HUMAN-AWARE ROBOT NAVIGATION USING THE SOCIAL FORCE MODEL

by

Yiğit Yıldırım

B.S., Computer Engineering, Boğaziçi University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2015

ACKNOWLEDGEMENTS

First of all, I would like to express my sincerest gratitude to my supervisor, Prof. H. Levent Akın. His guidance, support and unending patience made this thesis possible.

I am grateful to Assoc. Prof. Hatice Köse and Prof. A. C. Cem Say for kindly accepting to be members of my thesis committee and for their constructive suggestions.

I would like to dedicate this thesis to my dearest friend Görkem H. Tanrıverdi. Her presence in my life was a bliss. I miss her everyday.

I am indebted to Okan Aşık and Tekin Meriçli who helped me express my work much better.

I am thankful to my family and friends, who patiently supported me and during this journey and helped me through it.

I also would like to thank Gizem Ünsal who encouraged me unceasingly, stood by me during this time and supported me whenever I need.

This thesis has been supported by Boğaziçi University Scientific Research Fund project BAP 7361.

ABSTRACT

HUMAN-AWARE ROBOT NAVIGATION USING THE SOCIAL FORCE MODEL

Today, the long-lived dream of employing autonomous mobile robots in human environments is more close to coming into reality than ever. Most of the fundamental navigation problems are solved due to an extensive research on the field of mobile robotics. Today, there are many navigation strategies that enable robots to navigate through their surrounding environments in a collision-free manner. However, Human-Robot Interaction (HRI), suggests that, in addition to being safe, navigation of a robot must be understandable to humans, as well. Comprehensibility of the movements of a robot increases its acceptance among humans. This way, it is believed by many that, we will begin to internalize the presence of mobile robots in our daily lives. The concept of *human-aware navigation* refers to these shortcomings of the previous methods in this direction. Social Force Model was proposed to describe the motion of the pedestrians. We use this approach to develop a human-aware navigation module for our tour-guide robot. This model is a variant of a large-family, so-called *artificial potential fields*. The nature of the model makes it a good candidate for local path planning. In this work, we present the model and the differences of it from fundamental path planning algorithms. We explain the implementation of the model as a local path planner and show on simulated environment that a more socially-acceptable motion may be achieved using this model.

ÖZET

SOSYAL KUVVET MODELİ KULLANILARAK İNSAN-FARKINDA ROBOT NAVİGASYONU SAĞLANMASI

Otonom robotların insanlara ait yaşam alanlarında görevlendirilmesi hedefi, uzun soluklu bir tutku olmakla birlikte, günümüzde gerçekleşmeye - evvele kıyasla - çok daha yakındır. Robot navigasyonunda karşılaşılan temel sorunların birçoğu, bu alanda yapılan yoğun çalışmalar sonucu çözülmüştür. Literatürde, hareketli robotların hiçbir yere çarpmadan hedeflerine ulaşmalarını sağlayacak birçok navigasyon yöntemi bulunmaktadır. Bununla birlikte, İnsan-Robot Etkileşimi (İRE) çalışmaları, hareketli robotların çarpışmasızlık açısından güvenli olmalarının yanında, hareketlerinin insanlar tarafından anlaşılabilir olmasının da gerekli olduğunu ortaya koymaktadır. Hareketli robotların hareketlerindeki anlaşılırlık, onların insan nezdinde kabul görme oranını artıracaktır. Bu sayede, biz insanlar, gündelik hayattaki robot varlığını içselleştirmeye başlayacağız. *İnsan-farkında navigasyon*, eski navigasyon yöntemlerinin bu doğrultudaki eksikliklerine atfen ortaya çıkmıştır. Sosyal Kuvvet Modeli, ilk olarak, yayaların hareketlerini matematiksel olarak ifade etmek amacıyla ortaya atılmıştır. Bu yaklaşım, tur rehberi olarak görev alacak robotumuzun üzerinde çalışmasını planladığımız, insan-farkında navigasyon modülünü geliştirmek için kullanılmıştır. Bu model, *yapay potansiyel alanlar* ailesinin bir üyesidir; doğası itibariyle, robot navigasyonunda yerel patika planlayıcı olarak kullanılmaya uygundur. Çalışmada, söz konusu model sunulmakta ve onun temel patika planlama algoritmalarından farkları anlatılmıştır. Yaklaşımın, yerel patika planlayıcı olarak programlanış detayları aktarılmakta ve benzetim ortamında, sosyal olarak kabul edilebilirliği daha yüksek navigasyon üreten algoritmamızın işleyişi gösterilmektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . xii
LIST OF SYMBOLS . . . xiii
LIST OF ACRONYMS/ABBREVIATIONS . . . xv
1. INTRODUCTION . . . 1
2. BACKGROUND . . . 3
2.1. Overview . . . 3
2.2. Autonomous Robot Navigation . . . 3
2.2.1. Museum Tour-Guide Robots . . . 4
2.2.2. Other Public Installations . . . 8
2.3. Description of the Problem . . . 10
2.4. Human-Robot Interaction . . . 11
2.5. Human-Aware Navigation . . . 13
2.5.1. Increasing Comfort . . . 15
2.5.1.1. Distance . . . 15
2.5.1.2. Speed . . . 23
2.5.1.3. Direction . . . 24
2.5.1.4. Gaze, Appearance and Size . . . 24
2.5.2. Increasing Naturalness . . . 25
2.5.3. Increasing Sociability . . . 26
3. PLATFORM SPECIFICATIONS . . . 28
3.1. Software Platforms . . . 28
3.1.1. Robot Operating System . . . 28
3.1.2. V-REP Robot Simulator . . . 30
3.2. The Hardware Platform . . . 31
4. METHODOLOGY AND IMPLEMENTATION . . . 34

4.1. Methodology 34
4.1.1. Mathematical Description of the Social Force Model 34
4.1.2. Social Force Model as a Local Planner for Robot Navigation 36
4.2. Implementation 38
4.2.1. Infrastructure of the Local Path Planner 39
4.2.2. Implementation of the Local Path Planner in the Simulated Environment 41
4.2.2.1. Robot Tracker 41
4.2.2.2. People Tracker 42
4.2.2.3. Obstacle Tracker 42
4.2.2.4. Social Navigation Unit 45
4.2.2.5. Auxiliary Modules 53
4.2.3. Parameters Learning 55
5. EXPERIMENTS AND RESULTS 59
5.1. Overview 59
5.2. Parameters-Learning Experiments 60
5.2.1. Training Experiments with The First Data Set 62
5.2.2. Training Experiments with The Second Data Set 63
5.3. Validation Experiments 65
5.3.1. Experiments on an Empty World 65
5.3.2. Experiments with One Person 66
5.3.3. Experiments with Two Obstacles 69
5.3.4. Experiments with Two People 72
5.3.5. Experiments with One Person and One Obstacle 78
6. CONCLUSION 82
6.1. Future Work 83
REFERENCES 84

LIST OF FIGURES

Figure 2.1. Dynamic Window Approach. 4

Figure 2.2. Flow diagram of the path planner of *Robox*. 7

Figure 2.3. A social interaction experiment with *Maggie*. 12

Figure 2.4. An example scenario comparing safe navigation with human-aware navigation. 14

Figure 2.5. Distribution of the publications on socially-aware navigation over years. 14

Figure 2.6. Operation of the navigation system designed by Christensen and Pacchierotti. 17

Figure 2.7. Static and general comfort spaces. 18

Figure 2.8. Application of a Gaussian form of cost distribution. 19

Figure 2.9. Dynamic personal space. 20

Figure 2.10. Social conventions and corresponding rules of navigation. 21

Figure 2.11. Representation of the personal space as a sum of Gaussian distributions. 22

Figure 3.1. Tree of coordinate frames. 30

Figure 3.2. The robotic platform *TURGAY*. 33

Figure 4.1. Local minimum. 37

Figure 4.2. An example map. 38

Figure 4.3. The overall infrastructure. 40

Figure 4.4. A map and it's metadata. 43

Figure 4.5. The relation between resolution and *Number of Cells* within *ROI*. . . 44

Figure 4.6. 2D Ray-Tracing algorithm. 46

Figure 4.7. Fast hypotenuse algorithm. 47

Figure 4.8. Operation of the *Obstacle Tracker* module. 48

Figure 4.9. Calculation of the attractive force. 49

Figure 4.10. Repulsive force generated by humans. 50

Figure 4.11. Repulsive force generated by obstacles. 50

Figure 4.12. Calculation of the repulsive force. 51

Figure 4.13. Calculated social forces. 52

Figure 4.14. *Force* coordinate frame. 53

Figure 4.15. The comparison of people-generated and obstacle-generated potentials. 56

Figure 4.16. The comparison of the cumulative magnitudes of people-generated and obstacle-generated repulsive forces. 57

Figure 5.1. Social forces affecting the agent. 60

Figure 5.2. Example scenarios in simulated environment. 61

Figure 5.3. Summary of the first linear model. 62

Figure 5.4. Summary of the second linear model. 64

Figure 5.5. Simulated environment with no obstacles and no people. 65

Figure 5.6. Trajectories in the environment with no obstacles and no people. . . . 67

Figure 5.7. Comparison of average paths and average cumulative errors in the environment with no obstacles and no people. 68

Figure 5.8. Simulated environment with no obstacles and one person. 69

Figure 5.9. Trajectories in the environment with no obstacles and one person. . . 70

Figure 5.10. Comparison of average paths in the environment with no obstacles and one person. 71

Figure 5.11. Performance comparison with respect to the *path length* and *proxemic intrusion* metrics in the environment with no obstacles and one person. 71

Figure 5.12. Simulated environment with two obstacles and no people. 72

Figure 5.13. Trajectories in the environment with two obstacles and no people. . . 73

Figure 5.14. Comparison of average paths in the environment with two obstacles and no people. 74

Figure 5.15. Performance comparison with respect to the *path length* metric in the environment with two obstacles and no people. 74

Figure 5.16. Simulated environment with no obstacles and two people. 75

Figure 5.17. Trajectories in the environment with no obstacles and two people. . . 76

Figure 5.18. Comparison of average paths in the environment with no obstacles and two people. 77

Figure 5.19. Performance comparison with respect to the *path length* and *proxemic intrusion* metrics in the environment with no obstacles and two people. 77

Figure 5.20. Simulated environment with one obstacle and one person. 78

Figure 5.21. Trajectories in the environment with one obstacle and one person. . . 79

Figure 5.22. Comparison of average paths in the environment with one obstacle and one person. 80

Figure 5.23. Performance comparison with respect to the *path length* and *proxemic intrusion* metrics in the environment with one obstacle and one person. 80

LIST OF TABLES

Table 2.1. Interpersonal distances as proposed by Hall. 16

Table 5.1. SFM parameters as presented by Zanlungo *et al*. 59

Table 5.2. SFM parameters as presented by Ferrer *et al*. 60

Table 5.3. First set of weights calculated by multiple linear regression. 62

Table 5.4. Second set of weights calculated by multiple linear regression. 64

LIST OF SYMBOLS

A_q	Intensity of the interaction
B_q	Range of the interaction
d_{ij}	Distance between the pedestrian i and entity j
$\hat{e}_i$	Desired direction of the motion of pedestrian i
$\vec{F}_i$	Cumulative force applied to pedestrian i
$\vec{f}_i^{attr}$	Attractive force applied to pedestrian i
$\vec{f}_i^{rep}$	Repulsive force applied to pedestrian i
$\vec{f}_{i,O}^{rep}$	Repulsive force applied to pedestrian i that are generated by obstacles
$\vec{f}_{i,P}^{rep}$	Repulsive force applied to pedestrian i that are generated by other pedestrians
$g(.)$	Regression hypothesis
i	The pedestrian
j	The entity
m_i	The mass of pedestrian i
N	The number of elements in the set
$\hat{n}_{ij}$	Unit vector pointing from pedestrian i to entity j
$p-value$	A statistic for testing the hypothesis
q	The type of the entity
Q	The set of all obstacles and pedestrians
$\vec{r}_i$	Current position of pedestrian i
$\vec{r}_i^0$	The position of the navigation target
r^n	Component of the given velocity command
R^2	A statistic for measuring the closeness of data and the model
v_i^0	The desired speed of pedestrian i
$\vec{v}_i$	Velocity of pedestrian i
$\vec{v}_i^0$	The desired velocity of pedestrian i
$\vec{w}$	Vector of weight coefficients
$\vec{x}^n$	Vector of the calculated force values

$\mathscr{X}$	Data set
α	Weight coefficient for attractive force
β	Weight coefficient for obstacle-generated repulsive force
γ	Weight coefficient for human-generated repulsive force
ε	Noise
π	Mathematical constant
τ	Relaxation time

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
AMCL	Adaptive Monte Carlo Localization
API	Application Program Interface
CS	Comfort Space
DPS	Dynamic Personal Space
DWA	Dynamic Window Approach
HRI	Human-Robot Interaction
IPS	Information Process Space
IRL	Inverse Reinforcement Learning
MCL	Monte Carlo Localization
ND	Nearness Diagram
NVC	Non-Vocal Communication
NF1	Navigation Function 1
PF	Potential Field
ROI	Region of Interest
ROS	Robot Operating System
RRT	Rapidly-exploring Random Tree
SFM	Social Force Model
SLAM	Simultaneous Localization and Mapping
VFH	Vector Field Histogram
μDWA	model-based Dynamic Window Approach

1. INTRODUCTION

Navigation is a fundamental competence for a mobile robot. The capability of autonomous navigation becomes essential when deployment of mobile service robots in dynamic environments is considered. The field of mobile robotics has been researching the incorporation of artificial intelligence strategies to mobile robot navigation in order to achieve autonomy. Recent developments in technology have corroborated the idea that service robots would soon become an inseparable part of our social environments. Moving the robots out from the large production facilities where they have been held in cages, towards our daily environments, requires these robots to be autonomous and mobile, most of the time. While this new situation opens up a wide range of new fields of application for robots, it also poses a new set of challenges for the researchers.

Researchers have been working on developing robots which are intended to carry out the tasks that require robots to engage in social interaction, such as guiding a group of visitors in a museum or serving food to the customers in a restaurant. In highly dynamic environments populated by humans, the safety of the people has been the primary consideration for a long time. Many robust navigation strategies have been developed so as to ensure a collision-free motion for the mobile robot and the safety of the people that are sharing their environment with it.

In recent years, the emergence of Human-Robot Interaction (HRI) as a proven research field has also had implications on the concept of safety, in addition to the other aspects of robotics studies. From the viewpoint of HRI, safety may refer not only to the physical but also to the mental safety of the people. Therefore, mobile robot navigation is now regarded as an interdisciplinary problem. The sciences of Sociology and Psychology now provide invaluable input to mobile robot navigation.

The concept of human-aware navigation refers to this recent perspective. It points out the shortcomings of the previous methods in autonomous navigation, such as treating a humans as yet another dynamic obstacle. Although there are statistical approaches which

have been developed quite recently to achieve human-aware navigation, models from human sciences claim to offer an increase in the performance in terms of social acceptance of these robots.

Social Force Model (SFM) is such a model. It was initially proposed to describe the motion of pedestrians [1]. The model suggests that people move under the influence of several sources, such as other pedestrians, obstacles, and the navigational target. According to the model, representation of these effects as abstract forces and treating the pedestrian as a mass obeying the laws of Newtonian mechanics may offer a mathematical solution to the complex nature of the human motion. Social Force Model is a highly renowned model and its application to the navigation framework of a mobile robot is expected to create more human-like motion patterns for the robot.

The remainder of this thesis is organized as follows: In Chapter 2, we present an extensive literature survey. The survey starts with the earliest applications of mobile robots in public domains. The focus of the survey is the navigation frameworks of the previous researches. Definition of the problem addressed in this study is given in this chapter. Afterwards, we present the state of the art in human-aware navigation research. Chapter 3 is dedicated to introduce the platforms used in this study. We introduce Robot Operating System (ROS) and V-Rep robot simulator as the centerpieces of this work. We, later, introduce our robotic platform Turgay. In Chapter 4, we present Social Force Model and its mathematical definition. Several modules exist and work cooperatively in the navigation system we developed in order to achieve motion obeying the Social Force Model. These modules and other implementation details are also given in this chapter.

We conducted two sets of experiments with the developed framework. The first set aims to learn the parameters used in the calculation of the combined force that manages the movements of the robot. The second set shows the performance of the framework using these parameters. These experiments are given in Chapter 5.

In Chapter 6, we present a discussion about the results and several future directives in hopes of improving the current status of the work.

2. BACKGROUND

2.1. Overview

Mobile robotics is the research area that studies the autonomous and semi-autonomous vehicles. The distinction of mobile robotics from the domain of conventional robotics comes from the integration of artificial intelligence strategies in order to achieve autonomy [2]. In contrast to teleoperated vehicles, which are commanded by human operators, autonomous vehicles operate without human intervention; the decision for their next set of actions is based merely on their perception.

Navigation is the ability to physically move from one state to another in the configuration space. In the intersection of the two concepts, autonomy and navigation, lies autonomous navigation. Navigating through the space in a deliberate way is an indicator of intelligence. This competence is common to most of the animals. In fact, researchers have been studying the dynamics of the motion of animals for a long time, hoping to form various models to explain the mechanisms behind this ability and then, to exploit these models to develop autonomous navigation frameworks for mobile robots [3].

The aim of this study is to develop an improved autonomous navigation framework for socially assistive mobile robots, one which is far more suitable for the environments populated by people. Thus, in the remaining of this chapter, we first present a review of the conventional methods in autonomous robot navigation and their applications. Later, we will introduce the concept of human-robot interaction and describe the problem of social navigation which is the subject to this thesis. Finally, we will present a literature survey and a taxonomy on the strategies employed to realize social navigation.

2.2. Autonomous Robot Navigation

In comparison to the research for industrial robotic manipulators, mobile robotics is a relatively younger field. One of the most alluring aspirations in this field has been to deploy

autonomous robots in human environments. Early studies on mobile robots, that were designed to work in public places, addressed more fundamental issues in the robot's motion. In these studies, safety and reliability aspects of the robots were the primary concerns. Therefore, researchers were obliged to develop navigation systems with robust collision avoidance functionalities. In the following sections, such research efforts will be reviewed with the principal consideration of the employed navigation frameworks.

2.2.1. Museum Tour-Guide Robots

The majority of the experiments in the literature that are conducted in public places involve installations of tour-guide robots at the museums. Below is a survey of those experiences, listed chronologically.

In [4], Burgard *et al*. describes the autonomous mobile prototype Rhino. It is explicitly stated in the article that safe and reliable navigation was the most essential competency that the robot must have. Rhino uses predetermined paths to navigate. These paths however, are not directly executed but given to the collision avoidance mechanism which acts as a local planner. To avoid obstacles, Rhino employs the dynamic window approach (DWA) [5]. An example scenario displaying the operation of DWA is given in Figure 2.1.

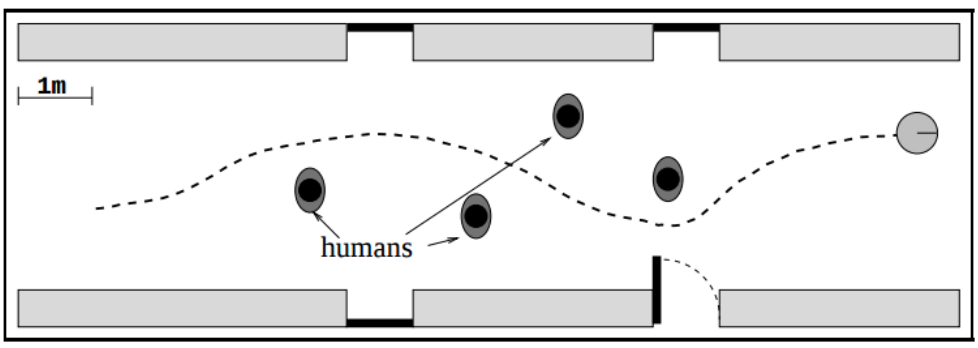


Figure 2.1. An example scenario showing the operation of the reactive collision avoidance technique Dynamic Window Approach [5].

In [6], Thrun *et al.* introduces the progress of the same team on a new prototype, named Minerva. It is the improved version of their first attempt Rhino. Minerva's path planner is called a *coastal planner* [7], which generates the global paths. This planner does not produce the shortest paths; however, by using these paths, the robot minimizes the risk of getting lost. μDWA, a derivation of DWA, is employed for the purpose of collision avoidance. It is used as a local path planner; and it produces velocity commands for the robot to follow. These commands are generated according to the predefined constraints. Whereas hard constraints require robot to make a full stop, soft ones represent the trade-offs between moving towards the goal and away from obstacles. The approach of μDWA is explained in detail in [8].

Sage is another museum tour-guide robot [9] which uses a very simple and a reactive obstacle avoidance strategy. According to the current readings of the sonar sensors, the robot calculates a reactive velocity that causes the robot to decelerate when there are people nearby. The robot makes a full stop if its path is blocked. The path planner employed in Sage's navigation system is rather simple, too. Previously placed landmark are followed during the operation of the robot. The robot turns toward the next landmark whenever it reaches one.

Chips and Mobot museum robot prototypes are introduced in [10]. It is explicitly stated in the publication that the safety of the people and the robot itself is the main requirement during the experiments. Relying on the distance readings from sonar sensors, the robots in these experiments employ a very simple method for obstacle avoidance. They reactively move around the detected obstacles diverging from the previously determined trajectory. This simple method leads to the situations in which the robots become stuck. Due to the infrequency of these situations, the authors neglect the issue and do not consider any further investigation.

In [11], Graf and Barth introduce their tour-guide robot prototype. Installed in the *Museum für Kommunikation* in Berlin, the robot uses a laser scanner to localize itself and detect obstacles. Two navigation modes are employed for the motion of this robot. One is the reactive navigation mode using target positions for the robot which are constantly recalculated momentarily in reaction to its environment. The second mode is the preplanned

path. When the robot is provided with a certain goal position, it calculates the shortest path using the strategy described in [12] based on the static map of the environment. For safety purposes, a three layer redundant safety system is implemented. On level one, the velocity of the robot is decreased whenever an obstacle is detected in the vicinity of the robot. If the obstacles is closer than a predefined distance, the robot makes a full stop and waits for the area to become clear again. On the second layer, the PolarBug algorithm [13] is implemented. The authors state that since this algorithm is faster than the common obstacle avoidance methods, this choice leads to a higher efficiency. On the third layer, there is an emergency stop behavior for the cases of software failures.

As described in [14], a 3-layered path planner is used in the tour-guide robot prototype Robox. A global planner is at the topmost level generating global paths by employing graph search algorithms. The specific map representation allows global planner to work this way in which the locations of interest are represented as nodes and edges denote traversability between these nodes forming a graph. The second layer uses the *NF1* method [15] in a local grid around the robot. *NF1* is a wave propagation technique that eliminates local minima on the map. However, the poor geometry of the paths produced by *NF1* method, enforces a third layer in which paths are smoothened using the elastic band method [16]. For obstacle avoidance, Robox also employs DWA method just as Rhino does with two modifications. The overall framework is given in Figure 2.2, explained in [17]. In this paper, the authors focus on the local path planner and the obstacle avoiding mechanism of their tour-guide robot by describing the above-mentioned methods and the alterations they did on these methods.

The museum tour-guide robot Jinny is introduced in [18] and the detailed description of the navigation framework is given in [19]. Accordingly, there is a hybrid path planner employed in Jinny's navigation system including both a global and a local path planner. Since the topological maps built for Jinny's operations represent the environment using nodes and edges, the global path planner employs graph search strategies to find an optimal path of nodes to the target location. Then, the local path planner generates shortest collision-free path between the nodes. In cases where the path is blocked by a dynamic obstacle, such as a human, the robot stops and waits for the generation of a new collision-free path.

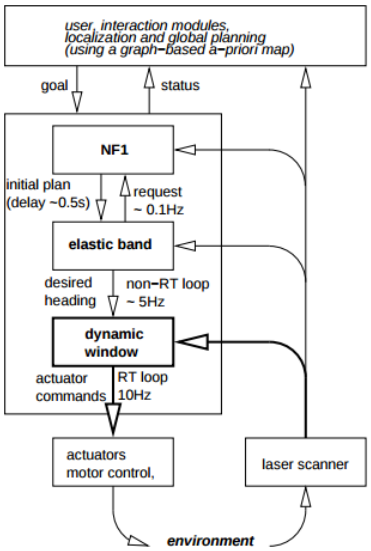


Figure 2.2. 3-layered architecture of Robox's path planner [17].

Urbano, which is the improved version of the interactive mobile robot Blacky [20], is introduced in [21]. In Urbano, a similar planner is used as in Jinny. A^* search algorithm is employed to produce global paths and a reactive controller moves the robot to the next way-point slightly diverging from the path in the case of encountering an obstacle. For safety purposes, they limit the robot to approach only within some predetermined range.

The navigation system of the humanoid tour-guide robot platform Robotinho, introduced in [22], is another example of the adoption of two-layered path planners. In its global path planner, Robotinho applies the Dijkstra algorithm to find the shortest path. Again, this path comprises several way-points. The local planner moves the robot towards the next way-point by using the Potential Field (PF) approach [23], which also avoids obstacles at the same time.

This survey on tour-guide robots includes the majority, if not all, of the notable applications in the literature. We can confidently say that, the most recent trends in path planner choices used in navigation frameworks indicate that the adoption of two-layered hybrid path planners including both a global and a local path planner are the most suitable choice for public places, such as museums. There are other publications in which robots are used in

other public places. In the next part, a list of these studies will be given.

2.2.2. Other Public Installations

Below is a list of some of the prominent studies in which mobile robots are installed in public places other than museums. Aside from tour-guide tasks, mobile robots have been used in exhibitions, train stations, shopping malls, etc. where the density of the people is high. Once again, the principal consideration in these applications is the employed navigation frameworks. In addition, this list is chronologically sorted, too.

FIRST is a mobile robotic system employed in hospitals to carry out transportation tasks [24]. FIRST uses predetermined trajectories that are learned by the robot in a teleoperation phase. The authors rely on laser rangefinders to detect the obstacles and their types. If the obstacle is a mobile one, then it is treated differently. Two types of behaviors are implemented for these two type of confrontations. For static obstacles, firstly, the free space around the obstacle is roughly modeled. Then FIRST slightly deviates from the initial trajectory to avoid collision. In case of an encounter with a mobile obstacle, the speed of the robot is decreased.

MAid is a robotic wheelchair platform, introduced in [25]. In the experimental setup, the preferred environment is a train station in the rush hour. Therefore, motion planning and obstacle avoidance becomes more significant. In MAid, the velocity obstacle method [26] is used to calculate the trajectories of the people in the vicinity of the robot. Then the robot chooses a velocity so that it could avoid imminent collisions by prioritizing the obstacle avoidance. Avoidance maneuvers are done by selecting a velocity accordingly from the reachable avoidance velocities set calculated by the velocity obstacles method. Hybrid approaches might have been successful in this application. However, the authors prefer task prioritization. The robot moves towards the goal only when no imminent collision is predicted.

In [27], Montemerlo *et al*. introduce two robotic platforms, Pearl and Flo, that are developed for serving the elderly in an assisted living facility. It is stated that navigating in

the presence of the elderly posed an enormous risk of unintentional collisions. Therefore the navigation system employed in Pearl and Flo are similar to the ones used in Rhino [4] and Minerva [6] with modifications that emphasize the detection of the people.

GRACE, introduced in [28], is another mobile robot that operates in dynamic environments. A two-layered navigation system is adopted in GRACE. The low-level planner translates directional heading commands into velocity commands for the motors. The high-level planner, on the other hand, is an implementation of a Markov Decision Process planner [29] which is used to produce the paths with minimum costs. By reevaluating the costs at 10 Hz. the navigation framework can successfully handle dynamic obstacles by reproducing the minimum-cost path.

Rackham [30] is another guide robot, which is used in several exhibitions and other public places. Rackham uses a laser scanner to detect obstacles. A layered path planning strategy is also used in this instance. The global path planner computes trajectories using precomputed visibility graphs of the environment to which dynamic obstacles can be integrated, when necessary. A local path planner integrates the collision avoidance behavior. Finally, another layer provides reference velocities for the servo-motors. However, this implementation relies on occasional supervisor intervention.

Toomas is a shopping guide mobile robot introduced in [31]. In the paper, the dynamicity of the environment is attributed to the moving customers and staff and carts. Toomas employs a hybrid path planner. Its global path planner uses A^* search to create global paths. The obstacle avoidance mechanism is integrated into the local path planner as in most of the recent applications. In the local path planner, the Vector Field Histogram (VFH) method [32] is used in combination with a visual obstacle detection algorithm to achieve collision-free motion control.

We can see the contemporary trend of the adoption of hybrid path planners in [33], as well. In this study, the researchers aim to develop a service robot that attends to the various needs of the pedestrians in urban environments. Within its global path planner, the robot uses graph search methods to find a global path from the current position to the target. Again, the

obstacle avoidance functionality is integrated into the local path planner module.

Another navigation framework which is developed for the service robots in urban environments is described in [34]. The authors of this paper prefer to use a 3-layered path planner. The low-level planner is designed to act as a local path planner. The high and the intermediate layers operates as a global planner together, employing a different strategy. This approach, as reported in [35], leads to the generation of suboptimal paths in favor of the far shorter calculation time.

We can say that over the years a similar convergence towards the adoption of two-layered hybrid path planners have occurred by looking at these studies, as well. In some studies, the researchers did detect people in the environment. However, treating them differently than they did to obstacles did not seem to take place until the emergence of human-robot interaction as a profound research field. In the next section we will elaborate on this problem.

2.3. Description of the Problem

Some of the most influential and pioneering studies of installations of autonomous mobile robots in public places are mentioned in Section 2.2. Early studies involving the installations of autonomous mobile robots in public places address more fundamental problems of the robot's motion, such as path planning and collision avoidance. The authors explicitly refer to the safety of the people as the primary concern.

The majority of these studies had already been aiming to attain a collaboration between humans and machines, such as tour-guide robots. Although the remaining studies were not seeking for collaboration, they had required the robots to share their operation environment with people. Although many successful strategies were developed to create reliable navigation frameworks with robust collision avoidance mechanisms within the context of these studies, the misconstruction in the navigational approaches was evaluating humans as dynamic obstacles. However, co-existence and co-operation between a human and an autonomous mobile robot involves more than conversation and simple gestures. It demands coordination in the motions of both agents. Actions of both agents must be planned and

adapted correspondingly [36]. The implications of these circumstances to autonomous robot navigation reveal the obvious problem; as the autonomous robots move, they do not differentiate between people and obstacles, in contrast to how people move. This condition stands out as one of the reasons why robots are considered by people as brainless mechanical devices rather than social entities.

The goal of this study is to develop a robot navigation framework that generates more human-like motion. The underlying assumption is that humans prefer to interact with machines in the same way that they interact with other people [37]. From human-like motion we do not mean bipedal locomotion, but a general strategy. This is called *socially* or *human-aware navigation* in the domain of mobile robotics. The notion also falls within the area of expertise of human-robot interaction, which is explained in the next section.

2.4. Human-Robot Interaction

Human-robot interaction (HRI) is the field of study which addresses the understanding, design, and evaluation of robotic systems that involve communication with humans [38]. The capabilities of the robot is the determinant factor on the type of the communication. Although communication between a human and a robot may take various forms, we can classify the interaction according to the physical distance between the agents as either remote or proximate. Whereas remote interaction usually involves controlling the operation of the robot such as in teleoperation, proximate interaction may emerge as physical or social communication.

The term HRI has usually been misconceived for being restricted to conversations and simple gestures. This perspective ignores most of the fundamental interaction types. Nonverbal communication (NVC) among humans is one of these types and has been widely studied in psychology and sociology [39–41]. In [39], NVC is also listed as an important means of human social interaction since it can convey messages without speech. Figure 2.3 illustrates a scenario in which a social interaction experiment takes place. Maggie changes its movements in accordance with the input from her dance partner and achieves social interaction through NVC [42].

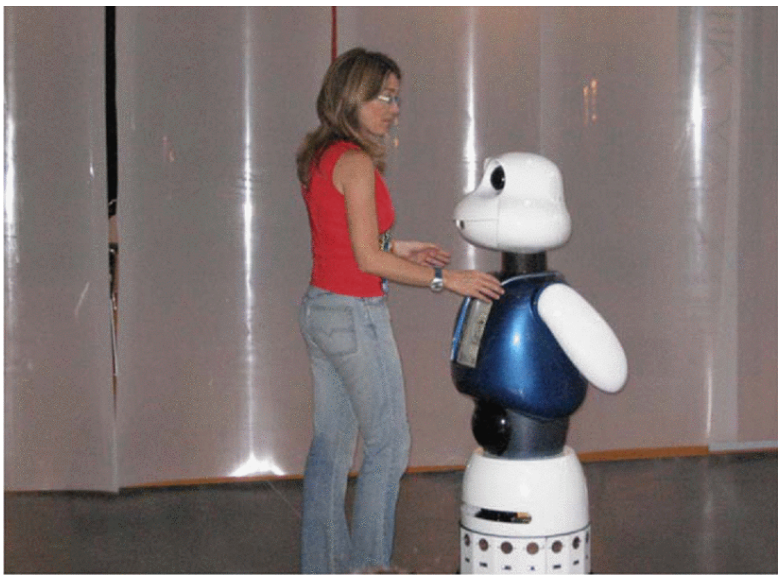


Figure 2.3. A social interaction experiment with Maggie [42].

Social interaction is formed either as a purpose, as in the case of tour-guide robots, or as a side effect, as in an encounter of a surveillance robot with a human. A human who is being distracted by the navigation of a robot is in a form of social interaction with the robot. Intimidation of the people by the motion of the robot is another example of this type of interaction. Social interaction can be established either implicitly or explicitly. Whereas explicit interaction is made with the purpose of achieving a reaction by the partner, implicit interaction defines modification of one's actions in the presence of a partner with no intended reaction [43].

Within the scope of this thesis, we consider proximate, implicit interactions; social implications of the navigation of a robot, to be precise. From this perspective, navigation is not only a task that the mobile robot performs on its own. The motion of the robot socially influences the humans around the robot and therefore, it creates a social interaction, too.

The navigation of the robot is included as a form of social interaction, also in the highly cited survey of Fong *et al*. [37]. It is stated that socially interactive robots may present social competence in reaction to the human behaviors by modeling these behaviors and responding to them. In addition, according to the taxonomy given in [44], a robot with a socially-aware

navigation system is a member of the *Social Interface* subclass. These cases in the literature encourage further research in this area.

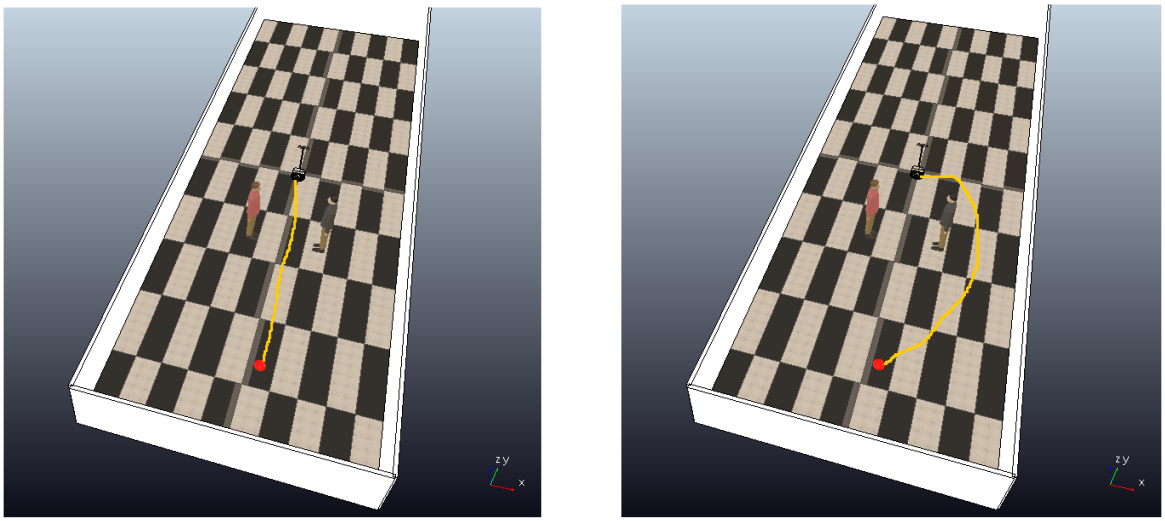
In the next section, we will give a list of studies that involve the installation of autonomous robots which pay regard to human-aware navigation.

2.5. Human-Aware Navigation

Human-aware navigation lies in the intersection of two concepts: navigation and HRI. It describes improving the navigation of the robot for the purpose of enhancing the comprehensibility of its movements by humans. In Section 2.2, some of the prominent studies which were able to develop robust navigation frameworks, that enabled the robots to move in a safe way, are given. The word safe is purported to reflect the physical safety, as in avoidance from collisions. In [45], physical and mental aspects of safety are separately evaluated. This separation reveals the need to question the psychological efficiency of navigation systems of mobile robots. Keeping in mind the assumption that humans prefer to interact with machines in the same way that they interact with other people, in order to achieve a natural integration to environments populated by people, mobile robots must be developed to be not only safe but also comprehensible. To illustrate the idea, an example scenario of human-aware navigation is given in Figure 2.4. On the left hand side, the robot plans a collision-free path, which is perfectly safe and efficient in terms of the length of the path. On the right, however, there is another path, which is safe but not so efficient. This path is more human-friendly; in that, it respects the personal and activity spaces.

Thanks to those pioneering studies and the advances in technology, the researchers have recently been focusing on the social interaction aspects of autonomous navigation. In Figure 2.5, a plot is given to illustrate the trends in the number of publications over years. The plot demonstrates the still-increasing interest to the topic.

In their comprehensive survey, Kruse *et al.* classify the strategies toward the enhancement of the comprehensibility of robot navigation into three categories as follows:



(a) A perfectly safe and efficient path to the destination.

(b) Another perfectly safe path which is less efficient but more respectful to personal spaces.

Figure 2.4. An example scenario comparing safe navigation with human-aware navigation.

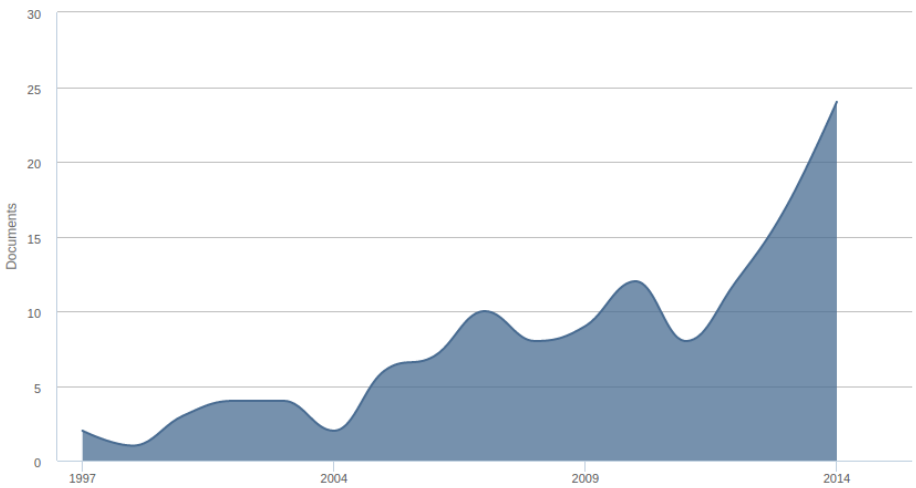


Figure 2.5. Distribution of the publications on socially-aware navigation over years [46].

- *Increasing comfort*: Removing the causes of discomfort.
- *Increasing naturalness*: Imitation of the humans to increase similarity.
- *Increasing sociability*: Exhibiting high-level cultural behaviors.

This is a reasonable taxonomy to follow for creating our own literature survey, which is given in the following.

2.5.1. Increasing Comfort

The aforementioned assumption prompts researchers to explore human behavior. People have a different perspective than the other animals in terms of comfort-distance relationship. Humans neither attack nor escape when others violate their personal space [47].

The vast majority of the studies on the concept of socially-aware navigation address comfort issues. In these studies, the researchers target eliminating the causes of discomfort. According to our extensive literature survey, a further classification can be applied to the studies that aim to maximize the comfort of robots' motions, as four main categories stand out in these studies. These categories are:

(i) Distance
(ii) Speed
(iii) Direction
(iv) Other factors, such as gaze, appearance and size

In the following, these classes will be introduced and a comprehensive list of the corresponding studies is given.

2.5.1.1. Distance. Distance is the most addressed issue in the above-mentioned studies. One of the earliest studies on socially acceptable navigation is presented in [48]. Despite taking a rather straightforward approach, this research is regarded as one of the leading studies in this domain. Nakauchi and Simmons developed an assistant robot to be used

in shopping malls that can stand in line as people do. The algorithm detects people and recognizes the end of the line firstly. Their motion planner creates a trajectory for the robot to get in the line while paying regard to the personal spaces. Therefore the robot does not intimidate people in the line.

Incorporation of the models presented by human science helps the researchers achieve further improvements in human-aware navigation. The most influential studies on the field of *human spatial behavior* is done by Hall [49]. He proposed the concept of imaginary regions around a person and coined the term *proxemics* to describe these regions. In human-human encounters, one instinctively respects this space. As expected, the actual dimensions of this area differs from one culture to another; for instance, Arabs and Americans do not follow the same customs. As he puts it:

"Americans and Arabs live in different sensory worlds much of the time and do not use the same senses even to establish most of the distances maintained during conversations."

Therefore, we can say that these ranges can change with respect to several factors, such as culture, age etc. However, they still represent the concept approximately. In the case of Northern Americans, interpersonal spaces during daily encounters can be classified as in Table 2.1:

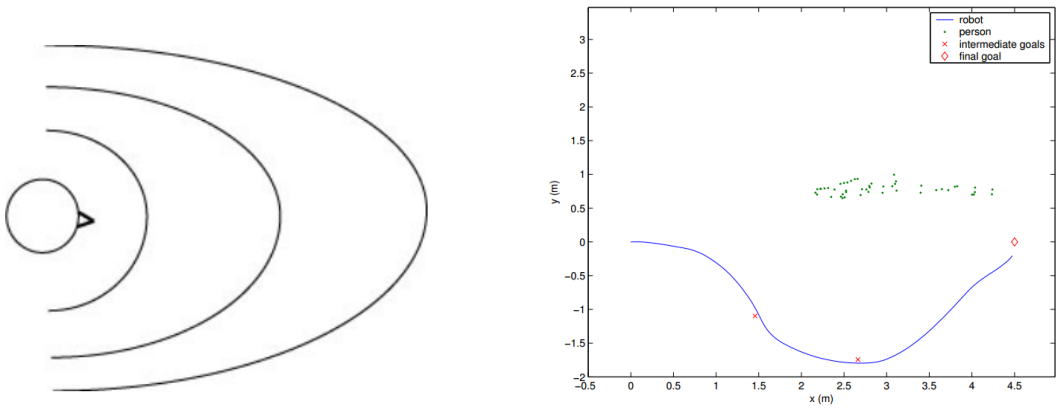
Table 2.1. Interpersonal distances as proposed by Hall in [49].

Intimate Distance	0 - 45cm
Casual/Personal Distance	45 - 120cm
Social Distance	120 - 350cm
Public Distance	beyond 350cm

The concept of proxemics have influenced many other studies in the last decade. In [50], Christensen and Pacchierotti implemented a navigation system and conducted several experiments on a robot which is passing a human walking in the opposite direction. The formalization of the passing behavior is as follows:

- Upon detecting the human, the navigation system moves robot to the right with respect to its own reference frame so that robot does not enter the personal space of the human.
- The robot then, waits there until the human passes by and returns its original trajectory afterward.

Researchers use Nearness Diagram (ND) based trajectory follower for path planning [51] to realize this rather simple plan. The usage of proxemics is different, though. Interpersonal distances are represented as ellipses and not as circles to which authors do not refer in their paper. Illustration of them and the resulting motion generated by the navigation system are given in Figure 2.6.



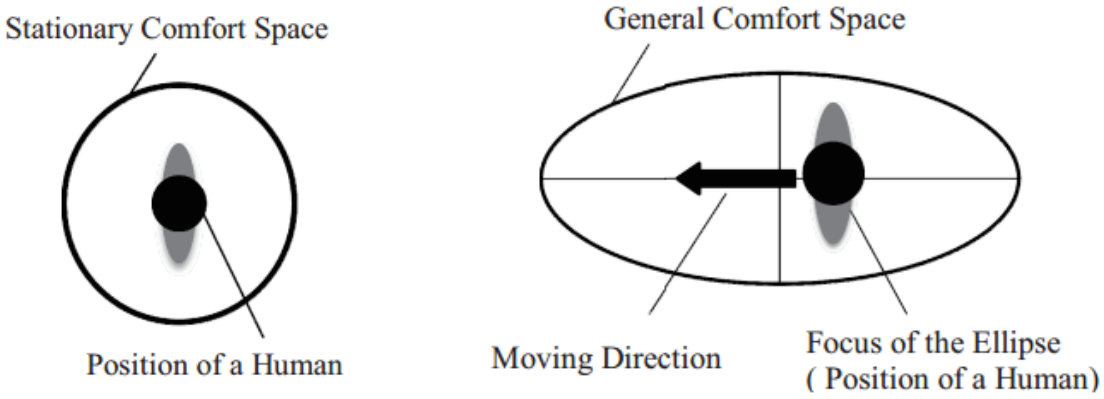
(a) Model of the proxemics zones of a moving human

(b) Resulting behavior of the navigation system.

Figure 2.6. Operation of the navigation system designed by Christensen and Pacchierotti [50].

In [52], researchers define another model of personal space based on the concept of proxemics and call it the comfort space (CS) because of the inadequacy of the of the notion of personal spaces to represent moving people. As a result, the shape of this space is affected by the velocity of the human. While the stationary CS is the same as the personal space, general CS is elliptic, as shown in Figure 2.7. Using this representation, they develop a motion controller which implements the ND method inside, and call this strategy as Human-Oriented Nearness Diagram.

In a follow-up study, Lam *et al.* improve the idea of comfort space and define sensitive



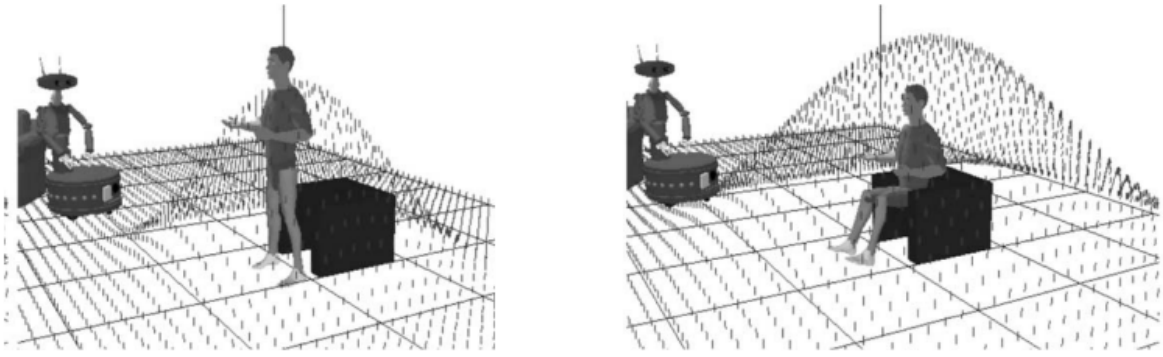
(a) Comfort space of a stationary person. (b) Comfort space of a moving person.

Figure 2.7. Static and general comfort spaces as shown in [52].

fields [53]. Six different fields are defined to represent different situations in human-aware navigation. These fields are formed according to the priorities that involve the security of both human and the robot and psychological feelings of the human, such as human-sensitive field, human-robot moving joint field, etc. This representation is integrated to a motion planner which implements ND and PF methods jointly. ND chooses free area that the robot should navigate to in order to decrease the discomfort of the people by avoiding their sensitive fields. Then, PF is used to establish different potential fields near that area so that the robot moves in that direction. Although the navigation framework is different, their idea is similar to the previous work.

Various other studies also investigate the effects of proxemics to HRI and aim to develop navigation systems that increase the social acceptance by improving the comfort of the people in its vicinity [54–57]. Some of the studies in the literature do not propose any methods to increase comfort but they present interesting findings in the scope of proxemics. In [58] and [59], researchers studied the distance preferences of the participants. One of the most important findings in these studies suggests that diameter of the personal space of a human subject decreases over the time as he/she has some experience with the robot.

Altogether, in these studies the researchers prefer strategies of defining areas around



(a) Cost applied to a standing person. (b) Cost applied to a sitting person.

Figure 2.8. Application of a Gaussian form of cost distribution, as shown in [60]. The height of the vertical lines represents the cost associated with each cell.

humans for the robot to avoid for the purpose of enhancing the comfort. However, it might be necessary to violate these areas when confined spaces taken into account. Thus, defining cost functions rather than forbidden zones around humans may be a more reasonable strategy.

Sisbot *et al.* explain the application of safety grids in [60]. These grids contains the human at the center, and a Gaussian form of cost distribution around her. Therefore each cell in the grid is associated with a cost that is inversely proportional to the distance to human. The authors also note that, as opposed to sitting, people feel less threatened when they are standing. This case is also reflected in the cost calculation via multiplying the cost by a weight for the cases of sitting people. In Figure 2.8, the applied cost distributions can be seen. After calculating the costs, the navigation task becomes a search problem of finding the minimum-cost path. They employ A^* search to find the minimum-cost path.

In [61], we observe the application of cost functions along with the concept of proxemics. In this study, the cost function is a combination of potentials. The overall potential of the environment is calculated as a summation of three other potentials:

- *Environmental potential:* Calculated according to robot's position in the environment.
- *Potential as a result of proximity to humans:* Humans are accompanied by potential

fields so as to create a repellent effect. These fields are calculated according to the interpersonal distances given in Table 2.1. The person is located at the center of the field. Since the approach from the front is preferred over the other sides, frontal potential is lower as is the cost of approaching from the front. The calculation of this potential is given in [62].

- *Potential attracting towards goal:* A potential field to draw robot into the goal position.

The use of potential fields introduces the problem of local minima. The researchers propose using Rapidly-exploring Random Trees (RRTs) [63] to overcome this problem. They modify the standard RRT algorithm to return the trajectory with the lowest cost.

A very similar approach is presented in [64]. The novelty with this study is the introduction of two new terms; social costmap, which is very similar to the overall potential in the previous work, and the dynamic personal space (DPS), which is the same interpretation of the proxemics with a new name. Formally, DPS is the combination of static personal space, back space and personal space during motion. A depiction of DPS is given in Figure 2.9.

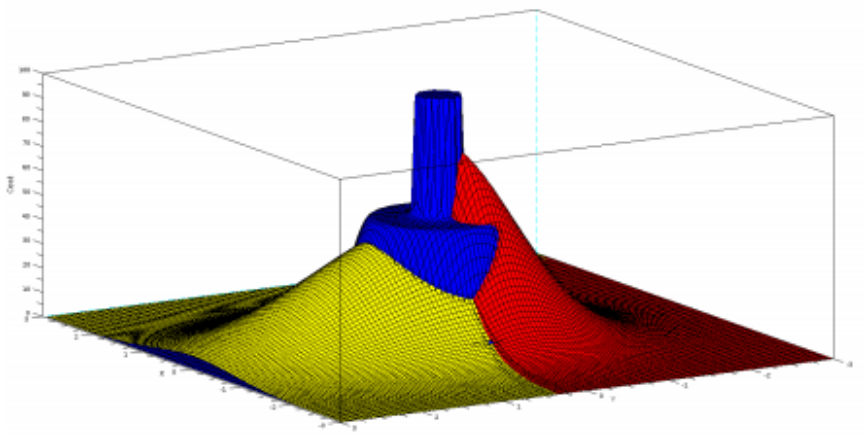


Figure 2.9. Dynamic personal space, introduced in [64].

The application of predefined cost functions represented as potential fields is an approach to model the expected behavior of humans in an encounter with robots. Integration of human sciences models is often criticized for relying on many strong assumptions. A

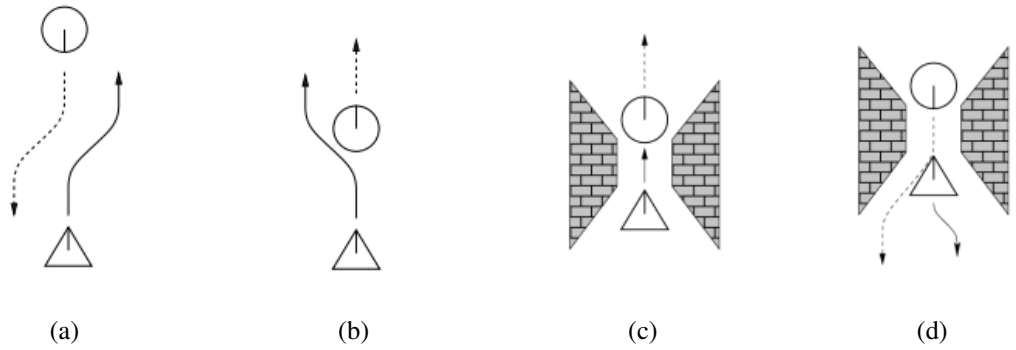


Figure 2.10. Social conventions and corresponding rules of navigation, shown in [69].

serious criticism questions whether these models scale to human-robot relations in uncontrolled conditions in [65]. On the other hand, learning methods aim to infer these models statistically. Especially, Inverse Reinforcement Learning (IRL) is a suitable choice for this particular problem. IRL allows to model the factors that motivate people's actions instead of the actions themselves. The examples of the application of IRL technique to this domain in the literature are rather recent ones, such as [65–67].

Some unconventional interpretations of the problem of enhancing the comfort in the literature also attract notice and therefore, is worth mentioning. In [68], the authors adopt a different method to enhance the comfort of the navigation task. Their approach is to generate a spatial affordance map which contains the learned information about place-time relations in humans' activities and try to avoid these places in the corresponding time intervals, unless it is necessary.

Another unique idea is to predefine social conventions and the corresponding rules to define the actions. Although the dynamics contributing to the social navigation are not yet known completely, it is a surprising fact that there is a set of different types of human-robot encounters. In [69], Dylla *et al.* defined several conventions, some examples are given in Figure 2.10, and corresponding rules. Using classification strategies, when a new encounter situation arises, the nearest corresponding action is realized by the path-planner.

Various aspects of the approach behavior are also addressed in many instances in the

literature. In [70], Kessler *et al.* aim to find a more comfortable way for a mobile robot to approach a human. Using the findings of [71], the researchers develop an approach behavior from front but with a slight deviation from the direct front since it is found to be a more preferable direction. The behavior of the robot is as follows: it drives around the person in a comfortable distance until the front position is reached. When it is reached, the robot turns towards the person and then approaches her. The comfortable distance in the previous sentence imply the use of personal space. This space is represented as a sum of Gaussian distributions, as shown in Figure 2.11. DWA is used in the navigation system. Furthermore, fast marching technique [72] is used to eliminate local minima.

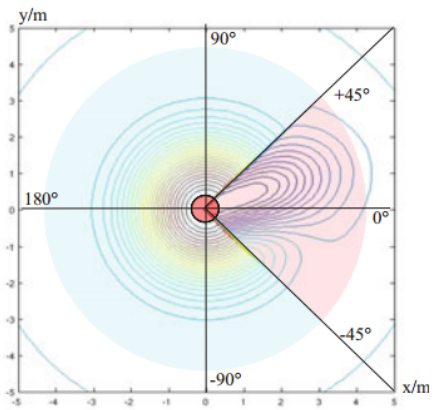


Figure 2.11. Representation of the personal space as a sum of Gaussian distributions, from [70].

In a follow-up study [73], Kessler *et al.* integrate the above-given personal-space into DWA. This time, they prefer the RRT algorithm to solve the local minima problem because random trees need to touch only a small area of the planning grid so that the computation time can be saved by evaluating only the needed cells.

Another important group of studies is formed by evaluation strategies. Several researchers have proposed various metrics for the purposes of assessing the performances of the navigation strategies, based on the concept of proxemics. A broad list of such metrics is given in [74]. In addition to these, four other evaluation metrics are defined in [65]. In this thesis, we use *path length* and *proxemic intrusion* metrics, defined by Vasquez *et al.*, in our

validation tests. Short descriptions of these metrics are given below:

(i) *Path length:* distance traveled by the robot during the navigation to the goal position.
(ii) *Proxemic intrusion:* number of intrusions into the interpersonal zones defined in [49]. We do not count violation of *public* and *social zones* as proxemic intrusion.

2.5.1.2. Speed. Some researchers target at adapting the speed of the robot to increase the comfort of the people around. These studies are mainly interested in the approach behavior of the assistive robots. In [75], the researchers explain the development of a simple human-aware navigation behavior. They claim to increase the comfort of the people in the environment, where the robot operates, by adapting the speed of the robot. This behavior is implemented as a different layer in the navigation system. The speed of the robot decreases within a predefined range when an encounter with humans occurs.

In [76], Butler and Agah present the result of their experiments with a standing person approached by a robot. The inspected behaviors include the robot approaching a person, avoiding a person while passing, and performing non-interactive tasks in an environment populated with humans. Evaluation of the robot's behaviors is done by a questionnaire filled out by the human subjects. The results show that while passing, most of the subjects feel the most comfortable when the robot approaches and adjusts its course to the right of the subject, making a smooth and non-stopping pass. The non-stop avoidance behavior is implemented using predefined constant speed and distance parameters. As a result of their study, they state that subjects rate *1 m/s* as uncomfortable, *0.5 m/s* as acceptable, and *0.254-0.381 m/s* as comfortable which suggests human prefer slower speed when they are approached by a robot. Note that this may not be true for guiding tasks; since, the mean walking speed of humans is approximately *1.4 m/s*, the slow speed of the robot leads to a boring experience most of the time.

Another study investigates how the speed of a robot affects the people around and cause them to feel insecure. During their experiment, the subjects are approached by a robot at high speeds, such as *4.5 m/s*. Although the robot in these experiments safely stop at a

distance of *0.5 m* from the subjects, they expressed that they felt unsafe. Based on these findings, the study suggests that the maximum velocity of the robot must be expressed as a function of some social radius, $v(r) \leq kr$ where $r < r_{max}$.

2.5.1.3. Direction. It is believed by some researchers that direction of the approach of the robot also plays an important role. Such studies were given as a part of other studies in the previous sections; for example, Scandolo and Fraichard create dynamic personal space around a human in [64] according to the result of the studies on this topic.

In their research Dautenhahn *et al*. [77] aim to determine the preferred direction of the people when approached by a robot. They conducted experiments with seated subjects using different approach directions. The results indicate that most of the subjects prefer to be approached from either the left or right side, with a small overall preference for a right approach rather than a frontal approach. They disprove these findings in their follow-up study in [71]; a majority of the subjects prefer the frontal approach. The authors attribute this inconsistency to the inexperience of the subjects in the first set of experiments. They state that cohabitation of a human and a robot might have decreased the intimidation the robot causes.

2.5.1.4. Gaze, Appearance and Size. There are other causes influencing the comfort of the people sharing their environment with a navigating robot, such as gaze, appearance and the size of the robot. In [78], Satake *et al*. claim improving the success of the approach by paying regard to the gaze direction of the robot while navigating. They claim that gaze orientation of the robot influences the comfort of the people . Direct gaze notifies human about the robot's awareness of his/her existence. Also Walters *et al*. presents their findings on the effects of physical features of the robot to the human-robot proxemics in [79]. They state that, whereas the size of the robot changing between *120 and 140 cm* do not have a direct effect, human-like features, such as facial expressions, increase the comfort of the people around a navigating robot.

2.5.2. Increasing Naturalness

Imitation of the human behavior may increase the social acceptance of the robot in social environments. A considerable amount of studies in the literature aim to optimize human-robot interaction by making robots move as humans do. The authors rely on the fact that a more human-like movement is more comprehensible to humans. This is called as *natural motion*.

Naturalness in navigation has been the primary target in several studies. Althaus *et al*. present the results of their research on natural robot behavior in [80]. They developed methods for a robot to join a group of people engaged in a conversation. Their approach is based on regulating the speed of the robot as it approaches to the group and keep a certain distance to the closest person from the group using predefined behaviors such as *approach_human, keep_distance* etc. Their strategy is judged as a more natural way to interact with a group by the participants.

In [81], they note that in case of a conflicting situation in human-human encounters, people do not only take geometric and other physical properties into account while moving, but also reflect on the opponents intention. Thus, they presented the concept of *reflective navigation* for mobile robots. Imitation of the human behavior is achieved by treating humans as moving obstacles with intelligent decision-making capability. This competence of the humans is modeled and is introduced into the robot's capabilities for decision-making. Thereby, the robot employs the method of velocity obstacles [26], assuming that the people will follow the same rules.

Müller *et al*. [82] note that the ability to move in a way similar to the humans is an important factor for the social acceptance of the robot. The key idea in this research is that in highly crowded environments, people tend to follow the crowd. They developed a navigation framework to realize this idea on the robot. The path planner uses the A^* search method iteratively, after an appropriate leader to be followed is detected. The detection of the leader is done based on the navigational direction of the people in the environment [82].

In contrast to the model-based strategies to enhance the social acceptance of robot motion, Luber *et al.* uses statistical approaches to learn dynamic motion prototypes from the observed data [83]. They pose the problem as an unsupervised learning problem. The learned prototypes are, later, used to calculate a costmap which is used by the A^* search algorithm to generate the minimum cost path. Their approach is proven to create better estimates to the actual paths of people than pure proxemics implementations. On the other hand, the model based approaches are not limited to the proxemics theory. In several works, the researchers have successfully implemented navigation frameworks that imitate inner dynamics of human motion to enhance the comprehensibility of the motion of the robot using the *Social Force Model (SFM)* [84–89].

SFM was introduced by Helbing and Molnar. In [1], they expressed that pedestrians act according to virtual forces exerted by the other components in the environment. While the goal position of the navigation applies an attractive force towards itself, the obstacles and the other pedestrians exert repulsive forces. Thus, a combination of these so-called social forces controls the motions of the pedestrian. These forces are calculated by potential functions to represent the exponential nature of the attraction-repulsion phenomena. As in other applications of artificial potential functions, such as [23], a linear combination of the different types of potentials are used to calculate the overall potential. Regardless, SFM is a highly renowned strategy to describe the human motion and its application of the robot's navigation will help imitate the motion of the human and surely enhance the comprehensibility of the motions of the robot. The detailed investigation of this method will be given in Section 4.1.

2.5.3. Increasing Sociability

Some researchers believe that further social-acceptance can be achieved by making robots follow social conventions, and adhere to cultural norms. Walking on the right-hand side of the hallways, moving at a slower rate when the followed person approaches to a door entrance are examples from these studies. Standing in the line is another social convention. In [48], as well as carefully spacing itself from the personal zones of other people, the researcher achieve a sociability by following this humanly convention.

In [90], several social conventions, such as tending to one side of hallways, are described as constraints in the navigation framework of the robot. The authors of this paper believe that conventions should be handled at the top layer of the framework, in contrast to the majority of the applications that handle them in the local path planning layer. The constraints are therefore integrated with the actual cost function of an A^* search algorithm. The resulting motions of the robot tend to follow the defined conventions just as humans.

Pandey and Alami created a similar set of conventions for their research [91]. They define navigational social rules, such as moving on the right-hand-side of a narrow passage, passing a person from his left etc. According to these rules, the navigation task at hand is represented as a search problem of several milestones on a topological map. The connection between the milestones are originally the shortest path between two milestones. These paths are then connected together and processed through the B-spline interpolation technique for smoothening.

Several other studies can be found in the literature that modifying the map building process to represent spatial effects. In these studies, the heuristics used in path planning changed from shortest path to the most efficient path in terms of these effects. Such applications are found in [92, 93].

An extensive literature survey is given above. The current inclinations on the methods adopted to increase the overall social acceptance of the robots are given in detail. In the next two chapters, hardware and software platforms will be introduced and the methodology will be elaborated on.

3. PLATFORM SPECIFICATIONS

In order to reason about the design decisions made in the implementation phase, preferred software and hardware platforms will be introduced in this chapter.

In the *Software Platforms* section, the software tools used during the development of the system are introduced. As the centerpieces of this study, only the Robot Operating System [94] and the V-REP Robot Simulator [95] will be mentioned in Section 3.1. In Section 3.2, the hardware platform, Turgay will be introduced.

3.1. Software Platforms

3.1.1. Robot Operating System

Robot Operating System (ROS) is an open-source system, especially useful in the software development for robots. Although it provides many of the functionalities of an operating system, it is rather a framework or a middleware that runs on a host operating system. ROS presents diverse advantages in robot development. Due to these advantages listed below the navigation system in this study is developed using ROS.

- *Distributed computing*: Robot software is usually a huge and complex system. Various hardware units that are irrelevant to each other are controlled continuously. It mostly requires processing of different types of information from different types of sensors and yet, producing different types of actions, simultaneously. In cases of multiple computation units on the same robotic platform and multi-robot groups, these requirements are more apparent. Modular approaches are well suited in developing such systems. The concept of distributed computing refers to the operation of the components of a software on different processors, which are placed not necessarily on the same computation unit. ROS provides this functionality by its message-passing interface.
- *Support for several programming languages*: ROS has 3 main and several experimental

client libraries that enable developers to implement nodes and message-passing functionality in one of the supported languages; C++, Python and LISP or experimental ones; Java, Haskell, Go, Julia, R etc.

- *Package management and code reuse*: Standard packages for commonly used algorithms which are maintained through every ROS distribution saves users considerable amounts of time. In addition, drivers for the frequently used devices, such as RGB-D cameras and laser scanners, are also provided by ROS.

During development, many ROS packages have been used. In the following, these packages will be listed. The final item in this list includes the built-in message types used in this study.

- *gmapping*: This package is the ROS implementation of the GMapping simultaneous localization and mapping (SLAM) algorithm, described in [96]. It is used in this study to learn maps of the environments and save them as occupancy grids for later use.
- *amcl*: AMCL is the ROS implementation of the adaptive Monte Carlo Localization approach [97]. This technique is used to estimate the position of the robot in a known map. It also estimates the transformation between the global map and the robot's egocentric coordinate frame which is used to calculate the origin of the force frame.
- *tf*: A number of coordinate frames can be found in a typical navigation system. These frames move over time and this makes keeping track of them much harder. Also, we occasionally need to transform points and vectors from one frame to another. The *tf* package in ROS enables users to simplify these tasks by systematizing the operations among coordinate frames. Many coordinate frames exist in our system. Figure 3.1 illustrates these frames on a tree.
- *move_base*: This package provides the global path planner, which employs either *Dijkstra* or *A** search algorithms to find a path from the current position of the robot to the goal position in a given occupancy grid.
- *Built-in Message Types*: Many kinds of ROS messages are predefined as standard message types. These message types are useful in exchanging different types of information among different modules of the system. Below is the list of the predefined message

types used in this study:

(i) std_msgs/Int8

(ii) std_msgs/String

(iii) std_msgs/Header

(iv) geometry_msgs/Point

(v) geometry_msgs/PointStamped

(vi) geometry_msgs/PoseStamped

(vii) geometry_msgs/PoseWithCovarianceStamped

(viii) geometry_msgs/Twist

(ix) geometry_msgs/TwistStamped

(x) geometry_msgs/Quaternion

(xi) sensor_msgs/Joy

(xii) sensor_msgs/LaserScan

(xiii) nav_msgs/Odometry

(xiv) nav_msgs/OccupancyGrid

(xv) nav_msgs/Path

(xvi) visualization_msgs/Marker

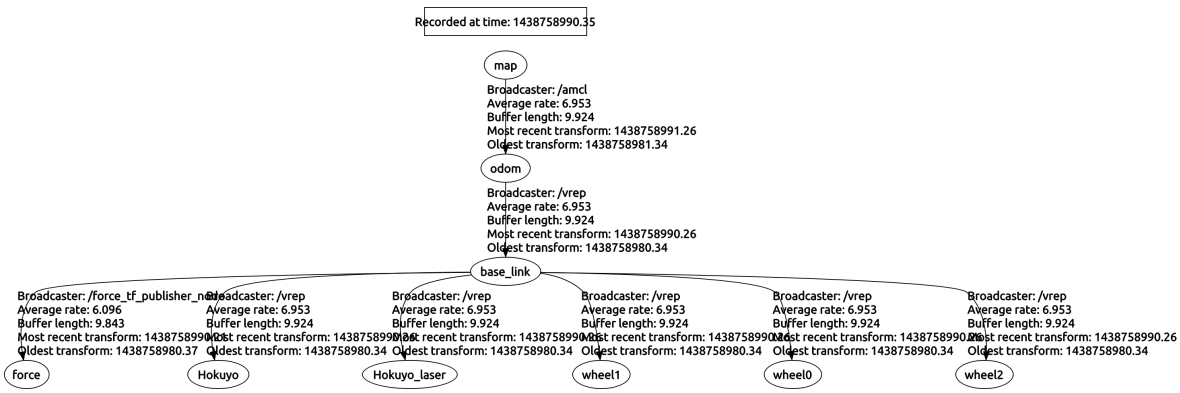


Figure 3.1. The coordinate frames in the navigation system.

3.1.2. V-REP Robot Simulator

V-REP is a robot simulator with an integrated development environment. It is able to use ODE [98], Bullet [99] and Vortex [100] physics engines. It has *regular* and *remote* APIs

with a ROS compatible interface that enables it to be controlled as a ROS node. Using the built-in ROS publishers and services, it allows fast development of the algorithms with ROS by letting the developer focus on the actual objective.

As noted above, V-REP provides many functionalities through its APIs and integration of them with ROS through publishers, subscribers and services. All can be invoked using scripts written in Lua language. Below is the list of API functions that are commonly used in this study:

- *simGetObjectHandle()*: Returns the *handle* of the object for later use.
- *simGetObjectPosition()*: Returns the position of the object.
- *simSetObjectPosition()*: Sets the position of an object in x, y and z coordinates.
- *simSetIntegerSignal()*: Sets the given integer as the value of an integer signal which is to be published later.
- *simGetStringSignal()*: Returns the value of a string signal.

In addition to these, below are two of the ROS API functions, which are used to publish the information to other ROS modules and subscribe to the topics to which other modules publish the information needed by the simulator.

- *simExtROS_enablePublisher()*: Enables a publisher.
- *simExtROS_enableSubscriber()*: Enables a subscriber.

3.2. The Hardware Platform

The hardware platform for the application of the navigation system is the Turgay robot. Turgay is an indoor mobile robot equipped with multiple computational units and developed by the Robotics Group of Boğaziçi University [101]. Turgay is a member of the Multi-Robot Tour-Guide System, which is a heterogeneous group of robots that works collaboratively to realize tour-guiding tasks at museums. The project is introduced in [102]. In the context of this study, Turgay is equipped with the following hardware:

- A laptop with 1.5 GHz. Intel Centrino processor and 4 GB RAM to carry out the processing of the large data
- 2 Raspberry Pi Model B computers [103] with 700 MHz. ARMv6 processors, Videocore 4 GPUs and 512 MB RAMs to host the ROS master which commands all devices on the platform and carry out network connectivity tasks.
- A wireless access point to enable connectivity to wireless networks.
- A Hokuyo URG-04LX laser scanner [104] to detect the obstacles around. URG-04LX has a range of 4095 mm, and a field of view of 240 degrees.
- An ASUS Xtion PRO LIVE RGB-D camera [105] to detect the people around.

Moreover, Turgay has a Festo Robotino robot [106] in its base to achieve mobility. Robotino has swedish wheels that serves the purpose of moving omnidirectionally. It also carries a tablet for the purpose of interacting with the people. Turgay is given in Figure 3.2.



Figure 3.2. The robotic platform *TURGAY*.

4. METHODOLOGY AND IMPLEMENTATION

This chapter of the thesis is dedicated to explain the preferred technique in much more detail and its implementation as a program that runs on the actual robotic platform. Although the model was touched on rather lightly in the Section 2.5, a more thorough explanation of it, including the mathematical foundations it is built atop, is given in Section 4.1. Subsequently, the implementation details are explained in Section 4.2.

4.1. Methodology

As mentioned earlier, SFM was designed to model the motion of the pedestrians. In addition, the application of SFM to robot navigation increases the naturalness of the motion of the robot. Implementing the model as a navigation system requires the mathematical descriptions of the so-called *social forces*. These are given in Section 4.1.1. Furthermore, the suitability of the model for local path planning concept is analyzed in Section 4.1.2.

4.1.1. Mathematical Description of the Social Force Model

In a free field, people are inclined to move towards their destination target by taking the shortest path. When obstacles are placed on this path, they alter their paths slightly to achieve a collision-free navigation. In addition to the obstacles, if there are people around, additional alterations are made. These phenomena are studied in [107].

The desire that drives the pedestrians towards their navigational goals is expressed as the *attractive force* in this model. The straight path starting from the current position of the pedestrian and ending at the navigational goal determines the *desired direction*, $\hat{e}$. The desired direction of the motion of pedestrian i is therefore, defined as in Equation 4.1.

$$\hat{e}_i(t) = \frac{\vec{r}_i^{\,0} - \vec{r}_i(t)}{\| \vec{r}_i^{\,0} - \vec{r}_i(t) \|} \,, \tag{4.1}$$

where $\vec{r}_i^{\,0}$ represents the position of the navigation target and $\vec{r}_i(t)$ represents the current

position of the pedestrian. Unless the motion is disturbed by the obstacles or other individuals, then the pedestrian i will move in the direction of $\hat{e}_i(t)$ with a *desired speed* v_i^0. Thus, the *desired velocity* is defined as in Equation 4.2.

$$\vec{v}_i^0(t) = v_i^0 \hat{e}_i(t) \tag{4.2}$$

Deviations from the velocity given in Equation 4.2 are caused by the obstacles and other pedestrians. On the other hand, the pedestrian is always inclined to reach the desired velocity from his or her current velocity, $\mathbf{v}_i(t)$, within a time interval, called the *relaxation time*, τ. These deviations can be reflected as an acceleration term; so the attractive force that pulls pedestrian i towards his or her navigational target can be described as in Equation 4.3.

$$\vec{f}_i^{attr}(t) = m_i \frac{1}{\tau}(v_i^0 \vec{e}_i(t) - \vec{v}_i(t)) \tag{4.3}$$

Note that the description of the attractive force complies with the Newton's laws of motion. In this study, we take m_i identical for all pedestrians which is the general tendency in most of the SFM applications, such as [107], [108], [109] and [110].

The other force affecting a pedestrian's motion is the repulsive one. It comprises the influences which are induced by other people and the obstacles in the environment. According to the model, deviations from the shortest path are the result of this type of forces, which are called *repulsive forces* within this context . According to [109] and [111], and assuming the circular specification, introduced in [112], repulsive forces are described in Equation 4.4.

$$\vec{f}_i^{rep} = \sum_{j \in Q,\ j \neq i}^{Q} A_q e^{-d_{ij}/B_q} \hat{n}_{ij} \ , \tag{4.4}$$

where i represents the pedestrian, j represents the entity, q represents the type of the entity, that is, $q \in \{obstacle, pedestrian\}$. $Q \equiv P \cup O$, such that P represents the set of pedestrians and O represents the set of obstacles, A_q is the intensity of the force in Newtons, B_q is the range of the social interaction in meters, d_{ij} is the minimum distance between i and j and $\hat{n}_{ij}$

is the unit vector pointing from i to j.

The values of A_q and B_q changes with respect to the type of the entity. Different range and intensity values are adopted for the obstacles and pedestrians; that is what distinguishes SFM from PF approach. Therefore, the cumulative repulsion exerted by the obstacles and the pedestrians are calculated separately. As explained in Section 4.2.3, a linear combination of these two types of repulsive forces is used to obtain the cumulative repulsive force, as given in Equation 4.5.

$$\vec{f}_i^{rep} = \vec{f}_{i,O}^{rep} + \vec{f}_{i,P}^{rep} \tag{4.5}$$

The cumulative force that governs the motion of the pedestrians is the superposition of the attractive and repulsive forces. The changes in velocity at time t is determined by this force. These are described in Equations 4.6 and 4.7, respectively.

$$\vec{F}_i(t) = \vec{f}_i^{attr}(t) + \vec{f}_i^{rep}(t) \tag{4.6}$$

$$m_i \frac{\mathrm{d}\vec{v}_i(t)}{\mathrm{d}t} = \vec{F}_i(t) \tag{4.7}$$

4.1.2. Social Force Model as a Local Planner for Robot Navigation

Navigation and mapping are two of the most important concepts in robot motion. The former is described as the problem of finding a collision-free motion for the robot, whereas the latter is described as exploring and learning the unknown environment so that this information may be utilized in navigation in [113].

Because SFM is a member of *artificial potential function* approaches, using it as the sole path planner of a navigation system introduces the risk of getting stuck at local minima;

an example is shown in Figure 4.1. To avoid encountering such cases, the simplest and most widely employed practice is to apply a hybrid path planner. An earlier example of the application of this type of planner is mentioned in [114]. The idea is to combine the efficiency of a global planner with the reflexivity of a local planner. Similarly, in [113] the suggested solution to elude local minima problems is also the use of *potential function* approaches as a guide to *search-based* global path planners. This explains the reason of the implementation of SFM as a local path planner.

As a derivative of artificial potential field approaches, SFM is an *online* planner that updates the potential field according to the position updates of the robot and the people around.

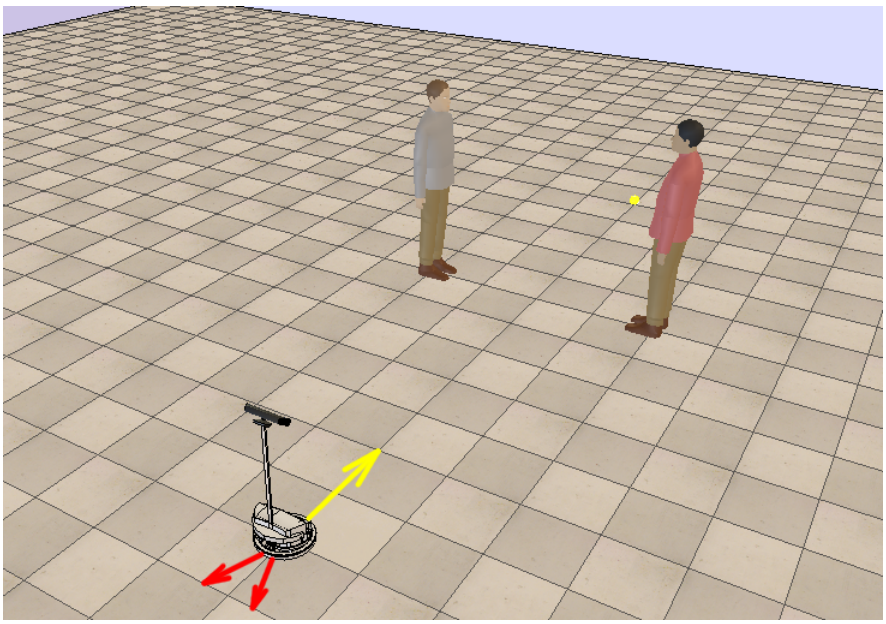


Figure 4.1. In a local minimum, the sum of the attractive and repulsive forces is zero.

In order to use graph search strategies to find a path from the current position to the target position, first the environment has to be represented as a graph. The occupancy grid representation of the environment is a convenient and widely used technique in mapping. This technique meets the forementioned requirements, as explained in [115].

A considerable amount of research has been done in the domain of mapping. Since

the problem is not in the scope of this study, a well renowned approach is directly adopted to learn the maps of the environments. *GMapping* is a SLAM algorithm and explained in Section 3.1.1. An example map is given in Figure 4.2. Moreover, since the maps built using this technique represent the environment as occupancy grids, graph search strategies, such as A^*, can be employed to create global paths which, in turn, are given to the local planner that works with the global planner in cooperation.



Figure 4.2. An example grid map built using GMapping technique explained in [96].

4.2. Implementation

When developing a navigation system, implementing directly on robots is not a reasonable way; in that it is a very time-consuming task to test each aspect of the whole system. More importantly, it is quite common to encounter errors which might lead to collisions during the testing phase. These collisions carry the risk of harming the hardware units on the platform. On the other hand, there exists an ever-increasing number of robot simulators with built-in physics engines in order for researchers to test the systems they develop. Use of such simulators expedites the programming and testing phases and may generate good starting points for optimization processes which are typically needed while developing a navigation system. In this study, the algorithm is tested using V-REP [95] robot simulator. The simulator is explained in Section 3.1.2, as well as the factors which are contributed in the selection of this simulator.

In the following section, the modules that altogether constitute the human-aware navigation behavior are described in detail.

4.2.1. Infrastructure of the Local Path Planner

In order to put SFM into practice, as a local path planner, four primary and several auxiliary modules were created. A modular design approach was adopted so that the navigation system could benefit from the advantages of distributed computing as explained in detail in Section 3.1.1. The forementioned primary modules can be characterized as follows:

- *Robot Tracker*: This module is responsible for providing the position information of the robot to the navigation system. The position information is delivered to the module either by the simulator or the localization node. The module, then, applies the necessary operations and outputs a standardized position information.
- *People Tracker*: This module provides the position information of the people in the environment in which the robot is navigating. After a vector of positions of the people in the environment is provided to this module, it applies the necessary transformations and produces a standardized vector of people positions.
- *Obstacle Tracker*: Since the robot is navigating in an environment of which we have already had a previously-built map; this module continually provides the information about the positions of the obstacles that are located in the vicinity of the robot. It wields the position information provided by the *Robot Tracker* to search for the obstacles around. It outputs a vector of obstacle positions in a standardized manner.
- *Social Navigation Unit*: Using the information provided by other modules, this one implements SFM approach to generate the velocity commands that will be sent to the robot.

The overall infrastructure – including the topic names, message types and contents of the messages which are exchanged among the primary modules – is depicted in Figure 4.3.

A more elaborate explanation on the implementation details for each primary module together with the algorithms used is given in the next two sections, as well as the descriptions

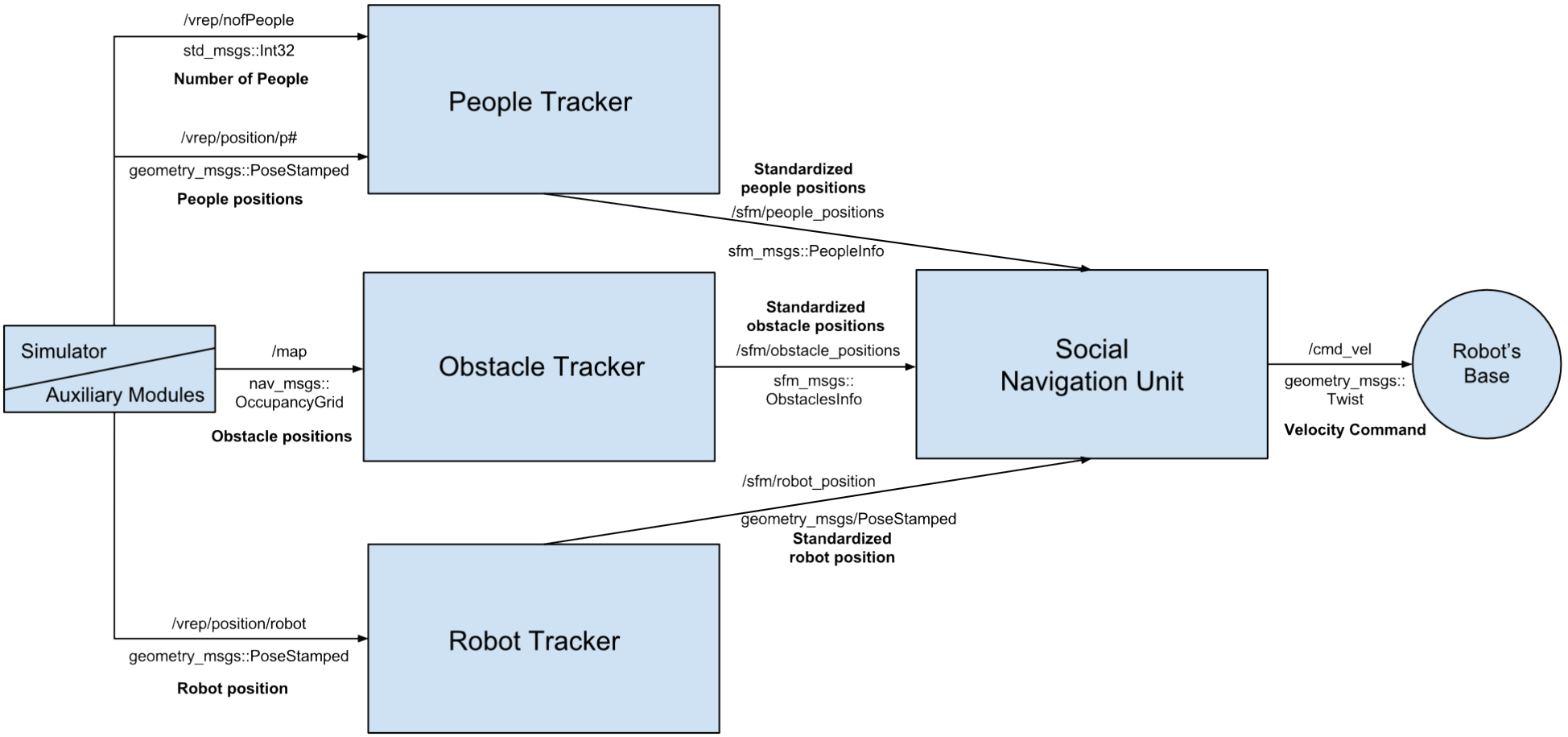


Figure 4.3. The overall infrastructure of the local path planner.

and explanations of the auxiliary ones.

4.2.2. Implementation of the Local Path Planner in the Simulated Environment

As described in the previous section, 4 primary modules play active roles in the implementation of SFM as a local path planner. In the following, each will be explained thoroughly.

4.2.2.1. Robot Tracker. Estimating the robot's position on a given map is a crucial part in almost all navigation tasks. As this study does not aim to solve the localization problem directly, we preferred to use one of the several readily available techniques, such as Monte Carlo Localization (MCL). In the case of the simulated environment, however, accessing the position of the robot is straightforward. As explained in Section 3.1.2, use of the corresponding API command did suffice to retrieve the position of the robot. Getting the position of the robot with respect to the environment, and publishing this information to a topic that the *Robot Tracker* module continuously listens is therefore done on the simulator side with a Lua script.

A publication to the topic which is listened by the *Robot Tracker* triggers the following callback function:

(i) *robotPoseCallback()*: Position updates for the robot which was published by the simulator or another auxiliary module, in the case of running the system on a real robot, are received via this function. The updated position is given to this method as a *geometry_msgs/PoseStamped* message.

Robot Tracker, after receiving the position updates, stamps the information with current time in nanoseconds and publishes the stamped information on another topic which is listened by the *Social Navigation Unit*.

4.2.2.2. People Tracker. To be able to treat people differently than the robot does obstacles, the algorithm must be provided with the position information of the people in the environment. In the case of the simulated environment, each person is an object just as the robot is. Therefore, the same command, that is used to get the position of the robot, is used for each person. As well as publishing the position updates for each person, the simulator also publishes the number of the people in the environment, so that *People Tracker* module knows how many topics to listen for position updates. Again, these publications are done on the simulator side by a Lua script.

The *People Tracker* receives the data via its callback functions. These functions are described below:

(i) *peopleCountCallback()*: The number of the people in the scene is published to a topic by the simulator or another auxiliary module, in the case of running the system on a real robot. *People Tracker* module gathers this information via this method. The data is given in the form of a *std_msgs/Int32* message.

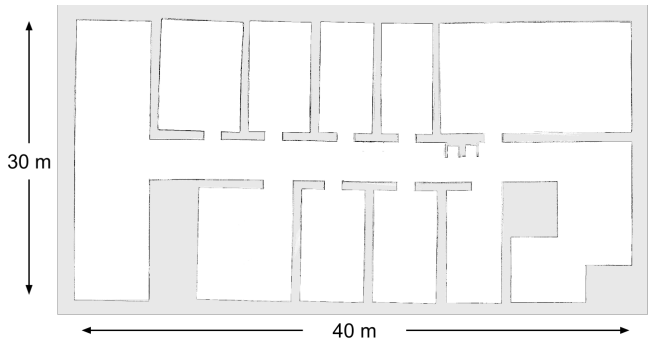
(ii) *poseCallback()*: Position updates for the people in the environment which was published by the simulator or another auxiliary module, in the case of running the system on a real robot, are received via this function. The updated position is given to the function as a *geometry_msgs/PoseStamped* message. The method also takes an integer parameter as the index of the person whose position information is given.

The *People Tracker*, just as the *Robot Tracker*, stamps the information it receives. Using a *sfm_msgs/PeopleInfo* message, passes the stamped information to the *Social Navigation Unit*.

4.2.2.3. Obstacle Tracker. Using a previously-learned map of the environment, this module informs the *Social Navigation Unit* about the static obstacles around the robot at that instant.

A typical map is shown in Figure 4.2. To elaborate more on some of the design decisions on the implementation of this module we need a closer look at the maps.

Another map and its metadata are given in Figure 4.4. As can be seen from the metadata, the resolution of this map in the example is *0.01 meters*.



(a) An example map, saved in Portable Gray Map (PGM) format.

```
image: last.pgm
resolution: 0.010000
origin: [-10.000000, -10.000000, 0.000000]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.196
```

(b) Metadata of the map in Figure 4.4(a).

Figure 4.4. A map and it's metadata.

The lower the resolution of the map, the higher the number of cells it contains. The number of the cells in the example can be calculated as follows:

$$Number\,of\,Cells = (40m \times 30m) \div (0.01m \times 0.01m)$$
$$= 12000000$$

In a typical map, such as this one, there usually exist more than 10^7 cells. Therefore, detecting the obstacles by visiting all cells becomes an unreasonable solution.

Since SFM was initially designed to model the motion of the pedestrians, the behavior of the people can be imitated to limit the obstacles that would be taken into consideration while calculating their effects. In [116], Kitazawa and Fujiyama investigates the notion of Information Process Space (IPS). Briefly, IPS describes the space within which a pedestrian is inclined to consider each objects as potential obstacles. The IPS of humans is limited to *4.5 meters* on average. Therefore, the smallest circle on 2D map that comprises the average IPS of the people is adopted as a reference, which is called throughout this study as Region of Interest (ROI). Using ROI, the number of map cells to be taken into consideration can be

decreased substantially:

$$Number\,of\,Cells = (\pi \times 4.5m \times 4.5m) \div (0.01m \times 0.01m)$$
$$\approx 635000$$

Despite this simplification, the approach still has weaknesses in terms of time complexity. Finding the cells included in the circle of radius 4.5 meters with the center at the robot's current position brings a massive overhead. Also, with lower resolutions (much detailed maps), the number of cells to be evaluated increase quadratically, as shown in Figure 4.5.

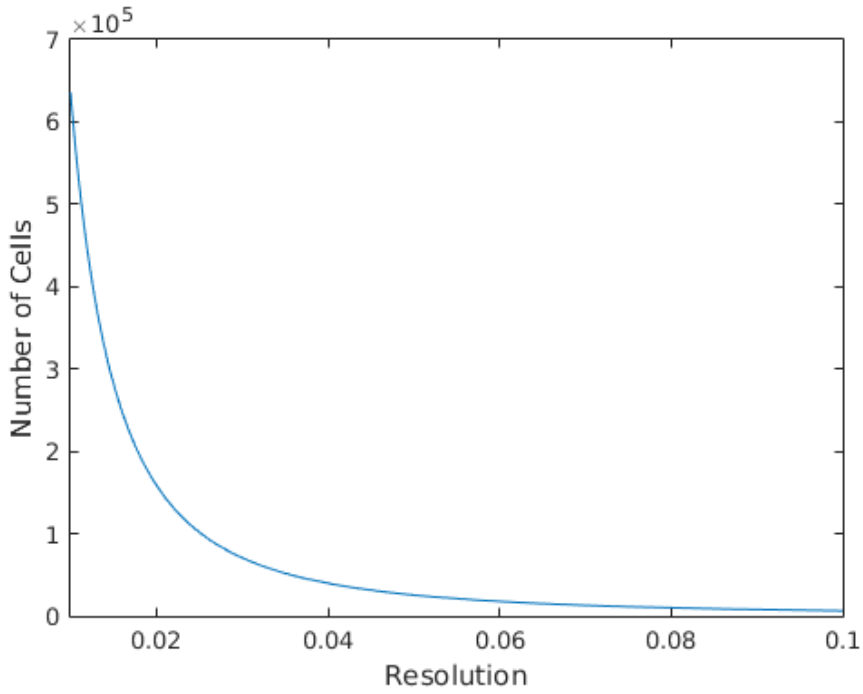


Figure 4.5. The relation between resolution and *Number of Cells* within *ROI*.

At this point, it is beneficial to mention another significant characteristic of the *Obstacle Tracker*. While the robot moves, this module is expected to continuously look at the cells on the map and report the occupied ones to the *Social Navigation Unit*. In addition, conducted experiments on TURGAY suggest that, it had to be provided with velocity commands

in – at least – every 0.143 seconds to achieve smooth movement.

To ensure this frequency regardless of the resolution of the map and decrease time complexity, 2D Ray-Tracing approach has adopted. The implemented algorithm is given in Figure 4.6.

Worst-case time complexity of this algorithm is $O(nlogn)$ which, in the case of the map example given above, evaluates to a much smaller number of cells to be evaluated, calculated as follows:

$$\begin{aligned} Number\,of\,Cells &= (Number\ of\ Rays) * [(4.5m \div increment) + log_2 increment] \\ &\approx 6000, \end{aligned}$$

where *Number of Rays = 360* and *increment = 0.4 meters*. These numbers are selected as a result of a series of experiments. Increasing the number of rays or decreasing the increment amount lead to more precise results when detecting obstacles, however; experiments had shown that the small amount of lack in the precision could safely be neglected in favor of decreasing the running time, since the continuous evaluation of the cells suffices to detect obstacles smaller than the size of the increment.

Further speed up was achieved by the application of $fastHypot()$ method, [117]. An estimate to the true distance between two points is deduced by this method, replacing expensive square root calculations at the cost of a small lack in the precision. Again, experiments had shown that this lack in the precision could safely be neglected. The $fastHypot()$ method is given in Figure 4.7. Thanks to the use of these methods to accelerate the obstacle detection process, as the bottleneck of the entire system, *Obstacle Tracker* module is guaranteed to work at least at 10 Hz. on the Turgay platform. The region of interest and the output of the module are illustrated in Figure 4.8.

4.2.2.4. Social Navigation Unit. This module can be regarded as the brain of the entire system. All the information which is gathered on the other modules is transmitted to this

Algorithm $RayTrace(start_x,\ start_y,\ end_x,\ end_y,\ increment)$
if increment $\leq$ maxError **then** ▷ Termination criterion
    $occupiedCells.add(Cell(end_x,\ end_y))$
    return
end if
$d_x \leftarrow abs(end_x start_x) \times mapResolution$ ▷ Distance(start,end) in meters
$d_y \leftarrow abs(end_y - start_y) \times mapResolution$
$d_x Cell \leftarrow end_x - start_x$ ▷ Distance(start,end) in cells
$d_y Cell \leftarrow end_y - start_y$
$d_h \leftarrow fastHypot(d_x,\ d_y)$ ▷ Length of hypotenuse in meters
$n \leftarrow ceiling(d_h \div increment)$ ▷ Number of iterations
$inc_x \leftarrow d_x Cell \div n$ ▷ Increment amount
$inc_y \leftarrow d_y Cell \div n$
$x \leftarrow start_x + inc_x$
$y \leftarrow start_y + inc_y$
for $j \leftarrow n, \ldots, 0$ **do**
    $occupancy \leftarrow visit(x, y)$
    if $occupancy = 0$ **then** ▷ Free cell
        $x \leftarrow x + inc_x$
        $y \leftarrow y + inc_y$
    else if $occupancy = -1$ **then** ▷ Unknown cell
        $raytrace((x - inc_x),\ (y - inc_y),\ x,\ y,\ increment \div 2)$;
        break
    else ▷ Occupied cell
        $occupiedCells.add(Cell(x,\ y))$
        break
    end if
end for

Figure 4.6. 2D Ray-Tracing algorithm.

Algorithm $FastHypot(a,\ b)$

$min \leftarrow a$

$max \leftarrow b$

if b $<$ a **then**

    $min \leftarrow b$

    $max \leftarrow a$

end if

$hypot \leftarrow (\sqrt{2}-1) \times min + max$

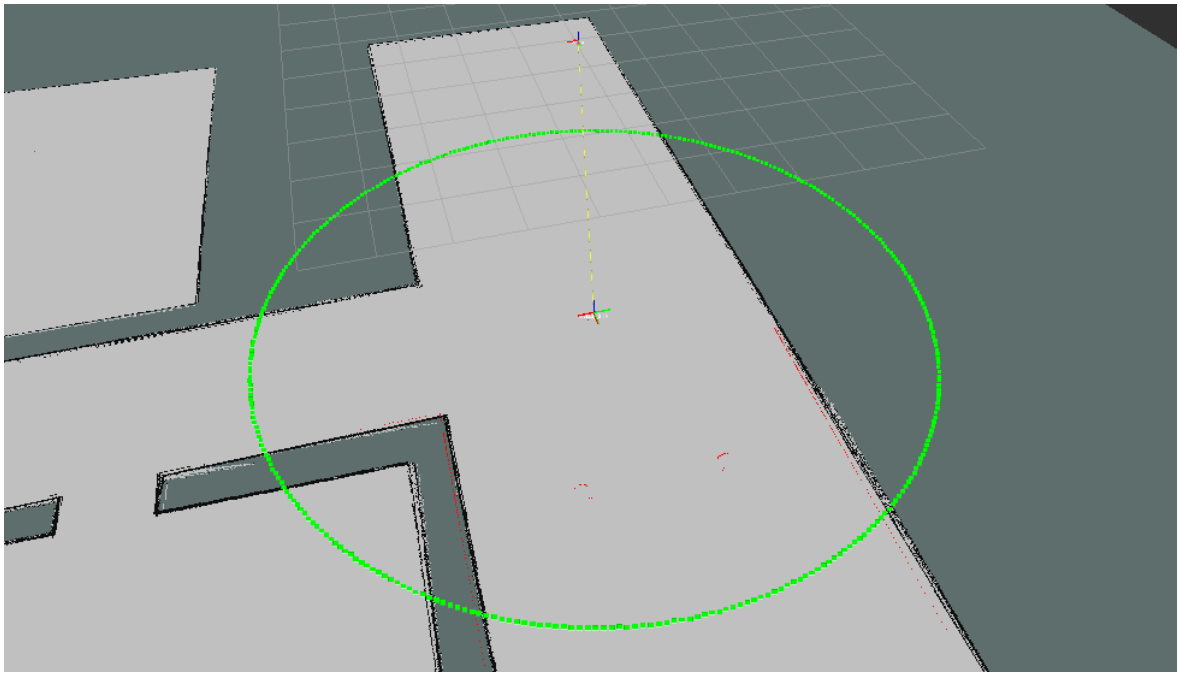
return $hypot$

Figure 4.7. Fast hypotenuse algorithm.

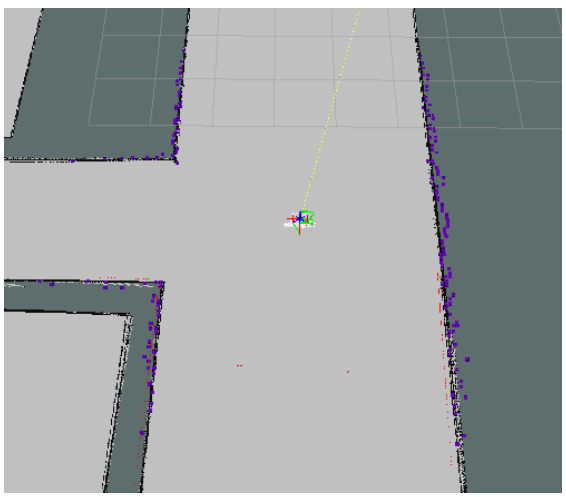
one. Using these data, the *Social Navigation Unit* first calculates the social forces which are explained in Section 4.1.1. Afterwards, with respect to the combination of these forces, the module generates a velocity command to be executed by the robot itself.

First, the callback functions of the module, with which it collects the information from other modules, are described below. There are four primary and two auxiliary callback methods implemented within this module, as listed below:

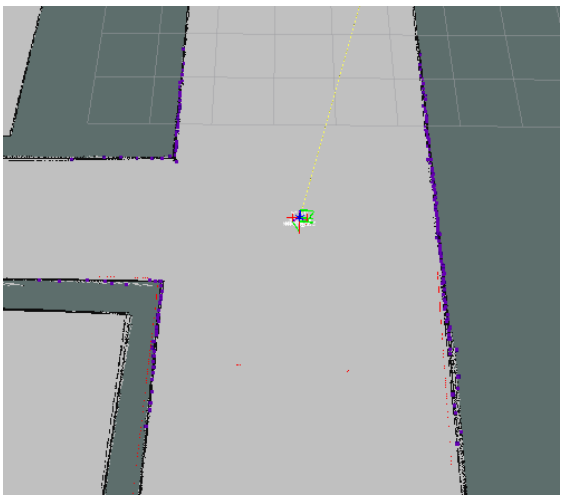
- Primary Callback Functions:
  (i) *planCallback()*: If the global path planner ever creates a new plan for any reason, the module is notified via this method. The function takes the new plan as a message of type *nav_msgs/Path*.
  (ii) *peopleLocationsCallback()*: Position updates for the people in the environment which was published by the *People Tracker* module are received via this function. The information is given to this function as a *sfm_msgs/PeopleInfo* message.
  (iii) *obstacleLocationsCallback()*: Positions of the obstacles in the ROI are reported to this module via this function. Obstacle Tracker transmits a message of type *sfm_msgs/ObstaclesInfo* to this module by giving the message as a parameter to this method.
  (iv) *robotLocationCallback()*: Position updates for the robot are produced by *Robot*



(a) Region of Interest



(b) Result for maxError = 0.1 meters



(c) Result for maxError = 0.02 meters

Figure 4.8. Operation of the *Obstacle Tracker* module.

Algorithm $CalculateAttractiveForce()$

▷ Implementation of Equation 4.3

$desiredDirection \leftarrow UnitVector2D(robotPose, nextSubgoal)$

$desiredVelocity \leftarrow desiredDirection \times desiredSpeed$

$actualVelocity \leftarrow Vector2D(lastVelInGlobal.x,\ lastVelInGlobal.y)$

$f_{attractive} \leftarrow (desiredVelocity - actualVelocity) \times relaxationTime$

return $f_{attractive}$

Figure 4.9. Calculation of the attractive force.

Tracker module, then given to the Social Navigation Unit via this method as a message of type *geometry_msgs/PoseStamped*.

- Auxiliary Callback Functions:
  - (i) *globalGoalCallback()*: For the sake of the simplicity of the assignment of a new navigation goal, this method is implemented. It takes a *geometry_msgs/Point* message as parameter and conducts the necessary operations to assign the position as the new target to the module.
  - (ii) *velCallback()*: Since parameters-learning part requires listening to the velocity commands given by the human operator, this function is created. It subscribes to the topic, on which the velocity commands are given to the robot, and records the messages. Additionally, transformation that are needed to convert the velocity commands from the robot's own coordinate frame into global coordinate frame are made in this method. It takes the commands as a message of type *geometry_msgs/Twist*.

Above explanations divulge the information-gathering procedures of *Social Navigation Unit*. Using this information, the module calculates *social forces*, using the algorithms detailed in Figures 4.9, 4.10, 4.11 and 4.12. In addition, the calculated social forces are depicted in Figure 4.13.

Algorithm $CalculatePeopleRepulsiveForce()$

$f_{repulsivePeople} \leftarrow Vector2D(0,0)$

for each position $p \in peoplePositions$ **do**

$f_{repulsivePeople} \leftarrow f_{repulsivePeople} + CalculateRepulsiveForce(p,\ TRUE)$

end for

return $f_{repulsivePeople}$

Figure 4.10. Repulsive force generated by humans.

Algorithm $CalculateObstacleRepulsiveForce()$

$f_{repulsiveObs} \leftarrow Vector2D(0,0)$

for each position $p \in obstaclePositions$ **do**

$f_{repulsiveObs} \leftarrow f_{repulsiveObs} + CalculateRepulsiveForce(p,\ FALSE)$

end for

return $f_{repulsiveObs}$

Figure 4.11. Repulsive force generated by obstacles.

Algorithm $CalculateRepulsiveForce(position, type)$

if $type = TRUE$ **then** ▷ Person

$intensity \leftarrow personIntensity$

$range \leftarrow personRange$

else ▷ Obstacle

$intensity \leftarrow obstacleIntensity$

$range \leftarrow obstacleRange$

end if

▷ Implementation of Equation 4.4

$direction \leftarrow Vector2D(position,\ robotPose)$

$minDistance \leftarrow direction.magnitude$

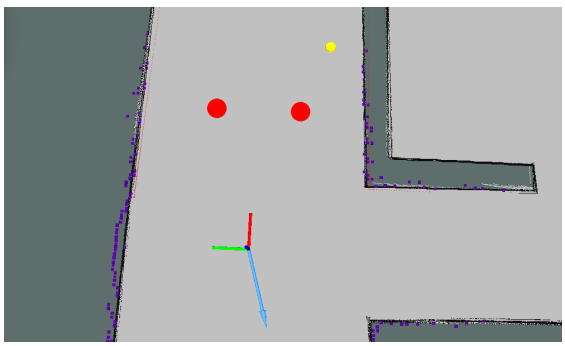
$forceDirection \leftarrow UnitVector2D(position,\ robotPose)$

$forceScale \leftarrow intensity \times exp(\frac{-minDistance}{range})$

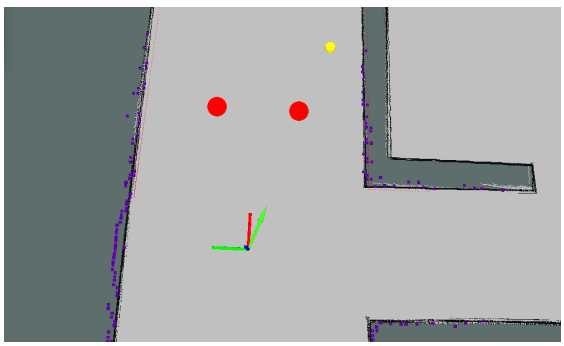
$result \leftarrow forceDirection \times forceScale$

return $result$

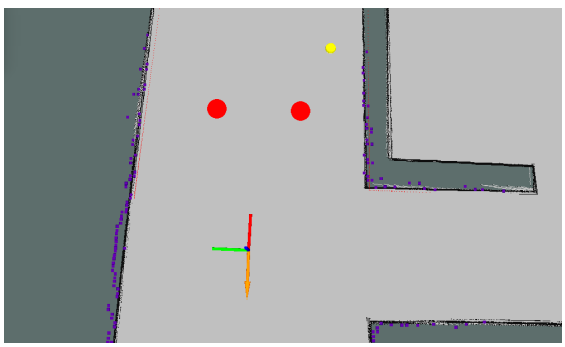
Figure 4.12. Calculation of the repulsive force.



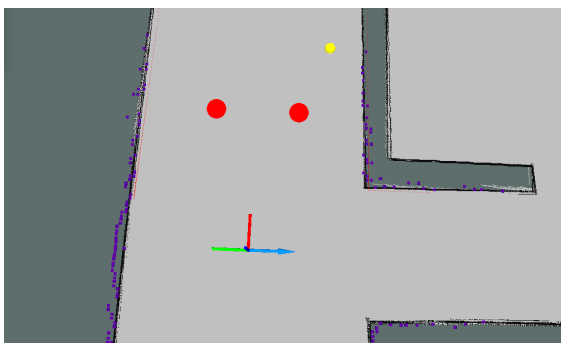
(a) Cumulative Force: Weighted Sum of Social Forces



(b) Attractive Force Exerted by the Navigation Target

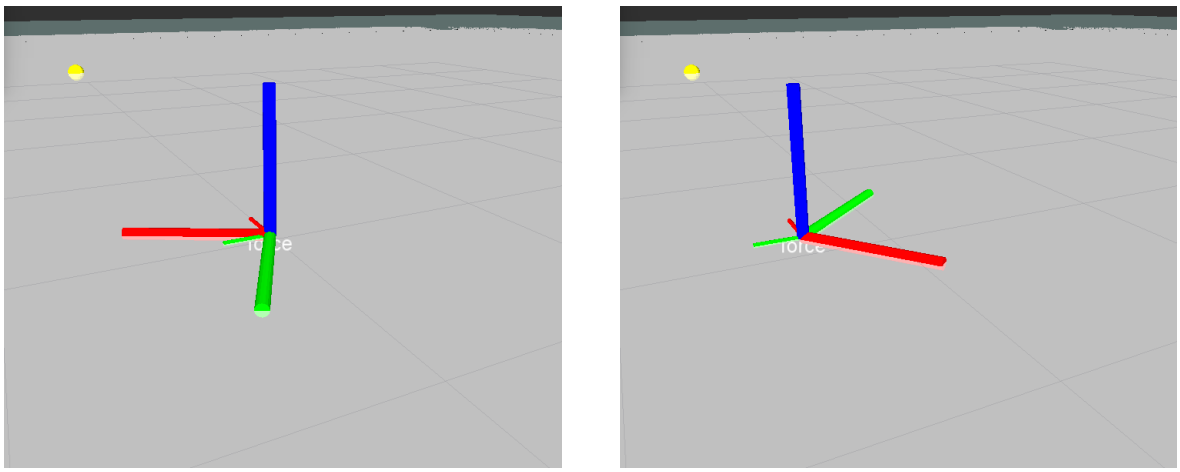


(c) Repulsive Force Exerted by People



(d) Repulsive Force Exerted by Obstacles

Figure 4.13. Calculated Social Forces. Red dots represent the people in the environment and yellow dot represents the navigation target.



(a) (b)

Figure 4.14. *Force* frame has a movable origin and a fixed orientation.

4.2.2.5. Auxiliary Modules. During the course of this study, in addition to the primary modules, several auxiliary modules have been implemented. This section is devoted to the explanations of these modules.

- *Force Transform Publisher*: As explained in Section 3.1.1, the tf package is extensively used throughout this work. One of the most significant uses of it is the creation of the *force* coordinate frame. The following reasons obliged us to create the *force* frame:
  (i) *Robot Tracker* and *People Tracker* modules report positions on the global coordinate frame. On the other hand, *Obstacle Tracker* operates in the robot's egocentric coordinate frame.
  (ii) Vector operations to calculate forces must be conducted on the same frame.
  (iii) Velocity commands which are calculated on some frame must be transformed into the robot's egocentric frame.

  These issues have been simplified, thanks to the creation of the *force* frame. The frame, shown in Figure 4.14, has a fixed orientation, the same as the global orientation, with its origin centered at the robot's current position.
- *Data Collector*: In order to examine the effects of current forces on the next velocity command, this module is implemented. It subscribes to the topics on which calculated force values and velocity commands are published to the robot and record these values by grouping them according to their time stamp. The generated data can later be

exploited in parameters learning phase.

- *sfm_msgs*: This module is created to define application-specific message types. These messages are used by the modules of the system to exchange information alongside with the built-in message types of ROS, which are given in Section 3.1.1. Newly defined message types and their contents are given in the following.

  (i) *sfm_msgs/PeopleInfo*:

  geometry_msgs/PoseStamped[ ] poses
  std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
  geometry_msgs/Pose pose
  geometry_msgs/Point position
  float64 x
  float64 y
  float64 z
  geometry_msgs/Quaternion orientation
  float64 x
  float64 y
  float64 z
  float64 w

  (ii) *sfm_msgs/ObstaclesInfo*:

  geometry_msgs/PointStamped[] poses
  std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
  geometry_msgs/Point point
  float64 x
  float64 y
  float64 z

(iii) *sfm_msgs/Vector2D*:

float64 x

float64 y

(iv) *sfm_msgs/Vector2DStamped*:

std_msgs/Header header

uint32 seq

time stamp

string frame_id

sfm_msgs/Vector2D vector

float64 x

float64 y

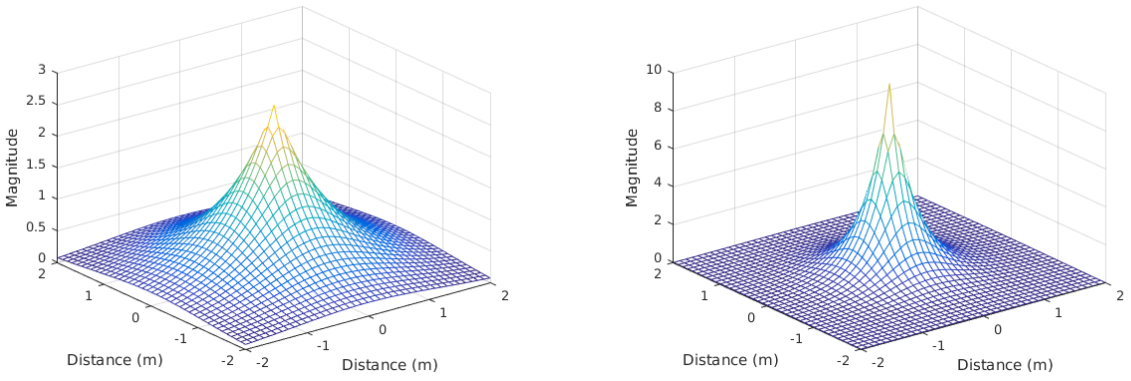
4.2.3. Parameters Learning

The application of social forces to robot navigation aims to increase the comprehensibility of the motion of the robots by imitating the inner dynamics of human motion. All of the calculated forces do contribute to the resulting velocity command in the same way; the cumulative force is a linear combination of them, as they have the same physical units in terms of Newtonian mechanics. However, their magnitude may differ largely from one application to another according to the changes in the representation preferences of the people, robot and, mostly, the obstacles.

Since the obstacles are represented as map cells in this study, the repulsive force stemming from them has a lot of addends, in contrast to that of the repulsive force stemming from the people. This situation poses a considerable difference in the magnitudes of the two repulsive force components. Magnitudes of the repulsive forces generated by a human and an occupancy grid cell are given in Figure 4.15. In addition, in Figure 4.16, the comparison of the cumulative magnitudes of these two types of repulsive forces over 12 trials on simulated environment is given. This situation reveals the need for a weighted sum of the calculated

forces as given in Equation 4.8.

$$\vec{F}^{cumulative} = \alpha \ \vec{f}^{attr} + \beta \ \vec{f}^{oRep} + \gamma \ \vec{f}^{pRep} \tag{4.8}$$



(a) People-generated potential.

(b) Obstacle-generated potential.

Figure 4.15. The comparison of people-generated and obstacle-generated potentials.

Inferring the weights which are going to be applied to each type of force may be done by the application of an appropriate approach from the field of artificial intelligence. In [109] the author mentions the use of sampling methods to learn the force weights. In [110] they use Markov Chain Monte Carlo Metropolis-Hasting algorithm to find the best weights by minimizing what they call the *social work* metric, which is basically the summation of the individual works done by the robot and each person in the scene.

In this study, supervised learning approach is taken to infer the weight coefficients of forces [118]. Considering the continuously linear nature of the problem at hand and real-valued output velocities, multiple linear regression may rationally be expected to succeed in modeling the relation between the velocity commands and calculated forces. In the first set of learning experiments, 30 real life scenarios were simulated and human operators were asked to move the robot past people and obstacles towards the target. Since the velocity commands which would be generated by *Social Navigation Unit* would be proportional not with the momentary cumulative but the integral of the force, only the initial velocity commands given

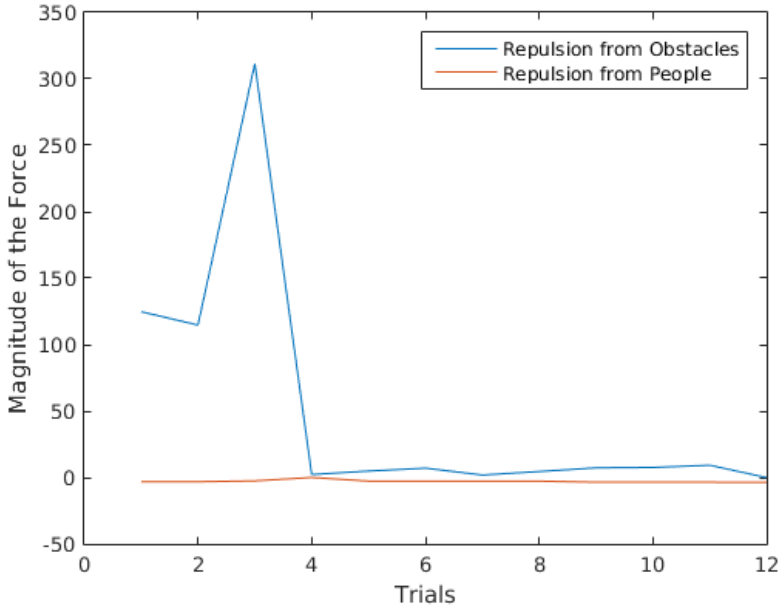


Figure 4.16. The comparison of the cumulative magnitudes of people-generated and obstacle-generated repulsive forces.

by the operators were recorded; because, the momentary cumulative force equals the integral of the force at the first step.

Equation 4.9 is used to form the data set which is used as the training data.

$$\mathscr{X} = \{\vec{x}^n, r^n\}_{n=1}^{N}, \tag{4.9}$$

where $\vec{x}^n$ denotes a 3-dimensional vector of either x or y component of the calculated force values from the same experiment and r^n is the corresponding component of the given velocity command. Thus, one may expect the Equation 4.10 to hold.

$$r^n = f(\vec{x}^n) + \varepsilon, \tag{4.10}$$

where ε represents the noise. The aim in a typical regression problem is to find a hypothesis $g(.)$ that minimizes the aggregate error between $g(\vec{x}^n)$ and r^n. This error is expressed in

Equation 4.11.

$$E(g|\mathscr{X}) = \frac{1}{N}\sum_{n=1}^{N}[g(\vec{x}^n) - r^n)]^2 \tag{4.11}$$

Under the assumption of the linear hypothesis, we have

$$g(\vec{x}) = \sum_{j=1}^{3} w_j x_j = \vec{w}^T \vec{x}. \tag{4.12}$$

In our case, $\vec{w} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$ and $\vec{x} = \begin{pmatrix} f^{attr} \\ f^{oRep} \\ f^{pRep} \end{pmatrix}$.

In order to fit a model to the training data, the R Environment [119] is used. R is an open software for statistical computing and graphics. A linear hypothesis is generated by R using *lm()* function. The resulting vector contains the weight values for each type of social force. Detailed examination of the data and results are given in Section 5.2.

5. EXPERIMENTS AND RESULTS

5.1. Overview

In the mathematical formulation of Social Force Model, there are several parameters. Prior studies on the model provide values for most of them. In [108], Zanlungo *et al*. present the results of their research on human-obstacle relation using this model. They calibrate the parameters using sampling methods in a series of simulation tests and the results are given in Table 5.1.

Table 5.1. SFM parameters as presented by Zanlungo *et al.* in [108].

	τ	A	B	d
Circular Specification	0.204 s	10 m/s^2	0.34 m	0.16 m

Note that, we ignore d in our experiments since we treat the robot and the people as 2D points on the map. Moreover, it is important to underline here that these parameters describe the reaction of people to obstacles.

The application of SFM to robot navigation aims to increase the comprehensibility of the motion of robots by imitating the motion of humans. Since the robot is expected to imitate humans, adoption of these parameters directly is reasonable.

The relation between robots and people is researched in [87]. Ferrer *et al.* conducted real life experiments in which subjects controlled the robots using a joystick according to their comfort and intimidation levels. Again, they used genetic algorithms to calibrate the parameters that are given in Table 5.2. We also use these parameters in our application.

In the following sections we give the experiments conducted within the course of this study.

Table 5.2. SFM parameters as presented by Ferrer *et al.* in [87].

	τ	A	B	d
Circular Specification	0.435 s	2.66 m/s^2	0.79 m	0.4 m

5.2. Parameters-Learning Experiments

To the best of our knowledge, the obstacle representation is specific to this study. In other applications of SFM, such as [87], [86] and [88], there is no mention about the representations of the obstacles. Although there is no mention about the technique they used, Figure 5.1 from [86], implies that there might be some preprocessing to describe the obstacles before the navigation algorithm runs.

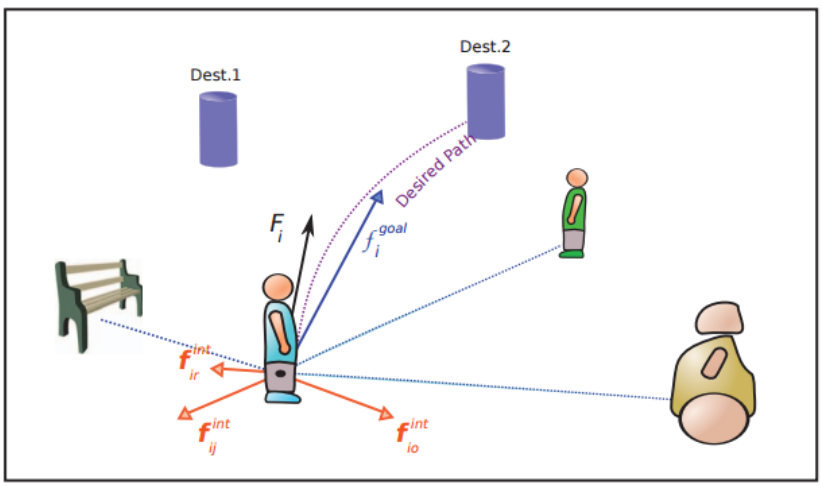
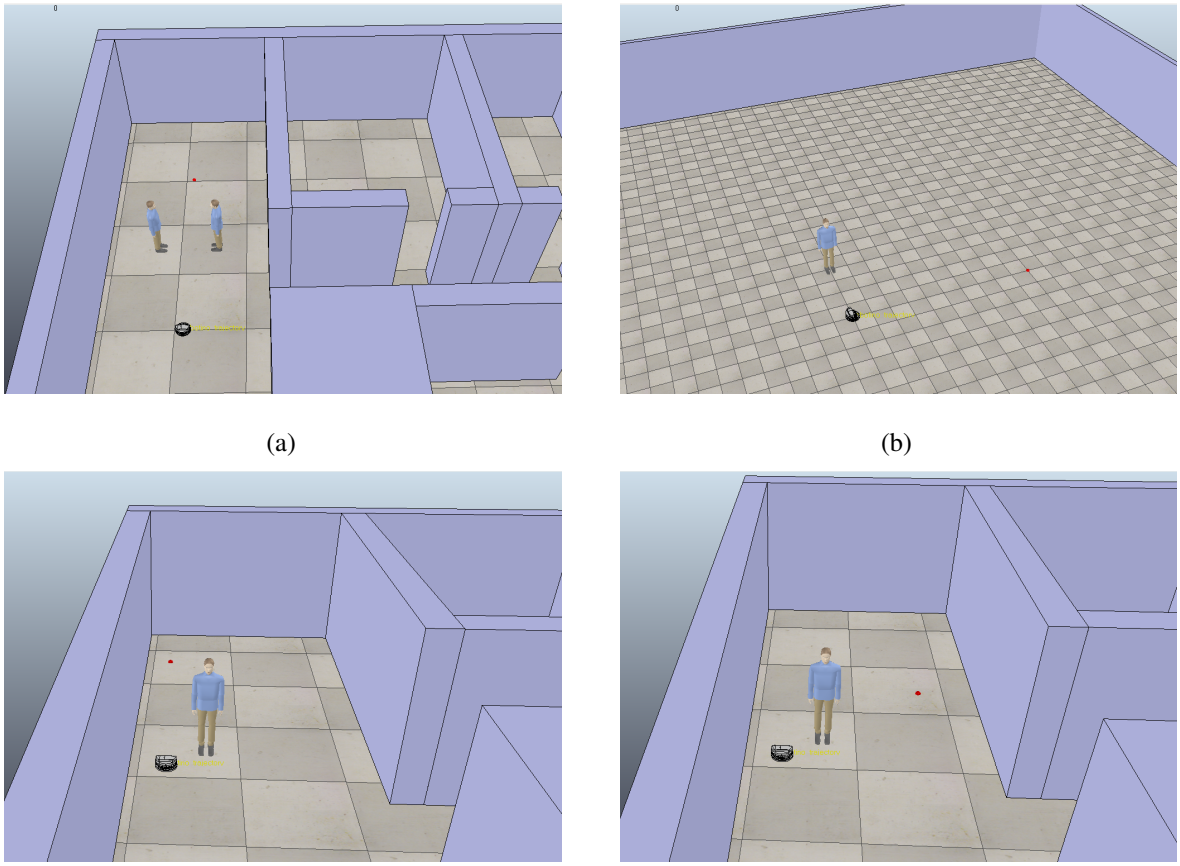


Figure 5.1. Social forces affecting the agent, from [86].

In this study, we represent the obstacles as occupied cells on the occupancy grid obtained after the mapping process. As explained in Section 4.2.2.3, the robot identifies the obstacles at run time using the learned map. On the other hand, as explained in Section 4.2.3, the sum of repulsive forces generated by obstacles has a lot of addends, in contrast to that of the repulsive force generated by the people. Therefore its magnitude is much greater than the latter. This condition, which is illustrated in Figure 4.16, prevents us from using the weights calculated in any of these studies.



(a) (b)

(c) (d)

Figure 5.2. Example scenarios in simulated environment.

Application of the social force model to robot navigation aims to increase the naturalness of the navigation of the robot. Since we desire the robot to imitate the human behavior, it is reasonable to learn these parameters from the recorded data of the experts. Therefore in the first set of learning experiments, 29 real life scenarios were simulated and human operators were asked to move the robot past people and obstacles towards the target. During the experiments three types of forces from the Equation 4.8 are recorded, as well as the velocity command provided by the human operator using a joystick. Example scenes from these scenarios are given in Figure 5.2.

We fit a linear model to the given data as described in Section 4.2.3. The summary of the model is given in Figure 5.3.

```
> summary(model3)

Call:
lm(formula = v ~ fa + fro + frp)

Residuals:
     Min       1Q   Median       3Q      Max
-0.50903 -0.08334 -0.02584  0.16295  0.43126

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0774795  0.0459303   1.687 0.097188 .
fa          0.0826568  0.0239888   3.446 0.001087 **
fro         0.0008746  0.0003589   2.437 0.017999 *
frp         0.0835058  0.0209839   3.980 0.000201 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2226 on 56 degrees of freedom
Multiple R-squared:  0.271,     Adjusted R-squared:  0.232
F-statistic:  6.94 on 3 and 56 DF,  p-value: 0.0004711
```

Figure 5.3. Summary of the first linear model.

5.2.1. Training Experiments with The First Data Set

The coefficients produced by the linear model are given in Table 5.3.

Table 5.3. First set of weights calculated by multiple linear regression.

Parameter	Value
α	0.0826568
β	0.0008746
γ	0.0835058

In addition to the coefficients, our procedure also outputs several statistical measures, such as *p-value*, R^2, etc. For $p - value < 0.05$ we reject the null hypothesis at 0.05 significance level. Therefore, we can say that there is a significant relationship between the variables in the linear regression model. Also, R^2 statistic is the ratio by which the variance of the errors is less than the variance of the dependent variable. In other words, it explains the percent of variance explained by the model. In the presented case, the $adjusted\ R^2 = 0.232$

which means approximately 23% of variation in velocity can be explained by these forces. Validation tests for this model is given in Section 5.3.

The latter statistic indicates that the model generated by multiple linear regression procedure, is not a very good candidate to represent the relation between the forces and the resulting velocity. This could depend on many things, rather than the failure of the linear models to explain the relation. We believe that one of the main reasons is the inexperience of the operators with the joystick with which the intermediate velocity values are quite hard to accomplish. That motivated us to extend the data set used for training and aim to learn the resulting direction, rather than both the direction and the magnitude, because of the difficulties in specifying the intermediate velocity values with the joystick.

5.2.2. Training Experiments with The Second Data Set

In these experiments we change the objective of the learning procedure, and we aim to learn only the direction of the motion; because, while the intermediate velocities are hard to produce for the human operators via a joystick leading to erroneous magnitude values, the direction of the velocity vectors do preserve the same mathematical relation. Therefore in the second set of the experiments, the same procedure is followed on the same set of simulated environments. This time each experiment is repeated 6 times yielding a set of 174 individual tests. Velocity commands produced by the human operators are recorded just as in the previous set of tests. However, the speed values are preprocessed before going further with the regression phase. All the velocity vectors are normalized so that they represent the direction of the motion that the human operator takes when he/she encounters such a scenario. The summary of the fitted model is given in Figure 5.4.

The linear model produced after this procedure is quite different than the previous one. The coefficients are given in Table 5.4.

Furthermore, with statistics $p - value = 2.2e - 16$, and $adjusted\ R^2 = 0.6711$ this model becomes a better candidate for explaining the relation among velocity and the force values. Validation experiments with these values are also given in Section 5.3.

```
> summary(model0Norm)

Call:
lm(formula = v ~ fa + fo + fp)

Residuals:
     Min       1Q   Median       3Q      Max
-1.06479 -0.23631 -0.01448  0.28878  0.77268

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0395113  0.0329519   1.199    0.231
fa          0.2740988  0.0137786  19.893   <2e-16 ***
fo          0.0024150  0.0002442   9.889   <2e-16 ***
fp          0.9866635  0.0539041  18.304   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4011 on 344 degrees of freedom
Multiple R-squared:  0.674,     Adjusted R-squared:  0.6711
F-statistic:   237 on 3 and 344 DF,  p-value: < 2.2e-16
```

Figure 5.4. Summary of the second linear model.

Table 5.4. Second set of weights calculated by multiple linear regression.

Parameter	Value
α	0.2740988
β	0.0024150
γ	0.9866635

5.3. Validation Experiments

Using the parameters which are taken directly from other studies as explained in Section 5.1, we can rewrite Equation 4.4 as follows:

$$\vec{f}_i^{rep} = \sum_{j \in O}^{O} 10e^{-d_{ij}/0.34}\hat{n}_{ij} + \sum_{j \in P}^{P} 2.66e^{-d_{ij}/0.79}\hat{n}_{ij} \tag{5.1}$$

5.3.1. Experiments on an Empty World

Velocity commands given to the robot are generated using Equations 4.7 and 4.8. We first experimented the navigation system in an empty world setting, as shown in Figure 5.5. We experimented with three different sets of weights;

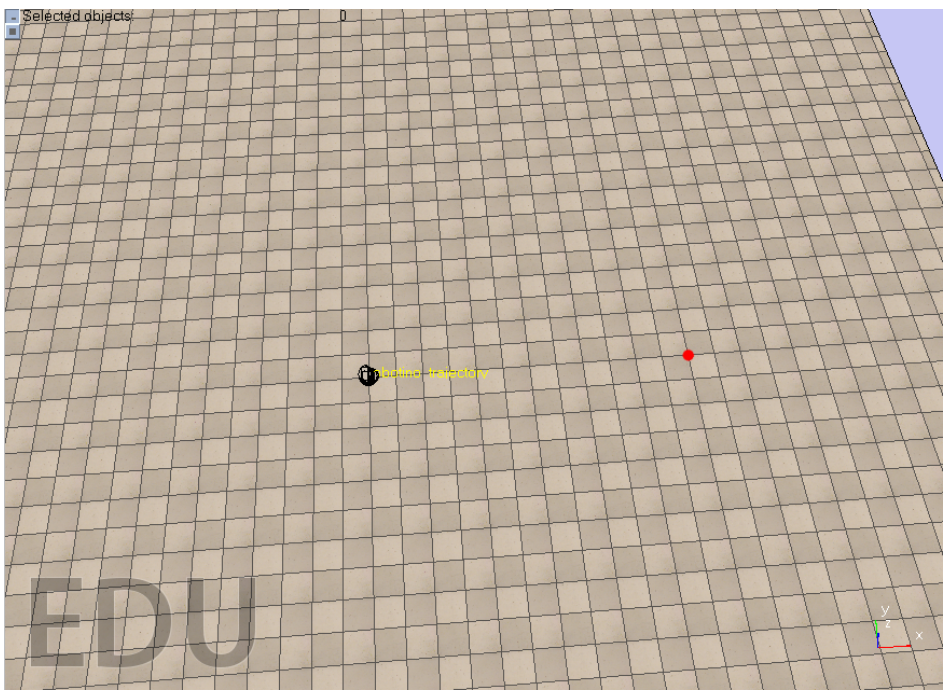


Figure 5.5. Simulated environment with no obstacles and no people.

(i) $\vec{w} = \begin{pmatrix} 1.0 \\ 0.2 \\ 3.18 \end{pmatrix}$, calibrated by Ferrer *et al.* in [86].

(ii) $\vec{w} = \begin{pmatrix} 0.0826568 \\ 0.0008746 \\ 0.0835058 \end{pmatrix}$, learned using first set of experiments.

(iii) $\vec{w} = \begin{pmatrix} 0.2740988 \\ 0.0024150 \\ 0.9866635 \end{pmatrix}$, learned using second set of experiments.

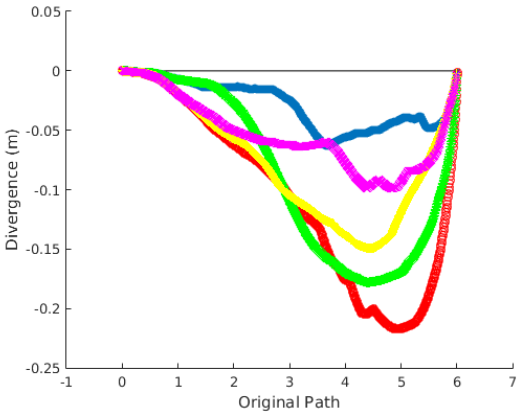
Five different observations are recorded with each set of weights. The divergences from the original paths are due to the systematic errors caused by the inaccuracies in the model of the robot on simulation. The plots in Figure 5.6 shows the ability of the weights used in the experiments to correct these errors and navigate the robot to the goal point. The comparison of the errors these coefficients produce is given in Figure 5.7.

These results indicate that our method for learn the weight coefficients outperforms the ones presented by Ferrer *et al.* in [86]. This indicates that the weights they use to generate the cumulative force that governs the changes in the velocity, shown in Equation 4.7 are not as successful as our weights while using our specific obstacle representation. The results also show that the coefficients learned using the second set of training data perform much better than the ones learned using the first set.

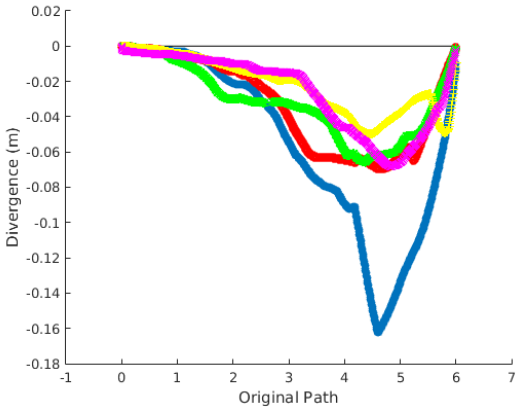
5.3.2. Experiments with One Person

These experiments aim to assess the effects of a single person on the performance of the algorithm and compare the performance of each set of weight coefficients in terms of proxemic metrics. The scenario is illustrated in Figure 5.8. Furthermore, Equation 5.1 is applied to calculate the repulsive force.

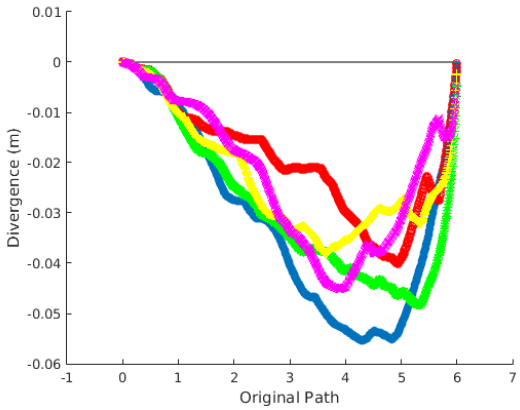
The path followed by the robot in each trial is given in Figure 5.9. Furthermore, the average path for each set of weight coefficients are given in Figure 5.10. Lastly, the personal



(a) Weights directly taken from [86].

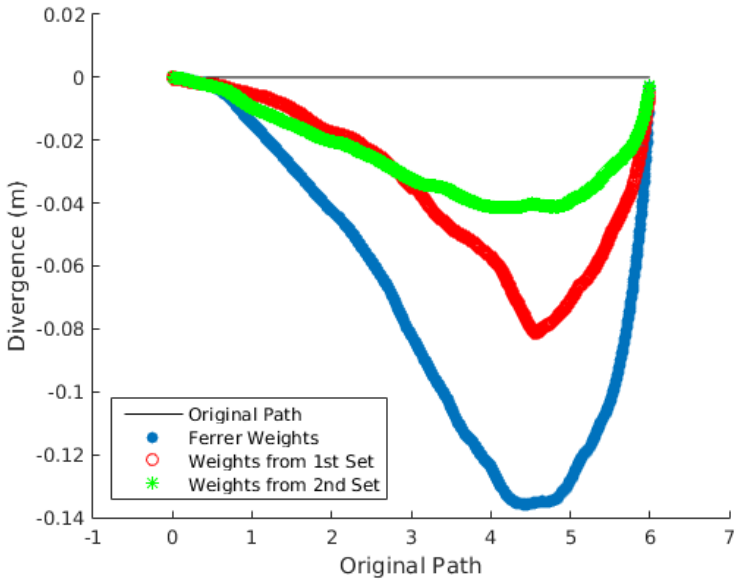


(b) Weights learned from the first set of experiments.

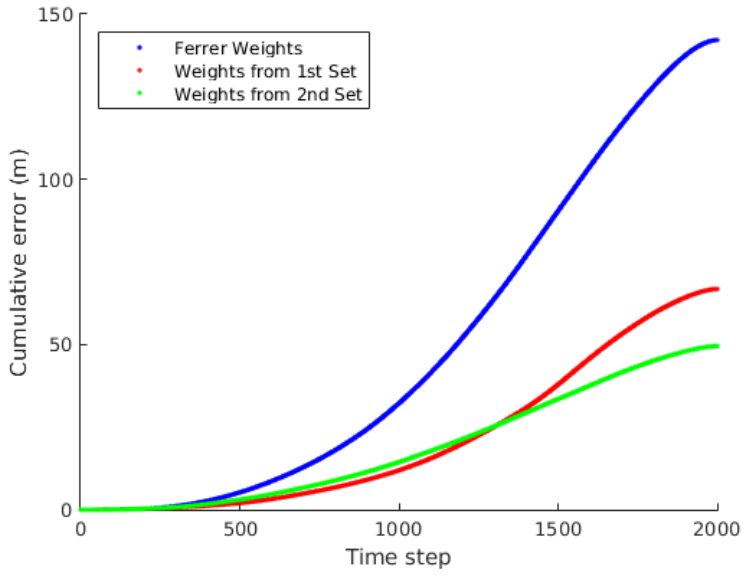


(c) Weights learned from the second set of experiments.

Figure 5.6. The trajectories followed by the robot in each experiment with different weight coefficients.



(a) Comparison of the averaged paths of 5 experiments.



(b) Comparison of the average cumulative error for each set of weights.

Figure 5.7. Comparison of average paths and average cumulative errors generated by three different set of weight coefficients.

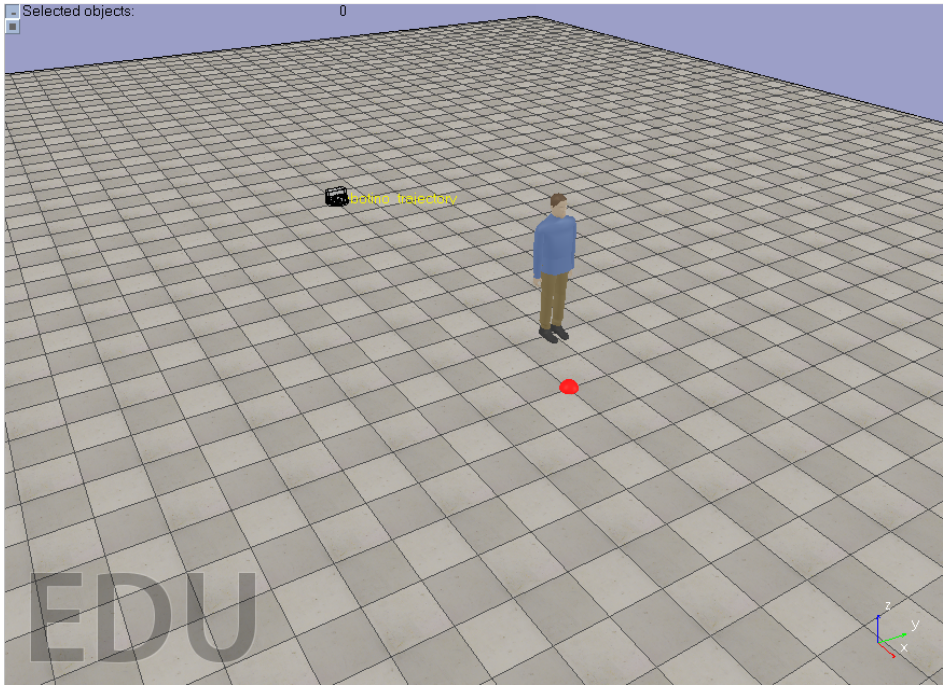


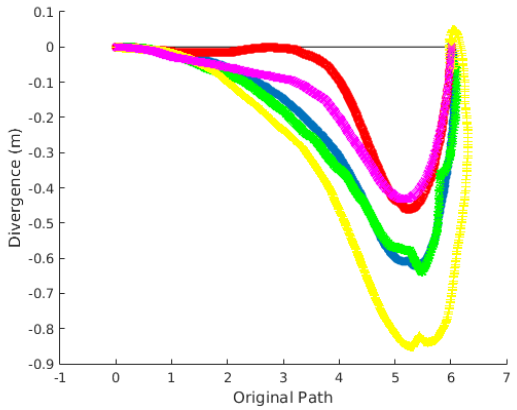
Figure 5.8. Simulated environment with no obstacles and one person.

zone and intimate zone intrusions and path lengths are illustrated in Figure 5.11.

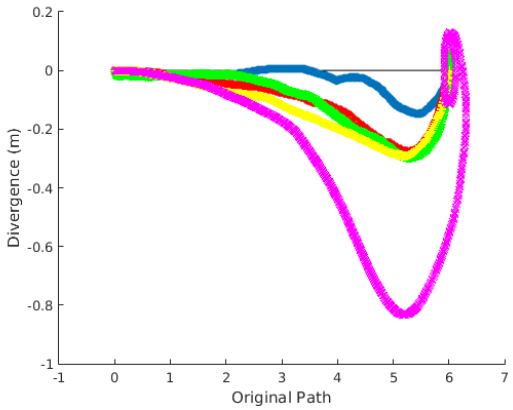
These results suggest that the most consistent paths are generated by the coefficients learned in the second training phase. Other sets of weight coefficients led to oscillation on multiple occasions around the navigation goal. This situation can be seen in Figure 5.9. Also, during the experiments, none of the weight sets led to the intrusion of the intimate zone, whereas the weights provided by Ferrer *et al.* led to the least number of personal zone intrusion, the absence of another entity prevents us from drawing concrete conclusions. The weights learned in second training phase led to the shortest path length on average.

5.3.3. Experiments with Two Obstacles

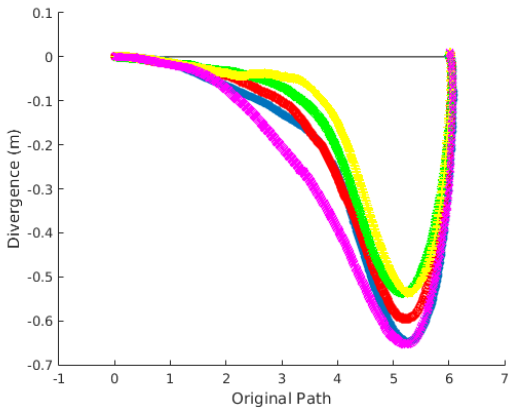
These experiments aim to assess the repulsive effects of obstacles on the performance of the algorithm and compare the performance of each set of weight coefficients in terms of applicable proxemic metrics. The scenario is illustrated in Figure 5.12. In this scenario, two obstacles are placed 1 meter apart on the planned route of the robot. The expected behavior



(a) Weights directly taken from [86].



(b) Weights learned from the first set of experiments.



(c) Weights learned from the second set of experiments.

Figure 5.9. The trajectories followed by the robot in each experiment with different weight coefficients.

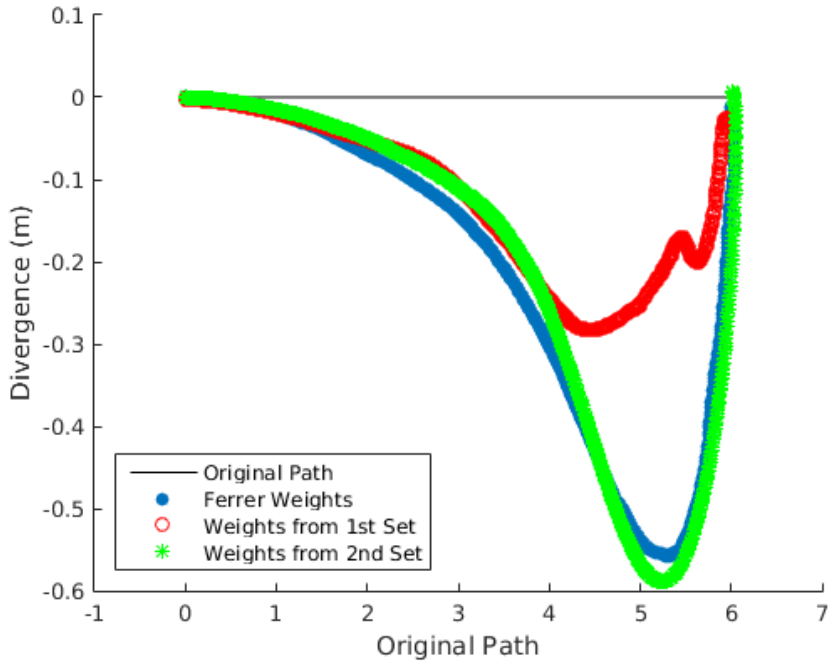
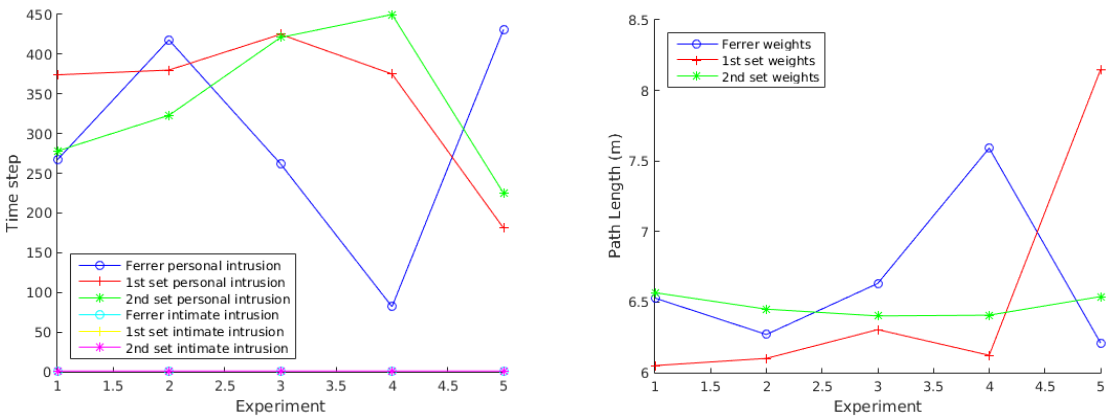


Figure 5.10. Comparison of average paths generated by three different set of weight coefficients.



(a) Personal zone and intimate zone intrusions.

(b) Traveled path lengths.

Figure 5.11. Comparison of weight coefficients with respect to the *path length* and *proxemic intrusion* metrics.

for the robot is to pass between the obstacles towards the target.

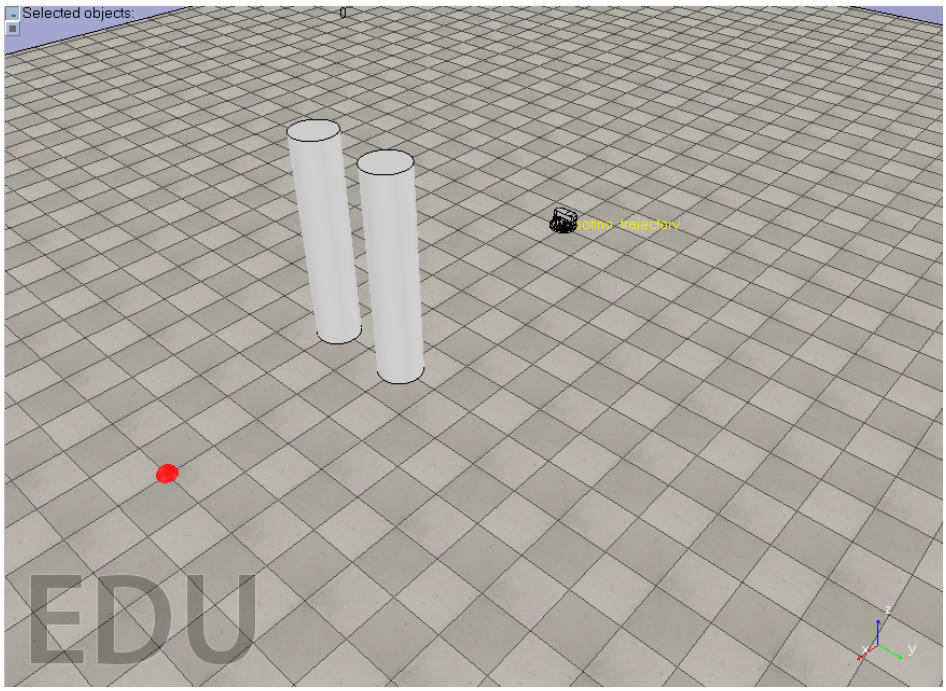


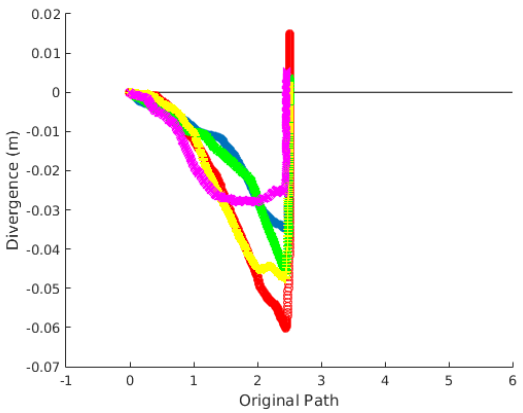
Figure 5.12. Simulated environment with two obstacles and no people.

The path followed by the robot in each trial is given in Figure 5.13. Furthermore, the average path for each set of weight coefficients are given in Figure 5.14. Since we cannot use personal and intimate zone concepts in this experiment, we only show the path lengths which are illustrated in Figure 5.15.

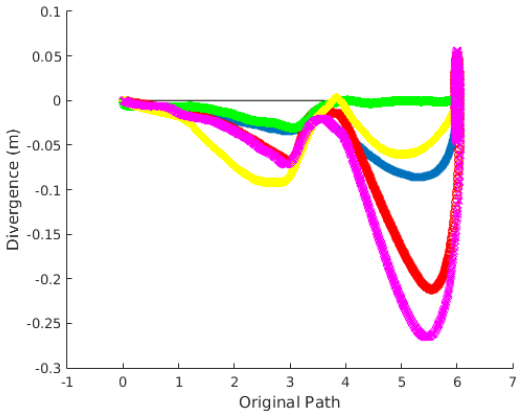
These experiments reveal the insufficiency of the weights taken from [86] to calculate obstacle-generated repulsive forces using our representation. In all of the five experiments, this set of weights failed to complete the navigation tasks. The weights learned using the second training data set led to almost-perfect paths with very small divergences from the original path resulting in rather short path lengths.

5.3.4. Experiments with Two People

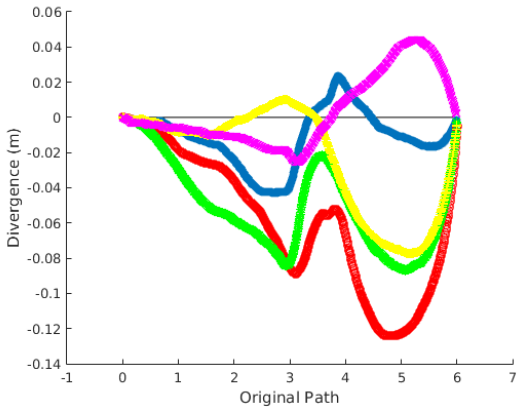
This series of experiments aim to compare the repulsive effects generated by obstacles with the ones generated by people and compare the performance of each set of weight



(a) Weights directly taken from [86].



(b) Weights learned from the first set of experiments.



(c) Weights learned from the second set of experiments.

Figure 5.13. The trajectories followed by the robot in each experiment with different weight coefficients.

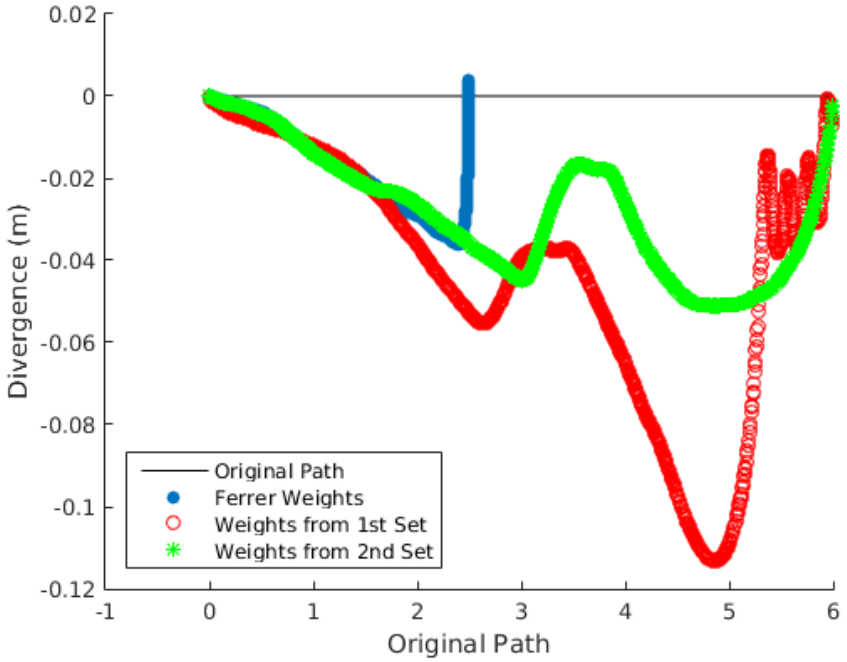


Figure 5.14. Comparison of average paths generated by three different set of weight coefficients.

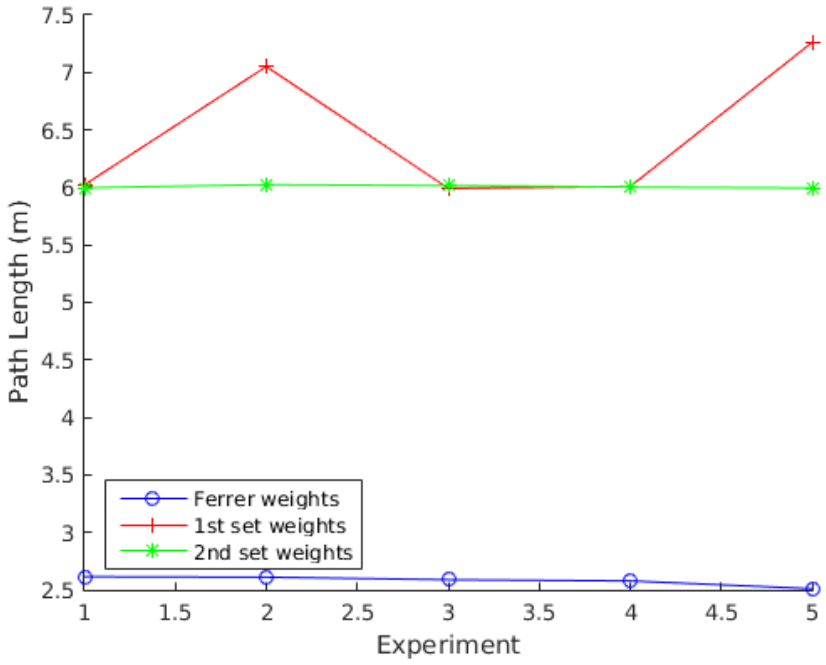


Figure 5.15. Comparison of weight coefficients with respect to the *path length* metric.

coefficients in terms of proxemic metrics. The scenario is illustrated in Figure 5.16. In this scenario, two people are placed 1 meter apart on the planned route of the robot which is quite similar to the previous scenario with the only difference being the type of the entities in the environment. The expected behavior for the robot is not to pass between people.

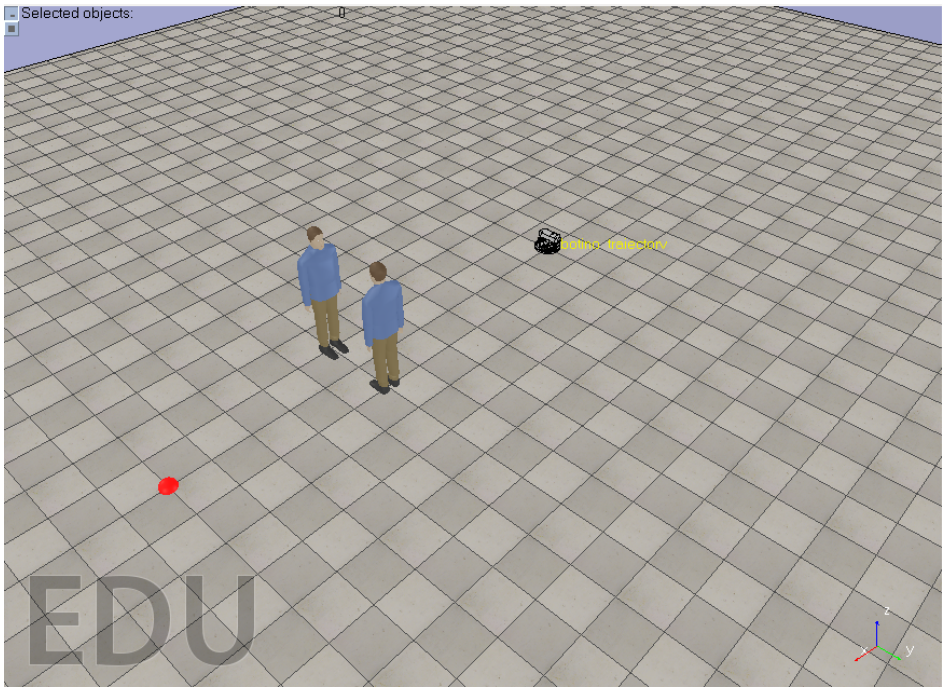
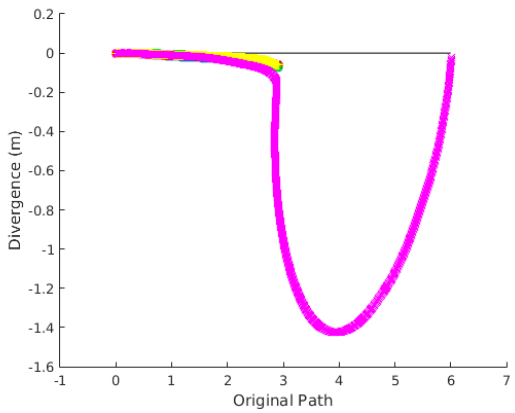


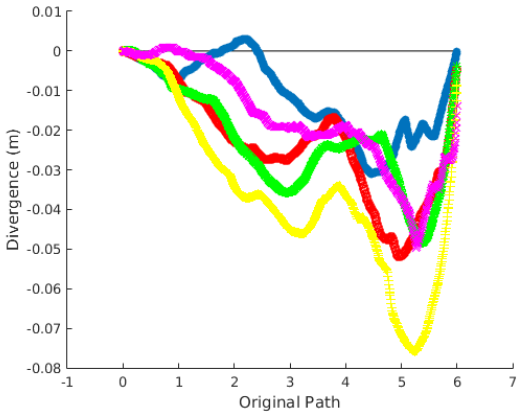
Figure 5.16. Simulated environment with no obstacles and two people.

The path followed by the robot in each trial is given in Figure 5.17. Furthermore, the average path for each set of weight coefficients are given in Figure 5.18. Lastly, personal zone and intimate zone intrusions and path lengths are illustrated in Figure 5.19.

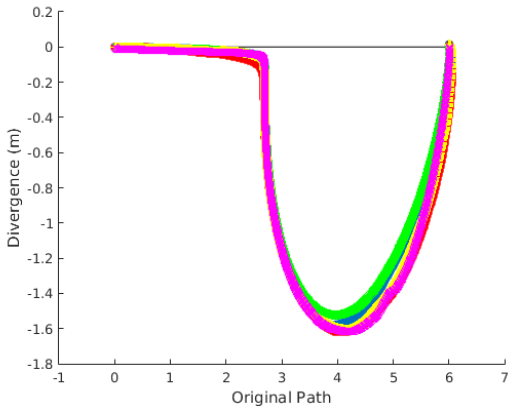
Although the robot could not complete the tests every time using the weights provided by Ferrer *et at.*, we should admit that it is one of the expected behaviors. In case of a local minimum, we expect the robot to abort its current path and change its direction to a less intrusive direction. Therefore, while average paths do not offer much to comment on, it is obvious that the weight coefficients learned in the first training phase fail to achieve the expected behavior, which is the most important deficiency of these parameters. On the other hand, in all of the experiments, the coefficients learned in second training phase were able to diverge from the path in reasonable time and reach to the target location. This set of



(a) Weights directly taken from [86].



(b) Weights learned from the first set of experiments.



(c) Weights learned from the second set of experiments.

Figure 5.17. The trajectories followed by the robot in each experiment with different weight coefficients.

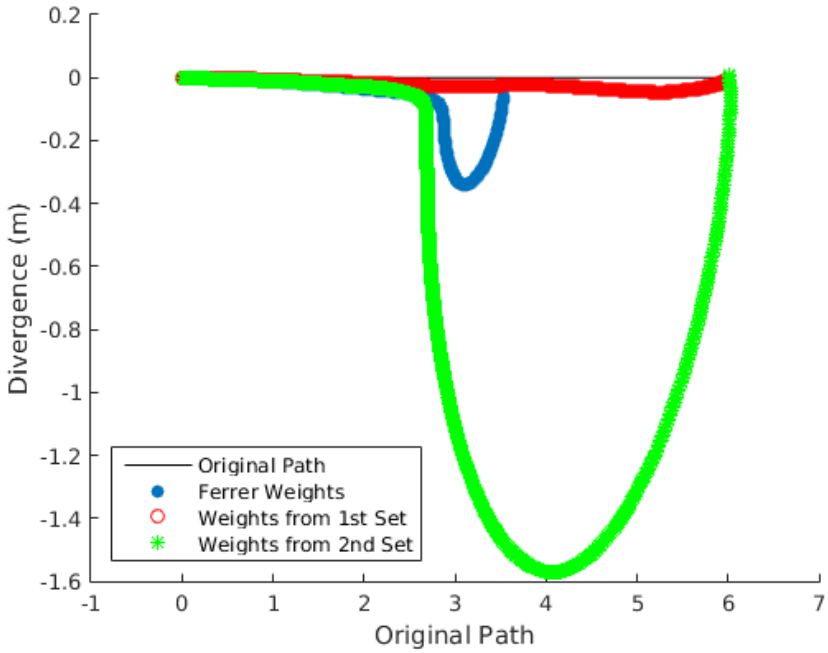
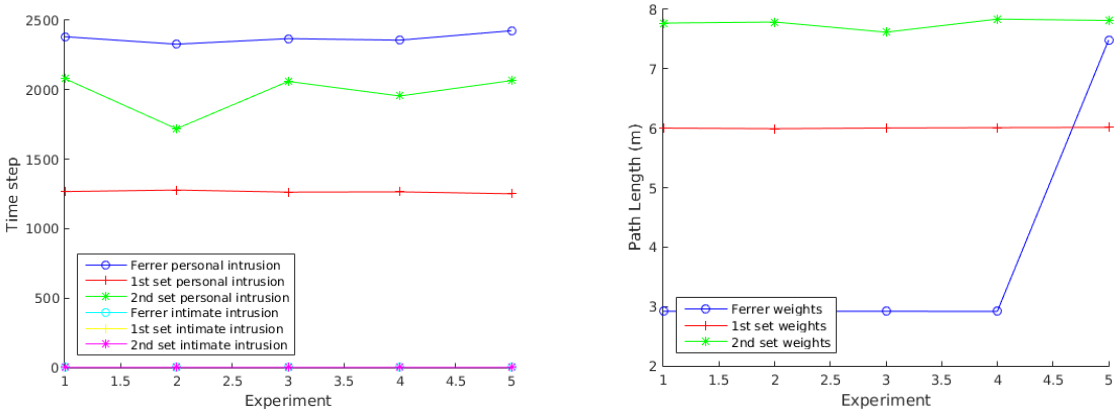


Figure 5.18. Comparison of average paths generated by three different set of weight coefficients.



(a) Personal zone and intimate zone intrusions.

(b) Traveled path lengths.

Figure 5.19. Comparison of weight coefficients with respect to the *path length* and *proxemic intrusion* metrics.

experiments attest to the success of these parameters.

We also cannot use path length metric to compare the performances of Ferrer-weights and the weights from second training set, due to the incomplete paths. Out of the two sets of coefficients that achieved the expected behaviors, it is obvious that the parameters learned in the second training set caused less intrusions. We can say that using the coefficients learned in the first training phase, the robot cannot diverge enough to present any kind of awareness.

5.3.5. Experiments with One Person and One Obstacle

These experiments aim to compare the repulsive effects generated by an obstacle and a person. The scenario is illustrated in Figure 5.20. In this scenario, one person and one obstacle are placed 1 meter apart on the planned route of the robot which is quite similar to the previous scenarios with the only difference being the type of the entities in the environment.

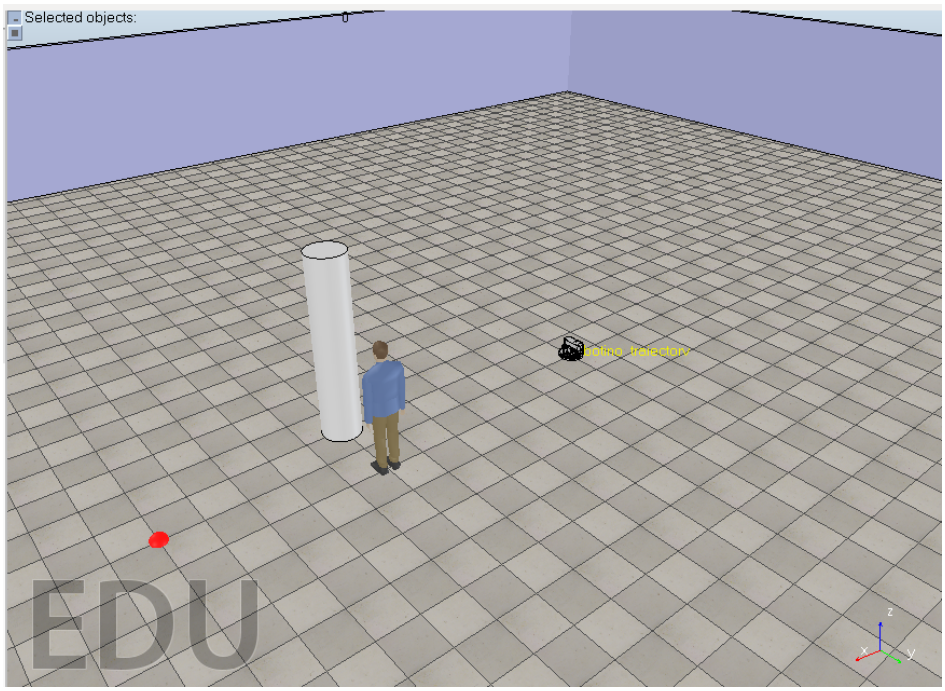
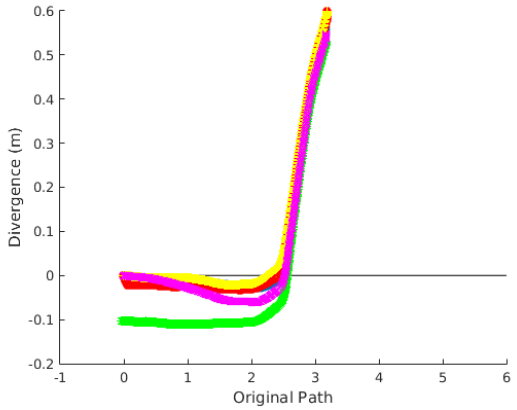
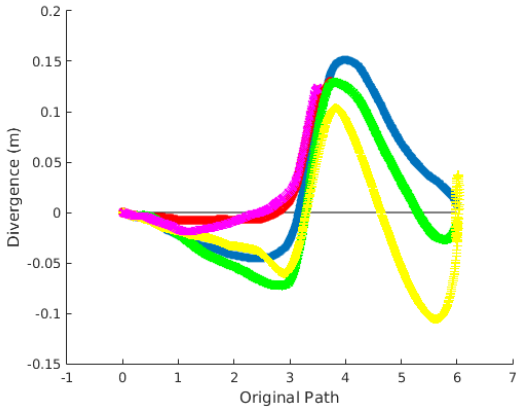


Figure 5.20. Simulated environment with one obstacle and one person.

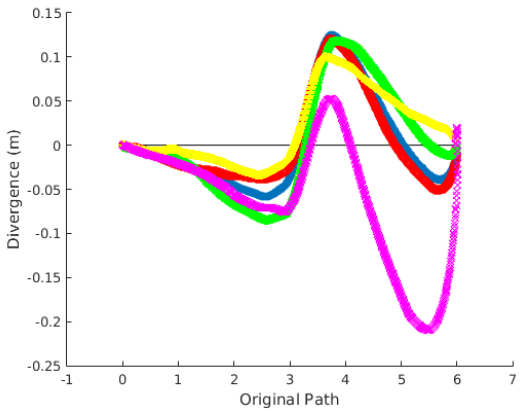
The path followed by the robot in each trial is given in Figure 5.21. Furthermore, the average path for each set of weight coefficients are given in Figure 5.22. Lastly, personal



(a) Weights directly taken from [86].



(b) Weights learned from the first set of experiments.



(c) Weights learned from the second set of experiments.

Figure 5.21. The trajectories followed by the robot in each experiment with different weight coefficients.

zone and intimate zone intrusions and path lengths are illustrated in Figure 5.23.

In this set of experiments, paths calculated using Ferrer-weights led to collision with the person. Weight coefficients learned in the first training phase also led to collision in two instances out of five experiments. Weights calculated in the second training phase again were able to finish the experiments without colliding into any entities; however, it led to a high number of personal and intimate zone intrusions. In all experiments, the robot diverged from the path towards the person, although one would expect it to pass the obstacle-person system by moving around the obstacle. This situation reveals the need for a better calibration

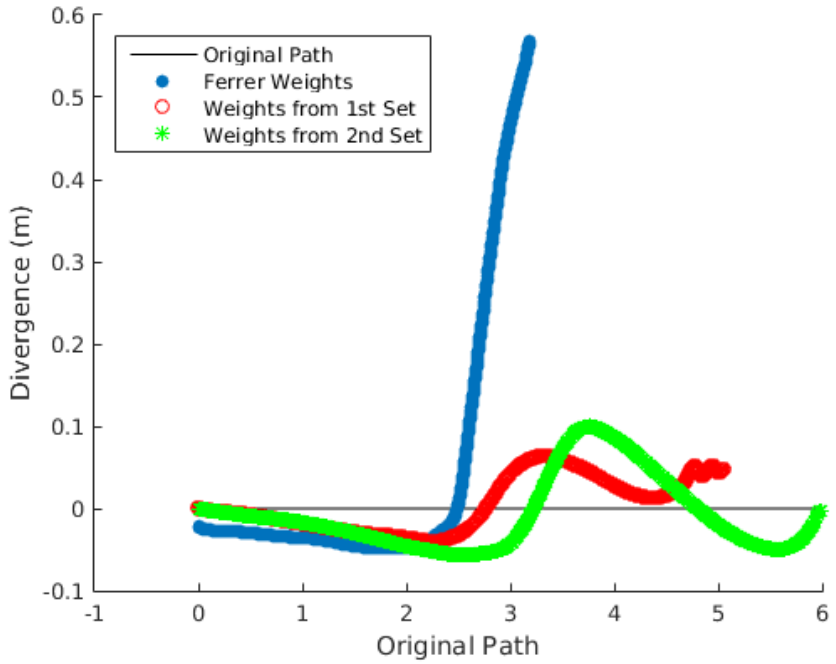
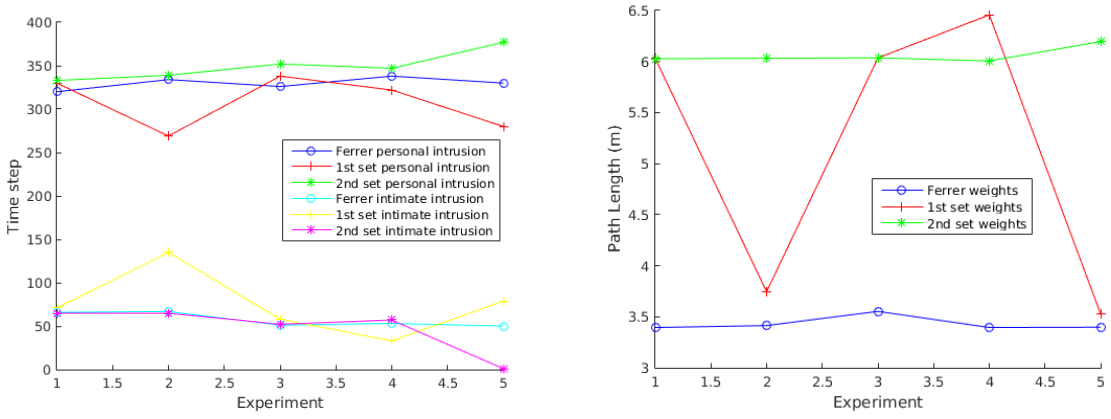


Figure 5.22. Comparison of average paths generated by three different set of weight coefficients.



(a) Personal zone and intimate zone intrusions.

(b) Traveled path lengths.

Figure 5.23. Comparison of weight coefficients with respect to the *path length* and *proxemic intrusion* metrics.

between A_q and B_q parameters of obstacles and people.

6. CONCLUSION

The need for more socially-aware navigation frameworks for autonomous mobile robots is obvious. In this thesis, we present an implementation of Social Force Model as a local path planner for mobile robots. The model is believed to describe the dynamics of human motion. For many years, it has been used widely by many researchers with various purposes. In the field of mobile robotics, until quite recently, the concept has been used to describe merely the motions of the humans for the robot to make meaningful predictions about their future paths. On the other hand, HRI studies suggest that we can further exploit this model on our robots to increase the similarity between the motion of robots and humans. It is explained in this thesis that, this is a proven way to enhance the comprehensibility and social acceptance of the robot by people that are sharing the same environment.

We present an extensive literature survey on the subject using studies from not only robotics science but also from psychology and sociology. As the subject is a relatively new one, we are yet to see holistic approaches to the problem of human-aware navigation and classification of the techniques to accomplish it. It is offered in this thesis to further extend the taxonomy on comfort related studies and classify them according to the adaptations they offer, such as direction of the approach or speed of the motion.

To the best of our knowledge, we use a novel representation of the static obstacles in the environment. Describing obstacles as occupied cells in the previously learned map and using 2D ray tracing technique to consider only the sides of the obstacles facing the robot is the most reasonable way to consider them. This specific representation restrained us to use the force calculation parameters obtained in other studies. Therefore, we chose to learn these parameters from an experienced human operator using supervised learning techniques. The results presented in Section 5.3 indicate that the learned weight coefficients outperform the ones proposed by Ferrer *et al.* in [86]. A reason of this may be the specific representation technique we used in this thesis. On the other hand, authors of that research do not provide enough information on their representation of obstacles; so, we cannot have an undeniable comparison.

6.1. Future Work

As a future work, we plan to implement and integrate a people detection unit on TURGAY in order to end the dependence of the navigation system to external sensory systems for people detection. For our *multi-robot tour-guide robot system* project [102], the independence of TURGAY has a vital importance.

The parameters of A_q and B_q that are explained in Section 4.1.1 also plays a role in the avoidance behavior. As depicted in Figure 4.15, these parameters decide on the range and the intensity of the repulsive potential. Calibration of these parameters was outside the scope of this thesis; therefore, they are taken directly from [86]. However, it is suspected that these parameters are robot-dependent. As another future work, we plan to deduce the parameters of our platform, TURGAY, with human-involved tests.

A lot of research has done on the personal spaces of the humans and assigning cost functions to these spaces that consist of different zones, as explained by the theory of proxemics. The social force model lacks the representation of interpersonal zones. As another future work, we will focus on representing the repulsive potential generated by the humans to reflect this idea, similar to dynamic personal space introduced in [64].

We also plan to research statistics-based strategies such as the application of inverse reinforcement learning to infer cost functions from the recorded data. Furthermore, we intend to make experiments with human subjects to come up with an explanation of the speed preferences in a guiding tasks. Initial experiments suggest that the approach speeds presented in [76], are not applicable to guiding tasks. People usually get bored and lose interest at that speeds.

The still-increasing interest of the robotics community in the problem may indicate the validity of the efforts on the problem. In the years-to-come we believe that there would be much more researchers working on the problem of human-aware navigation.

REFERENCES

1. Helbing, D. and P. Molnar, "Social Force Model for Pedestrian Dynamics", *Physical Review E*, Vol. 51, No. 5, p. 4282, 1995.

2. Dudek, G. and M. Jenkin, *Computational Principles of Mobile Robotics*, 2nd edn., Cambridge University Press, New York, NY, USA, 2010.

3. Milford, M. J., *Robot Navigation from Nature: Simultaneous Localisation, Mapping, and Path Planning Based on Hippocampal Models*, Springer Publishing Company, Incorporated, Berlin-Heidelberg, Germany, 2008.

4. Burgard, W., A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun, "The Interactive Museum Tour-Guide Robot", *AAAI-98 Proceedings*, pp. 11–18, AAAI, Wisconsin, USA, 1998.

5. Fox, D., W. Burgard and S. Thrun, "The Dynamic Window Approach to Collision Avoidance", *IEEE Robotics & Automation Magazine*, Vol. 4, No. 1, pp. 23–33, 1997.

6. Thrun, S., M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte and D. Schulz, "MINERVA: A Second-Generation Museum Tour-Guide Robot", *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, Vol. 3, pp. 1999–2005, IEEE, Michigan, USA, 1999.

7. Roy, N., W. Burgard, D. Fox and S. Thrun, "Coastal Navigation-Mobile Robot Navigation with Uncertainty in Dynamic Environments", *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, Vol. 1, pp. 35–40, IEEE, Michigan, USA, 1999.

8. Fox, D., W. Burgard, S. Thrun and A. B. Cremers, "A Hybrid Collision Avoidance Method for Mobile Robots", *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, Vol. 2, pp. 1238–1243, IEEE, Leuven, Belgium, 1998.

9. Nourbakhsh, I. R., J. Bobenage, S. Grange, R. Lutz, R. Meyer and A. Soto, "An Affective Mobile Robot Educator with a Full-Time Job", *Artificial Intelligence*, Vol. 114, No. 1, pp. 95–124, 1999.

10. Willeke, T., C. Kunz and I. R. Nourbakhsh, "The History of the Mobot Museum Robot Series: An Evolutionary Study", *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*, pp. 514–518, AAAI Press, Florida, USA, 2001.

11. Graf, B. and O. Barth, "Entertainment Robotics: Examples, Key Technologies and Perspectives", *Proceedings of the IEEE/RSJ IROS Workshop on Robots in Exhibitions*, IEEE, Lausanne, Switzerland, 2002.

12. Graf, B., J. H. Wandosell and C. Schaeffer, "Flexible Path Planning for Nonholonomic Mobile Robots", *Proceedings. 4th European Workshop on Advanced Mobile Robots (EUROBOT'01)., Fraunhofer Inst. Manufact. Eng. Automat.(IPS)*, pp. 199–206, Lund, Sweden, 2001.

13. Schraft, R. D., A. Traub, D. John and B. Graf, "PolarBug—Ein Effizienter Algorithmus zur Reaktiven Hindernisumfahrung im Dauereinsatz", *Autonome Mobile Systeme 2000*, pp. 96–101, Springer, 2000.

14. Tomatis, N., R. Philippsen, B. Jensen, K. O. Arras, G. Terrien, R. Piguet and R. Siegwart, "Building a Fully Autonomous Tour Guide Robot: Where Academic Research Meets Industry", *Proc. 33rd International Symposium on Robotics (ISR'02)*, ETH-Zürich, Stockholm, Sweden, 2002.

15. Latombe, J.-C., *Robot Motion Planning*, Kluwer, Boston, MA, USA, 1991.

16. Quinlan, S. and O. Khatib, "Elastic Bands: Connecting Path Planning and Control", *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pp. 802–807, IEEE, Georgia, USA, 1993.

17. Philippsen, R. and R. Siegwart, “Smooth and Efficient Obstacle Avoidance for a Tour Guide Robot”, *Robotics and Automation, 2003. Proceedings. ICRA ’03. IEEE International Conference on*, Vol. 1, pp. 446–451, IEEE, Taipei, Taiwan, 2003.

18. Kim, G., W. Chung, K.-R. Kim, M. Kim, S. Han and R. H. Shinn, “The Autonomous Tour-Guide Robot Jinny”, *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, Vol. 4, pp. 3450–3455, IEEE, Sendai, Japan, 2004.

19. Chung, W., G. Kim, M. Kim and C. Lee, “Integrated Navigation System for Indoor Service Robots in Large-Scale Environments”, *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*, Vol. 5, pp. 5099–5104, IEEE, New Orleans, LA, USA, 2004.

20. Rodriguez-Losada, D., F. Matia, R. Galan and A. Jimenez, “Blacky, an Interactive Mobile Robot at a Trade Fair”, *Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on*, Vol. 4, pp. 3930–3935, IEEE, Washington, DC, USA, 2002.

21. Rodriguez-Losada, D., F. Matia, J. M. Lucas, J. M. Montero, M. Hernando and R. Galan, “Urbano, an Interactive Mobile Tour-Guide Robot”, *Advances in Service Robotics*, pp. 229–252, 2008.

22. Faber, F., M. Bennewitz, C. Eppner, A. Görög, C. Gonsior, D. Joho, M. Schreiber and S. Behnke, “The Humanoid Museum Tour Guide Robotinho”, *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pp. 891–896, IEEE, Toyama, Japan, 2009.

23. Barraquand, J., B. Langlois and J.-C. Latombe, “Numerical Potential Field Techniques for Robot Path Planning”, *Systems, Man and Cybernetics, IEEE Transactions on*, Vol. 22, No. 2, pp. 224–241, 1992.

24. Sandt, F. and L.-H. Pampagnin, “Perception for a Transport Robot in Public Envi-

ronment", *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, Vol. 1, pp. 360–365, IEEE, Grenoble, France, 1997.

25. Prassler, E., J. Scholz and P. Fiorini, "Navigating a Robotic Wheelchair in a Railway Station During Rush Hour", *The international journal of robotics research*, Vol. 18, No. 7, pp. 711–727, 1999.

26. Fiorini, P. and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles", *The International Journal of Robotics Research*, Vol. 17, No. 7, pp. 760–772, 1998.

27. Montemerlo, M., J. Pineau, N. Roy, S. Thrun and V. Verma, "Experiences with a Mobile Robotic Guide for the Elderly", *Eighteenth National Conference on Artificial Intelligence*, pp. 587–592, American Association for Artificial Intelligence, Menlo Park, CA, USA, 2002.

28. Simmons, R., D. Goldberg, A. Goode, M. Montemerlo, N. Roy, B. Sellner, C. Urmson, M. Bugajska, M. Coblenz, M. Macmahon, D. Perzanowski, I. Horswill, R. Zubek, D. Kortenkamp, B. Wolfe, T. Milam, M. Inc and B. Maxwell, "GRACE: An Autonomous Robot for the AAAI Robot Challenge", *AI Magazine*, Vol. 24, pp. 51–72, 2003.

29. Konolige, K., "A Gradient Method for Realtime Robot Control", *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, Vol. 1, pp. 639–646, IEEE, Takamatsu, Japan, 2000.

30. Clodic, A., S. Fleury, R. Alami, R. Chatila, G. Bailly, L. Brethes, M. Cottret, P. Danes, X. Dollat, F. Elisei, I. Ferrane, M. Herrb, G. Infantes, C. Lemaire, F. Lerasle, J. Manhes, P. Marcoul, P. Menezes and V. Montreuil, "Rackham: An Interactive Robot-Guide", *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pp. 502–509, IEEE, Herthfordshire, UK, 2006.

31. Gross, H.-M., H. Boehme, C. Schroeter, S. Mueller, A. Koenig, E. Einhorn, C. Martin, M. Merten and A. Bley, "TOOMAS: Interactive Shopping Guide Robots in Everyday Use-Final Implementation and Experiences from Long-Term Field Trials", *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 2005–2012, IEEE, St. Louis, USA, 2009.

32. Borenstein, J. and Y. Koren, "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots", *Robotics and Automation, IEEE Transactions on*, Vol. 7, No. 3, pp. 278–288, 1991.

33. Trulls, E., A. Corominas Murtra, J. Pérez-Ibarz, G. Ferrer, D. Vasquez, J. M. Mirats-Tur and A. Sanfeliu, "Autonomous Navigation for Mobile Service Robots in Urban Pedestrian Environments", *Journal of Field Robotics*, Vol. 28, No. 3, pp. 329–354, 2011.

34. Kummerle, R., M. Ruhnke, B. Steder, C. Stachniss and W. Burgard, "A Navigation System for Robots Operating in Crowded Urban Environments", *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 3225–3232, IEEE, Karlsruhe, Germany, 2013.

35. Konolige, K., E. Marder-Eppstein and B. Marthi, "Navigation in Hybrid Metric-Topological Maps", *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3041–3047, IEEE, Shanghai, China, 2011.

36. Prassler, E., *Advances in Human-Robot Interaction*, Vol. 14, Springer Science & Business Media, Berlin, Germany, 2005.

37. Fong, T., I. Nourbakhsh and K. Dautenhahn, "A Survey of Socially Interactive Robots", *Robotics and Autonomous Systems*, Vol. 42, No. 3, pp. 143–166, 2003.

38. Goodrich, M. A. and A. C. Schultz, "Human-robot Interaction: A Survey", *Foundations and Trends in Human-Computer Interaction*, Vol. 1, No. 3, pp. 203–275, 2007.

39. Hinde, R. A., *Non-Verbal Communication*, Cambridge University Press, London, 1972.

40. Argyle, M., "Non-Verbal Communication in Human Social Interaction", R. Hinde (Editor), *Non-Verbal Communication*, Cambridge U. Press, London, 1972.

41. Payrató, L., "Non-Verbal Communication", *Key Notions for Pragmatics*, Vol. 1, pp. 163–194, 2009.

42. Salichs, M., R. Barber, A. Khamis, M. Malfaz, J. Gorostiza, R. Pacheco, R. Rivas, A. Corrales, E. Delgado and D. Garcia, "Maggie: A Robotic Platform for Human-Robot Social Interaction", *Robotics, Automation and Mechatronics, 2006 IEEE Conference on*, pp. 1–7, IEEE, Bangkok, Thailand, 2006.

43. Kruse, T., A. K. Pandey, R. Alami and A. Kirsch, "Human-Aware Robot Navigation: A survey", *Robotics and Autonomous Systems*, Vol. 61, No. 12, pp. 1726–1743, 2013.

44. Breazeal, C., "Toward Sociable Robots", *Robotics and Autonomous Systems*, Vol. 42, No. 3, pp. 167–175, 2003.

45. Nonaka, S., K. Inoue, T. Arai and Y. Mae, "Evaluation of Human Sense of Security for Coexisting Robots Using Virtual Reality. 1st Report: Evaluation of Pick and Place Motion of Humanoid Robots", *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, Vol. 3, pp. 2770–2775, IEEE, New Orleans, LA, USA, 2004.

46. Elsevier, "Scopus", `http://www.scopus.com/`, [Accessed August 2015].

47. Rios-Martinez, J., A. Spalanzani and C. Laugier, "From Proxemics Theory to Socially-Aware Navigation: A Survey", *International Journal of Social Robotics*, Vol. 7, No. 2, pp. 137–153, 2014.

48. Nakauchi, Y. and R. Simmons, "A Social Robot that Stands in Line", *Autonomous Robots*, Vol. 12, No. 3, pp. 313–324, 2002.

49. Hall, E., *The Hidden Dimension*, Anchor Books, New York, NY, US, 1966.

50. Christensen, H. I., E. Pacchierotti and K. T. Hgskolan, "Embodied social interaction for robots", *Proceedings of the 2005 Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB-05)*, pp. 40–45, Hertfordshire, UK, 2005.

51. Minguez, J. and L. Montano, "Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios", *Robotics and Automation, IEEE Transactions on*, Vol. 20, No. 1, pp. 45–59, 2004.

52. Huang, K.-C., J.-Y. Li and L.-C. Fu, "Human-Oriented Navigation for Service Providing in Home Environment", *SICE Annual Conference 2010, Proceedings of*, pp. 1892–1897, IEEE, Taipei, Taiwan, 2010.

53. Lam, C.-P., C.-T. Chou, C.-F. Chang and L.-C. Fu, "Human-Centered Robot Navigation—Toward a Harmoniously Coexisting Multi-Human and Multi-Robot Environment", *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1813–1818, IEEE, Taipei, Taiwan, 2010.

54. Syrdal, D. S., K. L. Koay, M. L. Walters and K. Dautenhahn, "A Personalized Robot Companion?-The Role of Individual Differences on Spatial Preferences in HRI Scenarios", *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pp. 1143–1148, IEEE, Jeju Island, Korea, 2007.

55. Mead, R. and M. J. Mataric, "A Probabilistic Framework for Autonomous Proxemic Control in Situated and Mobile Human-Robot Interaction", *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, pp. 193–194, ACM, Boston, MA, USA, 2012.

56. Torta, E., R. H. Cuijpers, J. F. Juola and D. van der Pol, "Design of Robust Robotic Proxemic Behaviour", *Social Robotics*, pp. 21–30, Springer, 2011.

57. Asghari Oskoei, M., M. Walters and K. Dautenhahn, "An Autonomous Proxemic System for a Mobile Companion Robot", *Proceedings of the AISB 2010 Symposium on New Frontiers for Human Robot Interaction*, AISB, Leicester, UK, 2010.

58. Takayama, L. and C. Pantofaru, "Influences on Proxemic Behaviors in Human-Robot Interaction", *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 5495–5502, IEEE, St. Louis, USA, 2009.

59. Walters, M., M. Oskoei, D. Syrdal and K. Dautenhahn, "A Long-Term Human-Robot Proxemic Study", *RO-MAN, 2011 IEEE*, pp. 137–142, IEEE, Atlanta, Georgia, USA, 2011.

60. Sisbot, E. A., L. F. Marin-Urias, R. Alami and T. Simeon, "A Human Aware Mobile Robot Motion Planner", *Robotics, IEEE Transactions on*, Vol. 23, No. 5, pp. 874–883, 2007.

61. Svenstrup, M., T. Bak and H. J. Andersen, "Trajectory Planning for Robots in Dynamic Human Environments", *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 4293–4298, IEEE, Taipei, Taiwan, 2010.

62. Svenstrup, M., S. Tranberg, H. J. Andersen and T. Bak, "Pose Estimation and Adaptive Robot Behaviour for Human-Robot Interaction", *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 3571–3576, IEEE, Kobe, Japan, 2009.

63. Kuffner, J. J. and S. M. LaValle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning", *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, Vol. 2, pp. 995–1001, IEEE, San Francisco, CA, USA, 2000.

64. Scandolo, L. and T. Fraichard, "An Anthropomorphic Navigation Scheme for Dynamic Scenarios", *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 809–814, IEEE, Shanghai, China, 2011.

65. Vasquez, D., B. Okal and K. O. Arras, "Inverse Reinforcement Learning Algorithms and Features for Robot Navigation in Crowds: An Experimental Comparison", *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 1341–1346, IEEE, Chicago, Illinois, USA, 2014.

66. Ramon-Vigo, R., N. Perez-Higueras, F. Caballero and L. Merino, "Transferring Human Navigation Behaviors into a Robot Local Planner", *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, pp. 774–779, IEEE, Edinburgh, Scotland, 2014.

67. Kim, B. and J. Pineau, "Socially Adaptive Path Planning in Human Environments Using Inverse Reinforcement Learning", *International Journal of Social Robotics*, pp. 1–16, 2015.

68. Diego, G. and K. O. Arras, "Please Do Not Disturb! Minimum Interference Coverage for Social Robots", *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 1968–1973, IEEE, San Francisco, California, USA, 2011.

69. Dylla, F., A. Kreutzmann and D. Wolter, "A Qualitative Representation of Social Conventions for Application in Robotics", *2014 AAAI Spring Symposium Series*, AAAI, Palo Alto, California, USA, 2014.

70. Kessler, J., C. Schroeter and H.-M. Gross, "Approaching a Person in a Socially Acceptable Manner Using a Fast Marching Planner", *Intelligent Robotics and Applications*, pp. 368–377, Springer, 2011.

71. Koay, K. L., E. A. Sisbot, D. S. Syrdal, M. L. Walters, K. Dautenhahn and R. Alami, "Exploratory Study of a Robot Approaching a Person in the Context of Handing Over an Object.", *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, pp. 18–24, Palo Alto, California, USA, 2007.

72. Sethian, J. A., "A Fast Marching Level Set Method for Monotonically Advancing Fronts", *Proceedings of the National Academy of Sciences*, Vol. 93, No. 4, pp. 1591–

1595, 1996.

73. Kessler, J., A. Scheidig and H.-M. Gross, “Approaching a Person in a Socially Acceptable Manner using Expanding Random Trees.”, *ECMR*, pp. 94–100, Örebro, Sweden, 2011.

74. Mead, R., A. Atrash and M. J. Matarić, “Proxemic Feature Recognition for Interactive Robots: Automating Metrics from the Social Sciences”, *Social Robotics*, pp. 52–61, Springer, 2011.

75. Calderon, C. A. A., C. Zhou and R. E. Mohan, “Development of an Autonomous Service Robot for Social Interactions”, *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, pp. 1–6, IEEE, Irbid, Jordan, 2011.

76. Butler, J. T. and A. Agah, “Psychological Effects of Behavior Patterns of a Mobile Personal Robot”, *Autonomous Robots*, Vol. 10, No. 2, pp. 185–202, 2001.

77. Dautenhahn, K., M. Walters, S. Woods, K. L. Koay, C. L. Nehaniv, A. Sisbot, R. Alami and T. Siméon, “How May I Serve You?: A Robot Companion Approaching a Seated Person in a Helping Context”, *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pp. 172–179, ACM, New York, NY, USA, 2006.

78. Satake, S., T. Kanda, D. F. Glas, M. Imai, H. Ishiguro and N. Hagita, “How to Approach Humans?-Strategies for Social Robots to Initiate Interaction”, *Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on*, pp. 109–116, IEEE, La Jolla, CA, USA, 2009.

79. Walters, M. L., K. Dautenhahn, R. Te Boekhorst, K. L. Koay, D. S. Syrdal and C. L. Nehaniv, “An Empirical Framework for Human-Robot Proxemics”, *Proceedings of New Frontiers in Human-Robot Interaction*, pp. 144–149, AISB, Edinburgh, Scotland, 2009.

80. Althaus, P., H. Ishiguro, T. Kanda, T. Miyashita and H. Christensen, “Navigation for Human-Robot Interaction Tasks”, *Robotics and Automation, 2004. Proceedings.*

ICRA'04. 2004 IEEE International Conference on, Vol. 2, pp. 1894–1900, IEEE, New Orleans, LA, USA, 2004.

81. Kluge, B. and E. Prassler, "Reflective Navigation: Individual Behaviors and Group Behaviors", *IEEE International Conference on Robotics and Automation*, pp. 4172–4177, IEEE, New Orleans, LA, USA, 2004.

82. Müller, J., C. Stachniss, K. O. Arras and W. Burgard, "Socially Inspired Motion Planning for Mobile Robots in Populated Environments", *International Conference on Cognitive Systems (CogSys'08)*, Karlsruhe, Germany, 2008.

83. Luber, M., L. Spinello, J. Silva and K. O. Arras, "Socially-aware robot navigation: a learning approach", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'12)*, pp. 902–907, IEEE, Vilamoura, Portugal, 2012.

84. Martinez-Garcia, E., O. Akihisa and S. Yuta, "Crowding and Guiding Groups of Humans by Teams of Mobile Robots", *Advanced Robotics and its Social Impacts, 2005. IEEE Workshop on*, pp. 91–96, IEEE, Nagoya, Japan, 2005.

85. Tamura, Y., P. D. Le, K. Hitomi, N. P. Chandrasiri, T. Bando, A. Yamashita and H. Asama, "Development of Pedestrian Behavior Model Taking Account of Intention", *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 382–387, IEEE, Vilamoura, Portugal, 2012.

86. Ferrer, G., A. Garrell and A. Sanfeliu, "Social-Aware Robot Navigation in Urban Environments", *Mobile Robots (ECMR), 2013 European Conference on*, pp. 331–336, IEEE, Barcelona, Spain, 2013.

87. Ferrer, G., A. Garrell and A. Sanfeliu, "Robot Companion: A Social-Force Based Approach with Human Awareness-Navigation in Crowded Environments", *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 1688–1694, IEEE, Tokyo, Japan, 2013.

88. Ratsamee, P., Y. Mae, K. Ohara, M. Kojima and T. Arai, "Social Navigation Model Based on Human Intention Analysis Using Face Orientation", *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 1682–1687, IEEE, Tokyo, Japan, 2013.

89. Shiomi, M., F. Zanlungo, K. Hayashi and T. Kanda, "Towards a Socially Acceptable Collision Avoidance for a Mobile Robot Navigating Among Pedestrians Using a Pedestrian Model", *International Journal of Social Robotics*, Vol. 6, No. 3, pp. 443–455, 2014.

90. Kirby, R., R. Simmons and J. Forlizzi, "Companion: A Constraint-Optimizing Method for Person-Acceptable Navigation", *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pp. 607–612, IEEE, Toyama, Japan, 2009.

91. Pandey, A. K. and R. Alami, "A Framework for Adapting Social Conventions in a Mobile Robot Motion in Human-Centered Environment", *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–8, IEEE, Munich, Germany, 2009.

92. Chung, S.-Y. and H.-P. Huang, "Incremental Learning of Human Social Behaviors with Feature-Based Spatial Effects", *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2417–2422, IEEE, Vilamoura, Portugal, 2012.

93. Rios-Martinez, J., *Socially-Aware Robot Navigation: Combining Risk Assessment and Social Conventions*, Ph.D. Thesis, PhD thesis, Université de Grenoble, 2012.

94. Open Source Robotics Foundation, "Robot Operating System", `http://www.ros.org/`, [Accessed August 2015].

95. Coppelia Robotics, "Virtual Robot Experiment Platform", `http://www.coppeliarobotics.com`, [Accessed August 2015].

96. Grisetti, G., C. Stachniss and W. Burgard, "Improved Techniques for Grid Mapping

with Rao-Blackwellized Particle Filters", *Robotics, IEEE Transactions on*, Vol. 23, No. 1, pp. 34–46, 2007.

97. Fox, D., "KLD-Sampling: Adaptive Particle Filters", *Advances in Neural Information Processing Systems, NIPS 2001*, pp. 713–720, Vancouver, Canada, 2001.

98. Smith, R. L., "Open Dynamics Engine", `http://www.ode.org/`, [Accessed August 2015].

99. Bullet Physics Library, "Bullet Physics Library", `http://bulletphysics.org/`, [Accessed August 2015].

100. CM Labs, "Vortex Simulation Software", `http://www.cm-labs.com/`, [Accessed August 2015].

101. Boğaziçi University, "Robotics Group of Boğaziçi University", `http://robot.cmpe.boun.edu.tr`, [Accessed August 2015].

102. Yıldırım, Y., O. Aşık, B. Görer, N. E. Özkücür and H. L. Akın, "Tur Rehberi Çoklu Robot Sistemi", *Türkiye Otonom Robotlar Konferansı*, Ankara, Turkey, 2014, in Turkish.

103. Raspberry Pi, "Raspberry Pi 1 Model B", `https://www.raspberrypi.org/products/model-b`, [Accessed August 2015].

104. Hokuyo Automatic Co., Ltd, "URG-04LX Laser Scanner", `https://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html`, [Accessed August 2015].

105. Asus, "Asus Xtion Pro Live", `https://www.asus.com/us/Multimedia/Xtion_PRO_LIVE/`, [Accessed August 2015].

106. Festo, "Robotino", `http://www.festo-didactic.com/int-en/learning-systems/education-and-research-robots-robotino`, [Accessed August 2015].

107. Helbing, D. and P. Molnár, "Social Force Model for Pedestrian Dynamics", *Physical Review E*, Vol. 51, pp. 4282–4286, 1995.

108. Zanlungo, F., T. Ikeda and T. Kanda, "Social Force Model with Explicit Collision Prediction", *EPL (Europhysics Letters)*, Vol. 93, No. 6, p. 68005, 2011.

109. Ferrer, G., A. Garrell and A. Sanfeliu, "Social-Aware Robot Navigation in Urban Environments.", *ECMR*, pp. 331–336, IEEE, Barcelona, Spain, 2013.

110. Garrell, A. and A. Sanfeliu, "Cooperative Social Robots to Accompany Groups of People", *The International Journal of Robotics Research*, Vol. 31, No. 13, pp. 1675–1701, 2012.

111. Saboia, P. and S. Goldenstein, "Crowd Simulation: Applying Mobile Grids to the Social Force Model", *The Visual Computer*, Vol. 28, No. 10, pp. 1039–1048, 2012.

112. Helbing, D. and A. Johansson, "Pedestrian, Crowd, and Evacuation Dynamics", R. A. Meyers (Editor), *Encyclopedia of Complexity and Systems Science*, pp. 1–28, Springer New York, 2014, `http://dx.doi.org/10.1007/978-3-642-27737-5\_382-5`.

113. Choset, H., K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Cambridge, MA, US, 2005.

114. Wang, L. C., L. S. Yong and M. H. Ang Jr, "Hybrid of Global Path Planning and Local Navigation Implemented on a Mobile Robot in Indoor Environment", *Intelligent Control, 2002. Proceedings of the 2002 IEEE International Symposium on*, pp. 821–826, IEEE, Vancouver, Canada, 2002.

115. Elfes, A., "Using Occupancy Grids for Mobile Robot Perception and Navigation", *Computer*, Vol. 22, No. 6, pp. 46–57, 1989.

116. Kitazawa, K. and T. Fujiyama, "Pedestrian Vision and Collision Avoidance Behavior:

Investigation of the Information Process Space of Pedestrians Using an Eye Tracker", W. W. F. Klingsch, C. Rogsch, A. Schadschneider and M. Schreckenberg (Editors), *Pedestrian and Evacuation Dynamics 2008*, pp. 95–108, Springer Berlin Heidelberg, 2010.

117. LaMothe, A., *Tricks of the 3D Game Programming Gurus: Advanced 3D Graphics and Rasterization*, Sams Publishing, Indiana, US, 2003.

118. Alpaydin, E., *Introduction to Machine Learning*, 3rd edn., MIT press, Cambridge, Massachusetts, 2014.

119. The R Foundation, "The R Project for Statistical Computing", `http://www.r-project.org`, [Accessed August 2015].