THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *03/10/2017* **par :**

Harmish KHAMBHAITA

Human-aware space sharing and navigation for an interactive robot

JURY

Alberto SANFELIU	Professeur d'Université	Rapporteur
Anne SPALANZANI	Maître de Conférences	Rapporteur
Luis MERINO	Professeur Associé	Examinateur
Amit Kumar PANDEY	Head Principal Scientist	Examinateur
Thierry SIMEON	Directeur de Recherche	Examinateur
Rachid ALAMI	Directeur de Recherche	Directeur de thèse

École doctorale et spécialité :
MITT : Domaine STIC : Intelligence Artificielle
Unité de Recherche :
Laboratoire d'analyse et d'architecture des systèmes (LAAS)
Directeur de Thèse :
Rachid ALAMI
Rapporteurs :
Alberto SANFELIU et *Anne SPALANZANI*

Abstract

The methods of robotic movement planning have grown at an accelerated pace in recent years. The emphasis has mainly been on making robots more efficient, safer and react faster to unpredictable situations. As a result we are witnessing more and more service robots introduced in our everyday lives, especially in public places such as museums, shopping malls and airports. While a mobile service robot moves in a human environment, it leaves an innate effect on people about its demeanor. We do not see them as mere machines but as social agents and expect them to behave humanly by following societal norms and rules. This has created new challenges and opened new research avenues for designing robot control algorithms that deliver human-acceptable, legible and proactive robot behaviors.

This thesis proposes a optimization-based cooperative method for trajectory-planning and navigation with in-built social constraints for keeping robot motions safe, human-aware and predictable. The robot trajectory is dynamically and continuously adjusted to satisfy these social constraints. To do so, we treat the robot trajectory as an elastic band (a mathematical construct representing the robot path as a series of poses and time-difference between those poses) which can be deformed (both in space and time) by the optimization process to respect given constraints. Moreover, we also predict plausible human trajectories in the same operating area by treating human paths also as elastic bands. This scheme allows us to optimize the robot trajectories not only for the current moment but for the entire interaction that happens when humans and robot cross each other's paths. We carried out a set of experiments with canonical human-robot interactive situations that happen in our everyday lives such as crossing a hallway, passing through a door and intersecting paths on wide open spaces. The proposed cooperative planning method compares favorably against other stat-of-the-art human-aware navigation planning schemes.

We have augmented robot navigation behavior with synchronized and responsive movements of the robot head, making the robot look where it is going and occasionally diverting its gaze towards nearby people to acknowledge that robot will avoid any possible collision with them. At any given moment the robot weighs multiple criteria according to the social context and decides where it should turn its gaze. Through an online user study we have shown that such gazing mechanism effectively complements the navigation behavior and it improves legibility of the robot actions.

Finally, we have integrated our navigation scheme with a broader supervision system which can jointly generate normative robot behaviors such as approaching a person and adapting the robot speed according to a group of people who the robot guides in airports or museums.

Résumé

Les méthodes de planification de mouvements robotiques se sont développées à un rythme accéléré ces dernières années. L'accent a principalement été mis sur le fait de rendre les robots plus efficaces, plus sécurisés et plus rapides à réagir à des situations imprévisibles. En conséquence, nous assistons de plus en plus à l'introduction des robots de service dans notre vie quotidienne, en particulier dans les lieux publics tels que les musées, les centres commerciaux et les aéroports. Tandis qu'un robot de service mobile se déplace dans l'environnement humain, il est important de prendre en compte l'effet de son comportement sur les personnes qu'il croise ou avec lesquelles il interagit. Nous ne les voyons pas comme de simples machines, mais comme des agents sociaux et nous nous attendons à ce qu'ils se comportent de manière similaire à l'homme en suivant les normes sociétales comme des règles. Ceci a créé de nouveaux défis et a ouvert de nouvelles directions de recherche pour concevoir des algorithmes de commande de robot, qui fournissent des comportements de robot acceptables, lisibles et proactifs.

Cette thèse propose une méthode coopérative basée sur l'optimisation pour la planification de trajectoire et la navigation du robot avec des contraintes sociales intégrées pour assurer des mouvements de robots prudents, conscients de la présence de l'être humain et prévisibles. La trajectoire du robot est ajustée dynamiquement et continuellement pour satisfaire ces contraintes sociales. Pour ce faire, nous traitons la trajectoire du robot comme une bande élastique (une construction mathématique représentant la trajectoire du robot comme une série de positions et une différence de temps entre ces positions) qui peut être déformée (dans l'espace et dans le temps) par le processus d'optimisation pour respecter les contraintes données. De plus, le robot prédit aussi les trajectoires humaines plausibles dans la même zone d'exploitation en traitant les chemins humains aussi comme des bandes élastiques. Ce système nous permet d'optimiser les trajectoires des robots non seulement pour le moment présent, mais aussi pour l'interaction entière qui se produit lorsque les humains et les robots se croisent les uns les autres. Nous avons réalisé un ensemble d'expériences avec des situations interactives humains-robots qui se produisent dans la vie de tous les jours telles que traverser un couloir, passer par une porte et se croiser sur de grands espaces ouverts. La méthode de planification coopérative proposée se compare favorablement à d'autres schémas de planification de la navigation à la pointe de la technique.

Nous avons augmenté le comportement de navigation du robot avec un mouvement synchronisé et réactif de sa tête. Cela permet au robot de regarder où il va et occasionnellement de détourner son regard vers les personnes voisines pour montrer que le robot va éviter toute collision possible avec eux comme prévu par le planificateur. À tout moment, le robot pondère les multiples critères selon le contexte social et décide de ce vers quoi il devrait porter le regard. Grâce à une étude utilisateur en ligne, nous avons montré que ce mécanisme de regard complète

efficacement le comportement de navigation ce qui améliore la lisibilité des actions du robot.

Enfin, nous avons intégré notre schéma de navigation avec un système de supervision plus large qui peut générer conjointement des comportements du robot standard tel que l'approche d'une personne et l'adaptation de la vitesse du robot selon le groupe de personnes que le robot guide dans des scénarios d'aéroport ou de musée.

Contents

Chapter 1

Introduction

Contents

Meet *Spencer*. A strapping but gentle robot, recently recruited by KLM, is patiently waiting near Gate B18. The flight from Vilnius is about to land, albeit with a delay of half an hour from the scheduled time. While other KLM staff prepare for welcoming the arriving passengers, Spencer and its co-worker robots are recharging their batteries. It is really busy at Amsterdam airport Schiphol, chances are high that some transfer passengers will lose their way during the transfer. "'Spencer! kom hier alsjeblieft.", calls Suzanna from the gate-desk. Spencer quickly responds by looking at her, nods in agreement and starts approaching her. During the approach Spencer constantly monitors Susanna's posture and adjusts its own approaching trajectory such that it remains visible to her. Stopping in front of Susanna, Spencer greets her with "Hoi!". "Kun je de familie gaan naar Munchen begeleiden?" she asks. "Jazeker", Spencer acknowledges. It figures out that the next flight to Munich is leaving from gate C17 in 40 minutes and moves near the gate to make itself

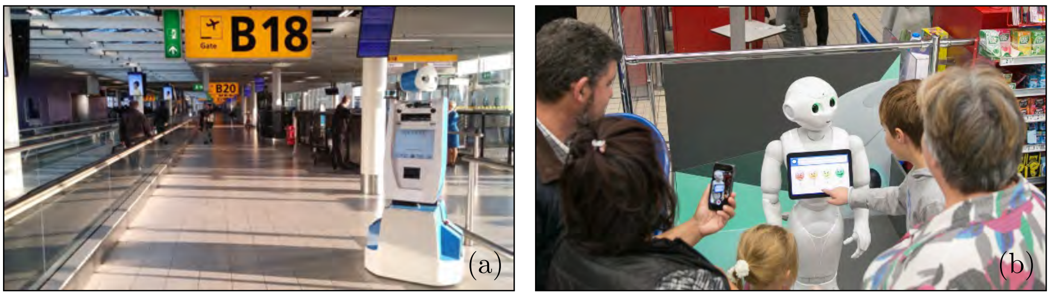


Figure 1.1 – Social service robots. (a) Spencer preparing to guide passengers at Amsterdam airport Schiphol. (b) Pepper helping and entertaining customers in a shopping mall. (Image credits: (a) www.spencer.eu, (b) www.mummer-project.eu)

noticeable to the arriving passengers. Apart from cordial facial expression and sturdy trunk, Spencer sports a portable tablet size touch-screen display on its front torso, which now shows a message for catching attention of Praßler family.

Soon after, the airplane lands and Praßler family rushes out of the gate, bit stressed about where to find the gate C17. Surprised by Spencer's welcome message, the family stops at the robot. Spencer greets the family and kindly offers them guidance inside the airport. It is primarily looking at mother, as it thinks she is the one who is going to decide. She gladly accepts the offer, so Spencer asks them to scan their boarding pass through the built-in bar-code reader. Meanwhile, Spencer had already calculated a quickest route to gate C17 which should take about 25 minutes. The family feels reassured as Spencer has already informed the KLM staff at gate C17 who will wait for them to reach.

Initially Spencer was moving fast, however, soon it sees that the children are not able to keep up with the pace so it slows down a bit. While smoothly marching through the airport swamped with rushing people, Spencer constantly keeps an eye on the Praßler family making sure they are still following. A long corridor connects pier-B with pier-C where temporarily parked trolley carts are narrowing down the moving space. Once Spencer sees a group of people running from the opposite side, it quickly moves behind the park trolleys to let them cross as they seem to have greater urgency. Praßler family thought it was a smart move by Spencer, who swiftly avoided a traffic-jam situation.

The youngest kid of the family wanted to visit toilets, so Spencer promptly made a small detour to the nearest lavatory. After the short break they resume their walk towards the departure gate. Entering the pier-C Spencer sees a long queue at an earlier gate. Rather then going around the queue, Spencer politely asks to make some space so they can pass through. Reaching timely at gate C17, the Praßler family thanks Spencer who has to soon start its next guiding mission.

At all times when moving Spencer remains very reactive to its surroundings, proactively responds to unexpected situations and makes sure it does not block anybody's way. Moreover, it takes extra care in explicitly making its movement readable by humans and showing where it is about to drive using its head movement. Passengers say that they can easily read where Spencer is going and whether Spencer has noticed them, thanks to its active and attentive face.

Now, Meet *Pepper*. A dwarfish but competent robot, new attraction at the Ideapark shopping mall in Lempäälä. Pepper has earned a designated working space near the welcome-desk. It's festival season and a small group of friends is looking for shoe-shops at the mall-map near the entrance. Pepper steps towards the group to tell them about current special-offers at the mall. The group makes some space for Pepper to join them and it subtly complies. Pepper helped the group to find the shop they were looking for and showed them way to reach there.

Later in the day an elderly woman asks Pepper "Hei, onko täällä apteekki?". "Tule mukaani!", Pepper aids her by taking her to the nearby pharmacy shop. Bustling shopping mall is not easy to traverse, so Pepper remains on rather straight path towards the pharmacy while constantly updating its speed to let the passersby placidly cross its path. At the same time, it also keeps a comforting distance from the elderly woman whom Pepper is helping.

Everyday Pepper helps dozens of visitors finding their way around the mall. Approaching and joining a group of people to start a conversation, negotiating its way around open spaces as well as densely crowded corridors are some of the daily activities of Pepper. Unlike Spencer, Pepper usually involves in short and specialized interactions. Still, like Spencer, Pepper shows proactive motion that follow social norms and uses modality of gaze to make itself better understandable to the people.

Both Spencer and Pepper are among the new breeds of *social service robots*. They go far beyond *safe*-motion paradigm, which, of course, is essential for any robot that is going to run in a human-centric environment. They studiously devise their movements to maximize human comfort and comprehension. Instead of just reacting to the circumstances, they show anticipatory and change-oriented behavior. Embedded social norms govern their demeanor while they co-exist and co-operate with humans in public places.

The level of autonomy, intelligence and sociability what we just described with airport and shopping mall scenarios is not a far-fetched dream. Robotics hardware research have already yielded compliant and fail-safe actuators as well as highly accurate and real-time human-motion capture sensors. On the other hand, studies of human social psychology have determined how pedestrian navigate in public places and what are the underlying physical factors that govern human interactions. What needed is novel planning techniques that adopt these sociological findings to design robot behaviors that resemble human awareness and proactivity. Therein we find the motivation for this thesis: **Attaining safe, natural and legible robot navigation behavior in public places along with multi-modal interaction.**

1.1 Human-Aware Navigation Planning

A **service robot** is “a robot that performs useful tasks for humans”, defines the International Organization for Standardization [ISO 2012]. It also defines a **personal service robot** as “a service robot for personal use by lay persons”. To do the *useful task* the robot needs to interact with *lay persons* in various ways, exchange information and respond to their actions. Which means, the robot should perceive its environment, create an action-plan and act, generally known as the iterative *sense-plan-act* method. While sensing of human posture and actions as well as robot’s own place in its surroundings is itself an independent domain of research, in this thesis our focus is on the planning and acting phases.

While planning the robot generally has an occupancy-grid-map of its operating environment. The path-planning algorithms plan a shortest path that is not passing through any grid-cell marked as *occupied*. And during the acting phase the robot calculates drive-velocities that keep the robot on previously computed path and satisfy any kinodynamic constraints imposed by the drive mechanism. Now, for human-aware planning the robot could incorporate human presence either during path planning by temporarily marking grid-cells around human position as occupied or shortly deviating from pre-computed path that make it avoid colliding with the human during the acting phase. For either case, there are certain challenges arise considering the dynamic and unpredictable nature of human motion, which we will summarize in the next section.

1.2 Challenges for Socially Compliant Navigation

The first challenge in human-aware navigation is to accurately detect and track humans. Nowadays, there are a variety of sensors used for detecting and classifying humans such as laser-scanners, color and depth sensing cameras, optical motion-capture systems and others. Data from different sensors is generally fused to calculate a person's position, orientation and velocity which is than passed to the navigation planning modules. In this thesis we have used off-the-shelf human tracking software, however, augmented them with velocity calculation subroutines. An associated challenge is to predict destinations for each tracked persons in the environment as well as to predict their short-term future trajectories. Precise prediction of immediate human trajectories can greatly aid the robot navigation planning system and lead to better path conflict management.

Almost all navigation planning schemes include either a cost-grid for representing the robot operating environment or use some constraint based optimal path planning algorithm, or both. Defining what makes up a good cost or constraint is the key challenge here. For claiming the robot navigation behavior as socially compliant, a navigation scheme has to capture the most pertinent criteria for calculating these costs or constraints. Furthermore, deciding which costs are most pertinent to which scenario is still an open question that begs for further investigation. Studies of pedestrian navigation behaviors are certainly useful in this regards and thus the criteria we use in our work for optimally planning the robot trajectories are directly inspired from the research on human locomotion in public places.

Robots that run in shared spaces need to respond fast to the changes in their surroundings. In the iterative planning process the last calculated trajectory could become invalidated as soon as the environment changes, particularly when the change is not what predicted by the robot. There are multiple reasons for this change: a previously unknown obstacle suddenly appears in front of the robot, a tracked-human does not move as predicted by the robot, robot encounters different terrain forcing it to move slowly, and so on. The challenge is then to re-plan the trajectory within some bounded time such that the robot behavior looks smooth and continuous. Therefore the trajectory planning algorithm must function with real-time constraints.

Another challenge for social compliance is producing the robot behaviors that people are easily and quickly able to understand, that is, making the robot behavior legible. Legibility is sometimes in conflict with geometrical optimality of the robot path, however, it improves fluency of human-robot cooperative motion and perceived safety by humans [Kruse 2014]. Hence, the robot navigation scheme has to strike a balance between legibility and path optimality.

While explicit signaling and cues to the actual robot intension increases legibility, it is different from predictability. Predictability requires an observer to envisage the expected future state, and it is determined by other factors than the robot internal state [Dragan 2013a] such as robot goal and its past movements. Sudden changes in robot direction or speed endangers predictability. Therefore, a robot has to consider applying hysteresis in its decision-making process.

In summary, here are the main challenges that one needs to solve while designing socially acceptable and desirable robot navigation behaviors.

- Precise perception of human motion.
- Finding most effective and pertinent costs/constraints for human-awareness.
- Timely response to environmental changes.
- Improving predictability and legibility of the robot behavior.

1.3 Contributions Summary

This goal of this thesis is to advance the human-aware navigation research by crafting cooperative and proactive navigation behavior for social robots that move and help humans in public places. We aim to close the gap between state-of-the art robot navigation frameworks and the hypothetical scenarios discussed earlier with Spencer and Pepper robots. As discussed in previous section, it requires to solve a number of challenges and advancements in several research areas including perception, mapping, localization, navigation, natural language processing, supervision and user-interface to fully synthesize such robot behaviors. Nevertheless, we stay focused on navigation planning, plan execution and motion planning of robot gaze.

Figure 1.2 shows a plausible architecture and different software modules required for human-aware navigation planning. In fact, this architecture is a generalization of the robot system architecture we have developed and used within the European projects SPENCER (Social situation-aware perception and action for cognitive robots)[1] and MuMMER (MultiModal Mall Entertainment Robot)[2].

The figure also shows contributions of this thesis with respect to affected software modules. Most of the contributions fall into the robot trajectory controller the path planning modules. With additional contributions to online behavior adaptation, learning normative behaviors and evaluation through user studies. We have also contributed to the aspects of simulation and visualization of the robot behavior as supportive contributions.

[1] http://spencer.eu/

[2] http://mummer-project.eu/

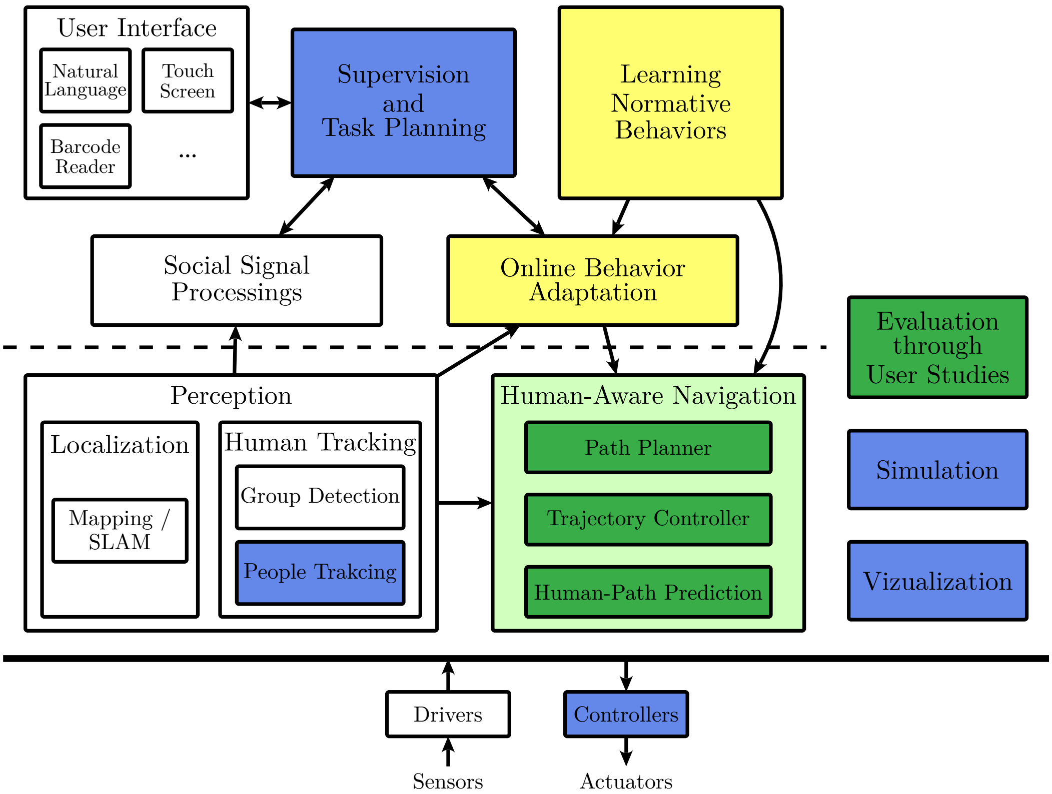


Figure 1.2 – A software architecture for human-aware navigation. The software modules shown in green are where this thesis makes major contributions. Modules with minor contributions are shown in yellow and modules in blue are the software components which we improved to support main contributions. The dashed line separates information flow. The communication is continuous in terms of software messages for the components below the dashed line, where the components above this line generally communicates using service requests.

1.3.1 Contributions to Navigation Planning

This thesis features two main contributions in navigation planning. First is modeling and implementing robot navigation in human environments as a cooperative activity. And second is the refined analysis of the constraints and criteria for such an activity.

We propose a cooperative and proactive navigation framework using an optimal planning scheme. The framework composes a set of mathematical constraints that model safety for nearby humans, social conventions for path and velocity planning as well as legibility of robot behavior along with usual kinematic constrains required by the robot drive mechanism. The fundamental idea of this framework is to combine planning of robot trajectories and prediction of plausible human trajectories within the same optimization framework while utilizing the same environmental map that robot perceives through its sensors. Thus our planning scheme incorporates estimation of others' behavior within the planning step. With this design the framework is able to produce robot behavior that is similar to what humans would do in identical situations.

We have introduced *time-to-collision* and *directional* constraints in this framework in addition to the typical *safety* constraints. We argue that with these constraints our planning scheme produces socially compliant robot behavior in a multitude of tasks that require tight human-robot co-operation such as traversing through hallways, passing through doors, crossing paths with humans as well as navigating in large semi-crowded areas. We also compare results with other state-of-the-art human-aware planning schemes and demonstrate the advantages of our framework in terms of quality of the generated trajectories and sociability of the robot. Additionally, we have carried-out a user study that shows how the *directional* constraint improves the legibility and perceived comfort by the participants in a human-robot path crossing scenario.

We have developed and extensively used a pedestrian movement simulator to test and validate navigation planning algorithms. Numerous software modules that support the main contributions were also developed during the course of this thesis such as laser-scanner data filters, motor drivers for Spencer head mechanism, improvements and re-implementation of some of the previous work that is compatible to the middleware used by Spencer and Pepper robots, low-level subroutines for collision checking and others.

1.3.2 Contributions to Normative Social Behavior

Acting socially is not enough, a robot also need to communicate its activity to the people in its vicinity, which affects the legibility of the robot behavior. For improving legibility with its gaze, this thesis proposes a novel method to control the head movements while the robot is navigating. We treat the gaze-control problem as multi-criteria decision-making problem and solve it with analytic hierarchy process. A video-based user-study has validated the effectiveness of the proposed head-control procedure.

A method to dynamically adapt the robot speed according to the walking speed of a group of people for a guide robot and a machine learning based algorithm to approach humans in a socially appropriate way are among the minor contributions of this thesis.

1.3.3 Publications

The work presented in this thesis is partly based on following prior publications.

Harmish Khambhaita and Rachid Alami. “Viewing Robot Navigation in Human Environment as a Cooperative Activity”. In: *Proc. International Symposium on Robotics Research (ISRR)*. Dec. 11–14, 2017.

Harmish Khambhaita and Rachid Alami. "Assessing the Social Criteria for Human-Robot Collaborative Navigation: A Comparison of Human-Aware Navigation Planners". In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN).* Aug. 28–Sept. 1, 2017.

Harmish Khambhaita and Rachid Alami. "A Human-Robot Cooperative Navigation Planner". In: *Proc. Companion of the ACM/IEEE International Conference on Human-Robot Interaction (HRI).* Mar. 6–9, 2017, pp. 161–162. ISBN: 978-1-4503-4885-0.

Harmish Khambhaita, Jorge Rios-Martinez, and Rachid Alami. "Head-Body Motion Coordination for Human Aware Robot Navigation". In: *Proc. International Workshop on Human-Friendlly Robotics (HFR).* HFR. Sept. 29–30, 2016.

Omar A. Islas Ramírez, Harmish Khambhaita, Raja Chatila, Mohamed Chetouani, and Rachid Alami. "Robots Learning How and Where to Approach People". In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN).* Aug. 2016, pp. 347–353.

Michelangelo Fiore, Harmish Khambhaita, Grégoire Milliez, and Rachid Alami. "An Adaptive and Proactive Human-Aware Robot Guide". In: *Proc. Seventh International Conference on Social Robotics (ICSR).* Oct. 26–30, 2015, pp. 194–203.

Rudolph Triebel, Kai Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, Daniel Cremers, Vanessa Evers, Michelangelo Fiore, Hayley Hung, Omar A. Islas Ramírez, Michiel Joosse, Harmish Khambhaita, Tomasz Kucner, Bastian Leibe, Achim J. Lilienthal, Timm Linder, Manja Lohse, Martin Magnusson, Billy Okal, Luigi Palmieri, Umer Rafi, Marieke van Rooij, and Lu Zhang. "SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports". In: *Proc. Conference on Field and Service Robotics (FSR).* Ed. by S. David Wettergreen and D. Timothy Barfoot. June 24–26, 2015, pp. 607–622. ISBN: 978-3-319-27702-8.

Séverin Lemaignan, Marc Hanheide, Michael Karg, Harmish Khambhaita, Lars Kunze, Florian Lier, Ingo Lütkebohle, and Grégoire Milliez. "Simulation and HRI Recent Perspectives with the MORSE Simulator". In: *Proc. International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR).* Ed. by Davide Brugali, Jan F. Broenink, Torsten Kroeger, and Bruce A. MacDonald. Oct. 20–23, 2014, pp. 13–24. ISBN: 978-3-319-11900-7.

Thibault Kruse, Alexandra Kirsch, Harmish Khambhaita, and Rachid Alami. "Evaluating Directional Cost Models in Navigation". In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI).* HRI. Mar. 3–6, 2014, pp. 350–357.

1.3.4 Dissertation Outline

The next chapter (**Chapter 2**) will introduce generic concepts of human-aware navigation and present an extensive literature review for the same. This chapter will also introduce a planning framework and additions software components that we have developed during the course of the this thesis to support the main contributions.

Chapter 4 will present the prominent contribution of the thesis, namely the cooperative and reactive navigation planning system. Tests in real environments with real robots in everyday path crossing situations help assess its capabilities and confirm its effectiveness. We also compare the proposed cooperative planner with other state-of-the-art human-aware navigation systems in canonical interactive situations. Moreover, we will present the results of a user-study that encouraged the use of *directional* constraint for cooperative planner.

In **Chapter 5** we give details of robot gaze control method that augments navigation planning system in improving legibility of the overall robot behavior. We discuss the reasoning behind proposing the gaze control as multi-criteria decision-making problem and how given social context can be used to weigh these criteria. This chapter provides details of two behavioral functions *look-at-path* and *glance-at-human* and synchronization of these functions for fluent and competent gaze behavior of a robot during navigation.

Chapter 6 focuses on techniques for producing normative robot behaviors. Specifically, we introduce a method for adapting the robot speed during collaborative actions such as guiding a person in large indoor environments as well as a machine learning based method for approaching people in socially acceptable fashion.

Chapter 7 concludes the thesis with brief discussion on most important outcomes and gives an outlook on the future work.

Chapter 2

Related Work

Contents

Our interest lies in creating interactive robots that seamlessly share their workspaces with humans. When waking among people a robot has to collaborate its actions while observing social conventions, and it is suppose to offer multi-modal interaction. As such, this thesis draws on the work from the fields of human-robot interaction, motion planning, optimization to proxemics theory, social psychology and pedestrian dynamics. This chapter introduces most relevant concepts and related work.

We will first summarize algorithms and planning methods for robot navigation. Following that, we summarize models for describing how people move and cooperate in shard spaces. Third section of this chapter will combine ideas from robot navigation and pedestrian models to introduce various human-aware planning techniques proposed in the literature.

2.1 Robot Navigation

Planning for navigating requires choosing next motor commands which move the robot closer to its goal in an iterative process. As characterized in the airport navigation scenario in Chapter 1, for moving between two locations the robot Spencer has to come up with some path to follow in the airport map and it has to avoid any obstacles that it encounters while moving on this path. Thus, the iterative process of planning is usually divided in two stages: *path planning* and *obstacle avoidance*. In other words, the robot has to calculate a path for its long-term goal and plan for short-term deviations from the long-term path to avoid colliding with previously unknown obstacles.

The well-known and versatile framework `move_base` [MarderEppstein][1] of the robot operating system (ROS)[2] for navigation also implements this two-stage strategy. It differentiates between *global planning* for generating a path from the start position to the goal position in arbitrarily large environment, and *local planning* for avoiding immediate obstacles by locally modifying the robot trajectory. We have extensively used this framework as it follows modular software development approach that allows us to gradually replace planning modules and test human-aware navigation planning scheme. In the next sections we give a rundown of widely used path searching and obstacle avoidance algorithms.

2.1.1 Path Search

At both stages of planning discussed above, the robot has to create a model of its operating environment. Standard practice for path planning is to create an occupancy-grid map that separate traversable and non-traversable areas and find a traversable path from start to goal using search algorithms such as A^* [Hart 1968] or Dijkstra's algorithm [Dijkstra 1959]. For the obstacle avoidance part, the robot augments the occupancy map with current sensor observations to plan immediate trajectory that makes the robot move both near to the previously calculated path and far from the detected obstacles. Here the planning process also involves searching a collision-less trajectory that for a short time deviates the robot from its path, eventually moving the robot back to its original path. There are several other algorithms proposed in the literature that overcome some of the difficulties with A^*, such as R^* [Likhachev 2008] which scales better for larger search spaces, D^* Lite [Koenig 2002] and $LLS-LRTA^*$ [Koenig 2009] that can plan paths in real-time, RTR^* [Cannon 2012] for dynamic situations, and others.

Another method for path planning, referred as sampling based planners, find possible paths by randomly adding points to a tree of configurations. Rapidly exploring random tree (RRT) [LaValle 2006] and probabilistic roadmaps planner (PRM) [Kavraki 1996] are sampling based popular algorithms to efficiently search the configuration space to find a path between starting and goal configurations. RRT is generally considered as single query algorithms, as it grows the tree from the start point until one of its branch hit the goal point. Whereas PRM is referred to as multi query algorithm, as it create the tree by randomly sampling points in the space and connecting them by conditioning on robot kinematics. The advantage of RRT over A^* is in the planning time, RRT algorithms are significantly faster than A^*-like algorithms. Furthermore, RRT allows to use other distance measure than the Manhattan or Euclidean distance between two configurations. Since the

[1] `http://wiki.ros.org/move_base`
[2] `http://www.ros.org/`

introduction of RRT, it quickly became popular among motion planning community and several researchers have provided extensions to it. The RRT^* [Karaman 2011] and $LQR - RRT^*$ [Perez 2012] variant are asymptotically optimal extensions to the standard RRT algorithm.

Another variant RRT^X [Otte 2015] is able to quickly re-plan a path by refining and repairing the same search-graph, thus it can also be used for dynamic obstacle avoidance. Thus the differentiation between path planning and obstacle avoidance algorithms is not a strict one, however obstacle avoidance algorithms mostly works on sub-space around the robot to keep the iterative process fast enough and make the overall robot motion smoother.

What important is the trajectory calculated by the planner should satisfy kinodynamic requirements of the robot motion controller. The kinodynamic problem involves finding a trajectory from one point with some initial velocity to the goal point with some end velocity in minimum-time. [Donald 1993] provides one of the first approximation algorithms to solve the kinodynamic motion planning problem. Their method reduces the motion planning problem to a graph search problem where the graph containing point-velocity tuples. They consider bounded velocity, bounded acceleration and obstacles as constraints. [Webb 2013] gives a kinodynamic variant of RRT^* algorithm and [Palmieri 2014] discusses an extend function specifically designed to suit kinodynamics of a wheeled robot.

2.1.2 Obstacle Avoidance

Typically the planning frameworks perform continuous re-planning of the robot path to cope up with dynamic situations. This means, the re-planning treats each situation as static snapshot of what is unfolding in the world and plans robot paths only for that particular snapshot, without any consideration of what might change in the very next snapshot. This scheme sometime suffers from issues of confusing behaviors where the robot path oscillates around moving humans [Kruse 2014] and leads the robot to entangled situations (figure 3.8 shows an example of such erratic behavior). Furthermore, re-planning the whole path from start (or current) position of the robot to the goal position might be computationally expensive often not necessary.

Several obstacle avoidance methods proposed in the literature tries to alleviate computational cost by reducing the area in which they plan the robot path. Therefore, these methods take the global path computed by a path planning algorithm as input and only modify small part of this path to avoid any previously unknown obstacles. Obstacle avoidance methods also work iteratively, albeit they are designed to run at higher frequency such that overall robot behavior remain reactive enough. The elastic bands [Quinlan 1993] and the dynamic window approach [Fox 1997] are examples of well-known obstacles avoidance algorithms used for mobile robot navigation.

Additionally, the obstacle avoidance methods also takes care that the trajectories they generate or the velocity commands they apply to the robot motor controller

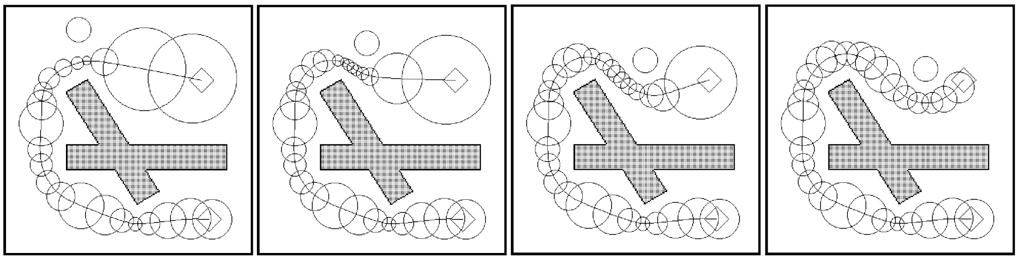


Figure 2.1 – The elastic band approach for obstacle avoidance. As an obstacle moves the elastic band changes to balance the repulsive force of the obstacle and contractive internal force. (Image modified from [Quinlan 1995])

are within the limits of the robot drive mechanism. Therefore, these methods are also knows as *local planners* in the `move_base` framework.

Elastic Band

Elastic band is a well-studied approach for dynamic obstacle avoidance that only locally modifies the robot path to keep a safe distance from previously unknown obstacles [Quinlan 1993]. The original idea of elastic band approach is to represent the robot path as a series of "bubbles", whose radii depend on how far the center of the bubble is from an obstacle. As an obstacle approaches the robot, it applies a repulsive force to these bubbles and cause them move away from them. This process in turn elastically modifies the robot path to balance the repulsive force of the obstacle and internal contraction force of the bubble-band. The iterative process of elastic band can add and remove these bubbles to keep the whole path fully covered by overlapping bubbles. Figure 2.1 shows how these bubbles and the robot path changes as an obstacle approaches the robot.

However, the modified path often does not satisfy the kinodynamic constraints of the robot. Therefore, a general scheme is to use a controller module that takes the output path (elastic band) and generates feasible trajectories that the robot can follow [Khatib 1997].

Recent proposal of *timed elastic band* evades this problem by explicitly considering temporal information [Rösmann 2012; Rösmann 2013]. Thus, it augmented with a series of time-difference values between each successive poses, instead of a purely geometric path. This timed elastic band can deform both in its shape, that means changing the 2D positions of the poses, or it can deform in time, that means adjusting the robot velocity between two consecutive poses. *Timed elastic band* makes it easy to take kinodynamic and nonholonomic constraints into account, formalizing the optimization problems as a non-linear least squares problem. This timed elastic band is then optimized by the graph optimization software g^{2}o [Kümmerle 2011]. This approach directly gives kinodynamically possible trajectories that a robot can start executing. Figure 2.2 shows example trajectories generated by this approach. In their recent work, the authors have extended the approach to optimize the timed

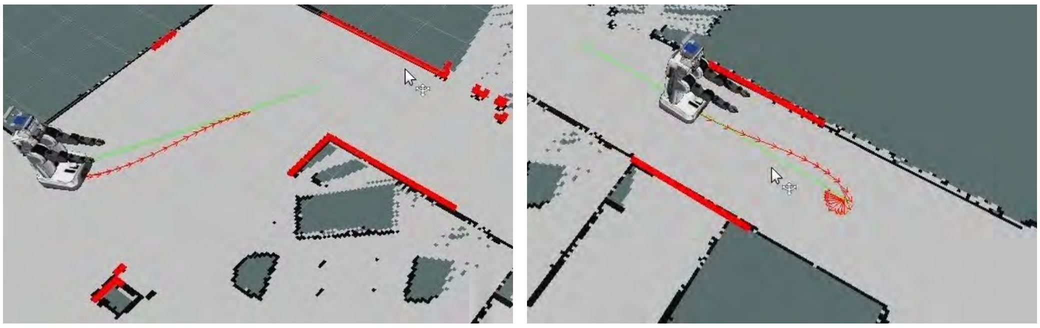


Figure 2.2 – Paths generated by the timed elastic band approach for obstacle avoidance. This approach modifies the global path such that it avoid obstacles and it gives kinodynamically possible trajectories for the robot driving mechanism.

elastic band for multiple homotopy classes [Rösmann 2015].

Dynamic Window Approach

The dynamic window approach (DWA) [Fox 1997] is another widely used approach for dynamic obstacle avoidance. The dynamic window represents admissible translational and rotational velocities for a robot when it is moving in an environment with obstacles. A velocity (linear and angular) is chosen from this dynamic window that maximizes certain objective function. This objective function includes a measure of progress towards a goal location, the forward velocity of the robot and distance to the next obstacle in the environment.

This approach is able to generate smooth trajectories in multitude of situations. As a result, an implementation of DWA based local planner comes pre-installed with the `move_base` framework. We have adopted this approach for test with the Spencer robot (for details see Section 3.1) with a velocity adaptation approach that we tested being effective in human-robot path crossing scenario with a user study (for details see Section 3.2).

The trajectory rollout algorithm [Gerkey 2008] is similar and widely used obstacle avoidance algorithm, since it is one of the pre-installed algorithms available in the `move_base` framework. It uses local predictive controller that encodes robot's kinodynamic constraint, including finite acceleration and non-zero turning radius. The controller samples control points from the vehicle's control space and predicts possible trajectories with discrete-time simulation up to the planning horizon. The trajectories calculated from sampled control points are evaluated with a multi-objective criteria that encodes several component of the planner's cost structure. The best trajectory is chosen to move the robot at each iteration, where the iterative process runs at about 10 hertz frequency. The approach is different from the DWA in which search space it operates, trajectory rollout algorithm searches in the space of feasible controls instead of searching the space of feasible trajectories. The predicted trajectory is a sequence of 5-dimensional $(x, y, \theta, \dot{x}, \dot{y})$ states with a discrete-time approximation of the vehicle's dynamics.

Temporal Planning

Although here we are discussing as part of obstacle avoidance method, temporal planning is more a technique for scheduling. Temporal planning can augment the path panning to help avoid unknown obstacles by maintaining multiple path possibilities for moving from start to goal positions, and invalidating some of these possible path configurations if an obstacle obstructs them. [Kushleyev 2009] introduce a temporal planning algorithm that first predicts how a detected obstacle is going to move with some uncertainly for certain time horizon, thus being a local planning technique. The prediction information helps to create and maintain a graph structure of motion primitives (called time-bounded lattice) for given time horizon. This technique reduces oscillations in robot behaviors and avoids robot being stuck in some local minima due to obstacles. [Phillips 2011] is an improved variant of this approach that aims to reduce the search space by using *safe and unsafe time intervals* (a time period for configurations without or with collision).

Another approach is to divide search space into regions based on possibility of collision [Vemula 2016]. The time dimension is only considered for regions where collisions may occur, and plan in low-dimension state space elsewhere. The time scaled collision cone based approach [Gopalakrishnan 2014] aims to reduce the computational cost by only computing homotopic trajectories based on predicted trajectories of obstacles and evaluating a symbolic formulae over it.

The time horizon for temporal planning should at least allow robot to smoothly avid the obstacle. Increasing the time horizon, however, leads to combinatorial explosion of search space and often it becomes unsuitable for real-time solutions.

Multi-Robot Collison Avoidance

When there are multiple robots in the environment, they need to avoid colliding with each other as well. Reciprocal n-body collision avoidance [Berg 2009] is a method to simultaneously avoid obstacles of multiple robots. One can consider humans as just another robot and act using this method, assuming that humans will also follow similar scheme. However, such a scheme may not be suitable for socially acceptable navigation since it assumes that each agent has perfect sending and is able to infer the exact shape, position and velocity of other agents in the environment, which is not usually the case due to uncertainty of human motions.

When a mobile robot encounters people, with a simple planning scheme, it would treat them as any other obstacle. This mostly leads to asocial robot behavior, not what people expect or demand from a social service robot. For example equipped with a naïve obstacle avoidance algorithm the robot can pass uncomfortably near to humans, it neither follows social norms such as passing others by moving to the right nor does it care if its actions are legible. Therefore, we need to embed the social compliance directly in the navigation planning architecture. In this thesis we have not only developed novel techniques for socially compliant trajectory planning but also addressed the issue of legibility of robot behavior by supplementary gaze behavior. But before we can say that our robot navigates in socially correct way, we ought to understand how pedestrians move in public places.

2.2 Modelling Pedestrian Motion

Pedestrian behaviors, nonverbal communication, human psychology, crowd behaviors are among the fields of research that can help understand what affects the decision-making process in humans while they are navigating in social environments. In this section we look into some of these research that is particularly relevant to human-aware robot navigation.

2.2.1 Proxemics

People, when interacting, usually respect the *personal space* of each other. The theory of *proxemics* introduced by [Hall 1966], formalizes this personal space as a region surrounding a person from about 0.45 to 1.2 meters. People often feel uncomfortable or threatened when others, especially a robot, invade personal space.

The two experiments of [Torta 2013] verifies the comfortable distance for a robot to approach a person. In the first experiment a person is sitting on a chair and the robot approaches the person from different directions (with angles $-75°$, $-30°$, $0°$, $30°$ and $75°$), and the person stops the robot when it reaches a comfortable distance. The person was standing for the second experiment while the robot approached at same angles as the first experiment. They concluded that for approaching a person the distance preferred by humans is larger than the communication distance suggested by [Hall 1966], being 182 cm while the person is sitting and 173 cm while the person is standing. The authors further defined the best robot distance for approaching a person as a quadratic function depending on the distance and direction, for both sitting and standing postures. [Sardar 2012] performed a similar user study where a robot invaded a standing person's personal space by approaching them at two different speeds and compared them against another person invading the personal space. They observed that the participants displayed more compensatory behavior when a robot invaded their personal space. These results suggest that in HRI we might need to consider larger safety distances compared to human-human interaction scenarios and be aware of the fact that humans will respond differently when a robot invades their personal space compared to other human being.

The survey of [RiosMartinez 2014] discusses other social spaces configurations such as information processing space (IPS): space in which all objects are considered as potential obstacles when a pedestrian is planning their paths, activity space: a space defined by current human activity such as taking a photograph and affordance space: the space in which humans may move or stay to interact with an object (for example information screens). For a robot to behave in socially normative manner, it also need to detect and avid such spaces around humans. This survey further discusses the concepts of O-space and F-formations [Kendon 1990]. These concepts are useful for describing interactive configuration of multiple persons, such as taking in a group, joining a group, guiding.

2.2.2 Social Forces

To describe pedestrian movements [Helbing 1995] have proposed the *social force model* (SFM). The social force model considers a person as a particle moving in open space where several forces acting upon them decide their next movement. These forces include an attractive force towards a goal, a repulsive force from other pedestrians and a third force describing acceleration towards their desired velocity. Additionally, the social force model have shown to describe also certain self-organizing crowd dynamics such as lane formation or effects at bottleneck [Helbing 2011]. Some extension to the social force model are also proposed in literature, CP-SFM is an extension that explicitly predicts time and place of the next collision to avoid it [Zanlungo 2011].

The social force model was recently refuted in favor of power law relation between pedestrian movements and their projected time to a potential future collision [Karamouzas 2014]. In this paper, the authors use statistical analysis to measure the interaction energy between pedestrians. They apply this analysis to two large datasets of pedestrian motions, one in outdoor environment (an area of about 8 to 10 meter) with 1146 trajectories and second dataset of a bottleneck situations (of 1 and 2.5 meter) with 354 trajectories of pedestrians. For statistical analysis they define a pair distribution function as the observed probability for two pedestrian to have relative separation divided by the expected probability density for two non-interacting pedestrians to have the same separation. Their finding suggest that this pair distribution function can be accurately parametrized by a single variable that describes how imminent potentially upcoming collisions are, that is, *time-to-collision*.

In either case, social force model or time-to-collision, there seems to exist simple mathematical model to fully describe the causal links between continuous primitive-movement-actions of people in public spaces. It is, however, advisable to use different collision avoidance methods for individual level and crowd level motions [Park 2013].

Another way to understand spatial human motions is qualitative trajectory calculus [Van de Weghe 2004]. [Hanheide 2012] augments this framework by including direction of moving agents. They use signs of movement of two agents with respect to each other and with respect to the reference line which passes through both agents in their representation of quantitative relations. Using these relations as state for a Markov Decision Process (MDP), they were able to model path crossing behavior of humans in a corridor.

While planner motions are explicit form of interaction, humans also use implicit motion through their gaze and gestures to convey auxiliary information about their future actions [Kruse 2015]. Gaze or head direction is predominantly used to convey walking direction [Castiello 2003; Kitazawa 2010]. This modality of gaze has been largely ignored in human-aware navigation literature. On the other hand we are witnessing more and more mobile robots equipped with a head or gaze module, which will force us to find useful ways of integrating control of robot gaze within the robot navigation frameworks.

If we look carefully, the research on human movements has two categories: how we behave while being stationary and how we navigate. Theory of proxemics [Hall 1966] and F-formations [Kendon 1990] fall into the category of understanding static behavior, while the social force model [Helbing 1995] and other research on how humans avoid collision belongs to the category of dynamic behavior. In Section 3.1 we also use this observation to treat moving and static humans differently in our human-aware navigation framework.

2.3 Integrating Human-Awareness

It is clear that social navigation-planning schemes have to treat humans differently than ordinary obstacles. The research on pedestrian movements is a starting point for designing socially acceptable behaviors for a mobile robot. At the very least a mobile robot needs to keep a safe distance from people and avoid entering into their personal space. Consequently some of the earlier work in human-aware navigation mainly concentrates on generating paths that keep a safe distance from humans. To do so, the theory of proxemics [Hall 1966] have been widely adopted by robotics researchers.

The human-aware navigation planner (HANP) described by [Sisbot 2007] is one of the first attempts to purposely integrate human-awareness within the navigation-planning algorithm. It already provide safe paths considering not only proxemics distance but also other social criteria like visibility and hidden zones around static humans in the environment. The proposed algorithm inflates area around humans with *artificial costs* in the robot's metal model of the environment, that is the occupancy-grid map. Specifically, it generates *social costs* in a grid-map structure around humans to facilitate A^*-like graph-search algorithms [Hart 1968] to find paths that minimize such social costs. This grid-map is result of a weighted-sum of mainly three component functions. Reasoning behind these component functions being *safety*: planning the robot path that avoids invading personal space of humans as defined by the proxemics theory, *visibility*: it is better for the robot to make itself visible to the human for most part of their trajectory, and *hidden zones*: robot should not move very near behind, for example, a wall such that it can shock humans. This strategy causes the path planning algorithm to plan paths that are farther from humans compared to other obstacles. Since these social costs are added around instantaneous position of humans in the map, for dynamic situation their strategy is to continuously re-plan the paths. The robot navigation planner we discuss in the next chapters also make use of the safety criteria while planning the robot trajectories.

The survey of [Kruse 2013] describes human-aware navigation as the intersection between research on human robot interaction (HRI) and robot motion planning. The

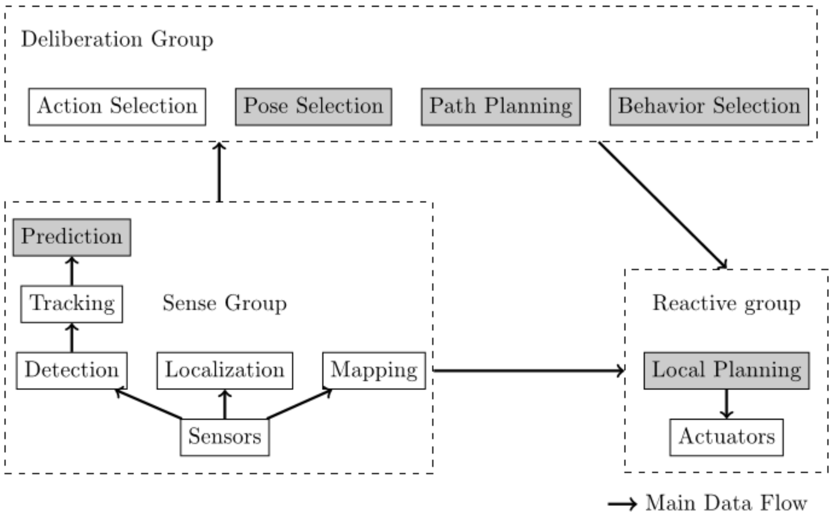


Figure 2.3 – A possible architecture showing modules involved in human-aware navigation planning. The arrows show main data flows, many more connections are possible. (Image taken from [Kruse 2013])

motion planning algorithms deal with finding a shortest path that allow the robot to move from one configuration to another without colliding with obstacles given a workspace and obstacles in the workspace. While in HRI, the assumption is that the shortest or most energy-efficient path is not necessarily the most desirable. Instead the intention is to find a path that is also sufficiently safe, comfortable, natural and legible to humans in the area. The authors classified literature that deal with human-aware motion in mainly two categories, first approach is to represent desirable path attributes (like proxemics of [Hall 1966]) as cost-functions for global path-planning, another approach is to interleave prediction of human motion and planning of robot motion to identify potentials for joint efforts in collision avoidance. They have identified functionalities that are relevant as research items in human-aware navigation, shown in figure 2.3.

As the figure 2.3 shows, the literature in human-aware navigation suggest to modify one of the four functionalities of this general navigation framework: pose selection, path planning, behavior selection and local planning. A complete approach also needs to make a set of predictions for each alternative decision that the robot can make as its decision will directly influence human's decision. Therefore, the authors classify prediction as also one of the core functionalities of a human-aware navigation planning. An important observation made by the authors is that in HRI the shortest or most energy-efficient path is not necessarily the most desirable, instead HRI aims to find a path that is safe, sufficiently comfortable, natural and legible.

A way to deal with the dynamics of human environments is to use some methods that can predict probable human paths for short-term future and already take it into account while planning the robot path. The reliability of any prediction method quickly degrades in time, and there are necessarily many occasions of unexpected human motions even more so when a robot moves nearby. Therefore, even with rather reliable prediction, re-planning during execution is inevitable. It requires to develop planning methods that scale well.

2.3.1 Robot as a Social Agent

An acclaimed approach used by the human-aware robotics researchers is *social force model* [Helbing 1995]: a method to describe crowd dynamics. These methods predict a class of homotopically distinct trajectories for humans and design a planner that, from human demonstrations, can learn navigation policies for robot to move on human-like trajectories [Kuderer 2012]. However, it may require re-learning of the model parameters for specialized or constrained situations such as crossing long corridors or passing through a door. The robot navigation planners explained in [Ferrer 2013] uses the *social force model* to cope with uncertain human motions by planning the robot paths itself as if the robot is an agent that also follows the social force model.

A further improvement that uses an *extended social force model* with a sampling based planner to make the robot motions proactive while respecting kinodynamic constraints of the robot [Ferrer 2014b; Ferrer 2015]. In this approach, the robot acts as an agent following the social force model, which computes kinodynamically feasible acceleration for the robot at each iteration (with some damping factor to reduce oscillations). The scheme for computing acceleration using RRT works as following:

- Several goals are randomly sampled on a circle around the robot whose radius defines the local planning horizon.

- Position of these random goals on the circle depends on the density of humans around the robot. When no human is present the sampling is biased towards the global path direction.

- Then the RRT tree propagates towards the sampled random goals, human positions are also propagated according to the SFM.

- The best path is the one that results into least amount of total *social work* carried out by the humans and robot as well as the linear and angular distance from the local goal for every added vertex. The *social work* takes into account the steering force and social-forces due to nearby humans and obstacles.

Thus, this approach interweaves prediction of human positions and planning of robot path in to the planning process which, we believe, is an essential design choice for any good human-aware navigation planner. Figure 2.4 shows operating of extended social force model based planner of [Ferrer 2014b] with a simulated PR2 robot and a person.

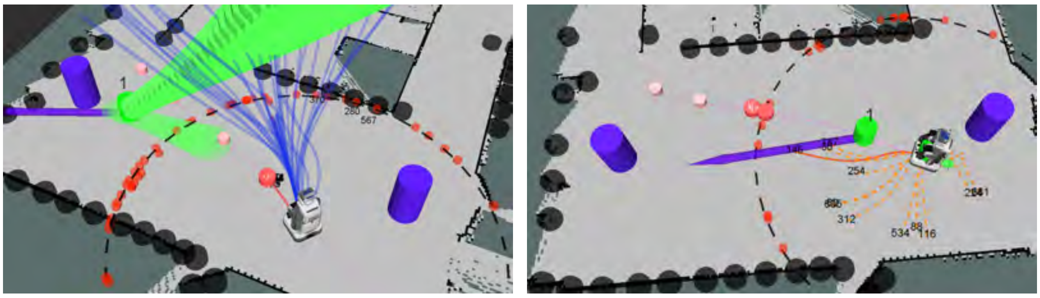


Figure 2.4 – Paths generated by the extended social force model based local planner. Green cylinder shows human positions and dashed circle around the robot depict the planning horizon for the local planner. Image on the left also shows predicted human positions, where time is plotted on the vertical axis and uncertainty in human motion is represented by the radius of the flat green cylinders. Image on the right shows different trajectories of RRT tree in dashed orange lines, trajectory with solid red line is what robot eventually selects to follow.

[Shiomi 2014] have used the CP-SFM [Zanlungo 2011] to plan robot motion in crowded environments. The authors also claim that by adjusting parameters of their algorithm the system can be collision free in the pedestrian density of interest and provides noiseless trajectories. They have tested the usefulness of the proposed algorithm with two filed trials, first where human moves from a to b and the robot moved from b to a in a straight line of about 20 meter. In the second field test, robot moved from point a to b and then b to a (about 20 meter in distance) autonomously, and the people can move freely in this areas. In both cases the robot behavior was judged, when human passed by in 5 meter range, unsafe if human have to move quickly and safe otherwise. They compared their proposed method against time varying dynamic window approach [Seder 2007], where the proposed method turns out significantly more acceptable by the participants.

The planning schemes based on the *social force model* works fine in large or open spaces where the robot have enough latitude to move away from predicted human paths. However, as we will see in Section 4.2, it performs rather poorly in constrained situations.

In a similar approach by [Bordallo 2015], the robot navigation planner uses counterfactual reasoning to calculate probabilities over a possible set of navigational goals. Using such probability set, this approach predicts human motion towards most probable goal and generates locally optimal motions for multiple robots.

The *RiskRRT* approach [RiosMartinez 2011; Spalanzani 2012] integrates probable collision risk while exploring the RRT tree. For assessing the risk of collision it makes predictions on future path of an obstacle (generally a human) using Gaussian process models [Tay 2007]. Since the robot cannot know when and where will it encounter a human, the robot maintains an RRT tree as it moves in the environment. When the robot detects a human and predicts its trajectory, it can quickly switch a new branch of the RRT tree such that any collision with the detected human is

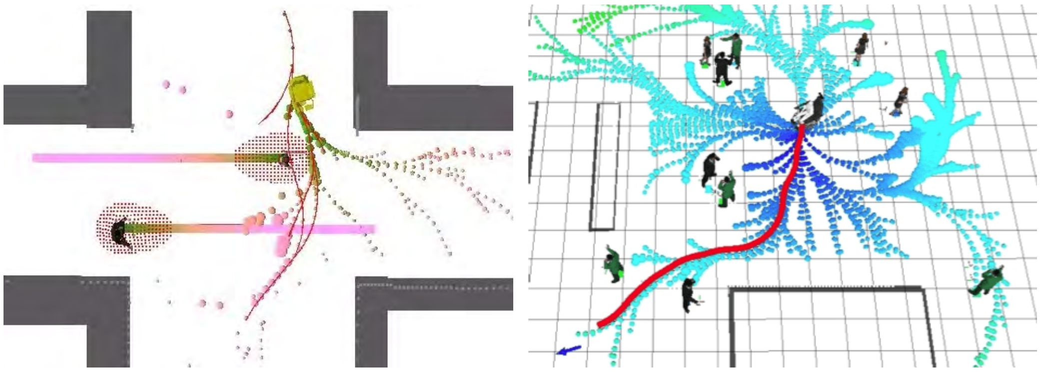


Figure 2.5 – Paths generated by the *RiskRRT* based local planner.

avoided. Additional advantage of this method is that it can also incorporate the social costs like the notion of *risk of disturbance* in the risk matrix used for growing the RRT tree. In a way this method is similar to the temporal planning methods we discussed in Section 2.1.2.

The *RiskRRT* based planning have shown to effectively navigate around humans in simulation as well as real world settings [Garzón 2013]. Figure 2.5 shows how *RiskRRT* generates a tree for social navigation. Although being effective in populated environments, this method may also make the robot oscillate based on the planning horizon and obstacles at the boundary of the planning horizon.

Concerning the ability for the robot to plan not only for itself but also for its human partner, [Alili 2009; Lallement 2014] have developed a task planner called HATP planner (*Human Aware Task Planner*). The HATP planning framework extends the traditional hierarchical task network planning domain representation and semantics by making them more suitable to produce plans which involve humans and robots acting together toward a joint goal. HATP is used by the robot to produce human-robot *shared plans* which are then used to anticipate human action, to suggest a course of action to humans, or possibly to ask help from the human if needed.

This effectively enriches the interaction capabilities of the robot by providing the system with what is in essence a prediction of the human behavior. This prediction is also used by the robot execution controller to monitor the engagement of the human partner during plan achievement. Another key property is to produce plans that would be possibly preferred by the human partner. For instance, HATP includes cost-based plan selection as well as mechanisms called *social rules* to promote plans that are considered as suitable for human-robot interaction. The question they tackle is where to perform the task and how to balance between the efforts of the human and the robot [Mainprice 2012; Waldhart 2015].

Similarly, the planner we propose in Chapter 4 manages explicitly one elastic band per agent and plans for all. A number of social constraints have been specially devised to produce plans that would be possibly preferred by the human encountering the robot. The robot and the human bands tightness are different in order to force the robot take most of the load.

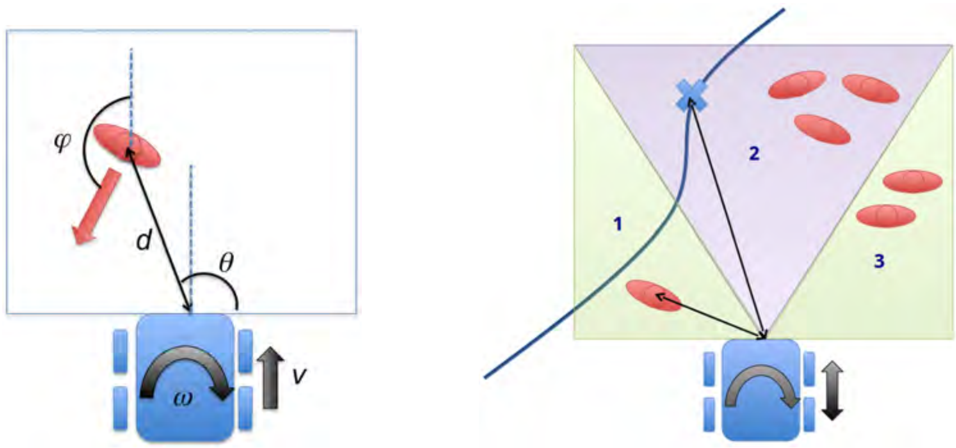
As the literature on human movements differs for stationary and moving people, social navigation architectures should also handle static and dynamic humans differently. One way to deal with moving people is to continuously re-plan the robot path as fast as the robot can perceive human positions. However, this quickly leads to erratic robot behaviors, especially in human-robot path-crossing situations. Therefore, conceptualization and development of novel algorithms specifically for robot navigation in dynamic settings is the core theme of this thesis.

2.3.2 Learning from Human Motion

In recent years, several machine learned based navigation methods appeared in the literature. The general scheme is to record human trajectories in the target environment and learn some parameters that influence trajectory planning and obstacle avoidance algorithm, or use the recorded trajectories to predict future human paths in populated environments to help robot plan paths that avoid the predicted human trajectories. [Kuderer 2012] presents an approach to predict future human trajectories based on a learned set of features that capture characteristics of human trajectories in populated environment. Predicted human trajectories in turn used for the robot to evaluate multiple topological variants of its own trajectory online and produce socially compliant robot trajectories. The authors have used a feature learning (data fitting) algorithm to fit following features obtained from human movements to generalize human motion in crowded environments: time, acceleration, velocity, collision avoidance (depends on distance between agents and their velocities) and topological variants (like moving on the right side of a corridor).

Inverse reinforcement learning (IRL) is a machine learning method that learns optimal policy that a demonstrator is following given several demonstrations and an MDP model of the situation. [Ng 2000] and [Zhifei 2012] discusses several algorithm for inverse reinforcement learning. [RamónVigo 2014; PérezHigueras 2014] uses inverse reinforcement learning as a tool to transfer the typical human navigation behavior to the robot local navigation planner. This method is able to generate low-level velocity controls for the robot using the learned policy. For this, the most relevant aspect is how they define the MDP model. The state for MDP includes relative position and orientation of persons to the robot (or expert teacher while learning), while the actions are discretized linear and angular velocities. For learning they have used openly available datasets of pedestrian motions, selecting some persons in the datasets as “experts” among the pedestrians. The experiments with this model suggest that the IRL policy is better than the planner that only uses proxemics costs, and the robot was able to anticipate smooth avoiding maneuvers when people are walking opposite to the robot.

An important aspect of analysis in [PérezHigueras 2014] is whether to consider only the nearest person or person density in the robot surroundings. Figure 2.6 shows how the MDP state is formed for each of these cases. The first model considering



(a) The state model that only considers nearest person.

(b) The state model that considers densities of person in the area in front of the robot.

Figure 2.6 – IRL states used for learning robot navigation strategy from pedestrian datasets by [PérezHigueras 2014]. Images taken from [PérezHigueras 2014]

only the nearest pedestrian sometimes results into excessive avoiding maneuvers. When mixing the first model with the model of pedestrian densities, the avoiding maneuvers becomes smoother and thus more suitable for crowded environments. This analysis also underlines the point that often different strategies are necessary to for social navigation in crowded and uncrowded situations.

Earlier IRL algorithms were limited to learning the reward function as linear combination of set of features. [Levine 2011] introduced a method to use Gaussian process to represent the reward as a nonlinear function of feature values, that is, the function if modelled as Gaussian process and its structure is determined by its kernel. The authors used a diagonal matrix kernel (also called automatic relevance detection kernel) with a regularization term. However, they assume that the training conditional (the reward function) is deterministic (having zero variance) to ease the computation. The results of simulated experiments compare favorable to other IRL algorithms such as maximum entropy IRL [Ziebart 2008] and feature construction IRL [Levine 2010].

Another interesting approach is to directly use the sensor data to extract features such as density and velocity of people around the robot (without direct human perception and tracking) that can represent the state vector for IRL [Kim 2016]. The IRL algorithm then provides primitive actions for the robot to execute during navigation by a low-level controller. More recent approach integrates the concept of homotopy within the IRL framework and learns which homotopy class should the robot select in given navigation situation in addition to the primitive actions [Kretzschmar 2016]. Thus this approach uses a joint mixture distribution to capture discrete decision-making aspect of selecting the relevant homotopy and continuous properties of the trajectory such as higher-order dynamics with IRL. Both of these methods were successfully tested in a hallway navigation scenario.

2.3.3 Legibility

One of the aims of human-aware navigation is to achieve more human friendly behaviors that are comfortable and legible for persons interacting with the robot. Even when the robot is paying utmost attention to human safety and social norms, if it is unable to convey its actions, particularly its motion direction, the visible robot behavior could lead to perceived discomfort or even danger for nearby people [Pacchierotti 2006]. A social robot thus has to act legibly as well. In that respect, more recent work goes beyond the safety criteria for richer interactive robot motions.

The concept of legibility describes one of the qualities beside safety that make robot behavior agreeable. Legibility expresses how easily an uninstructed person can estimate internal state of the robot from observations. It also enables an observer to quickly infer the robot intent and goals [Dragan 2013b]. In [Lichtenthaler 2012] the authors have carried out a video-based user study comparing human-aware and non-human-aware navigation algorithms in situations where a human crosses the path of a robot. Users saw first-person perspective videos of crossing situations with a robot using different navigation algorithms and predicted the robot motion and rated their own emotions during the interaction shown in the videos. They found correlation between legibility and predictability. Although correlated, legibility is different from predictability. Predictability requires an observer to imagine the most likely future state, determined by many other factors than the robot internal state [Dragan 2013a].

The directional cost model introduced by [Kruse 2012] have shown to increase legibility of the robot motions, where a robot attempts to solve a spatial conflict by adjusting velocity instead of path when possible. From an extensive user-study, it is clear that humans prefer robot following this strategy, particularly in 90° path crossing situations [Kruse 2014]. We exploit this result and introduce *directional costs* in the optimization framework used in the cooperative planners we introduce in Section 4.1.

2.3.4 Normative Behaviors

Leader-follower is a model that some robot navigation researchers adopted for scenarios like moving in a long hallway. [Müller 2010] is an example. In this work the authors first implemented a method to track people using on board laser data. For path planning, they first plan a path considering only static obstacles (with A* algorithm), that is, not including humans. Then they include humans into the occupancy map and if they are not interfere with plan trajectory robot carries on the planned path. In other case (humans interfering the robot path) the robot identifies finds a person who is not considered as obstacle to become a leader and follow them. For that they compare the short-term predicted track of people with computed path in previous step, and choose a person with maximum deviation of 30°.

[Morales 2013; Morales 2015] describes a way to generate "visibility" costmaps for indoor environments. This visibility map is generated using a visibility index that depends on how far the laser scanners can "see" at particular position on the map. Effectively, it adds costs on the corners of corridors in the environment. The costmap provides a layer for the layered costmaps approach introduced by [Lu 2014b]. Another layer they add is a static layer, costs of which depends on how far a map cell is from the walls. After that the default `move_base` planners can generates paths for a robot (a robotic wheelchair in this case) to follow. 30 Japanese participants who used robotic wheelchair rated their approach more comfortable compared to an approach without any social costmap layers.

While designing socially normative behaviors robot, one should note that these behaviors may differ from what we consider normative behaviors for people [Joosse 2013]. The authors have suggested to use the following scheme for experiments:

- Define social behavior norms from behavioral science and social psychology literature.
- Experiment with human and robot either adhering or violating the specific social norm.
- Measure participants' subjective (attitudinal) and objective (behavioral) response to determine whether participants find the robot's behavior as socially normative.

2.4 Summary

As we have seen that human-robot navigation is an amalgamation of many research areas. Our interest is on improving cooperativeness and coordination of robot movements with human in its surrounding. We made use of the proxemics and other social costs (such as visibility) for global planning. For local planning we utilized the timed elastic band approach and created a cooperative and proactive navigation planner that works well in different canonical human-robot path crossing scenarios. Finally, to improve the legibility of overall robot motion we designed a robot-gaze behavior that works well with and complements the navigation planners.

Chapter 3

Reactive Human-Aware Navigation

Contents

In this chapter we introduce a human-aware navigation framework that we have comprehensively and successfully testes in simulated and real-work experiments. This human-aware navigation framework includes several software components that we have developed throughout the course of this thesis. Apart from core software modules such as navigation planners, social costmaps and a head controller, the full architecture includes auxiliary software modules for human perception, prediction of human motions, simulation, visualization and user-study setup.

The second part of this chapter introduces a local planning scheme that prefers to adapt the robot speed over changing its path. This scheme takes motivation from the directional costs notion that we duly introduce in the Section 3.2.1. With a thorough user study of human-robot path crossing scenario we justify the effectiveness of this scheme.

3.1 Human-Aware Navigation Framework

From the literature review in the last section, it is clear that only advancement in path planning and obstacle avoidance is not enough to synthesize robot behavior that is human-like and human-acceptable in a broad range of scenarios particularly in large public places. Therefore in the comprehensive robot navigation framework we contributed also to the aspects of human path prediction, inference from human perception, social cost models, legibility of robot behavior as well as integration of component, simulation of human locomotion, testing and visualization of planning procedure. For that reason, we discuss here the human-aware navigation framework we used throughout this thesis.

3.1.1 Planning

In the last section we discussed several social cost function such as safety, visibility and hidden zones that can help path planning algorithms to compute human-aware navigation paths [Sisbot 2007]. These social cost functions were originally developed to work with the open source motion planning library Move3D[1]. Since we are extensively using the ROS framework `move_base` in our work, we have ported the safety and visibility functions to work as a plugin for the costmap library[2] of `move_base`. This plugin adds a layer in the multi-layer construction of the costmap [Lu 2014b; Lu 2014a]. The layer is fully configurable in terms of what costs to use and the weights to set for both the safety and the visibility layer.

Figure 3.1 shows examples of how this layer generates cost around tracked humans in the environment[3]. This costmap layer can easily be combined with other social cost layers to add further social behavior in robot navigation. This costmap layer using Gaussian function to add costs around humans, the parameters of this function are fully configurable. Tuning of these parameters can greatly affect the precise behavior of the robot. [Lu 2013] has a good discussion on the effect of various parameters on generated paths using Gaussian cost functions.

We have already mentioned earlier that it is important to differentiate between moving and static people and treat them appropriately in navigation planning. This costmap layer is only meant to deal with people who the robot considers static in the environment by computing a global path that ensures their safety and takes into account self visibility to the humans. Since, there is usually some error involved perceiving speed of detected persons, we also added a configurable threshold below which speed the robot will consider the person static.

Local planning modules are the core contributions of this thesis. We have developed two different local planners, first planner uses directional cost model and velocity adaptation scheme that we call a reactive human-aware planner and learning from the experiments with this reactive planner later we developed a cooperative and proactive planner that integrates social constraints in an optimization framework. Details of both of these planners are given in sections 3.2 and 4.1 respectively. Both of the planners are fully integrated into the `move_base` framework, thus the directly output desired velocity for the robot motion controllers at each iteration of planning.

[1] `https://www.openrobots.org/wiki/move3d`
[2] `http://wiki.ros.org/costmap_2d`
[3] Source code for the HANP costmap layer: `http://harmish.in/HANP/code/layer`

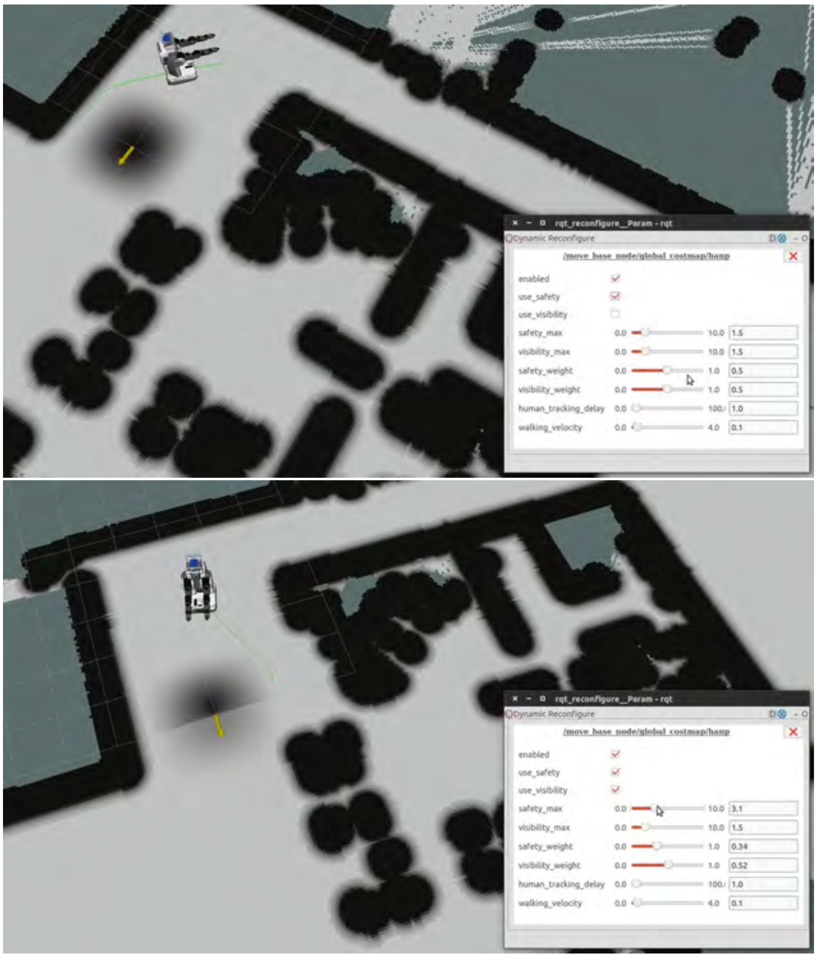


Figure 3.1 – Proxemics costs for path planning (in shades of black to white, darker shades denote higher costs) and generated path (green). Arrow (yellow) denote tracked position of a person.

Both the local and global planner continuously need information from human perception module. We have installed a motion capture system from OptiTrack[4] at our premises with multiple cameras in an area of around 80 m^2. The motion capture system detects certain reflective markers, for that reason we ask participants to wear a light weight helmet with markers stitched on them during our experiments in this area. We have also deployed the reactive planner during the real-world tests with the Spencer robot in Amsterdam airport Schiphol, however, there we have utilized output from onboard human detection and tracking module. Figure 3.2 shows the map of the experimentation area, motion capture cameras and helmet worn by a person.

A further requirement for treating humans and ordinary obstacle differently is to detect them differently. On board laser scanners usually detect ordinary obstacles,

[4] `http://www.optitrack.com/`

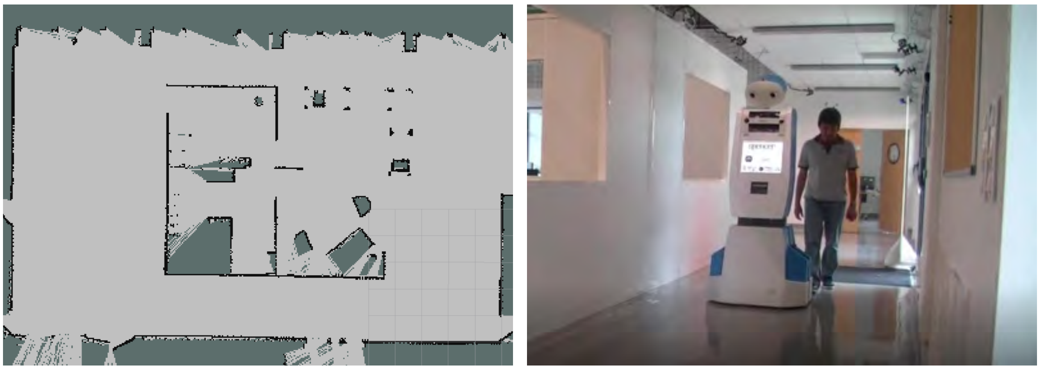


Figure 3.2 – Experiment area with motion capture system to detect human positions and velocities. Figure on the left shows an image of a person being guided by the robot, the robot tracks the helmet that the person is wearing with the motion capture cameras Figure on the right shows map of an area where we have installed the motion capture system.

but the sensor data also includes area occluded by humans. So, we add a laser scan filter that filters our human data from the scan data to give us only the scan points that belong to non-human obstacles. This data is then fed to an obstacle costmap layer which the planners use to avoid these obstacles.

3.1.2 Human Motion Prediction

The local planners also require predicted human path to better plan its human-aware trajectories. Velocity obstacle [Fiorini 1998] is one method to predict future human positions whose uncertainly increases with the future time of prediction. We have developed a separate module that continuously receives tracking information from human perception module and can return predicted human positions on request. The local planner usually requests this prediction information at each iteration of planning.

Apart from velocity obstacle, we have added another method from predicting future human positions at any given time in the future from their current position and velocity. This method also scales the current velocity to add uncertainty around future human positions in case the human decides to slow down or speed up. Figure 3.3 shows examples of both of these prediction methods.

Sometimes we can predict a goal for the human in the same environment where the robot operating and we want to know probable path that the human may take to reach this goal. So we have added a third method which simply takes current human position a probable goal position and gives an A^* path between these positions as probable path that the human may take. An interesting extension to this method would be to include sophisticated algorithms that can estimate a list of different possible goals for a human based on their past motion, as suggested by [Ferrer 2014a].

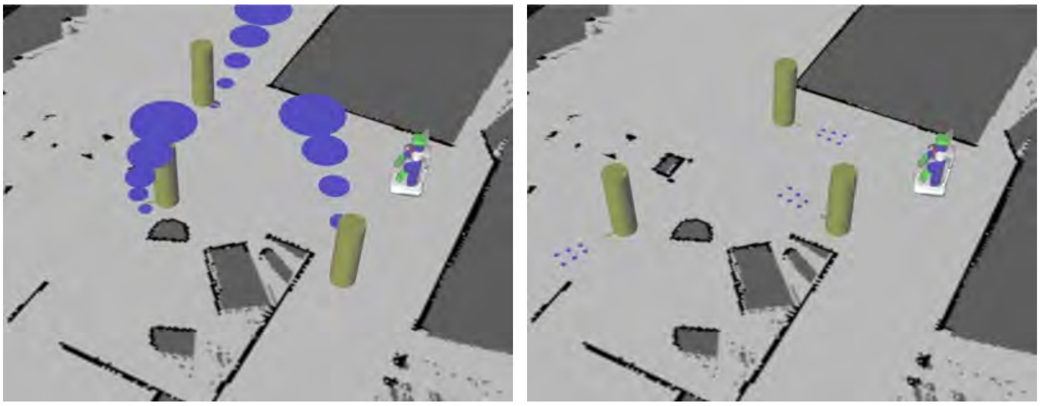


Figure 3.3 – Human pose prediction. Figure on the left shows predicted human positions with velocity-obstacle method, radius of the blue circles defines uncertainty and height of each circle represents future time of prediction. Figure on the right shows predicted human positions based on the instantaneous velocities of the tracked persons with some uncertainly added in linear and angular velocities.

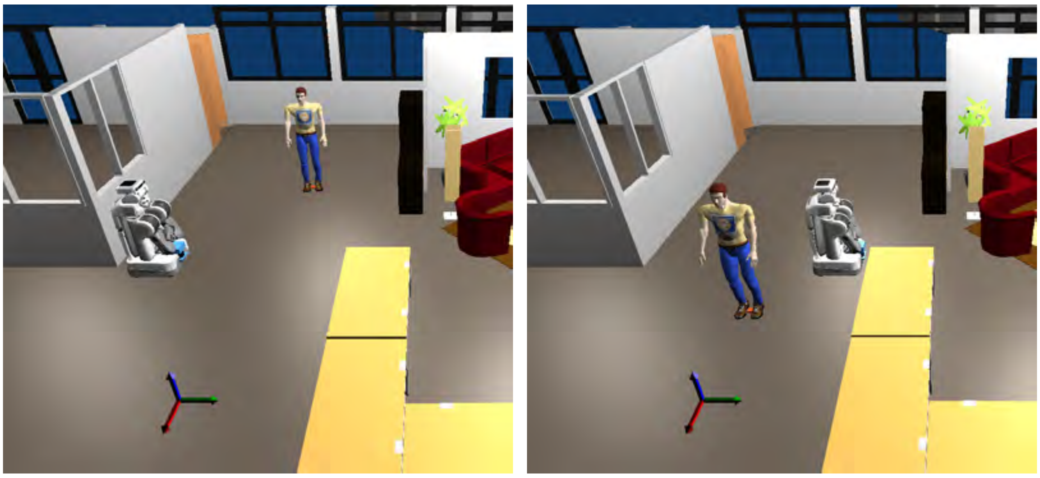


Figure 3.4 – A human-robot path crossing scenario created in MORSE simulator. The figure on the left shows the situation a few seconds before the situation of figure on the right. The robot and human had crossed their paths during this time.

3.1.3 Simulation

MORSE[5] is an open source, lightweight and easy-to-use robotic simulator with particular focus on human-robot interaction experiments [Lemaignan 2014]. We have extensively used the MORSE simulator to test and compare human-aware navigation algorithms. Figure 3.4 shows an example simulation environment created for human-robot path crossing scenario.

We have designed a human navigation simulator with a framework similar to `move\_base`, thereby using separate global and local planning modules. The global

[5] `https://www.openrobots.org/morse`

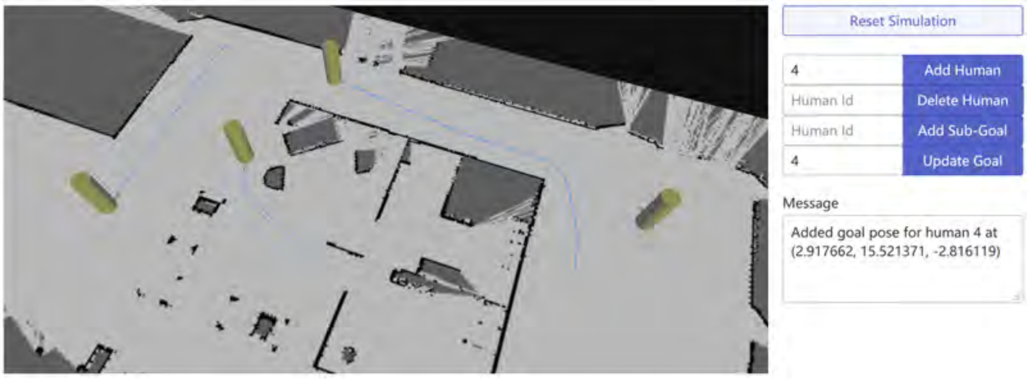


Figure 3.5 – A human motion simulator that can simulate 2D motion of multiple humans. Cylinders represent current human position and associated radius. The blue lines show the paths that each simulated person is following. The panel on the right side allows to manipulate number of simulated persons and their goal positions.

planning module uses the same global costmap that of the robot, for planning global paths for humans between given start and goal positions using the same default global planner as `move_base`. As the local planning module we have developed simple teleport controller to update human position and velocity, making human move on the global path at nominal walking velocity. Figure 3.5 shows a screenshot of this human motion simulator. This human simulator can simultaneously simulate motion of multiple humans[6].

As the figure 3.5 shows, we purposely represent human as cylinders. This is due to two reasons, 1. human tracking sensors generally only track people as two-dimensional points with some velocity associated with it, and 2. human-aware planners take this position and add generally assume some radius around the tracked person positions (and some nominal height for each person) that the planner should avoid for safety. Thus, we represent the humans as what the planner actually sees rather than displaying animated human characters. The simulator reads a configuration file with person identification numbers and their start and goal positions at start for easier setting and re-setting of the simulator. Each human can have one goal position, reaching that position will make the human stay at that position. However, for each simulated human we can add multiple ordered sub-goals that it must visit before reaching the goal position. This facility to add sub-goals allows us to quickly manipulate human paths to generate different human behaviors while testing particular robot navigation algorithms.

Additionally, the local planner can subscribe to trajectories provided by other modules. In this mode it teleports the human according to the velocity embedded into the trajectory information. This mode helps us to test the cooperative planner we proposed in this thesis (Section 4.1) effectively, by simulating human motions as predicted by the optimization process.

[6]Source code for the human simulator is available at: `http://harmish.in/HSIM/code`

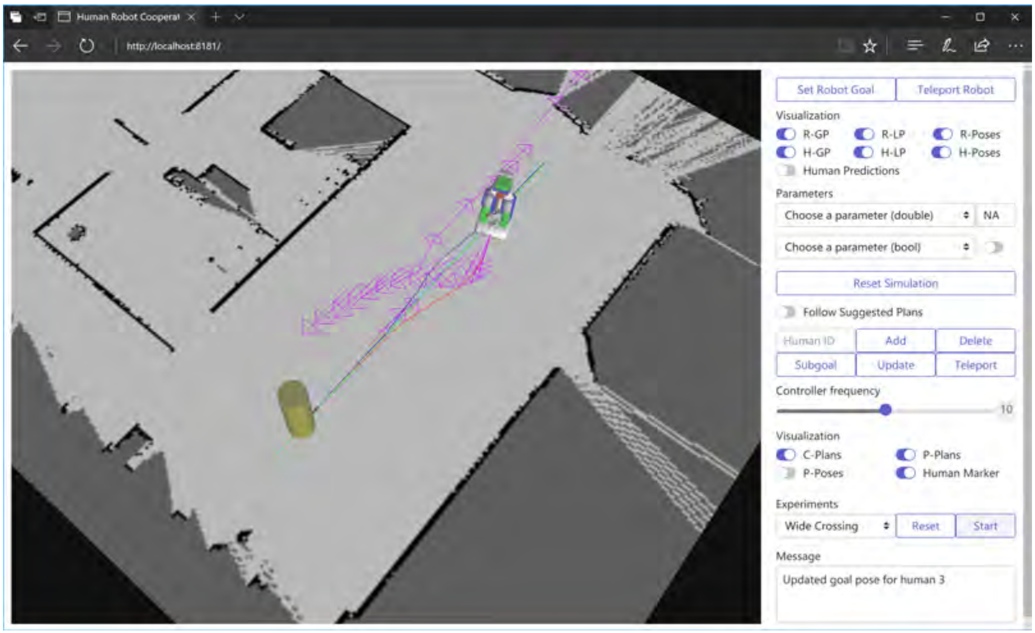


Figure 3.6 – User interface for visualizing and controlling experiments for human-aware navigation planners. This interface uses and extends the software components developed by ROS community as part of the robot-web-tools efforts [Toris 2015].

3.1.4 Visualization

For some of the human-aware planners introduced in the next chapter, we define several message type for communication between modules (such as robot trajectory, human trajectories and velocities)[7]. We have also developed software modules to visualize these messages. Additionally for testing the cooperative planner we have created a user-interface to control the experiments. This user-interface utilizes the web tools [Toris 2015] developed by the ROS community[8] which allows to visualize and control operating parameters from any web browser. Figure 3.6 shows a screenshot of the user interface.

All of this software modules working together results into desired human-aware behaviors. We have testes this framework in Amsterdam airport Schiphol during the course of the SPENCER project. Figure 3.7 shows the Spencer robot guiding the passengers during busy airport hours. Here we have only used the reactive planner which reduces its speed when humans are crossing in the front and once human passes-by it re-accelerates. We also had certain situations where people stop in front of the robot. In these cases the robot quickly detects them as static people and in the next iteration of global planning it computes a new path to safely go around them.

Having introduced the main software modules for our human-aware navigation

[7]Message definitions for human-aware navigation: `http://harmish.in/HANP/code/msgs`
[8]`http://robotwebtools.org/`

Figure 3.7 – Impressions of the reactive planner guiding people in Amsterdam airport Schiphol.

framework, in the next section we present the reactive local planner that favors velocity adaptation against changing paths depending on a directional cost model. First we will discuss the directional cost model and later we present a user study showing that with use of the directional cost model the robot can improve the legibility and perceived comfort for co-existing humans in the robot operating environment.

3.2 Evaluating Directional Cost Model

When we follow the common approach of using spatial cost functions the resulting robot behavior often appears confusing. This is due to the unpredictably of dynamic scenarios which may constantly invalidate the last calculated path as situation unfolds. Figure 3.8 shows an example of such behavior.

Temporal planning can help in this type of uncertain environments if it can predict the human path with sufficient accuracy and already select a path well in advance that avoid the robot oscillating around human. However, it quickly runs into combinatorial explosions of search space as we have discussed in the previous chapter. The concept of directional cost model [Kruse 2012] is supposed to mitigate this problem with directional cost functions. To reduce the directional costs our robot will attempt to solve spatial conflicts by adjusting its speed instead of its path. We conducted a user study with a PR2[9] robot and human participants to evaluate the robot behavior resulting from the use of this directional cost based velocity adaptation strategy.

[9]`http://wiki.ros.org/Robots/PR2`

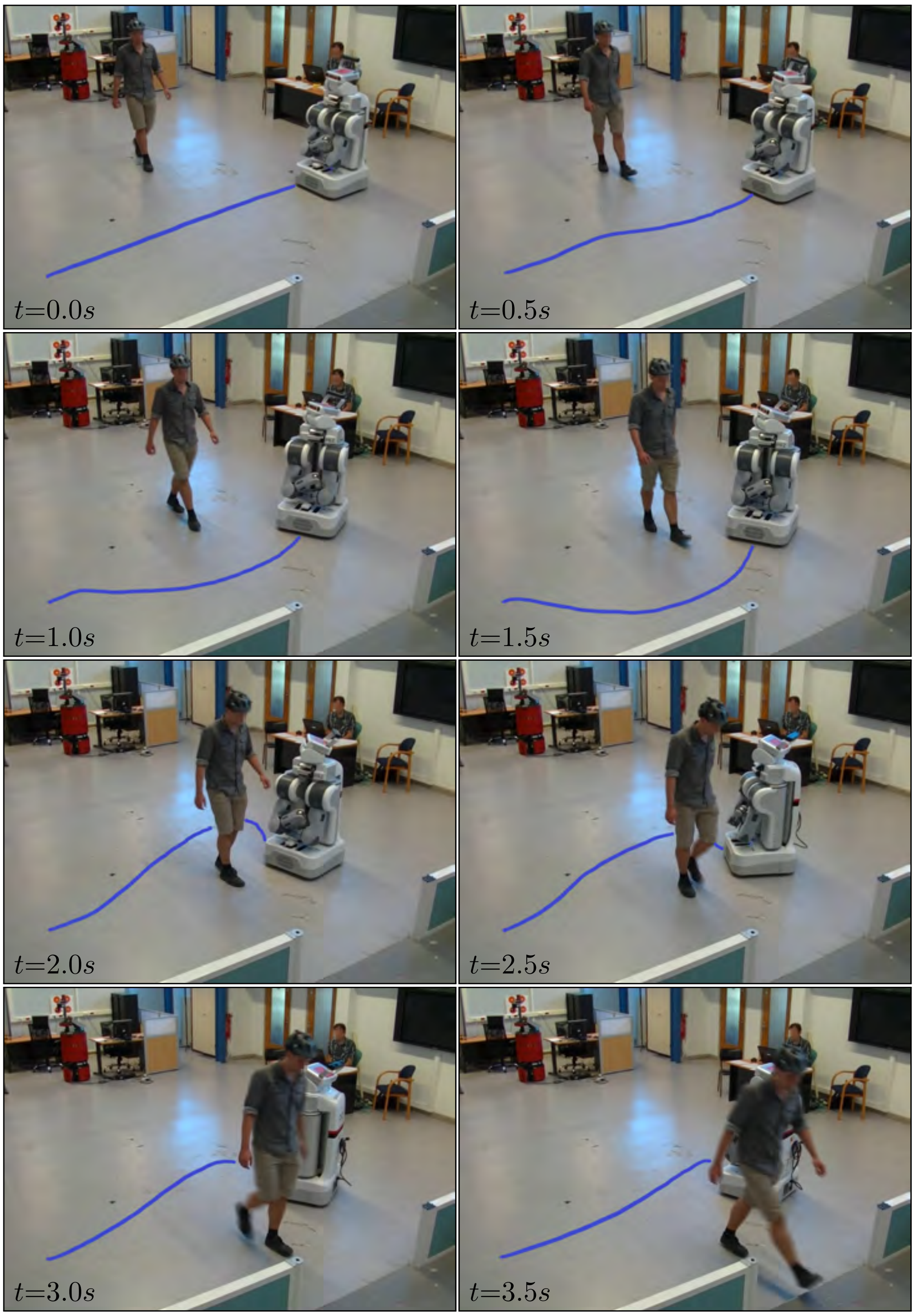


Figure 3.8 – Effects of re-planning with spatial costs for human safety. The robot is planning paths going from right to left, a person walks from up to down without stopping for the robot. The blue line shows the planned path of the robot. As the person moves near to the robot, its shortest path towards the goal passes in front of the person. However, because the person does not stop the safety costs around the person forces the robot to plan a path that passes behind the person. This movement of the robot also causes some discomfort to the person when he passes near the robot. The lower left corner shows the time relative to the first image.

H	position of human
$\hat{hd}$	heading of human
$\hat{hd}'$	translated $\hat{hd}$
$\hat{hd}^{-1}$	inverse $\hat{hd}'$
w_i	i^{th} waypoint on path
P	closest point on human path to w_i
d_P	distance of P and w_i
$\hat{rd}$	potential heading of robot at w_i
α	angle between $\hat{rd}$ and $\hat{hd}^{-1}$

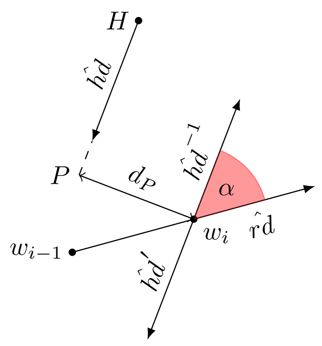


Figure 3.9 – Context dependent costs model *ContextCost*, from [Kruse 2012].

3.2.1 The Directional Cost Model

The cost model *ContextCost* [Kruse 2012] defines social costs for a path search in a grid that depends on the motion direction of present humans and of the potential movement of the robot. It gives a measure of compatibility of the situation, meaning it quantifies how agreeable the situation is if the robot and the human continue moving on their predicted path with their current velocities. While calculating the compatibility it may discard the social costs for moving towards humans in path crossing situations. Which in turn requires a local planner to adjust the robot velocity, Section 3.2.2 explains the strategy that the local planner uses.

When using extended proxemic costs (including costs for safety, visibility, hidden zones) different specialized functions f_i calculate spatial costs for each human H for each grid cell of a costmap. The final cost of each grid cell could be calculated using the function:

$$\varsigma'_{Static}(H, w_i) = \max\left(f_1(H, w_i), f_2(H, w_i), \ldots, f_k(H, w_i)\right) \tag{3.1}$$

The purpose of the user study was to evaluate the usefulness of the cost model *ContextCost* with velocity adaptation strategy in dynamic real-world situations over standard cost models with only proxemic costs for safety (denoting this model as *Static*). Both cost models use a proxemic model of equation 3.1 for social distancing to guide search for a path on a 2D grid by, that is, for global planning. The *Static* model makes no difference between moving or static people, and relies only on continuous planning for avoiding collision with moving people. However, while using *ContextCost* model we use a compatibility measure for each moving human to adapt the robot velocity with a local planner.

Figure 3.9 illustrates the geometry for calculating compatibility using the cost model *ContextCost*. In this equation default values useful for most of the human-robot co-navigation situations are 1.0 m for d_{low}, 3.0 m for d_{high} and 1.4 rad for $\alpha_{\max}$. The incompatibility function ϕ (see Equation 3.2) calculates a number $\in [0, 1]$ which we call the compatibility measure. In the broader concept of directional cost model we can multiply the costs of the model *Static* with this compatibility measure to get new costmap to search a robot path. However, for purpose of this study we only use this compatibility measure for adjusting the robot speed.

$$\phi(H, w_i, w_{i-1}) = \begin{cases} 1 & \text{if} \quad \hat{hd} \quad \text{undefined} \\ 1 & \text{if} \quad d_P \leq d_{low} \\ 0 & \text{if} \quad d_P \geq d_{high} \\ 0 & \text{if} \quad \alpha \geq \alpha_{\max} \\ \dfrac{d_P - d_{low}}{d_{high}} \cdot \dfrac{\alpha}{\alpha_{\max}} & \text{otherwise} \end{cases} \tag{3.2}$$

With equation 3.2, when a robot path segment should not conflict with a predicted human motion, social costs calculated by ϕ do not apply. Thus robot considers its own path direction incompatible with human direction if the human and robot could frontally run into each other. In all other cases, the motions are compatible, meaning it is likely the situation can work out well if the robot just reduces its velocity on the path while the human is in the way. This strategy shifts the burden of obeying social distancing rules around moving humans from global planning to local planning and thus makes the robot follow more goal-directed paths. We assumed that such strategy will make robot's navigation intention more legible, to find support for this assumption we carried out a user study (presented in Section 3.2.3 and Section 3.2.4). This user-study was a collaborative work with [Kruse 2015] and it has been partly published in [Kruse 2014].

3.2.2 Velocity Adaptation

When robot and human paths are spatially compatible, the robot can stay on its intended (straight) path. Only in incompatible situations, it has to take specific avoidance action. Thus, for the experiments we developed a waypoint following algorithm that adapts the robot velocity to prevent collisions with perceived humans. The waypoint follower causes the robot to visit each waypoint of the global plan, by turning towards each waypoint in turn and then moving forward to reach it (as opposed to allowing sideways or backward motions). Using the cost model *ContextCost*, the robot changed it velocity so that with projected motion of the robot and the human their mutual distance do not decrease below 1.3 m. Primary trials yielded this distance that humans felt safe while still keeping the robot close enough that can create a crossing conflict situation.

Based on the preliminary trials, we found the linear velocity profile shown in figure 3.10 is ideal for crossing the path of a human when giving them way. There are three distinct phases for braking, waiting, and restarting. Ideally, there is no acceleration during the "Brake" phase, and no significant acceleration during the "Wait" phase. These phases should be clearly observable by the human, and the timing of the phases could be crucial to human comfort.

Figure 3.11a shows the transition to the "Brake" phase. During the preliminary trials, when the robot decelerated late or had the slightest acceleration motion, par-

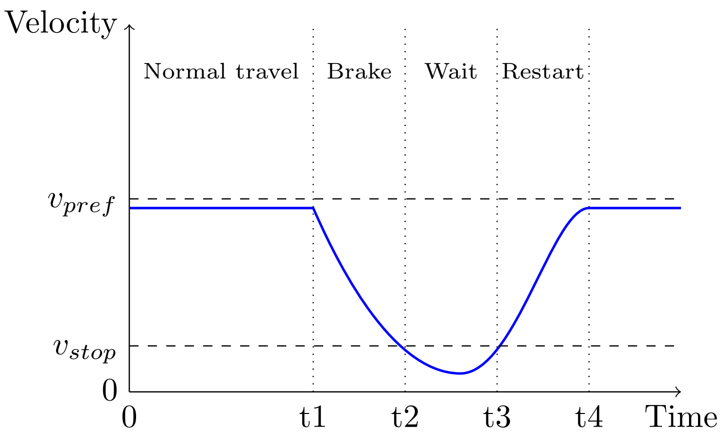


Figure 3.10 – Idealized velocity profile scheme for 90° crossing situations with cost model *ContextCost*. v_{pref} indicates the preferred travel speed, v_{stop} a velocity so low the robot is considered not traveling.

ticipants felt the situation was uncomfortable. We believe this is because participants fear the robot to accelerate to full speed at the smallest symptom of acceleration. Noisy data and prediction was found the cause for short-time robot accelerations, so we suppressed it by adding uncertainty to the perception data by predicting low, nominal and high human velocities, and selecting a velocity that avoids collision with all three prediction models. This caused the robot to select low velocities more often, and made participants less nervous. Figure 3.11b shows the “Restart” phase. Sometimes the participants felt uncomfortable expressing the robot moved *too early*. We believe again that the participants fear the robot quickly accelerating to full speed, so we corrected this behavior by projecting the robot position not at the desired speed but at full speed, which caused the robot to accelerate late and not discomforting the participants anymore. Using both increased uncertainty about the future human positions and projecting the robot at full speed, the robot behavior in the crossing situation became rather similar to the scheme in figure 3.10.

3.2.3 User Study Setup

The context dependent cost model has already been compared in simulation to human motion strategies in 2-agent orthogonal crossing scenarios [Basili 2013]. Encouraged by the results, we wanted to further evaluate the cost function in a real world setting when presenting experiment participants to a real world autonomously moving robot. As mentioned earlier we tested the robot behavior on two strategies as summarized in table 3.1.

17 participants were given the task to act as interfering walkers with a robot moving from one spot to another in an area without static obstacles. We used a motion capture system to track human participants based on four passive infrared markers placed on a lightweight helmet the participants wore while walking, as

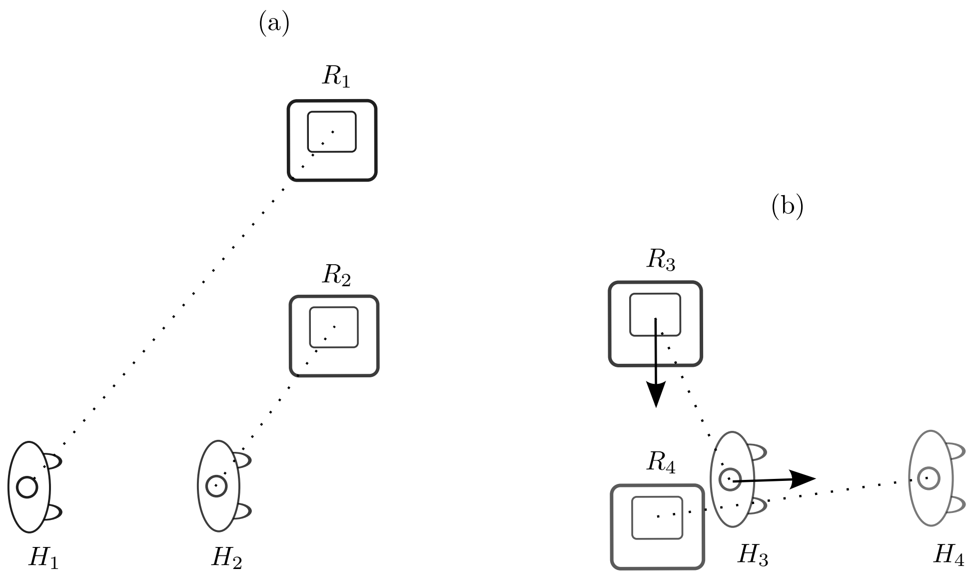


Figure 3.11 – Situations relevant for comfort while crossing with cost model *ContextCost*. R_t and H_t are positions of the robot and human at time-step t respectively. Stopping late in (a) or starting early in (b) cause discomfort.

Cost model:	*Static*	*ContextCost*
Path:	Changing paths according to the costmap	Straight line path towards the goal position
Velocity:	Maximum possible speed without collision	Reduce speed to keep distance

Table 3.1 – Comparison of experiment conditions.

shown in figure 3.12. The room had an experimental area with offices and desks surrounding it, during the experiment people were quietly working around the experiment area.

The participants were all given the same set of written instructions to read. The instructions stated that the robot will use multiple different strategies, but not how many or what variations to expect. The procedure was the following:

- In the preparation stage, the robot and the participant moved independently to their starting positions. A mark on the floor showed the starting position of the participant, as illustrated in figure 3.12.
- When both were ready, the experiment instructor pressed a button to start the execution phase. This triggered the robot motion planning and made the robot say "go".
- The participant and the robot would start moving towards their respective goal positions at roughly the same time. The robot planned its path at about 10 Hz

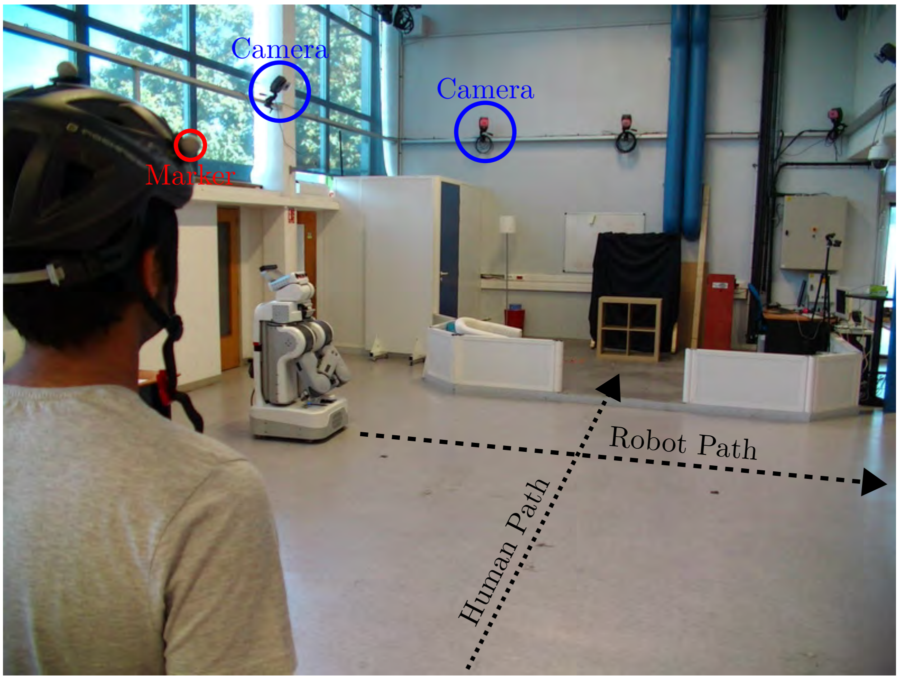


Figure 3.12 – Experiment setup showing wall-mounted cameras (2 out of 10), participant at start position with helmet and passive marker (1 out of 4), robot at start position, the intended paths crossing at 90°

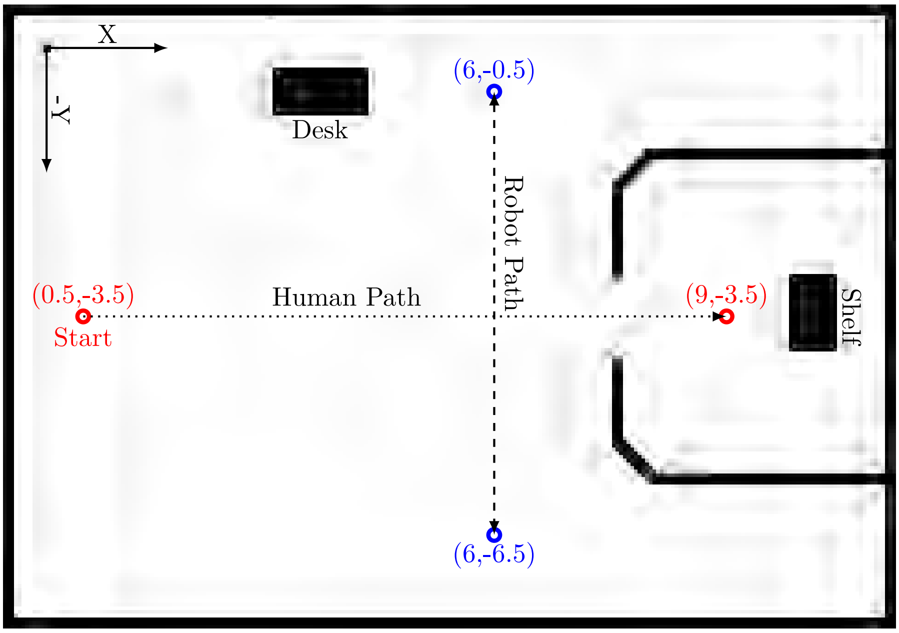


Figure 3.13 – Top-Down view on the room layout, showing the experiment path, the instructor's desk, and the shelf where the participants marked their answer after each trial

frequency. The map in figure 3.13 shows the coordinates. The participant goal was a small shelf on which lay a questionnaire. Going towards a shelf for a purpose rather than going to a marked spot seemed to benefit a natural walking behavior.

- The goal area of the participant was slightly elevated (10 cm), and there was a low barricade with a gate spanning 1.5 m, seen in white in figure 3.12. The participants started their movement at coordinates (0.5, −3.5) on the map, but would not be visible to the motion capture system before reaching roughly (2.5, 3.5). Again we decided to maximize the walking distance because we were mostly interested in ratings about the robot, not human gait.
- When the participants reached the shelf, they could then immediately rate the robot behavior on the questionnaire placed on the shelf. Once the participant and the robot had both reached their goal positions, the execution phase ended. Then a new cycle would begin with the preparation phase.
- An instructor was present and visible during the experiment, sitting at the desk shown in figure 3.13, to react in cases of emergencies or disturbances, but also to reassure the participant and verify they were following the procedure. Also the instructor modified the robot maximum speed between 0.45 m/s and 0.55 m/s according to the participant's walking speed during the first 3 trials to create a spatial conflict without imposing a walking speed on the participants.
- The robot would approach alternately from the left or the right hand side. Each participant performed the same task 13 times, and knew that the first 3 cycles would not be taken into account, but serve to get used to the situation.
- We randomized the robot strategy but never the same three times in a row. All participants experienced both strategies to allow them a comparative opinion.

After each run, the participants were asked to rate the robot's performance according to predefined questions:

Q1 Please rate the robot behavior (clear vs. confusing)

Q2 Please rate the crossing situation (comfort)

The participants were given a Semantic Differential Scale from 1 to 5, the extremes labeled *clear*, *confusing* for Q1 and *comfortable*, *uncomfortable* for Q2.

Based on the design of the cost model *ContextCost*, the hypotheses for the outcome of the experiment were the following:

Hypothesis 3.1 (H3.1): *The robot behavior observed in simulation for Static also occurs in the experiment.*

Hypothesis 3.2 (H3.2): *Participants rate path adaptation as more confusing than velocity adaptation.*

Hypothesis 3.3 (H3.3): *Participants rate path adaptation as more uncomfortable than velocity adaptation.*

The alternatives are trivially negations of the given hypothesis. Hypotheses H3.2 and H3.3 would be confirmed if significant difference between the reported ratings were found.

3.2.4 User Study Results

The user study involved 17 Participants:

- Number of Participants: 2 female, 15 male
- Age range: 22 — 34
- Mother language: 11 French, 6 Other
- Education level: 11 Master, 5 PhD
- Robot experience: 11 None, 6 with less than 6 years

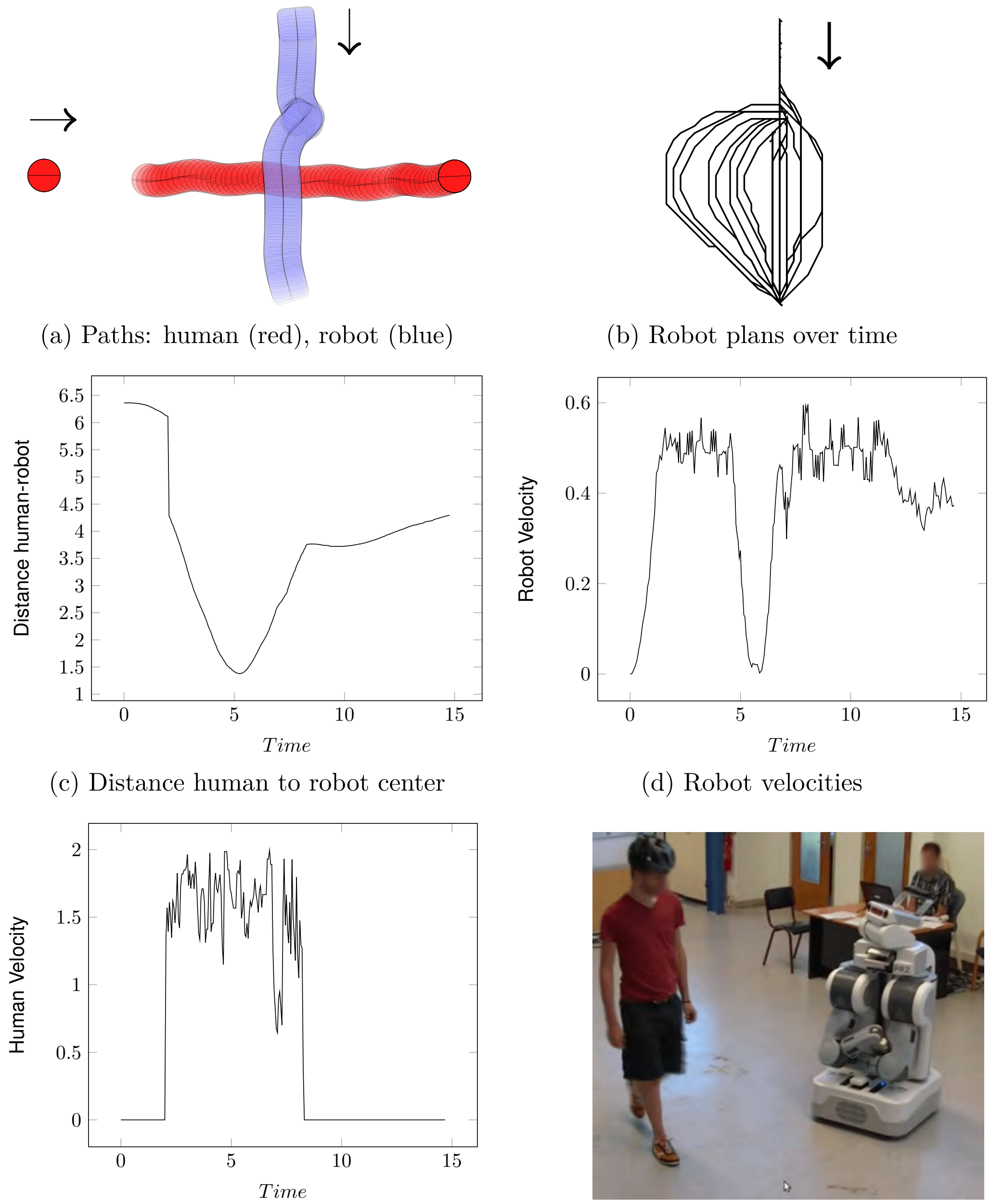
We collected data from 170 valid trial runs as shown in table 3.2. Figure 3.14 shows summary plots of a representative run using cost model *Static*, and figure 3.15 shows the same plots for a representative run using *ContextCost*. Figures 3.14a and 3.15a show the positions of robot and human over time for two representative samples we selected. Since we could not measure the human position from the start, we added the starting position as a single circle in the plot. Figures 3.14a and 3.14f shows how the robot at a distance of 1.2 m slightly deviates from the straight line towards its left and rotates. The human passes during that time. Once the human has passed, the robot plans paths behind the human as we can see in figure 3.14b, thus turning to its right before eventually regaining the straight path. Compare that to figures 3.15a and 3.15f, where the robot also decelerated to a minimal distance of 1.2 m, but never rotated, and never planned anything but a straight path as shown in figure 3.15b.

In figure 3.15d at time 3.8 there is a small increase of velocity. We could not pin down the cause of this, which could be with the prediction algorithm or robot wheel control. We point it out only to show that such a small and short acceleration already counted as a cue to some participants that the robot was maybe re-accelerating.

Figure 3.16 and 3.17 summarizes participant's answers to the questionnaire. One-way ANOVA indicates that the users rated trials with cost model *ContextCost*

	Static	*ContextCost*
robot start at $(6, -0.5)$	39	45
robot start at $(6, -6.5)$	41	45

Table 3.2 – Conditions for the 170 valid trials.



(a) Paths: human (red), robot (blue)

(b) Robot plans over time

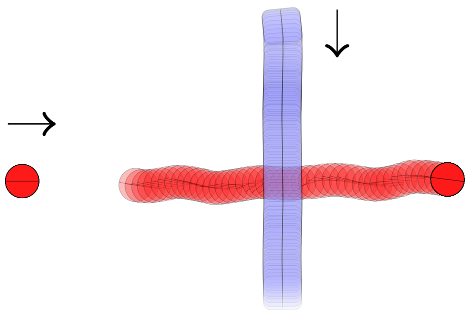
(c) Distance human to robot center

(d) Robot velocities

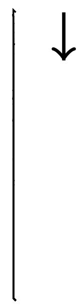
(e) Human velocities

(f) Position at minimum distance

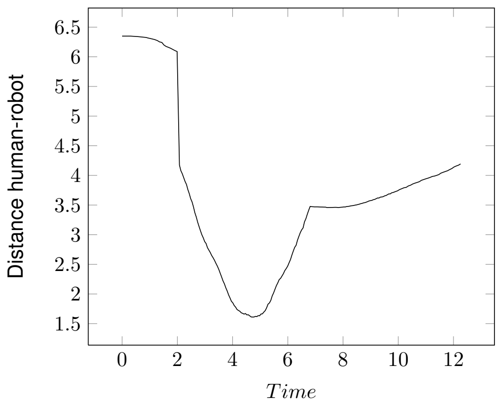
Figure 3.14 – A sample run with cost model *Static*. In (a) the round positions are the human moving from left to right. (b) shows what the diverse plans the robot generate while moving.



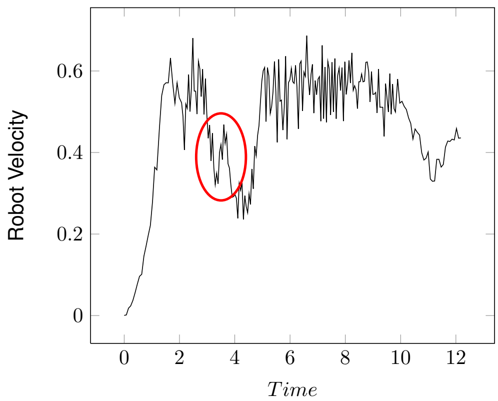
(a) Paths: human (red), robot (blue)



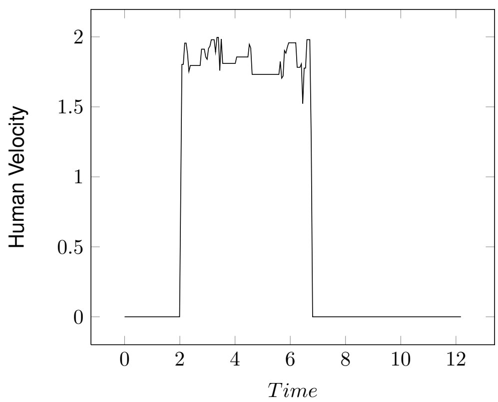
(b) Robot plans over time



(c) Distance human to robot center



(d) Robot velocities



(e) Human velocities

(f) Position at minimum distance

Figure 3.15 – A sample run with cost model *ContextCost*. Same plots as in figure 3.14.

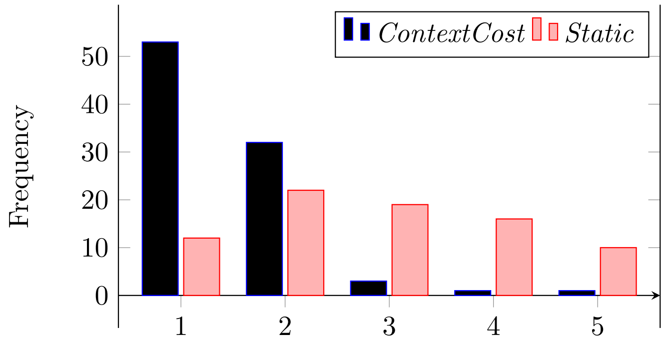


Figure 3.16 – Participant ratings of situation: (1=Comfortable, 5=Uncomfortable).

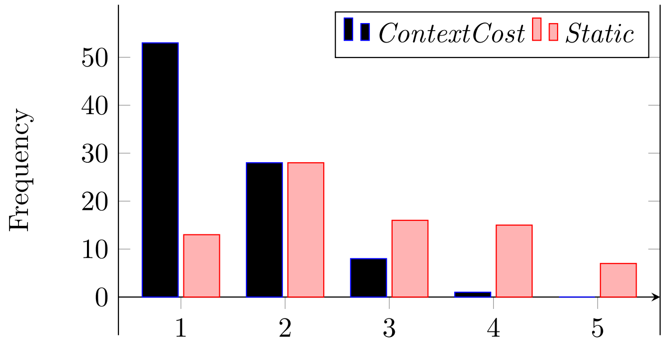


Figure 3.17 – Participant ratings of robot behavior: (1=Clear, 5=Strange).

more comfortable ($M = 1.5, SD = 0.723$) than for *Static* with $p < 0.001$ ($M = 2.873, SD = 1.265$). Similarly, the users rated trials with cost model *ContextCost* more clear ($M = 1.522, SD = 0.707$) than for *Static* ($M = 2.684, SD = 1.215$), $p < 0.001$. The ratings for clarity and discomfort were also correlated as revealed by Pearson's R test ($R = 0.771, p < 0.001$). Hence, hypothesis H3.1 is validated by the robot path behavior we saw, as displayed in figures 3.14. The distinct responses for clarity and comfort also confirm hypotheses H3.2 and H3.3.

The user study found not statistically significant differences between the cases of robot approaching from the right or left-hand side of the human. The observed human average walking velocities mainly varied between 1.5 m/s and 1.7 m/s with no clear trend over different trials. While figures 3.15e and 3.14e appear different, no repetitive pattern seemed evident over all 170 runs.

3.2.5 Analysis of Robot Behavior

Reported discomfort by participants seemed to always be related to uncertainty about future collisions. Participants felt generally disturbed when the robot base moved (or rotated) while the robot appeared it is waiting for them to pass. Legibility of robot motion perhaps a quality that only becomes relevant once humans feel sufficiently reassured about their safety when a robot is present in their surrounding.

Participants reported the robot base rotations as uncomfortable when it happened very close to the participants (due to the square shape of the robot, a rotation could also cause the robot collide with person's legs), but when it was at least 1.5 m away, the participants reported no discomfort.

Using the *ContextCost* model, the robot behaved very similar for each run, even when participants walked at different speeds or hesitated. This consistency made the robot behavior also very predictable. Which could also be the reason participants let the robot go first in 20 out of 80 runs when robot was using the *Static* model but never did so in 90 runs when the robot was using the *ContextCost* model.

This study did not validate the *ContextCost* model with velocity adaptation strategy for other conflict situations such as human and robot crossing at different angles or moving at different speeds. Such studies would demand a larger experimental area than what was available to us. Also the study did not compare the straight path following behavior with a behavior where robot passes behind the human with a curved path, finding such a path would require the robot to more precisely predict medium term human paths and evaluate over future paths that the robot that can take to resolve future conflicts.

While participants' primary focus was on the robot path behavior, we saw that minor acceleration and deceleration were also used as significant cues by them, so a social robot has to avoid presenting such cues to prevent confusion. One possibility is to develop a social commitment behavior such that the robot commits itself to stopping or moving for at least given time-span of one or two seconds. Auditory cues like hectic motion of PR2 caster wheels during low velocities or starting of cooling vents do not clearly express a specific robot intent, but they are signals that can confuse people.

During the experiments we used a simple waypoint following algorithm without any special obstacle avoidance capabilities. However, because of these promising results later we fully integrated this directional cost model with a local planner that uses the dynamic window approach (DWA) [Fox 1997] for avoiding other dynamic obstacles during global planning. In this scheme we clearly differentiate between static and moving people, specifically for static people we only use the costs provided by the *Static* model and do not filter them with the compatibility measure calculated by the *ContextCost* model. For dynamic people we use similar strategy of lowering the robot speed such that projection of robot's and human's path do not lead to collision. However added advantage of using DWA is we can also avoid other obstacles than humans, and be able to cope with situation where on-board person tracing modules fail to detect a person in the semi-crowded environments. We have successfully tested this local planner on both the PR2 and the Spencer robot and used it during the preliminary tests of the Spencer robot in Amsterdam airport Schiphol as shown in figure 3.7.

3.3 Summary

In this chapter we have introduced a reactive planning framework with a local planner that tries to avoid colliding with moving people by reducing its own speed and thus allowing them to pass in front of the robot. For static humans in the environment, we use a social costmap layer that causes the robot to plan paths to go around them in safe and socially acceptable way. The reactive framework also includes a software module for predicting future human paths based on their current linear and angular velocities.

A real world user study with 17 participants and 170 trials gave us the impression that such a strategy can help improve the legibility and perceived comfort by humans about the robot's navigation behavior. The study also reveals that for legibility, it is not only crucial to provide useful cues to the robot internal state but also to avoid accidentally creating misleading cues.

The strategy of slowing down the robot to avoid collision, however, could lead to somewhat sluggish behavior of the robot in crowded or spatially constrained environments. A complete solution would also need the robot to change its paths such that it allow humans to comfortably navigate around robots. It would be even better if a robot can ensure that humans have a valid paths to reach their goals and its own path is not obstructing paths of people around it. In other words, socially acceptable robot need not only have to remain reactive but also become cooperative and sometimes proactive to plan solutions to situations rather than just its own paths. In the next chapter we will introduce our approach of treating the human-ware robot navigation as a cooperative task.

Chapter 4

Cooperative and Proactive Human-Aware Navigation

Contents

This chapter provides the core contributions to human-aware navigation planning of this thesis, while the following chapter will put forward contributions to head and body coordination that supplement to this work.

In the previous chapter we have identified that combining a path planner (global planner) with obstacle avoidance module (local planner) is a usual and pertinent choice to achieve soft real-time motion planning in dynamic environments. We have also embraced this principle for the human-robot cooperative navigation planning method that we introduce this chapter. As global planner we use the same module discussed in the last chapter that utilizes a costmap layer for imposing social costs during path planning phase. For local planning, we introduce a novel method for cooperatively planning robot trajectories and predicting human trajectories within the same optimization framework. In the cooperative planning framework we have brought several new cost-factors. We have also tested effectiveness and applicability of these cost-factors with comprehensive user-study and also compared our results with previous work in social navigation planning.

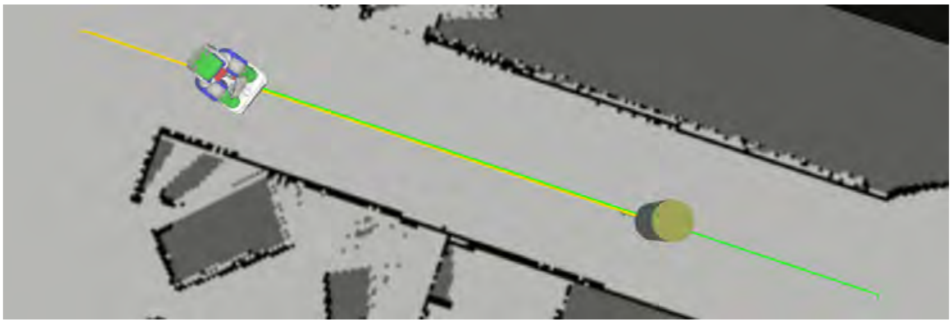


Figure 4.1 – A corridor-crossing scenario where the robot fails to find a path using standard planner when predicted human path (yellow) is blocking the way.

4.1 An Adaptive and Cooperative Navigation Planner

The most of the state-of-the-art human-aware navigation planners add proxemics costs around humans in a grid-based map representation of the robot operating environment [Kruse 2013; RiosMartinez 2014]. As we have seen in Chapter 2, using this costmap the path planning algorithms can generate paths that lower the total cost over the entire path, thereby keeping a safe distance from humans to maximize human comfort and visibility of the robot [Sisbot 2007]. The re-planning techniques used to cope up with dynamic situations are, however, only applied to smaller, immediate path of the robot to reach real-time compliance. The resulting robot motions are robust and safe but not necessarily social. The robot often oscillates or stops completely while moving near humans [Kruse 2014]. The requirement for human motion prediction arises when we intend to design a robot navigation system that is socially acceptable in dynamic settings where humans and robot are moving in a shared common space.

Prediction of human trajectories independent of robot plans, however, does not alleviate the problem of purely reactive robot behavior [Ferrer 2014b]. For example, consider a corridor situation where a robot and a person could only cross each other in a side-by-side configuration (figure 4.1). If the person is walking in the middle of the corridor, due to their predicted path, the robot will often fail to find a collision free trajectory using a simple reactive planner. Therefore, while predicting human behavior the planner has to consider that the humans do see the robot and will also try to contribute to collision avoidance with the robot by modifying their own trajectories. In other words, there is a need for a planner that can proactively suggest a solution to the cooperative navigation situations. The human behavior prediction scheme should not only consider dynamics of human motion but also the shared environment with common obstacles that both human and robot need to avoid. The more “natural” and “habitual” the predicted human path is, the more probable it is, in the sense that the it will correctly predict the human behavior.

The work presented in this section has been partly published in [Khambhaita 2017a] and [Khambhaita 2017c].

4.1.1 A Case for Cooperation

Figure 4.1 shows a typical case where both robot and human can safely and smoothly pass each other only if they cooperate and facilitate the other party by giving enough space to move. Thus, navigation in human environments is a cooperative task that a robot should treat as such. Humans concurrently aid and comply with each other while moving in a shared space. Cooperation helps pedestrians to efficiently reach their own goals and respect the personal space of others. To meet human comparable efficiency, a robot needs to predict the human trajectories and plan its own trajectory correspondingly in the same shared space. In this section, we present a navigation planner that is able to plan such cooperative trajectories, simultaneously enforcing the robot's kinematic constraints and avoiding other non-human dynamic obstacles using an optimal planning framework.

Taking inspiration from the *joint action* literature [Sebanz 2006; Clodic 2014] and from our previous contributions on robot planning abilities for human-robot task achievement [Lemaignan 2017], we propose a reactive navigation planner that builds and maintains a set of streams of execution for the robot and the humans in its close vicinity. Indeed, this research has shown that it is sometimes pertinent to endow the robot with the ability to plan not only for itself but also for its human partner. This ability takes its full meaning and pertinence when it is necessary that both act in order to solve a problem. In navigation, this corresponds to very constrained environments.

Research in psychology [Knoblich 2011] and philosophy [Pacherie 2012] have led to a good understanding of human behavior during joint actions and collaboration and have helped to identify the key elements for human-robot joint actions [Clodic 2014]. [Tomasello 2005] defines a goal as the representation of a desired state and an intention as an action plan chosen in order to reach the goal. [Bratman 1993] adds that if there is a *shared intention* to perform an action, the partners should agree on the meshing sub-parts of a *shared plan*, which is elaborated based on *common ground* [Clark 1983].

Thus we claim that navigation in a populated environment can generally be modeled as a kind of coordination activity, since each individual has his own goal and all share an environment. This very same activity can be transformed into a problem that needs cooperation of two or several individuals when the environment becomes very constrained: a given individual cannot find his path unless another individual participates and helps in finding a solution. Thus in a shared space, both the robot and the human have to take into account that avoiding a collision is a joint task. It is the duty of both to avoid a collision and help the other party to advance towards their destination. In a joint task all the agents need to find a common ground with known roles and abilities and use it to predict what the other will do [Lemaignan 2017].

This is exactly the kind of problems we want to tackle:

- We would like to develop a robot navigation system that is able to manage usual coordination issues, but that is also able to manage intricate situations.

- Besides, we would like, to come up with a scheme that allows the robot to be proactive by proposing an acceptable solution and, whenever possible, to take "most of the load" when the human and the robot have to share the load to solve a problem.
- And, finally, we would like the robot to take into account human acceptability and comfort issues.

With such scheme the robot computes a shared plan that includes probable paths for the humans and its own path, and continuously monitors the human actions to further rectify and improve the shared plan. Of course, robot and humans are not equal, therefore the shared plan entails as less effort as possible for humans.

One way to intertwine human motion prediction with robot motion planning is using an iterative simulation process where robot computes its paths for a fraction of total time-horizon base on current positions of human in the environment and uses this plan to predict new human positions for the same fraction of time. This process of planning and prediction repeats itself for a specified time-horizon [Ferrer 2014b]. With this scheme, effectively, prediction of human motion depends only on last planning iteration. The resulting robot behavior, although being locally correct, is sometimes sub-optimal when we look at the interaction episode as a whole.

While incorporating human motion prediction with the robot path planning scheme is a step towards socially acceptable navigation at each instance of planning cycle, we aspire to make the full interaction period agreeable by more tightly coupling prediction and planning stages. Moreover, first hand experiences of working from purely reactive to legibility enhancing and social force model based human-aware navigation planners convinced us that we will have to strive for integrating the human motion prediction within the planning algorithm. Thus, our approach of cooperative human-robot navigation focuses on providing a solution for entire human-robot co-navigation situation by combining robot trajectory calculation and human path prediction into a unified planning framework. Furthermore, the constraints that we are using also takes human velocity into account and it can predict a human path where the human can accelerate or slow down (within specified bound based on human motion kinematics). This method results into robot behavior that not only optimal for the entire interaction episode but also comfortable and legible for the humans.

Borrowing the elastic band notion from [Quinlan 1993], we imagined a planning situation where it can help the robot safely navigate around humans: a wide corridor where the robot and the human are crossing each other's path albeit from opposite direction. While entering at one end of the corridor the robot had already a calculated a feasible path to reach the other end of the corridor. However, as the robot encounters the human, the previously calculated path becomes invalid, since the human is blocking that path. Here the elastic band approach helps the robot to deform its path in such a way that it avoids colliding with the human, and at the same time avoiding collision with any other static obstacles in the environment such as walls. This situation assumes that the human would continue moving on their original path. Elastic band based navigation planner has already shown to produce collision free paths in relatively open spaces where the robot have bigger latitude to move away from the humans. Therefore, elastic band has become one of the most widely used navigation approaches for mobile robots.

Now, when we think of a narrower corridor where the human is moving in the middle of the corridor, the elastic band would fail to find any path beyond the human position as it cannot deform the robot path to pass on either side of the human because of the risk of collision with the corridor walls. This situation is similar to what we have already depicted in figure 4.1. Here, the assumption that 'the human will continue moving on their original path' is improbable and unrealistic. It is highly possible that the human would move little on either side of the corridor yielding enough space for the robot to pass. And, in fact, this is the human behavior that we have also observed time and again in our field tests with different navigation planners. We have seen that the humans act as their paths are also elastic and these paths can deform with similar constraints that make the robot path deform. Such is the expected motion of two humans crossing each others' path in a narrow corridor as well. In other words, humans cooperatively solve the navigation problem by sharing the workload among themselves. With this critical insight we started working on treating not only the robot path but also coarsely predicted human path as an 'elastic band'.

Figure 4.2 shows how the proposed cooperative planner would find a solution for the corridor crossing situation. It assigns an elastic band to both of the agents, the robot and the human. Then the cooperative planner simultaneously calculates a trajectory for the robot and predicts a plausible trajectory for the human. Here we can see that the planner only slightly deforms the predicted human trajectory that gives enough room to the robot to pass beside the human. Therefore in this corridor crossing situation, still the robot takes most of the workload by deforming its own trajectory to a greater extent.

Since we ascribe bands to each social agent in the environment, we are now in a place to regulate the elasticity of these bands separately. In this manner, this approach provides additional advantage of streamlining robot navigation behavior from open spaces to very constrained environmental conditions.

Furthermore, in highly confined spaces sometime robot does not have any latitude to move until human gives it some space. An example of such situation is when robot need to move to a connected room through a door but a human is blocking the door. With the propose scheme the robot is able to predict a trajectory of the human where human momentarily moves away and unblocks the door for the robot to pass, the robot plan involves waiting at the door until the human gives a way. Such waiting behavior is not explicitly programmed in to the robot memory, rather it is the outcome of the same generic cooperative planner that also solves the corridor crossing problem described above. And thinking about how a robot should solve door crossing problem such waiting behavior seems logical, similar to how

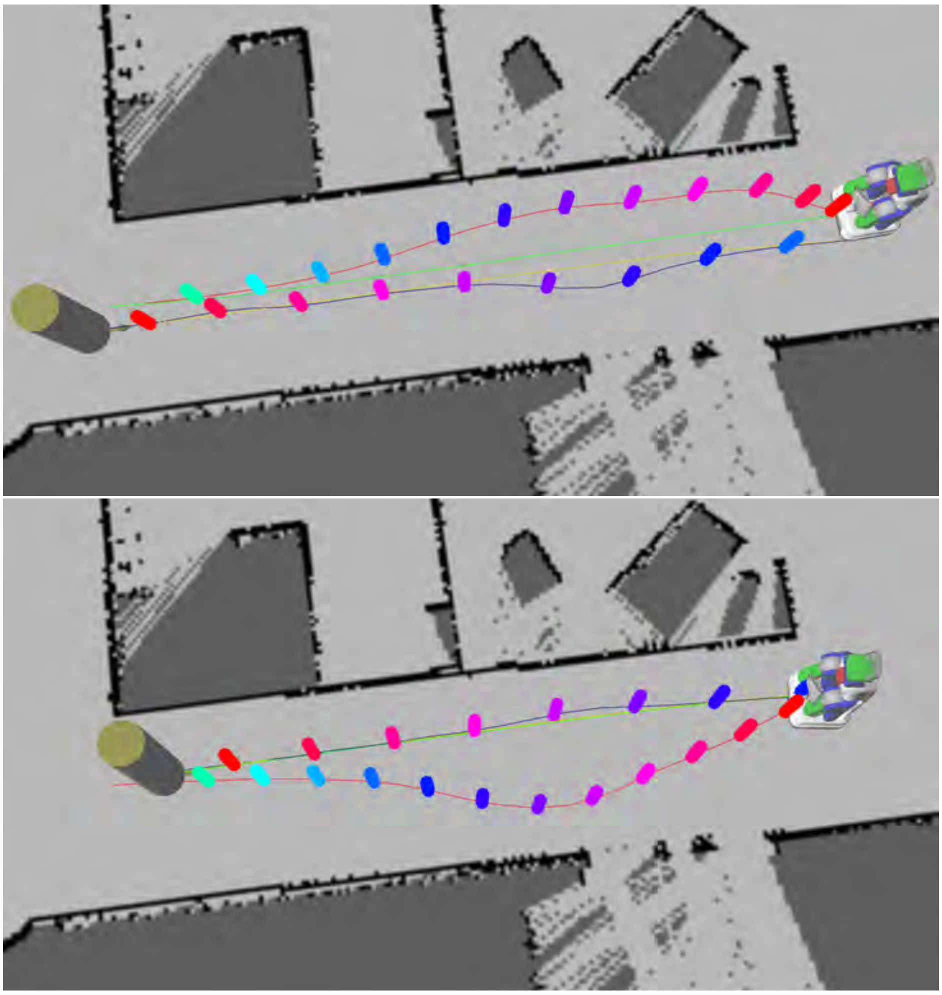


Figure 4.2 – A corridor-crossing scenario with proposed cooperative planner. The robot is able to calculate its own trajectory (in red) and proposes a trajectory for the human (in blue) that solves the co-navigation problem. Here the robot assumes that human wants to go to the other end of the corridor. We have added cylindrical shaped landmarks on the planned robot trajectory and predicted human trajectory by the cooperative planner. The landmarks shows future human and robot positions at every second. The color of the landmark on the human and robot trajectories correspond to same future time. We can see that the robot moves to its right well in advance (even if it is not absolutely necessary at the moment the robot sees the human). With such proactive behavior the robot is offering the solution to the co-navigation situation. The robot does not accelerate until the human passes so that the human do not feel threatened, in this example it passes the human at approximately 0.6 m/s and later accelerates to 0.8 m/s. If the human reject the solution (bottom figure) suggested by the robot and move in the opposite direction to what the robot has predicted, the robot is able to quickly react by changing its path that adheres to human's wish. Robot's global path is shown in green, where as a velocity based prediction of human path is shown in yellow.

humans would behave in comparable situations. Due to this waiting behavior the cases where the planner fails to find a path are highly reduced. With cooperative planner, waiting beside the door becomes part of the robots navigation plan.

With the proposed planner a solution does not only include the contributions of the robot but also of the human. Hence, the robot is able to envision the solution to the complete co-navigation problem and acts accordingly. This is why we claim that our system is a human-robot cooperative planner. Besides, we will also show that human and robot are not treated equally in this scheme. Generally the robot takes, when possible, all or most of the effort to avoid collision with humans.

Concretely, the proposed cooperative navigation planner predicts a plausible trajectory for the humans and accordingly plans for a robot trajectory that satisfies a set of constraints. These constraints and their relative weight eventually determines how the planner deforms the elastic band trajectories. Apart from usual kinodynamic constraints we have introduced a set of social constraints that make the overall robot navigation behavior socially adequate. These social constraints include projected time to a possible future collision, compatibility of human-robot motion direction, and proxemics. By adding the kinodynamic and the social constraints we turn the trajectory generation into a non-linear least squares optimization problem. It is crucial that generation of the robot and human trajectories is represented as a solitary multi-constrained problem. The optimization problem is, in turn, solved using an iterative algorithm. We use an optimization library that represents trajectory points as nodes and constraints between them as edges of a graph structure, hence falls into the category of graph-based optimization solvers. It generates both robot and human trajectories, thus facilitating both agents to avoid any other static or dynamic obstacle present in the shared space.

The proxemics based constraint ensures safety distance between the robot and the humans throughout their optimized trajectories. The directional compatibility constraint takes inspiration from the *ContextCost* model described in Section 3.2.1, which increases the perceived safety and comfort felt by humans. While recent advances in human locomotion research motivates our third social constraint, namely the *time-to-collision* constraint. The *time-to-collision* constraint makes the robot behave proactively, that is, taking motion decisions early enough such that they are easily recognized by the humans. With these constraints our planner is able to mirror human-like navigation behavior not only in open spaces but also in confined areas.

One of the key aspect of the proposed planner is that apart from adapting the robot trajectory, the planner is also able to propose co-navigation solutions by jointly computing human and robot trajectories within the same optimization framework. We demonstrate richness and performance of the cooperative planner with simulated and real world experiments on multiple interactive navigation scenarios. Our experiments also show improvements in the fluency of the interaction. Our approach aims to balance and tune the efforts between the human and the robot to solve a co-navigation task, in its spirit similar to the previously proposed approaches for geometric [Waldhart 2015] and symbolic [Shah 2011] planning systems.

Thus, this work on cooperative planner provides following key contributions:

1. An optimization based framework for computing the robot trajectory and predicting probable trajectories of nearby humans that respect motion and social constraints.

2. Judiciously devised social constraints for (a) safety, (b) time-to-collision and (c) directional compatibility of human-robot motion.

3. A demonstration of the success of proposed approach in everyday interactive navigation situations, especially in confined spaces.

4.1.2 Elastic Bands for Navigation

The elastic band approach, introduced by [Quinlan 1993] only locally modifies the robot path to keep a safe distance from previously unknown obstacles, thus suitable as a local planner. However, the modified path often does not satisfy the kinodynamic constraints of the robot. Therefore, a general scheme is to use a controller module that takes the output path (elastic band) and generates feasible trajectories that the robot can follow [Khatib 1997]. Recent proposal of *timed elastic band* evades this problem by explicitly considering temporal information [Rösmann 2012]. It locally deforms the robot path and computes a trajectory augmented with a series of time-difference values between each successive poses, instead of a purely geometric path. *Timed elastic band* makes it easy to take kinodynamic and nonholonomic constraints into account, formalizing the optimization problems as a non-linear least squares problem.

We have substantially extended this work by introducing prediction and optimization of human trajectories in the same framework. Predicted human trajectories are also subject to kinodynamic constraints of human locomotion. Here the robot assumes a nominal velocity for the humans when predicting their trajectories, plus it also accounts for people slowing down or speeding up according to the situation. By changing the nominal velocity of the humans while predicting human trajectories, the robot can take into account people who prefer to walk slow or rushing because of some time pressure. Thus, the overall optimization problem that we are solving not only enforce kinodynamic constraints for human and robot but also plans for dynamic changes in the environment. Besides, we have brought in carefully selected social constraints into the optimization framework. These constraints not only keep the robot at a safe distance from human throughout its trajectory but also allow the robot to show proactive and legible navigation behaviors such as moving to a side of a corridor sufficiently ahead of time or waiting at doors that allow humans to comfortably pass.

It is clear that robot navigation among humans requires minimizing multiple cost-functions using some optimization framework. We argue that it is imperative to also include prediction of plausible human trajectories within the same optimization framework. We address the social navigation task by applying least squares optimization to simultaneously minimize multiple cost-functions that represent costs associated with human-robot cooperation as well as robot dynamics. And we reiterate that the optimization framework explicitly includes prediction of plausible trajectories within the same non-linear least squares problem.

Compared to earlier work [Rösmann 2013], we create *elastic bands* also for the humans and extend their hyper-graph structure (figure 4.3) to handle humans separately from ordinary obstacles using well-grounded *human-aware planning constraints*. The resulting scheme combines in one step the *robot-plans*, *human-plans* and *robot-reacts* process. Using this scheme a robot is able to balance between *changing-path* and *changing-speed* options to achieve human legible behaviors.

4.1.3 Optimizing Robot Trajectories

The *timed elastic band* approach [Rösmann 2012] augments each of the $n \in \mathbb{N}$ poses of the robot path with time interval between each consecutive poses. The resulting trajectory forms a tuple, a list of n trajectory points and a list of $n-1$ time-difference values between each consecutive trajectory points. One full trajectory (denoted as $\mathcal{B}$), is called a *timed elastic band*. This *timed elastic band* is subject to deformation by the optimization algorithm.

$$\mathcal{B} := \{\mathcal{P}, \mathcal{T}\}$$

where, the $\mathcal{P}$ is the 2D pose sequence with orientation of the trajectory,

$$\mathcal{P} := \left\{ p_i = [x_i, y_i, \theta_i]^T \right\}_{i=0}^{n}$$

and $\mathcal{T}$ is a sequence representing the time intervals.

$$\mathcal{T} := \{\Delta t_j\}_{j=0}^{n-1}$$

In addition to the robot trajectory $\mathcal{B}_{\mathcal{R}}$, we also represent a set of human trajectories $\{\mathcal{B}_{\mathcal{H}_k}\}_{k=0}^{m}$ as *timed elastic bands*, when there are $m \in \mathbb{N}$ humans in the robot vicinity.

The gist of our approach is to jointly optimize the robot and all human trajectories in terms of social and kinodynamic constraints. It requires to solve the following multivariate multi-objective optimization problem using the weighted-sum model:

$$f(\mathcal{B}_{\mathcal{R}}, \mathcal{B}_{\mathcal{H}_k}) = \sum_a \gamma_a f_a(\mathcal{B}_{\mathcal{R}}) + \sum_b \gamma_b f_b(\mathcal{B}_{\mathcal{H}_k}) + \sum_c \gamma_c f_c(\mathcal{B}_{\mathcal{R}}, \mathcal{B}_{\mathcal{H}_k}) \tag{4.1}$$

$$\{\mathcal{B}^*_{\mathcal{R}}, \mathcal{B}^*_{\mathcal{H}_k}\} = \underset{\{\mathcal{B}_{\mathcal{R}}, \mathcal{B}_{\mathcal{H}_k}\}}{\arg\min} f(\mathcal{B}_{\mathcal{R}}, \mathcal{B}_{\mathcal{H}_k}) \tag{4.2}$$

where $\{\mathcal{B}^*_{\mathcal{R}}, \mathcal{B}^*_{\mathcal{H}_k}\}$ denotes the set of optimized robot and human trajectories. The component objective functions for the robot trajectory, the human trajectories and the human-robot social constraints are denoted by f_a, f_b and f_c respectively.

We inherit the kinodynamic and nonholonomic constraints from [Rösmann 2012] imposed on the robot trajectory. These constraints:

- Enforce physical limits of velocity and acceleration between consecutive poses of the robot trajectory.
- Keep the robot at a minimum clearance distance from obstacles for each of the poses.
- Make the robot move near to the globally planned path.
- And, prefer fastest execution time for the whole robot trajectory.

We have added optimization of human trajectories to the overall optimization problem. Similar to the robot kinodynamic constraints we impose kinodynamic constraints on each human trajectory, with human velocity and acceleration limits obtained from empirical studies of pedestrians interaction data [Bohannon 1997]. However, while predicting the human trajectories the optimization algorithm tries to maintain nominal human velocity (and not the fastest velocity that human can move with). Constraints on human-robot cooperative motion are discussed in detail in Sec. 4.1.4. To summarize, the newly added constraints:

- Enforce physical limits of velocity and acceleration between consecutive poses of the each human trajectory.
- Prefer nominal velocity (which are lower than maximum velocity) for the humans.
- Keep each pose of the predicted human trajectories at a minimum clearance distance from obstacles.
- Make the humans move near to their globally predicted paths towards their goal positions.
- Prefer fastest execution time for the whole human trajectories (and withal respecting the nominal human velocity).
- Keep each pose of the predicted human trajectories at a minimum safety distance from the each pose of the robot trajectory with respect to their respective time step. That is, a safety distance is ensured between the human and the robot poses at one second in the future, at two seconds in the future and so on.
- Similar to keeping the safety distance between the robot and the human trajectories, also keep the specified safety distance between each pair of two human trajectories.

- Ensure that the time to predicted future collision does not go below specified limit for respective poses in time of the robot and all the human trajectories.
- Assure that robot and all the human trajectories remain compatible according to the compatibility measure discussed in Section 3.2.1.

Since, the resulting trajectory is an optimally deformed version of the initial path, this representation of trajectory is similar to an *elastic band*. The key aspect of proposed framework is to predict optimized human trajectories by the same hyper-graph. We have multiple *timed elastic bands*, one for the robot and one for each of the humans, that are optimized simultaneously. By adjusting weights on the constraints for robot and humans separately (the parameters γ_a and γ_b in equation 4.1), we can balance and tune the "tightness" of the elastic bands for effective effort sharing.

Introduction of all social and kinodynamic constraints for robot and humans results into a non-linear least squares optimization problem. Following the work of [Rösmann 2012], we have also adopted the general optimization framework g^{2}o [Kümmerle 2011] which requires mapping of the least squares problem into a graph representation. The g^{2}o software library makes it easy to introduce new constraints to the optimization problem by writing an error functions with specified interfaces for each constraint. The graph structure consists of nodes and edges that connect these nodes. A node represents a state variable to optimize and an edge represents an optimization constraint (in terms of error function). The framework makes it possible to write pair-wise separate error functions (that is, affecting only the node that an edge connects) for each of the optimization constraints. The edges that connect two nodes represent social and kinodynamic constraints, as shown in figure 4.3.

For the problem at hand each node in the graph represents a pose along the trajectory of either the robot or a human. However, the nodes represented with double circle in figure 4.3 are there only for depiction purposes. They are different from other nodes in the manner that position of these nodes cannot be modified by the optimization algorithm. In fact, the error function of the edge that connects a trajectory node with static obstacle node directly calls the collision-check subroutine that returns the distance of the trajectory node from the nearest obstacle in the environment.

The graph structure of figure 4.3 incorporates all of the kinodynamic and social constraints. The bottom row, containing rx_0, rx_1, . . . as well as $r\Delta T_0$ and o_1 nodes with c_{vel_r}, c_{acc_r}, c_{kin_r} and the bottom c_{obs} edges represents the structure similar to the one introduced by [Rösmann 2012]. The cooperative planner adds all other nodes relating to the human trajectory (h_0x_0, h_0x_1, h_1x_0, h_1x_1,. . .) and time difference between consecutive human trajectory poses ($h_0\Delta T_0$, $h_1\Delta T_0$, . . .). It also adds kinodynamic constraint edges (c_{vel_h}, c_{acc_h}, c_{kin_h}) and social constraint edges (c_{safety}, c_{ttc}, c_{dir}) as well as the c_{sep} edges for safety distance separation between humans

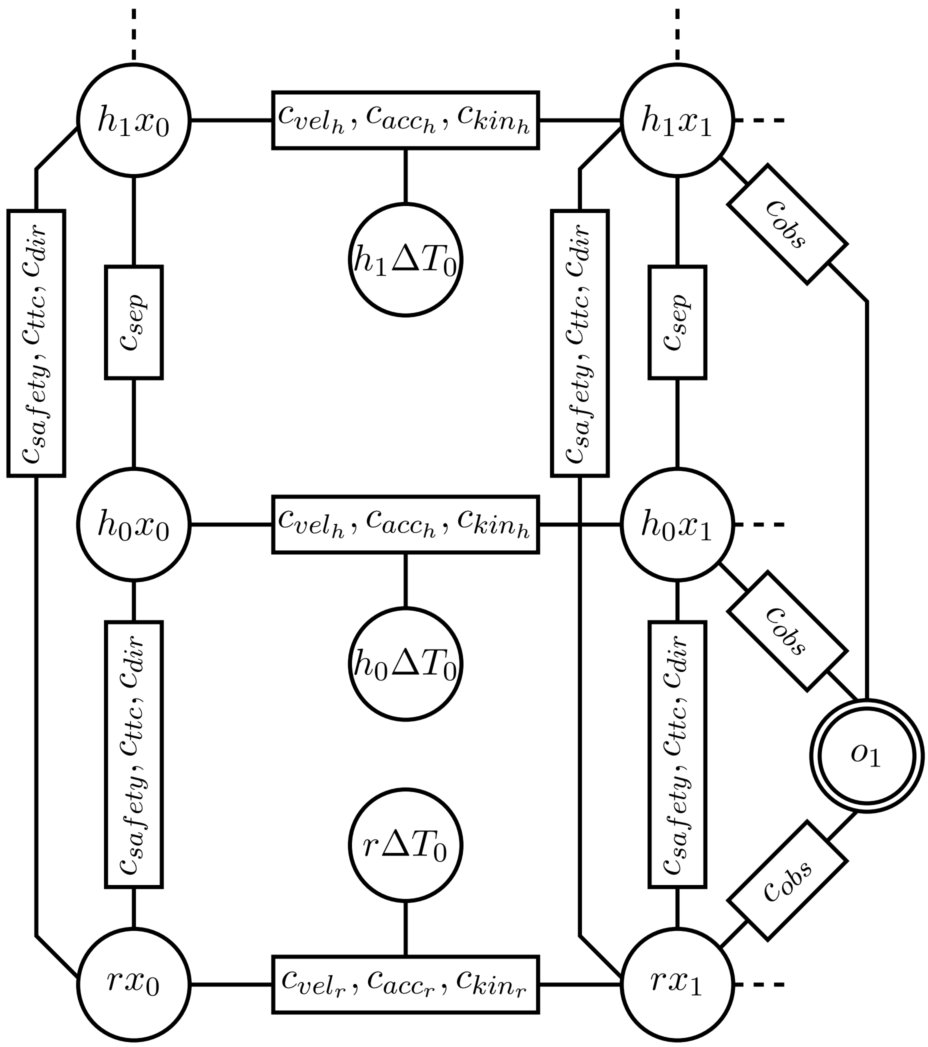


Figure 4.3 – Graph structure. The bottom row has consecutive nodes for robot trajectory (rx_0, rx_1, ...); edges enforcing velocity, acceleration, and kinodynamic constraints connects these nodes. We combined the three edges into one for easier depiction. Penalty imposed by these edges depends on time difference between consecutive nodes, therefore the *time-diff* node $r\Delta T_0$ connects with them. Pose and *time-diff* nodes are subject to change by the optimization process. Similarly, the middle and the top row represent trajectories for humans 0 and 1 respectively, albeit with different weights for constraints. Obstacles node (o_1), shown with double circle, is a fixed node, meaning the optimization process cannot alter its position. The edge c_{obs} shows the constraint for keeping minimum distance from obstacles. The edge c_{sep} represents the constraint to keep minimum separation between two humans. Nodes of the robot and a human that belong to the same time-step of their trajectories are connected by three edges (c_{safety}, c_{ttc} and c_{dir}) that impose social constraints.

and further c_{obs} edges for ensuring safety distance between obstacles and predicted human trajectories. Introduction of all these new nodes and edges now makes the graph structure suitable for cooperative planner.

Regarding the equation 4.1, the constraint edges c_{vel_r}, c_{acc_r}, c_{kin_r} and c_{obs} are types of function f_a that operates on robot's *timed elastic band* ($\mathcal{B}_{\mathcal{R}}$) constructed from rx_0, rx_1, ... as well as $r\Delta T_0$, $r\Delta T_1$, ... nodes. Similarly, the f_b type of functions are realized by the constraint edges c_{vel_h}, c_{acc_h}, c_{kin_h} and c_{obs}, and it operates on human's *timed elastic bands* ($\mathcal{B}_{\mathcal{H}_k}$) constructed from h_0x_0, h_0x_1, h_1x_0, h_1x_1,... as well as $h_0\Delta T_0$, $h_1\Delta T_0$, ... nodes. Remaining constraint edges c_{safety}, c_{ttc}, c_{dir} and c_{sep} typifies f_c functions that operates either between human and robot or each pairs of human *timed elastic bands*.

The graph based structure provides and easier interface to initialize and maintain the composition of variables and constraints used during the optimization process. One can see the graph structure as reducer of the non-linear least squares problem to a form with linear system of equations around the current state and iteratively solving it solves the original non-linear least squares problem. The g^{2}o framework employs Levenberg-Marquardt algorithm to solve the non-linear least squares problem defined by the graph structure. The Levenberg-Marquardt algorithm (also known as damped least squares) can be thought of as a linear combination of gradient-descent and Gauss-Newton methods, where the damping-factor introduction makes it balance between the two methods. Thus it combines advantages of gradient-descent and Gauss-Newton methods such that it increases convergence chance and prohibits divergence.

Once we write down the error functions for each state variables, the graph structure generated by g^{2}o framework represents the objective function for the least squares problem. The optimization framework then calculates the matrix of the second order derivative of the objective function (also known as the Hessian matrix or the information matrix). The iterative process of Levenberg-Marquardt algorithm uses this information matrix to converge towards the optimal solution by adding increments to the initial guess. Since the *local planner* itself runs at a certain frequency (generally about 10 Hz), it allows us to use the solution calculated in last iteration as the initial guess to the optimization algorithm at hand in current iteration. A second option is to use the path segment of the *global plan* that falls within the *local costmap* as initial guess. For cooperative planner, generally the second option works best in our tests.

When the optimization runs for the first time it takes the *global plan* of the robot to initialize the *timed elastic band*. The initialization process adds position nodes in the *timed elastic band* spaced at a specified distance from each other and it adds one time-difference node between each pair of position nodes. For the optimized trajectory to remain as close as possible to the *global plan*, the graph structure also includes a constraint to minimize the distance between optimized position node and their initial position (which corresponds to the *global plan*). Similarly, the graph structure also adds constraints to minimize the distance between each of the predicted (optimized) human trajectories with their corresponding human *global plans*. However, for simplicity, these particular constraints are not depicted in terms of edges in the figure 4.3.

Since most of the constraints are local to certain pose and a few neighboring configurations, the information matrix is typically sparse. Consequently the g^{2}o framework uses state-of-the-art sparse linear system solvers [Davis 2006]. Result of the optimization adjusts the position and orientation of each of the poses as well as time difference between the consecutive poses of the robot trajectory and predicted human trajectories such that each trajectories minimize the imposed constraints. This iterative optimization process solves the equation 4.2 and the result set contains an optimized robot trajectory (denoted by $\mathcal{B}^*_{\mathcal{R}}$) and predicted optimal human trajectories (denoted by $\mathcal{B}^*_{\mathcal{H}_k}$).

Contrary to [Rösmann 2012], in our approach we have modified the error function used for safety-clearance from obstacles as following,

$$f_{obs}(d, d_o, \epsilon, S) = \begin{cases} \dfrac{(d_o + \epsilon) - d}{Sd + 1} & \text{if} \quad 0 \leqslant d < (d_o + \epsilon) \\ (d_o + \epsilon) - d & \text{if} \quad d < 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

where d is the distance between obstacle and a robot pose during current iteration of the optimization process, d_o is lower bound for obstacle clearance, ϵ is a parameter to control the accuracy of the approximation. Equation 4.3 is non-linear between the lower bound and zero, and the parameter S adjust the non-linearity. Figure 4.4 plots the obstacle function f_{obs}, showing how it varies with respect to d_0 and S. Such construction of the error function enables us to manipulate relative importance of safety clearance between robot-obstacle and robot-human. That is, we can make robot *push* itself more towards an obstacle rather than towards a human in constrained situations. We are of the opinion that a human-aware navigation planner has to treat humans differently from other ordinary obstacles, which has motivated us to choose such error function for the *safety* constraints.

The optimization process runs in two computation loops. *Inner loop* corresponds to iterative loop for the least squares solver, that is, the loop of iterative process of the Levenberg-Marquardt algorithm. The user specifies the number of iterations for this inner loop. This is an optimization parameter, in our experience between five to eight inner iterations gives good results.

After each full run of the inner loop, the optimization process updates the graph structure using latest result of the solver. During this update, it adds new nodes between two nodes in the graph if the time-difference between the two nodes exceeds certain threshold (which an optimization parameter) in order to maintain similar time difference between each pair of neighbor nodes. Thus at each outer loop the trajectory is re-adjusted such that the time difference between each pair of neighboring poses are same as specified value for the temporal resolution of the

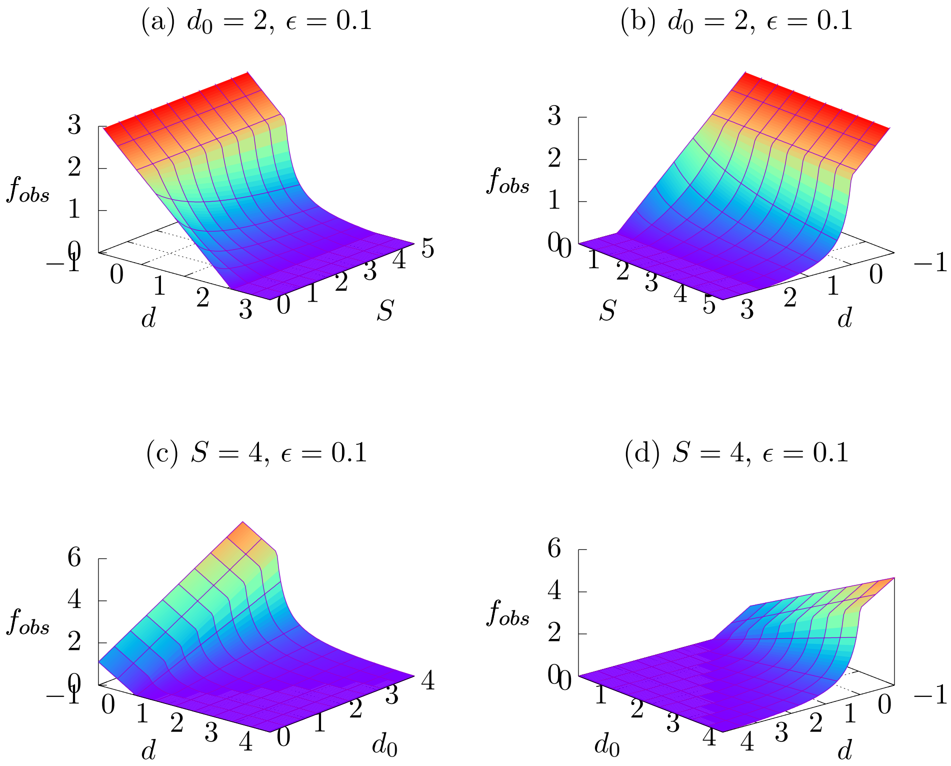


Figure 4.4 – Error function used for safety clearance. Figures (a) and (b) shows how f_{obs} of equation 4.3 varies when d_0 is fixed. Figures (c) and (d) shows variability of f_{obs} when value of S is fixed.

trajectory. Therefore, the human and robot time-difference nodes, shown as $h_0\Delta T_0$ and $r\Delta T_0$ in the graph structure are synchronized during the *outer loop*. In our experiments about four outer loop iterations give good results. For the newly added position nodes the outer loop also adds a constraint-edge that minimizes distance of optimized position node with nearest node on the initial *timed elastic band*, similar to the constraint-edges that are added during initialization, as discussed earlier.

This synchronization is necessary for each of the social constraints (that is, constraints between human and robot trajectories) to produce the intended result. In our implementation we apply these social constraints between respective poses (in terms of time from start of the trajectory) of human and robot trajectories. Another option would be to have the same time-difference node between the human and robot trajectories (specifically, by merging $h_0\Delta T_0$ and $r\Delta T_0$ into one node). However, since the length of robot trajectory and each human trajectory is different, we prefer the synchronization of time-differences that happen during the outer loop.

Both inner and outer loops run for several iterations. Thus, the multiplication of inner and outer loop iteration number is the total number of times the least squares optimization process runs. The number of iterations directly affects produced trajectories and optimization time. Also, the threshold for time-difference nodes for removing or adding new nodes affects the quality of the resulting trajectories and the number of inner and outer loop iterations required. Smaller values of this threshold requires more iterations for the optimization process to converge. A value of 0.3 s gave good results for different tests that we carried out to evaluate the cooperative planner performance.

4.1.4 Social Constraints

We have selected the above mentioned graph-based solver because it enables us to introduce the social constraints and rules which are appropriate for efficient human-robot cooperative planning. Since we have the whole trajectories of human and robot at our disposal, we have added social constraints between human and robot nodes in the graph structure that correspond to the same time-step during their trajectories. That means, we add an edge for the *safety* constraint between n^{th} node of human and n^{th} node of robot trajectory (which corresponds to poses at same future time), another edge between $n+1^{\text{th}}$ nodes, and so on. Similarly, we also add edges for *time-to-collision* and *directional* constraints between corresponding poses of human and robot trajectories. Nevertheless, we only add edges to the nodes corresponding to part of the human path that falls within the *local planning* area centered at the robot.

It is needless to say that we only add the social constraints between corresponding human-robot nodes for up to the length of the shorter trajectory in each pair. Now we will look into the definition of these three social constraints.

(a) Safety Constraint

In extension to the earlier work on planning safe robot trajectories [Sisbot 2007], we also define a *safety* constraint in optimization framework around the trajectories of moving humans. The *safety* constraints uses proxemics based cost function to ensure minimum safety distance between corresponding human and robot poses. However, to avoid over-cautious behavior of the robot, as explained in the last section, we only apply the safety costs around the respective points in time along the human-robot trajectory pairs. That means, the optimization procedure ensures given safety distance between planned position of the robot and predicted position of the human at all future time points $t = 1$, $t = 2$, … up to the planning horizon.

Therefore the error function associated with the *safety* constraint is,

$$f_{safety}(d, d_s, \epsilon) = \begin{cases} (d_s + \epsilon) - d & \text{if} \quad d < (d_s + \epsilon) \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

where d_s is a lower bound on allowed safe distance between human and robot poses at the same time-stamp during the trajectory. d is the outer distance between human and robot.

The outer safety distance is calculated from the boundary of the robot to the boundary of the human footprint. There are multiple ways one can represent the footprint of the robot in this planning scheme, from simple circle to complex polygon. For faster distance calculations, we represent the human footprint as circle with a specified radius. Calculating the outer distance between a polygon shaped robot and a circle shaped human is computationally more expensive than calculating outer distance between human and robot that are both circle shaped. Thus, it is advisable to represent the robot with a simpler footprint.

The graph structure injects an edge for this *safety* constraint not only to each human-robot trajectory pair but also to each human-human trajectory pair, albeit with different parameters for the error function. The c_{sep} edges in the hyper-graph structure of figure 4.3 shows this constraint to have minimum separation between each pair of humans. Such construction of the graph structure ensures that the optimization process respects proxemics cost between humans as well. In our experiments we generally keep the safety distance between humans and the robot slightly higher compared to the safety distance between pairs of humans.

A nice side-effect of this human-human *safety* constraint was robot planning a queue forming behavior for humans in very constrained corridor crossing situations when only one human and the robot can pass on each-others side. Here, the robot was planning for solving the navigation situation where one human moves behind another human at a specified separation distance.

(b) Time-to-Collision Constraint

A novel social constraint used in the proposed scheme is *time-to-collision*, that is, the projected time to a possible future collision with a human. Empirical studies have shown that *time-to-collision* between self and other governs the pedestrian interaction across wide variety of situation [Karamouzas 2014]. We make use of these results by applying higher cost to human-robot configurations that result in smaller *time-to-collision* during the optimization.

Our hypothesis is, the *time-to-collision* constraint will push the robot to act early enough, thus clearly showing the robot motion intentions to the human counterpart. In other words, with the *time-to-collision* constraint the robot will not wait until the last-minute, before making a way for the human. Such behavior should give an early signal to the nearly humans that the robot has already taken care of any possible collision with them and is already altered its path to avoid it.

The error function associated with the *time-to-collision* constraint is,

$$f_{ttc}(ttc, \tau, \epsilon) = \begin{cases} \dfrac{[(\tau + \epsilon) - ttc]\, \alpha}{C^2} & \text{if} \quad ttc < (\tau + \epsilon) \\ 0 & \text{otherwise (including } ttc = \infty) \end{cases} \tag{4.5}$$

where τ is the lower bound on time at which the robot predicts a collision occurring with the human. In other words, $\tau = 8$ indicates that the robot starts adding costs between the configuration (node) of human and robot whenever the computed time-

Algorithm 4.1 ComputeTTC ($\vec{C}$, $\overrightarrow{vel_r}$, $\overrightarrow{vel_h}$, R)

if $\vec{C} \cdot \vec{C} \leq R^2$ **then**
  $ttc = 0$
else
  $\vec{V} = \overrightarrow{vel_r} - \overrightarrow{vel_h}$
  if $\vec{C} \cdot \vec{V} > 0$ **then**
    $f = \left(\vec{C} \cdot \vec{V}\right)^2 - \left(\vec{V} \cdot \vec{V} \times \left(\vec{C} \cdot \vec{C} - R^2\right)\right)$
    if $f > 0$ **then**
      $ttc = \dfrac{\vec{C} \cdot \vec{V} - \sqrt{f}}{\vec{V} \cdot \vec{V}}$
    else
      $ttc = \infty$
    end if
  else
    $ttc = \infty$
  end if
end if
return ttc

to-collision goes below 8 s. The parameter α is a scaling parameter for strengthening or weakening the penalty on particular human robot configurations due to computed time-to-collision and C^2 is the squared distance between human and robot center points.

ttc is the predicted time-to-collision between a human and the robot. For calculating the value of *ttc* we consider both human and robot having a disc-like shape, however, with different radius. Then the time-to-collision defined as the time when the boundary of these two moving disc meet based on their current linear velocities. Value of *ttc* is computed using algorithm 4.1, where $\vec{C}$ is the 2D vector from human center point to the robot center point, $\overrightarrow{vel_r}$ and $\overrightarrow{vel_h}$ are velocity vectors for robot and human velocities respectively and R is the sum of robot and human radius.

With this constraint our robot is able to proactively propose, a co-navigation solution sufficiently well ahead of time compared to other state-of-the-art approaches. Figure 4.18 shows an example of trajectories with and without the *time-to-collision* constraints enabled.

(c) Directional Constraint

We have also added the *directional* constraint, which makes use of a compatibility measure between human and robot configurations. Motivation for this social constraint comes from the user study of Section 3.2.4. Likewise the user study results, this constraint for the cooperative planner aims to make the robot motions more legible, especially in human-robot path crossing situations.

Similar to equation 4.4, the directional constraint is also lower bound by a threshold ς, and defined as,

$$f_{dir}(cost_{dir}, \varsigma, \epsilon) = \begin{cases} (\varsigma + \epsilon) - cost_{dir} & \text{if} \quad cost_{dir} < (\varsigma + \epsilon) \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where the directional cost is defined as,

$$cost_{dir} = \frac{\overrightarrow{v_{\mathcal{R}}} \cdot \overrightarrow{p_{\mathcal{R}} p_{\mathcal{H}}} + \overrightarrow{v_{\mathcal{H}}} \cdot \overrightarrow{p_{\mathcal{H}} p_{\mathcal{R}}}}{C^2}$$

$\overrightarrow{v_{\mathcal{R}}}$ and $\overrightarrow{v_{\mathcal{H}}}$ are robot and human velocity vectors respectively, $\overrightarrow{p_{\mathcal{R}} p_{\mathcal{H}}}$ defines the vector from robot position to human position in 2D vector space and C^2 is the squared distance between center points of robot and human. This measure penalizes motions where human and robot are very near and moving straight towards each other. Moreover, high relative velocity means higher penalty values.

The incentive for using these social constraints comes from our past experiences with different cost-functions used for different forms of human-aware navigation planners as well as recent studies of human locomotion behavior. We consider that each of the social constraint augments to the human-awareness behavior shown by our robot. Simultaneous use of these social constraints makes the robot behavior not only safe but also proactive and legible for the human. With these social constraints we have tested and evaluated the proposed planner's performance in simulation and on two real robotic platforms. The Sec. 4.1.6 discusses the results of our tests in detail.

It should be noted that the social constraints discussed here are suitable for the task where robot is navigating autonomously or carrying out a task such as guiding a person. For other social task, for example approaching a person, one need to define different social constraints. However, the framework laid out here makes it feasible to easily add task-specific social constraints.

4.1.5 Software Framework

The elastic band approach can only locally deforms the robot trajectory, thus it requires an initial path to bootstrap the optimization. A simple grid-based *global* planning algorithm, for example A^*, is suitable for computing this initial path.

We have adopted `move_base` for our robot navigation architecture within which the proposed human-robot cooperative planner fits as a *local planner*. Figure 4.5 depicts the full navigation architecture.

From our experience of moving the robot in semi-crowded environment like airports, we have learned that it is preferable to reason differently for moving and static humans in the robot operating environment. If position of any human is known at the time of global planning, it is better to incorporate the proxemics costs

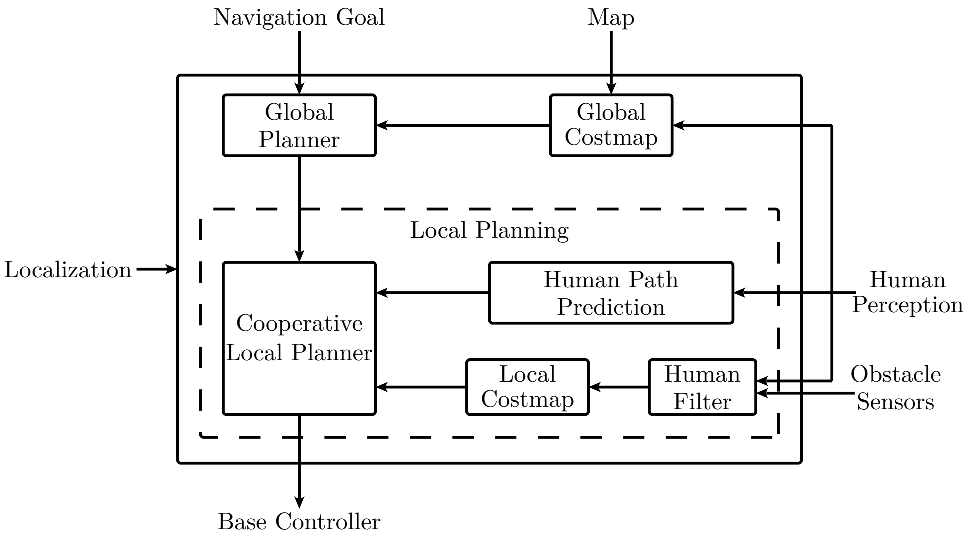


Figure 4.5 – Overview of the cooperative navigation software architecture.

around static humans already while calculating the global path. This way one can reduces the burden of avoiding static humans from the local planner. Hence, we have developed a plugin for `layered_costmap` library [Lu 2014b] (which is part of the `move_base` framework) that add safety and visibility costs introduced by [Sisbot 2007] around human positions in the occupancy grid-map.

All moving humans in the robot environment are handled by the cooperative planner (the local planner in our scheme). As we have explained earlier, the cooperative planner uses a global path to bootstrap the timed elastic band. This means we not only need to feed the cooperative planner with the global plan of the robot but also global plans for the humans. We are using a separate module to predict the first paths for tracked humans in the robot environment. This allows us to switch between different prediction methods depending on the interactive situation at hand. For example, in a corridor crossing situation we use the heuristic that the most probable goal of the human is to go to other side of the corridor and so we create and imaginary goal position behind the current robot position when a human was first detected. In open areas, we use the velocity-obstacle method [Fiorini 1998] to generate a short-term path for the tracked humans. Once this module selects the goal for each human, it uses the same A^*-based path search algorithm that is used by the robot global planner to find a path from human's current position to the goal position.

Since in this cooperative planning scheme we are treating humans differently than other obstacles in the environment, we filter out the human position data from the obstacles detection sensors (laser scanner in out experiments). This is done using a filter between the raw laser scanner data coming from the sensor and filtered data going to the `move_base` framework. This filter simply removes the laser scan points that fall within a specified radius from the tracked human positions.

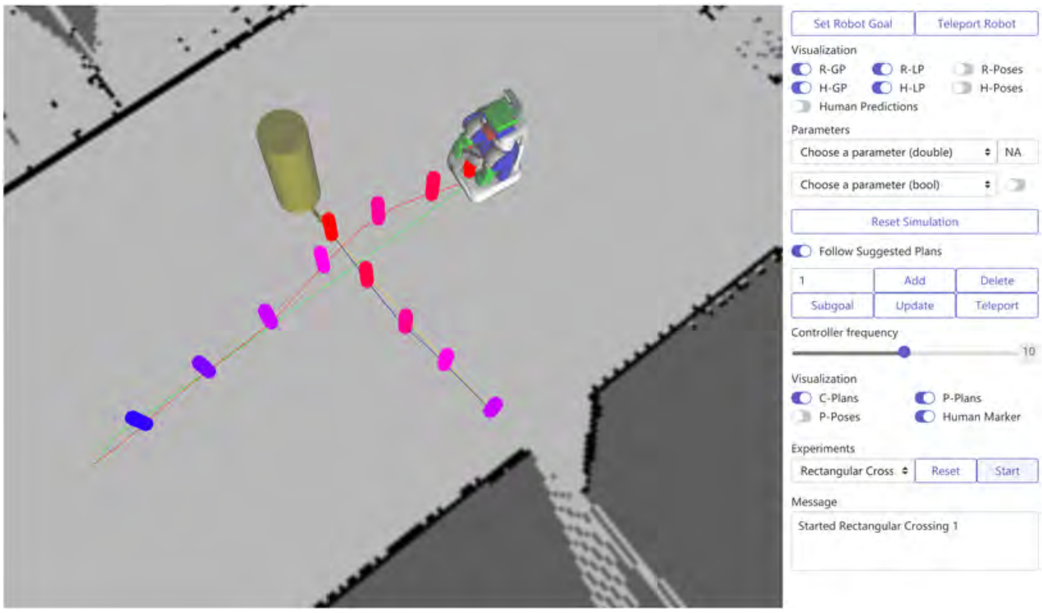


Figure 4.6 – Interface for testing the cooperative navigation planner in simulation.

4.1.6 Experiments

Before conducting experiments in the real world, we tested the proposed social constraints and validated them in a simulation environment. We have used the human motion simulator described in Section 3.1.3. By exposing the human trajectories from the cooperative planner to the human navigation simulator as new paths for the humans to follow, we can simulate full interactive navigation situations.

The human simulator can simulate motion of multiple humans, thus allowing us to test our planners also on semi-crowded environments. For simulating a PR2 robot, we have used the generic open source simulation engine MORSE. Figure 4.6 shows a screenshot of the simulation interface.

To test the cooperative planner in a static situation we introduced a mode in the planner where the optimization runs only once for given start and end positions of the robot and the human. This way one can test the effects of the each social constraints separately. This mode can show how the cooperative planner solves the optimization problem and what type of trajectory it generates, without actually moving the robot.

Figure 4.7 shows one of such 'froze' situation. Here we try to show each pose of the calculated robot and human trajectories. We have used the vertical (z) axis to plot the time of the each pose from the static pose. Figure 4.7 depicts two views of the same situation with different angles to have a better understanding of the timed elastic band trajectory. This view helped us to better understand the solution provided by the planner when sometime the 2D-trajectories of the robot and human look overlapping each-other.

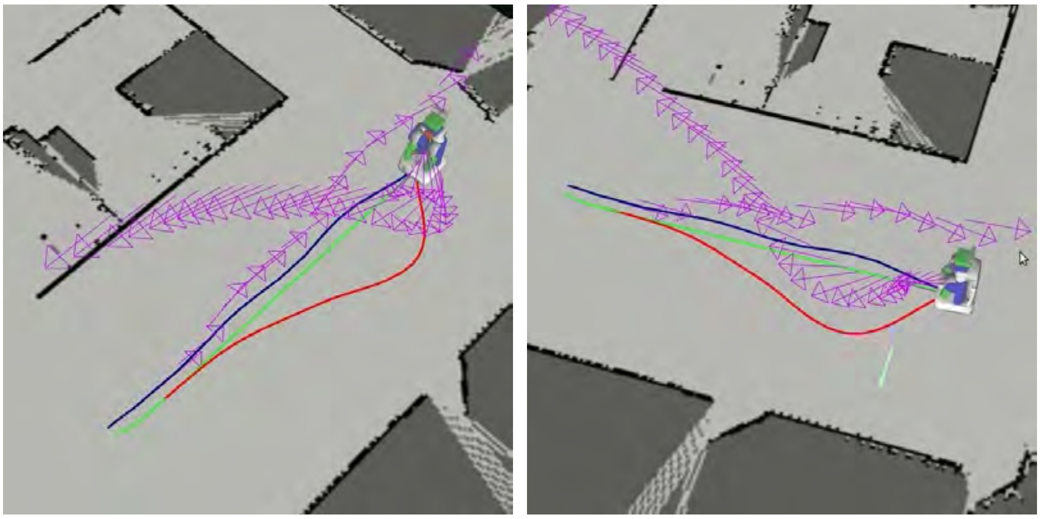


Figure 4.7 – Poses of planned robot trajectory and predicted human trajectories in 3D. The vertical (z) axis plots the time-difference of each pose from the starting (current) pose of each trajectories. The global paths for the human (blue) and robot (red) are also shown in the figure. (In this static simulation, the human start position is at the start of the blue line opposite to the robot.)

We have ported the cooperative planner to two service robotic platforms, the PR2 and the Pepper robot. For real world experiments, our objective is to validate the navigation planning system, not the human detection and tracking algorithms. Therefore we employed off-the-shelf motion capture system from OptiTrack[1]. It publishes positions and velocities of tracked humans at a certain frequency (10 Hz during our experiments[2]).

On the PR2 robot we have also activated a module for coordinating the head motion (see Chapter 5) with the navigation planner to facilitate communication of robot's navigational intentions.

Figure 4.8 demonstrates the capabilities of cooperative planning with a series of experiments in real world interactive situations. Here we demonstrate four of the typical situations that a social robot can encounter when moving in human environments. This figure shows camera view and planned trajectories side-by-side for the same time-point during navigation. It shows situations where robot and the human need to cross a narrow corridor (figures 4.8 a–1 and a–2), or to pass through a door (figures 4.8 d–1 and d–2) while going in opposite directions to each-other. Figures 4.8 b–1 and b–2 show another narrow corridor crossing situation with an additional obstacle previously unknown to the robot. And finally, figures 4.8 c–1 and c–2 shows a situation where robot avoids colliding with two persons.

[1] `http://www.optitrack.com/`

[2] Although the motion capture system delivers data at higher frequency (about 100 Hz), we apply a moving average filter and re-sample the filtered data at 10 Hz to have better estimate of velocities

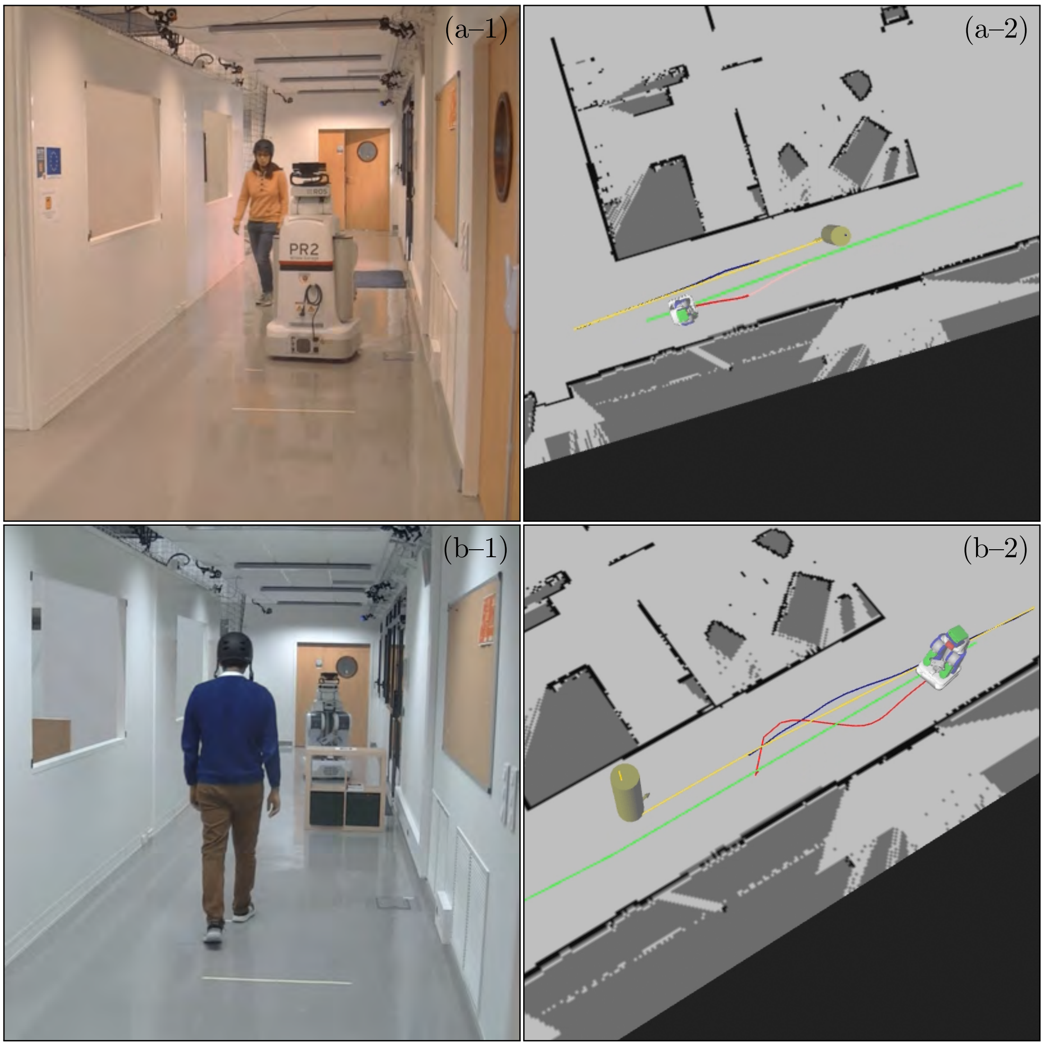


Figure 4.8 – Trajectories generated by cooperative planner for different interactive navigation tasks. (a–1 and a–2) A corridor crossing situation where human and robot share the effort to avoid colliding with each other. The person only slightly alters her path (blue) from what is projected path towards her goal (yellow). The robot's local path (red) is modified from its global path (green) such that the person can safely cross on the side. (b–1 and b–2) A more confined corridor crossing situation because of the drawer. Here the robot facilitates the human to cross the corridor by moving behind the drawer thus giving the person sufficient space to pass. (c–1 and c–2) An open area crossing situation where two human decides to move on either side of the robot. Although in this situation the original plan suggest by the robot was to pass both persons on its right side, when the persons decide otherwise the robot complied and quickly adapted its own path accordingly. (d–1 and d–2) A door crossing situation where the human wants to pass through a door, the robot facilitates the human by stopping near the door. The robot stops not because of a planning failure but because it has planned a cooperative strategy where it waits until the human passes through.

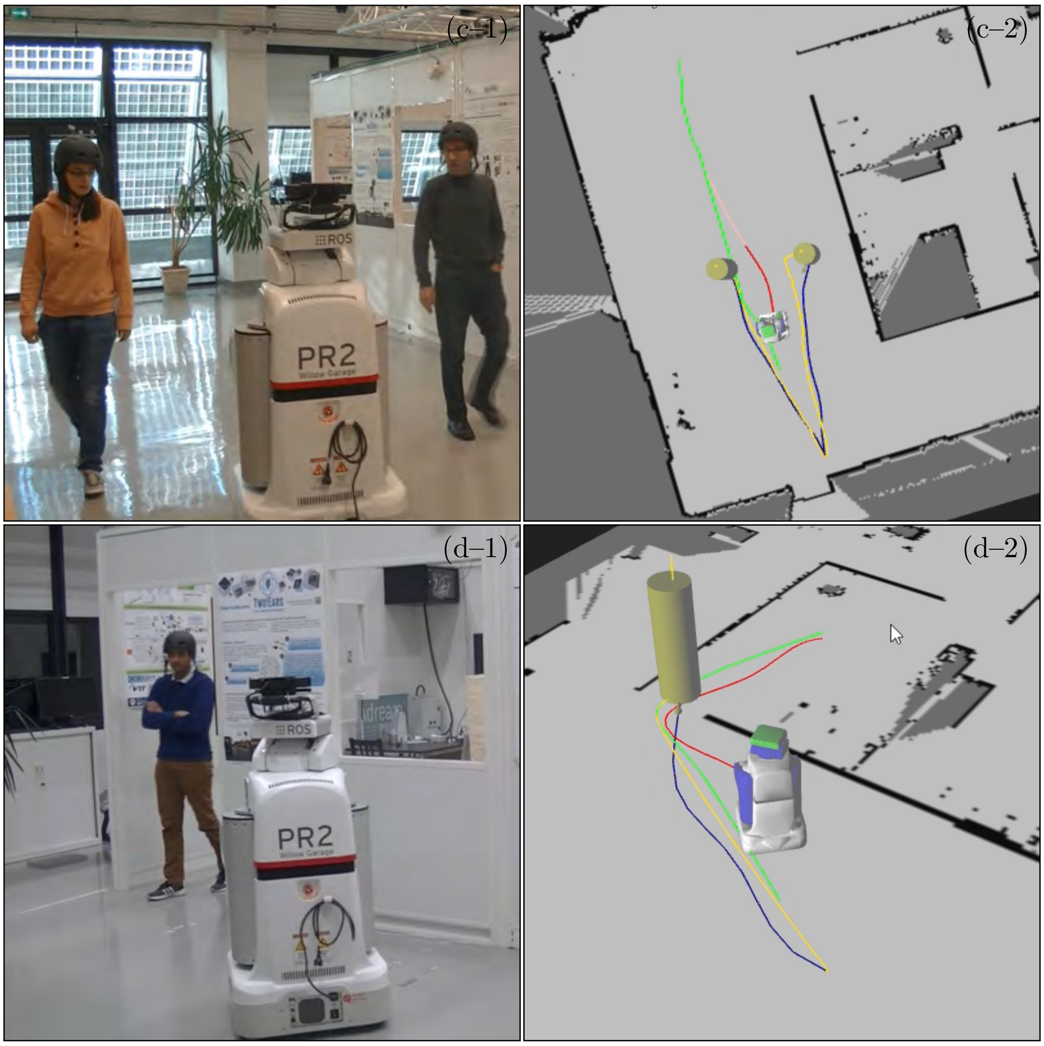


Figure 4.8 – Trajectories generated by cooperative planner for different interactive navigation tasks (continued from above).

We have observed the cooperative planner performing better compared to purely reactive planning schemes, both in terms of solving the co-navigation situation and optimally reaching its navigation goal. We have also learned from these experiments that different co-navigation situation sometimes require slightly different optimization parameters to show best social behavior, in other words, one set of optimization parameters do not fit all situations. During the experiments presented here, we generally converge to a set of optimization parameters that works best for the given situation after a few testing runs.

Please note that, for all interactive situations shown here the planner was not particularly "informed" about the task. The behavior such as stopping near the door and facilitating human in confined corridors emerged because of the integrated social constraints in the optimization framework.

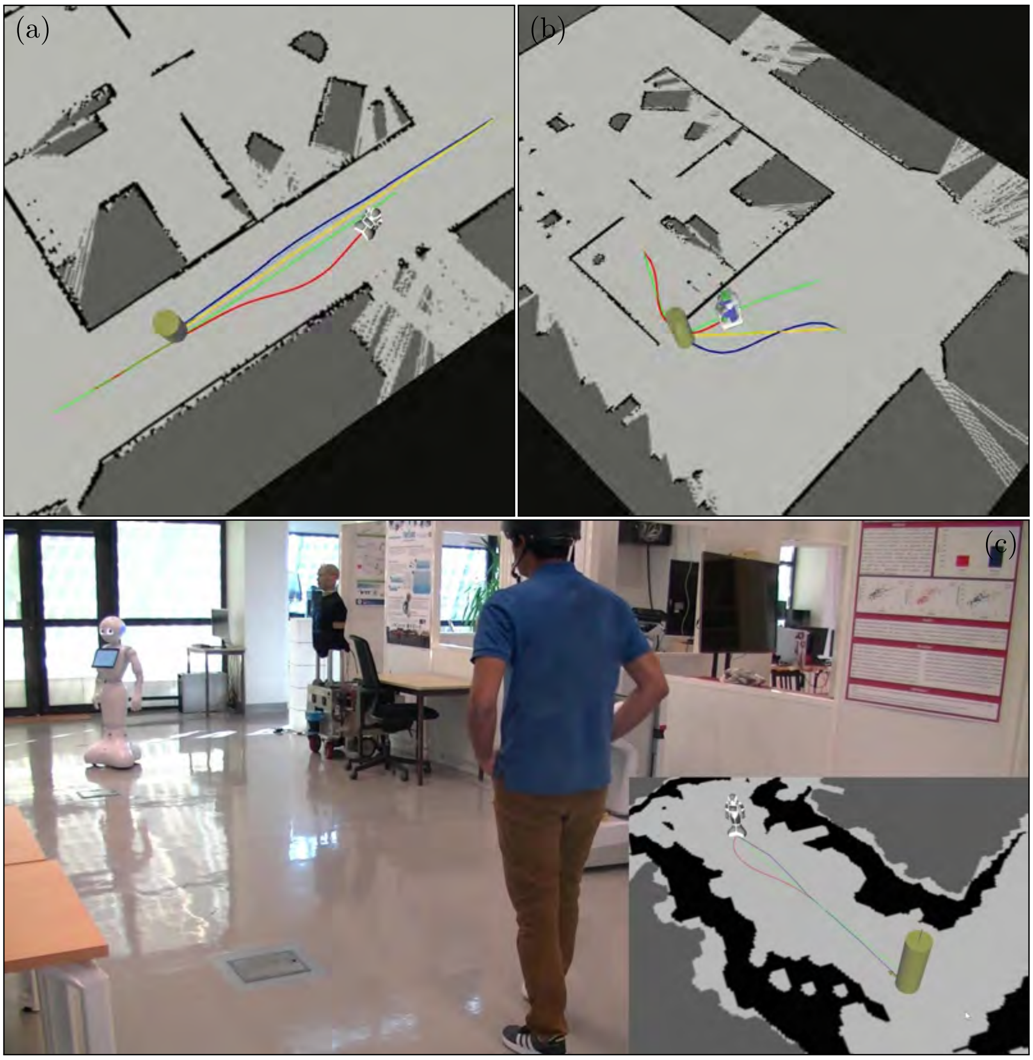


Figure 4.9 – Navigation experiments with the Pepper robot using the cooperative planner. (a) A narrow corridor crossing scenario in simulation, similar to the situation in figure 4.8 a–1 and a–2. Here also the person has to only slightly move to their left and the robot taking most of the effort in moving its left side to utilize the space given to it by the human. (b) A door crossing scenario in simulation, similar to the situation in figure 4.8 d–1 and d–2. The Pepper robot has planned a trajectory that comprise of a waiting behavior outside the door until the human comes out of the door. Once the human passes safely out of the door the robot continues its march towards its goal inside the room. (c) A wide corridor crossing scenario with a real Pepper robot. The Pepper robot moves with much lower speed compared to the human. Therefore, we can see that the robots plan is to immediately move towards a side of the corridor because it predicts that human will eventually cross the robot not too far from the human's goal position (which is also the start position for the robot).

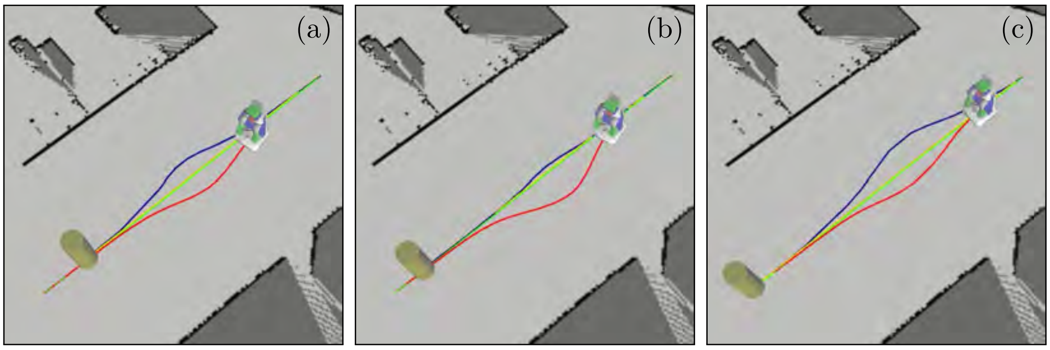


Figure 4.10 – Balancing the shared effort between human and robot. (a) Human and robot share equal effort. (b) The robot takes most of the effort for avoiding a collision and moves far away from predicted human path. (c) Although never used in real world situations, it is possible to design a rude behavior of the robot where it expects human to make more effort for avoiding a collision.

Figure 4.9 shows the cooperative planner working on the Pepper robot. Figures on the top shows a simulated environment with a narrow corridor crossing situation (4.9 a) and a door crossing situation (4.9 b), similar to the situation in figures 4.8 a–2 and d–2 with the PR2 robot.

Although, both of the PR2 and the Pepper robots have omnidirectional drive systems, in these experiments we discouraged the lateral movement of the robots by adding car-like kinodynamic constraints into the cooperative planner. In our view, limiting lateral movement of the robots makes them behave more like how humans move.

Furthermore, the cooperative planner enables balancing and tuning of the efforts between the human and the robot to solve a co-navigation task. The inspiration for designing a tunable navigation planner comes from previously-proposed approaches for geometric [Waldhart 2015], where the robot synthesizes a shared plan for the human and itself. By tuning, we mean to adjust the "elasticity" of underlying *timed elastic band*. Figure 4.10 show the effect of tuning the effort between a human and a robot for a shared navigation task.

During a navigation task, if the human decides to move on other path than one suggested by the robot (for example, choosing to pass by another side of the robot), the robot quickly adapts its trajectory. In situations where robot has enough space to move well advance in time, the robot proactively chooses a path that is both legible and comfortable for the human counterpart.

These results are certainly encouraging for us. We wanted to further test and compare this cooperative planning scheme against other state-of-the-art human-aware planning schemes. In the following section, we compare the cooperative planner against a dynamic window based planner which only used the directional cost model introduced in the previous section and one more human-aware planner that also integrates human motion prediction within its planning framework.

4.2 Assessing the Social Criteria for Collaborative Navigation

This section focuses on requirements for effective human robot collaboration in interactive navigation scenarios. We designed several use-cases where humans and robot had to move in the same environment that resemble canonical path-crossing situations. These use-cases include open as well as confined spaces. Three different state-of-the-art human-aware navigation planners were used for planning the robot paths during in all selected use-cases. We compare results of simulation experiments with these human-aware planners in terms of quality of generated trajectories together with discussion on capabilities and limitations of the planners. The results show that the *human-robot collaborative planner* introduced in the previous section performs better in everyday path-crossing configurations. This suggests that the criteria used by the *human-robot collaborative planner*, namely *safety*, *time-to-collision*, *directional-costs* are possible *good* measures for designing acceptable human-aware navigation planners. Consequently, we analyze effect of these social criteria in detail and draw perspectives on future evolution of human-aware navigation planning methods.

The work presented in this section has been partly published in [Khambhaita 2017b].

4.2.1 Comparison Methodology

We are witnessing a surge in social robots that are present in our everyday lives. Robots are offering guidance to passengers at airports [Triebel 2015], providing assistance to elderly persons at care centers [Hebesberger 2016], or even engaging in entertaining experiences in public spaces [Foster 2016]. Use of social robots in these domains show that situations where human and robot share and navigate in common space are becoming more and more important. An even higher degree of collaboration is necessary for fluent co-navigation, especially in confined spaces such as narrow corridors: here human and robot have to act cooperatively and help each-other for finding their way. Although there are several human-aware navigation planners proposed in the literature, as far as we know no substantive approach yet proposed to compare and evaluate the planning schemes.

In this work we consider several normative co-navigation situations as basis for comparing how different human-aware navigation planners perform in the same situation. Therefore, we will refrain from going into the implementation details (planning algorithms, robot control) and rather focus on assessment of paths generated by the planners and behavior of the robot during a full interactive navigation episode. Figure 4.11 shows the situations we have designed in a simulated environment for evaluating the planners.

For comparison we have selected three different planners based on how they model social constraints and what methods they use for finding a trajectory that satisfy these constraints. We are only comparing three different local planners here. All of

Figure 4.11 – Canonical human-robot path crossing situations. (a) Human and robot crossing each other's path on a corridor. (b) 90° path crossing situation. (c) Human coming out of a room where robot needs to go, they cross each-other at the door. (d) Similar to situation 4.11a, however the drawer is blocking the corridor in such a way that only one person (or robot) can pass next to the drawer at a time.

the planners utilize the ROS navigation architecture `move_base`. Thus, for global planning we used the standard ROS planner that uses Dijkstra's method [Dijkstra 1959] for calculating a path from start to goal position. The planners we compare are available as a local planner plug-in to the `move_base` and their source code is freely available online.

We have constructed a simulated environment with MORSE that resembles the real robotics lab at LAAS-CNRS. Figure 4.11 shows screen shots of the simulator and situations we have designed for testing the planners. We have used simulated version of the PR2 robot for these experiments. One of the aims of simulated experiments present in this paper is to get insights on workings of each planner in different co-navigation situations and prepare for a later user-study with real PR2 platform. An advantage of simulation is that it provides consistent and reproducible environment for testing different navigation algorithms. For simulating human motions, we used the simulator described in Section 3.1.

(a) Directional Cost Model

In our comparison of human-aware planners, the first planner we have compared uses the directional cost described in Section 3.2.1. Specifically, with this strategy the robot attempts to solve a spatial conflict by adjusting velocity instead of path when possible. This planner requires predicted human paths to calculate how compatible it is with the planned robot path. We are using simple velocity obstacle method [Fiorini 1998] to predict future human paths based on their current velocity. With this method the uncertainty of the predicted path increases with time. The local planner takes this uncertainty into account while calculating the compatibility, meaning it checks the compatibility against the full cone-shaped projection of human path.

(b) Social Force Model

Another acclaimed approach used by the human-aware robotics researchers is *social force model* [Helbing 1995]: a method to describe crowd dynamics. The robot navigation planners explained in [Ferrer 2013] and [Kuderer 2012] uses the *social force model* to cope with uncertain human motions. A further improvement that uses an *extended social force model* with a sampling based planner to make the robot motions proactive while respecting kinodynamic constraints of the robot [Ferrer 2014b] [Ferrer 2015]. In this approach, every iteration of planning step uses the human prediction information which is dependent on the path calculated during the previous iteration. This is the second planner we are using for our comparison[3]

An important concept brought in by these methods is to interlace prediction of human motions within the robot navigation planning framework. Because of superior results, we believe that any further human-aware navigation planner should also involve prediction of human motions in their planning architecture.

(c) Cooperative Trajectory Optimization

The cooperative planner introduced in previous section also interweaves prediction and optimization of human trajectories in the same framework. Besides, we have brought in carefully selected social constraints within the optimization framework. The resulting planning framework is cooperative in nature, that means the robot not only can react to the unfolding situation but it can also proactively suggest a solution to the co-navigation situation. Furthermore, the robot uses the same environmental

[3] We used the source code available at `https://devel.iri.upc.edu/pub/labr`.

map to coherently plan its own trajectory and predict human trajectories, therefore it always provides a solution where both human and robot can move optimally.

In this approach, our focus is on advanced interactive motions with single person to a small group of people. For open spaces, this cooperative planner gives comparable results to the *social force model* based approach. However, in confined spaces it give arguably better results by unnecessary detours and remaining adaptive to the human motions. This is the third planner we are using for comparison of human-aware navigation planners.

4.2.2 Experiments with Canonical Co-Navigation Situations

We have compared these three human-aware navigation planner in five customary human-robot path crossing situations. Thus, for each situation we give exact same start and goal positions of the robot and human for all three planners. As a shorthand we will use the names *directional-cost planner*, *AKP planner* (for the social force based planner) and *cooperative planner* throughout the following discussion.

All the figures in the following discussion shows both global (depicted in green) and local (depicted in red) paths of the robot. Whenever possible the predicted human paths are also shown in the figures (in blue)[4].

(a) Indoor Hall Navigation

In the first situation we have considered a hall size area where robot and human are crossing each other's path in a face-to-face configuration. Figure 4.12 shows a particular scene during the navigation task for each planner. Here trajectories generated by both AKP and cooperative planners make robot move away from the predicted human path thus keeping a comfortable distance from the human and require minimal or no effort on from human regarding changing their path. Since, the directional-cost planner only plans in velocity, it slows down as the human approaches near eventually stops completely. Once the human moves behind the robot, the robot continues on its path, thus it requires human to change its original path. The directional-cost based planner works in two steps, it uses the path generated by the *global planner* while slowing-down or speeding-up near moving humans. Thus, if we use the *global planner* with continuous re-planning scheme eventually the robot will change its direction as well, as soon as a new *global plan* is available. However, here we are purely comparing the *local planner*, so to remain fair to other two planner we are not using the re-planning mechanism for the *global planner*.

Although both AKP and cooperative planner successfully avoid the possible collision, we can see that the cooperative planner is already moving on away from the human path ahead in time compared to the AKP planner. This behavior shows that the cooperative planner proactively taking the decision on how it will avoid

[4]The purple cylinder shown in the AKP planner examples are the goal positions used for the human goal prediction.

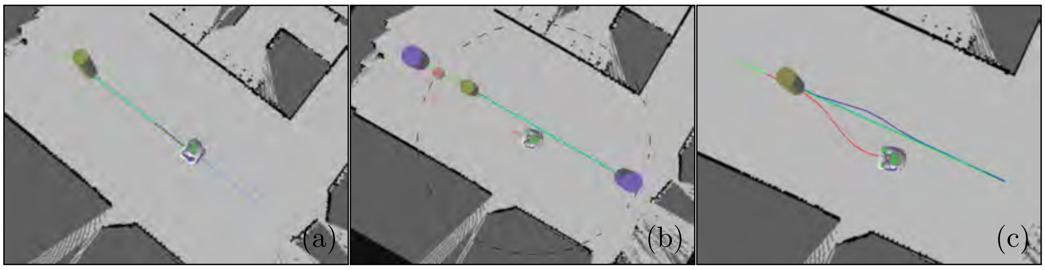


Figure 4.12 – Trajectories generated by human-aware planners for indoor hall navigation task. (a): Directional-cost planner. (b): AKP planner. (c): Cooperative planner. Here the robot and human are crossing each other's path in a face-to-face configuration. Since the area is relatively wide, the robot has enough latitude to move farther from the predicted human path in this situation.

a possible collision with the human. Furthermore, when coming near the human and passing on the side, the robot also slows down to make sure the human do to feel any danger, this is generally the effect of the *directional* constraint used in the cooperative planner.

(b) Corridor Crossing

Second situation is quite common in office, airport, or shopping mall like environments. Here the human and the robot are crossing their paths similar to the previous situation, however, since the passage is narrow it required some effort from both of human and robot to avoid a collision (4.13). In case of the directional-cost planner the robot again simply slows down as the human approaches, the *local plan* of the robot getting smaller. Thus, even this behavior is understandable for the human, it requires some effort from human to cross the robot. Since the direction-cost planner works in two levels, if the human stops in front of the robot the global planner will eventually find a path going around the human which the local-planner follows.

The AKP planner starts properly on its path, however, as the human comes near, it plans a trajectory backwards. Such back-tracking is common in *social force model* based planners. A possible explanation for this behavior would be that, while the repulsive force of the walls remaining same on both sides of the robot as human approaches the repulsive force from human gets larger than the attractive force towards the robot's goal, which eventually results into the robot moving backwards. Because of backtracking, and they was AKP planner designed and used for particular robot, only for this planner we had to add an imaginary laser scanner on the back of the PR2 robot in simulated environment.

As we can see in the figure 4.13c–1, the cooperative planner moves on the one side of the corridor well ahead in time. It does so because it has planned a human path in the same map that causes least effort on for human to change its path. It should be noted that, in this case it is necessary for both human and robot to change their paths to find a solution to the co-navigation problem. Here robot selects a

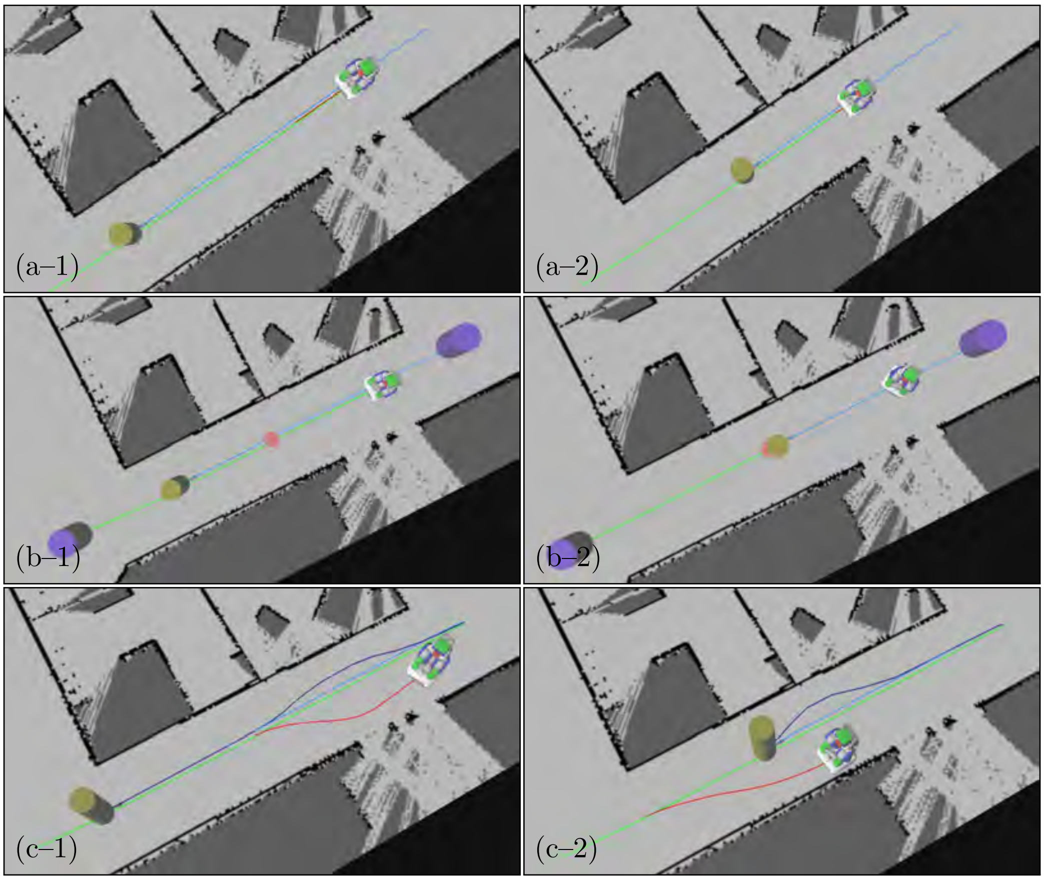


Figure 4.13 – Trajectories generated by human-aware planners for corridor crossing task. Here also the robot and human are crossing each other's path in a face-to-face configuration, however, in a confined space. (a–1) and (a–2): Directional-cost planner. (b–1) and (b–2): AKP planner. (c–1) and (c–2): Cooperative planner. Figures (a–1), (b–1) and (c–1) shows snapshots of situations that happen a few seconds before the situations (a–2), (b–2) and (c–2) respectively.

solution that causes minimal change to the human trajectory.

(c) 90° Path-Crossing

Figure 4.14 show a situation where robot and human are crossing their paths at a 90° angle. Here the directional-cost planner performs quite good as the robot slows down when human comes near, and then the robot again accelerates towards its goal, no additional effort required by the human here. It should be noted that the directional-cost planner was extensively tested on this situation, thus it generates paths that are legible and acceptable by the humans.

AKP planner first plans a path to cross the human from their front, but when the human is too near the planner eventually decided to pass behind the human. Here, we suspect this behavior is due to the particular workings of random exploration based

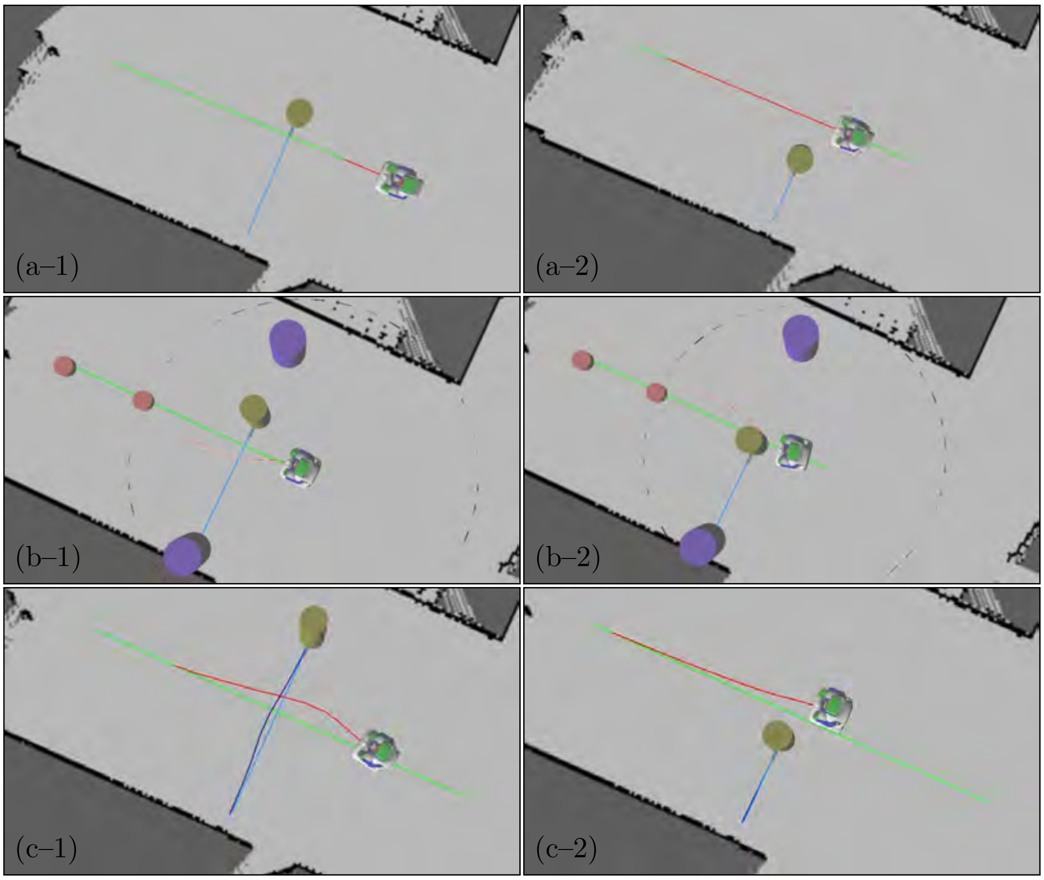


Figure 4.14 – Trajectories generated by human-aware planners for 90° path-crossing task. (a–1) and (a–2): Directional-cost planner. (b–1) and (b–2): AKP planner. (c–1) and (c–2): Cooperative planner. Figures (a–1), (b–1) and (c–1) shows snapshots of situations that happen a few seconds before the situations (a–2), (b–2) and (c–2) respectively.

algorithm used by the AKP planner. Nevertheless, A^*-like path search algorithm with continuous re-planning scheme also yields similar behavior.

The cooperative planner simultaneously changes its path while slowing down, thus suggesting human to pass before itself. Since, we are using both *directional* and *time-to-collision* constraints for this planner, this can be explained as combined effect of those social constraints. Therefore, the resulting behavior remains as legible as, or arguably even better than the directional-cost planner.

(d) Passing through a Door

As shown in figure 4.11c, sometimes in a door-crossing situation human is visible to the robot due to the window between the hall area and the room. Figure 4.15 shows trajectories generated by the planners when task of the robot is to move inside the room while the human is moving out. The directional-cost planner could induce

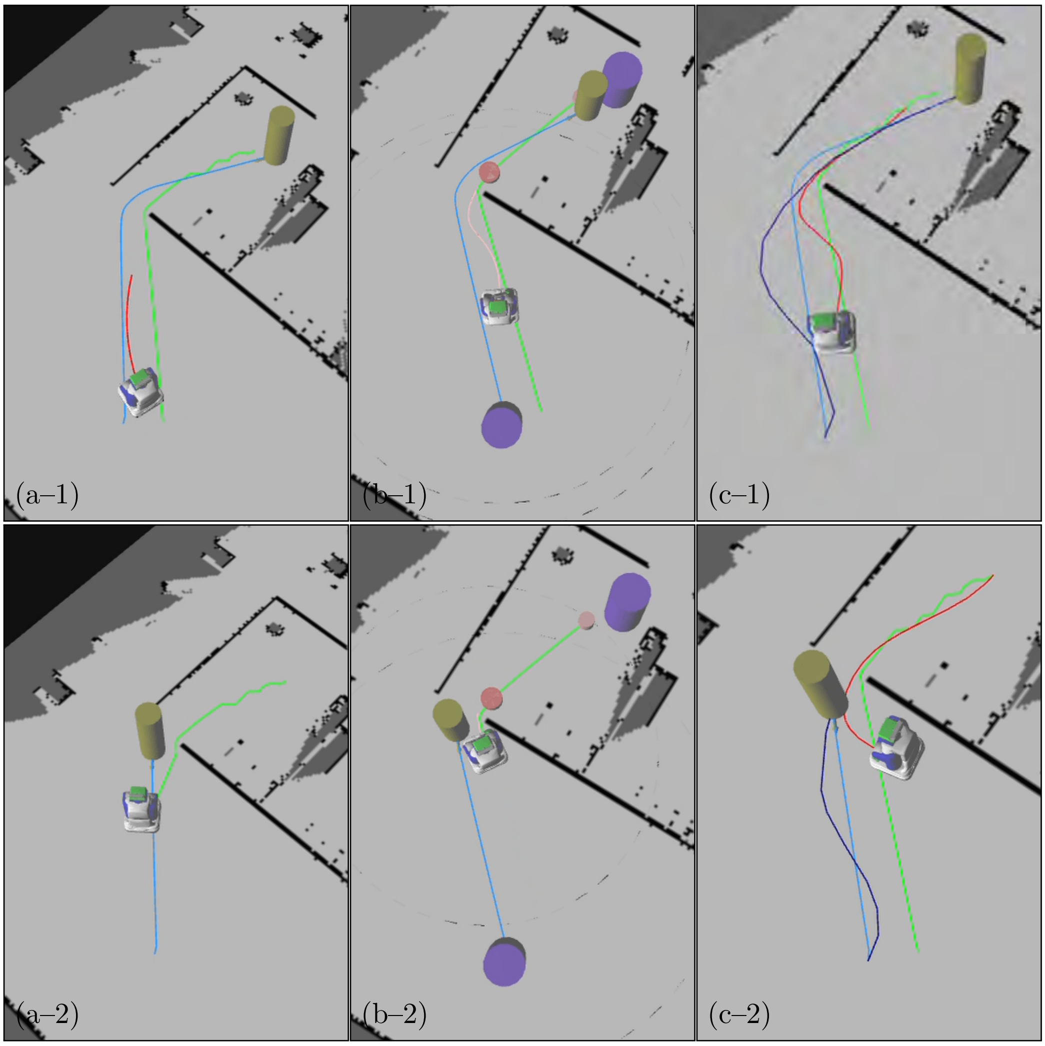


Figure 4.15 – Trajectories generated by human-aware planners for the robot passing through a door. (a–1) and (a–2): Directional-cost planner. (b–1) and (b–2): AKP planner. (c–1) and (c–2): Cooperative planner. Figures (a–1), (b–1) and (c–1) shows snapshots of situations that happen a few seconds before the situations (a–2), (b–2) and (c–2) respectively.

"bad" configurations depending on how the situation plays out, where it reaches near the door and slows down but does not give enough space to the human to move out of the room. This could lead to considerable effort by the human, making them move back to create space for the robot. In other cases, it still goes straight towards the door and slows down in front of it, as shown in the figure 4.15a–2, which may shock the human and creates discomfort for the human.

The AKP planner, because of its backtracking behavior could eventually make space for the human. However, as we can see in fig 4.15b–2 the robot goes very near to the door which could threaten the human.

With the cooperative planner the robot prefers waiting in a place where it limits, as much as possible, obstruction to the human motion. Instead of moving backwards, here the robot proactively plans a path that is not only far from the door when human is coming out but the robot trajectory inherently contains a “waiting” behavior.

(e) Constrained Corridor Crossing

The last situation we have considered for comparison is similar to the corridor crossing situation described in Section (b), however with an additional obstacle in the corridor. Figure 4.11d depicts this situation in the simulation environment. The corridor becomes a highly constrained space, where robot and human cannot even pass side-by-side. Only the human or the robot can cross the additional obstacle at a time, requiring other to either wait or backtrack. The additional obstacle (a drawer) is not known to the robot, that is, it is not in its pre-built map. Therefore, we are also showing laser scanner points (in yellow) to better understand the robot behavior.

Here both directional-cost and AKP planner come very near to the obstacle at the same time when human also reaching near it. Directional-cost planner hardly given enough space for the human to pass, while AKP planner backtracks a bit when human comes very near to the robot. Since with both of these planners the robot slows down to almost standstill, they are not violating the safety requirement, however, creates discomfort for the human. However, with both planners the human need to share most effort for avoiding a collision with the robot.

The cooperative planner performs particularly well in this situation. As shown in the figure 4.16c–1, the robot first moves near the drawer, and waits for the human to pass, and then continues its motion towards its goal.

4.2.3 Effect of Different Social Criteria on Robot Behavior

With simulated experiments in Section 4.2.2, we have shown capabilities and drawbacks of three different human-aware navigation planners. The cooperative planner performs generally good in both confined and non-confined cases. The behavior such as stopping near the door and facilitating human in confined corridors emerges due to the social constraints that are integrated in the optimization framework and because the cooperative planner is planning a path for both human and robot which allow them to avoid other static or dynamic obstacles. The cooperative planner is also highly reactive at the same time, so if during a navigation task the human decides to move on other path than one suggested by the robot (e.g., choosing to pass by another side of the robot), the robot quickly adapts its trajectory.

In the following, we briefly discuss the effect of the social constraints *safety*, *time-to-collision* and *directional-costs* separately on the quality of the trajectories generated by a collaborative planner.

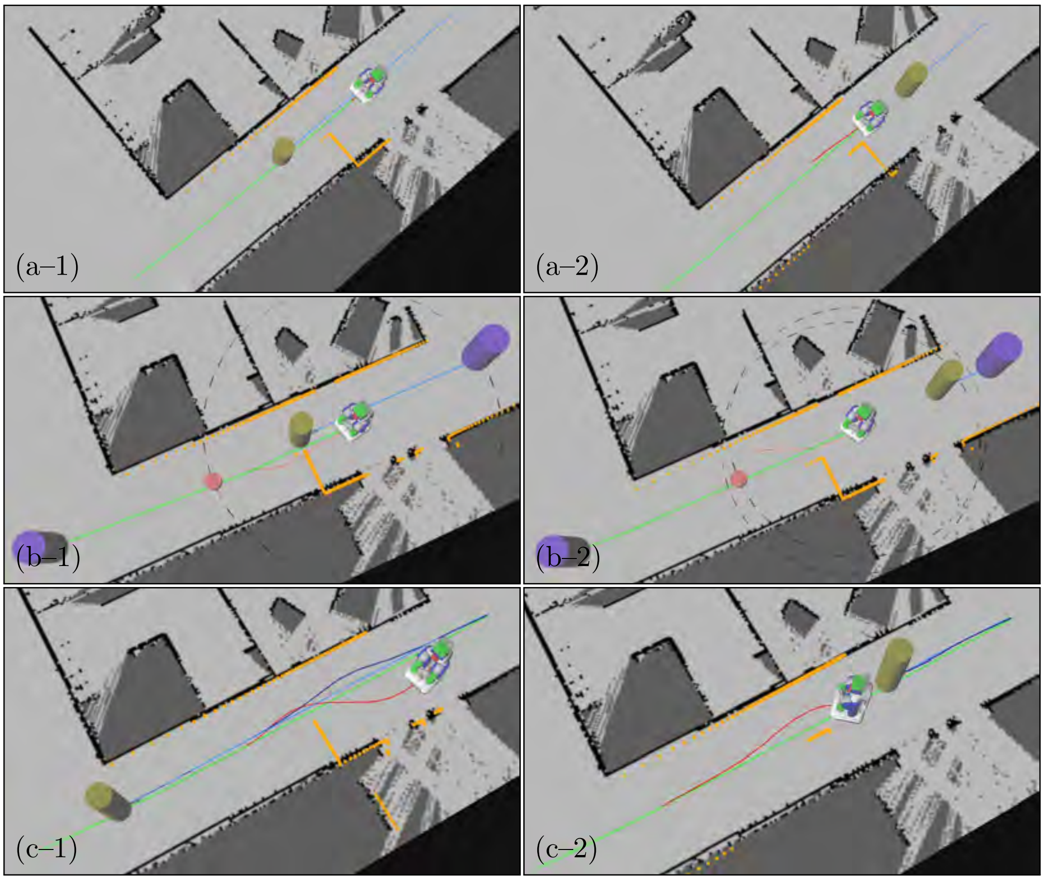


Figure 4.16 – Trajectories generated by human-aware planners for corridor crossing task with an obstacle in the middle of the hall. (a–1) and (a–2): Directional-cost planner. (b–1) and (b–2): AKP planner. (c–1) and (c–2): Cooperative planner. Figures (a–1), (b–1) and (c–1) shows snapshots of situations that happen a few seconds before the situations (a–2), (b–2) and (c–2) respectively. Since the obstacle was not known to the global planner, it must be avoided by the local planner. For better visualization we also show the laser points (yellow) as recorded by the simulated laser scanner.

(a) Effect of *safety* Constraint

The cooperative planner only applies the *safety* constraint between the respective point in time along the human and robot trajectory, so that slow and restrained robot behavior is avoided. With all other parameters being equal, figure 4.17 shows the effect of *safety* constraint on the robot path. With a single parameter we can tune how far the robot moves form the human. Since this constraint is only applied between configuration of the human and robot for the given time horizon, the robot is able to ensure the safety along the whole trajectory.

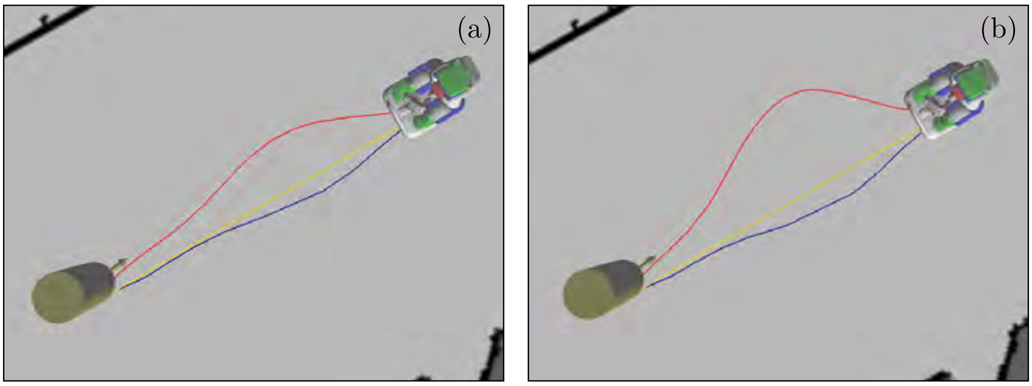


Figure 4.17 – Effect of *safety* constraint on robot paths. (a) Minimum safety distance between human and robot is set to 0.3 m meters. (b) Minimum safety distance is set to 0.7 m.

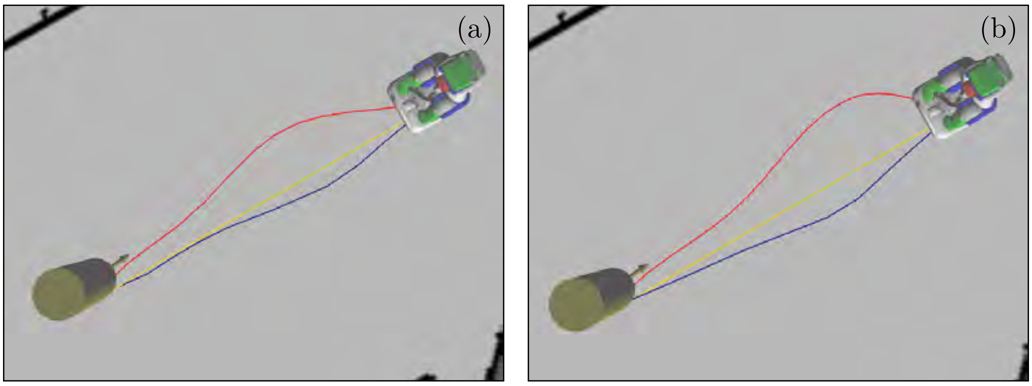


Figure 4.18 – Effect of *time-to-collision* constraint on robot paths. (a) Without *time-to-collision* constraint. (b) *time-to-collision* constraint is switched-on.

(b) Effect of *time-to-collision* Constraint

A novel constraint used by the cooperative planner is *time-to-collision*, defined as the projected time to a possible future collision with a human. Recent empirical studies of pedestrian interactions have shown that *time-to-collision* can uniquely describe how humans move in public places [Karamouzas 2014]. It is also interesting to note that, the same study also shows that the *social force model* rather poorly fits to the pedestrian interaction data, and thus being a weak model to design human-robot interactive navigation planners. Figure 4.18 shows an example of how this *time-to-collision* constraint affects the generated path. When *time-to-collision* constraint is switched-on the robot moves little early, thus making the robot behavior proactive.

(c) Effect of *directional* Costs

The cooperative planner exploits results of user study in [Kruse 2014] and introduces *directional costs* in the optimization framework. The *directional costs* discourages face-to-face motion towards a person. It is rather difficult to show the effect of *directional cost* constraint in an image, as the main effect of this constraint is to make the slow down when human and robot are in move opposite to each other. It also makes the robot slow down while moving very near to the human because of the modeled inverse proportionality of this constraint to the distance between the human and the robot. The full optimization scheme of cooperative planner, however, makes the robot balance between slowing down and changing the path.

4.2.4 Insights

In this section we have compared three human-aware planners against five different canonical co-navigation situation humans encounter in their everyday lives. Following are the lessons learned from this comparison:

- Directional-cost based planner performs well in 90° path crossing situation, however performs sub-optimally in constrained situations.
- Social force model based planners can perform well in open areas or large indoor environments, however they suffer from unnecessary detours due to the way repulsive and attractive forces are calculated in this model.
- The cooperative planner performs well both in non-constrained and constrained situations, while legible and cordial behavior emerging with use of this planner in confined areas.
- For a decent human-aware navigation planner it is imperative to tightly couple with human motion prediction algorithms.

Although simulation being a limited tool, repeatability of the same experiments makes it a suitable tool for evaluating different navigation algorithms on the same situations. We believe that our investigation provides preliminary basis for comparing human-aware navigation planners. Drawing fruitful insights from these experiment, we think the cooperative planner is ready for extended evaluation on the real robot. Therefore, we are planning a comprehensive user-study to further evaluate the cooperative planner and compare it against other human-aware navigation planners.

4.3 Summary

This chapter introduces an adaptive and cooperative human-aware navigation planning method. This cooperative planning scheme has several advantages over other state-of-the-art human-aware planning schemes. First, our robot does not stay purely reactive but now it can also propose a path for the human assuming she will consider the proposed solution that benefits both agents. This is crucial especially in confined spaces, such as corridors where two agents can navigate only in side-by-side configuration. Second, the cooperative planner is able to tune and balance the effort between human and robot in co-navigation situations by devising a shared plan that includes trajectory of robot and nearby humans. The optimization process works on the whole trajectory until specified time horizon. Moreover, instead of returning a planning failure when a human is blocking the robot's path, the cooperative planner deliberately plans a waiting behavior until human gives a way and the continues moving towards its goal. The cooperative planner uses the novel time-to-collision criteria for computing socially acceptable navigation trajectories in addition to safety and directional costs.

The computational cost of the cooperative planner increases with the number of humans surrounding the robot as well as number and type of social constraints used for optimization. This is a limiting factor for using the cooperative planner in crowded situation. We plan to improve over this limitation by parallelly evaluating the error functions and utilizing GPU capabilities provided by state-of-the-art sparse matrix solvers.

Finally, to assess the improvements in robot behavior using the cooperative planner, we have compared it against a directional cost based and a social force model based planner. We have provided analysis of behavior of the robot in each situation with each planners and discussed their capabilities and drawbacks.

Chapter 5

Head-Body Coordination during Navigation

Contents

While cooperative planning helps robots to effectively navigate in open as well as confined space, passive gaze can still leave people bewilder about its intents. **Legibility** of robot motion is crucial for higher acceptance and suitability of social robots in public places. Thus, in this chapter our focus is on planning and control of head-movements. We study the effect of synchronized head-body motion on legibility in the context of robot navigation. Here, we had adopted the working definition of legibility given by [Lichtenthaler 2012]: “Legibility is an attribute of robot behavior that is used by a human to infer the next actions, goals and intentions of the robot with high accuracy and confidence.”

A large number of mobile robots today have a pan-tilt head mounted at top of their structure. Spencer, Pepper and the PR2 robots are example of such robot having their appearance resembling a human being. It seem natural and pertinent to use their gaze direction to manifest its navigational intents for agreeable human-robot interaction, even if the head motion is not necessary for the robot in terms of perception. We frame control of robot gaze behavior as multi-criteria decision-making problem, and cater a solution to synchronize gaze control with robot's navigation planner. This approach is useful in the context of robot navigation where it may sometimes be awkward to display only a predefined gaze pattern, due to the dynamic nature of the scene.

The literature review on human gaze behavior lead us to propose this framework for the robot gaze behavior following a multi-criteria approach. By enabling two behaviors, *look-at-path* and *glance-at-human*, we show the effectiveness of our approach on a real robotic platform in a path crossing scenario. Furthermore, later in this chapter we also discuss the results of a video based user study conducted with 126 participants showing improved communication of robot's navigational intentions with the proposed approach.

The work presented in this chapter has been partly published in [Khambhaita 2016].

5.1 The Modality of Gaze

As we have seen in Chapters 2 and 4, human-aware navigation planners already produce safe and socially acceptable motion of a robot. Construction of directional cost functions [Kruse 2012] specifically increases legibility of robot motions, making a robot to solve spatial conflicts by adjusting its speed instead of its path when possible. The human counterparts prefer this approach [Kruse 2014], however, it is not enough to lower hesitation in humans and to assure legibility and acceptability of the robot behavior. We need to model the robot gaze to give explicit information about robot's plans and goals.

5.1.1 Where Do We Look When We Steer

It is well-known that humans predominantly use gaze as a cue for understanding intentions and mental states of others. In one of the most influential studies of human intention recognition, [Castiello 2003] concluded:

> Even in the absence of any overtly executed action, observers can still read other people's motor intentions, provided they can see a model's face, in particular his or her gaze direction.

Besides, research in human behavioral psychology reveals that the constant alignment of the head with walking direction, as well as alignment of gaze with other humans and with static objects (for obstacle avoidance) [Kitazawa 2010] are the prominent gaze behaviors associated with locomotion. How a robot with a pan-tilt head unit could simultaneously manufacture such motions is still an open issue.

Based on the behavioral psychology findings, on the onset of our investigation, we imagined a typical sequence of events involved in a human-robot path crossing

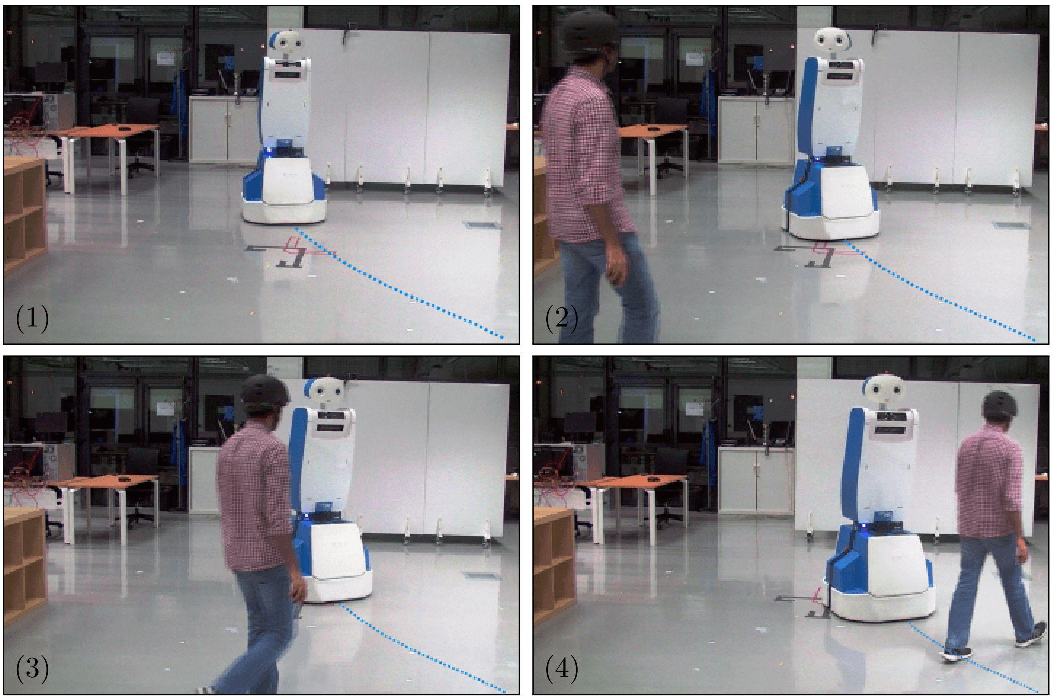


Figure 5.1 – A typical sequence of events during human-robot path crossing scenario. (1) The robot starts looking at the path. (2) A human enters in the area, and the robot detects a possible interference; the robot turns its head towards him. (3) After acknowledging the human, the robot turns its head back towards the path and slows down at the same time. (4) The human passes in front of the robot, the interaction episode ends.

situation as shown in figure 5.1. To bring about such human-robot interaction, we will now look into relevant parameters and social context, explored so far, associated with effective gazing behavior.

5.1.2 Related Work in Robot Gaze Behavior Planning

In fact, the larger part of nonverbal human-robot interaction literature belongs to the studies of robot gaze and its utility for social signaling. Researchers have successfully used robot gaze to increase attention and thus engagement of users [Mutlu 2006], [Zaraki 2014]. Moreover, sometimes deliberate delays [Admoni 2014] or premeditated motion patterns [Gharbi 2015] are required to capture partner's attention. Gaze direction can also communicate attentiveness and visual awareness of the robot [Breazeal 2005]. Human-like head movements, adapted to the context of an interaction task, can help improve the fluency of the interaction. Furthermore, coordinated head and body movements as well as facial expression can also display different emotional state of the robot [Amaya 1996; Inderbitzin 2011; Barliya 2013].

In a squared-table interaction scenario, [Al Moubayed 2012] have compared how humans perceive robot gaze direction from eyes and head orientation. Conditions for their experiment were, only orienting eyes towards a target, or only orienting head (while keeping eyes look straight ahead). It turns out that head orientation is an accurate signal of where the robot is looking on the lateral axis (that is, sideways), as perceived by the participants. Although these experiments feature a non-mobile robot, the results are helpful for us to deduce that head orientation is more useful compared to combined orientation of head and eyes.

A big part of robot gaze research relates to situated interaction settings. It is only recently the HRI community have begun to explore the effects of robot gaze in navigation context. As robot moves, its motion leaves an innate effect on humans about the robot's "mental state" and intentions. It is, therefore, indispensable to look at the robot head motion jointly with the body. [Fiore 2013] have found that gaze affects the perceived emotional states in a human-robot path crossing scenario. Nonetheless, their findings imply that the gaze of a robot is not as important a social signal for that particular scenario, meaning the gaze have no interactive effects on perceived social presence. The timing of particular gaze behavior execution could be the reason behind these findings, where the robot head was orienting towards human only in the beginning of an interaction episode. Our results, presented in Section 5.3.3, beg to differ in this regard.

[May 2015] have compared head orientation and visual light indicators for communicating turning signals, with a pre-scripted behavior for controlling the head orientation. Their results hint at a positive impact on comfort felt by humans using both communication modalities, with participants favoring the indicators over head orientation. It is, however, difficult to show whether the robot is making a slight or sharp turn with such indicators. [Lu 2014a] have also tested a scripted *glance* behavior during the task of jointly navigating in a hallway. In their implementation the robot looks at a passing humans for a certain amount of time and then looks back in front of the robot. However, they remain inconclusive whether this behavior was effectively giving the person acknowledgment that the robot saw them.

5.1.3 Why Robots Need Better Head/Gaze Behavior?

A simple answer is, for robot to look like a social agent. Effective use of different modalities for communicating navigational actions and goals to co-existing humans is the broader context of our investigation. At the same time, we would like interaction be *natural*, make it easy for an uninstructed person to estimate internal state of the robot from observations. [Kruse 2013] defines *naturalness* as "the similarity between robots and humans in low-level behavior patterns". Aspects of the lower-level robot behavior patterns that convey actual robot intentions are also helpful in increasing the legibility. It is important for an interaction that the robot shows synchrony between the different expressive modalities. As shown by [Dragan 2015], isolated motions threaten legibility of the robot's behavior and may lead to wrong beliefs and expectations about it.

Task-dependent robot head expression can enhance the robot behavior. However, while navigating among humans our robot is executing multiple parallel tasks. Our robot continuously looks ahead at its path to help humans forecast its immediate navigation plans. It also performs a *saccade* like behavior towards other pedestrians to convey that the robot has seen them and it is going to avoid possible path conflicts. Only by taking navigation and gaze planner as a whole, our system can produce more acceptable and legible overall robot behavior.

Timing of integrated head movements plays a crucial role, for instance, in communicating intended handover region in an object handover task [Hart 2014; Gharbi 2015]. When well synchronized with rest of the body, gaze can even manifest forethought to improve robot's readability [Takayama 2011]. That being discussed, implementations of particular gaze behaviors lack generality and remain largely ad-hoc in nature. We want to avoid another ad-hoc implementation.

5.1.4 Parameters for Gaze Control

Determining to what extent head, eyes, and body orientation contribute to a gaze behavior is crucial. One of the first user studies in this regard [Imai 2002] suggests that knowing the head orientation (pan and tilt axes) is enough for a person to notice the robot gaze. In a round-table situation with a robot in the center, reported required precision of the robot gaze (pan axis) is within 22.5° to either side of the relative human standpoint.

Search for the essential factors for regulating gaze control leads us to investigate further the research on the gaze behavior associated with human locomotion. Humans start turning their head in the travelling direction before their body during locomotion. This anticipatory nature of the head orientation is known for some time [Patla 1999]. It is also clear that humans begin head orientation control several meters before the turn [Hicheur 2005]. With a series of experiments [Bernardin 2012] have quantified the anticipation They have observed that at normal walking speed the gaze shifts by 500 to 700 milliseconds in advance compared to the body. Besides that, the angle for gaze anticipation into the heading direction increases with the increasing curvature of the path. A recent user study [Unhelkar 2015] supports these findings and points out that in addition to *head orientation*, *body velocity* is also statistically important for anticipating a turning motion in humans. However, in their following work these anticipatory signals are only used for predicting future human positions and they show how it helps collision avoidance with a series of simulated experiments.

[Kitazawa 2010] have also investigated gaze patterns in a collision avoidance scenario with multiple pedestrians moving in a wide hallway shape area. Observed participants were fixating their gaze on other pedestrian and static obstacles for avoiding a collision. Reported average distance of fixation was 3.97 meters (SD = 0.54) for approaching participants and 1.9 meters (SD = 0.71) for leading participants (average time for the fixation is not reported). Furthermore, during the experiment the participants fixated their gaze on other pedestrians when it was really necessary for collision avoidance. Put succinctly, the two most significant factors for the robot gaze during navigation are robot's own path-direction and presence of humans in the proximity. Nonetheless, their significance also depends on the task at hand, for example approaching or accompanying a person.

5.1.5 Deciding Where to Look

Because there are multiple factors to take into account for controlling the robot's head direction during navigation, the decision for switching the head direction becomes complex. A behavioral framework developed by [Srinivasan 2015], makes use of the social context for efficient generation of head behaviors. Authors have classified several types of social head gaze behaviors based on earlier studies in human-robot interaction literature. We have adopted their proposed vocabulary in our implementation (discussed in Section 5.2).

The control of robot gaze in navigation context is rather under-explored in robotics literature. A general and extensible framework requires the social context included within the gaze control system. [Zaraki 2014] have attempted a layered approach putting an attention layer between perception and gaze control layers. The attention layer selects the human with highest weighted sum of specific social features, and the robot turns its gaze toward the selected human. The specialized function for our *glance-at-human* task (see Section 5.2.2) takes inspiration from this work.

Only recently [Yoo 2015] have used a multi-criteria framework for gaze control, analogous to our proposal in Section 5.2.1, however for a static robotic head. Their proposed algorithm considers all possible discrete gaze (pan and tilt) positions in the visual field as a candidate goal for the head controller. Seven of these discrete gaze positions are periodically selected as criteria, based on certain factors that affect the human gaze, for example, faces, objects. Fuzzy measures reflecting user-defined weights of these factors are then applied to these criteria points and the point with the greatest value determines the pointing direction for the robot head. Since these criteria points are of the same type as other candidate points, the solution process is comparable to that of a prioritization method, which is the critical difference from our proposed method. As one will see in Section 5.2, we use several candidate gazing functions to select the most pertinent one depending on current perceptual inputs and context. Each function independently generates gazing points while adhering to the timing constraints required by particular head behaviors. Our criteria to assess these candidate gazing functions are dynamically rated on perceived social context.

We formulate head motion control on a mobile-robot platform as a multi-criteria decision-making problem and solve it using the analytic hierarchy process, which is a novel contribution of this thesis. We also supply the implementation specifics for a synchronized head-behavior module that model *look-at-path* and *glance-at-human* behaviors. A video-based pilot user study that evaluates our approach in real-world scenario is an auxiliary key contribution.

Before we explain our approach in the next section, a note on using augmented reality for communicating robot's navigation plans. Augmented reality projection can positively affect people's prediction of robot actions, as demonstrated by [Coovert 2014]. Although tested for being useful for humans close to a robot, effect of this method is unproven on human that are farther from the robot and looking the robot from different vantage points. Apart from that, arguably, augmented reality methods intrinsically lack the *naturalness* aspect in their application. Therefore, in this thesis we have not explored augment reality base methods for reinforcing legibility of the robot behavior.

5.2 Approach

Different studies have shown that multiple, often competing, objectives govern human head/gaze behavior. Selection of gaze direction depends on the prominent objective in the current *social context* that robot derives from its perceptual events. [Srinivasan 2015; Srinivasan 2012; Srinivasan 2011] have neatly classified actions that robot a social agent should serve. This kind of organization of different gazing-functions into behavior-groups is particularly helpful for devising a generic gaze control architecture. We make use of four social head gaze behaviors:

- *Communicating Social Attention*, where robot shows interest by looking at a human.
- *Manifesting an Interaction*, where gaze turns towards objects relevant for current task, for instance obstacle-avoidance.
- *Projecting Mental State*, which includes expression of motion intent.
- *Establishing Agency*, where head gaze reinforces human-like aliveness.

In this work, our interest lies in robot gaze behavior during navigation and therefore we omitted the fifth class of actions: *regulating the interaction process*. Regulation of the interaction process is necessary only when the robot is stationary while interacting or conversing with humans, for actions like turn-taking, story-telling and so forth.

These social behaviors are the criteria upon which we evaluate each of the available alternative functions for pointing the robot gaze. Here, one should not confuse social behaviors with behavioral functions (discussed in Section 5.2.2). A behavioral function can manifest several social behaviors, for example the action where robot momentarily glances at nearby person can communicate its attention towards the person at the same time showing human-like aliveness.

Process of updating the most relevant *social behavior* is usually continuous and rapid, especially for mobile robots as they navigate through highly dynamic environments like shopping malls, museums or airports. Notably, previous work appears to neglect this high dynamic nature of robot gaze/head behavior.

5.2.1 Framework for Effective Robot Gaze Behavior

We formulate the complete *robot head behavior* during navigation task as a multi-criteria decision-making problem. One can think of a robot having multiple choices for pointing its head, such as looking into the direction where it is going to move the next moment; looking at an object to act upon; simply moving the head down to express sadness etc. The choice of the best among several alternatives requires evaluating each alternative against a set of criteria indicative of the social gaze behaviors. How important these criteria are may change as the situation, goal or task changes. Weights (a list of user-defined scalar values in current implementation) on each criterion quantifies this importance.

Distinctive choices for gazing points usually originate from different types or sources of information. For example, human detection module provides relative human location for the robot to look at. Similarly, manipulation planning module can provide information about the next object to grasp. However, often there are more than one alternatives that arise from the same source of information. Consider a situation where there are multiple humans in front of the robot, who should the robot look at? Selection within these alternatives is subject to dedicated computation methods. Thus, we propose a scheme which processes information from distinct sources using respective *behavioral functions*.

Each of the behavioral function takes a set of candidate 3D points (C) belonging to the same source and computes a single candidate point (p_f) for directing the robot gaze. Therefore, these functions are of the following signature,

$$f : C \subset \mathbb{R}^3 \times \mathcal{D} \rightarrow \mathbb{R}^3 \times \mathbb{R}^k \tag{5.1}$$

where, $\mathcal{D}$ is a set of domain-dependent factors and k is the number of criteria under consideration. Taking the example of whom to look at when the robot is navigating in area full of people, here the source of information will be a human perception module which gives a list of 3D points referring to estimated position of each tracked person's face[1]. Then the behavioral function will take this list of points and selects one candidate point representing most urgent person in the crowd to look at. For calculating the candidate person this function could use factors ($\mathcal{D}$) like speed and orientation of the tracked persons. Furthermore, this function also gives a set of weights ($\mathbb{R}^k$) that quantify importance of the function for all social gaze-behaviors.

The analytic hierarchy process (AHP), introduced by [Saaty 1990], finds a solution for multi-criteria decision-making problems by assessing and prioritizing the options. It uses a multi-level hierarchical structure of objectives, criteria, sub-criteria, and alternatives. Solution using AHP involves paired comparison of involved criteria, which gives us a weight vector (w_j). At present, however, we only consider one level of criteria for our purpose.

First the AHP subroutine computes the weight vector, and subsequently it

[1]For simplicity of implementation, human perception module could calculate approximate geometrical center that represent human face as a single 3D point

chooses an alternative that receives the maximum score. The pan-tilt control unit receives the selected 3D point (P_{head}) and starts moving the robot head towards it. Therefore, the equation for selecting the final gazing point takes the following form,

$$P_{head} = \arg\max_{p_f} \sum_{j=1}^{k} v_j(p_f) \cdot w_j \tag{5.2}$$

where $v_j(p_f)$ is a value vector assigned by the behavioral functions, it represents importance of the candidate point p_f with respect to each criteria.

A well-designed implementation is necessary to demonstrate capabilities of this framework. To this end, we have implemented two gaze behaviors that we found most pertinent for social navigation. It should, however, be clear that the framework is not limited by the number of gaze behaviors or behavioral function it can cope with.

5.2.2 Behavioral Functions

As we have observed during our review of where humans look when they navigate (Section 5.1.1), the most relevant gazing behaviors are:

(a) *look-at-path*: looking at the planned navigational path.

(b) *glance-at-human*: acknowledging nearby people with a short glance at them.

We will now detail the associated behavioral functions along with the underlying communication architecture.

(a) The *look-at-path* Behavior

In previous chapter we have seen that the standard practice in navigation planning is to differentiate between generation of a geometric path with a static map of the environment (*global-planning*), and computation of the motor commands avoiding dynamic obstacles (*local-planning*). For *look-at-path* functionality, we use output trajectory from the *local-planning* module, as it gives us the best estimate of where the robot is going in the immediate future. We reckon this functionality corresponds to two social gaze behaviors, *Projecting Mental State* and the *Establishing Agency*. Therefore, the alternative point provided by this functionality has a higher score (1.0 during the experiments) for the corresponding attributes. In our current implementation we have only used binary values for each of the attributes, but the framework is certainly allows to conceptualize and use any weight values.

The modular design of this framework for behavioral functions makes them portable and tunable across robots. However, it also requires designing these behavioral functions that can easily be tailored to a specific robots. Some robotic heads are capable of turning full 360° around the pan axis, including the robot we have used in our experiments. For human-like gazing behavior we have to limit how much the robot head can turn. Angle between the movement direction and

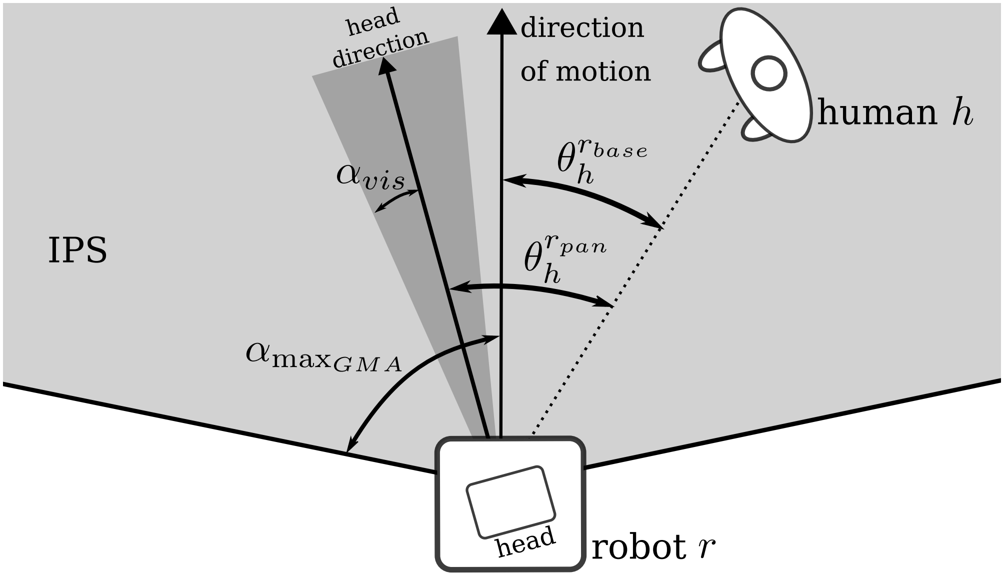


Figure 5.2 – Illustration of angles involved in gazing point computation for the robot head. $\alpha_{\max_{GMA}}$: maximum gaze-movement angle, α_{vis}: visibility angle, $\theta_{h_L}^{r_{base}}$: robot base to human angle, $\theta_{h_L}^{r_{pan}}$: robot head-pan axis to human angle, IPS: information process space (area in light grey color).

the gaze direction, formalized as *gaze-movement angle* (GMA) by [Park 2013], is illustrated in figure 5.2. Limits on this GMA value define an *information process space* (IPS), which is a conceptual area that determines the spatial boundary for observing humans. IPS is the area which humans use for avoiding potential collisions and shown to depend on the current velocity. We likewise enforce limits on robot GMA to enforce naturalness of head motions.

When the path-planner gives a trajectory in which the last point of the trajectory is outside this maximum allowed GMA ($\alpha_{\max_{GMA}}$), we limit the robot pan-angle to remain within the IPS. That means we enforce artificial joint limits on the pan joint of the robot head mechanism that resemble how much humans can turn their head sideways.

(b) The *glance-at-human* Behavior

When the robot detects humans in its environment, it activates a second function for calculating an alternative gazing point that leads the robot to perform a saccade like behavior towards the human. Since each of the behavioral functions provides only one alternative for the decision process, when the robot detects multiple humans it favors the one that requires the most urgent attention.

For estimating which human to look at first the robot takes into account both the relative position and velocity of all tracked humans, line 5 in algorithm Algorithm 5.1. This ranking scheme takes inspiration from the experiments of [Kitazawa 2010], which shows that it is preferable to look at the human who is coming towards the robot compared to the one who is going away. Additionally, we also use velocity difference between robot and human in this calculation which makes the robot to first look at the person who is coming faster towards the robot compared to the one who are walking slowly.

Algorithm 5.1 codifies this relatively straightforward procedure. This algorithm is running iteratively on every update from the human tracking module. Here, the COMPUTE_ANGLES subroutine simply calculates the angles shown in figure 5.2. In addition to that, we also need to keep track of humans that are already acknowledged by the robot (in set $\mathcal{H}_L$) to avoid triggering multiple saccade behaviors on new human position updates.

Algorithm 5.1 HUMANSUPDATE $(\mathcal{H}, \vec{r}, \mathcal{J})$

$[\theta_{h_L}^{r_{pan}}, \theta_{h_L}^{r_{base}}] =$ COMPUTE_ANGLES $(\mathcal{J}, h_L)$
if $\theta_{h_L}^{r_{pan}} < \alpha_{vis}$ **or** $\theta_{h_L}^{r_{base}} > \alpha_{\max_{GMA}}$ **then**
    $\mathcal{H}_L \leftarrow \mathcal{H}_L \cup h_L$
else
    $$h_L = \underset{h \in \mathcal{H} \setminus \mathcal{H}_L}{\arg\max} \frac{(\vec{r} - \vec{h}) \cdot (\dot{\vec{r}} - \dot{\vec{h}})}{\|\vec{r} - \vec{h}\|^2}$$
end if
return h_L

The procedure HUMANSUPDATE receives a list of position of currently tracked humans $(\mathcal{H})$, current robot position $(\vec{r})$ and positions of robot joints $(\mathcal{J})$. The algorithm returns single human (h_L) position for the robot to perform a saccade behavior towards that human. We have defined a visibility angle α_{vis}, the robot considers that it has acknowledged a human once found within this angle, consequently the *tracking-id* of acknowledgment human gets added to $\mathcal{H}_L$set. The robot suspends triggering of further saccade behaviors during an active saccade. The vector $\vec{h}$ represents the position vectors for the human at whom the robot is currently executing a saccade behavior.

Three of the social gaze behaviors, *Establishing Agency*, *Communicating Social Attention* and *Manifesting an Interaction* afford this *glance-at-human* functionality, therefore, we set the respective weight attributes to a higher value (1.0 during the experiments). We do not prune the head-tilt angle (β) in this function, the head-tilt motor controller in our robot already enforces these limits.

We also programmed the head motor controller for the Spencer robot used in our experiments. On each point-head request the head motor controllers compute the pan and the tilt-joint velocities with a hyperbolic function that is proportional to the difference between current $(\mathcal{J})$ and required joint states. Such control made the robot head move fast and smooth, somewhat looking like how humans turn their heads.

5.2.3 Implementation Details

We have used the Spencer robot for testing our approach. Full software architecture of the Spencer robot includes several software components for different capabilities including person and group perception, mapping, localization, social signal processing and so on. Figure 5.3 shows a simplified architecture diagram with components that are relevant for head-body coordination during navigation. Nevertheless, this scheme can easily adapted to any mobile robot that possesses a pan-tilt head unit.

For path planning during our experiments, we have adopted directional-cost based algorithm presented in Chapter 4. This human-aware path planning algorithm makes the robot slow down when in the likely event of a possible collision with nearby human, while avoiding abrupt changes in the path direction. Since the *look-at-path* behavior is tightly coupled with the local-planning module, sudden changes in the path direction would result in an unwanted trembling head motion. The directional cost-based algorithm avoids such abrupt change in motion direction and, therefore, it is particularly useful for our application.

As the local-planner is responsible for dynamic obstacle avoidance, at every iteration it generates a trajectory (a series of 2D points) and computes velocity commands for the robot base controller. The *Head Behavior* receives this planned trajectory, and selects the endpoint of the trajectory to compute the next gazing point for the *look-at-path* behavior. We add a scalar constant value to the Z-coordinate to adjust the height of the gazing point, and thus make it fall within the turning limit of the head-tilt axis. To regulate the anticipation distance for small trajectories, we added a parameter that extends the trajectory provided by the local-planner up to a specified (configurable) distance.

We employed the same OptiTrack motion capture system that we used while testing navigation planner also here for human tracking. It publishes positions and velocities of detected humans (denoted by set $\mathcal{H}$) at a certain frequency (10 Hz during our experiments)Furthermore, we also used velocity-obstacle based human pose prediction module and directional-costs based human-aware path planning scheme during the experiments and the user-study. In the architecture schema illustrated in figure 1.2, the *Head Behavior* module is the original contribution for this work. Other software components required for full experiment to work (mapping, perception, localization as well as sensor drivers and motor controllers modules) are comprised of our previous work and widely used open-source ROS modules.

The *Head Behavior* software module is also well-integrated with the ROS middleware's standard framework for navigation, `move_base`, and tested on Spencer robot that we deployed for duration of three weeks at Amsterdam airport Schiphol for guiding passengers [Triebel 2015]. Nevertheless, the module is not limited to that particular robot. In fact, any robot with a *head-pan* and *head-tilt* joints would benefit from it[2]. It was a trivial task to integrate this gazing behavior scheme in the PR2 robot during the experiments for cooperative navigation planner (presented in Chapter 4).

[2]Source code for the module is available at `http://harmish.in/HBC/code`

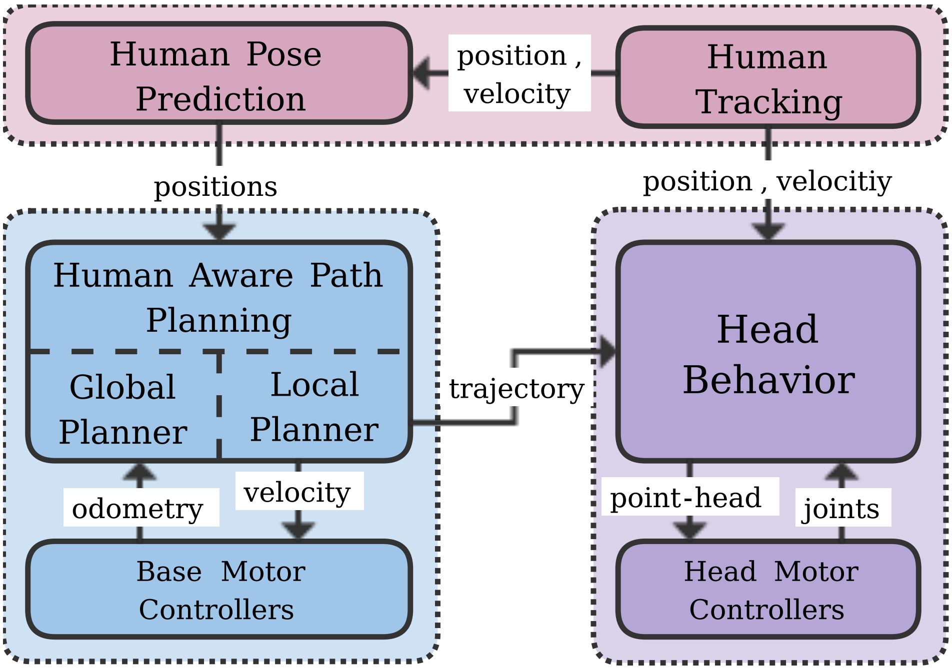


Figure 5.3 – Software components and data flow for robot gazing behavior scheme.

5.2.4 Synchronization of Head and Body Motion

Synchronization between head and body joints of a robot, as well as between robot joints and tracked human positions is very important to achieve a desired and meaningful behavior. It is only when all the expressive modalities achieve synchrony we see a robot behavior which is not only fluent in interaction but also adds the *naturalness* aspect to robot movements. It requires substantial engineering and programming efforts to accomplish good harmony of whole body movements. Perhaps this is one of the reasons, why we see a big disproportion in the implementations of gazing behaviors among static and dynamic situations.

With our implementation, we were able to achieve the desired collaboration in human-robot path crossing scenario as shown in figure 5.1. Time-series plot of commanded and actual pan-angles presented in figure 5.4 is for a scenario in which human moves across the robot's path in front of the robot. The robot starts with moving and looking towards its navigational plan; it executes the *glance-at-human* behavior when the human appears on the scene, and after finishing the glance behavior the robot again turns its head towards the plan until it reaches the goal position.

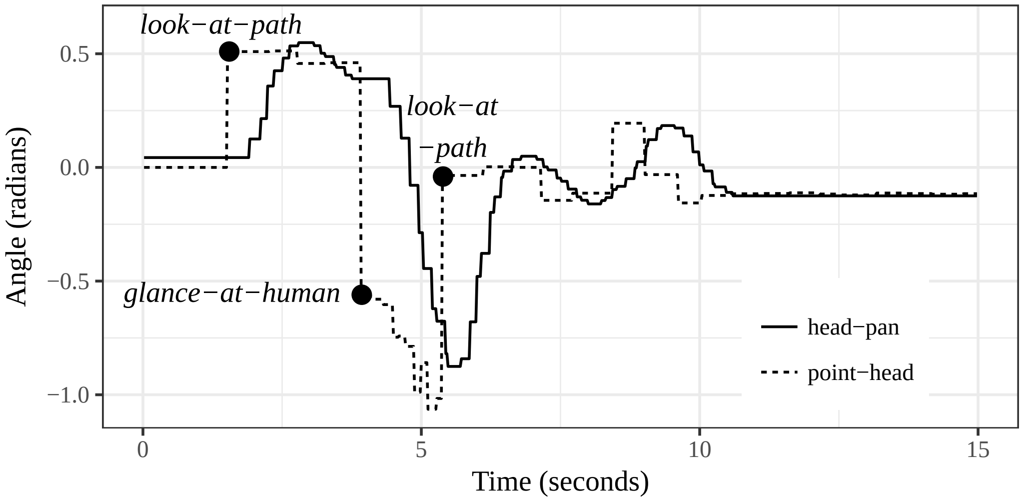


Figure 5.4 – An example time-series plot of commanded gaze points (point-head) and the angle between the front position of robot base and head-pan axis (head-pan). Marker-points indicate change of behavioral functions.

5.3 Evaluation through User Study

Based on this design of head behaviors, we anticipate that human's perception of the robot motion will improve both in interactive and non-interactive situations. We set-up a video based user study to determine whether it is true that the proposed head movements visibly improves the robot motion quality. Our hypotheses for the outcome of the study were following:

Hypothesis 5.1 (H5.1): *Anticipatory head movements during navigation will positively affect the perception of robot's navigation intents.*

Hypothesis 5.2 (H5.2): *Head behavior with both of the look-at-path and glance-at-human functionalities will be evaluated as more favorable over no head movements.*

Hypothesis H5.1 is related to objective improvement in robot motion, whereas the hypothesis H5.2 is linked to the subjective improvement since it concerns participants' subjective choice about the robot behavior.

5.3.1 User Study Design

There is a precedent for video based studies performed with real [Syrdal 2008] as well as a simulated [Takayama 2011] robot for human-robot interaction. Results of the study by [Syrdal 2008] suggest that video prototyping is an excellent source to gain insights regarding user experiences in assessing the human-robot interactions. Furthermore, the video prototyping allowed us to engage geographically diverse set of participants while providing a consistent experience for each of them. We used freely available online service[3] to create a web-based form for the user study. Web-based user study made it relatively easy for us to produce the questionnaire in four different languages: English, French, German and Spanish as well as invite participants from three different countries: France, Germany and Mexico.

The Spencer robot platform includes a two degrees-of-freedom head. Head pan joint can fully turn towards the back of the robot, while tilt joint is physically limited to move up to 30° up and down from the straight looking position. Head of the robot has two passive eyes that can only move horizontally to the left and the right, however, they were not used during our experiments. The robot is also equipped with a display, which was switched-off during the video recordings.

We recorded seven videos in an experimental area of about 5.5 by 9 meters. Surrounding place to the experimental area was partially covered with large wooden boards to reduce any distraction caused by it. Lighting conditions were maintained same throughout the recording. And we removed the audio from the recorded video files before it embedding them to the online questionnaire for the participants. During experiments the weight for *Establishing Agency* criteria was assigned the value $w = 0.9$, remaining weights were set to the value 1.0. Earlier we mentioned that the total distance where the robot looks depends on the immediate path of the robot, in these experiments the robot was planning for a time-window of 4 seconds and thus watching the point where it will reach 4 seconds in the future. These values were empirically found to best suite our task, however, they can experimentally be learned and adapted on-line. Using machine learning techniques to find out weights for used criteria in the given social context is a research direction we aim to explored in future.

The first four of the recorded videos did not involve any human. Here the task of the robot was to move from the start position as shown in figure 5.5a to one of the end positions, either going towards the door or the corridor. For both of the goal positions, we recorded two videos, one with the *look-at-path* behavior enabled and other without (head position fixed, looking straight ahead). After recording, we cropped all four videos until the frame where the body of the robot starts moving towards the goal direction. Only the two videos with *look-at-path* behavior the robot head starts moving at the marked position in figure 5.5a.

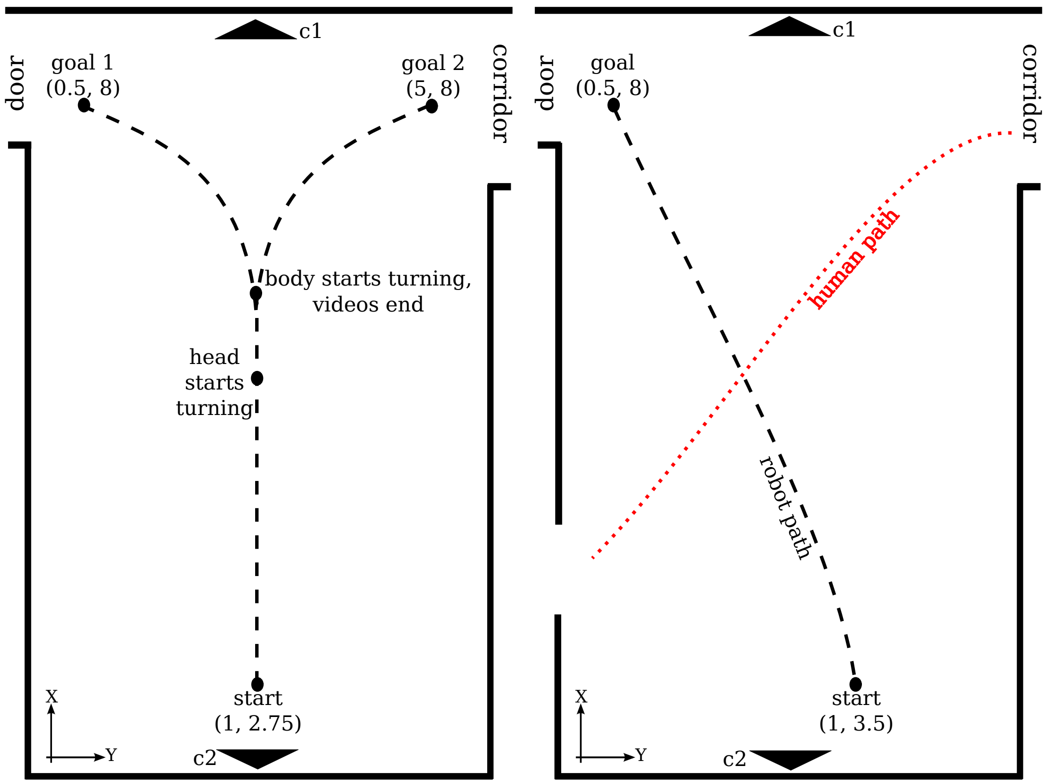
In the last three videos we recorded, a human played the role of an interferer who crosses the robot path as shown in the figure 5.5b. The human wore a light-weight helmet with four passive reflective markers placed on it. These markers gave the tracking position of the human head. We manipulated the condition during the recording of these three videos as following:

(A) Robot head position fixed relative to the body looking straight in front of the robot with both head-behavior functions disabled.

(B) Only the *look-at-path* function enabled.

(C) Both the *look-at-path* and *gaze-at-human* enabled.

[3] The service we for carrying out online user study:`https://www.typeform.com/`



(a) Layout for recording videos where robot goes turns either right or left. The robot head start turning towards the eventual goal position a little before the body when *look-at-path* behavior is switched-on. This situation do not involve any human being during the recordings.

(b) Layout for human-robot path crossing videos. The human comes from the corridor when the robot is going towards the door.

Figure 5.5 – Illustration of the layout of experiment space.

All of the videos in the study were recorded with two viewpoints, one in the front of the robot and another in the back, denoted by c1 and c2 respectively in Fig. 5.5[4]. As previously mentioned, we embedded these videos in a web-based questionnaire[5] Each participant saw the experiment videos from both viewpoints before answering respective question. We created several web-forms with all possible order for showing the videos and translated them in all four languages we used to conduct the user study. To avoid fake or bogus answers to the online questionnaire, we send an invitation to the participants only via e-mail messages. Email message contained links for language the participant prefer to answer. Once participant clicked on the link for their selected language, the link randomly redirected them to one particular web-form with one particular order for showing the videos. The online web-form also included a short introductory message about the study procedure and questions for

[4]Link to view all the recorded videos in full length: `http://harmish.in/HBC/video`.
[5]Link to an example on-line form used as questionnaire: `http://harmish.in/HBC/form`.

collecting demographic data before showing the videos. Figure 5.6 shows screenshots of the web-form.

5.3.2 Study Procedure

The study procedure was divided into two parts:

Part 1 where we only manipulated the *look-at-path* behavioral condition.

Part 2 where we manipulate both the *look-at-path* and *glance-at-human* behavioral conditions.

For Part 1, participants were randomly assigned to watch videos where the actual navigation goal was one of the two positions shown in figure 5.5a, either the *door* or the *corridor*. This way we remove any effect on the goal direction caused by the experimental setup or the robot appearance. Once assigned to an option, participants watched one video where movements of the robot head looked straight in front of the robot and another video of robot exhibiting the *look-at-path* behavior. The order of showing these videos was also counterbalanced across participants, to avoid carry-over effects in this independent measures experiment. As discussed earlier, to remove any influence of robot's body-movements on participants' answers, we stopped the video just at the moment where the body starts turning towards the goal position, figures 5.7b and 5.7a show the snapshots of these moments.

After watching each video, the web-form posed the following question:

Q1,2 Where is the robot going?

To respond to this question, the web-form asked participants to choose one of the two goal positions of the robot, as shown in figures 5.8a and 5.8b.

Participants' response to these videos will help us test the hypothesis H5.1. While recording these videos we have added a couple of virtual obstacles in the planning cost-map to "force" the path planning system to produce a path that makes the robot move straight forward for a certain distance before initiating the goal-directed turning motion.

The second part of the procedure concerns the hypothesis H5.2. After answering the above mentioned question, participants saw a video with the conditions (A), (B), and (C), as listed before. Again, we counterbalanced the order of these conditions as they appear in the video across participants. After watching this video, the participants chose which behavior of the robot they like the most and the least, by answering following two questions:

Q3 (a) Which behavior of the robot do you like the most?

Q3 (b) Which behavior of the robot do you like the least?

Once participants register their liking, we requested them to see the videos with condition (A) (that is, with no head behavior) and condition (C) (that is, both

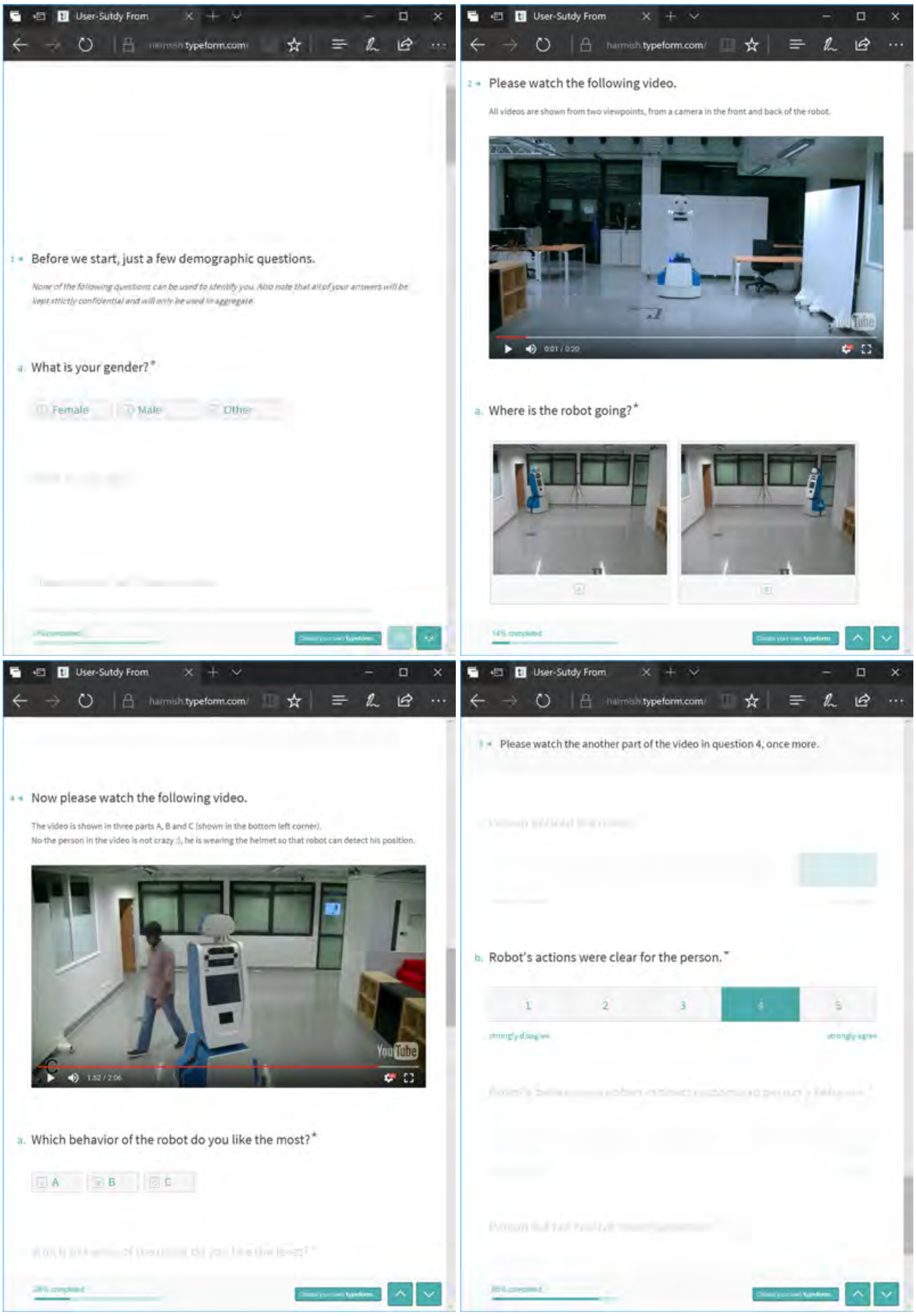


Figure 5.6 – Example screenshots of the web-form used for the online user study. The questions follow one after another in a modern one-page website that can easily be viewed on different screen sizes.

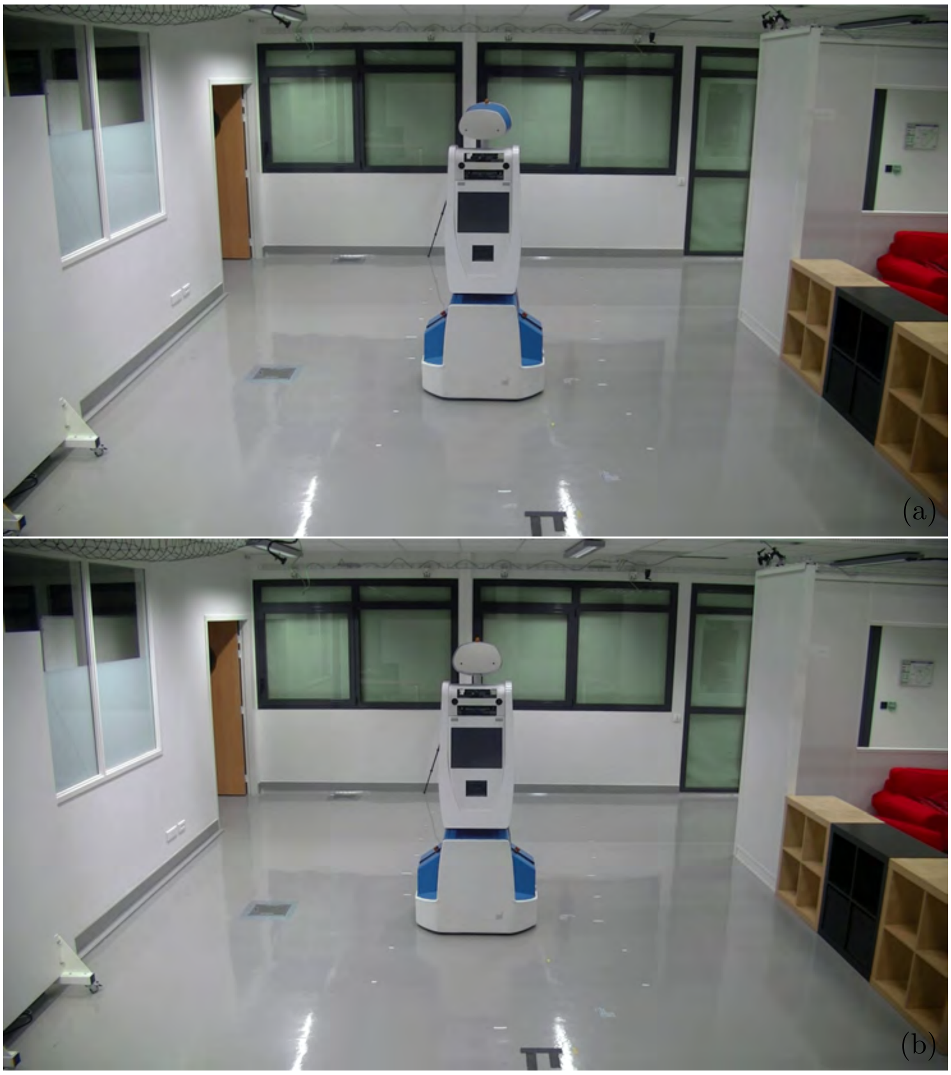


Figure 5.7 – Moment where the videos stopped during Part 1 of the user study. Figure 5.7a shows the case when the *look-at-path* behavior was enabled, while figure 5.7b shows robot pose without any head behavior.

look-at-path and *glance-at-human* behaviors enabled) once more, in counterbalanced orders. After watching each video participants rated their responses to the following statements on a five-level Likert scale ranging from *strongly disagree* (= 1) to *strongly agree* (= 5):

Q4,5 (a) Robot noticed the person.

Q4,5 (b) Robot's actions were clear for the person.

Q4,5 (c) Robot's behavior was often in direct response to person's behavior.

Q4,5 (d) Person did not receive robot's attention.

(a) Robot at goal 1 (door)

(b) Robot at goal 2 (corridor)

Figure 5.8 – Goal positions for Q1,2 *look-at-path* behavior.

These statements where participants rated sociality of the robot behavior took inspiration from the social presence measures introduced by [Harms 2004]. Since our study captures a third person's perspective, we have selected only four (of total six) dimensions of the social presence measurement test; removing the dimensions that measure affective and emotional aspects of the interaction episode. The four selected dimensions are *co-presence* (Q4,5 (a)), *perceived message understanding* (Q4,5 (b)), *perceived behavioral interdependence* (Q4,5 (c)), and *attentional allocation* (Q4,5 (d)).

5.3.3 Results

We collected data from N = 126 participants between age 17 and 59 ($Mean = 27.82, SD = 6.44$) who volunteered for the video based user study. Out of 126 participants, 67 choose to answer in English, 36 in French, 20 in Spanish and 3 in German.

For the first two videos we manipulated the behavioral condition *look-at-path*, our aim was to get an objective measure of improvement in robot motions. In the condition without any head behavior, we would expect the ratio of the participants predicting the correct goal position be around 50% (that means the null hypothesis is true). Results of answers to Q1,2, plotted in figure 5.9, show an increase in the ratio of participants who inferred the correct goal when *look-at-path* condition was *true*. The difference in participants' predictions of the robot goal is significant according to McNemar's chi-squared test ($\chi^2(DF = 1, N = 126) = 50.704,\ p < 0.0001$). On the other hand, when the head behaviors are absent, the null hypothesis (that is, the head behavior has no effect on how participants perceived robot's navigation intents) is true.

Since the length of videos with and without *look-at-path* behavior was same, these results show that, in given amount of time the *look-at-path* behavior results in more accurate perception of robot's navigational goal. Therefore, we found support for hypothesis H5.1 concerning accuracy. These results also confirm that without

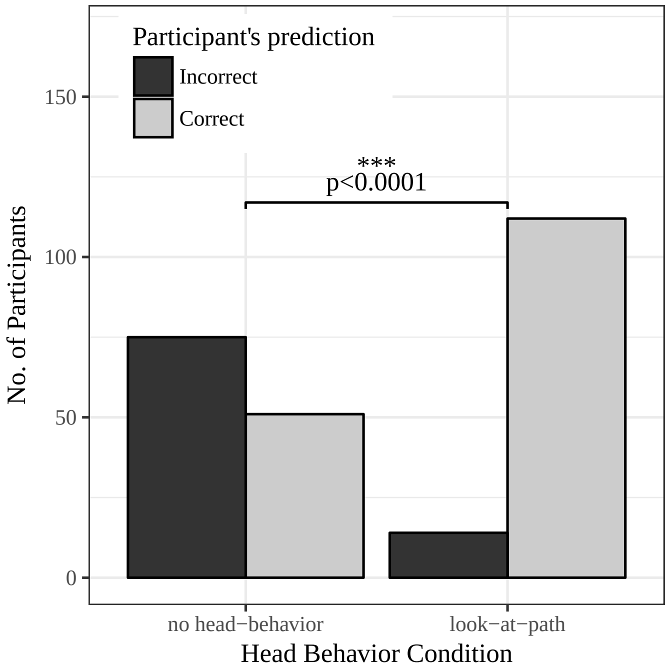


Figure 5.9 – Bar chart showing success-ratio of 126 participants' prediction of robot goal. It indicates significant improvement in correct prediction of the robot goal with *look-at-path* behavior.

any head behavior cue the probability of successful prediction of the robot goal is equivalent to what is expected due to chance.

Concerning the Part 2 of the user study involving a human-robot path crossing scenario, figure 5.10 shows the box plot of participants' liking about above listed three behavioral conditions (A): the robot head looking straight ahead, (B): the robot head looking at its path, and (C): the robot head looking at path and throwing a glance at the human. The Friedman rank sum test reveals significant differences in participants' liking about the three types of robot motions, $\chi^2(DF = 2, N = 126) = 50.683, p < 0.0001$. Moreover, the post hoc analysis with Wilcoxon signed rank test shown significant differences between all three pairs of conditions as well (all tests returned $p < 0.0001$). That means, the participants significantly preferred the robot motions condition (C) over the other two variants.

Finally the figure 5.11 shows participants' responses to the social-presence-measure questionnaire among four selected dimensions. The comparisons are, for easier interpretation, annotated with significance levels using "*", *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$. We see significant improvement along the *co-presence* dimension with (C) condition. Complementary to this, we also see an increase in perceived attention given to the human by the robot (note the negative nature of Q4,5 (d) for *attention allocation*). Both of the above mentioned results indicate an increase in robot's awareness towards the human.

Participants' response towards *Perceived message understanding* measure hints

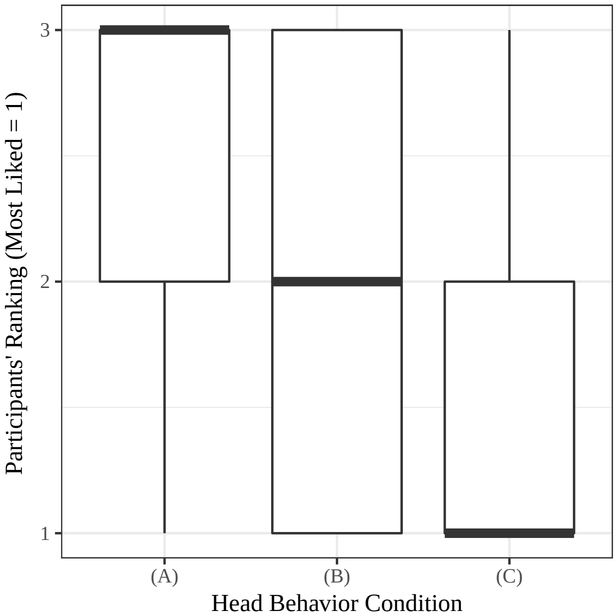


Figure 5.10 – Box plot of 126 participants ranking over three head behavior conditions (A), (B) and (C). Condition (C) where both *look-at-path* and *glance-at-human* where enabled, turns out to be the most liked behavior by the participants.

at overall enhancement in fluency of the interaction episode, which is probably due to little jerks showed by the robot while slowing down and a slight overshoot in the *glance-at-human* procedure (resulting from lower planning frequency). The *perceived behavioral interdependence* dimension also shows some improvement, meaning the robot being more responsive to the human presence.

These results combined with participants' ranking partially support the hypothesis H5.2.

5.3.4 Limitations

There are several limitations present in the design of this study. It gives us a third person's perspective on human-robot interactions, which is not sufficient to adequately assess the interaction episode quality with all dimensions of the social-presence-measure test. The results are nonetheless meaningful, as suggested by [Syrdal 2008] it gives valuable insights regarding user experiences related to how participants assessed human and robot interactions.

A comprehensive study should also include effect of auditory clues which could directly affect the perceived safety of individual participants. To focus on effect of head behaviors on the person directly interacting with the robot, this study involved only one person. Navigation in areas like shopping malls, airports would involve interaction with more than one person at a time, proper assessment of how this head control algorithm makes robot behave in-the-wild requires a larger scale user-study.

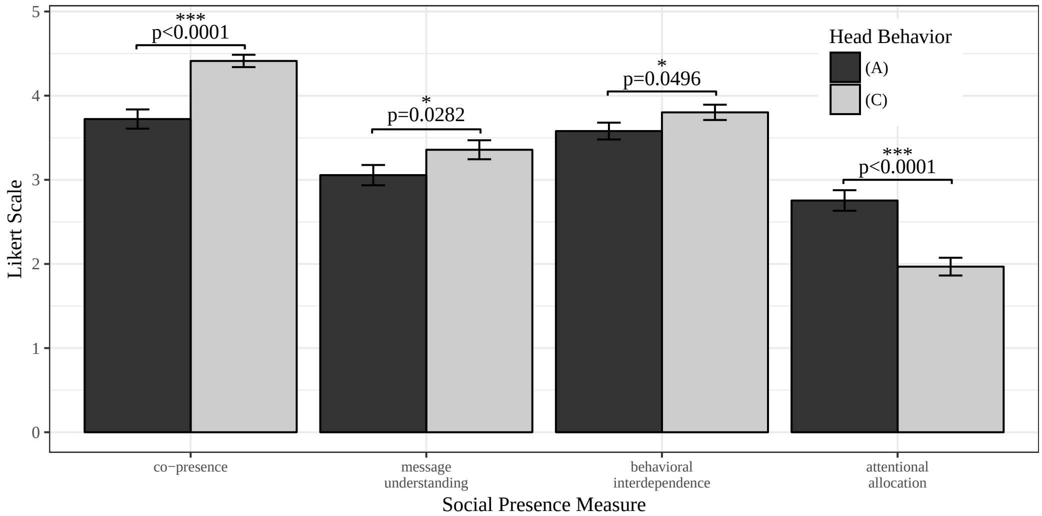


Figure 5.11 – Bar charts of participants' responses to social-presence-measure questionnaire.

Nonetheless, we have successfully tested this head behavior for the period of one week while testing the robot Spencer at Amsterdam airport in May 2016.

The context of the experiment, a path crossing scenario within a relatively small area, limits how much we can generalize our results. Subsequent experiments would need to involve other shared navigation situations to further elaborate the relationships between a robot's head expressions and how humans interpret them as cues for robot's navigational intents.

We are also aware of the limitations of the current implementation. The robot head-pan axis sometimes overshoots while glancing at a human, particularly when the human is moving at higher velocities, because the *Human Pose Prediction* module is not yet fully integrated with the *Head Behavior* module. Secondly, the *glance-at-human* behavior should mark all humans that pass within the α_{vis} angle as seen while a saccade towards another human is underway; a ray-tracing based technique would come to rescue here.

Possible Learning Effect

During the study participants watched the videos multiple times, once for rating the behaviors and a second time while responding to the social-presence-measure questionnaire. Therefore, there arise a possibility of carryover or learning effects. Just before completion of the study, we showed an additional video to each of the participants similar to that of the Part 1 videos, however with a different starting pose of the robot. This new starting pose had the same position as in figure 5.5b but the front of the robot oriented towards the negative Y-axis. This way we showed the robot-head already moving at the start of the motion as the robot orients itself to reach either of the goal positions. The purpose here was to test if it leaves any "learning" effect on the participants about the robot behavior. Results of participants' prediction show no significant improvement over the results of figure 5.9, McNemar's chi-squared test returns p-value of 0.5. No improvement in accuracy of robot goal prediction by participants suggests a lack of "learning" effect associated with the *look-at-path* behavior, that means participants don't need to observe robot behavior for a longer time to accurately infer robot's navigational intents.

5.4 Summary

This chapter contributes in two aspects.

Based on our literature review on gaze behaviors from multiple disciplines including human-robot interaction, psychology, neurobiology, and computer simulation, we put forward a plausible framework to determine dynamically an optimal gazing point. We formulated the robot gaze selection as a multi-criteria decision-making problem and solved it with the use of an analytic hierarchy process based algorithm. The proposed algorithm first employs dedicated functions to provide gazing points based on different sources of information. Each candidate functions are then globally evaluated on account of perceived social context. Along with the framework, we provided implementation details that testify our proposal.

Secondly, to assess the resulting gazing behavior of the robot, we carried out a video based user study that supports our hypotheses and suggests improvements in participants' perception of the robot motions both objectively and subjectively.

Chapter 6

Normative Navigation Behaviors

Contents

Human-robot interaction research aims at *humanizing* robots by reproducing some aspects of humanity in their character. The hope is to make robots that can socialize, live alongside us, talk to us, work and walk with us. Humans have devised mechanisms for interacting with each-other over millions of years of evolution. They are largely comprised of hidden social rules and norms as built-in instinct in us. For robots, being social agents means they also have learned and ingrained these rules and norms in them. In this chapter we introduce our approach to learn and imitate a couple of the social behaviors that are pertinent to navigation process.

One of the task we do everyday is approaching someone for initiating a conversation. Intuitive as it is to us, it involves certain social norms we follow subconsciously like creating eye contact, focusing our attention whom we are approaching and moving in a way that makes us fully visible to them. We introduce two navigation strategies, first where the robot learns a proper way to approach a person in open area without any other obstacles, and second where robot creates a cost-map of the areas surrounding itself and the person it is approaching and leverage standard path planning techniques to plan a path in this map.

The service robots like Spencer need to master the task of guiding people. While guiding it need to constantly estimate if people are still actively following and react to their needs in terms of their ability to move faster to slowly. In this chapter we also provide our solution for effectively planning and executing a guiding behavior. Furthermore, we also investigate how a robot can influence behavior of the people being guided according to urgency of task, willingness of the people to follow or other environmental stimulus.

6.1 Approaching a Person

One way to deal with the subject of approaching people is focusing on heuristics to calculate a point in the front of the human moving straight to that point, which requires dedicated method of detecting *pedestrians' intentions* [Satake 2009; Kato 2015]. Another way is to generate an intermediary waypoint that make robot move towards humans facing them in the front [Shomin 2014], however, applicability of such methods in highly dynamic situation is questionable. A more generic way of generating human like approaching behavior is to use data of people approaching each other and create a value map around the human position that drives the robot to move towards the low value region with a greedy planner [Avrunin 2014]. This method, however do not consider other obstacles in the environment that may need entirely new value map for different situations. In our work, we also focus on how robot shall move to reach an engagement given previously generated demonstrations. A critical difference with the related works is that we find the goal position given the demonstrations instead of hard-coding it.

6.1.1 Modelling Approach Task

Inverse Reinforcement Learning (IRL) method enables a robot to learn a policy using discrete and finite Markov Decision Process (MDP). We modeled the MDP states such that they define robot's relative position and orientation with respect to a human, similarly to the work of [RamónVigo 2014] where their robot learns to navigate among humans based on recorded trajectories of people. We specifically address the problem of *how to approach people to interact with them*, therefore our algorithm is able to find a plausible goal for the robot and a socially acceptable path to reach this goal. This requires a specific model representing the space around the humans and appropriate trajectories for homing on them.

We assume that the expert from which we want to learn the approach behavior is modeled by an MDP. Thus, our problem defines the tuple $\langle S, A, T, R, D, \gamma \rangle$, which is a standard MDP model with finite sets of states S and actions A, a transition probability function T, a reward function R, a discount factor γ plus the added variable D which represents demonstrations given by an expert. An IRL algorithm finds an optimal policy $\pi^*(s)$ (suggesting which action should robot take when in state s) that maximizes the probability of reward. While finding the optimal polity, the IRL algorithm utilizes the demonstrations by an expert, thus the optimal policy would represent what the expert demonstrator would do in any given state. This is a standard procedure applied in IRL based algorithms, for detailed description of different IRL algorithms and how an IRL finds the optimal policy we can refer to [Ng 2000].

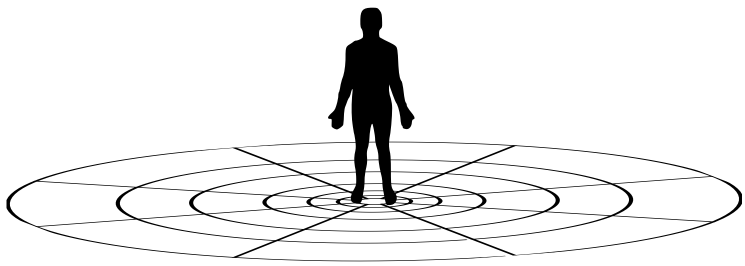


Figure 6.1 – Human Centered State in MDP

The important detail is how we define our states and actions. The definition of states and actions depends on the strategy for learning, which we discuss in the following two sub-sections.

6.1.2 Learning Path for Approaching

First of the two methods we developed to tackle the approaching behavior is a path planner that generates paths directly based on the response of the IRL algorithm.

States

We design MDP state robot's relative pose with respect to the person it approaching in a human-centered polar representation of the area. Figure 6.1 depicts this representation. The state has two component: a distance component d and a state angle component θ. We d along a quadratic based function dividing area around human in to n sections allowing us to have higher precision in the region near the person. For the angle component θ, we divided the region into m sections with each section being $2\pi/m$ radians wide. Thus, the a state of the MDP is in $\mathbb{R}^{n\times m}$.

Actions

We define a set of 5 actions described below.

1. (θ_c, d_c): stay in the same place.
2. $(\theta_c + 1, d_c)$: move right in $\theta_c + 1$.
3. $(\theta_c - 1, d_c)$: move left in $\theta_c - 1$.
4. $(\theta_c, d_c + 1)$: move forward in $d_c + 1$.
5. $(\theta_c, d_c - 1)$: move backward in $d_c - 1$.

Where θ_c represents the current angular state and d_c the current distance state.

Based on the states and actions definitions, the transition matrix is simply an agglomeration of the 5 actions for all the $n \times m$ states, and thus having a sparse matrix of size $S \times A \times S$. In this method the number of features is equal to the number of states multiplied by the number of possible actions.

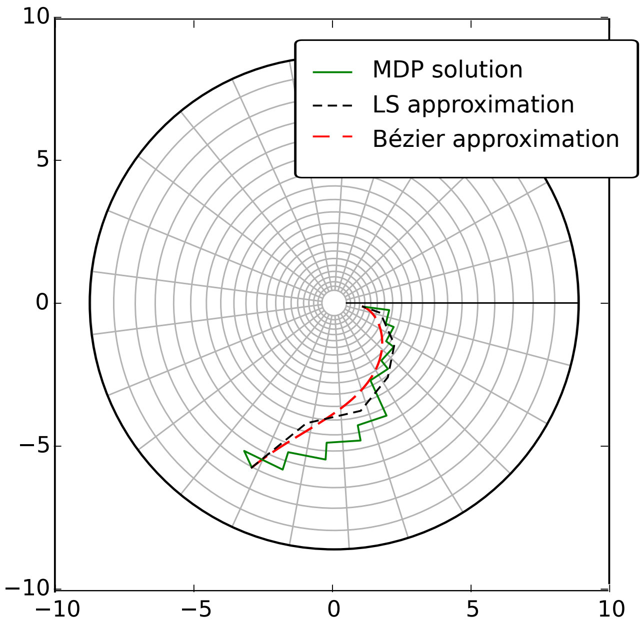


Figure 6.2 – Path generated by IRL and post-processing. The green line represent the optimal policy generated by from the MDP learned with IRL. The dashed black line represents the least square approximation of the path in green. The dashed red line is a Bézier curve created from the set of parametric points representing the path in green and the initial orientation of the robot.

From the demonstrations given by an expert, IRL algorithm returns and MDP with learned state transition probabilities. Therefore finding the optimal polity in this MDP will make robot move along a sequence of states which form the optimal trajectory for approaching a person. Connecting geometrical centers of the areas covered by each state gives us a zigzag line pattern as shown in green in figure 6.2. These trajectories are the path that what this method learns.

Smoothing the Learned Path

Since the learned trajectories are discontinuous in terms of velocities, we apply smoothing filter on them. First we transform the paths into the global map frame, and then re-transform with two parametric data sets such that the first (x, y) coordinate give use two points (t, x) and (t, y) for all time-points $t = 1$, $t = 2$ and so on. Next, we can apply least squares function approximation to both parametric data sets resulting into a smoother path, shown with dashed black line in figure 6.2.

However, the least squares based function approximation do not take into account orientation of the robot. Instead, Bézier curves can smooth the trajectory to respect robot orientation. We use the aforementioned parametric data set version of the path as control points for Bézier curve generation with additional control point

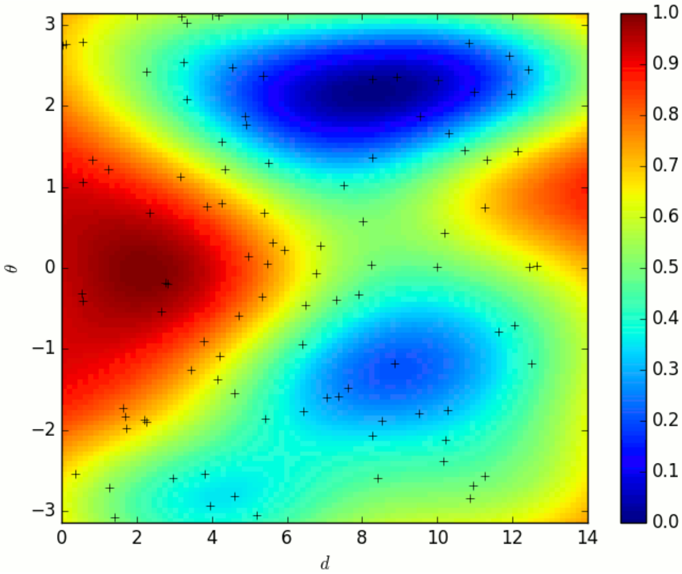


Figure 6.3 – Costmap generated with $w^T\Phi(s)$ in an unfolded polar map. The blue + signs represent the center of all the RBF used in this task.

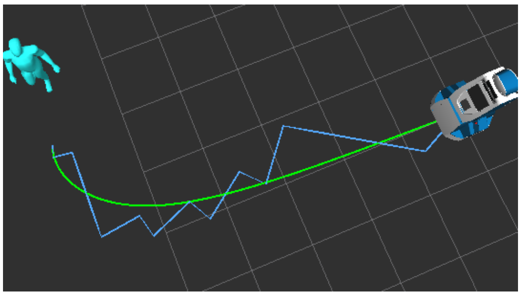
that represent initial orientation of the robot. The resulting path is also shown in figure 6.2 in dashed red line.

6.1.3 Learning Costmap for Approaching

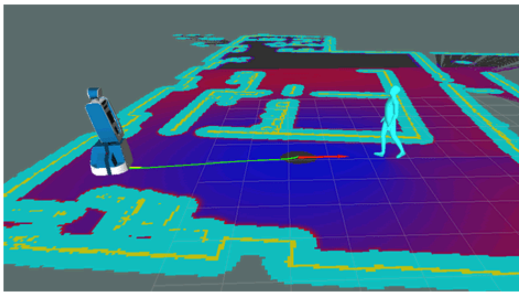
In the second method we build a costmap for the approach scenario. The main difference with the previous case is the use of (Gaussian) Radial Basis Functions (RBF) for representing features of continuous state space. Here we sample n random 2D points from ranges $r_d \in [0, 14]$ and $r_\theta \in [-\pi, \pi)$ for each dimension. These random points serve as the mean for the 2D Gaussian we use as RBFs. Standard deviation for the Gaussian RBF were set to the quarter of the range for each axis. Thus, the state transition representation is of the form $\Phi(s) = [\phi_1(s_{\text{coord}}), \phi_2(s_{\text{coord}}), \ldots, \phi_n(s_{\text{coord}})]$, where $\phi_i(s_{\text{coord}})$ is the i^{th} RBF and s_{coord} is the cartesian center of the state s. Then we set $\Phi(s, a) = \Phi(s)$ given than we intended to use this information in a costmap, which is only represented by the states and not the actions, differently from previous strategy of learning a path. For MDP it means that the there is only one action possible in every state, and IRL effectively learning the state transition probabilities from expert data.

The learning process gives us a weight vector w vector for the basis functions. With that we can directly generate a costmap of the area centered at a human having smaller costs of the grid cells that the robot should traverse to display a social approaching behavior. Since we use polar coordinates of representation for learning, we must convert it to cartesian space before the robot can use the costmap to plan a path.

An advantage of this strategy is we can now combine this costmap with other



(a) Path learned by finding optimal MDP policy (violet) and smooth curved generated with least squares fitting and Bézier lines (green).



(b) Layered costmap based navigation with IRL learned layer.

Figure 6.4 – Experiments with proposed strategies to plan a path for approaching a person.

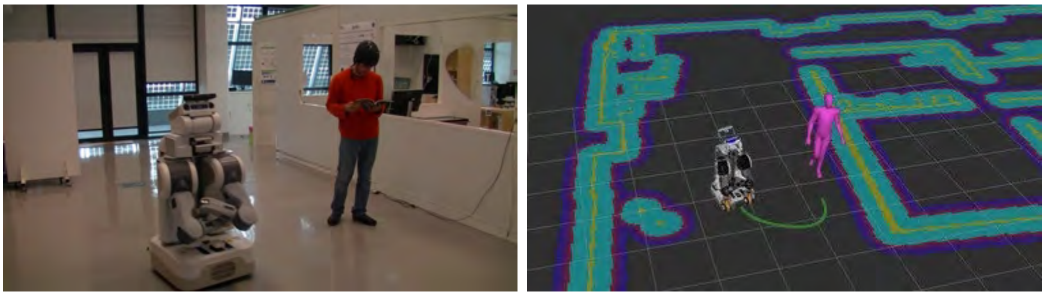
costs in the are representing other static or moving obstacles. We created a costmap layer plugin for the `move_base` framework of ROS, which is based on the work of [Lu 2014b]. Our plugin sets the cost of particular cell only if it is higher than one already set by the previous layer. Then we use Dijkstra's algorithm [Dijkstra 1959] (as used by default in the `move_base` framework) to calculate the best path while setting the goal position of the planner the position in which the maximum reward is found in the costmap layer and which orienting opposite the human.

Figure 6.3 shows an example costmap learned with this strategy. It is feasible generate costmap with arbitrary resolution because the features we use here are continuous functions. Even when we have discrete number of bases functions, the values of the coordinate system is in $\mathbb{R}$ for distance and angle.

6.1.4 Experiments

We employed ROS message mechanism to simulate the human movements by setting their positions and velocities. We also used it to generate sample trajectories of robot while approaching humans, where he robot is manually controlled during different approaching scenarios. We collected a set of demonstrations with this experimental platform for the learning process. Figure 6.4 shows paths generated by the path learning strategy as well as costmap learning strategy.

As an early stage tests, we integrated the algorithm with the PR2 root and tested in a close area. We used the same motion capture setup for detecting and tracking the person as during the tests of cooperative planner discussed in Chapter 4. Figure 6.5a shows a person wearing a helmet with markers that we use for tracking. While figure 6.5b shows the visual representation of the robot, human and the proposed path generated by the path learning strategy. This work on learning to approach people have been published in [Islas Ramírez 2016].



(a) Scene where robot approaching a person wearing the helmet is detectable by motion capture system.

(b) Visualization of the computed path based on the path learning strategy.

Figure 6.5 – Testing with real scenario.

6.2 Speed Adaptation while Guiding

Another interesting problem in human-robot interaction is developing robots that are able to guide humans, by offering a tour of attractions in an area or simply by helping humans to reach a destination. While there are different robot guides studied and tested in real environments, from pioneers like Rhino [Burgard 1998] and Minerva [Thrun 1999] to robots with more social and interactive capabilities like Rackham [Clodic 2006] and FROG [Evers 2014], there is a need to for guiding robots with better reactive and collaborative planning capabilities. Human-aware navigation in a museum guide situation presented in [Samejima 2015] works by building environmental maps, which include information learnt from human trajectories and postures, in order to plan safe paths that do not disturb the humans present in the area. We believe that most robot guide systems are focusing either on the social aspects of the problem or on human-aware navigation, without fully considering the fundamental aspects of joint actions. Guiding is a collaborative task, where the robot doesn't need only to reach a destination, but also to ensure that its followers reach it, while providing a socially acceptable experience to them. In order to achieve this goal, the robot needs to constantly monitor its users, adapt to their behaviors and be ready to proactively help them.

In this section, we present a robot guide which is able to lead a group of people to a destination. The novelty of our approach is that the robot is able to show both an adaptive and a proactive behavior. While guiding the robot tries to select a speed that pleases its users, moreover it can also propose a new speed using environmental and task related needs.

6.2.1 Methodology

The overall system implements these behaviors using a *situation assessment* component which gathers data from different sources and maintains symbolic information, a *supervision and task planning* based on hierarchical Mixed Observability Markov Decision Process (MOMDPs) [Ong 2010] and, a human-aware navigation planning component that allows the robot to reactively navigate in populated environments.

Situation Assessment

The situation assessment module [Milliez 2014] reasons on agents and objects that are present in the environment and produces different kind of information, such as distance and orientation of a human relative to the robot, variation of the distance between a human and the robot, whether the human is currently moving, and so on. It is also able to create activity areas in the environment and link them to different types of computations. An activity area is a polygonal or circular area, which is either fixed or linked and updated with an entity's (object, human or robot) position.

For the purpose of speed adaptive guide, we employ the situation assessment module to detect group, the robot will consider as "group" the persons closer than a predefined threshold at the start of the scenario. We consider the group's size as a dynamic entity during the scenario, allowing the group to shrink when part of its members leaves for more than a predefined period of time. In this case the robot will re-evaluate the participants of the group, considering only the remaining members as part of it.

Task Planning and Supervision

With the reasoning abilities provided by Situation Assessment, the robot should guide the group toward its goal, which could be predefined or negotiated with the users at the start of the scenario. We defined a set of modules, called *Collaborative Planners*, able to choose which proactive or adaptive actions the robot should perform at each moment. This framework uses hierarchical MOMDPs as in this task system state is split in an observable set and a hidden set (which cannot be fully observed rather inferred from other observations). Moreover, we use a hierarchical framework [Pineau 2001], where the system model is split into a main MOMDP module and several MOMDP sub-models, each one related to a different action and solved separately. For the guide robot, we have a *guide model* as main MOMDP module and the *speed adaptation model* as sum-module along with other sub-modules like *task suspension model*. Here we only discuss the speed adaptation task and thus only the *speed adaptation model*, a broader work with explanation of other sub-modules has been published in [Fiore 2015].

We argue that one aspect for being socially acceptable is the robot able to adapt its speed to the group it is guiding. By setting its own pace at the start of the scenario the robot would risk of being too slow, annoying the users, or too fast, which would lead the robot to constantly stop to wait for the group, producing an awkward behavior. The robot defines a desired range of distance r from the group. The distance of group members from the robot influence its actions in the way that: 1) If there is a member of the group farther than r the robot will *decelerate*, giving priority to people who like a slower speed. 2) If that is not the case and majority of people in the group are closer to the robot than r, the robot will *decelerate*. 3) And finally, if both of the above cases are false, the robot will continue at its current pace. In this work, r was a predefined vector of numbers, but its values could be learnt and adapted to the users during the task, since different people could prefer following the robot at different distances and positions. The robot should also not constantly change speed, in order to give time to users to adapt to its new chosen speed, and so we defined a temporal threshold in which we don't allow the robot to repeat an *accelerate* or *decelerate* action.

In this scenario we also studied the idea that the robot can try to influence the speed of the group. We studied two situations in which this idea is useful. A) There is a time limit to reach the destination. In this case the robot must balance the desire to satisfy the group with the task urgency. For example, in an airport scenario, the robot could prioritize arriving on time, warning users if their speed would render the goal not achievable, while in other situations the robot could try to arrive in time but still avoid to adopt speeds that are uncomfortable for the group. B) The rules of the current environment limit the robot's speed. In this case the robot will avoid accelerating over a set speed even if it detects that its current velocity is too slow for the group. For example, the robot could be navigating in a construction zone.

This reasoning happens inside the *speed adaptation* MOMDP module, which activates when the higher *guide model* chooses to keep guiding the group.

Motion Planning

A guiding robot need to plan safe and socially acceptable trajectories. This requires continual integration of high-level social constraints with the low-level constraints of the robot vehicle.

We use the architecture discussed in Section 3.1 for navigation, replacing the global planner as suggested in [Sisbot 2007] to have proxemics based costs in the grid-map around the detected humans who are static in the environment. The local planner, the module responsible for generating motor commands, is the directional cost based algorithm suggested in [Kruse 2014] as this module continuously calculates the *compatibility* of the robot path by predicting and avoiding future collisions with moving persons and simultaneously keeping the robot as close as possible on the planned global path.

In our guiding experiments, the humans are mostly moving behind the robot and therefore the situation remains compatible for the local planner most of the time. During compatible situations the robot simply follows the way-points on the planned global path. The nominal velocity of the robot is set by the supervision system to achieve the desired behavior of slowing-down or speeding-up, as required by the situation.

Test Name	$\bar{d}$	Δv	$Var(\bar{d})$	$Var(\Delta v)$
Adapting Slow	2.82	-0.03	0.64	0.02
Adapting Fast	1.38	0.00	0.29	0.01
No Adaptation	3.08	-0.09	1.04	0.07
Proactive Slow	1.45	-0.06	0.04	0.10
Proactive Fast	2.66	-0.11	0.63	0.01

Table 6.1 – Experiment results: average distance and average speed difference between the robot and the user as well as variance in average distance and average speed difference. Distances are in meters, velocities are in meters/second.

6.2.2 Experiments

We first performed a set of experiments with a single user following a robot on a predefined path, in order to test the behaviors of the robot. After that, we tested the system by having a group of three users follow the robot. In this section, we show the results of our experiments on single user data, since they show in a clearer way the behavior of the system. The experiments included following cases:

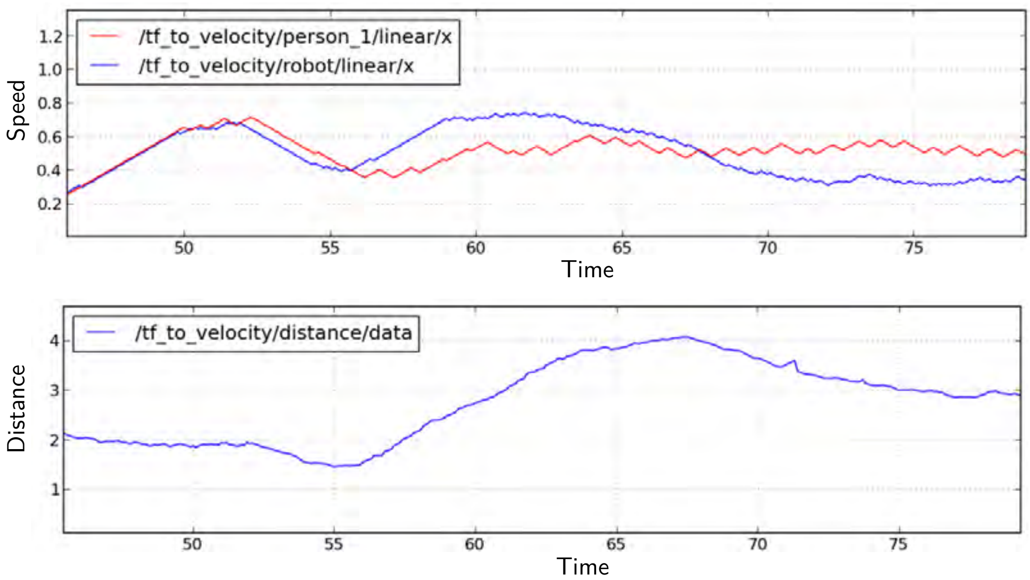
- Adapting Slow or Fast: In these two tests we used our system to reactively guide a user who liked to move at a slow pace, and a user who liked to move at a faster speed.
- No Adaptation: In these experiments the robot won't adapt to the speed of the user, setting its own pace and instantly stopping if it is too far.
- Proactive Slow and Fast: During this task, the robot proactively chooses to change pace, in the first case by slowing down and in the second by accelerating.

Table 6.1 shows results of these experiments. Looking at the data we can see that our system shows lower values for speed and distance variance, which means that after a certain time it is able to establish equilibrium with the human follower. The No Adaptation system shows a significantly higher variance for both values, since the robot stopped several times to wait for a user. In proactive behavior tests the user adapted after some seconds to the robot's pace, but this behavior needs further in-depth user studies. Figure 6.6 shows example plots of the speed of the robot and the human as well as distance between them when robot is adapting to the speed of human.

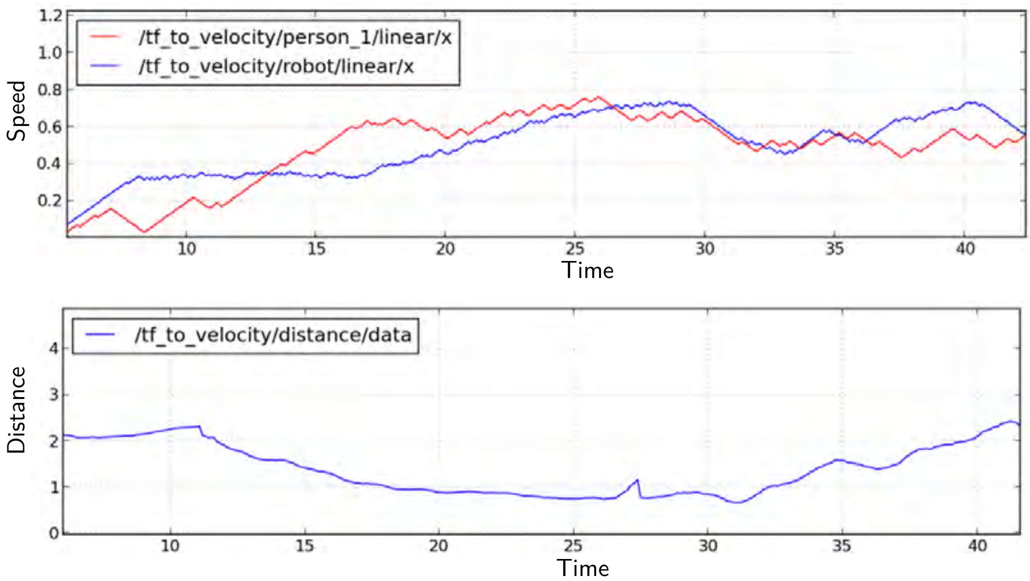
6.3 Summary

In this chapter we presented our collaborative work that uses the human-aware planning systems introduced in Chapter 4 for constructing different normative navigation behaviors.

First, we presented two path planning strategies to approach a person. One that uses an IRL algorithm to directly learn the social-approach-paths which require



(a) Adapting robot speed to a slow user. The upper figure shows the speed of the user (red) and of the robot (blue), and the lower figure shows distance between them. The robot starts slowing down at $Time = 60$, when its distance from the user is growing, until it finds an equilibrium with the user's speed. Notice that there is a turn in the path, at $Time = 50$, that causes the robot and the user to slow down.



(b) Adapting robot speed to a user walking faster. As in above figure a, this figures show the robot and user's speed (above) and their distance (below). The robot starts accelerating at $Time = 15$ when its distance from the user becomes small.

Figure 6.6 – Speed adaptation experiments. Distances are in m, velocities are in m/s, and time in seconds.

smoothing before executing on the robot. In this strategy we quickly recalculate the paths as the person moves. Computation time for planning depends on resolution of discretized state space. Another strategy produces a costmap layer based of the IRL result using RBF state space. Advantage of this strategy is that we can use any path planning techniques on the generated costmap and even mix this costmap with other social cost representations. An important feature of this work is that in both methods also selects the final position to go for finishing the approach behavior.

Later, we introduced a robotic system that is able to guide groups of users to a destination in a human-aware manner. This system is able to estimate if human users are currently engaged in the task and it is able adapt its actions to their behaviors as well as proactively help them. Through a set of experiments we showed that the robot is able to adapt its speed to its followers which is considered a socially acceptable behavior. We also began to study how the system can influence its users, by proposing a new speed, based on environmental and task related stimulus.

CHAPTER 7

Conclusions and Outlook

Contents

Human-aware robot navigation in shared spaces poses new challenges at every link of the robot navigation toolchain. Starting from perception and prediction of human motion to social signal processing and situation assessment up to trajectory and whole body motion planning. New solutions are needed at all levels for transforming a mobile robotic platform into a living and socially-aware individual. It even changes the very definition of what used to known as optimal navigation behavior in terms of shortest and fastest path to goal avoiding environmental obstacles. While moving among humans the desired robot behavior is the one that is acceptable, legible and desirable by the human beings. This shifts the optimality criteria to social measures like human comfort in the presence of a robot, cooperativeness shown by the robot behavior, legibility and predictability of the robot demeanor. The success of social robots is eventually evaluated through responses and ratings of people who share space with them and experience their conduct in everyday situations.

7.1 Conclusions

In Chapter 3 we have presented a reactive planning framework with all necessary components for human-aware navigation. Our fundamental insight is that while both path-planning and obstacle-avoidance modules should integrate human-awareness, it is better to tackle static people during path-planning by respecting social conventions such as proxemics, group formations and activity areas, and handle reaction to moving persons during obstacle avoidance phase. Nevertheless, obstacle avoidance has to differentiate between human and ordinary non-human obstacles to maximize the human comfort, follow the social norms and improve the legibility of the robot behavior.

As a first step towards this framework we ported and integrated the *safety* and the *visibility* criteria [Sisbot 2007] into the occupancy grid-map used by the path-planning algorithm as social costs. Introduction of these social costs already provide safe and comfortable paths around stationary persons in the robot environment. To cope up with dynamic people, we introduced a speed adaptation algorithm based on the *directional cost model*. This model continuously evaluates human and robot path compatibility based on their relative position and velocity. Robot motions which lead the robot on a collision path with the human are treated incompatible. In such situations, the robot reduces its speed without changing its direction until the situation turns compatible again. This strategy proves useful and humans prefer such robot behavior for path crossing situations. According to the user-study we presented in Section 3.2 with 17 participants and the PR2 robot, the *directional cost model* based speed adaptation strategy is both more comfortable and legible compared to the case where robot continuously tries to change its path and thus its direction at while moving at possible maximum speed.

Following the results of the user-study we integrated the *direction cost model* with the well-known dynamic window approach for obstacle avoidance. This integration gave us a local-planning module that can not only efficiently avoid any non-human obstacles, but also treat a human as a “special” case of dynamic obstacle by employing the speed adaption behavior whenever the robot encounters crossing its path with a person. We have deployed and successfully tested this local-planner on the Spencer robot during the experiments in the Amsterdam airport Schiphol.

During the course of the SPENCER project we further investigated two socially normative robot behaviors for human-aware navigation: approaching a person to initiate a conversation and adapting the robot behavior while guiding a group of people such as passengers at an airport. Specifically, we used inverse reinforcement learning technique to determine the robot goal position for approaching a human as well as to generate a costmap that can lead the robot to this approach position while executing human-like trajectories. With this method robot proffered to approach the person from the front and avoided moving too near to the person when coming from their back, which is the case how humans usually approach other humans and evident from the data we recorded for learning.

We also integrated the *directional cost* based planner with a supervision module for guiding a group of people in large indoor environments like airports or shopping malls. While guiding, the robot continuously monitors whether the group is still following the robot and modifies its own speed that suit the preferred speed of the group. This means, the robot can slow down when the group is legging behind or speed up if groups is trying to “push” the robot to move faster by moving very closely behind the robot. Furthermore, the robot can proactively encourage the group to hurry-up if the task as hand has external time constraints, such as the group need to reach a gate at the airport before the gate closes. Chapter 6 gives implementation details of both approaching and guiding behaviors.

The SPENCER project deliberately opted for a passive pan and tilt head while designing the robot. By passive we mean that the head did not include any sensors for perception. The purpose of a pan-tilt head with horizontally moving eyes was to use it for human-robot interactions, not only while taking with person or a group but also during navigation. With this goal in mind, we looked into the human psychology research to understand how humans use their head and gaze during locomotion. This lead us to develop a multi-criteria decision-making based method to control the robot gazing behavior during navigation. Specifically, our robot can choose different gazing behaviors depending on the environmental context and can select a gazing-point from a set of candidate points to induce the desired behavior. In Chapter 5 we presented this gaze control method and an online video-based user study that evaluates the effectiveness of the method. The results of the user-study with 126 participants suggest that they preferred the robot behavior where the robot looks at its path while it is moving and occasionally glances at nearby humans to indicate the robot has seen them and it is going to avoid any possible collision with them.

One of the lessons we learned from the experiments with the Spencer robot in real, semi-crowded environment like airports is that although the *direction cost model* is quite effective in making the robot behavior more legible, the robot sometimes also need to change its path for making its behavior more fluent. A social robot has to balance between adapting its speed and adapting its path. This balance is even more important in confined spaces such as long and narrow corridors or bottlenecks. In constrained cases a social robot should behave in a manner that is both safe and comfortable for the humans and avoid moving into entangled configurations. In other words, we view the human-robot co-navigation situation as a joint-task where both human and robot need to cooperate and help each-other to efficiently reach their navigational goals.

In Chapter 4, we introduced a robot navigation planner that can provide solution to human-robot co-navigation problems. This cooperative planner represents the planned robot trajectory and predicted human trajectories as virtual elastic bands and uses least-squares optimization that "deform" these elastic bands such that both the robot and the human can safely and comfortably move and avoid colliding with each other in a shared space. At the same time, the planner makes sure that both the human and the robot are also able to avoid any other static and dynamic obstacles present in the environment.

With the cooperative planner the robot shows human friendly behaviors due to the social constraints of *safety*, *time-to-collision* and *directional-costs* that we incorporated within the optimization framework. These social constraints also allows the robot to show proactive behaviors such as moving well in advance on a side of a corridor or waiting and allowing human to pass before in bottleneck situations. While displaying proactive behaviors the robot does not compromise reactivity towards unfolding situation. In case if the human does not move on the trajectory that the robot had predicted, it can quickly re-evaluate the situation and find new solution to comply with human's wish.

The cooperative planner enables us to plan for the human-robot joint action of navigating in a shared space, where we can tune and balance the effort required by both the human and the robot. Naturally, during experiments we let the robot to take most of the load and give the human maximum comfort. We demonstrated the strength and versatility of the proposed cooperative planner in simulation as well as real world human-robot crossing situations. We have also compared the cooperative planner with other state-of-the-art human-aware navigation planners in five canonical path crossing situations in simulation. The results show the cooperative planner outperforming especially in constrained situations. This planner is quite suitable for intricate navigation scenarios such as in shopping malls or museums involving single person or a group of people. Developing such sophisticated interactions is also one of the objectives of the MuMMER project.

7.2 Summary of Contributions

This thesis is a step towards injecting human-awareness into tomorrows mobile-service-robots like Spencer and Pepper. The main scientific contributions of this thesis are:

- An adaptive and cooperative planner that is able to not only plan robot trajectories that satisfy a set of social constraints but also predict plausible human trajectories in the same environment that robot perceives through its sensors. In turn it uses the predicted human trajectories to modify its own path that cause minimal deviation human paths thus ensuring human comfort and safety. It works particularly well in demanding situation where several human-like behavior emerges such as robot moving away from its path to give the human way at doorways, or the robot taking an initiative to offer a solution the co-navigation problem in confined spaces.

- The cooperative planner is compares favorably in canonical path crossing situations against other state-of-the-art human-aware planning schemes in our preliminary experiments. This results are reassuring for the approach we take to solve social navigation problem, however, it required further user studies to demonstrate its full potential.

- We identified the problem of “where the robot should look?” while navigating among humans as multi criteria decision-making problem and proposed a solution using atomic behavioral functions that are activated based on the social context the robot finds itself in at given moment during navigation. With this approach the robot performs human-like saccade behaviors and indicates it future path with its gaze, such behavior is agreeable for people in robot surroundings.

- The cooperative planner also makes use of directional constraints, effectiveness of which is confirmed by a user study with uninstructed participants in a real work path crossing scenario.

- We have also proposed methods for developing normative social behaviors for approaching a person to initiate an interaction and adapting speed of a robot guide depending on users will or other environmental conditions.

7.3 Outlook

We have made significant strides towards our vision of an autonomous, intelligent and social service-robot that we contemplated in Chapter 1. Though we are not done just yet! Here we list down some of the interesting future research directions that builds upon the current work of this thesis.

Explore novel and pertinent social constraints: the cooperative planning approach with optimal planning mechanism at its base opens up new possibilities for quickly testing and comparing different social constraints. We plan to define further novel social constraint that will enable further interactive motion scenarios such as actively approaching a group of people and joining the group, queuing behavior in long hallway like environments with multiple humans, carrying out hand and body gestures while navigating, and more.

Integrate additional modalities with navigation: we have identified that only after augmenting navigation algorithms with other modalities such as gaze, gesture, speech, a robot can truly become *socially intelligent* and *socially acceptable*. Building on our work on head and body coordination (Chapter 5), we are already exploring and testing timely gestures such as pointing to a landmark or addressing particular person of a group, for a robot like Pepper while it navigates among people. We plan to explore other gaze behaviors like robot acknowledging a group of people, showing certain emotions (surprise, anticipation, trust), and functions that explicitly take affordances of the environment into account. A machine learning based approach for dynamically adapting the criteria weight vector coupled with higher level perception modules is also and exiting avenue for the future work.

Testing usability in real-world situations: We are preparing for a series of real-world experiments and a user study which compares the cooperative planner against other proactive planning approaches. Our aim will be to evaluate effect of the robot behaviors resulting from the use different social constraints in different situations. In the online user study for head behavior, we did not include any *placebo* behavior (for example robot looking at random points), testing such behavior against the proposed approach, and that with a first-person user study is also what we are planning next.

Exploit the cooperative planner for creating normative behaviors: the cooperative planner makes it easy to directly generate kinodynamic trajectories. We would like to integrate the cooperative planner with the approaching behavior that we discussed in Section 6.1. Figure 7.1 shows the first step where we manually compute the goal position for the robot to be directly in front of the human and use cooperative planner to calculate a trajectory to reach that goal. Instead, we intend use the goal positions computed by the IRL based algorithm while keep relying on the cooperative planner to generate human-aware trajectories. Similarly, we aim to integrate the cooperative planner with a supervision system by sharing rich information like planned time to execute the full trajectory, planned velocities at certain point during

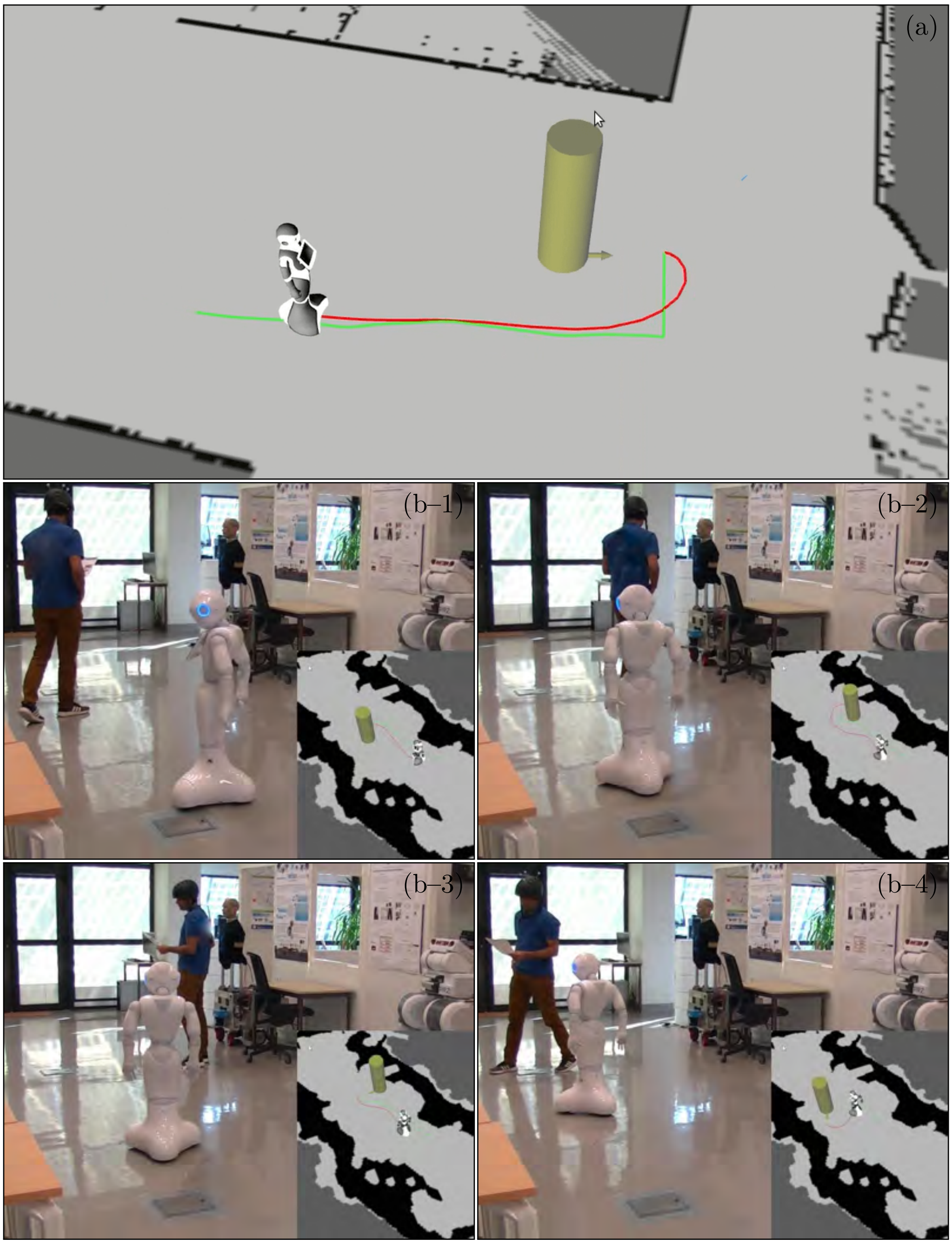


Figure 7.1 – Approaching behavior where we manually set the robot goal at specific distance in front of the human and use the cooperative planner (Chapter 4) to generate human-aware trajectories to reach this goal. Figure (a) shows a sample trajectory for approaching a person in a simulated environment. Figures (b) to (e) show a sequence of trajectories generated by a real Pepper robot to approach a human while the human is continuously moving. We can see that the robot is able to dynamically adapt its trajectories such that it can always approach the human from front.

the trajectory or the time robot planned to wait for human to pass, for example, through a door.

Learning for social navigation: the cooperative planner has several parameters that directly affect the quality of the generated trajectories. Different simulation demand different values for this parameters to display best robot behavior. Manually setting the values for this parameter may not be the most effective way. Supervised machine learning can help in determining parameter values that suit the situation under consideration.

Bibliography

[Admoni 2014] Henny Admoni, Anca Dragan, Siddhartha S. Srinivasa, and Brian Scassellati. “Deliberate Delays During Robot-to-Human Handovers Improve Compliance with Gaze Communication”. In: *Proc. 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2014, pp. 49–56. ISBN: 978-1-4503-2658-2 (Cited on page 93).

[Al Moubayed 2012] Samer Al Moubayed and Gabriel Skantze. “Perception of Gaze Direction for Situated Interaction”. In: *Proc. 4th Workshop on Eye Gaze in Intelligent Human Machine Interaction Held at ICMI*. 2012, 3:1–3:6. ISBN: 978-1-4503-1516-6 (Cited on page 93).

[Alili 2009] Samir Alili, Rachid Alami, and Vincent Montreuil. “A Task Planner for an Autonomous Social Robot”. In: *Distributed Autonomous Robotic Systems 8*. Ed. by Hajime Asama, Haruhisa Kurokawa, Jun Ota, and Kosuke Sekiyama. 2009, pp. 335–344. ISBN: 978-3-642-00644-9 (Cited on page 23).

[Amaya 1996] Kenji Amaya, Armin Bruderlin, and Tom Calvert. “Emotion from Motion”. In: *Proc. Graphics Interface*. May 1996, pp. 222–229. ISBN: 0-9695338-5-3 (Cited on page 93).

[Avrunin 2014] Eleanor R. Avrunin and Reid Simmons. “Socially-Appropriate Approach Paths Using Human Data”. In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. Aug. 25–29, 2014, pp. 1037–1042 (Cited on page 116).

[Barliya 2013] Avi Barliya, Lars Omlor, Martin A. Giese, Alain Berthoz, and Tamar Flash. “Expression of Emotion in the Kinematics of Locomotion”. In: *Experimental Brain Research* 225.2 (2013), pp. 159–176. ISSN: 1432-1106 (Cited on page 93).

[Basili 2013] Patrizia Basili, Murat Sağlam, Thibault Kruse, Markus Huber, Alexandra Kirsch, and Stefan Glasauer. “Strategies of Locomotor Collision Avoidance”. In: *Gait Posture* 37.3 (2013), pp. 385–390. ISSN: 0966-6362 (Cited on page 40).

[Berg 2009] Jur van den Berg, Stephen J. Guy, Ming C. Lin, and Dinesh Manocha. “Reciprocal N-Body Collision Avoidance”. In: *International Symposium on Robotics Research (ISRR)*. ISRR. Aug. 31–Sept. 3, 2009, pp. 3–19 (Cited on page 16).

[Bernardin 2012] Delphine Bernardin, Hideki Kadone, Daniel Bennequin, Thomas Sugar, Mohamed Zaoui, and Alain Berthoz. “Gaze Anticipation during Human Locomotion”. In: *Experimental Brain Research* 223.1 (2012), pp. 65–78. ISSN: 0014-4819 (Cited on page 95).

[Bohannon 1997] Richard W. Bohannon. “Comfortable and Maximum Walking Speed of Adults Aged 20—79 Years: Reference Values and Determinants”. In: *Age and Ageing* 26.1 (Jan. 1, 1997), pp. 15–19. ISSN: 0002-0729 (Cited on page 60).

[Bordallo 2015] Alejandro Bordallo, Fabio Previtali, Nantas Nardelli, and Subramanian Ramamoorthy. “Counterfactual Reasoning about Intent for Interactive Navigation in Dynamic Environments”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 28–Oct. 2, 2015, pp. 2943–2950 (Cited on page 22).

[Bratman 1993] Michael E. Bratman. “Shared Intention”. In: *Ethics* 104.1 (1993), pp. 97–113. ISSN: 00141704, 1539297X. JSTOR: 2381695 (Cited on page 53).

[Breazeal 2005] Cynthia Breazeal, Cory D. Kidd, Andrea Lockerd Thomaz, Guy Hoffman, and Matt Berlin. “Effects of Nonverbal Communication on Efficiency and Robustness in Human-Robot Teamwork”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Aug. 2005, pp. 708–713 (Cited on page 93).

[Burgard 1998] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. “The Interactive Museum Tour-Guide Robot”. In: *Proc. Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*. 1998, pp. 11–18. ISBN: 0-262-51098-7 (Cited on page 121).

[Cannon 2012] Jarad Cannon, Kevin Rose, and Wheeler Ruml. “Real-Time Motion Planning with Dynamic Obstacles”. In: *Proc. Fifth Annual Symposium on Combinatorial Search (SOCS)*. July 2012 (Cited on page 12).

[Castiello 2003] Umberto Castiello. “Understanding Other People’s Actions: Intention and Attention”. In: *Journal of Experimental Psychology: Human Perception and Performance* 29.2 (Apr. 2003), pp. 416–430 (Cited on pages 18 and 92).

[Clark 1983] Herbert H. Clark, Robert Schreuder, and Samuel Buttrick. “Common Ground at the Understanding of Demonstrative Reference”. In: *Journal of Verbal Learning and Verbal Behavior* 22.2 (1983), pp. 245–258. ISSN: 0022-5371 (Cited on page 53).

[Clodic 2006] Aurélie Clodic, Sara Fleury, Rachid Alami, Raja Chatila, Gérard Bailly, Ludovic Brethes, Maxime Cottret, Patrick Danes, Xavier Dollat, Frédéric Elisei, Isabelle Ferrane, Matthieu Herrb, Guillaume Infantes, Christian Lemaire, Frédéric Lerasle, Jérôme Manhes, Patrick Marcoul, Paulo Menezes, and Vincent Montreuil. “Rackham: An Interactive Robot-Guide”. In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. Sept. 2006, pp. 502–509 (Cited on page 121).

[Clodic 2014] Aurélie Clodic, Rachid Alami, and Raja Chatila. "Key Elements for Human-Robot Joint Action". In: *Sociable Robots and the Future of Social Relations*. Vol. 273. Aug. 2014, pp. 23–33 (Cited on page 53).

[Coovert 2014] Michael D. Coovert, Tiffany Lee, Ivan Shindev, and Yu Sun. "Spatial Augmented Reality as a Method for a Mobile Robot to Communicate Intended Movement". In: *Computers in Human Behavior* 34 (2014), pp. 241–248. ISSN: 0747-5632 (Cited on page 97).

[Davis 2006] Timothy Davis. *Direct Methods for Sparse Linear Systems*. 2006 (Cited on page 64).

[Dijkstra 1959] E. W. Dijkstra. "A Note on Two Problems in Connexion with Graphs". In: *Numerische Mathematik* 1.1 (Dec. 1959), pp. 269–271. ISSN: 0029-599X (Cited on pages 12, 78 and 120).

[Donald 1993] Bruce Randall Donald, Patrick G. Xavier, John F. Canny, and John H. Reif. "Kinodynamic Motion Planning". In: *Journal of the ACM* 40.5 (1993), pp. 1048–1066 (Cited on page 13).

[Dragan 2013a] Anca D. Dragan, Kenton C. T. Lee, and Siddhartha S. Srinivasa. "Legibility and Predictability of Robot Motion". In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Mar. 3–6, 2013, pp. 301–308 (Cited on pages 5 and 26).

[Dragan 2013b] Anca Dragan and Siddhartha Srinivasa. "Generating Legible Motion". In: *Proc. Robotics: Science and Systems*. June 24–28, 2013 (Cited on page 26).

[Dragan 2015] Anca D. Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S. Srinivasa. "Effects of Robot Motion on Human-Robot Collaboration". In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. International Conference on Human-Robot Interaction. Mar. 2015, pp. 51–58. ISBN: 978-1-4503-2883-8 (Cited on page 94).

[Evers 2014] Vanessa Evers, Nuno Menezes, Luis Merino, Dariu Gavrila, Fernando Nabais, Maja Pantic, and Paulo Alvito. "The Development and Real-World Application of FROG, the Fun Robotic Outdoor Guide". In: *Proc. Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*. 2014, pp. 281–284. ISBN: 978-1-4503-2541-7 (Cited on page 121).

[Ferrer 2013] Gonzalo Ferrer, Anaís Garrell, and Alberto Sanfeliu. "Robot Companion: A Social-Force Based Approach with Human Awareness-Navigation in Crowded Environments". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 1688–1694 (Cited on pages 21 and 79).

[Ferrer 2014a] Gonzalo Ferrer and Alberto Sanfeliu. "Behavior Estimation for a Complete Framework for Human Motion Prediction in Crowded Environments". In: *Robotics and Automation (ICRA), 2014 IEEE International Conference On*. May 2014, pp. 5940–5945 (Cited on page 32).

[Ferrer 2014b] Gonzalo Ferrer and Alberto Sanfeliu. "Proactive Kinodynamic Planning Using the Extended Social Force Model and Human Motion Prediction in Urban Environments". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 18, 2014, pp. 1730–1735 (Cited on pages 21, 52, 54 and 79).

[Ferrer 2015] Gonzalo Ferrer and Alberto Sanfeliu. "Multi-Objective Cost-to-Go Functions on Robot Navigation in Dynamic Environments". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IROS. 2015, pp. 3824–3829 (Cited on pages 21 and 79).

[Fiore 2013] Stephen M. Fiore, Travis J Wiltshire, Emilio J. C. Lobato, Florian G. Jentsch, Wesley H. Huang, and Benjamin Axelrod. "Toward Understanding Social Cues and Signals in Human-Robot Interaction: Effects of Robot Gaze and Proxemic Behavior". In: *Frontiers in Psychology* 4.859 (2013). ISSN: 1664-1078 (Cited on page 94).

[Fiore 2015] Michelangelo Fiore, Harmish Khambhaita, Grégoire Milliez, and Rachid Alami. "An Adaptive and Proactive Human-Aware Robot Guide". In: *Proc. Seventh International Conference on Social Robotics (ICSR)*. Oct. 26–30, 2015, pp. 194–203 (Cited on page 122).

[Fiorini 1998] Paolo Fiorini and Zvi Shiller. "Motion Planning in Dynamic Environments Using Velocity Obstacles". In: *The International Journal of Robotics Research* 17.7 (1998), pp. 760–772 (Cited on pages 32, 70 and 79).

[Foster 2016] Mary Ellen Foster, Rachid Alami, Olli Gestranius, Oliver Lemon, Marketta Niemelä, Jean-Marc Odobez, and Amit Kumar Pandey. "The MuMMER Project: Engaging Human-Robot Interaction in Real-World Public Spaces". In: *Proc. 8th International Conference on Social Robotics (ICSR)*. Ed. by Arvin Agah, John-John Cabibihan, Ayanna M. Howard, Miguel A. Salichs, and Hongsheng He. 2016, pp. 753–763. ISBN: 978-3-319-47437-3 (Cited on page 77).

[Fox 1997] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. "The Dynamic Window Approach to Collision Avoidance". In: *IEEE Robotics and Automation Magazine* 4.1 (1997), pp. 23–33. ISSN: 10709932 (Cited on pages 13, 15 and 48).

[Garzón 2013] Mario Garzón, Efstathios P. Fotiadis, Antonio Barrientos, and Anne Spalanzani. "RiskRRT-Based Planning For Interception of Moving Objects in Complex Environments". In: *Proc. First Iberian Robotics Conference (ROBOT)*. Nov. 2013, pp. 489–503 (Cited on page 23).

[Gerkey 2008] Brian P. Gerkey and Kurt Konolige. “Planning and Control in Unstructured Terrain”. In: *Proc. Workshop on Path Planning on Costmaps at the IEEE International Conference on Robotics and Automation (ICRA)*. 2008 (Cited on page 15).

[Gharbi 2015] Mamoun Gharbi, Pierre-Vincent Paubel, Aurélie Clodic, Ophelie Carreras, Rachid Alami, and Jean-Marie Cellier. “Toward a Better Understanding of the Communication Cues Involved in a Human-Robot Object Transfer”. In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2015, pp. 319–324 (Cited on pages 93 and 95).

[Gopalakrishnan 2014] Bharath Gopalakrishnan, Arun Kumar Singh, and K. Madhava Krishna. “Time Scaled Collision Cone Based Trajectory Optimization Approach for Reactive Planning in Dynamic Environments”. In: *International Conference on Intelligent Robots and Systems (IROS)*. IROS. Sept. 14–18, 2014, pp. 4169–4176 (Cited on page 16).

[Hall 1966] E. T. Hall. *The Hidden Dimension: Man’s Use of Space in Public and Private*. 1966 (Cited on pages 17, 19 and 20).

[Hanheide 2012] Marc Hanheide, Annika Peters, and Nicola Bellotto. “Analysis of Human-Robot Spatial Behaviour Applying a Qualitative Trajectory Calculus”. In: *The 21st IEEE International Symposium on Robot and Human Interactive Communication, IEEE RO-MAN 2012, Paris, France, September 9-13, 2012*. 2012, pp. 689–694 (Cited on page 18).

[Harms 2004] Chad Harms and Frank Biocca. “Internal Consistency and Reliability of the Networked Minds Measure of Social Presence”. In: *Proc. 7th Annual International Workshop on Presence*. Ed. by Mariano Alcaniz and Beatriz Rey. 2004 (Cited on page 110).

[Hart 1968] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (July 1968), pp. 100–107. ISSN: 0536-1567 (Cited on pages 12 and 19).

[Hart 2014] Justin W. Hart, Brian Gleeson, Matthew Pan, AJung Moon, Karon MacLean, and Elizabeth Croft. “Gesture, Gaze, Touch, and Hesitation: Timing Cues for Collaborative Work”. In: *Proc. "Timing in Human-Robot Interaction" Workshop Held at HRI*. 2014 (Cited on page 95).

[Hebesberger 2016] Denise Hebesberger, Christian Dondrup, Tobias Körtner, Christoph Gisinger, and Jürgen Pripfl. “Lessons Learned from the Deployment of a Long-Term Autonomous Robot as Companion in Physical Therapy for Older Adults with Dementia - A Mixed Methods Study”. In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2016 (Cited on page 77).

[Helbing 1995] Dirk Helbing and Péter Molnár. "Social Force Model for Pedestrian Dynamics". In: *Physical Review E* 51.5 (May 1995), pp. 4282–4286 (Cited on pages 18, 19, 21 and 79).

[Helbing 2011] Dirk Helbing and Anders Johansson. "Pedestrian, Crowd and Evacuation Dynamics". In: *Extreme Environmental Events*. Ed. by Robert A. Meyers. 2011, pp. 697–716. ISBN: 978-1-4419-7694-9 (Cited on page 18).

[Hicheur 2005] Halim Hicheur, Stéphane Vieilledent, and Alain Berthoz. "Head Motion in Humans Alternating between Straight and Curved Walking Path: Combination of Stabilizing and Anticipatory Orienting Mechanisms". In: *Neuroscience Letters* 383 (1–2 2005), pp. 87–92. ISSN: 0304-3940. (Visited on 07/29/2005) (Cited on page 95).

[Imai 2002] Michita Imai, Takayuki Kanda, Testuo Ono, Hiroshi Ishiguro, and Kenji Mase. "Robot Mediated Round Table: Analysis of the Effect of Robot's Gaze". In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2002, pp. 411–416 (Cited on page 95).

[Inderbitzin 2011] Martin Inderbitzin, Aleksander Väljamäe, Jose Maria Blanco Calvo, Paul F.M.J. Verschure, and Ulysses Bernardet. "Expression of Emotional States during Locomotion Based on Canonical Parameters". In: *Proc. 1st International Workshop on Emotion Synthesis, rePresentation, and Analysis in Continuous Spaces (EmoSPACE) in Conjunction with the 9th IEEE Automatic Face & Gesture Recognition Conference*. 2011, pp. 809–814. ISBN: 978-1-4244-9140-7 (Cited on page 93).

[Islas Ramírez 2016] Omar A. Islas Ramírez, Harmish Khambhaita, Raja Chatila, Mohamed Chetouani, and Rachid Alami. "Robots Learning How and Where to Approach People". In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. Aug. 2016, pp. 347–353 (Cited on page 120).

[ISO 2012] ISO/TC 184 ISO. *Robots and Robotic Devices — Vocabulary*. ISO 8373:2012. 2012 (Cited on page 3).

[Joosse 2013] Michiel Joosse, Aziez Sardar, Manja Lohse, and Vanessa Evers. "BEHAVE-II: The Revised Set of Measures to Assess Users' Attitudinal and Behavioral Responses to a Social Robot". In: *International Journal of Social Robotics* 5.3 (2013), pp. 379–388 (Cited on page 27).

[Karaman 2011] Sertac Karaman and Emilio Frazzoli. "Sampling-Based Algorithms for Optimal Motion Planning". In: *International Journal of Robotics Research* 30.7 (2011), pp. 846–894 (Cited on page 13).

[Karamouzas 2014] Ioannis Karamouzas, Brian Skinner, and Stephen J. Guy. "Universal Power Law Governing Pedestrian Interactions". In: *Physical Review Letters* 113.23 (Dec. 2014), p. 238701 (Cited on pages 18, 67 and 87).

[Kato 2015] Yusuke Kato, Takayuki Kanda, and Hiroshi Ishiguro. "May I Help You? - Design of Human-like Polite Approaching Behavior". In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2015, pp. 35–42. ISBN: 978-1-4503-2883-8 (Cited on page 116).

[Kavraki 1996] Lydia E. Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark H. Overmars. "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces". In: *IEEE Transactions on Robotics and Automation* 12.4 (Aug. 1996), pp. 566–580. ISSN: 1042-296X (Cited on page 12).

[Kendon 1990] Adam Kendon. *Conducting Interaction: Patterns of Behavior in Focused Encounters*. 1990 (Cited on pages 17 and 19).

[Khambhaita 2016] Harmish Khambhaita, Jorge Rios-Martinez, and Rachid Alami. "Head-Body Motion Coordination for Human Aware Robot Navigation". In: *Proc. International Workshop on Human-Friendlly Robotics (HFR)*. HFR. Sept. 29–30, 2016 (Cited on page 92).

[Khambhaita 2017a] Harmish Khambhaita and Rachid Alami. "A Human-Robot Cooperative Navigation Planner". In: *Proc. Companion of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Mar. 6–9, 2017, pp. 161–162. ISBN: 978-1-4503-4885-0 (Cited on page 52).

[Khambhaita 2017b] Harmish Khambhaita and Rachid Alami. "Assessing the Social Criteria for Human-Robot Collaborative Navigation: A Comparison of Human-Aware Navigation Planners". In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. Aug. 28–Sept. 1, 2017 (Cited on page 77).

[Khambhaita 2017c] Harmish Khambhaita and Rachid Alami. "Viewing Robot Navigation in Human Environment as a Cooperative Activity". In: *Proc. International Symposium on Robotics Research (ISRR)*. Dec. 11–14, 2017 (Cited on page 52).

[Khatib 1997] Maher Khatib, H. Jaouni, Raja Chatila, and Jean-Paul Laumond. "Dynamic Path Modification for Car-like Nonholonomic Mobile Robots". In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. Apr. 20–25, 1997, pp. 2920–2925 (Cited on pages 14 and 58).

[Kim 2016] Beomjoon Kim and Joelle Pineau. "Socially Adaptive Path Planning in Human Environments Using Inverse Reinforcement Learning". In: *International Journal of Social Robotics* 8.1 (2016), pp. 51–66. ISSN: 1875-4805 (Cited on page 25).

[Kitazawa 2010] Kay Kitazawa and Taku Fujiyama. "Pedestrian Vision and Collision Avoidance Behavior: Investigation of the Information Process Space of Pedestrians Using an Eye Tracker". In: *Pedestrian and Evacuation Dynamics 2008*. Ed. by Wolfram W. F. Klingsch, Christian Rogsch, Andreas Schadschneider, and Michael Schreckenberg. 2010, pp. 95–108. ISBN: 978-3-642-04503-5 (Cited on pages 18, 92, 95 and 100).

[Knoblich 2011] Günther Knoblich, Stephen Butterfill, and Natalie Sebanz. "Psychological Research on Joint Action". In: *Psychology of Learning and Motivation* 54 (2011), pp. 59–101. ISSN: 0079-7421 (Cited on page 53).

[Koenig 2002] Sven Koenig and Maxim Likhachev. "D*Lite". In: *Proc. Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence.* 2002, pp. 476–483 (Cited on page 12).

[Koenig 2009] Sven Koenig and Xiaoxun Sun. "Comparing Real-Time and Incremental Heuristic Search for Real-Time Situated Agents". In: *Autonomous Agents and Multi-Agent Systems* 18.3 (June 2009), pp. 313–341. ISSN: 1387-2532 (Cited on page 12).

[Kretzschmar 2016] Henrik Kretzschmar, Markus Spies, Christoph Sprunk, and Wolfram Burgard. "Socially Compliant Mobile Robot Navigation via Inverse Reinforcement Learning". In: *The International Journal of Robotics Research* (2016) (Cited on page 25).

[Kruse 2012] Thibault Kruse, Patrizia Basili, Stefan Glasauer, and Alexandra Kirsch. "Legible Robot Navigation in the Proximity of Moving Humans". In: *Proc. IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO).* ARSO. 2012, pp. 83–88 (Cited on pages 26, 36, 38 and 92).

[Kruse 2013] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. "Human-Aware Robot Navigation: A Survey". In: *Robotics and Autonomous Systems* 61.12 (2013), pp. 1726–1743. ISSN: 0921-8890 (Cited on pages 19, 20, 52 and 94).

[Kruse 2014] Thibault Kruse, Alexandra Kirsch, Harmish Khambhaita, and Rachid Alami. "Evaluating Directional Cost Models in Navigation". In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI).* HRI. Mar. 3–6, 2014, pp. 350–357 (Cited on pages 5, 13, 26, 39, 52, 88, 92 and 123).

[Kruse 2015] Thibault Kruse. "Planning for Human Robot Interaction". Technische Universität München, Université de Toulouse, 2015 (Cited on pages 18 and 39).

[Kuderer 2012] Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. "Feature-Based Prediction of Trajectories for Socially Compliant Navigation". In: *Proc. Robotics: Science and Systems (RSS).* RSS. July 9–13, 2012 (Cited on pages 21, 24 and 79).

[Kümmerle 2011] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. "G2o: A General Framework for Graph Optimization". In: *Proc. IEEE International Conference on Robotics and Automation (ICRA).* ICRA. May 2011, pp. 3607–3613 (Cited on pages 14 and 61).

[Kushleyev 2009] Aleksander Kushleyev and Maxim Likhachev. "Time-Bounded Lattice for Efficient Planning in Dynamic Environments". In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. May 2009, pp. 1662–1668 (Cited on page 16).

[Lallement 2014] Raphaël Lallement, Lavindra de Silva, and Rachid Alami. "HATP: An HTN Planner for Robotics". In: *CoRR* abs/1405.5345 (2014) (Cited on page 23).

[LaValle 2006] Steven M. LaValle. *Planning Algorithms*. 2006 (Cited on page 12).

[Lemaignan 2014] Séverin Lemaignan, Marc Hanheide, Michael Karg, Harmish Khambhaita, Lars Kunze, Florian Lier, Ingo Lütkebohle, and Grégoire Milliez. "Simulation and HRI Recent Perspectives with the MORSE Simulator". In: *Proc. International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*. Ed. by Davide Brugali, Jan F. Broenink, Torsten Kroeger, and Bruce A. MacDonald. Oct. 20–23, 2014, pp. 13–24. ISBN: 978-3-319-11900-7 (Cited on page 33).

[Lemaignan 2017] Séverin Lemaignan, Mathieu Warnier, E. Akin Sisbot, Aurélie Clodic, and Rachid Alami. "Artificial Cognition for Social Human–robot Interaction: An Implementation". In: *Artificial Intelligence* 247 (2017), pp. 45–69. ISSN: 0004-3702 (Cited on page 53).

[Levine 2010] Sergey Levine, Zoran Popović, and Vladlen Koltun. "Feature Construction for Inverse Reinforcement Learning". In: *Proc. International Conference on Neural Information Processing Systems*. 2010, pp. 1342–1350 (Cited on page 25).

[Levine 2011] Sergey Levine, Zoran Popović, and Vladlen Koltun. "Nonlinear Inverse Reinforcement Learning with Gaussian Processes". In: *Proc. International Conference on Neural Information Processing Systems*. 2011, pp. 19–27 (Cited on page 25).

[Lichtenthaler 2012] Christina Lichtenthaler, Tamara Lorenz, Michael Karg, and Alexandra Kirsch. "Increasing Perceived Value between Human and Robots - Measuring Legibility in Human Aware Navigation". In: *Proc. IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO)*. May 2012, pp. 89–94 (Cited on pages 26 and 91).

[Likhachev 2008] Maxim Likhachev and Anthony Stentz. "R * Search". In: *Proc. 23rd National Conference on Artificial Intelligence (AAAI)*. Vol. 1. 2008, pp. 344–350 (Cited on page 12).

[Lu 2013] David V. Lu, Daniel B. Allan, and William D. Smart. "Tuning Cost Functions for Social Navigation". In: Proc. 5th International Conference on Social Robotics (ICSR). 2013, pp. 442–451 (Cited on page 30).

[Lu 2014a] David V. Lu. "Contextualized Robot Navigation". Washington University in St. Louis, 2014. 150 pp. (Cited on pages 30 and 94).

[Lu 2014b] David V. Lu, Dave Hershberger, and William D. Smart. “Layered Costmaps for Context-Sensitive Navigation”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 14–18, 2014, pp. 709–715 (Cited on pages 27, 30, 70 and 120).

[Mainprice 2012] Jim Mainprice, Mamoun Gharbi, Thierry Siméon, and Rachid Alami. “Sharing Effort in Planning Human-Robot Handover Tasks”. In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2012, pp. 764–770 (Cited on page 23).

[MarderEppstein] Eitan Marder-Eppstein. *`move_base`: A ROS Package That Lets You Move a Robot to Desired Positions Using the Navigation Stack.* In collab. with David V. Lu and Michael Ferguson (Cited on page 12).

[May 2015] Alyxander David May, Christian Dondrup, and Marc Hanheide. “Show Me Your Moves! Conveying Navigation Intention of a Mobile Robot to Humans”. In: *Proc. European Conference on Mobile Robots (ECMR)*. Sept. 2015, pp. 1–6 (Cited on page 94).

[Milliez 2014] Grégoire Milliez, Matthieu Warnier, Aurélie Clodic, and Rachid Alami. “A Framework for Endowing an Interactive Robot with Reasoning Capabilities about Perspective-Taking and Belief Management”. In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2014, pp. 1103–1109 (Cited on page 122).

[Morales 2013] Yoichi Morales, Nagasrikanth Kallakuri, Kazuhiro Shinozawa, Takahiro Miyashita, and Norihiro Hagita. “Human-Comfortable Navigation for an Autonomous Robotic Wheelchair”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 2737–2743 (Cited on page 26).

[Morales 2015] Yoichi Morales, Atsushi Watanabe, Florent Ferreri, Jani Even, Tetsushi Ikeda, Kazuhiro Shinozawa, Takahiro Miyashita, and Norihiro Hagita. “Including Human Factors for Planning Comfortable Paths”. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 6153–6159 (Cited on page 26).

[Müller 2010] Jörg Müller, Cyrill Stachniss, Kai O. Arras, and Wolfram Burgard. “Socially Inspired Motion Planning for Mobile Robots in Populated Environments”. In: *Cognitive Systems*. 2010 (Cited on page 26).

[Mutlu 2006] Bilge Mutlu, Jodi Forlizzi, and Jessica Hodgins. “A Storytelling Robot: Modeling and Evaluation of Human-like Gaze Behavior”. In: *Proc. IEEE-RAS International Conference on Humanoid Robots*. Dec. 2006, pp. 518–523 (Cited on page 93).

[Ng 2000] Andrew Y. Ng and Stuart Russell. “Algorithms for Inverse Reinforcement Learning”. In: *Proc. Seventeenth International Conference on Machine Learning (ICML)*. 2000, pp. 663–670. ISBN: 1-55860-707-2 (Cited on pages 24 and 116).

[Ong 2010] Sylvie C. W. Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. "Planning Under Uncertainty for Robotic Tasks with Mixed Observability". In: *International Journal of Robotics Research* 29.8 (July 2010), pp. 1053–1068. ISSN: 0278-3649 (Cited on page 121).

[Otte 2015] Michael Otte and Emilio Frazzoli. "RRT-X: Asymptotically Optimal Single-Query Sampling-Based Motion Planning with Quick Replanning". In: *International Journal of Robotics Research* (2015) (Cited on page 13).

[Pacchierotti 2006] Elena Pacchierotti, Henrik Christensen, and Patric Jensfelt. "Evaluation of Passing Distance for Social Robots". In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (Sept. 2006), pp. 315–320 (Cited on page 26).

[Pacherie 2012] Elisabeth Pacherie. "The Phenomenology of Joint Action: Self-Agency Vs. Joint-Agency". In: *Joint Attention: New Developments*. Ed. by Seemann Axel. 2012 (Cited on page 53).

[Palmieri 2014] Luigi Palmieri and Kai O. Arras. "Efficient and Smooth RRT Motion Planning Using a Novel Extend Function for Wheeled Mobile Robots". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2014 (Cited on page 13).

[Park 2013] Jin Hyoung Park, Francisco Arturo Rojas, and Hyun Seung Yang. "A Collision Avoidance Behavior Model for Crowd Simulation Based on Psychological Findings". In: *Computer Animation and Virtual Worlds* 24 (3-4 2013), pp. 173–183. ISSN: 1546-427X (Cited on pages 18 and 100).

[Patla 1999] Aftab E. Patla, Allan Adkin, and Tonya Ballard. "Online Steering: Coordination and Control of Body Center of Mass, Head and Body Reorientation". In: *Experimental Brain Research* 129.4 (1999), pp. 629–634. ISSN: 0014-4819 (Cited on page 95).

[Perez 2012] Alejandro Perez, Robert Platt, George Konidaris, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. "LQR-RRT*: Optimal Sampling-Based Motion Planning with Automatically Derived Extension Heuristics". In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. 2012, pp. 2537–2542 (Cited on page 13).

[PérezHigueras 2014] Noé Pérez-Higueras, Rafael Ramón-Vigo, Fernando Caballero, and Luis Merino. "Robot Local Navigation with Learned Social Cost Functions". In: *Proc. 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. International Conference on Informatics in Control, Automation and Robotics. Sept. 2014 (Cited on pages 24 and 25).

[Phillips 2011] Mike Phillips and Maxim Likhachev. "SIPP: Safe Interval Path Planning for Dynamic Environments". In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. May 2011, pp. 5628–5635 (Cited on page 16).

[Pineau 2001] Joelle Pineau, Nicholas Roy, and Sebastian Thrun. “A Hierarchical Approach to POMDP Planning and Execution”. In: *Proc. Workshop on Hierarchy and Memory in Reinforcement Learning (ICML)*. 2001 (Cited on page 122).

[Quinlan 1993] Sean Quinlan and Oussama Khatib. “Elastic Bands: Connecting Path Planning and Control”. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. ICRA. May 1993, pp. 802–807 (Cited on pages 13, 14, 54 and 58).

[Quinlan 1995] Sean Quinlan. “Real-Time Modification of Collision-Free Paths”. PhD thesis. Stanford University, 1995 (Cited on page 14).

[RamónVigo 2014] Rafael Ramón-Vigo, Noé Pérez-Higueras, Fernando Caballero, and Luis Merino. “Transferring Human Navigation Behaviors into a Robot Local Planner”. In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2014, pp. 774–779. ISBN: 978-1-4799-6764-3 (Cited on pages 24 and 116).

[RiosMartinez 2011] Jorge Rios-Martinez, Anne Spalanzani, and Christian Laugier. “Understanding Human Interaction for Probabilistic Autonomous Navigation Using Risk-RRT Approach”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2011, pp. 2014–2019 (Cited on page 22).

[RiosMartinez 2014] Jorge Rios-Martinez, Anne Spalanzani, and Christian Laugier. “From Proxemics Theory to Socially-Aware Navigation: A Survey”. In: *International Journal of Social Robotics* 7.2 (Sept. 2014), pp. 137–153. ISSN: 1875-4791 (Cited on pages 17 and 52).

[Rösmann 2012] Christoph Rösmann, Wendelin Feiten, Thomas Woesch, Frank Hoffmann, and Torsten Bertram. “Trajectory Modification Considering Dynamic Constraints of Autonomous Robots”. In: *Proc. German Conference on Robotics (ROBOTIK)*. ROBOTIK. May 2012, pp. 1–6 (Cited on pages 14, 58, 59, 60, 61 and 64).

[Rösmann 2013] Christoph Rösmann, Wendelin Feiten, Thomas Wösch, Frank Hoffmann, and Torsten Bertram. “Efficient Trajectory Optimization Using a Sparse Model”. In: *Proc. European Conference on Mobile Robots (ECMR)*. ECMR. Sept. 25–27, 2013, pp. 138–143 (Cited on pages 14 and 59).

[Rösmann 2015] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. “Planning of Multiple Robot Trajectories in Distinctive Topologies”. In: *Proc. European Conference on Mobile Robots (ECMR)*. Sept. 2015, pp. 1–6 (Cited on page 15).

[Saaty 1990] Thomas L. Saaty. “How to Make a Decision: The Analytic Hierarchy Process”. In: *European Journal of Operational Research* 48.1 (1990), pp. 9–26. ISSN: 0377-2217 (Cited on page 98).

[Samejima 2015] Ippei Samejima, Yuma Nihei, Naotaka Hatao, Satoshi Kagami, Hiroshi Mizoguchi, Hiroshi Takemura, and Akihiro Osaki. “Building Environmental Maps of Human Activity for a Mobile Service Robot at the ”Miraikan” Museum”. In: *Proc. Conference on Field and Service Robotics (FSR).* Ed. by Luis Mejias, Peter Corke, and Jonathan Roberts. Vol. 105. 2015, pp. 409–422. ISBN: 978-3-319-07487-0 (Cited on page 121).

[Sardar 2012] Aziez Sardar, Michiel Joosse, Astrid Weiss, and Vanessa Evers. “Don’t Stand so Close to Me: Users’ Attitudinal and Behavioral Responses to Personal Space Invasion by Robots”. In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI).* Mar. 2012, pp. 229–230 (Cited on page 17).

[Satake 2009] Satoru Satake, Takayuki Kanda, Dylan F. Glas, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. “How to Approach Humans? - Strategies for Social Robots to Initiate Interaction”. In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI).* Mar. 2009, pp. 109–116 (Cited on page 116).

[Sebanz 2006] Natalie Sebanz, Harold Bekkering, and Günther Knoblich. “Joint Action: Bodies and Minds Moving Together”. In: *Trends in Cognitive Sciences* 10.2 (2006), pp. 70–76. ISSN: 1364-6613 (Cited on page 53).

[Seder 2007] Marija Seder and Ivan Petrovic. “Dynamic Window Based Approach to Mobile Robot Motion Control in the Presence of Moving Obstacles”. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA).* 2007, pp. 1986–1991 (Cited on page 22).

[Shah 2011] Julie Shah, James Wiken, Brian C. Williams, and Cynthia Breazeal. “Improved Human-Robot Team Performance Using Chaski, A Human-Inspired Plan Execution System”. In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI).* HRI. Mar. 6–9, 2011, pp. 29–36 (Cited on page 58).

[Shiomi 2014] Masahiro Shiomi, Francesco Zanlungo, Takayuki Kanda, and Kotaro Hayashi. “Towards a Socially Acceptable Collision Avoidance for a Mobile Robot Navigating Among Pedestrians Using a Pedestrian Model”. In: *International Journal of Social Robotics* 6.3 (2014), pp. 443–455 (Cited on page 22).

[Shomin 2014] Michael Shomin, Bhaskar Vaidya, Ralph Hollis, and Jodi Forlizzi. “Human-Approaching Trajectories for a Person-Sized Balancing Robot”. In: *Proc. IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO).* Sept. 2014, pp. 20–25 (Cited on page 116).

[Sisbot 2007] Emrah Akin Sisbot, Luis F. Marin-Urias, Rachid Alami, and Thierry Siméon. “A Human Aware Mobile Robot Motion Planner”. In: *IEEE Transactions on Robotics* 23.5 (Oct. 2007), pp. 874–883. ISSN: 1552-3098 (Cited on pages 19, 30, 52, 66, 70, 123 and 127).

[Spalanzani 2012] Anne Spalanzani, Jorge Rios-Martinez, Christian Laugier, and Sukhan Lee. "Risk Based Navigation Decisions". In: *Handbook of Intelligent Vehicles*. Ed. by Azim Eskandarian. 2012, pp. 1459–1477. ISBN: 978-0-85729-085-4 (Cited on page 22).

[Srinivasan 2011] Vasant Srinivasan and Robin Murphy. "A Survey of Social Gaze". In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2011, pp. 253–254. ISBN: 978-1-4503-0561-7 (Cited on page 97).

[Srinivasan 2012] Vasant Srinivasan, Cindy L. Bethel, Clifford Nass, and Robin Murphy. "Validation of a Behavioral Robotics Framework for Social Head Gaze". In: *Proc. "Gaze in Human–Robot Interaction" Workshop Held at HRI*. Mar. 2012 (Cited on page 97).

[Srinivasan 2015] Vasant Srinivasan, Robin R. Murphy, and Cindy L. Bethel. "A Reference Architecture for Social Head Gaze Generation in Social Robotics". In: *International Journal of Social Robotics* 7.5 (2015), pp. 601–616. ISSN: 1875-4805 (Cited on pages 96 and 97).

[Syrdal 2008] Dag Sverre Syrdal, Nuno Otero, and Kerstin Dautenhahn. "Video Prototyping in Human-Robot Interaction: Results from a Qualitative Study". In: *Proc. European Conference on Cognitive Ergonomics (ECCE)*. 2008 (Cited on pages 104 and 112).

[Takayama 2011] Leila Takayama, Doug Dooley, and Wendy Ju. "Expressing Thought: Improving Robot Readability with Animation Principles". In: *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Mar. 6–9, 2011, pp. 69–76 (Cited on pages 95 and 104).

[Tay 2007] Meng Keat Christopher Tay and Christian Laugier. "Modelling Smooth Paths Using Gaussian Processes". In: *Proc. Field and Service Robotics (FSR)*. July 2007, pp. 381–390 (Cited on page 22).

[Thrun 1999] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B. Cremers, Frank Dellaert, Dieter Fox, Dirk Hahnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, and Dirk Schulz. "MINERVA: A Second-Generation Museum Tour-Guide Robot". In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 3. 1999, 1999–2005 vol.3 (Cited on page 121).

[Tomasello 2005] Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne, and Henrike Moll. "Understanding and Sharing Intentions: The Origins of Cultural Cognition". In: *Behavoral and Brain Sciences* 28.5 (2005), pp. 675–691 (Cited on page 53).

[Toris 2015] Russell Toris, Julius Kammerl, David V. Lu, Jihoon Lee, Odest Chadwicke Jenkins, Sarah Osentoski, Michell Wills, and Sonia Chernova. "Robot Web Tools: Efficient Messaging for Cloud Robotics". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2015, pp. 4530–4537 (Cited on page 35).

[Torta 2013] Elena Torta, Raymond H. Cuijpers, and James F. Juola. “Design of a Parametric Model of Personal Space for Robotic Social Navigation”. In: *International Journal of Social Robotics* 5.3 (2013), pp. 357–365 (Cited on page 17).

[Triebel 2015] Rudolph Triebel, Kai Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, Daniel Cremers, Vanessa Evers, Michelangelo Fiore, Hayley Hung, Omar A. Islas Ramírez, Michiel Joosse, Harmish Khambhaita, Tomasz Kucner, Bastian Leibe, Achim J. Lilienthal, Timm Linder, Manja Lohse, Martin Magnusson, Billy Okal, Luigi Palmieri, Umer Rafi, Marieke van Rooij, and Lu Zhang. “SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports”. In: *Proc. Conference on Field and Service Robotics (FSR).* Ed. by S. David Wettergreen and D. Timothy Barfoot. June 24–26, 2015, pp. 607–622. ISBN: 978-3-319-27702-8 (Cited on pages 77 and 102).

[Unhelkar 2015] Vaibhav V. Unhelkar, Claudia Pérez-D’Arpino, Leia Stirling, and Julie A. Shah. “Human-Robot Co-Navigation Using Anticipatory Indicators of Human Walking Motion”. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA).* May 26–30, 2015, pp. 6183–6190 (Cited on page 95).

[Van de Weghe 2004] Nico Van de Weghe. “Representing and Reasoning about Moving Objects: A Qualitative Approach”. PhD thesis. Ghent University, 2004 (Cited on page 18).

[Vemula 2016] Anirudh Vemula, Katharina Muelling, and Jean Hyaejin Oh. “Path Planning in Dynamic Environments with Adaptive Dimensionality”. In: *Proc. Ninth International Symposium on Combinatorial Search (SoCS).* July 2016 (Cited on page 16).

[Waldhart 2015] Jules Waldhart, Mamoun Gharbi, and Rachid Alami. “Planning Handovers Involving Humans and Robots in Constrained Environment”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Sept. 28–Oct. 2, 2015, pp. 6473–6478 (Cited on pages 23, 58 and 76).

[Webb 2013] Dustin J. Webb and Jur van den Berg. “Kinodynamic RRT*: Asymptotically Optimal Motion Planning for Robots with Linear Dynamics”. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA)* (May 2013), pp. 5054–5061. ISSN: 1050-4729 (Cited on page 13).

[Yoo 2015] Bum-Soo Yoo and Jong-Hwan Kim. “Fuzzy Integral-Based Gaze Control of a Robotic Head for Human Robot Interaction”. In: *IEEE Transactions on Cybernetics* 45.9 (Sept. 2015), pp. 1769–1783. ISSN: 2168-2267 (Cited on page 96).

[Zanlungo 2011] F. Zanlungo, T. Ikeda, and T. Kanda. “Social Force Model with Explicit Collision Prediction”. In: *Europhysics Letters* 93.6 (Mar. 2011), p. 68005. ISSN: 0295-5075 (Cited on pages 18 and 22).

[Zaraki 2014] Abolfazl Zaraki, Daniele Mazzei, Manuel Giuliani, and Danilo De Rossi. "Designing and Evaluating a Social Gaze-Control System for a Humanoid Robot". In: *IEEE Transactions on Human-Machine Systems* 44.2 (Apr. 2014), pp. 157–168. ISSN: 2168-2291 (Cited on pages 93 and 96).

[Zhifei 2012] Shao Zhifei and Er Meng Joo. "A Review of Inverse Reinforcement Learning Theory and Recent Advances". In: *IEEE Congress on Evolutionary Computation (CEC)*. Congress on Evolutionary Computation. June 2012, pp. 1–8. ISBN: 978-1-4673-1510-4 (Cited on page 24).

[Ziebart 2008] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. "Maximum Entropy Inverse Reinforcement Learning". In: *Proc. Twenty-Third AAAI Conference on Artificial Intelligence (AAAI)*. 2008, pp. 1433–1438 (Cited on page 25).