University of Nevada, Reno

Multi-Context Socially-Aware Navigation Using Non-Linear Optimization

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in
Computer Science and Engineering

by
Santosh Balajee Banisetty

Dr. David Feil-Seifer/Dissertation Advisor

May 2020

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

SANTOSH BALAJEE BANISETTY

Entitled

Multi-Context Socially-Aware Navigation Using Non-Linear Optimization

be accepted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

David Feil-Seifer, Ph.D., Advisor

Monica Nicolescu, Ph.D., Committee Member

Hung La, Ph.D., Committee Member

Emily Hand, Ph.D., Committee Member

Jacqueline Snow, Ph.D., Graduate School Representative

David W. Zeh, Ph.D., Dean, Graduate School

May 2020

Abstract

This presents a framework for a novel Unified Socially-Aware Navigation (USAN) architecture and motivates it for Socially Assistive Robotics (SAR) applications. This approach emphasizes interpersonal distance and how spatial communication can be used to build a unified planner for a human-robot collaborative environment. Socially-Aware Navigation (SAN) is vital for helping humans to feel comfortable and safe around robots; HRI studies have shown the importance of SAN transcends safety and comfort. SAN plays a crucial role in the perceived intelligence, sociability, and social capacity of the robot, thereby increasing the acceptance of the robots in public places. Human environments are very dynamic and pose serious social challenges to robots intended for interactions with people. For robots to cope with the changing dynamics of a situation, there is a need to detect changes in the interaction context.

We present a context classification pipeline to allow a robot to change its navigation strategy based on the observed social scenario. Most of the existing research uses different techniques to incorporate social norms into robot path planning for a single context. Methods that work for hallway behavior might not work for approaching people, and so on. We developed a high-level decision-making subsystem, a model-based context classifier, and a multi-objective optimization-based local planner to achieve socially-aware trajectories for autonomously sensed contexts. Our approach augments the navigation stack of Robot Operating System (ROS) utilizing machine learning and optimization tools. Using a context classification system, the robot can select social objectives that are later used by Pareto Concavity Elimination Transformation (PaCcET) based local planner to generate safe, comfortable, and socially-appropriate trajectories for its environment. Our method was tested and validated in multiple environments on a Pioneer mobile robot platform; results show that the robot was able to select and account for social objectives autonomously.

We also developed new scales for observing HRI that can measure the perceived social intelligence (PSI) of robots. We validated our PSI scale by evaluating our PaCcET-based local planner; a bystander experiment showed that people perceived robots with socially appropriate navigation strategies as more socially intelligent when compared to robots using traditional navigation strategies.

Acknowledgements

I would like to thank Dr. David Feil-Seifer, Dr. Monica Nicolescu, Dr. Hung La, Dr. Emily Hand, and Dr. Jacqueline Snow for being on my committee, with special thanks to Dr. David Feil-Seifer for giving me the opportunity to do research in the Socially Assistive Robotics Group (SARG). I would like to thank Dr. Logan Yliniemi and Dr. Kimberly Barchard, for their collaboration. I would also like to thank my lab-mates for helping collect data and run the experiments for this research. Lastly, I would like to thank my friends and family for helping to support me through this tiresome time in my life.

This material is based in part upon work supported by: The National Science Foundation (NSF, #IIS-1719027, #IIS-1757929), Nevada NASA EPSCoR (#NNX15AK48A), and the Office of Naval Research (ONR, #N00014-16-1-2312, #N00014-14-1-0776). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the Office of Naval Research or Nevada NASA EPSCoR.

Contents

[1] Theoretical contribution

List of Figures

Chapter 1

Introduction

In this chapter, we are going to discuss the following

- A brief introduction to Socially-Aware Navigation (SAN).
- The motivating problem.
- Limitations of existing work and discuss briefly how the contributions of this dissertation address these limitations.

The introduction of service and assistive robots in homes and public places poses opportunities as well as challenges. Whether or not such robots meet the social expectations determines the naturalness of interaction with humans. A robot that meets social expectations can be viewed as a social entity, which is vital for long-term acceptance of robots in public places. To share space with people, these assistive and service robots must be designed with human safety as well as comfort and social norms in mind. Depending on the design, purpose, and capabilities of robots, the interaction dynamics between humans (one or more) and robots (one or more) have many complexities and inter-dependencies. For example, a robot designed to navigate a human environment, complexities include comfort, safety, etc. Inter-dependencies can be how walking on the right side of the hallway influence comfort. The prime focus of this dissertation is to augment a socially-aware robot navigation system to respond to various interaction dynamics associated with navigating various scenarios in human environments.

For SAN, spatial communication, in which personal space and distance are used to *communicate* with a human partner [1], can have an impact on the acceptance of assistive robots in human environments. Spatial communication is both voluntary and involuntary act of communicating intent among interacting humans. When navigating a narrow doorway, if a person steps back (voluntarily), that means he/she is offering the other person to pass first; This voluntary spatial communication is considered polite behavior in an HHI navigation setting. When invited for a group discussion, people usually tend to form a closed circle (involuntarily) as the discussion progresses, this behavior shows how invested and engaging the group is. Spatial communication is one of several subcategories of nonverbal communication; it is a study of the effects of spatial separation among individuals and plays an essential, but unconscious, role in everyday communication. Spatial communication is both voluntary and involuntary act of communicating intent among interacting humans.

For human-human interaction (HHI), humans can understand social norms via spatial communication as a normal consequence of social development. For human-robot interaction (HRI) to match this HHI property, the spatial interface between a human and a robot should be natural in order to achieve human-friendly navigation. Successful robot behaviors, including navigational behaviors, must be appropriate for a given social circumstance for the robot's long-term acceptance in human collaborative environments like hospitals, airports, and other public places. Failing to adhere to social norms that humans tend to follow might lower the quality of HRI.

The basis of this work is a model-based socially-aware navigation planner utilizing distance-based spatial features [2]. In previous work, we demonstrated that a similar model-based approach could discriminate human navigation behaviors. We used Gaussian Mixture Models (GMMs) with distance-based features in a hallway to distinguish various hallway interactions like *passing down the corridor*, *meeting*, *Walking together towards a goal*, and *walking together away from a goal* [3]. We developed an architecture [4] that utilizes our HHI model [3] to generate socially-aware trajectories by scoring every future trajectory point, at a local planning stage, against the HHI model.

Like other SAN planners, our prior work [4] assumes a linear relationship among spatial features. To account for any non-linear relationship in spatial communication in HRI related to navigation, we implemented a SAN planner [5] incorporating Pareto Concavity Elimination Transformation (PaCcET) [6]. In previous work [5], we showed that a non-linear multi-objective optimization method like PaCcET could be used to select an optimized future trajectory point, associated with a non-linear combination of costs. We showed in a simulation that the PaCcET-based trajectory planner not only can avoid collisions and reach the intended destination but also considers a person's personal space (i.e., rules of proxemics [7]) in the trajectory selection process. While this considered a simple single person, two objective interaction scenarios, we further extended PaCcET based SAN planner to complex interactions like *joining a group*, *waiting in a queue*, and *art gallery interaction* which involve more than two objectives [8].

1.1 Motivation

An ethnographic study [9] showed that problems caused due to inappropriate human navigation in HHI could also be caused due to inappropriate robot navigation in HRI. Inappropriate navigation behavior in a hospital environment caused users not to adopt the robot for long-term use. The study found that the design challenges of autonomous mobile robots go beyond aesthetics and usability, the responses of the participants of this long-term study show that social aspects such as *not walking on the right side*, *getting too close to people* played a significant role in how workers in a workplace used, interacted and perceived the robot. Participants felt "disrespected" by the robot as the robot took precedence in the hallways. This inappropriate behavior is because a traditional navigation planner [10] treats any detected occlusion

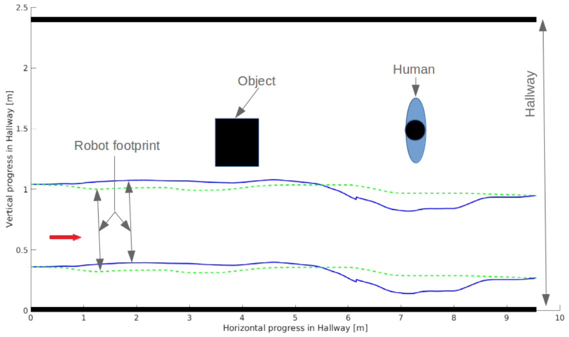


Figure 1.1: *Ideal social behavior*: SAN trajectory (blue) should be able to distinguish, by its navigation behavior, between people and objects. The traditional planner (green) fails to do so. This result demonstrates that the robot went close to the object but maintained distance with the human. Direction of the robot is shown with a red arrow.

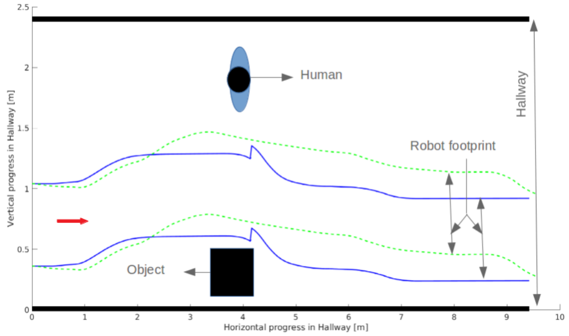


Figure 1.2: *Ideal social behavior*: Similar behavioral discrimination in tight spaces by SAN planner (Blue). Traditional planner (Green) fails to execute socially-aware behavior. This demonstrates that the robot prioritizes personal space even in tight spaces. Direction of the robot is shown with a red arrow.

as an obstacle. It is acceptable to treat furniture as a static obstacle, but people may feel uncomfortable interacting with a robot if it does not openly communicate its intentions by respecting traditional social norms.

Robots should not treat humans through their navigation behavior as dynamic objects [11]. Figures 1.1 and 1.2 illustrate a comparison between a traditional planner (green trajectory), which optimizes for performance (like time, distance) and a socially-aware planner (blue trajectory), which optimizes for social norms along with performance objectives. The traditional planner treats both people and objects in the same way (does not account for personal space around human). Not able to behaviorally distinguish humans and objects can lead to HRI missteps; it is acceptable to get close to an object, but similar behavior around a person is not acceptable. Proxemics [7] provides a theoretical model of personal space, yet there is ample evidence that human navigation preferences go beyond a mere distance of separation and include motion [12]. Effective SAN will investigate methods to integrate the rules of interpersonal motion into robot navigation behavior.

Problems such as these have made SAN an active area of research [13]. The goal of HRI researchers interested in SAN is to aid the acceptance of assistive robots by improving the robots' navigation behavior. The robots may better perform a navigation task by respecting the space and social norms of their human partners. By recognizing the social and personal space of people, a robot should adapt to the environment treating humans as social (and mobile) beings rather than mere obstacles.

1.2 Limitations of the State-of-the-Art SAN Methods

Traditional navigation algorithms can generate a collision-free path and maneuver a robot on that path to get to a goal. However, these algorithms do not consider social interactions that occur while navigating in highly dynamic human environments. There is a rapidly growing HRI community that is addressing SAN related challenges. The solutions to SAN associated problems range from simple cost functions to more advanced deep neural networks. Socially-Aware Navigation planners, including our method, consider the theory of proxemics and other social norms such that the robot exhibits socially appropriate behaviors. Proxemics [7] codifies this notion of personal space; researchers interested in SAN are investigating methods to integrate the rules of proxemics into robot navigation behavior. Current approaches to Socially-Aware Navigation have the following limitations:

- Many current approaches depend on exocentric sensing, hence, limiting the robot's services to a particular environment [2].
- Approaches may require a large amount of training data for low-level planning tasks [14–18].
- The environment/scenario is a singleton, i.e., Only a hallway, a room is considered, or considers only an approach behavior or a passing behavior [2, 19].
- Planners are optimized for single or few objectives with linear combination or weighted sum [20, 21].
- Lack of validation infrastructure that precisely measures social intelligence capabilities of robots using social navigation [22, 23].

1.3 Contributions

Accomplished contributions, directly related to this dissertation are listed below:

- Early work involved a model-based approach to discriminate navigation actions (scenario) [3]. The same model was used to achieve socially-aware navigation by modifying the local planner of ROS navigation stack [4].

- A local planner utilizing non-linear multi-objective optimization was implemented [5] and validated our PaCcET-based local planner in different contexts (*hallway, art gallery, joining a group and waiting in a line*) [8] both in simulated environment and on a Pioneer mobile robot.

- HRI instruments like Bartneck *et al.* [22], Nomura *et al.* [23] do not take into account Perceived Social Intelligence (PSI). We are bridging the gap and investigated the role of PSI in HRI [24–26].

- We used newly developed scales to measure the perceived social intelligence of robots incorporating proposed SAN planner as compared to the traditional navigation approach [27].

- A theory paper, identifying critical components of a Unified Socially-Aware Navigation planner architecture [28].

- Autonomous context classification system that can classify an on-going interaction and account for social norms that matter most for that particular context [29].

1.4 Summary

The introduction of service and assistive robots in homes and public places poses opportunities as well as challenges. Whether or not such robots meet the social expectations determines the success of its co-existence with humans. To share space along with people, these assistive and service robots should be designed, keeping in mind not only human safety but also human comfort and social aspects. Depending on the design and capabilities of robots, the interaction dynamics between humans (one or more) and robots (one or more) has many possibilities. The interaction dynamics related to navigation, Socially-Aware Navigation (SAN), is the prime focus of this work. The basis of this work is a model-based socially-aware navigation planner utilizing distance-based spatial features [2].

We have some accomplished contributions leading up to this dissertation; they are:

- We developed a local planner [5] that can generate SAN trajectories in multiple contexts (*hallway, art gallery, joining a group, and waiting in a line*) [8].

- We developed USAN architecture [28] that can autonomously senses the interaction context and execute SAN behaviors in various contexts like *hallways*, *art galleries*, *O-formations*, and *waiting in line* [29].
- We developed HRI instruments to measure Perceived Social Intelligence (PSI) of robots [26], which was used to measure PSI of robots with SAN behavior [27].

By implementing a Unified Socially-Aware Navigation architecture, both holonomic and non-holonomic robots can perform socially-aware navigation behavior in an autonomously sensed context. The contributions of our approach (listed above), address the limitations of the recent SAN research.

This dissertation demonstrates context-appropriate socially-aware navigation. Using autonomous context classification and a PaCcET-enabled local planner, we can achieve socially-aware navigation behaviors not just for a single context but for multiple contexts. We realize and validate our unified socially-aware navigation (USAN) architecture [28]. The remainder of this dissertation is structured as follows. In the next chapter, we review related work. In Chapter 3, we discuss our prior work on a model-based socially-aware navigation planner. In Chapter 4, we present a non-linear multi-objective optimization method to achieve socially-aware navigation in multiple contexts. In Chapter 5, we discuss the technical details of the architecture and validate it with a real robot that can exhibit social navigation behavior for an autonomously sensed context. In Chapter 6, we evaluate our SAN planner by measure perceived social intelligence. Finally, in Chapter 7, discussion and future directions are presented.

Chapter 2

Background

In this chapter, we are going to discuss the following:

- Brief history of traditional robot navigation, social navigation, and everything in between.
- We broadly classify existing SAN work into learning and non-learning approaches.
- Existing survey instruments in HRI, their limitations, and the role of PSI scales in bridging the gap.

In this chapter, we first brief about the behavioral difference between a robot with traditional navigation and a robot that uses socially-aware navigation methods. Second, we discuss the history of robotics, related to robot navigation, technical breakthroughs that lead to the state-of-the-art robot navigation methods that are widely used in present-day robots. Third, we discuss the social limitations of such traditional navigation methods when used by social robots that work along with people. Fourth, we classify the existing work, based on the methods used in implementing social navigation, into learning approaches and non-learning approaches. Lastly, we discuss the need for measuring the social intelligence of robots and its role in evaluating the social behaviors of robots using social navigation.

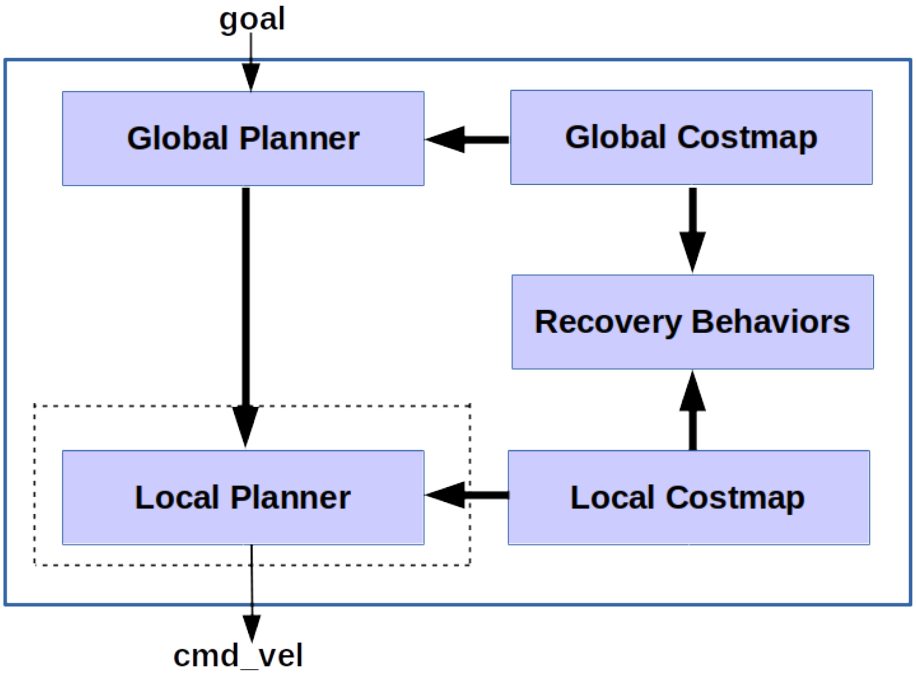


Figure 2.1: Block diagram of nav_core package from ROS navigation stack. We base our two SAN approaches on this nav_core functionality. The local planner (denoted by dotted lines) is modified using a social model, presented in Chapter 3. The local planner is modified using PaCcET, a non-linear optimization tool and is discussed in detail in Chapter 4. More details about the ROS navigation stack can be found at http://wiki.ros.org/navigation/

Traditional navigation algorithms, as shown in Figure 2.1 can generate a collision-free path and maneuver a robot on that path to get to a goal. However, these algorithms are not sophisticated enough to deal with social interactions that occur while navigating in highly dynamic human environments. For example, when a robot with traditional navigation exits a doorway with people standing in line, it would cut the line and exit through the doorway as it is cost function includes getting to the goal as quickly as possible. On the other hand, a robot with social navigation would wait in the line and let other people exit first hence considering the social norm of *waiting in line*. Proxemics [7], social rules for interpersonal distance, is an essential aspect of navigation; researchers interested in SAN are investigating methods to integrate the rules of proxemics into robot navigation behavior.

Social norms, such as driving on the right or left side of the road (depending on the country one lives in), turn-taking rules at four-way stops and roundabouts, holding doors for people behind us, and maintaining an appropriate distance when interacting with another person (actual distance depending on the type of interaction), are crucial in our day-to-day interactions. People use these actions as signals that they are participants in the social order, violating these principles is jarring at best (i.e., a person becoming confused at another person's behavior), and can provoke hostility at worst (i.e., getting upset at someone for cutting in line).

Table 2 shows a brief history of events related robot navigation in human environments that lead to evolution of socially-aware navigation. In this chapter, we present the solutions to SAN that range from simple cost functions to more advanced deep neural networks. Later in this chapter, we will see the most recent work in socially-aware navigation.

Table 2.1: Evolution of Socially-Aware Navigation

1949	Before 1950, robotics was a science fiction [30].
1950	The early 1950s is an early stage of artificial intelligence that drew inspiration from animal intelligence/behavior [31]. Autonomous mobile robots in the 50s were basically line-following carts (a robot that followed a line painted on the floor or ceiling) or autonomous robots with randomized motion or robots that follow the light.
1966	Hall [7] studied Proxemics, a formalism that specifies with the amount of space, for a given situation, that people feel necessary to set between themselves and others. With the advent of sophisticated robot navigation capabilities, today's researchers are looking into ways to incorporate Proxemics in HRI for a collaborative human-robot environment.

Table 2.1: Evolution of Socially-Aware Navigation

1972	Between 1960 and 1972, the artificial intelligence group at Stanford Research Institute conducted research on mobile robot platform called SHAKEY [32]. Shakey could move around the room; to a certain degree, it can respond to the environment and move in unfamiliar surroundings with some limitations due to poor sensing capabilities. Even with wobbly and cluttery movements (hence, Shakey), Shakey is an important milestone in modern mobile robotics. Some of the notable results from a decade-long research include famous A* algorithm, which later pioneered research in robot path planning and the famous feature extraction technique, Hough transforms, also an important invention that pioneered modern-day robot perception research.
1995	Early 90's, robot navigation dealt mostly with avoiding people (obstacle avoidance) in order to avoid robot accidents. Mere avoidance behavior poses serious challenges when interacting with humans. To the best of our knowledge, Tadokoro *et al.* [33] first studied coexistence and cooperation with humans in robot motion planning and navigation.
1998	RHINO [34], one of the first real-world service robot deployments. RHINO was deployed in "Deutsches Museum Bonn" where it offered guided tours to hundreds of visitors for six days.

TABLE 2.1: Evolution of Socially-Aware Navigation

1999	Burgard *et al.* [35] revisited RHINO's software architecture with particular emphasis on the design of interactive capabilities that appeal to interacting humans' intuition. Empirical results show reliable operation in human environments. This distributed software architecture provided a new means of human-robot interaction in crowded environments. Learnings from this lead to the development of MINERVA [36], a second-generation tour-guide robot.
2002	The early 2000s was a period when researchers pioneered Simultaneous Localization and Mapping (SLAM) [37, 38], which is one of the important advancements in mobile robotics.
2008	In an ethnographic study, Mutlu *et al.* [9] found that problems caused by inappropriate human navigation behavior can also be caused due to inappropriate robot navigation. The study found that robot navigation if performed without considering spatial communication can be detrimental for long-term human-robot interaction.

Table 2.1: Evolution of Socially-Aware Navigation

2010	Robust navigation in the indoor environment was made possible by Marder [10] when a PR2, a mobile manipulation robot completed a 26.2 mile run in an office environment. This state of the art navigation technique also has limitations related to human-robot interaction research. The availability of sophisticated navigation methods and availability of computational power opened new frontiers in Socially-Aware Navigation research which is presented in greater detail in the rest of the chapter.

With the late '90s, advancements in robot navigation that led to trail deployments of service robots in human environments, Nakauchi *et al.* [39], investigated social behaviors and their importance in crowded human environments. The authors demonstrated that a mobile robot (Xavier) could wait in a line just like people do when purchasing a cup of coffee. Performance results, such as *was the robot able to detect line formations* using a stereo camera, *was the robot able to reach the end of the line* were reported. Althaus *et al.* [40] developed methods that allow a robot to join a group of people having a conversation. The results show that the robot's movements are very similar to that of human movement patterns when joining a group. Three subjects in the experiment were required to answer a few questions about their impressions interacting with the robot. The subjects judged the moving patterns of the robot as natural, had the impression that the robot was engaging and perceived as intelligent. Gockley *et al.* [41] compared two different human following algorithms; both have similar characteristics regarding performance metrics (performance of laser-based people tracking, distance maintained with a human partner). However, in a user study, participants rated the direction following behavior to be more natural and human-like when compared to path following behavior.

The current navigation methods in this new field of social navigation are quite diverse; very different ad-hoc approaches have been proposed for specific sub problems. For example, there exist learning-based algorithms for socially-aware navigation [14, 15], game theory based decision making methods [42], reactive and proactive navigation methods [43], rule-based planners [44], cost-based planners [19, 45], etc. If focused on the solution to a specific sub problem, Gockley *et al.* [41] proposed methods to following a human. Feil-Seifer *et al.* [1] demonstrated a learning-based algorithm that enables a robot to encourage a human to follow it. Mead *et al.* [12] proposed a method that enables a robot to improve human-robot proxemics with respect to "interaction potential" and behavior to approach a human to start an interaction [46] are some of the recent problems that are being studied. Some of the research mentioned above will be discussed in greater detail in Section 2.0.1 and 2.0.2.

In this section, we review the state-of-the-art SAN methods. We broadly classify these recent trends into two categories namely *Non-learning approach*, and *Learning approach* and the pros and cons of both the categories. The ones that use a non-learning approach to SAN include the utilization of cost functions, optimization tools, and constraint-based planning. Learning approach to SAN includes classic Machine Learning (ML) models, Inverse Reinforcement Learning (IRL), Reinforcement Learning (RL), and Deep learning methods. Section 2.1 dedicated to evaluation methods, we discuss different evaluation methods used to evaluate SAN systems. Most of them are tailored to specific needs. We also identify the gaps in such evaluation methods and propose the need for measuring perceived social intelligence.

2.0.1 Non-learning approaches

Garrell *et al.* [47] presented a combination of Discrete-Time Motion (DTM) and a cost function for a group of robots to minimize the work required for leading and regrouping people. In simulation results, an analysis of forces actuating among all the agents was presented for various situations (guiding in open areas, areas with obstacles, etc.) and different cooperative robot behaviors. The results show that the system of robots satisfactorily guided a group of people through a path. Lam *et al.* [44] presented a navigation algorithm called Human-Centered Sensitive Navigation (HCSN) that considers states of multiple people (tracked using a laser scanner) and robots to establish a harmonious coexisting environment. HCSN considers the fact that both robots and humans have proxemic zones and model these zones based on several harmonious rules such as a collision-free rule, waiting rule, human rule, to name a few. This rule-based approach yielded socially acceptable movements in simulation and real-world environments. Rios *et al.* [48] demonstrated in a simulation that a navigation algorithm called Risk-RRT (a modified RRT) could incorporate proxemic constraints. Risk-RRT based method used both the traditional notion of risk of collision and risk of disturbance to achieve socially-aware navigation behavior. Demonstrations in simulations show that the robot generated trajectories that did not get in-between two conversing people, thereby adhering to the social norms. Turnwald *et al.* [42] presented a game-theoretic approach to SAN utilizing concepts from non-cooperative games and Nash equilibrium. The authors evaluated whether their proposed approach generated human-like motion behavior by conducting two experiments, video-based simulations shown to passive observers, and an in-person experiment where the participants interacted with a robot with different planners in virtual reality. The authors evaluated the game theory-based SAN planner against established planners such as reciprocal velocity obstacles or social forces, a variation of the Turing test was administered which determines whether participants can differentiate between human movements and artificially generated motions.

In another work, Lu *et al.*[19] modified the existing ROS navigation to make the robot navigate in a socially appropriate manner by adding Gaussian-based cost values around a detected human. This cost-based approach caused the robot to take a socially appropriate path in a hallway setting; the method was evaluated using ”Passage Behavior Parameters” [49]. Gaze behavior was also implemented for enhanced interaction, and the results show that gaze plays an important role in intent communication. In another work Lu *et al.* [50] proposed a layer costmaps approach to achieve context-sensitive navigation. In this method, the authors developed layered costmaps in contrast to traditional approaches that use a single costmap. Layered costmaps makes it possible to represent complex cost values in order to create navigation behaviors in many contexts. Each layer represents and tracks a particular type of obstacle or a constraint, and then a master costmap is computed to be used for path planning. In both [19, 50], laser-based people detection was implemented. Santana *et al.* [51] presented a human-aware navigation system for industrial mobile robots targeting cooperative intra-factory logistics scenario. The authors used cost functions to model assembly stations and operators in layered cost maps [50] to improve overall efficiency. Experimental results show that the overall efficiency of the human-robot teams improved due to predictability and comfort. Bordallo *et al.* [52] developed a multi-agent framework that utilizes counter-factual reasoning to infer and plan according to the movement intentions of goal-oriented agents. This interactive dynamics model is constructed on the notion of Hybrid Reciprocal Velocity Obstacles (HRVO). The authors validated their proposed framework on KUKA YouBots with experiments involving human-robot navigation, and the performance of the planner was shown in experimental results. The authors also validated their visual people tracking on openly available pedestrian datasets.

Many of the non-learning approaches discussed above work well without the need for any training data. However, most of the time, the context is a singleton, i.e., only a hallway behavior or an approach behavior is demonstrated. These non-learning approaches are well suited for low-level planning tasks, as demonstrated in this section. In the next section, we will discuss learning-based approaches that are widely used due to the advent of breakthroughs in machine learning and deep learning. Learning-based approaches to socially-aware navigation can be utilized in multiple contexts but require a lot of training data. Later in the next section, we will see how a hybrid approach utilizing both learning and non-learning approaches can address the challenges associated with a unified socially-aware navigation planner.

2.0.2 Learning approaches

While interacting with humans, a robot should be able to perceive its surroundings, predict intended human behavior, and act accordingly. Satake *et al.* [46] developed a model of approach behavior that anticipated the future behavior of people. SVMs (Support Vector Machines) were utilized to classify 2-second snippets of a trajectory into four behavior classes: fast-walking, idle-walking, wandering, and stopping. An evaluation of the system conducted with human users in a shopping mall found that people enjoyed interacting with the robot; the results also show that the robot's performance in successfully initiating a conversation has improved significantly. Feil-Seifer *et al.* [2] demonstrated that user state could be determined using autonomously sensed distance-based features and that such an approach resulted in more "leading," more "helpful," and more "attentive" than a standard navigation planner. In this system, Gaussian Mixture Models (GMMs) were utilized as opposed to SVM to capture the integrated multi-modality of interpersonal navigation data. The results from real-world experiments with autistic children show promising evidence that SAR systems might be used as therapeutic partners. We have used a similar approach to classify a person's navigation behavior from a set of human demonstrated actions [3], later a local planner using a learned social model using human-human navigation data was able to execute navigation behaviors that mimic human navigation actions in regards to social norms in various hallway maneuvers like *passing, meeting, walking together* [4]. Experimental results in simulation show that the trajectories generated by the planner confined to the social model along with the planner performance (time, distance traveled, etc.) in comparison with a traditional DWA planner.

Inverse Reinforcement Learning (IRL) has gained popularity in the machine learning community. IRL can be used to train human navigation behavior policy in order for the robot to emulate social behavior [15, 53–55]. Ramirez *et al.* [53] proposed two planners that use the layered costmap approach in combination with IRL based on the Markov Decision Process (MDP) utilizing expert demonstrations via teleoperation to solve the problem of "how and where to approach a person." Metrics such as time, trajectory length, Trajectory Difference Metric (TDM) - a modified version of Mean Square Error (MSE) were used to evaluate the approach behaviors. Kuderer *et al.* [54] proposed a feature-based maximum entropy IRL to achieve a navigation policy from teleoperated interactions with humans. The resulting policy maintained a probability distribution of the trajectories that lead the robot to avoid collisions with humans.

One interesting outcome is that the robot learned which sides the agents pass each other. The IRL method was implemented on a Pioneer robot that successfully navigated an office environment relying on an on-board laser sensor for human tracking. Ramon *et al.* [55] used Gaussian Process Inverse Reinforcement Learning (GPIRL) to train and evaluate a control policy on a publicly-available dataset [56]. Transfer learning using human navigation behaviors was achieved and presented an analysis of the performance of the learned policy compared to a heuristic cost-based proxemics method. Kim *et al.* [15] proposed an IRL based framework that can perform adaptive path planning in a hallway setting with people. The framework consists of three modules: a feature extractor, a learning module, and a path planning module that generated human-like trajectories in dynamic human environments. Performance of the proposed approach in comparison with DWA planner and human-generated trajectories was performed with the closest distance to humans, avoid distance, time to the goal as metrics.

Okal *et al.* [14] presented a Bayesian Inverse Reinforcement Learning (BIRL) based approach to achieving socially normative robot navigation using expert demonstrations. The authors of that work extend BIRL to include a flexible graph-based representation to capture relevant social task structure that relies on collections of sampled trajectories. The authors of the work conducted experiments both on a real robot and a pedestrian simulator; the results show that the approach was able to learn complex social normative behaviors like avoiding activity spaces. Experiments include both objective and subjective metrics such as path length, cumulative heading changes, smoothness of the trajectory, and the number of proxemic intrusions. Kretzschmar *et al.* [16] proposed a method to learn policies from demonstrations; it learns the model parameters of cooperative human navigation behavior that match the observed behavior concerning user-defined features. They used Hamiltonian Markov chain Monte Carlo sampling to compute the feature expectations. To adequately explore the space of trajectories, the method relied on the Voronoi graph of the environment from start to target position of the robot. The proposed model learned to mimic the behavior of pedestrians, i.e., it replicated a specific behavior that was taught by expert teleoperation. Furthermore, the authors performed cross-validation to validate the learned model and performed a Turing test to determine if the planned trajectories are human-like. Hamandi *et al.* [18] developed a novel approach using deep learning (LSTM) called DeepMoTIon, trained over well-known pedestrian surveillance data [56] to predict human velocities. This work used a trained model to achieve human-aware navigation, where the robots imitate humans to navigate in crowded environments safely. The experiments show that DeepMoTIon 24% reduction in time series-based bath deviation over the best approach and outperformed all the benchmarks related to human imitation. When compared to other models like the Social Force Model (SFM) and Generalized Reactive Planner (GRP), the proposed model reached the target location 100% of the time. Other metrics like Squared Path Difference (SPD), proximity were analyzed. Chen *et al.* [57] used deep reinforcement learning for motion planning that accounts for social norms when navigating. The robot observed and learned a policy continuously for an optimal path that will avoid collisions with humans and objects. The proposed approach is implemented on a mobile robot and shown to enable autonomous navigation in a dense human environment. When deployed in a human environment, the authors used metrics such as time to goal, the minimum separation distance.

Aroor *et al.* [58] formulated a Bayesian approach to develop an online global crowd model using a laser scanner. The model uses two new algorithms, CUSUM-A* (to track the spatiotemporal changes) and Risk-A* (to adjust for navigation cost due to interactions with humans), that rely on local observation to continuously update the crowd model. Evaluation in simulation shows that both algorithms generated paths that reduce proximity to humans, thereby increasing safety and inspire the human partners' trust. Other common metrics, such as mean success time, total time, distance, are reported between the two algorithms. Silva *et al.* [59] presented a Reinforcement Learning approach, where a robot learns a policy to share the effort required to avoid collision with a human. The results of the simulated experimental evaluation state that the robot mutually solves the collision avoidance problem with a human partner. Runtime analysis of both online and offline learning was reported to show that the simulation results can be realized in a real robot environment. Johnson *et al.* [60] presented a SAN implementation on a smart wheelchair robot using topological map abstraction, which lets the robot learn generalizable social norms. Furthermore, the authors compared their SAN planner with a baseline collision-free motion planner; the results show that a robot with SAN planner influenced the behavior of pedestrians around it by analyzing the pedestrian path and transitions.

The authors conjecture that legible robot navigation improves pedestrians' ability to predict the robot's future actions, making the pedestrians likely follow the social norm.

The work that exists deals only with a single context when addressing SAN, to the best of our knowledge, no method can handle multiple SAN contexts on the fly. Lu *et al.* work on layered costmaps is an approach that closely relates to the goals of USAN [50]. However, it has limitations such as maintaining multiple costmaps can be memory intensive, computation of a master costmap from a subset of costmaps for a particular context can be computationally expensive. Also, this approach does not include a method to sense a context autonomously; hence, costmaps associated with a specific context cannot be selected automatically. On the other hand, IRL based approaches are promising in a single context and can be trained to handle multi-context SAN but will require a lot of human training data for each context. Chapter 5 explains our approach towards a unified planner for socially-aware navigation that could overcome the said limitations.

2.1 Evaluation Methods

In this section, we list some of the SAN planner evaluation methods such as *In-person experiments*, *Naturalness of a trajectory*, *Performance metrics*, *Proxemic intrusions*, and *Observer experiments*.

Robot navigation planners are improving their roles in the real world. By investigating more than just performance-based metrics and the primary social challenges of these planners, we can validate the nature of a robot's social intelligence. Robot navigation in crowded spaces was demonstrated in a densely populated museum [34]. The robot successfully gave museum tours to visitors for six days. The researchers evaluated the method using performance metrics such as hours of continuous operation and average speed and evaluated the interaction with people by metric such as an increase in visitors count, number of web visitors [35].

Inverse reinforcement learning-based SAN uses human demonstrations of socially appropriate navigation [15]. Metrics such as closest distance to human, avoid distance, time to goal, comparison of a human-generated path, path generated by DWA planner, and their proposed IRL method were used. Althoff *et al.* [61] presented a probabilistic framework for reasoning about the safety of trajectories generated by robots in a dynamic environment with uncertain data about the moving objects in the environment. Probabilistic collision cost was used as a safety assessment cost metric that considers the motion of the moving objects in the environment. Human-robot proxemic preferences, using an HRI study, related to comfort, approach distance and approach angle were collected. A fuzzy-based human-robot proxemic model was built using the data collected from the HRI study, and the model's cross-validation results were reported [62].

The adoption of socially assistive robots can suffer if the robots do not follow social norms that people value [9]. One open question for socially-aware navigation is "*How do we evaluate social mapping/navigation techniques?*" [63]. We posit that SAN approaches related to proxemic rules can be evaluated in the following ways with some advantages and disadvantages associated with them:

In-person experiments: Participants can be asked to fill out questionnaires such as Negative Attitudes towards Robots Scale (NARS) [23] and the Godspeed Questionnaire Series (GQS) [22]. Examples: [64].

Naturalness of a trajectory: SAN methods that are model-based or learning-based approaches can compare trajectories to human-human interaction (HHI) data. Trajectory difference metric (TDM) [53], is a modified Mean Square Error (MSE), which evaluates every point of SAN trajectory to the closest point in the trajectory of HHI. For model-based methods, the probability that a trajectory confined to a particular interaction from HHI can be a great metric to determine if the robot confined to a particular social norm [4].

Performance metrics: These can include: the number of times the robot was able to generate a collision-free social path, time taken to reach a goal, efficiency of the trajectory, efficiency of the algorithms used, etc. Examples: [65].

Proxemic intrusions: Number of times the robot intruded social zones such as intimate, personal, social, and public space. Examples: [14].

Observer experiments: Measures can be obtained through outside observation of the human-robot dynamic as a navigation action occurs. Measures could be obtained either by presenting unaltered video (from side-view or overhead angles) and having a person rate the robot's behavior. An alternative approach, which can be used to control for a viewer's perception of a robot in these tasks, is to use Heider & Simmel-style videos [66] that preserve the spatial relationships of the agents performing navigation actions, but conceal whether those agents are humans, robots, or neither. Observers can then rate agents' behavior for several subjective factors related to the spatial behavior communicated by their movement [2].

In recent work on SAN, most of the validation methods are tailored to specific applications or methodologies used in the implementation of SAN. While such custom validation methods capture insights into the robot side of the interaction, they do not adequately assess the human's perception of the robot's social performance. Many factors influence our perception of socially-aware behavior [67]. Mobile robots need to take into account social conventions as a whole to be successful, socially intelligent agents [67]. The importance of a robot demonstrating human-like social conventions (safe and understandable behaviors) may play a more significant role in affecting a human's perception [67]. Taking these social conventions into account, we can examine our perception of a robot with SAN.

Understanding how humans perceive a robot's social intelligence may be crucial to

HRI research. When people interact, they orient themselves in a direction and distance that feels most comfortable for them. Research on human space has helped us understand how a robot invading different "zones" of space influences a human's perception of the robot [7]. Counting the number of proxemic intrusions might not validate a social planner as the theory of proxemics is complicated and depends on many demographic factors such as gender, cultural differences, etc.

Validation methods like comparing robot trajectories to that of human-generated trajectories are not always unique as there are uncontrolled factors like skills of humans operating the robot, etc. There is a need for standardized measurement tools for human-robot interaction research. Previous work in measurement tools such as as [22] measures five key concepts in HRI, namely *anthropomorphism*, *animacy*, *likeability*, *perceived intelligence*, and *perceived safety*. Another work [23] the authors developed scales that measure negative attitude towards robots specifically, *negative attitudes toward situations of interaction with robots*, *negative attitudes toward the social influence of robots*, and *negative attitudes toward emotions in interaction with robots*. Well-validated HRI surveys such as the Negative Attitudes towards Robots Scale (NARS) [23] and the Godspeed Questionnaire Series (GQS) [22] are missing the unique evaluation of a robots social intelligence. The GQS measures general intelligence. While there are other standardized questions, they get further away from our interest in measuring the perceived social intelligence of robots. The NARS measures the negative attitudes one might already have towards robots, which gets further away from our interest in robots' perceived social intelligence.

In Chapter 6, we discuss our PSI measurement tools and its validation by evaluating our socially-aware navigation planner.

2.2 Summary

Existing approaches to social navigation are quite diverse, and we have seen different approaches adopted by researchers. We broadly classify these recent trends into two categories, namely *Non-learning approach*, and *Learning approach* and the pros and cons of both the categories. The ones that use a non-learning approach to SAN include the utilization of cost functions, optimization tools, and constraint-based planning. The learning approach to SAN includes classic Machine Learning (ML) models, Inverse Reinforcement Learning (IRL), Reinforcement Learning (RL), and Deep Learning (DL) methods.

The algorithms presented above only work for a single navigation context; to the best of our knowledge, no method can handle multiple SAN contexts on the fly. Lu *et al.* work on layered costmaps is an approach that closely relates to the goals of USAN [50]. However, it has limitations such as maintaining multiple costmaps can be memory intensive, computation of a master costmap from a subset of costmaps for a particular context can be computationally expensive. Also, this approach does not include a method to sense a context autonomously; hence, costmaps associated with a specific context cannot be selected automatically. On the other hand, IRL based approaches are promising in a single context and can be trained to handle multi-context SAN but will require a lot of human training data for each context. Chapter 5 explains our approach towards a unified planner for socially-aware navigation that could overcome the said limitations.

Well-validated HRI surveys such as the Negative Attitudes towards Robots Scale (NARS) [23] and the Godspeed Questionnaire Series (GQS) [22] are missing the unique evaluation of a robots social intelligence. The GQS measures general intelligence. While there are other standardized questions, they get further away from our interest in measuring the perceived social intelligence of robots. The NARS measures the negative attitudes one might already have towards robots, which gets further away from our interest in robots' perceived social intelligence. Understanding the perceived social intelligence (PSI) of a robot is crucial when these robots are designed to work alongside humans. PSI allows researchers to improve the general social behavior of the robots; we designed PSI scales to measure the social intelligence of robots utilizing SAN. We provided motivation and a need for developing scales that measure the perceived social intelligence of robots. Later, in Chapter 6, we will see how we used the newly developed scales to measure the social intelligence of robots with SAN behavior. With an experiment design, we validate our hypothesis that *"Participants who observe a socially-aware navigation planner will perceive the robot as more socially intelligent than the one that is utilizing a traditional navigation planner."* Validation of such a hypothesis would only be possible with our newly developed measurement scales that address the gaps in existing methods [22, 23].

Chapter 3

Model-based SAN

This chapter details our prior work in SAN, a model-based socially-aware navigation planner.

- Our approach to a model-based SAN method.
- Experimental results of model-based approach.
- Limitation of model-based SAN.

In the previous chapter, we discussed the state-of-the-art approaches to SAN and broadly classified them into learning and non-learning based approaches. In this chapter, we will see our implementation of a model-based socially-aware navigation planner. We study if learning a model can help discriminate actions, which in turn can be used to select an appropriate behavior for a mobile robot. For human-human interaction, significant social and communicative information can be derived from the interpersonal distance between two or more people. If Human-Robot Interaction reflects this human-human interaction property, then the interpersonal distance between a human and a robot may contain similar social and communicative information. An effective robot's actions, including actions associated with interpersonal distance, must be suitable for a given social circumstance. We use autonomously detected environmental and distance-based features to develop such an interpersonal model using a Gaussian Mixture Model (GMM) and demonstrate that such a learned model can discriminate different human actions. This model-based, socially-aware navigation planner is a modification of *nav_core* package of Robot Operating System (ROS) in such a way that all the future trajectories are scored for appropriateness again a social model learned from human navigation data. Experimental validation of our approach in a simulation showed that the model-based SAN produced trajectories that are similar to the learned social model.

3.1 Model-based SAN

In this section, we revisit a real-time socially-aware navigation planner [2] by extending it to use on-board sensors, which helps a mobile robot to navigate alongside humans in a socially acceptable manner. This navigation planner is a modification of *nav_core* package of Robot Operating System (ROS), based upon earlier work and further modified to use only egocentric sensors. The planner can be utilized to provide safe as well as socially appropriate robot navigation. Interpersonal distance features between the robot and an interaction partner and features of the environment (such as hallways detected in real-time) are used to reason about the current state of an interaction. Gaussian Mixture Models (GMM) are trained over these features from human-human interaction demonstrations of various interaction scenarios as shown in Figure 3.1. This model is both used to discriminate different human actions related to their navigation behavior and to help in the trajectory selection process by providing a social-appropriateness score for a potential trajectory. In the next section, we will see how a learned model can discriminate navigation actions using Gaussian Discriminant Analysis (GDA).

3.1.1 Action Discrimination using GMM

Humans learn social conventions along many years of social interactions, robots may not have the same timeframe to learn, but they can learn rather quickly with a lot of training data. We collected human-human navigation data to construct a model using a Gaussian Mixture Model. For action discrimination in hallway setting, we extended the findings from [2] by creating a more robust and sophisticated model using more features to capture social interactions specific to hallway navigation behaviors. The model using distance-based features was successful in discriminating between a set of possible 2-agent social actions in real-time with an overall accuracy of 94.74% [3].

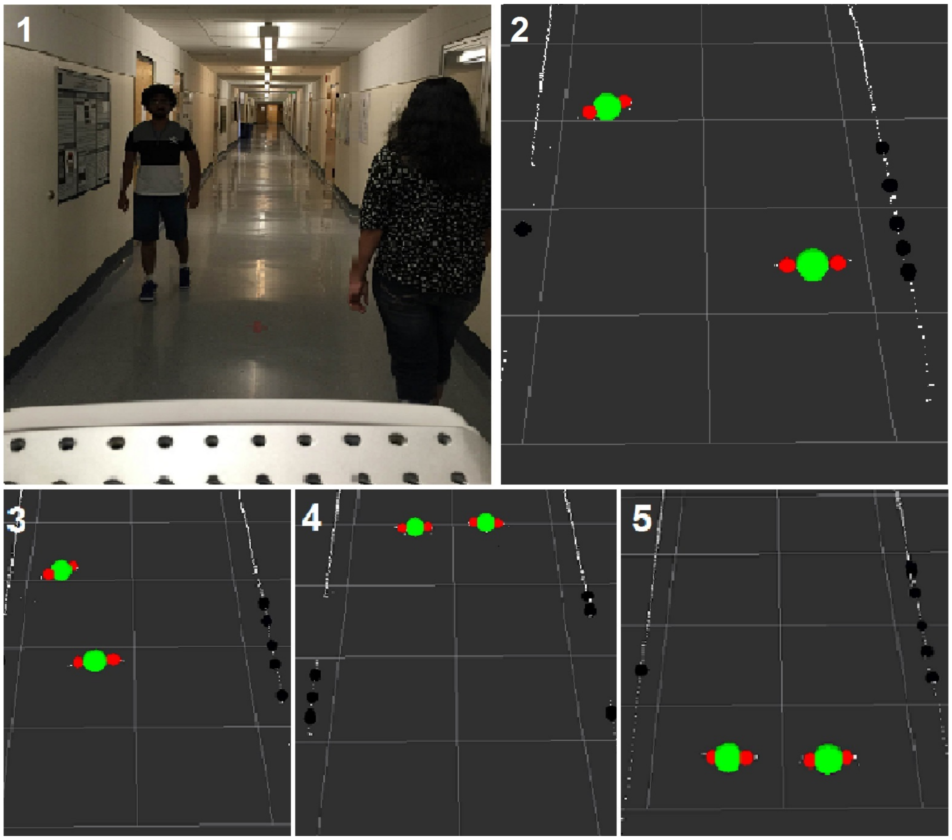


Figure 3.1: **1**: Shows the real picture of PR2 watching people. **2,3,4,5**: shows the RVIZ screen capture of PR2 tracking people in all the four scenarios, Passing, Meeting, Walking towards a goal and Walking away from a goal respectively.

Models of appropriate social behavior were learned from logged data of human-human interaction. We collected a training set of 24 recordings for four scenarios *(S1: people passing in a hallway, S2: people meeting in a hallway, S3: people walking together towards a goal and S4: people walking together away from a goal)* recorded from a floor-level 30m laser scanner, as shown in Figure 3.1. We identify people and relevant aspects in the environment (in this case, the location of hallways) and use relative distances as features of the model. Hallways were detected from the laser data using Hough Transforms to find parallel straight lines. People were found and tracked using laser scanner with the *leg_detector*, *people_msgs* ROS packages [68]. This model was

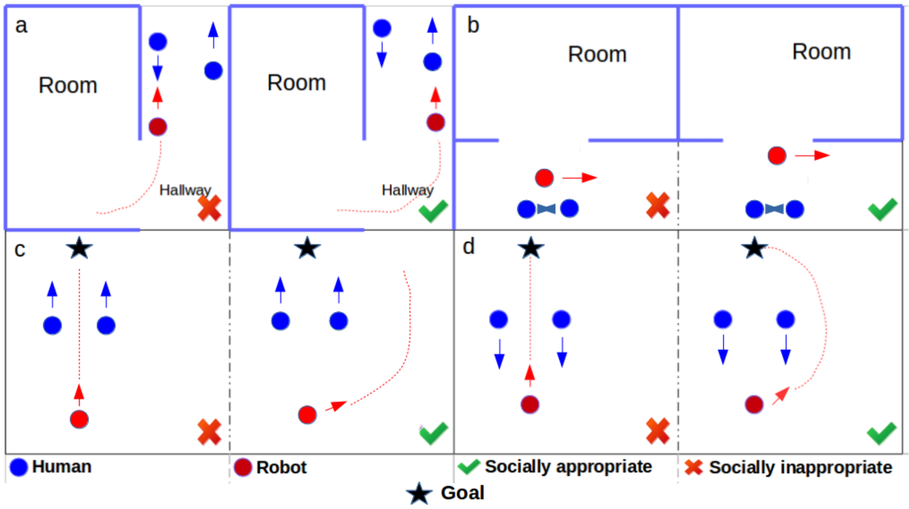


FIGURE 3.2: Some examples of socially appropriate/inappropriate navigation scenarios. a) People passing in a hallway, b) People meeting in a hallway, c) people walking towards a goal and d) people walking away from a goal.

then used to classify social action as one of the trained models.

We recorded the positions of two people exhibiting the given navigation behavior using a floor-level laser scanner. We used 20 of the 24 recordings as the training set for each scenario. We used the remaining four recordings as a test set for each scenario to test the model, and Figure 3.3 shows the results of the accuracy of our GMM model in all four scenarios in a confusion matrix. The training set of S1, S2, S3, and S4 consists of 3904 sample trajectory points on average, and the test set consists of 803 sample trajectory points on average. Figure 3.3 shows that the GMM model was able to distinguish between *people passing each other, people meeting, people walking towards a goal location, and people walking away from a goal location*. The probability of the observed data was much higher in the respective models when compared with the probability of the observed data in other models. In other models, the probability is either 0 or very small. We were able to mathematically discriminate between people

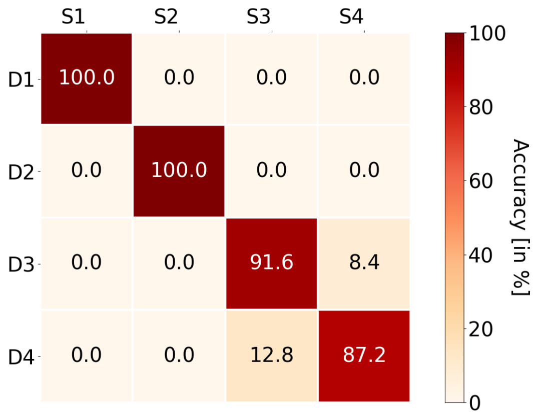


Figure 3.3: Confusion matrix of accuracies (as a heatmap) of the GMM in all the four scenarios.

walking with appropriate/inappropriate hallway behavior, people meeting situations, and we were able to identify if two people are walking together, as shown in figure 3.2. As the system can detect what people are doing, it can select its actions appropriately. For example, if the system identifies two people walking together, it will not pass between them provided if there is enough space around them. If not, it will have to ask permission to pass between them.

In the rest of the chapter, we discuss how a learned social model could account for social norms at a local planning stage.

3.1.2 Model-based Local Planner

Our model-based SAN planner has three main modules: the feature extractor, the SAN model [3], and the modified trajectory planner, as shown in Figure 3.4. The feature extractor module collects distance-based features that build our model, which

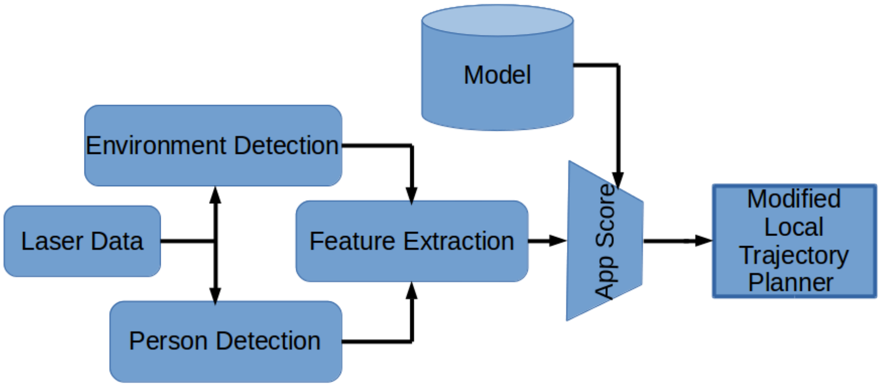


FIGURE 3.4: Block diagram explaining the overview of our approach

represents various social scenarios. The modified trajectory planner detects the current social scenario, scores every possible trajectory point to get an appropriateness score for that scenario. The trajectory point with the highest appropriateness score is chosen to execute a socially appropriate path to the goal. We chose a simulated PR2 for implementation of this model-based approach, but it could easily be implemented on any robot that is *nav_core* compliant and has egocentric sensing.

3.1.2.1 Feature Extractor

The goal of our model-based approach was to improve upon prior work [2], which utilized ubiquitous sensing to detect features in the environment. In this work, we only utilized on-board sensing to detect features relevant to the social scene. Information regarding where an agent was located with respect to the hallway, with respect to the other agents in a scene, and how much a given action has progressed was necessary in order to observe the social scenario. We hand-selected several of these distance-based features, used them to build the social model, and later at the local planning stage, scored for appropriateness for a given social scenario.

These features included:

- The normalized time.
- Distance traveled by the robot.
- The lateral position of the human with respect to the hallway
- The distance between the robot and human.
- The lateral distance between humans and the robot with respect to the hallway.

A SAN feature extractor node was developed that calculates a set of required features based on the possible future trajectory points the robot could select and published them for the developed model to analyze. The detection of obstacles, environment features such as hallways, people was achieved using a floor-level laser scanner on-board the PR2 robot as described in Section 3.1.1.

3.1.2.2 Model

Our model-based SAN planner utilizes multiple models of human-human social interaction to choose more socially-appropriate trajectories for a robot to reach a given goal. Human-human navigation data for the three out of four scenarios described in Section 3.1.1 were collected [3]. A model for each social scenario was constructed using a Gaussian Mixture Model (GMM), as discussed in Section 3.1.1. GMM was chosen over other methods as it can handle models that are not unimodal. Appropriateness can then be determined using Gaussian Discriminant Analysis (GDA). A given position's conformity to a given model can be derived from the Mahalanobis distance of a candidate point w to a given component k of the model:

$$\delta_M(w,k|\phi) = \frac{(w-\mu_{\phi(k)})^T \sum_{\phi(k)}^{-1}(w-\mu_{\phi(k)})}{2} \tag{3.1}$$

This term, Equation 3.1, is the standardized distance from an individual component of the GMM, taking into account the variance of that component. $\delta_M(w,k|\phi)$ can then be used to calculate the probability that w is part of the model:

$$p(w,k|\phi) = \frac{1}{(2\pi)^{n/2}|\sum_{\phi(k)}|^{1/2}} e^{-\delta_M(w,k|\phi)} \tag{3.2}$$

The probability that w conforms to a given model ϕ is the sum of the probabilities that it conforms to each of the k components of that model:

$$p^{'}(w|\phi) = \sum_k p(w,k|\phi) \tag{3.3}$$

The modified trajectory planner can then use this score of the appropriateness of a potential path given the scenario. The appropriateness score is then used to choose trajectories to reach a goal in a socially appropriate manner, the planner is discussed in next section.

3.1.2.3 Modified Local Trajectory Planner

The modified trajectory planner module is a modification of *nav_core* package of the ROS navigation stack, which operates by enumerating all possible trajectories, scoring them for the amount each trajectory moves toward the goal and the deviation of each trajectory from a globally-planned path [10]. While this does effectively navigate in complex and dynamic environments, no social information is considered. We have modified this planner to utilize conformity to a social model [3], built using human-human interaction data, in addition to these more utilitarian metrics.

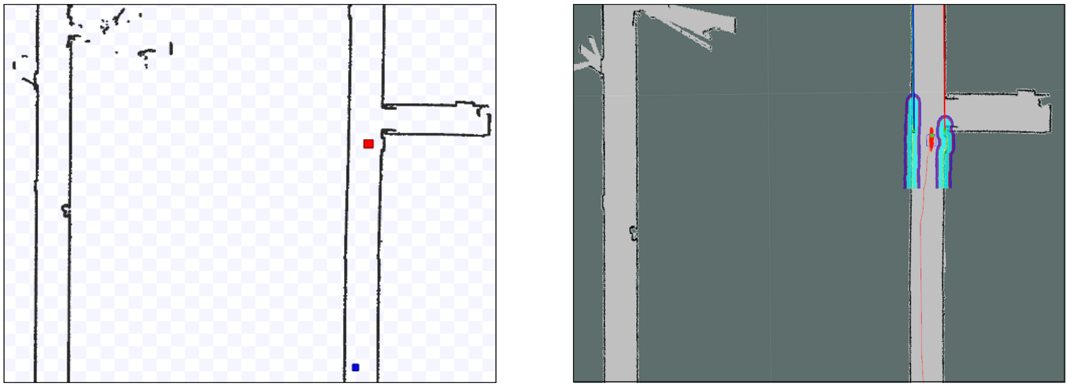
The above navigation planner [10] solves two problems to operate in an uncertain environment. First, by using *a priori* map of the environment, a *global* path plan is derived (usually by using wavefront planning to find a shortest possible path from point A to point B). However, while this plan will be the optimal solution, following it exactly will not account for dynamic obstacles in the scene. In this case, a *local* path planner utilizes egocentric sensor data from the robot (augmented by known obstacles from the *a priori* map of the environment). This local planner works by determining all possible future trajectory points. Our modification to this local planner allows scoring all these future trajectory points for social appropriateness, thus making the planner socially-aware of the ongoing interaction.

This local planner works by weighing candidate trajectories (v_x, v_y, v_θ) for progress toward the goal and adherence to the global plan. v_x and v_y are translational velocities along the robot's x and y axes respectively (non-holonomic robots have a v_y of zero), and v_θ represents the rotational velocity. In order to make the *nav_stack* planner more socially appropriate, we have modified this local planner to weigh trajectories based not only on the above metrics of path adherence and goal-directedness, but also adherence to models of human-human social interaction (see GDA approach in the previous section). The GMM based model [3] is used to score the appropriateness of the possible trajectories, and the trajectory with the highest score is chosen as the navigation behavior for a particular scenario. The modified planner will execute simultaneously socially appropriate and goal-directed behavior until the robot achieves its navigation objective. Thus, the modified trajectory planner plays a crucial role in driving the robot towards the goal in a socially appropriate manner.

Next section shows the evaluation of our model-based approach carried out on a simulated PR2 to demonstrate that our proposed method performed better than traditional navigation planner.

3.1.3 Results

To conduct the experiment and validate the architecture, we used a simulated PR2. The validity of the planner's operation under predictable conditions can be observed through testing in a simulated environment. The simulated environment made it possible to incorporate mobile obstacles as well as immobile obstacles into the system. Figures 3.5a and 3.5b show the screenshots of the simulator and visualization of the simulated environment, respectively. In the simulated environment, there were two robots, red and blue. The red robot utilized the SAN trajectory planner, and the blue robot was programmed to act according to human-human interaction norms for a given social scenario, as was recorded earlier. The simulated robot used data from a laser scanner for feature detection, localization, and obstacle avoidance. Since the simulated PR2 utilized identical sensing to the actual PR2, the navigation module of the simulated robot is compatible with the real PR2. Navigation planning began when the robot was given a navigation goal, and a social scenario to adhere to. The SAN planner then autonomously detected the features of the scene, such as hallway position and partner distance in the current simulator scene.



(A) Simulated environment showing both human agent (blue) and a PR2 robot (red) involving in a spatial interaction in a hallway scenario.

(B) Rviz screen capture showing robot navigation in the simulated environment and the detected hallways represented by line markers.

FIGURE 3.5: Simulated environment used in our model-based approach.

We evaluated the system by assessing the differences in task performance (in this case, time to complete a navigation action) between the traditional and the SAN planner.

Concerning the SAN planner, we wished to know if the SAN planner adhered to the social model better than the traditional planner. Seven different metrics have been defined to evaluate the performance of the system:

- m1: Robot task efficiency, time taken by the robot to navigate from point A to point B.
- m2: Human task efficiency, Human task efficiency is equally important as robot task efficiency and is often neglected. So, we will calculate not only robot task efficiency but also human task efficiency, which is the time taken for the human participant to navigate from point B to point A.
- m3: Combined task efficiency, time taken by both robot and human to navigate from point A to point B, point B to point A, respectively.
- m4: Distance covered by the robot to reach its goal location, which is point B.
- m5: Distance covered by the human participant to reach his goal location, which is point A.
- m6: The minimum distance the robot kept with the human participant during the course of its interaction with the human. Maintaining social distance while interacting with a human is an important factor in human-robot interaction studies [7].
- m7: The average probability that a trajectory is appropriate for a given situation.

Planner	Metric	Scenario I	Scenario II	Scenario III
SAN Planner	m1 (in sec)	49.4	100.18	43.67
	m2 (in sec)	49.5	49.79	42.28
	m3 (in sec)	98.96	149.99	86.02
	m4 (in m)	10.77	10.92	10.40
	m5 (in m)	8.86	9.76	10.49
	m6 (in m)	6.38	8.09	7.43
	m7	0.31	0.96	0.86
Traditional Planner	m1 (in sec)	50.12	45.17	42.28
	m2 (in sec)	50.4	41.6	45.78
	m3 (in sec)	100.53	86.81	92.63
	m4 (in m)	10.84	10.67	10.75
	m5 (in m)	8.91	9.80	11.39
	m6 (in m)	6.02	8.80	8.37
	m7	0.26	0.92	0.81

Table 3.1: Table showing a comparison of the observed results for the validation metrics in SAN planner and traditional planner

We conducted ten trials (command the simulated PR2 to a goal) for each of the three scenarios (*meeting, passing, and walking together towards a goal*) using the traditional navigation planner and ten trials with the SAN planner in the simulated environment; the results are shown in Table 3.1. To evaluate the performance of the approach, we compared the results from each planner using the metrics. Predictably, and most importantly, the data for metric 7 (the average probability that a trajectory is appropriate for a given situation) shows that for each of the three scenarios, the robot adhered more to the norms of human-human interaction with the SAN planner than with the traditional planner. For scenarios, I (*meeting*) and II (*passing*), the planner was more efficient. Which makes sense since the robot was actively weighing its behavior to conform to the social model. Additionally, the lower times for metric m1 demonstrate that the robot reaches its goal faster using the SAN planner, travelling a shorter distance (m4). This makes sense, since the robot uses the information from the social model that inherently predicts where the person will be over time, where the traditional planner does not. These results demonstrate that the SAN planner acts in a more socially appropriate way when compared to the traditional planner in *meeting* and *passing* scenarios without being significantly different for other performance metrics.

With this work [3, 4], we show that model-based SAN was able to generate trajectories for a given context, in this case, a hallway setting. However, scalability is an issue with such methods as they require large datasets for all the contexts the robot might encounter in a human environment. In the next chapter, we will see an optimization-based SAN planner that requires no training data.

3.2 Summary

In Chapter 2, we broadly classified existing SAN methods into two categories, namely, Learning-based methods and non-leaning based methods. In this chapter, we discussed our implementations of a model-based approach uisng human-human navigation data to construct a model using a Gaussian Mixture Model. The model using distance-based features was successful in discriminating between a set of possible 2-agent social actions in real-time with an overall accuracy of 94.74%. Our model-based SAN planner has three main modules: the feature extractor, the SAN model [3], and the modified trajectory planner, as shown in Figure 3.4. The feature extractor collects distance-based features that build our model, which represents various social scenarios. The modified trajectory planner detects the current social scenario, scores every possible trajectory point to get an appropriateness score for that scenario. The trajectory point with the highest appropriateness score is chosen to execute a socially appropriate path to the goal [4].

With our accomplished contributions, summarized above, we have the following capabilities that contribute towards dissertation:

- Local sensing of environmental features for distance-based interaction using onboard sensors as opposing to using external sensors.
- An understanding of the need for high-level decision-makers, in our case, a context classifier that can make decisions on social objectives involved in robot navigation.

In the next chapter, we will discuss an optimization-based socially-aware navigation planner that requires no training data.

Chapter 4

Optimization-based SAN

In this chapter, we will discuss the following topics:

- An optimization-based socially-aware navigation planner.
- Experimental results in multiple context such as *hallway*, *art gallery*, *waiting in a queue*, and *O-formations*.
- The role of optimization-based SAN planner in a USAN architecture.

In Chapter 2, we discussed the state-of-the-art approaches to SAN and broadly classified them into learning and non-learning based approaches. In this chapter, we will discuss our prior work on an optimization-based socially-aware navigation planner using a non-linear multi-objective optimization method [5, 8]. We used Pareto Concavity Elimination Transformation (PaCcET) to optimize a local trajectory planner by accounting for social norms using spatial information. Before looking at the technical details, the next section will provide some background information on PaCcET.

4.1 Background

In some cases, optimizing a single objective does not yield the desired performance, and therefore multiple objectives need to be considered when evaluating a policy's fitness. A standard method is to multiply a preset scalar value to each objective's fitness score and then add them all together. In some domains (such as planning from point A to point B), this standard method can lead to an optimal set of policies. In some complex domains (like planning from point A to point B with social constraints); however, this method will yield sub-optimal policies. A solution to this is to use a multi-objective tool, such as PaCcET, to evaluate policies on multiple objectives [6, 69] properly. PaCcET works by first obtaining an understanding of the solution space and finding the Pareto optimal solutions. Next PaCcET transforms the solution space and then compares each solution giving a single fitness score representative of how well each solution performed in the transformed space. At a high level, PaCcET works by transforming the Pareto front in the objective space in a way that it is forced to be convex. Transforming to objective space allows the linear combination of transformed objectives to find a new Pareto optimal point. PaCcET iteratively updates this transformation to always force non-explored areas of the Pareto front to be more highly valued than points dominated by the Pareto front or points that are on the explored areas of the Pareto front.

PaCcET has seen a variety of applications: it has been used to extend the life of a fuel cell in a hybrid turbine-fuel cell power generation system [70], the operation of the electrical grid on naval vessels [71], the coordination of multi-robot systems [72], and for the efficient operation of a distributed electrical microgrid [73], where a series of small power generation systems coordinate to meet the demands of consumers. In each of these applications, it has been shown that PaCcET functions at or above the solution quality of other techniques like NSGA-II or SPEA2 [6], with as low as one-tenth of the run-time.

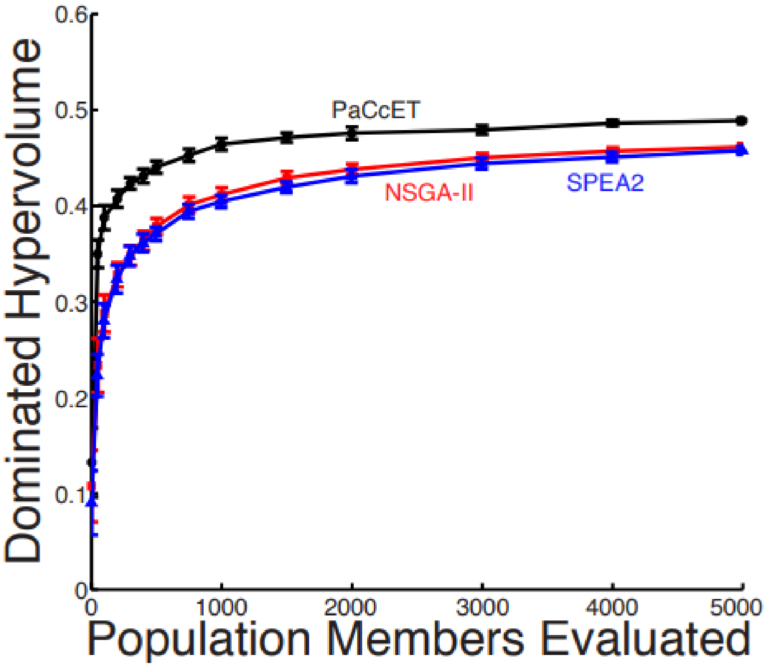


Figure 4.1: *PaCcET computational speed* - Percentage of hypervolume dominated in Kursawe's (KUR) problem in comparison with two successful multi-objective methods, SPEA2 and NSGA-II. This plot that PaCcET proceeds faster towards the Pareto front.

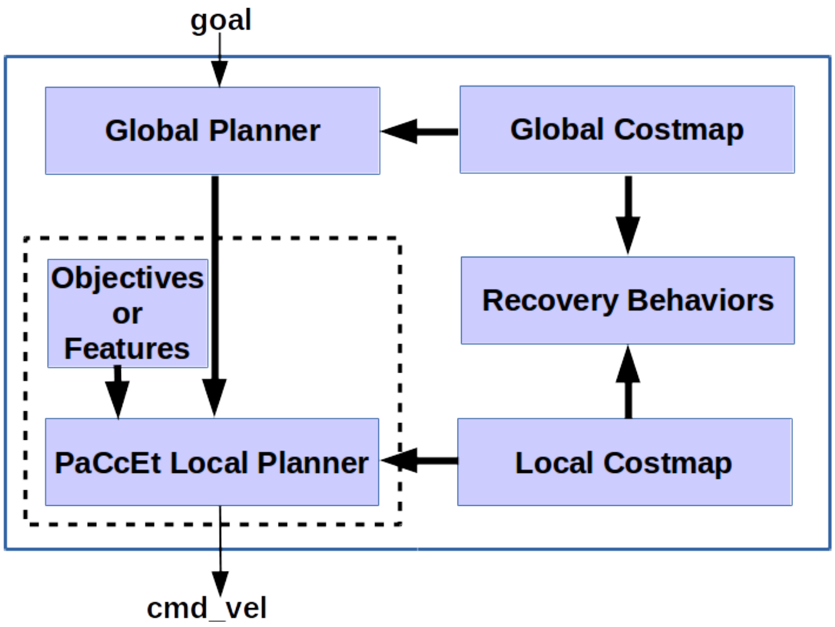


Figure 4.2: Block diagram of SAN using PaCcET, a modification of ROS navigation stack's local planner using PaCcET based non-linear optimization.

For this project, PaCcET was used over other multi-objective tools because of its computational speed [6], as shown in Figure 4.1. Another reason is that by assuming a spatial relationship among agents as linear, we might lose some crucial information. However, by using PaCcET as our optimization tool, we are not only considering non-linearities but also achieving solutions in real-time so that we can implement it on a robot. Figure 4.2 shows the overall high-level block diagram of the proposed approach. It is built on top of the well established ROS navigation stack by modifying the local planner to perform PaCcET transformation. The overall function of the local trajectory planner at each time step is to generate an array of possible future trajectory points and evaluate each future trajectory point based on a predefined feature set, as shown in Figure 4.3. In previous work [10], the features were assumed to have either no relationship or a simple linear relationship with one another; however, this is not always the case, and therefore we need to consider the possibility that the features are not only dependent on each other but also have nonlinear relationships.

4.2 Non-linear Multi-objective Optimization for Local Planning

The modified local planner [5, 8] using PaCcET [6] can be summarized as follows:

1. Discretely sample the robot control space.
2. Depending on the type of the robot, for each sampled velocity (V_x, V_y, and V_{theta}), perform a forward simulation from the robot's current state for a short duration to see what would happen if the sampled velocities were applied.
3. Score the trajectories based on metrics.
   (a) Score each trajectory from the previous step for metrics like distance to obstacles, distance to a goal, etc. Discard all the trajectories that lead to a collision in the environment.

   (b) For all the valid trajectories, calculate the social objective fitness scores like interpersonal distance and other social features and store all the valid trajectories.

4. Perform Pareto Concavity Elimination Transformation (PaCcET) on the stored trajectories to get a PaCcET fitness score and sort the trajectories from lowest to highest PaCcET fitness score.

5. For each time step, select the trajectory with the highest fitness score.

In the above working illustration of our low-level planner, step 3b is where the social objectives are accounted for while choosing the future valid trajectory points, as shown in Figure 4.3.

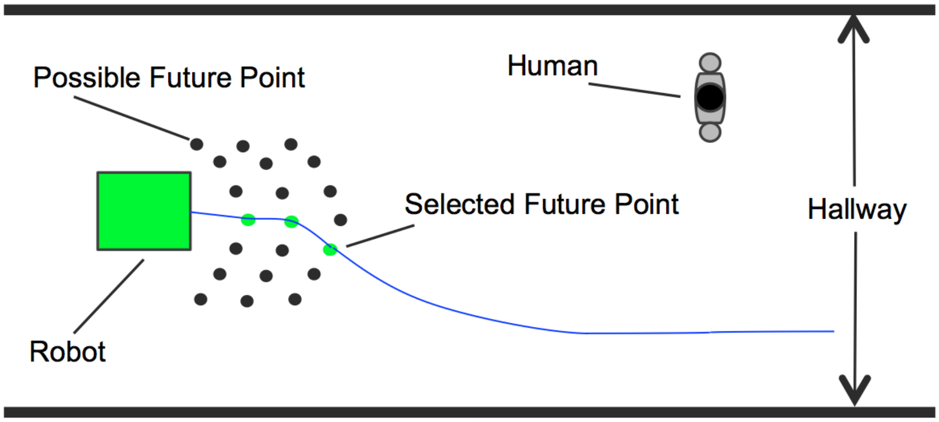


FIGURE 4.3: *Navigation Planner* - The navigation planner selects a short-term trajectory (green points represent potential trajectory end-points) from the pool of possible trajectories (black points), optimized for adherence to a long-term plan (blue line), obstacle avoidance, and progress toward a goal, and in the case of this paper, interpersonal distance.

In a traditional navigation planner, the features extracted were each assigned their own cost (e.g., the path distance cost (Δ_{path}), the length that the robot has already traveled, the goal distance cost (Δ_{goal}), the distance the robot is from the goal) [10]. The path distance will have a linear relationship with the goal distance since the change in one has a direct linear impact on the other. Once each feature has a cost associated with it, each cost is multiplied by a pre-tuned scalar and then added together, thus giving a linear combination, or weighted sum, in this case, the cost function shown in Equation 4.1. We can think of this cost function as an objective, where each possible future trajectory point has a cost or fitness associated with that objective. Since the purpose is to minimize the overall cost function, the planner will take the best path possible that minimizes the function, which in this case, will minimize both features.

$$\text{cost}(v_x, v_y, v_\theta) = \alpha(\Delta_{path}) + \beta(\Delta_{goal}) \tag{4.1}$$

More recently, this cost function has been adapted to include a heading difference ($\Delta_{heading}$) feature, and an occupancy (Δ_{occ}) cost feature, where the heading difference is the distance that the robot is from the global path and the occupancy cost is the cost used to keep the robot from hitting something. The same approach as in the previous cost function is taken in Equation 4.2. By taking a closer look at just the heading difference and how that might affect the path distance or the goal distance, it becomes less clear if there is only a linear relationship between the four. For example, if there is an obstacle in the robot's path, it will try and minimize goal distance by changing its heading, thus increasing the heading feature cost. In turn, this also increases the path distance cost, though this may or may not be linear.

$$\begin{aligned}\text{cost}(v_x, v_y, v_\theta) = \alpha(\Delta_{path}) + \beta(\Delta_{goal}) + \gamma(\Delta_{heading}) \\ + \delta(\Delta_{occ})\end{aligned} \tag{4.2}$$

Building upon prior work done in this area, we include socially-aware navigation features such as interpersonal distance (ID), distance from a group (GD), and distance from a social goal (SGD). As a way to dissuade the robot from getting too close to a human, a cost function was developed to penalize the robot at an exponential rate as the interpersonal distance decreases, as seen in Equation 4.3 (for every human in the interaction scenario). Although we could penalize the robot based on this at all times, it is not necessary if the interpersonal distance is so significant that it would not be considered as a socially inappropriate distance. Therefore the robot is only penalized if the interpersonal distance is less than or equal to 1.5 meters. The interpersonal distance threshold was chosen to be 1.5 meters to ensure that the robot remains in social space and does not invade the personal space of the person [7].

$$ID_f = e^{1/ID} \tag{4.3}$$

In order for the robot to not get too close to a group of people or not to get in between them, we penalized the robot based on GD whenever it is close to a group using Equation 4.4.

$$GD_f = e^{1/GD} \tag{4.4}$$

Contrary to Equation 4.3 and 4.4, Equation 4.5 is like a reward to get closer to a social goal rather than an actual goal. With this feature in place, the robot tends to reach a social goal for a particular scenario while still adhering to the final goal location. For example, the social goal for reaching the front of a desk when others are waiting in a line is the end of the line. So, the robot will reach the social goal first (end of the line) and eventually reaches the desk when it is the robot's turn.

$$SGD_f = e^{SGD} \tag{4.5}$$

Instead of adding these features cost into the previous cost function, Equation 4.2, we assume that its relationship with other features might be nonlinear and therefore gets treated as separate objectives. Since we know that the above cost function, Equation 4.2, works sufficiently enough from previous work [10], we can treat it as a single objective. Instead of optimizing just one objective, we need to optimize multiple objectives, hence our multi-objective approach. Using a multi-objective tool like PaCcET requires computational time, and since this is intended to work in real-time, any chance to improve the computation time should be utilized. Treating the first four features used in the previous cost equation 4.2 as a single objective not only speeds up this process but, in turn, allows for the possibility to add even more features to our local trajectory planner. Using PaCcET to do the multi-objective transformations, we essentially get a new cost function with a PaCcET fitness denoted by P_f, which was modeled under the assumption of nonlinear relationships between the objectives. Equation 4.6 shows how P_f is a transformation function dependent on multiple variables.

$$P_f = T_f(Obj_1, Obj_2, ..., Obj_n) \tag{4.6}$$

In this work, we are only interested in objectives like interpersonal distance, distance from a group, and distance from the social goal. The first objective is the original cost function (Equation 4.2), which is the linear combination of the path distance, goal distance, heading difference, and occupancy cost. The remaining objectives are the social features, such as interpersonal distance, distance from the social goal. Equation 4.7 shows the PaCcET fitness function with our proposed objectives.

$$P_f = T_f(cost(v_x, v_y, v_\theta), ID_{f1}, .., ID_{fn}, GD_f, SGD_f) \tag{4.7}$$

where, ID_{f1}, .. ID_{fn} are the cost functions associated with interpersonal distance between n people and the robot.

4.2.1 PaCcET Local Planner Algorithms

In this section, we will see algorithms that makeup our PaCcET-based local planner. Algorithm 1 shows the primary functions of the local trajectory planner and how the future trajectory points were stored to be used with PaCcET. The trajectory planner is called every time step, which in this case, is every 0.1 second. Once the trajectory planner is called, the `Transform_Human_State` function is called to transform human pose to the robot's Odom reference frame, which allows the interpersonal distance corresponding to each possible trajectory to be calculated in the `Generate_Trajectory` function. Now there are two methods of calculating the possible trajectories. The first is assuming that the robot can only move forward, backward, and turn. To produce the possible trajectories for this physical setup, we loop through every combination of a sample of linear velocities (V_x) and angular velocities (V_θ) to generate trajectories (For a holonomic robot, a slight change in V_y is also used to generate possible trajectories). Once a trajectory is created, we determine if it is valid based on the constraints for the first objective. For example, trajectories that would make the robot hit a wall, obstacle, or human are not considered strong trajectories and therefore, will not be stored in the `Store_Trajectory` function. By not storing these invalid trajectories, the speed at which PaCcET runs can be improved.

The second method assumes that the robot is capable of holonomic movement can translate with any V_x, V_y, V_θ that are less than velocity limits. Given these movements, we again loop through all the possible movements given the predefined number of V_x

Algorithm 1: `Local Trajectory Planner Algorithm`. The trajectory planner generates multiple trajectories (T) given a number of V_x samples and V_θ samples and calculates the independent cost for each feature. The cost for each feature is based on the robots sensing of the human's state (H_s) and the robot's state (R_s). At the end of a time step the best trajectory ($\mathcal{T}_\mathcal{B}$) is returned.

```
Input: $V_x$ samples, $V_\theta$ samples, $H_s$, $R_s$
Output: $Best\_Trajectory$ ($\mathcal{T}_\mathcal{B}$)
for Each time step do
    $Transform\_Human\_State(H_s, R_s)$
    for Each $V_x$ do
        $\mathcal{T} \leftarrow Generate\_Trajectory(T,\ H_s)$
        if valid trajectory then
            $Store\_Trajectory(\mathcal{T})$
        for Each $V_\theta$ do
            $\mathcal{T} \leftarrow Generate\_Trajectory(T,\ H_s)$
            if Valid Trajectory then
                $Store\_Trajectory(\mathcal{T})$
    if Holonomic Robot then
        $\mathcal{T} \leftarrow Generate\_Trajectory(T,\ H_s)$
        if Valid Trajectory then
            $Store\_Trajectory(\mathcal{T})$
    $Run\_PaCcET(\mathbb{T})$
    Return $\mathcal{T}_\mathcal{B}$
```

samples, V_y samples, and V_θ samples. Again, if the trajectories are valid, they are stored. Once all the valid trajectories are stored for all possible movements, the `Run_PaCcET` function runs, giving back the best possible trajectory, ($\mathcal{T}_\mathcal{B}$), based on its multi-objective transformation process.

In order to run a multi-objective tool like PaCcET, each objective's fitness needs to be calculated. Algorithm 2 details the `Generate_Trajectory` function from Algorithm 1. The first function that needs to be performed is the `Calculate_State` function as the robot's position, and velocity are used to determine the fitness values for the objectives. Using the state information the `Compute_Path_Dist`, `Compute_Goal_Dist`, `Compute_Occ_Cost`, and `Compute_Heading_Diff` functions are used to calculate the fitness values associated with the four pieces of the first objective. Using the computed fitness values, the first objective's fitness is calculated by the `Compute_Cost` function. Distance-based features like interpersonal distance of each person, group distance, and social goal distance are calculated, as shown in Algorithm 2 lines 8 - 10. Once all the objectives have their fitness values, the trajectory along with the fitness values is returned to the local trajectory planner algorithm, which saves all the valid trajectories and calls PaCcET Algorithm 3 to perform optimization.

Algorithm 2: `Generate Trajectory Algorithm`. The generate trajectory function take in an instance of a trajectory (T) and the humans' state (H_s) to compute the cost function for each feature. The trajectory (T) is then returned to the local trajectory planner.

```
   Input: $T$, $H_s$
   Output: $T$
1  $\mathcal{S} \leftarrow Calculate\_State(\mathcal{T})$
2  $path\_dist \leftarrow Compute\_Path\_Dist(\mathcal{S})$
3  $goal\_dist \leftarrow Compute\_Goal\_Dist(\mathcal{S})$
4  $occ\_cost \leftarrow Compute\_Occ\_Cost(\mathcal{S})$
5  $heading\_diff \leftarrow Compute\_Heading\_Diff(\mathcal{S})$
6  $cost \leftarrow Compute\_Cost(path\_dist,\ goal\_dist,\ occ\_cost,\ heading\_diff)$
7  for $Each\ person$ do
8      $\mathcal{ID} \leftarrow Calculate\_Interpersonal\_Distance(H_s,\ \mathcal{S})$
9  $\mathcal{GD} \leftarrow Calculate\_Group\_Distance(H_s,\ \mathcal{S})$
10 $\mathcal{SGD} \leftarrow Calculate\_Social\_Goal\_Distance(H_s,\ \mathcal{S})$
11 $Return\ Trajectory(\mathcal{T})$
```

At the end of Algorithm 1, all the valid trajectories have been stored along with their objective fitness scores in a vector of type trajectory. Algorithm 3 details the primary functions for determining a single fitness value from multiple objectives. In order to run PaCcET, the objectives for each trajectory must be stored in a vector of type double, which is done in the `Store_Objectives` function. Before running PaCcET's primary functions, an instance of PaCcET must be created. Next, the solution space and Pareto front are created by giving each trajectory to the `Pareto_Check` function. Now that the Pareto front and its geometry has been calculated, PaCcET can transform the solution space and give a single fitness value for each trajectory in the `Compute_PaCcET_Fitness` function. Once each trajectory has its PaCcET fitness, they are sorted from best to worst in the `Sort_Trajectories` function, which allows the function to not only ascertain the best trajectory easily but is also useful for debugging purposes. Algorithm 3 concludes by returning the best trajectory to the local trajectory planner algorithm.

Algorithm 3: `PaCcET Alogrithm`. PaCcET (P),takes in the vector of valid possible trajectories $\mathbb{T}$ to compute the multi-objective space and the PaCcET fitness (P_f) for each trajectory.

```
Input: $\mathbb{T}$
Output: $\mathcal{T}_{\mathcal{B}}$
for Each trajectory do
    Store_Objectives($\mathcal{T}$)
$\mathcal{P} \leftarrow$ Initialize_PaCcET()
for Each trajectory do
    Pareto_Check($\mathcal{T}$)
for Each trajectory do
    $P_f \leftarrow$ Compute_PaCcET_Fitness($\mathcal{T}$)
Sort_Trajectories($\mathbb{T}$)
Return $\mathcal{T}_{\mathcal{B}}$
```

An identification of social goals, such as: *end of the line* for *waiting in a queue* context and *a suitable position on the circle* for *O-formation*, is crucial for the PaCcET planner to exhibit appropriate social navigation behavior. An optimization planner is only as good as its objectives, so it is important to identify and efficiently compute these objectives (social goal). In the next section, we will see how our proposed system can identify social goals using simple mathematical modeling.

4.2.2 Social Goal Computation

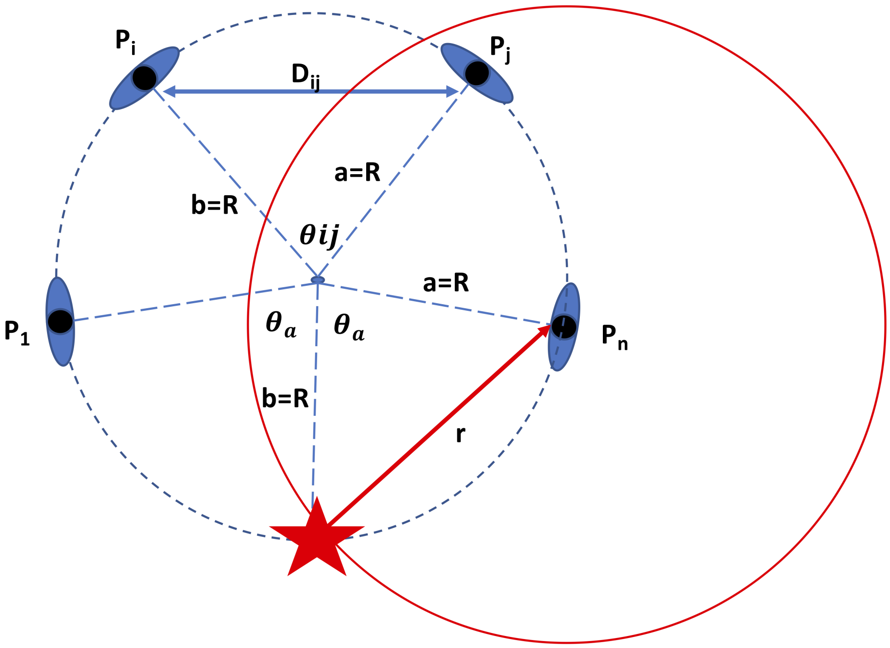


FIGURE 4.4: Figure illustrating the computation of social goal in O-formation scenario. The red star represents the social goal.

Computing the social goal location is important because often, the actual goal location may not be an appropriate location for interaction, and explicitly commanding the social goal would not be possible or highly variable. A social goal can be defined as an appropriate location for a robot to involve in human-robot interaction. For example, in a front desk-like scenario, the end of the line can be considered as a social goal. For this work, the social goals for each interaction scenario were geometrically computed, for *O-formation scenario (joining a group)*, we fit a circle with the people in a group and find a socially appropriate spot to join the group as shown in Figure 4.4.

We find angle made by every person with the center of the formed circle using the law of cosines, equation 4.8 as shown below:

$$c^2 = a^2 + b^2 - 2ab * cos(\theta) \tag{4.8}$$

$$\theta_{ij} = cos^{-1}[(2R^2 - D_{ij})/2R^2] \tag{4.9}$$

Where D_{ij} is the Euclidean distance between person i, j, and R is the radius of the circle formed by all the people in the group. Out of all the θ_{ij}'s, we pick one half of the widest angle as approach angle denoted by θ_a. Now, joining a group problem (O-formation) boils doing to finding the intersection of two circles, one formed by the people in the group and the other formed in the wide-open sector with the center as the locations of either of the people making the widest sector. The equations of the two circles to solve for are given as follows:

$$(x - h)^2 + (y - k)^2 = R^2 \tag{4.10}$$

$$(x - h_p)^2 + (y - k_p)^2 = r^2 \tag{4.11}$$

Where, (h, k) is the center of the circle formed by the group of people, (h_p, k_p) is the location of one of the person that formed the widest sector. The radius r in Equation 4.11 is obtained by solving for c in Equation 4.8 where, $\theta = \theta_a$, a and b equals R, radius of the group formation. There are two solutions when solving Equations 4.10 and 4.11, we further filter one social goal from the two solutions.

Similarly, we can fit a straight line as shown in Figure 4.5 for *waiting in a queue scenario*, and social goal location would be the end of the line considering personal space of the last person in the line. Hence, in this case, the solution boils down to solving for the intersection of a line and a circle.

The equation of the line formed by the people can be found by fitting a line of form $y = mx + c$ with the humans' locations. The circle formed using the last person's location as the center and a comfortable distance that the robot should maintain

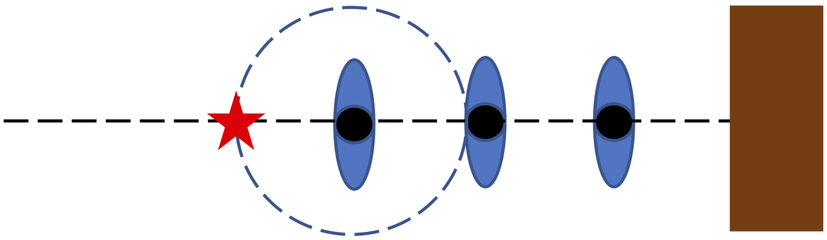
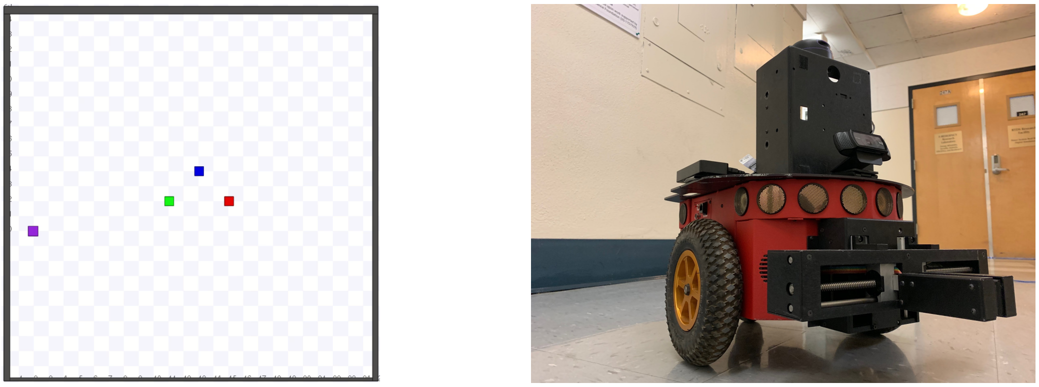


FIGURE 4.5: Figure illustrating the computation of social goal in waiting in a queue scenario. The red star represents the social goal.

around the last person as the radius is of the form $(x-k)^2+(y-h)^2=r^2$. The two solutions to the line and circle intersection can be obtained using quadratic roots, and the social goal is further filtered to the solution farthest to the actual goal (desk).

For the *art gallery* context where for appropriate social behavior, the robot needs to avoid activity space (space between the artwork and the visitor). The objective of avoiding the activity space is a social goal, which can be computed by fitting a straight line with the human and artwork locations as two points.

4.3 Results



(A) Stage, a 2D simulator.

(B) Upgraded Pioneer 3DX robot.

FIGURE 4.6: Platforms used to validate our proposed PaCcET local planner and associated USAN architecture.

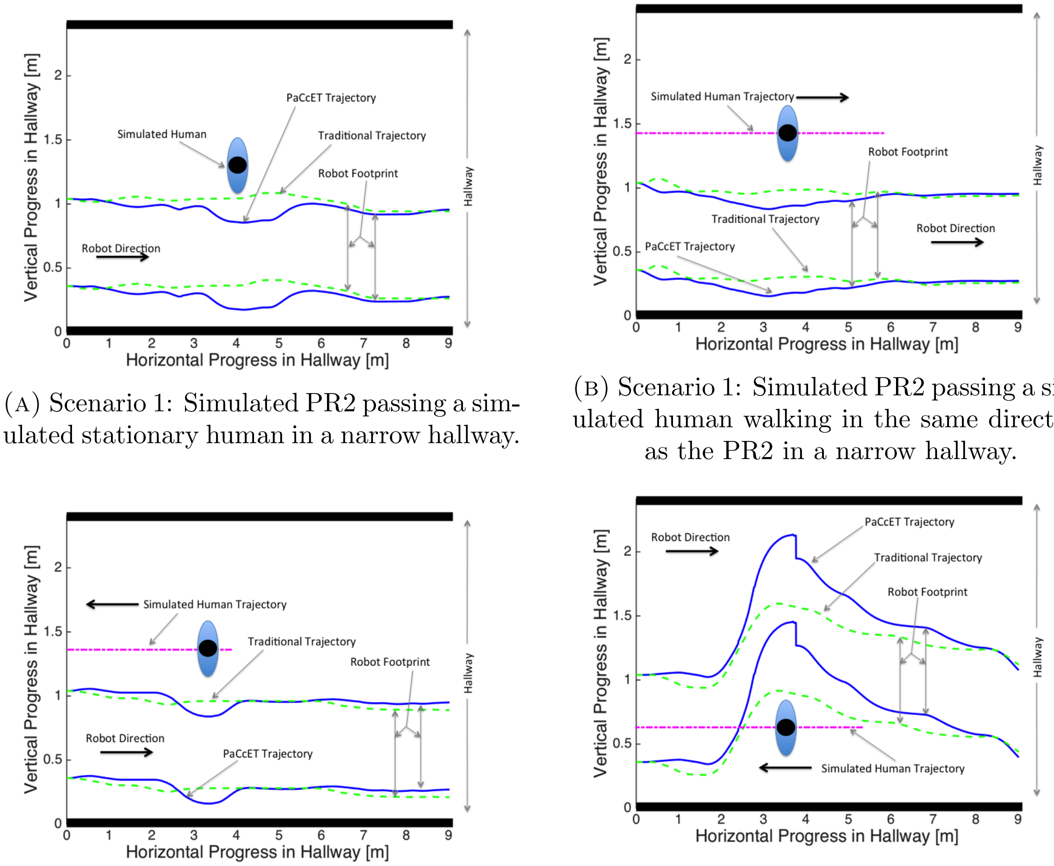
In order to validate our proposed approach, we considered four different scenarios, namely, *a hallway*, *art gallery*, *forming a group*, and *waiting in a queue*. We validated our approach both in simulation and on a real robot to see if our proposed PaCcET local planner can account for social norms in various interaction contexts. For example, in an *art gallery* scenario, we expect the robot (with SAN planner) to account for activity space by not traversing the activity space between the human and artwork. Similarly, in a *waiting in a line* scenario, we expect the robot (with SAN planner) not to cut the line and join the line instead.

For simulation experiments, we used the 2D simulator, Stage [74] on a machine with an Intel 6th-generation i7 processor @3.4 GHz, 32 GB of RAM. The simulated environment for each hallway experiment [5] was the second-floor hallway of the Scrugham Engineering and Mines building at the University of Nevada, Reno. The map of the building used in the simulation was built using the gmapping package for SLAM on the PR2. The simulated PR2 is comparable to the real-world PR2 for sensing and movement capabilities and is using AMCL for localization on the map. The simulated PR2 used a 30-meter range laser scanner that is identical to the real PR2 robot's laser scanner's capability—the humans in the simulation exhibit very simple motion behaviors. Follow-up scenarios are simulated in a 25m x 25m open space in the stage environment, as shown in Figure 4.6a. The PR2 robot was simulated to run both traditional planners and our modified PaCcET based planner. In Figure 4.6a, the purple agent is the simulated PR2, and the rest of the agents are humans formed as a group.

For real-world validation, we used an upgraded Pioneer 3DX platform, shown in Figure 4.6b. The Pioneer robot that we used was equipped with an RPLIDAR-A3, a 30-meter range laser scanner with a 360°field of view and a webcam as sensors for perception. For detecting people using a laser scanner, we used the people detector developed by Leigh *et al.* [75]. The robot's computational unit was also upgraded to a laptop with an Intel Core i7-7700HQ CPU @ 2.80 GHz x 8 processors, 16 GB RAM, and GeForce GTX 1050 Ti GPU with 4GB of memory. The pioneer robot also uses AMCL for localization on the map. The *hallway* scenarios on the real robot were validated in the same location as the simulation experiments. *Art gallery*, *waiting in line*, and *group formation* scenarios were validated in the lobby area (7m x 7m approx.) situated on the first floor of the Scrugham Engineering and Mines building of University of Nevada, Reno.

4.3.1 Simple Context - Two Objectives

In the first experiment, the robot was tasked with getting to a goal while passing close to a static simulated human. Figure 4.7a shows that when using the traditional planner, the robot made sure to avoid a collision with the simulated human, but did not consider any social distance. The same will be the case for the other experiments as well since the traditional planner does not consider interpersonal distance into its cost function. The PaCcET-based planner did consider interpersonal distance, and therefore the robot deviated from a more straight-lined path as a way to satisfy the second objective (interpersonal distance). Once the threshold for the interpersonal distance was no longer an issue, the robot only needed to minimize the first objective;



(A) Scenario 1: Simulated PR2 passing a simulated stationary human in a narrow hallway.

(B) Scenario 1: Simulated PR2 passing a simulated human walking in the same direction as the PR2 in a narrow hallway.

(C) Scenario 1: Simulated PR2 encounters a simulated human passing on the appropriate side of a narrow hallway.

(D) Scenario 1: Simulated PR2 encounters a simulated human walking on the inappropriate side of a narrow hallway in opposite direction.

FIGURE 4.7: Simple two objective optimization scenarios with a single simulated human. The simulated human trajectory is shown using a dotted magenta line, trajectory of traditional planner is represented using a dotted green lines (two lines to represent footprint of the simulated PR2) and PaCcET based SAN trajectory is represented using a solid blue line (two lines to represent footprint of the simulated PR2). Direction of simulated human and PR2 are represented using arrows.

therefore, returning to a straight-line path. It is worth noting that in all the experiments conducted, the robot also considered a wall as an obstacle and was required to disregard trajectories that would lead to a collision with the wall, which is why the robot did not deviate from the global trajectory even more.

We developed the second experiment to mimic a passing scenario where the robot has a set goal but needs to pass by a simulated human who is traveling much slower in the same direction. Figure 4.7b shows that with the traditional trajectory planner, it merely made sure that a collision would not take place as it tried to minimize its cost function. The PaCcET-based planner deviated from its global trajectory in order to consider the interpersonal distance objective, then returned to the global trajectory once the threshold for the interpersonal distance was no longer an issue.

Similar to the previous experiment, the third experiment involves both the simulated human and robot moving; however, in this case, the simulated human is now moving at a normal walking speed in the opposite direction of the robot—the robot and simulated human pass close to one another but not close enough to cause a collision. Figure 4.7c shows that the traditional trajectory planner altered its path ever so slightly to ensure that a collision would not happen, where as the PaCcET-based trajectory planner not only ensured that a collision would not take place but also considered interpersonal distance and provided the simulated human with additional space while passing.

The previous experiments show that when using a PaCcET-based trajectory planner interpersonal distance can be considered in selecting a local trajectory in both static and dynamic conditions where a collision is not imminent; however, the case of a collision that would occur unless either the simulated human or the robot moves out of the way also needs to be considered. This experiment considers a simulated human who is not paying attention or unwilling to change their course and walking directly

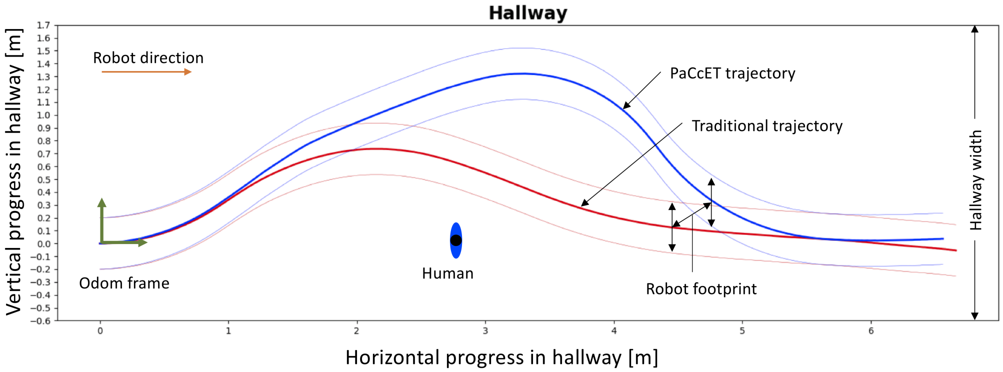


FIGURE 4.8: Scenario 1: (real-world interaction) Pioneer robot encounters a stationary human standing in the path of the robot in a hallway.

towards the robot. Figure 4.7d shows that the traditional trajectory planner was successful at avoiding the collision as expected; however, it did so while minimizing its cost function as much as possible, which caused the robot to get very close to the simulated human. When using the PaCcET-based trajectory planner, the robot not only avoided the collision but also gave the simulated human additional space to satisfy the interpersonal distance objective. It is worth noting that once the interpersonal distance threshold was no longer an issue, the robot used its holonomic movement for a short time as a way to quickly minimize the heading difference portion of the original cost function objective.

We extended hallway scenarios to the real-world by implementing our proposed approach on a Pioneer 3DX robot and validating it in both static and dynamic environments. Figure 4.8 shows a real-world hallway situation where a human is standing in the path of a robot that is attempting to go down the hallway. The robot, when using the traditional planner, treated the human as a mere obstacle and avoided a collision but violated the personal space rule of the human. On the other hand, our approach

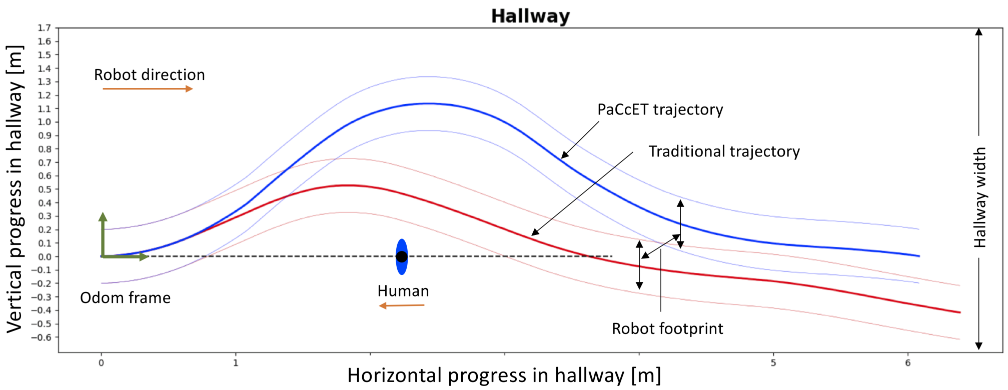


Figure 4.9: Scenario 1: (real-world interaction) Pioneer robot encounter a human walking in the opposite direction and on the side of the hallway.

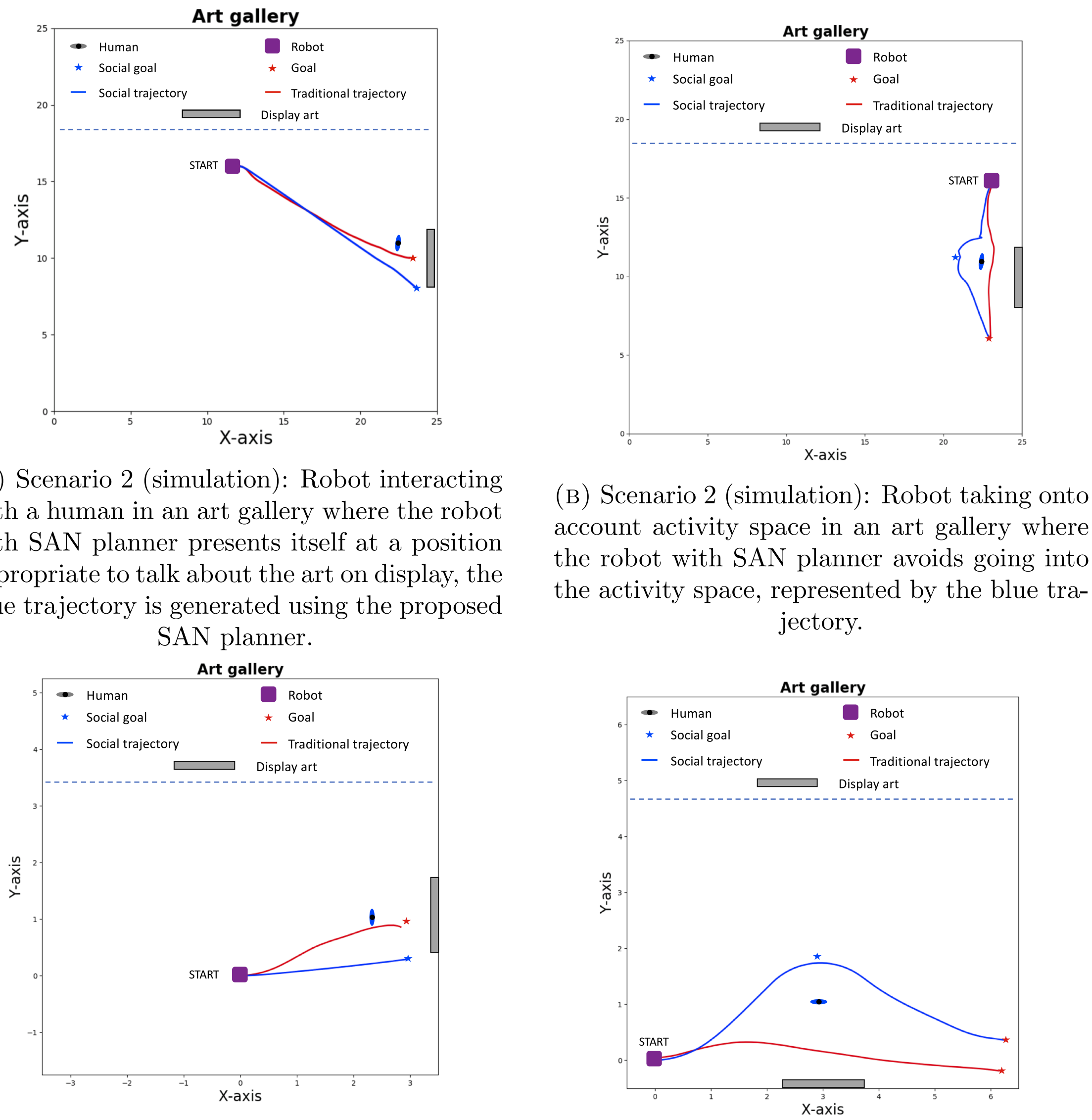
using PaCcET-based local planning considered the stationary human's personal space using interpersonal distance objective and deviated from the global trajectory in such a way that the personal space rule is obeyed. In Figure 4.8, the blue trajectory is generated by our proposed approach, and the traditional approach generates the red trajectory.

Figure 4.9 shows a real-world hallway interaction like the previous one, but in this case, the human is moving as opposed to a static human. In this experiment, the human is walking in the opposite direction of the robot and also on the wrong side of the hallway. As one can observe, the traditional planner (red trajectory) managed to avoid a collision with the human but went very close to the person, thereby intruding into the human's personal space. Our proposed approach not only avoided a collision but also maintain a safe distance while trying to avoid the human walking on the wrong side of the hallway. Unlike the PR2, Pioneer is a non-holonomic robot; hence, the holonomic behavior, as seen in Figure 4.7d, is not seen in the real-world interaction. It is worth noting that in Figures 4.8 and 4.9, the robot with PaCcET trajectory planner showed signs of legibility of movements. In both these cases, the efforts of the robot trying to clear the human's personal space are clearly seen using our method as opposed to the traditional planner.

4.3.2 Complex Contexts - Multiple Objectives

In our prior work, Section 4.3.1, we showed that by just considering one social feature, i.e., interpersonal distance, our approach was able to account for personal space while navigating a hallway (with different maneuvers of a human partner) [5]. In this section, we will see the results of our approach applied to complex social scenarios like *art gallery interactions* (Figure 4.10), *waiting in a line* (Figure 4.11) and *joining a group of people* (Figure 4.12) in simulation and on a Pioneer mobile robot. The context is manually selected and given to the PaCcET local planner, and the results of the planner executing socially appropriate trajectories in multiple contexts, both in simulation and in real-world situations, are shown. These scenarios are representative of both human-human and human-environment interactions that occur in normal social discourse.

Figure 4.10a shows the behavior of our social planner and traditional planner in an *art gallery* situation (three objectives) in simulation. We considered an *art gallery scenario*, but it can be generalized to other similar scenarios like a tour guide robot in a museum or an attraction. For this scenario, we staged a human-robot interaction



(A) Scenario 2 (simulation): Robot interacting with a human in an art gallery where the robot with SAN planner presents itself at a position appropriate to talk about the art on display, the blue trajectory is generated using the proposed SAN planner.

(B) Scenario 2 (simulation): Robot taking onto account activity space in an art gallery where the robot with SAN planner avoids going into the activity space, represented by the blue trajectory.

(C) Scenario 2 (real-world): Pioneer robot interacting with a human in an art gallery where the robot with SAN planner presents itself at a position appropriate to talk about the art on display, the blue trajectory is generated using the proposed SAN planner.

(D) Scenario 2 (real-world): Pioneer robot taking onto account activity space in an art gallery where the robot with SAN planner avoids going into the activity space, represented by the blue trajectory.

FIGURE 4.10: Validation results of Scenario 2 (art gallery) in both simulation and real-world.

consisting of a robot presenting a piece of art (hanging to a wall) to a human standing nearby. Both the traditional planner (red line) and the SAN planner (blue line) were given the same goal (represented as a red star) and start (START) locations. The traditional planner steered the robot to the goal location, cutting the standing person from the back (inappropriate). On the other hand, the SAN planner steered the robot to a location that is appropriate to present the details of the art to the human (social goal). The SAN planner approached the social goal, leaving enough personal space based on the interpersonal distance feature.

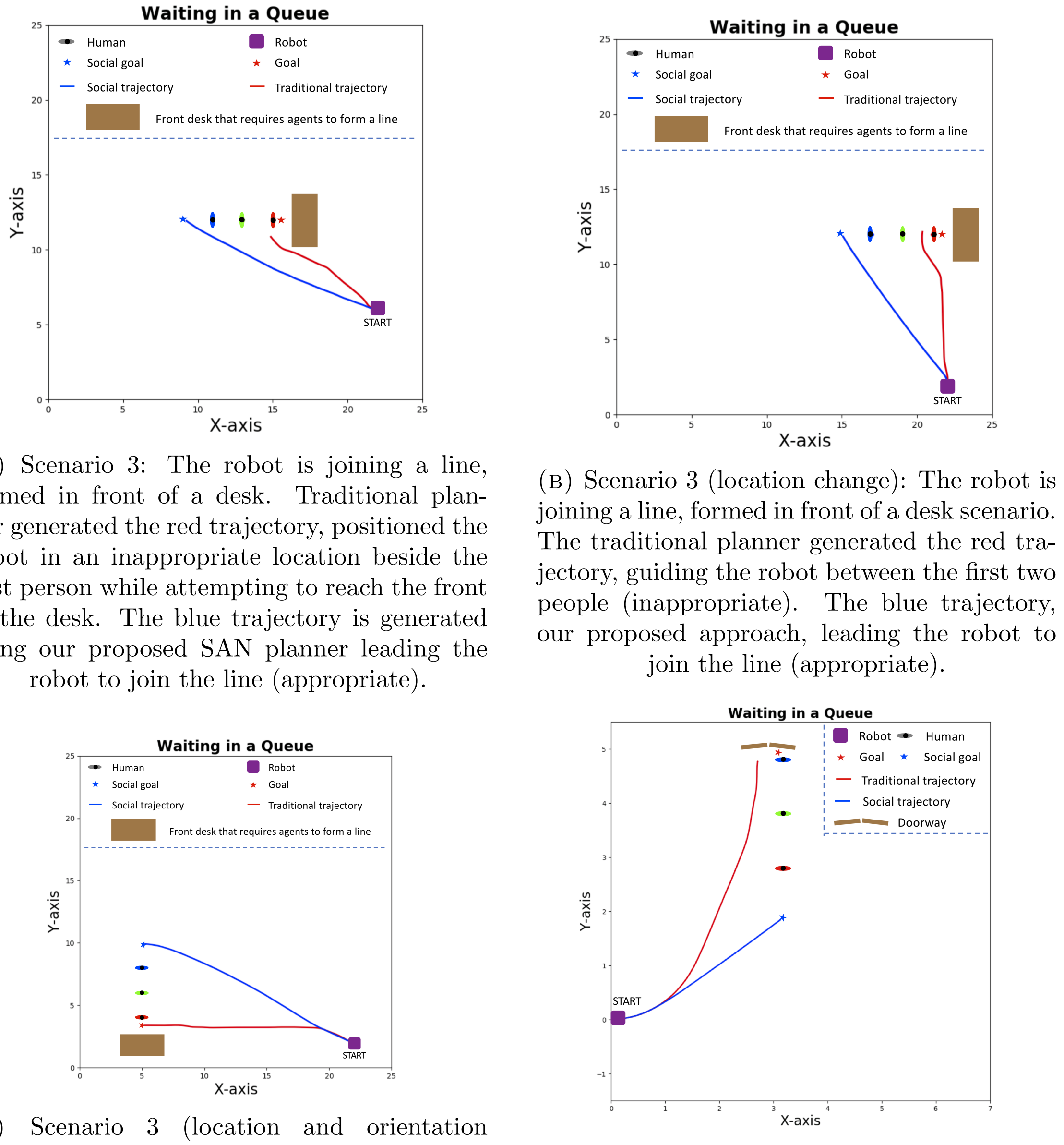
Art gallery interactions are not always presenting the artwork on display. While navigating an art gallery, one should consider the affordance and activity spaces between the artwork and an individual looking at the art. Activity space is a social space linked to actions performed by agents [76]. For example, the space between the subject and a photographer is an activity space, and we humans generally avoid getting in the way of such activity spaces. Affordance space is defined as a social space related to a potential activity provided by the environment [67]. In other words, affordance spaces are potential activity spaces. An environment like an art gallery provides numerous locations as affordance spaces (place in front of every piece of art is an affordance space). When a visitor steps into once such affordance space, that space between the artwork and the interacting human becomes activity space.

In Figure 4.10b, we demonstrated an appropriate behavior around activity space in simulation using our proposed SAN planner. For this scenario, we staged a human-robot interaction consisting of a human interacting with a piece of art working hanging to the wall. Both the traditional planner (red line) and the SAN planner (blue line) were given the same goal (represented as a red star) and start (START) locations.

The traditional planner steered the robot to the goal but did not account for the activity space, i.e., the robot traversed through the activity space (inappropriate). On the other hand, the PaCcET-based SAN planner steered the robot to the goal location while avoiding the activity space (appropriate social behavior). The social goal while avoiding an activity zone is not an end goal where the robot would stop but is more like a social goal that acts as a way-point in reaching the end goal.

Similarly, the two art gallery behaviors (presenting art and avoiding activity space) is implemented and validated on a Pioneer robot, and the results are shown in Figure 4.10c and Figure 4.10d

Figure 4.11a shows the behavior of our social planner and traditional planner in the *waiting in a queue* situation (five objectives) in simulation. Here, we considered a front desk interaction, but this can be generalized to other similar social scenarios where a robot or a human is required to form a line before reaching the goal. For example, social scenarios like getting coffee from a public coffee machine, taking an elevator, etc. In this context, we staged a human-robot interaction consisting of a robot that wants to interact with a front desk representative of an office building where other people were being served on a first-come-first-served basis. Both the traditional planner (red line) and the SAN planner (blue line) were given the same goal (represented as a red star) and start (START) locations. The traditional planner tried to steer the robot to the goal location and stopped at an inappropriate location (besides the person currently being served) as the traditional planner treated the



(A) Scenario 3: The robot is joining a line, formed in front of a desk. Traditional planner generated the red trajectory, positioned the robot in an inappropriate location beside the first person while attempting to reach the front of the desk. The blue trajectory is generated using our proposed SAN planner leading the robot to join the line (appropriate).

(B) Scenario 3 (location change): The robot is joining a line, formed in front of a desk scenario. The traditional planner generated the red trajectory, guiding the robot between the first two people (inappropriate). The blue trajectory, our proposed approach, leading the robot to join the line (appropriate).

(C) Scenario 3 (location and orientation change): The robot is joining a line, formed in front of a desk scenario. The traditional planner generated the red trajectory, guiding the robot to the front of the desk, cutting the line (inappropriate). The blue trajectory, our proposed approach, leading the robot to join the line (appropriate).

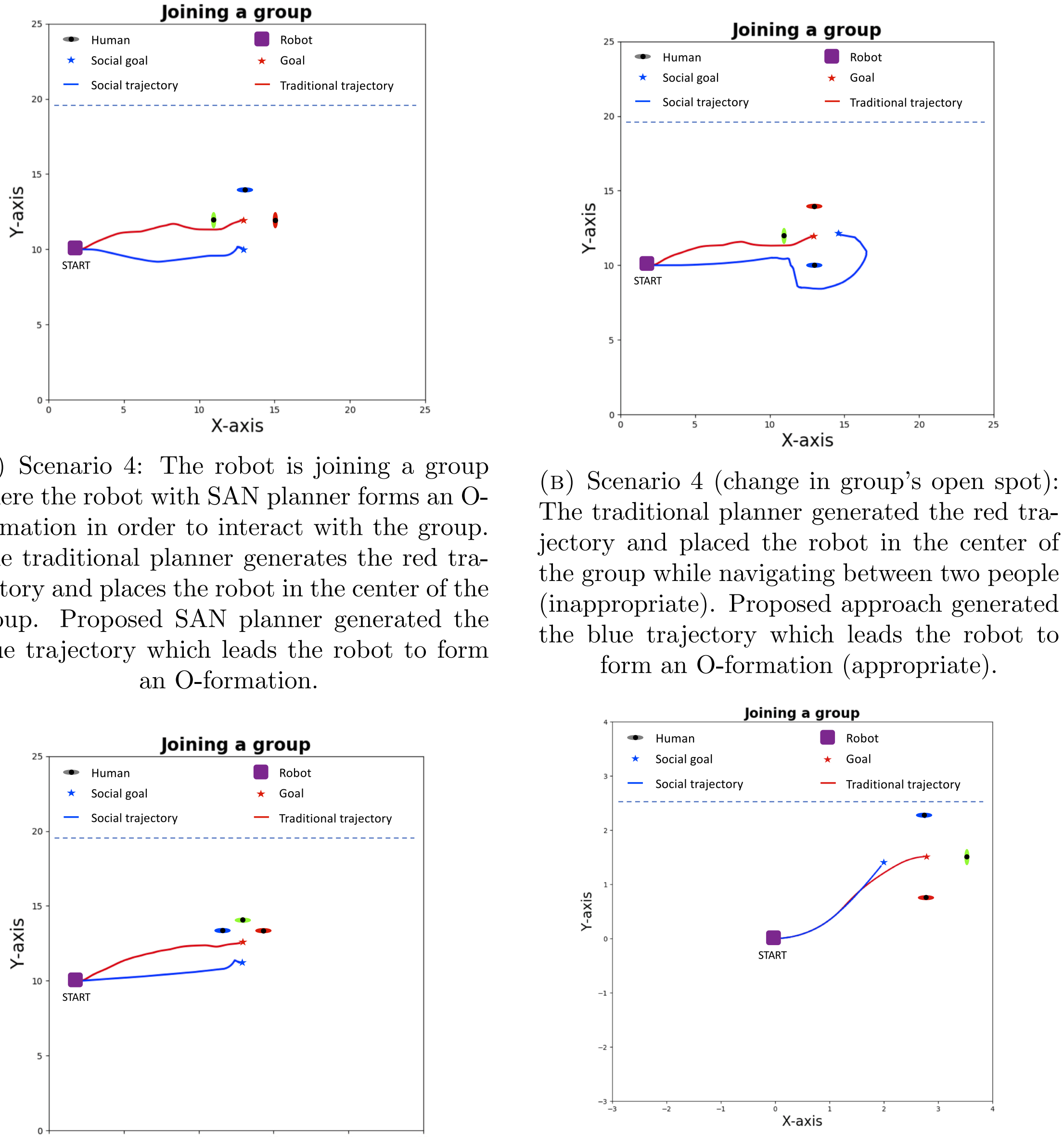
(D) Scenario 3 (real-world): Pioneer robot is joining a line, formed in front of a doorway scenario. The traditional planner generated the red trajectory, guiding the robot to a location besides the first person (inappropriate), cutting the line. The blue trajectory, our proposed approach, leading the robot to join the line (appropriate).

FIGURE 4.11: Validation results of Scenario 3 (waiting in a queue) in both simulation and real-world.

human as an object. On the other hand, the SAN planner steered the robot to an appropriate location, i.e., end of the line positioning the robot behind the last person (social goal), considering personal space as well.

Figure 4.11b and 4.11c shows results with variations in scenarios 3 (*waiting in a queue*). The variations are the locations of people and the orientation of the queue formed by them, figures 4.11b and 4.11c show that our method is robust. Figure 4.11d shows the behavior of our social planner and traditional planner in the *waiting in a queue* situation (five objectives) in the real-world. Here, in a doorway social situation where we humans expect to go one after the other and not rush or cut the line. Both the traditional planner (red line) and the SAN planner (blue line) were given the same goal (represented as a red star) and start (START) locations. The traditional trajectory planner tried to steer the robot to the goal location and stopped at an inappropriate location (besides the first person in front of the door). On the other hand, the SAN planner steered the robot to an appropriate location, i.e., end of the line positioning the robot behind the last person (social goal), considering personal space as well.

Figure 4.12a shows the behavior of our social planner and traditional planner in *Joining a group* situation in simulation. Here, we considered an HRI situation where the robot is required to join a group of three people. However, this can be generalized to interact with more people. Both the traditional planner (red line) and the SAN planner (blue line) were given the same goal (represented as a red star) and start



(A) Scenario 4: The robot is joining a group where the robot with SAN planner forms an O-formation in order to interact with the group. The traditional planner generates the red trajectory and places the robot in the center of the group. Proposed SAN planner generated the blue trajectory which leads the robot to form an O-formation.

(B) Scenario 4 (change in group's open spot): The traditional planner generated the red trajectory and placed the robot in the center of the group while navigating between two people (inappropriate). Proposed approach generated the blue trajectory which leads the robot to form an O-formation (appropriate).

(C) Scenario 4 (robot leading the group's conversation): The traditional planner generated the red trajectory and placed the robot in the center of the group. Proposed approach generated the blue trajectory which leads the robot to form an O-formation (appropriate).

(D) Scenario 4 (real-world): Pioneer robot with SAN planner is joining a group, forms an O-formation in order to interact with them. The traditional planner generates the red trajectory and places the robot in the center of the group. Proposed SAN planner generated the blue trajectory which leads the robot to form an O-formation.

FIGURE 4.12: Validation results of Scenario 4 (joining a group) in both simulation and real-world.

(START) locations. The trajectory planner steered the robot to position it at an awkward location (middle of an interacting group) as the traditional planner did not account for group proxemics and group dynamics. On the other hand, the SAN planner steered the robot to an appropriate location, i.e., a vacant spot on the circle, considering group proxemics.

Figure 4.12b shows a variation in the open spot where the robot needs to join. In this case, the open spot is in a tricky location as the robot has to approach the group from the back. Our proposed approach found a way around the group to the social goal. On the other hand, the traditional planner leads the robot to the center of the group while getting in between two people (blue and green shirts).

Figure 4.12c not only differs in the size of the circle formed but also is a variation of O-formation where the robot is leading the conversion opposing to joining a conversation. When joining a group for discussion, we tend to maintain a uniform spacing between every member of a group wherein joining a group to lead the conversation, all the members of the group except the lead squish together so that the leader can make eye contact with all the members (with an as little field of view as possible). In this case, the traditional planner guides the robot to the center of the group, whereas the proposed method guides the robot to a social goal location where the robot can effectively interact with the group.

Figure 4.12d shows the results of our method implemented on a Pioneer robot executing an appropriate social behavior of *joining a group*. Both the traditional planner (red line) and the SAN planner (blue line) were given the same goal (represented as a red star) and start (START) locations. The trajectory planner steered the robot to position it at an inappropriate location (the middle of an interacting group) as the traditional planner did not account for group proxemics and group dynamics. On the other hand, the SAN planner steered the robot to an appropriate location, i.e., a vacant spot on the circle, considering group proxemics.

So far, the results shown are for multiple complex contexts where the context is not sensed but given. Our PaCcET based local planner was able to demonstrate that taking into account spatial features in multi-objective optimization problem can yield socially appropriate trajectories in various contexts. In the next chapter, we will see autonomous context classification and associated social navigation behaviors for that particular context utilizing the contributions of our PaCcET planner.

4.4 Summary

For our non-learning based SAN planner [5], we chose a non-linear multi-objective optimization tool called PaCcET [6, 69]. For this project, PaCcET was used over other multi-objective tools because of its computational speed [6], as shown in Figure 4.1. Another reason is that by assuming a spatial relationship among agents as linear, we might lose some crucial information. Figure 4.2 shows the overall high-level block diagram of the proposed approach. It is built on top of the well established ROS navigation stack by modifying the local planner to perform PaCcET transformation. The overall function of the local trajectory planner at each time step is to generate an array of possible future trajectory points and evaluate each future trajectory point based on a predefined feature set, as shown in Figure 4.3.

With our PaCcET local planner contributions, we have the following capabilities that contribute towards dissertation:

- Local detection of features such as interpersonal distance, social goals.
- Methods to compute social goals for various contexts.
- A demonstrated local planner that takes into account social objectives to execute socially appropriate navigation behavior, given the context.

In the next chapter, we will see our PaCcET local planner in conjunction with a context classifier to achieve socially normative navigation in multiple autonomously sensed contexts.

Chapter 5

Unified Socially-Aware Navigation

In this chapter, we will discuss the following topics:

- A bird's-eye view algorithmic intuition of the USAN approach implemented in this work.
- Individual subsystems that make up the USAN architecture.
- Results of the perception system of USAN architecture.
- Results of socially-aware navigation behavior appropriate for autonomously sensed contexts.

Previously, in Chapter 3, we demonstrated that a model-based approach to SAN could achieve socially-aware trajectories in a hallway setting. One drawback of model-based approaches is that it requires a lot of training data. In Chapter 4, we were able to achieve socially-aware trajectories in a hallway setting using an optimization method that does not rely on training data. Our experience with both model-based SAN and optimization-based SAN shows that model-based approaches are more suitable for high-level decision making and optimization-based approaches are more suitable at low-level planning stages. This chapter will detail how we implemented a unified socially-aware navigation architecture that used a model-based perception system to make high-level decisions related to a context and an optimization-based local planner that can execute low-level navigation tasks for an autonomously sensed context.

This chapter presents the framework for a novel Unified Socially-Aware Navigation (USAN) architecture and motivates its need in Socially Assistive Robotics (SAR) applications and mobile social robots in general. Our approach emphasizes environmental features, interpersonal distance and how spatial communication can be used to build a unified planner for a human-robot collaborative environment. SAN is vital for helping humans to feel comfortable and safe around robots; HRI studies have shown the importance of SAN transcends safety and comfort. SAN plays a crucial role in the perceived intelligence, sociability, and social capacity of the robot, thereby increasing the acceptance of the robots in public places. Human environments are very dynamic and pose serious social challenges to robots intended for interactions with people. For robots to cope with the changing dynamics of a situation, there is a need to infer intent and detect changes in the interaction context. SAN has gained immense interest in the social robotics community; however, to the best of our knowledge, no planner can adapt to different interaction contexts spontaneously after autonomously sensing the context. Most of the recent efforts involve social path planning for a single context. In this chapter, we propose a novel approach for a unified architecture to SAN that can plan and execute trajectories for an autonomously sensed interaction context that are human-friendly. Our approach augments the navigation stack of the Robot Operating System (ROS) utilizing machine learning and optimization tools. We modified the ROS navigation stack using a machine learning-based context classifier and a PaCcET based local planner for us to achieve the goals of USAN. In this chapter, we discuss the architecture in detail.

Our goal is to have a system architecture for USAN as described in [28], Figure 5.1 shows the block diagram of the proposed architecture. Many SAN approaches use vast amounts of data to mimic human navigation behavior and apply it to a single scenario with multiple people. However, these methods need new training data for every new scenario, and training a USAN planner can be tedious and hardly scales when the environment changes. Our approach to USAN is that a low-level planner using non-linear optimization will handle the proxemics using spatial features, and a data-driven scenario classifier makes high-level decisions on selecting the objectives that matter most for a sensed human-robot navigation scenario. For a robot to navigate socially in human environments and to achieve USAN goals, we state the architecture needs the following sub-systems:

1. *A non-linear multi-objective optimization method for local planning* - Most of

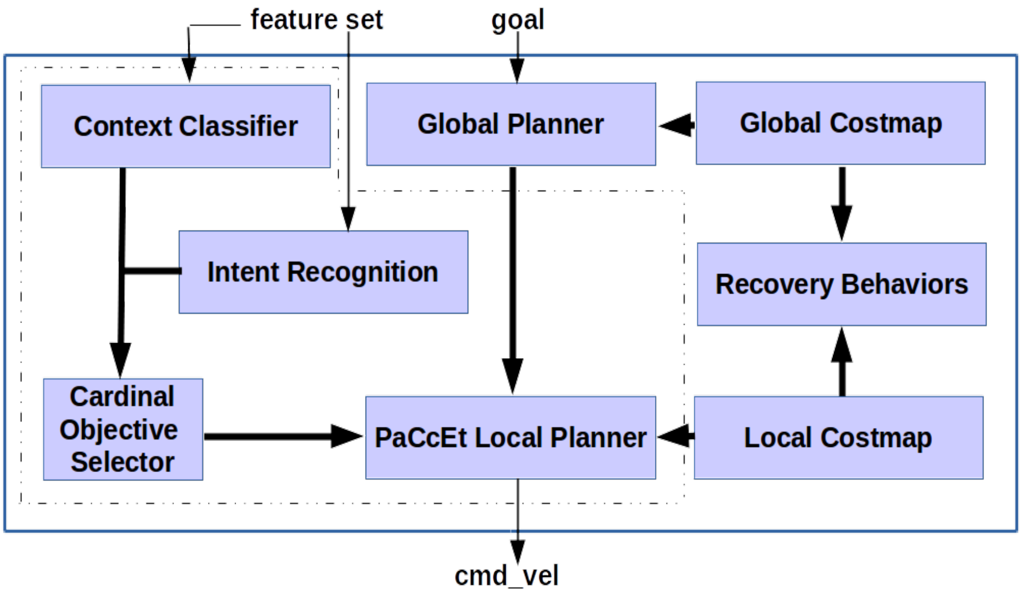


FIGURE 5.1: An overview of the Unified Planner for Socially-Aware Navigation (UP-SAN). Modules within the dotted lines are the modification to ROS navigation stack.

SAN methods assume a linear relationship between objectives related to spatial information. However, It is unclear if any information is lost by assuming the relationship to be linear. To deal with non-linear low-level spatial communication and to scale with the complexity of scenarios, we need a non-linear multi-objective optimization tool such as PaCcET [5, 8].

2. *Intent recognition system* - To detect, track and predict human behavior. Humans predict other people's navigation behavior/intent and adjust our behavior accordingly for efficient and effective navigation. Similarly, an intent recognition sub-system is of utmost importance in USAN when it comes to user experience in HRI (Future work and theoretical contribution).

3. *Scenario/Context classifier* - To classify the type of interaction in real-time and

to select relevant objectives to feed the local planner. As the transformation of objectives and solution generation has to happen in real-time, we can use the classification label of the scenario to select a subset of objectives that are relevant to a particular sensed scenario.

5.1 A Non-linear Multi-objective Optimization Planner

The robot's trajectory can be broken into three parts, the global planner, the local planner, and low-level collision detection and avoidance. The global trajectory planner works by using knowledge of the map to produce an optimal route given the robot's starting position and the goal position. The global path is created as a high-level planning task based upon the robot's existing map of the environment; this is regenerated every few seconds in order to take advantage of shorter paths that might be found or to navigate around unplanned obstacles. The role of the traditional local planner is to stay in line with the global path unless an obstacle makes it deviate from the global path. The low-level collision detector works by stopping the robot if it gets too close to an object[1]. We use the global trajectory planner, and low-level collision detector [10] and make adaptations to the local trajectory planner to incorporate interpersonal distance features using PaCcET (refer to Chapter 4 for implementation details).

[1] More details about the ROS navigation stack can be found at http://wiki.ros.org/navigation/

The modified local trajectory planner gives the USAN architecture the following capabilities:

- Given a context, the planner can account for multiple non-linear social objectives to achieve social navigation.
- The output of the modified planner is command velocities that drive the robot in a socially appropriate manner.

The planner does not have the information about the objectives that matter most for a given situation. This information is given by a context classifier (Section 5.3). After the context classifier determines the high-level decision of navigational context, the cardinal objectives that matter most are selected. Selected objectives are then utilized by our modified local planner to account for social norms to navigate an environment socially.

5.2 Intent Recognition System[2]

The environments in which SAR will be deployed are complex and involve other decision-making agents, such as other robots or humans. In public places, a reactive social planner will only yield a sub-optimal human-robot interaction experience. These complex environments call for an intent recognition module integrated into the

[2] Theoretical contribution

planning pipeline, as shown in Figure 5.1. Intent recognition in SAN is not new; researchers in recent times have explored it [77]. However, intent recognition in a unified planning architecture is not only novel but also a key component in achieving the objectives of USAN.

The SAN problem, to some extent, boils down to a multi-agent optimization problem. When all the agents in a multi-agent system are robots, it is easy to solve as we have similar sensors and standard communication protocols. However, when robots are in a human environment, there is no such standard communication protocol between people and robots. On the other hand, when people interact, we utilize spatial communication to communicate our intent and infer the intent of others. Similarly, there is a need for an intent recognition system that can understand human navigation intentions in a human-robot interaction scenario. Our group is developing an intent recognition module that utilizes OpenPose [78] along with time-series predictive models like Hidden Markov Models (HMM) and Long Short-Term Memory (LSTM). The intent recognition system will answer questions, such as:

- Do particular people belong to a group?
- Does a person belongs to a waiting queue, or he/she is just standing talking to another person?
- Is someone interested in interaction?
- Are the group dynamics changing?

5.3 Context Classifier

A unified socially-aware navigation method must dynamically sense interpersonal and environmental features to identify a context. For this purpose, we previously employed a model-based method to determine the context based on a feature set (environmental features like walls, doorways, and distance-based features like interpersonal distance, distance from a wall). Our prior works [3], used Gaussian Mixture Models (GMM) to implement the context classification functionality trained on a set of distance-based features. Our GMM model was able to distinguish between different scenarios (Passing, Meeting, Walking together towards a goal, and Walking together away from a goal) with an accuracy of 94.74%. While the GMM-based model demonstrated good classification accuracy for different scenarios in a hallway context, it will not scale to other contexts (contexts that rely on camera input) such as an art gallery/museum interactions, joining a group. Unlike GMMs, one advantage of neural nets is that we do not have to handpick the features and hence is an ideal choice for context classification and scene understanding. We investigated neural net-based perception methods that can classify complex scenarios like art gallery interaction, joining a group.

For the Cardinal Objective Selector, the context classifier provides a probability distribution over any potential navigation scenarios. We can use this probability score w_i in selecting the cardinal objectives $(Obj_1, .., Obj_n)$ or emphasize on how much each cardinal objective $(w_1 * Obj_1, w_2 * Obj_2, .., w_n * Obj_n)$ is important based on the sensed context. Using the PaCcET local planner, we can optimize for a subset of objectives that matter most for a sensed context. By selecting cardinal objectives for a context will help filter out irrelevant factors in interaction as well as speed computation to meet real-time constraints.

5.3.1 Convolutional Neural Network

Before continuing with the model architecture and the dataset, we briefly describe general information on CNN, a type of neural network that is used for context classification in this paper.

CNN uses backpropagation [79] to learn the weights of the neural network that can predict different kinds of recommendations. Convolution layers, pooling layers, and fully connected layers as the core components of a CNN network architecture. State-of-the-art classification tasks [80] and end to end systems [81] widely used CNNs at their core. CNN use cases extend to various domains like perception, speech recognition, and autonomous vehicles, to name a few.

Convolution layer is at the core of a CNN, which utilizes convolution operation to extract features in an image. Pooling layers, a down-sampling process, reduces the size of the input array. There are two types of pooling mechanisms; namely, max pooling [82] and mean pooling. In max pooling technique, the maximum value in the region is picked. Whereas, in mean pooling, the average of all the elements in the region is picked. At the end of the CNN are the fully connected layers that have full connections from the previous layer to make a classification decision based on activation. In addition to the mentioned layers, CNN often uses different regularization techniques to eliminate over-fitting in its learning process; one such technique is Dropout [83]. In the next section, the dataset for context classification is detailed.

5.3.2 Context Dataset

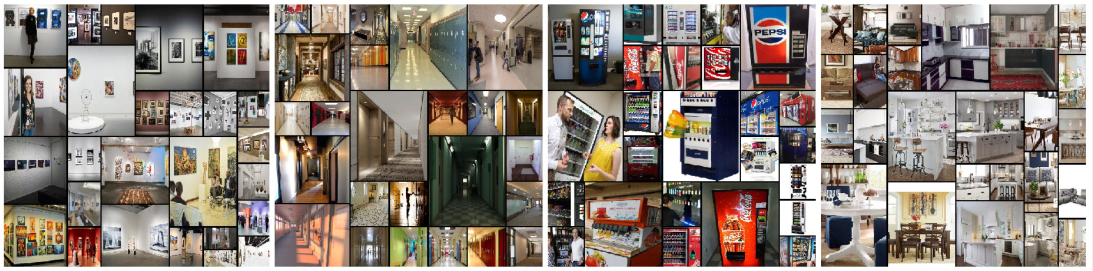


FIGURE 5.2: A sample of images from the internet that constitute images of hallways, artwork, vending machines and other categories used for training our model.

We trained a CNN model to distinguish between four contexts (classes), art gallery, hallway, vending machine, and others (anything which is not a hallway, art gallery, or vending machine - we utilized images of kitchens, living rooms, and dining rooms). We collected a total of 4773 images from the internet, as shown in Figure 5.2 and split them into training (.75), validation data (.25), and further kept aside 400 images for testing on the model as shown in Table 5.1. The images collected were all in color, resized to 256x256, and normalized before feeding to the network. As the dataset is relatively small, data augmentation was incorporated to ensure model generalization. Augmented data includes image manipulations like zoom, shear, a shift in width, a shift in height, horizontal, and vertical flip.

Apart from the data collected from the internet, we collected real-world data at the University of Nevada, Reno, to further test the model. The locations on campus, where we collected data, include buildings in the Colleges of Engineering, Science, and Humanities. The real-world data used for testing, but not part of the training process, includes all the classes - hallway, art gallery, and vending machines.

Class	Train	Validation	Test
Art Gallery	1080	360	100
Hallway	804	268	100
Other	793	265	100
Vending Machine	602	201	100
Total	3279	1094	400

Table 5.1: Amount of data collected for training and testing.

Other social contexts do not depend on the environmental features, but depend on the non-verbal spatial information of people – for example, social contexts like *waiting in a queue* and *O-formations when joining a group*. To account for such non-verbal spatial communication, we collected both simulation and real-world data of people standing in a queue and O-formations using a laser scanner. We collected approximately 170 samples of each context (173 queue context and 168 O-formation). A total of 341 samples, split into 80% training and 20% test data, are collected both from simulations and real-world interactions.

For the real-world samples, the *leg_tracker* package [75] detected the positions of people that were later used to calculate circularity and linearity features to train a

Support Vector Machines (SVM) model to distinguish between standing in a queue and group formations.

5.3.3 Context Model

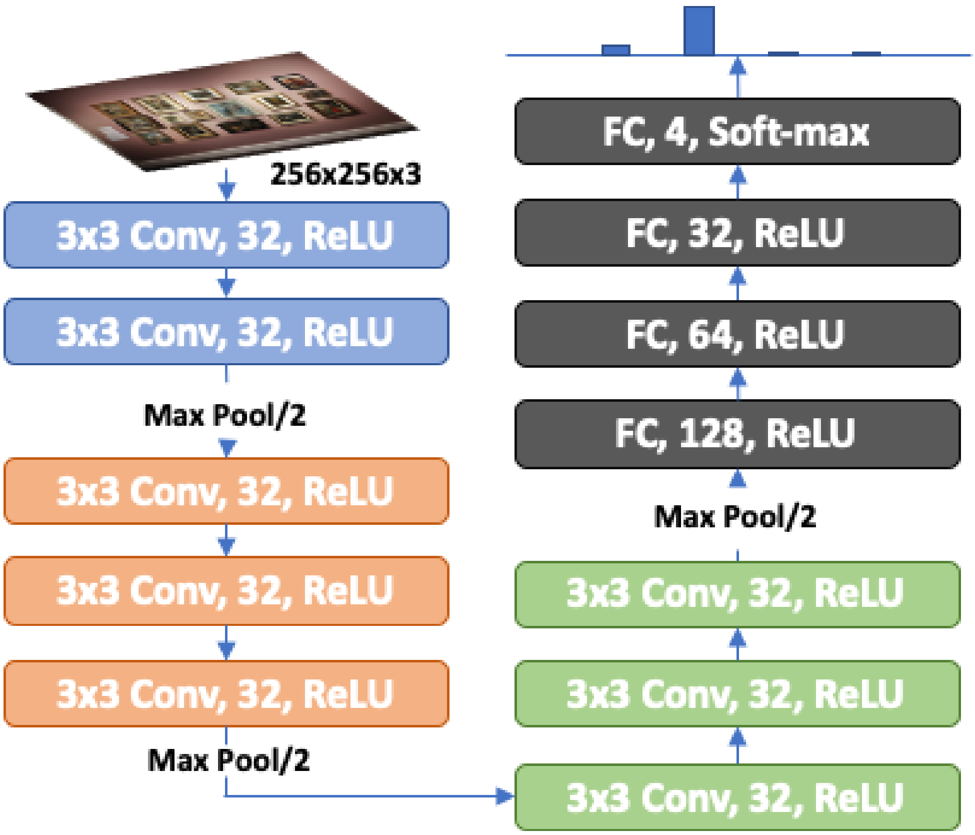


FIGURE 5.3: USAN Context Classifier neural network architecture with 8 convolution layers, 3 max-pooling layers and 4 fully connected layers.

USAN can utilize context information to properly select the objectives specific to the sensed context for a low-level planner [5] to work with. Our approach to a context classifier is a mix of classical machine learning and neural networks. For contexts that include environmental features like hallways, we used images with a CNN architecture that resembles VGGnet [81] but with a shallow depth. For contexts that depend on non-verbal spatial information like waiting in a queue, we used laser scanner data with a linear SVM. CNN takes a 3-channel color image as input and outputs a probability that the image belongs to one of the four classes, as shown in Figure 5.3. The proposed CNN model consists of 8 convolution layers, each with 32 filters, a kernel size of 3, a stride of 1x1, same padding, and ReLU activation. There are three max-pooling layers with a pool size of 2x2 to downsample between layers 2-3, 5-6, 8-9, as shown in Figure 5.3. The network also includes dropout regularization with ever max-pooling layer and between layers 9 and 10 (between first two fully connected layers). All the fully connected layers use ReLU activation except for the last layer, which uses soft-max activation to make the predictions.

When applied to video classification task (continuous frames), the CNN model produced flickering in the predictions of the scene, a common problem in video classification. We used a rolling average method on the prediction probabilities to get a smooth prediction result of the scene.

As discussed earlier, there are some social contexts, such as *group formations* and *waiting in queue*, which are difficult to be studied by 2-dimensional cameras. However, laser data can also be used to understand spatial information, so we used laser scan data to detect and track people in a scene [75]. The positions of the tracked people were used to calculate the following features which were later used in training a linear SVM to distinguish between *waiting in line* and *group formations*:

Circularity: It is used to describe how close a set of points should be to a true circle. The circularity of an irregular polygon formed by a set of points is given by:

$$C = (4 * \pi * area)/perimeter^2 \tag{5.1}$$

Where, area and perimeter of an irregular polygon are:

$$area = 1/2 \sum x_{i+1} * y_i - y_{i+1} * x_i \tag{5.2}$$

$$perimeter = \sum \sqrt{(x_{i+1} * y_i)^2 - (y_{i+1} * x_i)^2} \tag{5.3}$$

Linearity: It is the property by which a set of points can be graphically represented as a straight line. The linearity of a set of points is given by:

$$L = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sum x^2 - \frac{(\sum x)^2}{n}} \tag{5.4}$$

Where, n is the number of points/people.

The range of values for C and L is [0, 1]. People forming a group (circle-like) will have a C value towards 1 and L value towards 0. People forming a line will have a C value towards 0 and L value towards 1. With Circularity and Linearity features, the data is linearly separable, and hence, a linear SVM is one of the simple and ideal models for such data.

The CNN model using camera input and the SVM model using laser data are two distinct models. When the CNN model detection confidence falls below a threshold in *hallway* and *art gallery* contexts, then the robot uses the SVM model to check if the on-going interaction is either *O-formation* or *waiting in a queue* context. In any other context, the planner switches to sub-optimal traditional planning.

Scikit-Learn [84] and Keras [85] with Tensorflow [86] backend was used to implement the proposed context classifier (SVM and CNN). The models were built on a computer with an Intel Core i7-8700K CPU @ 3.70GHz x 12 processors, 32 GB of RAM, and GeForce GTX 1070 Ti GPU with 8GB memory. The CNN model was trained for 500 epochs with a batch size of 64 on the GPU and took approximately two hours. The model was evaluated for accuracy; the training process included Adam optimizer with a categorical cross-entropy loss function. The SVM model for spatial data is built on the same hardware with a linear support vector classification kernel. In the next section, we will see results of our USAN approach.

5.4 Results

In this section, we will see the results of our USAN method implemented on a Pioneer mobile robot. Section 5.4.1 shows the high-level decision to detect the ongoing interaction context. Section 5.4.2 shows a timeline view of various objectives that our system chose based on the detected context. Section 5.4.3 shows appropriate social navigation behaviors in multiple autonomously detected social contexts.

5.4.1 Perception

Our CNN based context classification model was evaluated on validation data, unseen test data and real-world test data. The results are shown in sections 5.4.1.1, 5.4.1.2 and 5.4.1.3 respectively. The results of the SVM model distinguishing *waiting in a queue* and *O-formations* are presented in section 5.4.1.4. To validate our context classifier, we used the following metrics:

- Confusion matrix, defined as a matrix with elements C_{ij} representing the percentage of observations known to be in class i but predicted as class j. For a good classifier, the main diagonal elements should have the highest percentage.

- Precision, intuitively defined as the ability of a classifier not to label a negative sample as positive. It is the ratio $t_p/(t_p + f_p)$.

- Recall, intuitively defined as the ability of a classifier to find all the positive samples. It is the ratio of $t_p/(t_p + f_n)$.

- F-1 score, can be interpreted as a weighted harmonic mean of precision and recall. Where 1 being best and 0 being worst.

Where t_p is the number of true positives, f_p the number of false positives, and f_n the number of false negatives.

5.4.1.1 Validation Set

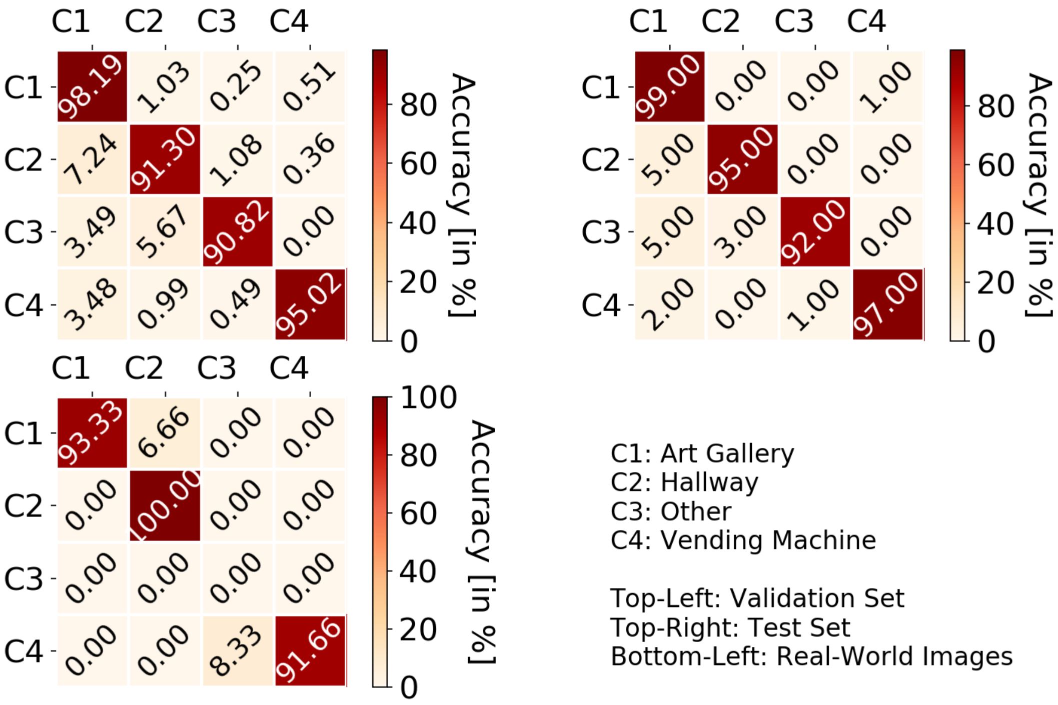
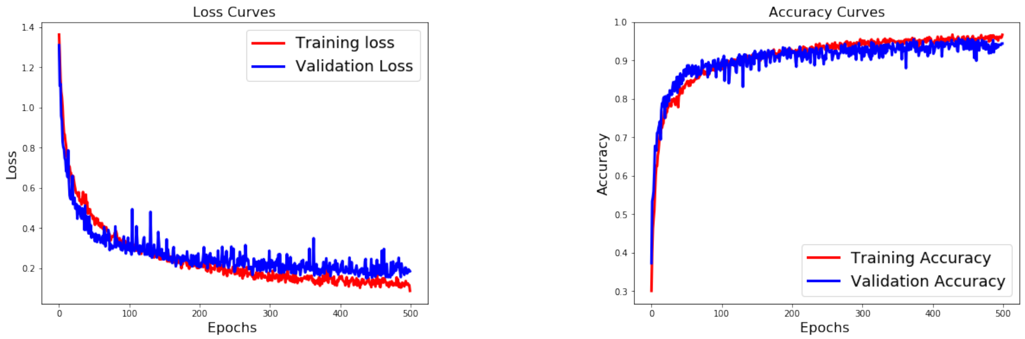


Figure 5.4: Confusion matrix of validation set, test set and real-world images, showing accuracy (in percentage) for all four context.

The model was trained on the training set and validated on the validation set over 500 epochs; the results of loss and accuracy are shown in Figure 5.5. Figure 5.5a shows a plot of training (red) and validation loss (blue) for all the epochs. The loss curves show that the model converges over time. Figure 5.5b shows a plot of training and validation accuracy for all the epochs. Our model achieved a 96.44% training accuracy and 94.33% accuracy on validation data.



(A) Loss Curves: validation loss (blue) and training loss (red) vs Epochs

(B) Accuracy curves: validation accuracy (blue) and training accuracy (red) vs Epochs

FIGURE 5.5: Training and validation curves: categorical cross entropy loss and accuracy curves

The confusion matrix of the validation set, shown in Figure 5.4, shows that the model was able to learn to distinguish between an art gallery, a hallway, a vending machine, and other contexts with an accuracy of 98.19%, 91.30%, 95.02%, and 90.82% respectively. Table 5.2 shows performance on the validation set.

Validation set / Test set / Real-world data			
Class	Precision	Recall	F1-Score
C1	0.92 / 0.89 / 1.0	0.98 / 0.99 / 0.93	0.95 / 0.94 / 0.97
C2	0.93 / 0.97 / 0.97	0.91 / 0.95 / 1.0	0.92 / 0.96 / 0.99
C3	0.98 / 0.99 / 0.00	0.91 / 0.92 / 0.00	0.94 / 0.95 / 0.00
C4	0.98 / 0.99 / 1.0	0.95 / 0.97 / 0.92	0.97 / 0.98 / 0.96
C1: Art Gallery, C2: Hallway, C3: Other, C4: Vending Machine			

TABLE 5.2: Performance of the CNN based context classifier.

5.4.1.2 Unseen Test Set

The confusion matrix of the unseen test set (Images from the internet that we kept aside) is shown in Figure 5.4 shows that the model was able to generalize to unseen data and was able to distinguish between an art gallery, a hallway, vending machine, and other contexts with an accuracy of 99.00%, 95.00%, 97.00%, and 92.00% respectively. Table 5.2 shows performance on the unseen test set.

5.4.1.3 Real-World Data

To see if the model generalizes to real-world images that it has not seen, we collected 15 art gallery, 33 hallway, and 12 vending machines, a total of 60 images on campus. The "other" category is only a place-holder for any other context apart from the learned hallway, art gallery, and vending machine, so we omitted it from this test set. When in an unknown context, the planner can select default, but likely sub-optimal, objectives that will reward safe movement from one place to another. As seen in Figure 5.4, the model performed well on real-world images as well. The accuracy of an art gallery, hallway, and vending machine categories are 93.33%, 100.0%, and 91.66%, respectively. The performance on real-world data is presented in Table 5.2.

5.4.1.4 Group and Queue Formations

We trained a linear SVM on the people's location data collected from a laser scanner to classify if a group of people as *waiting in a queue* or forming a *O-formation*. We selected features like circularity, linearity, and the radius of the best-fit circle (with standardization). Later, we trained the SVM omitting the radius feature as circularity and linearity are sufficient to differentiate between the two classes, as

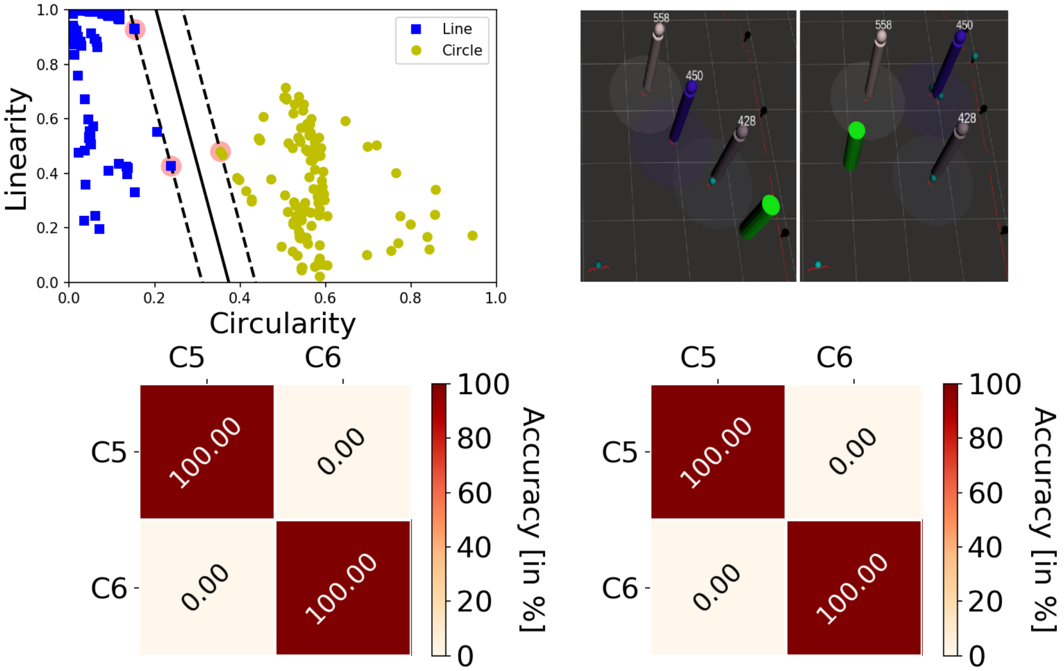
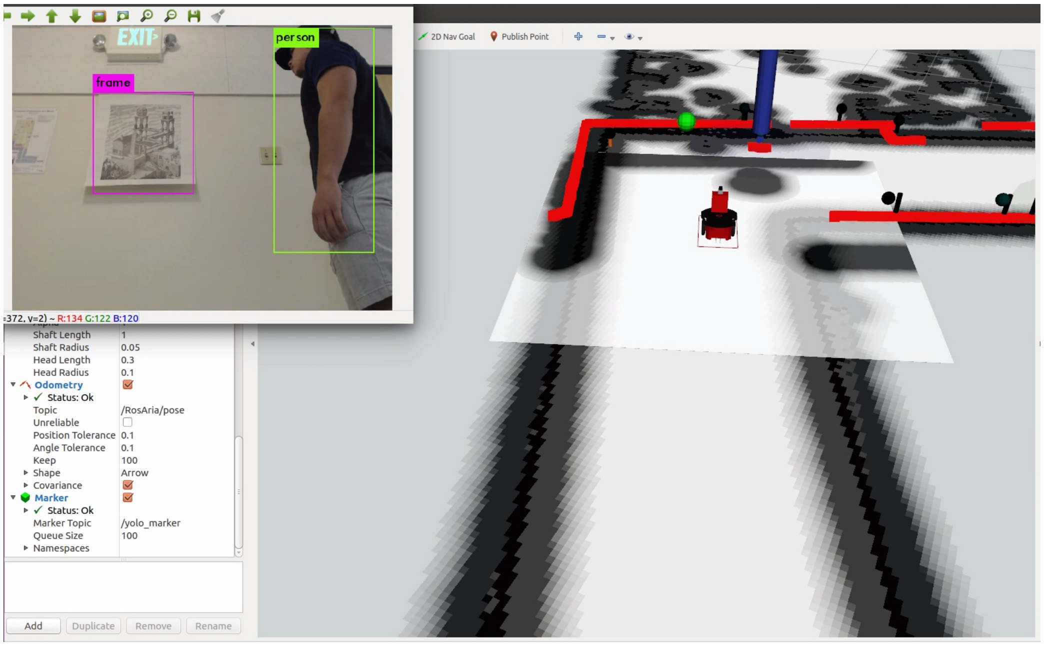


FIGURE 5.6: **Top-left:** trained SVM classifier, **top-right:** Social goal determined by the robot in *waiting in queue* and *O-formation* contexts, **bottom-left and bottom-right** confusion matrices of training and test set respectively.

shown in Figure 5.6 (top-left). The trained SVM achieved 100% accuracy on both training and test data, confusion matrices of the training set with three-fold cross-validation, and the test set is shown in bottom-left and bottom-right of Figure 5.6 respectively. Precision, recall, and f1-scores are all 1.00 for both training and test sets. Figure 5.6 (top-right) shows an rviz screenshot of the mathematically computed social goal (green cylindrical marker) determined by the robot in *waiting in queue* and *O-formation* scenarios (as discussed in Chapter 4).



=372, y=2) ~ R:134 G:122 B:120

Shaft Length 1
Shaft Radius 0.05
Head Length 0.3
Head Radius 0.1
Odometry ☑
Status: Ok
Topic /RosAria/pose
Unreliable □
Position Tolerance 0.1
Angle Tolerance 0.1
Keep 100
Shape Arrow
Covariance ☑
Marker ☑
Status: Ok
Marker Topic /yolo_marker
Queue Size 100
Namespaces

FIGURE 5.7: Image showing object detection and tracking using YOLO-v3 and *leg_tracker* package. The image window (top-left) shows artwork and human detection using YOLO-v3. RVIZ screenshot shows human detection (dark blue cylindrical marker) using *leg_tracker* package and localization of artwork in laser data (green spherical marker).

5.4.1.5 Object Detection and Tracking

For detection and tracking of people using a laser scanner, we used a people tracker package [75] by Leigh *et al.* To visually detect and track picture frames (for art gallery interactions), we trained YOLO-v3 [87] on Open Images Dataset. To track picture frames in 3D, we used (x, y) pixel locations in the camera to calculate the depth in the laser scanner data.

5.4.2 Cardinal Objective Selection

We teleoperated the robot in an environment with *hallways*, *artwork*, *people in O-formations*, and *people waiting in queues* to test if the models can select objectives related to detected context. The results of the robot deciding on the objectives for an autonomously sensed context are shown in Figure 5.8, the transitions from one context to the other are shown using the vertical grid lines. Figure 5.8 shows that the robot is considering personal space and activity space in an *art gallery* situation. In a *hallway* situation, the robot accounts for personal space and staying on the right-side objectives. Similarly, in a *group (O-formation)* scenario, the robot considers the personal space of all the people, the O-space of the group, and the social goal of joining the group. In *waiting in a queue* context, the robot considers joining the end of the line along with the personal space of the people forming the line. It is also important to note that reaching the goal, and collision avoidance are other objectives of our PaCcET local planner.

The black box with the dotted line in Figure 5.8 shows the ambiguity of classification during the transition of the same group of people from *O-formation* to a *line formation*. This ambiguity is due to the quick change in the group dynamics, but mis-classification for a fraction of a second should not affect the overall social performance of the planner.

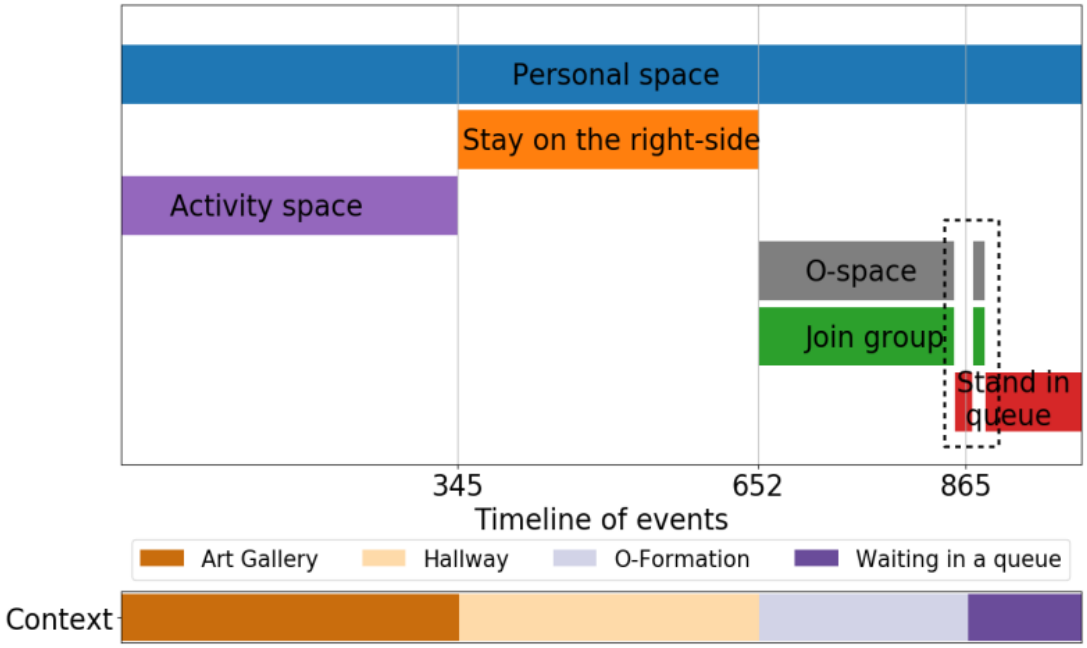


Figure 5.8: Timeline showing the social objectives selected by the robot when teleoperated in an environment with hallways, artwork, people in O-formations, and people waiting in queue contexts.

5.4.3 Multi-Context Socially-Aware Navigation

In Sections 5.4.1 and 5.4.2, we discussed the results of perception pipeline: performance of the CNN based visual classification, SVM based group scenario classification using laser data, by teleoperating the robot in an environment, we showed that our method was able to detect the context accurately and thereby was able to select the cardinal objectives for that particular context.

Figure 5.9 shows the robot's interaction in an *art gallery* followed by a *hallway* context. In the *art gallery* context, the robot encountered one spectator viewing the art. When switching to hallway context, the robot encountered a person in the narrow hallway. The green trajectory in figure 5.9 **a**, **c** represents the shortest global

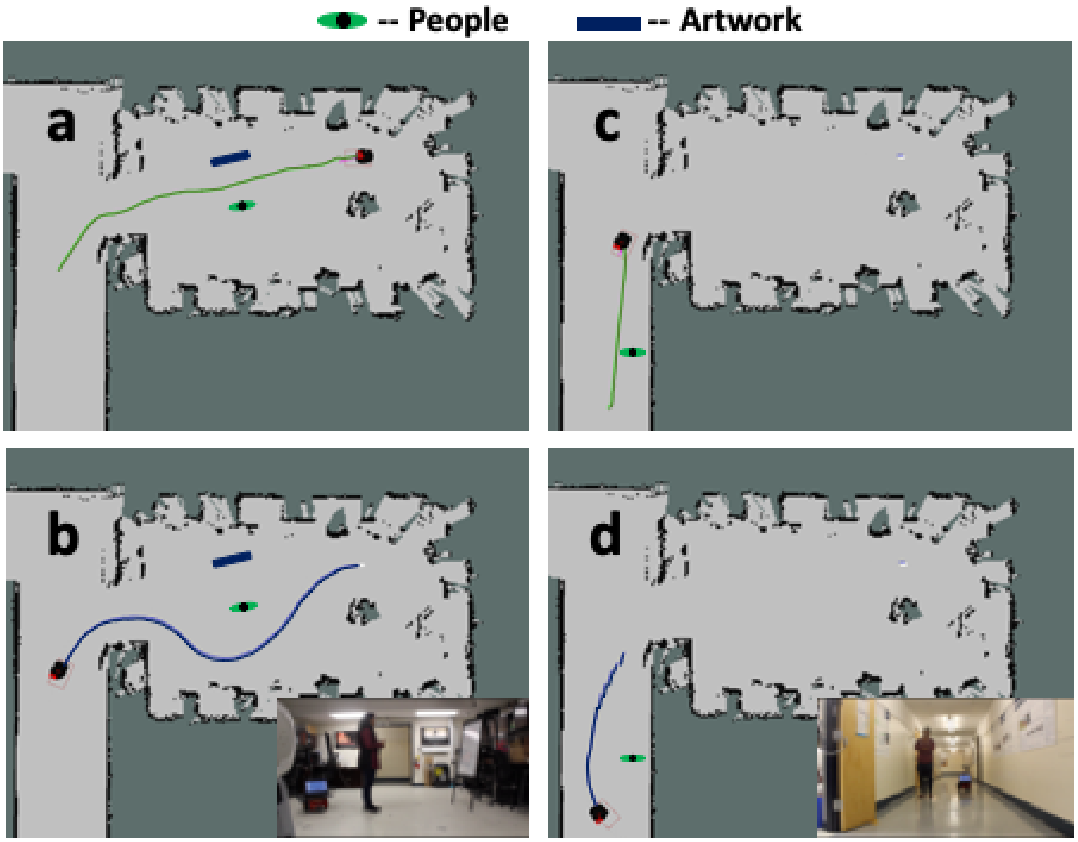


Figure 5.9: Sub figures **a**, **c** shows a non-social path a robot with traditional planner would take in an *art gallery* and *hallway* contexts respectively. Sub figures **b**, **d** shows the social path our SAN planner executed.

trajectory that a traditional local planner would closely follow. In figure 5.9 **a**, the trajectory violates the social rule of traversing in the activity zone (space between the artwork and the spectator). In figure 5.9 **c**, the trajectory violates the personal space around the human in a tight hallway. On the other hand, in figure 5.9 **b**, our social planner steered the robot away from the activity space, thereby executing a socially appropriate trajectory in an art gallery. Similarly, in figure 5.9 **d**, our social planner steered the robot in such a way that it does not violate a person's personal space.

Figure 5.10 shows the robot's interaction in an *O-formation* situation followed by

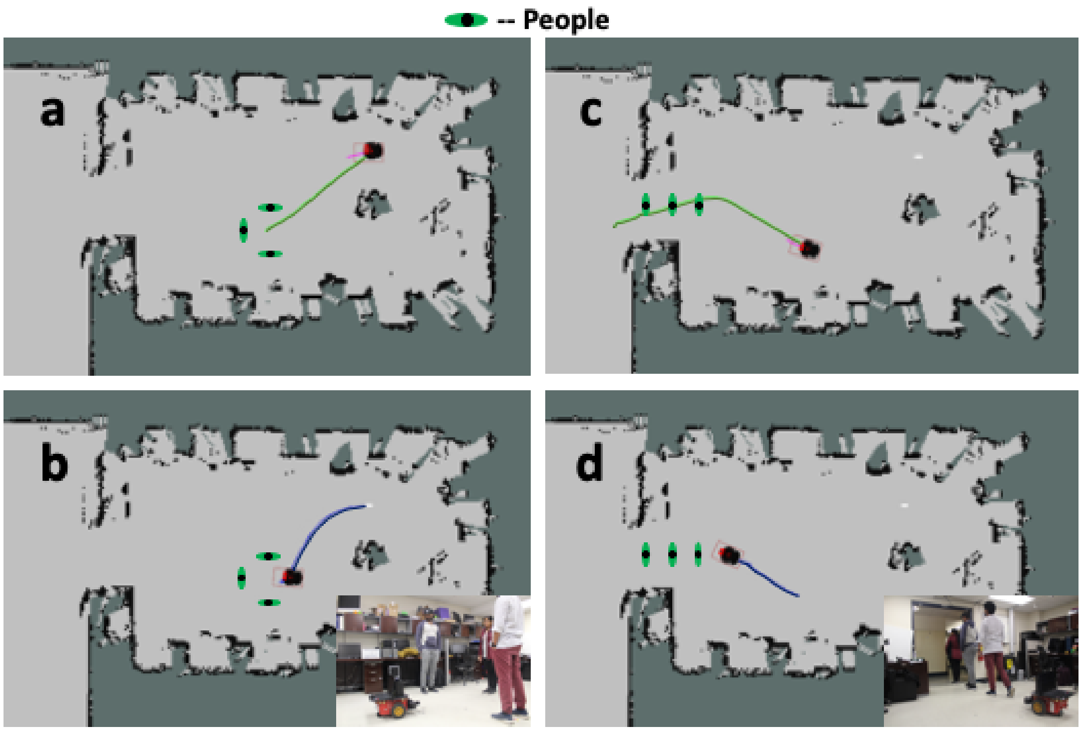


FIGURE 5.10: Sub figures **a**, **c** shows a non-social path a robot with traditional planner would take in an *O-formation* and *waiting in a queue* contexts respectively. Sub figures **b**, **d** shows the social path our SAN planner executed.

a *waiting in a queue* context. In both these contexts, the robot interacted with three humans. The green trajectory in figure 5.10 **a**, **c** represents the shortest global trajectory that a traditional local planner would closely follow. In figure 5.10 **a**, the trajectory steered the robot to the center of the group, placing it in an inappropriate location to meet with the group. In figure 5.10 **c**, the generated trajectory forces the robot cut the line which is socially inappropriate. On the other hand, in figure 5.10 **b**, our social planner steered the robot to an appropriate location on the circle formed by the group (social goal). Similarly, in figure 5.10 **d**, our social planner steered the robot to the end of the line formed by the people (social goal). The social goal calculation in *O-formation* and *waiting in a queue* context is determined by mathematical modeling [8], as discussed in Section 4.2.

5.5 Summary

Our goal is to have a system architecture for Unified Socially-Aware Navigation (USAN) as described in [28], Figure 5.1 shows the block diagram of the proposed architecture. Our approach to USAN is that a low-level planner using non-linear optimization will handle the proxemics using spatial features, and a data-driven scenario classifier makes high-level decisions on selecting the objectives that matter most for a sensed human-robot navigation scenario. For a robot to navigate socially in human environments and to achieve USAN goals, we state the architecture needs following subsystems:

1. *A non-linear multi-objective optimization method for local planning* - Most of SAN methods assume a linear relationship between objectives related to spatial information. However, It is unclear if any information is lost by assuming the relationship to be linear. To deal with non-linear low-level spatial communication and to scale with the complexity of scenarios, we need a non-linear multi-objective optimization tool such as PaCcET [5, 8].

2. *Intent recognition system* - To detect, track and predict human behavior. Humans predict other people's navigation behavior/intent and adjust our behavior accordingly for efficient and effective navigation. Similarly, an intent recognition

subsystem is of utmost importance in USAN when it comes to user experience in HRI (Future work or secondary contribution).

3. *Scenario/Context classifier* - To classify the type of interaction in real-time to select relevant objectives to feed the local planner. As the transformation of objectives and solution generation has to happen in real-time, we can use the classification label of the scenario to select a subset of objectives that are relevant to a particular sensed scenario.

With this chapter, we demonstrated the following capabilities of our proposed USAN architecture:

- We established a need for a high-level context classifier [3, 4]. We realized a context classification pipeline using CNN for image-based input and an SVM for laser-based input. We validated the context classifier in real-world scenarios.
- We developed a non-learning based local planner that requires no training data and uses a non-linear multi-objective optimization tool to achieve SAN behaviors in multiple contexts [8].
- Finally, we evaluated our USAN approach by demonstrating socially-aware navigation on multiple autonomously sensed contexts [29].

In Chapter 2, we reviewed state-of-the-art SAN methods and the evaluation methods used in evaluating such approaches. We established and identified the need for measuring the perceived social intelligence of robots. In the next chapter, we will discuss perceived social intelligence and its importance in evaluating SAN systems.

Chapter 6

Perceived Social Intelligence

- We will discuss Perceived Social Intelligence (PSI) and its importance as a measurement scale in HRI.
- Results of bystander's perception of social intelligence of robot with SAN planner.

The perception of the social intelligence (PSI) scale allows us to measure one's perception of robots' social intelligence with SAN more precisely than the measure of general intelligence. This is important because most robots are still incapable of processing social interactions, but a robot with SAN may be giving us the impression that they do. Navigation systems are becoming more natural (human-like) and less robot-like, which gives us the impression that a robot may be "thinking about" or "acknowledging" human presence when these robots have no cognition similar to what humans have. Because of this, the measures for evaluating a human's social intelligence are too advanced for current robot abilities [24]. For example, when a robot orients itself much like a human would, it gives us the impression that there is some higher-level cognition going on within the robot due to its actions. By evaluating different HRI scenarios with the PSI measure of social intelligence, we are expanding upon the idea that humans are influenced in basic areas of comfort, naturalness, and sociability when exposed to human-robot interactions [13]. The PSI also allows us to evaluate the perception of a robot by surveying a wide range of social intelligence capabilities [24].

Until now, not much has been done to research our perception of robots' social intelligence when interacting with SAN compared to traditional navigation. As socially-aware navigation techniques are continuously improving, we must understand how people react and perceive these robots that are integrating into more advanced roles. Social intelligence is defined as the ability to successfully interact or communicate with others to accomplish goals [88]. For example, in a navigational context, social intelligence is when a robot can inspire a human to assist it by removing obstacles in its path. Social intelligence is important for social robots that interact and communicate with people [89]. Social intelligence is critically important for any robot that will be around people, whether engaged in social or non-social tasks. Robots with high PSI can be perceived as less annoying and not rude, thereby improving the acceptance of robots for long-term deployment in public places. Some aspects of robotic social intelligence have been included in HRI research [22, 23, 90, 91], but current measures are brief and often include extraneous variables. We designed a comprehensive measure of the perceived social intelligence of robots [24–26].

The motivation to design PSI scales is two-fold. When evaluating a robot, several metrics already exist for examining how the robot behavior affects its perception by humans. Widely used HRI survey instruments either examine positive [22] or negative [23] feelings about a robot exhibiting social behavior. However, these scales lack in measuring the social intelligence of a robot with socially-aware behavior as perceived by a human. On the other hand, many scales measure the social intelligence of humans [92, 93]. In theory, these scales can be adapted to measure the social intelligence of robots provided the robots have certain necessary skills that are essential for smooth social interactions. All most all humans, including children, have these skills. For example, humans understand that other people have emotions, thoughts, and behavior. Humans can distinguish humans, and non-humans; can remember their previous conversations with individuals. Most robots do not have these basic skills, which makes it hard to adapt existing scales that measure human social intelligence to measure robots' social intelligence.

Apart from the architecture, there is a need for HRI metrics to evaluate navigation behavior from a human's standpoint. We propose a standardized PSI study design - To evaluate the system and the quality of interaction from a human's standpoint, special emphasis on Perceived Social Intelligence (PSI) [24, 25]. Performance metrics such as time, distance traveled provides insights into the efficiency of the planner.

However, when it comes to HRI, the humans' perception of the robot's behavior is of utmost importance [9]. We developed an experiment where a bystander observes overhead views of robots and humans interacting and rate the resultant robot behavior [27]. Here, we outline the scenarios we chose where the participant was able to be a bystander to the human-robot interaction, which allowed participants to perceive the robot's social intelligence based on the interaction provided (Socially aware vs. traditional navigation). These scenarios allow the participant to decide for themselves whether or not the robot is behaving in a socially intelligent way.

To measure social intelligence, we use the PSI short form [25]. This short-form consists of 20 statements having to do with measuring the robot's social intelligence. A high rating on the PSI indicates a strong level of social intelligence, while low rating indicates little to no social intelligence. Consequently, it would make sense if observers relate social-awareness in the navigation to social intelligence; the following hypothesis will be supported:

Hypothesis: Participants who observe a socially-aware navigation planner will perceive the robot as more socially intelligent than one that is utilizing a traditional navigation planner.

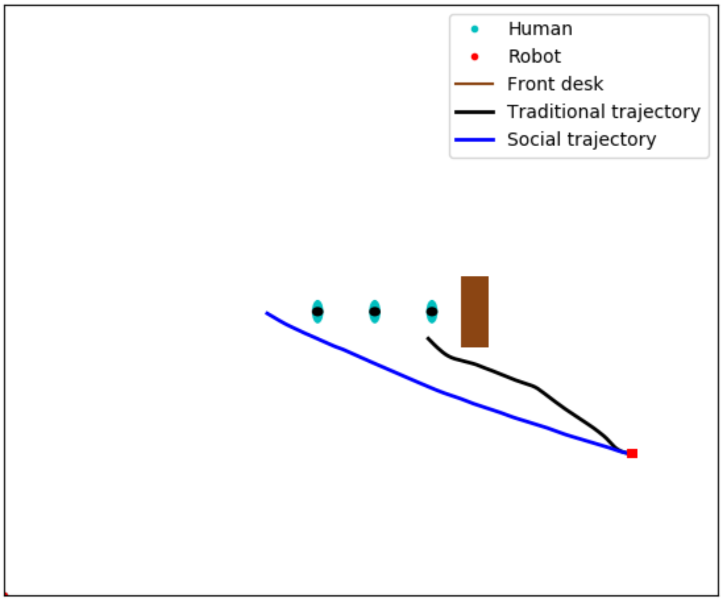


Figure 6.1: This diagram shows the robots' trajectory for socially aware and traditional models of navigation in the "waiting in a queue" scenario, showing a line forming in front of a desk.

6.1 Experiment Design

To test the above hypothesis, we asked participants to view videos of simulated robot movements near humans and environmental features relevant to the navigation task. These videos were animated renderings of the robot and person's positions on a white background (see Figure 6.1, 6.2, 6.3) about from trajectories from our prior work [8]. The figures are rendered as outlines of people or robots, thus simplifying the scene for the viewer. This overhead view also removed considerations of interaction features, like facial expressions or gestures, thus controlling only for movement effects. We

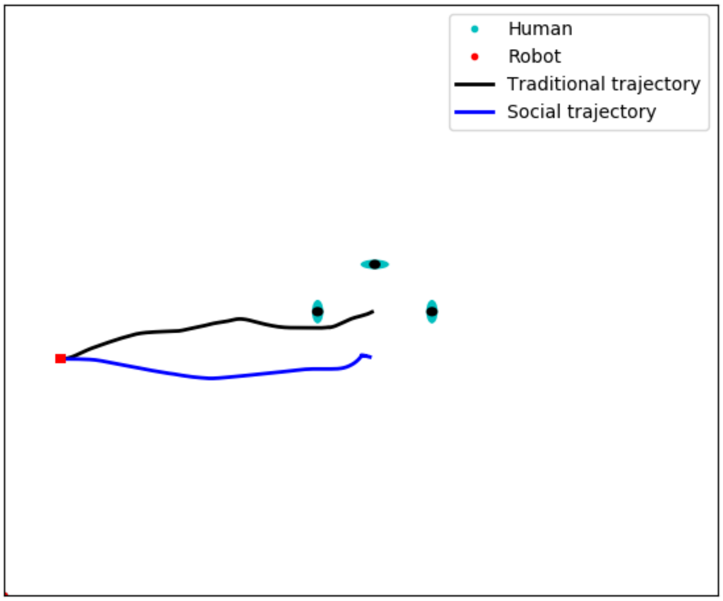


Figure 6.2: This diagram shows a robot with traditional navigation joining a group as indicated by the black line, and shows the robot with socially aware navigation joining the group as represented by the blue line.

asked participants to rate the robot's movement for each video for the perception of social intelligence (PSI). The study was conducted using the Qualtrics platform [94].

There were three simulation scenarios for each navigation category: Socially-Aware navigation and Traditional navigation. Participants were randomly assigned to one of these two categories through the online Qualtrics survey platform and given the PSI short form [24, 26] immediately after watching each video. At the beginning of the survey, we asked participants for their age, gender, and career/field of study.

We recruited a total of 70 participants, 25 Female, 43 Male, 1 Agender Flem Flux, and 1 Non-Binary. The age range was from 18-65 with a mean of 28. One participant was

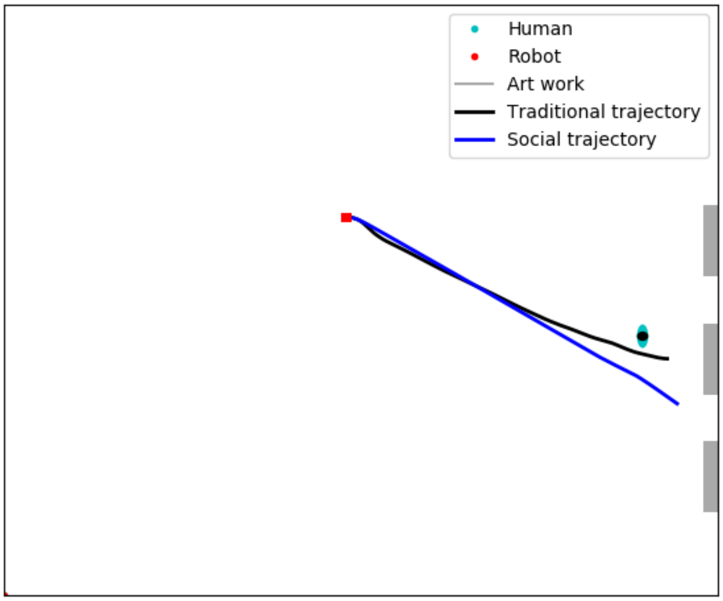


FIGURE 6.3: This diagram shows the robot with traditional and socially aware navigation joining a human observing art as indicated by the black and blue lines.

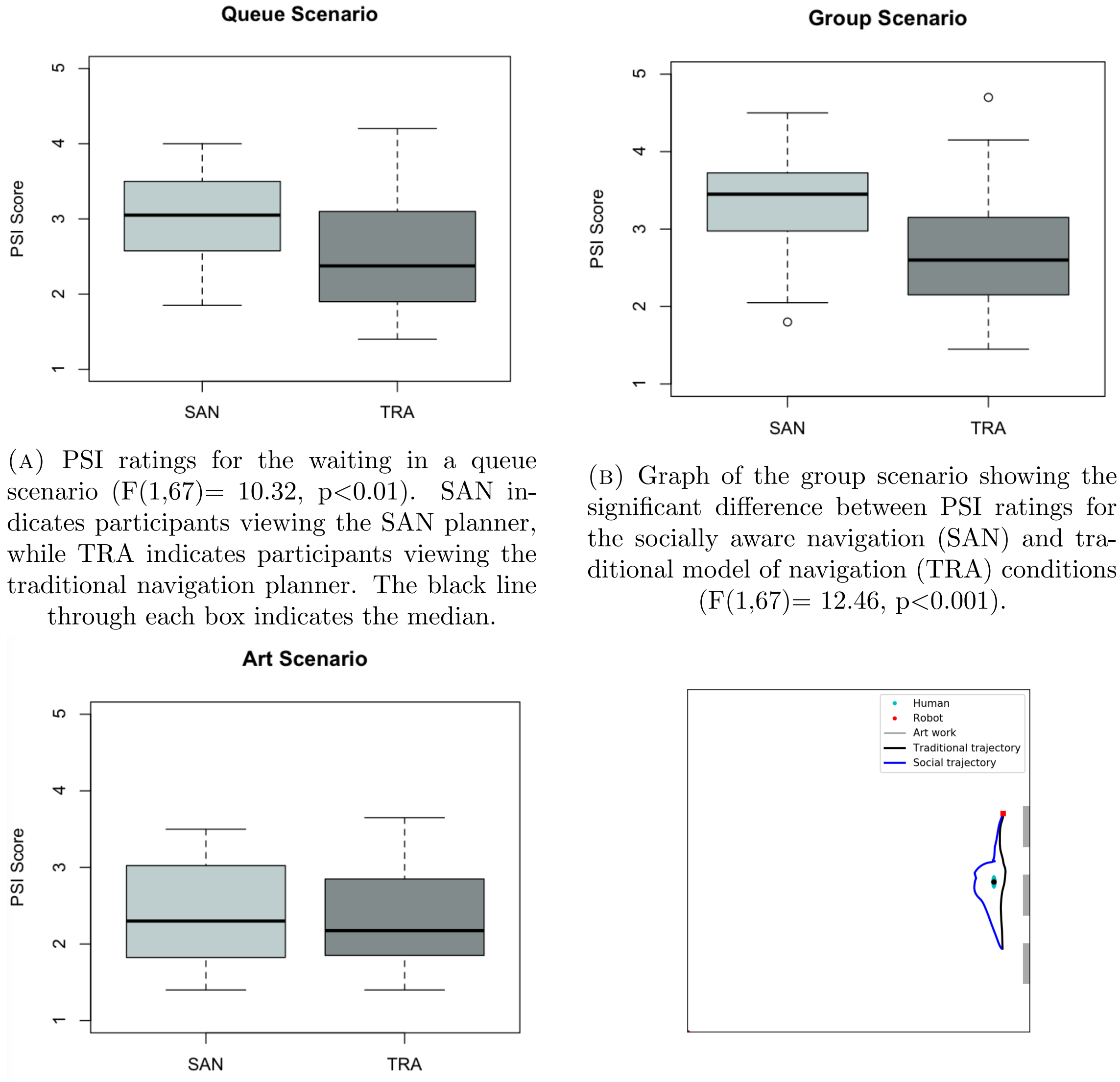
omitted from the data set due to the failure of answering all questions in the Group scenario PSI rating. Each participant was randomly assigned to one of the two interaction conditions, socially-aware navigation, or traditional navigation. A Shapiro-Wilk test was used in R [95] to determine if the data were normally distributed. Two out of the three conditions were normally distributed; therefore, an ANOVA was run for the normally distributed conditions (Queue and Group). A Kruskal-Wallice test was used for the non-normally distributed data for the Art scenario.

6.2 Results

Understanding the perceived social intelligence (PSI) of a robot is crucial when these robots are designed to work alongside humans. PSI allows researchers to improve the general social behavior of the robots; we designed PSI scales to measure the social intelligence of robots utilizing SAN [26]. The motivation for PSI scales is discussed in Chapter 2, Section 2.1, and the experiment design is discussed in detail in Section 6.1. In this section, we will discuss the results obtained from our PSI survey, where the participants (bystanders) rated the robot with SAN and robot with traditional navigation on perceived social intelligence.

After conducting an ANOVA on the *waiting in a queue* scenario, there was a statistical significance between PSI ratings of robots with socially-aware navigation compared to traditional navigation (F(1,67)= 10.32, p<0.01). Figure 6.4a shows the significant difference between the socially-aware (SAN) and traditional (TRA) navigation groups for PSI ratings. The robot with SAN was rated significantly higher on the PSI compared to the robot with traditional navigation.

In addition, there was a statistical significance after conducting an ANOVA on *joining a group* scenario. Robots with socially-aware navigation were rated as significantly more intelligent on the PSI than robots who demonstrated the traditional navigation (F(1,67)= 12.46, p<0.001). Figure 6.4b shows the significant difference in PSI ratings for the SAN and TRA groups.



(A) PSI ratings for the waiting in a queue scenario (F(1,67)= 10.32, p<0.01). SAN indicates participants viewing the SAN planner, while TRA indicates participants viewing the traditional navigation planner. The black line through each box indicates the median.

(B) Graph of the group scenario showing the significant difference between PSI ratings for the socially aware navigation (SAN) and traditional model of navigation (TRA) conditions (F(1,67)= 12.46, p<0.001).

(C) Graph of the art scenario showing no significance between the socially aware navigation group (SAN) and the traditional model of navigation group (TRA) (Chi-squared= 0.35, p>0.05).

(D) This diagram shows how a robot with the traditional navigation trajectory approaches the art in front of the human, but the social trajectory approaches the art by navigating behind the human that is already observing.

Figure 6.4: PSI results for *A. Queue*, *B. Group*, and *C. Art* scenarios. Alternative *Art* interaction shown in D.

There was no statistical significance after conducting a Kruskal-Wallis on the *art gallery* scenario. The PSI of a robot with SAN was rated similarly to the ratings of a robot with the traditional model of navigation (Chi-squared= 0.35, p>0.05). Figure

6.4c shows the similarities between PSI ratings for the SAN and TRA navigational groups.

The findings largely support the hypothesis that participants will rate a simulated agent higher if it is exhibiting socially-aware navigation behavior than behavior typical of a robot with a traditional planner. The distinct differences between social and traditional trajectories for these scenarios closely relate to the acceptable and unacceptable behaviors of human-human interaction. These results support the use of the PSI to note bystander judgments of the effectiveness of a SAN planner.

These effects were observed in both the *queue* and *group* scenarios. These are scenarios where the robot utilizing a traditional planner commits gross social violations (cutting in line, breaking into the center of a group of people that are in social proximity). The participants clearly viewed these actions as indicative of lower social intelligence.

In the art gallery scenario, there was no significance between the perceived social intelligence of robots with socially aware navigation compared to the traditional model of navigation. We attribute this to the minor difference between the socially-aware planner and the traditional planner in the simulations participants viewed. As we can see from figure 6.3, the robot with the traditional model of navigation (indicated in black) approaches the human getting quite close and orients itself in front of the human blocking or obstructing their view). However, in figure 6.3, the trajectory for the robot with socially-aware navigation gives the human more space than the traditional model when passing by, but proceeds to get closer to the artwork than the present human. This is a potential dilemma because if a robot (regardless of the model of navigation) proceeds to be closer to an object than the human can be perceived as not socially intelligent.

We have come up with an alternative art scenario (see Figure 6.4d) where the robot is approaching from one side of the person viewing the art/poster, thereby avoiding invasion of activity space. The socially-aware navigation trajectory is approaching the human looking at the art and navigates behind them, but the traditional navigation trajectory navigates in front of the human observing the art. We believe that a new approach scenario like this will more distinctly highlight a social violation, it will be similar to the other two scenarios where the robot is in the human's view prior to approaching them. It will be more like the queue and group scenarios where the robot approaches from the side and interacts in a face-to-face manner.

In the next chapter, we will proceed with a discussion and future directions.

6.3 Summary

There are many custom evaluation methods for SAN systems designed to evaluate a specific contribution, as discussed in Chapter 2, Section 2.1. However, there is a need for measurement scales to study the social intelligence of robots as perceived by humans. Well-validated HRI surveys such as the Negative Attitudes towards Robots Scale (NARS) [23] and the Godspeed Questionnaire Series (GQS) [22] are missing the unique evaluation of a robot's social intelligence. The GQS measures general intelligence. While there are other standardized questions, they get further away from our interest in measuring the perceived social intelligence of robots. The NARS measures the negative attitudes one might already have towards robots, which gets further away from our interest in robots' perceived social intelligence. With this chapter, we have the following contributions:

- *A standardized study design* - We identified a need for perceived social intelligence and the need for new scales to bridge the gap in current HRI scales such as [22, 23]. To evaluate the system and the quality of interaction from a human's standpoint, special emphasis on Perceived Social Intelligence (PSI) [24–26].
- An HRI study investigating the social aspect of the navigation behaviors [27] using PSI scales to evaluate our SAN planner [8] from a bystander's view and also simultaneously validated our newly developed PSI scale [26].

Chapter 7

Discussion & Future Work

In this chapter, we will discuss the following:

- Challenges associated with SAN systems.
- Future directions of USAN.

Prior to this chapter, we have seen our contributions, such as a preliminary work based on a model in Chapter 3, non-linear multi-objective optimization method in Chapter 4, a unified socially-aware navigation in Chapter 5, and method to measure the perceived social intelligence of a robot in Chapter 6. In this chapter, we will discuss challenges that are associated with SAN systems and propose some future directions in the USAN research space.

7.1 SAN Challenges

Robots operating alongside humans in a human environment have to address particular challenges. Here, we will limit our discussion to challenges faced by robots due to their navigation behaviors and how social motion planners can address them. Kruse *et al.* [13] identified Comfort, Sociability, and Naturalness as challenges that SAN planners should tackle in a collaborative human environment.

C1 - Comfort. *Comfort is the absence of annoyance and stress or the easing of a person's feelings of distress in human-robot interactions.*

C2 - Sociability. *Sociability is communicating an ability or willingness to engage in social behavior and adherence to explicit high-level cultural conventions and social norms.*

C3 - Naturalness. *Naturalness is the low-level behavior similarity between humans and robots.*

Adding to the challenges mentioned above, we would add safety, legibility, predictability, fluency, overall efficiency, and acceptance as critical challenges faced by robots using traditional planners:

C4 - Safety. *Safety is the robots unlikeliness to cause harm or injury to human during interaction.*

C5 - Legibility. *Legibility is defined as the clarity in what a robot is doing and can a human identify it.*

C6 - Predictability. *A bystander's ability to anticipate what the robot is currently doing and where it will go.*

C7 - Fluency. *Fluency is the smoothness of the chosen trajectory.* [96]

C8 - Overall Efficiency. *The combined task efficiency of both the robot and a human partner.*

C9 - Acceptance. *Acceptance is the willingness of human to interact with the robot.*

One might think robots with traditional planners are safe as they avoid collisions, but these planners avoid humans within close proximity, which is not always safe. When it comes to the comfort of people interacting with robots, a robot with a traditional planner does not make people feel comfortable as it invades personal space. The comfort and safety of humans around robots depend on the legibility and predictability of motion taken by the robot to reach its goal. Predictability and legibility are fundamentally different and often contradictory properties of motion [97]. In human-human interaction, we use legibility and predictability of a human's trajectory to mutually avoid collision and invasion of personal space by understanding the intent of a person. Similarly, both legibility and predictability of the robot's trajectory help the human partners in understanding the robot's intended actions and vice-versa.

7.2 Limitations and Future Work

Our prior work [5] proposed a non-linear multi-objective optimization based PaCcET local planner using two objectives that was able to execute socially-aware behavior in a hallway setting. We then extended it to include more than two objectives to show that our PaCcET local planner can scale and extend to complex social situations like avoiding activity zones, joining a group, and waiting in a line scenarios [8]. In our prior work [29], we concentrate on the PaCcET-enabled local planner in conjunction with a hybrid context classification method using CNN and SVM to demonstrate that architecture, shown in Figure 5.1 can be used to exhibit socially-aware navigation behaviors in multiple social contexts.

The idea of the USAN system is to have a modular and ROS compatible architecture; individual subsystems can be replaced with an already existing but better system or a novel method. Other researchers can replace one or more subsystems with a custom one and improve the overall system. Other SAN related areas such as planning, control are also emerging, and having a modular and customizable USAN architecture is our contribution to research teams interested in robot navigation in human environments. Real-world, long-term deployment of service robots requires a unified socially-aware navigation method that can exhibit social navigation behavior in every social situation it might encounter in a dense human environment. Our proposed work is novel yet has certain limitations/improvements that can push USAN methods in a real-world deployment. Possible improvements and future work include the following:

1. The trained CNN classifier works well for the trained contexts, but a better solution would be a combination of learning and reasoning. For example, the model learns what objects constitute a context, later when encountered a situation, it should reason about the correct context against a knowledge base from prior experience. Refer to Section 7.4.

2. The cardinal objectives are hand-picked for each trained context. A possible improvement would be to learn these objectives from human-human interactions without being explicitly told.

3. PaCcET works both with and without an evolutionary algorithm. In the proposed approach for a local planner, we chose the latter for computational speed. However, an evolutionary algorithmic (EA) approach to PaCcET at global planning can be advantageous, PaCcET with EA could be used to make copies of the top global plans and apply slight mutations like speed, direction, proxemics to get the best possible SAN global path.

4. When closely observed, human-human navigational interaction benefits from intent communication and intent recognition. An intent module that can both infer and communicate navigational intentions would make our proposed method a predictive system as opposed to a reactive system.

5. Gaze plays a vital role in socially-aware navigation, incorporating social gaze cues in an exciting area of research in SAN [19]. Gaze tracking of human partners and incorporating it in SAN architecture is an interesting area.

7.3 An Extended PSI Study

Social intelligence is the ability to interact effectively with others to accomplish one's goals [88]. Social intelligence is critically important for any robot that will be around people, whether engaged in social or non-social tasks. Some aspects of robotic social intelligence have been included in HRI research [22, 23, 90, 91], but current measures are brief and often include extraneous variables. We designed a comprehensive measure of the perceived social intelligence of robots [24–26]. It would be interesting to see the correlation of SAN challenges, discussed in Section 7.1, with perceived social intelligence. This study design not only helps SAN researchers to understand how a robot is perceived socially while performing its navigation tasks but also what factors are of most importance in navigation, for example, is legibility as important as comfort, etc.

7.4 Ontology-based Knowledge Graph

Our current approach to USAN can be improved by implementing an Ontology-based knowledge graph approach to understanding the context of navigation using multimodal sensor inputs such as visual, speech, and gaze. There is much research activity happening in explainable AI in self-driving car research due to the cost associated with any mishaps. Similarly, social robots not following social norms also have concerns related to human safety (mild when compared to driverless car technology) and comfort. It would be nice to have an explanation of why the robot has chosen particular objectives in a particular situation. Explainability can be of some use both for future tuning/understanding of the system and also the explainability part of the decision-making process itself.

The following subsystems are needed:

1. **Sensor input to text conversion**

   (a) **Automatic captioning of visual sensor input**

   For automatic captioning [98], utilizes training data (images with human-annotated captions) to caption unseen images automatically. Here, the input is an image, and the output is a text describing the image. For example, when a picture with a hallway and a person is given, the output might be something like "*A person walking down the hallway.*"

(b) **Speech to text**

Similar text can also be obtained using speech input, using a speech recognition system, we can convert simple speech into text for further processing. For example, a human partner speaking to a robot, "*Would you accompany me down the hallway?*" Open-source Deep Speech [99] library.

In both cases, above, the behavior has overlaps: accompanies someone might be one extra objective up on normal hallway behavior. When accompanying someone, other objective might be to walk together.

2. **An NLP engine**

   An NLP engine would use techniques like parsing, identifying keywords, etc. to better understand what exactly is "*A person walking down the hallway*" or "*Would you accompany me down the hallway?*" means, understanding what the output of step 1.

3. **Knowledge graph query system**

   The knowledge graph [100] is where the explainability part comes in. Based on the keywords from step 2, we query the knowledge graph to select the objectives for social navigation. This step can be optional if we do not want the explainability part. An alternative can be an ML system with keywords as features. The following capabilities will be needed:

   (a) *A mechanism to build the graph with new information*

(b) *A mechanism to query the knowledge graph*

4. **Multi-objective optimization based planner**

   Once we have the objectives, it is the planning stage where we use the objectives and derive a trajectory that makes social sense. As of now, we have a local planner [8] that does the job, but if needed, a similar method can be employed at global planning as well.

We think a multi-context SAN with a multimodal approach along with the explainability factor is novel and will have a more significant impact on the SAN research community.

7.5 Summary

In this chapter, we discussed challenges associated with SAN systems, namely, *Comfort, Sociability, Naturalness, Safety, Legibility, Predictability, Fluency, Overall Efficiency, Acceptance*. We presented the limitations of our proposed approach, possible improvements directing towards an improved USAN system. We discussed a possible extended perceived social intelligence (PSI) study to understand the correlation of PSI with challenges associated with SAN systems. We discussed a possible improvement in context understanding using an ontology-based knowledge graph method.

Chapter 8

Conclusion

Real-world deployment of social robots that can socially navigate in a human dense human-robot environment may be far off. However, social behavior in one context is not sufficient for long-term acceptance of service robots in a public place. With this work, we demonstrate how differing navigation behavior is appropriate, given different social and environmental contexts, and that visual and laser range information can be used to sense the context autonomously. While there are SAN planners that account for social norms for different contexts, there is a need for a unified architecture that can autonomously sense the ongoing navigational interaction and execute a trajectory that is socially appropriate for that particular interaction context. We presented a novel approach to a unified socially-aware navigation, discussed various subsystems, and implemented it on a differential drive robot.

In this work, we showed that a context classifier, along with a low-level planner utilizing PaCcET, could be used to generate socially optimal trajectories for an autonomously sensed social navigation context. The perception system has generalized to new data and had performed well in recognizing the contexts in real human environments. On the other hand, the navigation results show that the robot was able to account for the social norms while performing navigational actions in various social contexts such as *hallway interactions*, *art gallery situations*, *O-formations when joining a group*, and *waiting in queue situations*.

We identified pitfalls of existing HRI scales, developed perceived social intelligence (PSI) scale that can measure a human's perception of a robot's social intelligence. We validated the PSI scale by evaluating our SAN planner from a bystander's perspective. Participants of the study rated the robot using our proposed SAN planner as socially intelligent. Our bystander experiments support that robots with socially-aware navigation behavior are perceived as socially intelligent when compared to robots with traditional navigation behavior.

Recent research in SAN focused mostly on motion planning aspect, which led to many planners that can account for social norms in human environments. In this work, we implemented a local planner using multi-objective optimization to adhere to social norms in multiple scenarios. In order for robots to execute social navigation, understanding the context is crucial. We discussed various improvements to our proposed USAN method, one of which is an Ontology-based knowledge graph method to make high-level decisions. One advantage of such a method is that by building a knowledge graph, the robot can learn new contexts by adding a new graph node every time it encounters a new situation. A knowledge graph-based solution provides a multi-modal sensor-based approach that is reliable and scalable. An explainable USAN system can be achieved by keep track of knowledge graph queries.

The main contributions of this dissertation work are:

- A conceptualization of a modular Unified Socially-Aware Navigation architecture and its validation in a real robot environment. This contribution resulted in the following capabilities:
  - Most of the existing work in SAN either learning-based methods that need a lot of training data to scale or traditional non-learning based methods that assume linear relationship among social objectives. Our work contributes to a SAN planner that requires no training data and accounts for a non-linear relationship among social objectives.
  - Very few methods in SAN deal with contextual navigation. However, they work with prior information of the context, i.e., the context needs to be given manually by the user. Our work contributes to a learning-based method that can autonomously sense the on-going navigation context.
- Existing evaluation methods in socially-aware navigation either use performance metrics (time, distance traveled, etc.) or HRI measurement scales that are

inadequate in measuring the social intelligence of the robot as perceived by a human. This contribution resulted in the following capabilities:

- We designed new measurement scales to measure perceived social intelligence (PSI) of robots. The measure of social intelligence capabilities did not exist in the HRI community prior to our contribution.
- We validated our newly proposed PSI scales by evaluating our social planner to measure the perceived social intelligence as compared to a traditional planner.

8.1 Peer-Reviewed Publications

Publications that resulted from this dissertation work

[1] Santosh Balajee Banisetty and David Feil-Seifer. Towards a unified planner for socially-aware navigation. In *AAAI Fall Symposium Series: AI-HRI*, October 2018.

[2] Santosh Balajee Banisetty, Scott Forer, Logan Yliniemi, Monica Nicolescu, and David Feil-Seifer. Socially-aware navigation: A non-linear multi-objective optimization approach. ***In Review*** *Transaction on Interactive Intelligent Systems (TiiS) arXiv preprint arXiv:1911.04037*, 2019.

[3] Santosh Balajee Banisetty, Vineeth Rajamohan, Fausto Vega, and David Feil-Seifer. A deep learning approach to multi-context socially-aware navigation. In ***In Review***

IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, March 2020.

[4] Santosh Balajee Banisetty, Meera Sebastian, and David Feil-Seifer. Socially-aware navigation: Action discrimination to select appropriate behavior. In *AAAI Fall Symposium Series: AI-HRI*, November 2016.

[5] Kimberly A. Barchard, Leiszle Lapping-Carr, R. Shane Westfall, Santosh Balajee Banisetty, and David Feil-Seifer. Perceived social intelligence of robots. ***In Press*** *Transactions on Human-Robot Interaction (THRI)*, 2020.

[6] Scott Forer, Santosh Balajee Banisetty, Logan Yliniemi, Monica Nicolescu, and David Feil-Seifer. Socially-aware navigation using non-linear multi-objective optimization. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2018.

[7] AnaLisa Honour, Santosh Balajee Banisetty, and David Feil-Seifer. Perceived social intelligence: A tool to evaluate socially-aware navigation. In ***In Review*** *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, March 2020.

[8] Meera Sebastian, Santosh Balajee Banisetty, and David Feil-Seifer. Socially-aware navigation planner using models of human-human interaction. In *International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 405–410, Lisbon, Portugal, August 2017.

Publications not directly related to dissertation

[1] Janelle Blankenburg, Santosh Balajee Banisetty, S Pourya Hoseini Alinodehi, Luke Fraser, David Feil-Seifer, Monica Nicolescu, and Mircea Nicolescu. A distributed control architecture for collaborative multi-robot task allocation. In *Humanoid Robotics (Humanoids), 2017 IEEE-RAS 17th International Conference on*, pages 585–592. IEEE, 2017.

[2] Monica Nicolescu, Natalie Arnold, Janelle Blankenburg, David Feil-Seifer, Santosh Balajee Banisetty, Mircea Nicolescu, Andrew Palmer, and Thor Monteverde. Learning of complex-structured tasks from verbal instruction. In *Submitted to Proceedings of the International Conference on Intelligent Robots and Systems*, pages 585–592, 2019.

Bibliography

[1] David Feil-Seifer and Maja Matarić. People-aware navigation for goal-oriented behavior involving a human partner. In *Proceedings of the International Conference on Development and Learning (ICDL)*, Frankfurt am Main, Germany, August 2011. doi: 10.1109/DEVLRN.2011.6037331.

[2] David Feil-Seifer and Maja Matarić. Distance-based computational models for facilitating robot interaction with children. *Journal of Human-Robot Interaction*, 1(1):55–77, July 2012. doi: 10.5898/JHRI.1.1.Feil-Seifer.

[3] Santosh Balajee Banisetty, Meera Sebastian, and David Feil-Seifer. Socially-aware navigation: Action discrimination to select appropriate behavior. In *AAAI Fall Symposium Series: AI-HRI*, November 2016.

[4] Meera Sebastian, Santosh Balajee Banisetty, and David Feil-Seifer. Socially-aware navigation planner using models of human-human interaction. In *International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 405–410, Lisbon, Portugal, August 2017. doi: 10.1109/ROMAN.2017.8172334.

[5] Scott Forer, Santosh Balajee Banisetty, Logan Yliniemi, Monica Nicolescu, and David Feil-Seifer. Socially-aware navigation using non-linear multi-objective optimization. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2018.

[6] Logan Yliniemi and Kagan Tumer. Paccet: An objective space transformation to iteratively convexify the pareto front. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 204–215. Springer, 2014.

[7] Edward Twitchell Hall. *The hidden dimension*. Doubleday & Co, 1966.

[8] Santosh Balajee Banisetty, Scott Forer, Logan Yliniemi, Monica Nicolescu, and David Feil-Seifer. Socially-aware navigation: A non-linear multi-objective optimization approach. *arXiv preprint arXiv:1911.04037*, 2019.

[9] Bilge Mutlu and Jodi Forlizzi. Robots in organizations: the role of workflow, social, and environmental factors in human-robot interaction. In *Proceedings of the International Conference on Human-Robot Interaction (HRI)*, pages 287–294, Amsterdam, The Netherlands, 2008. ACM.

[10] Eitan Marder-Eppstein, Eric Berger, Tully Foote, Brian Gerkey, and Kurt Konolige. The office marathon: Robust navigation in an indoor office environment. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 300–307. IEEE, 2010.

[11] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 797–803. IEEE, 2010.

[12] Ross Mead and Maja J Matarić. Autonomous human–robot proxemics: socially aware navigation based on interaction potential. *Autonomous Robots*, pages 1–13, 2016.

[13] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726–1743, 2013.

[14] Billy Okal and Kai O Arras. Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2889–2895. IEEE, 2016.

[15] Beomjoon Kim and Joelle Pineau. Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics*, 8(1):51–66, 2016.

[16] Henrik Kretzschmar, Markus Spies, Christoph Sprunk, and Wolfram Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35(11):1289–1307, 2016.

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

[18] Mahmoud Hamandi, Mike D'Arcy, and Pooyan Fazli. Deepmotion: Learning to navigate like humans. *arXiv preprint arXiv:1803.03719*, 2018.

[19] David V Lu and William D Smart. Towards more efficient navigation for robots and humans. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference On*, pages 1707–1713. IEEE, 2013.

[20] Gonzalo Ferrer, Anais Garrell, and Alberto Sanfeliu. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In *Intelligent robots and systems (IROS), 2013 IEEE/RSJ international conference on*, pages 1688–1694. IEEE, 2013.

[21] Gonzalo Ferrer, Anaís Garrell Zulueta, Fernando Herrero Cotarelo, and Alberto Sanfeliu. Robot social-aware navigation framework to accompany people walking side-by-side. *Autonomous robots*, 41(4):775–793, 2017.

[22] Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International journal of social robotics*, 1(1):71–81, 2009.

[23] Tatsuya Nomura, Tomohiro Suzuki, Takayuki Kanda, and Kensuke Kato. Measurement of negative attitudes toward robots. *Interaction Studies*, 7(3):437–454, 2006.

[24] Kimberly A. Barchard, Leiszle Lapping-Carr, R. Shane Westfall, Santosh Balajee Banisetty, and David Feil-Seifer. Perceived social intelligence (psi) scales test manual. `https://ipip.ori.org/newMultipleconstructs.htm`, 2018.

[25] Kimberly Barchard, Leiszle Lapping-Carr, Shane Westfall, and David Feil-Seifer. Perceived social intelligence of robots. In *Society for Personality and Social Psychology*, Portland, Oregon, February 2019.

[26] Kimberly A. Barchard, Leiszle Lapping-Carr, R. Shane Westfall, Santosh Balajee Banisetty, and David Feil-Seifer. Perceived social intelligence of robots. *(In Press) Transactions on Human-Robot Interaction (THRI)*, 2020.

[27] AnaLisa Honour, Santosh Balajee Banisetty, and David Feil-Seifer. Perceived social intelligence: A tool to evaluate socially-aware navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, March 2020.

[28] Santosh Balajee Banisetty and David Feil-Seifer. Towards a unified planner for socially-aware navigation. In *AAAI Fall Symposium Series: AI-HRI*, October 2018.

[29] Santosh Balajee Banisetty, Vineeth Rajamohan, Fausto Vega, and David Feil-Seifer. A deep learning approach to multi-context socially-aware navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, March 2020.

[30] Karel Čapek. *RUR (Rossum's universal robots)*. Penguin, 2004.

[31] Ronald C Arkin, Ronald C Arkin, et al. *Behavior-based robotics*. MIT press, 1998.

[32] Nils J Nilsson. Shakey the robot. Technical report, SRI INTERNATIONAL MENLO PARK CA, 1984.

[33] Satoshi Tadokoro, Masaki Hayashi, Yasuhiro Manabe, Yoshihiro Nakami, and Toshi Takamori. On motion planning of mobile robots which coexist and cooperate with human. In *iros*, page 2518. IEEE, 1995.

[34] Wolfram Burgard, Armin B Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. The interactive museum tour-guide robot. In *Aaai/iaai*, pages 11–18, 1998.

[35] Wolfram Burgard, Armin B Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot. *Artificial intelligence*, 114(1):3–55, 1999.

[36] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B Cremers, Frank Dellaert, Dieter Fox, Dirk Hähnel, Charles Rosenberg, Nicholas Roy,

Jamieson Schulte, et al. Minerva: A second-generation museum tour-guide robot. In *Robotics and automation, 1999. Proceedings. 1999 IEEE international conference on*, volume 3. IEEE, 1999.

[37] Andrew J Davison and David W Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):865–880, 2002.

[38] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598, 2002.

[39] Yasushi Nakauchi and Reid Simmons. A social robot that stands in line. *Autonomous Robots*, 12(3):313–324, 2002.

[40] Philipp Althaus, Hiroshi Ishiguro, Takayuki Kanda, Takahiro Miyashita, and Henrik I Christensen. Navigation for human-robot interaction tasks. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1894–1900. IEEE; 1999, 2004.

[41] Rachel Gockley, Jodi Forlizzi, and Reid Simmons. Natural person-following behavior for social robots. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 17–24. ACM, 2007.

[42] Annemarie Turnwald and Dirk Wollherr. Human-like motion planning based on game theoretic decision making. *International Journal of Social Robotics*, pages 1–20, 2018.

[43] Jérôme Guzzi, Alessandro Giusti, Luca M Gambardella, Guy Theraulaz, and Gianni A Di Caro. Human-friendly robot navigation in dynamic environments. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 423–430. IEEE, 2013.

[44] Chi-Pang Lam, Chen-Tun Chou, Kuo-Hung Chiang, and Li-Chen Fu. Human-centered robot navigation—towards a harmoniously human–robot coexisting environment. *IEEE Transactions on Robotics*, 27(1):99–112, 2011.

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

[46] Satoru Satake, Takayuki Kanda, Dylan F Glas, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. How to approach humans?-strategies for social robots to initiate interaction. In *Proceedings of the International Conference on Human-Robot Interaction (HRI)*, pages 109–116. IEEE, 2009.

[47] Anaís Garrell and Alberto Sanfeliu. Local optimization of cooperative robot movements for guiding and regrouping people in a guiding mission. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference*

on, pages 3294–3299. IEEE, 2010.

[48] Jorge Rios-Martinez, Anne Spalanzani, and Christian Laugier. Understanding human interaction for probabilistic autonomous navigation using risk-rrt approach. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2014–2019. IEEE, 2011.

[49] Elena Pacchierotti, Henrik I Christensen, and Patric Jensfelt. Human-robot embodied interaction in hallway settings: a pilot user study. In *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pages 164–171. IEEE, 2005.

[50] David V Lu, Dave Hershberger, and William D Smart. Layered costmaps for context-sensitive navigation. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 709–715. IEEE, 2014.

[51] Pedro Santana. Human-aware navigation for autonomous mobile robots for intra-factory logistics. In *Symbiotic Interaction: 6th International Workshop, Symbiotic 2017, Eindhoven, The Netherlands, December 18–19, 2017, Revised Selected Papers*, volume 10727, page 79. Springer, 2018.

[52] Alejandro Bordallo, Fabio Previtali, Nantas Nardelli, and Subramanian Ramamoorthy. Counterfactual reasoning about intent for interactive navigation in dynamic environments. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2943–2950. IEEE, 2015.

[53] Omar A Islas Ramírez, Harmish Khambhaita, Raja Chatila, Mohamed Chetouani, and Rachid Alami. Robots learning how and where to approach people. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, pages 347–353. IEEE, 2016.

[54] Markus Kuderer, Henrik Kretzschmar, and Wolfram Burgard. Teaching mobile robots to cooperatively navigate in populated environments. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3138–3143. IEEE, 2013.

[55] Rafael Ramon-Vigo, Noe Perez-Higueras, Fernando Caballero, and Luis Merino. Transferring human navigation behaviors into a robot local planner. In *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, pages 774–779. IEEE, 2014.

[56] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009.

[57] Yu Fan Chen, Michael Everett, Miao Liu, and Jonathan P How. Socially aware motion planning with deep reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1343–1350. IEEE, 2017.

[58] Anoop Aroor, Susan L Epstein, and Raj Korpan. Online learning for crowd-sensitive path planning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1702–1710. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[59] Grimaldo Silva and Thierry Fraichard. Human robot motion: A shared effort approach. In *Mobile Robots (ECMR), 2017 European Conference on*, pages 1–6. IEEE, 2017.

[60] Collin Johnson and Benjamin Kuipers. Socially-aware navigation using topological maps and social norm learning. In *AAAI*, 2018.

[61] Daniel Althoff, James J Kuffner, Dirk Wollherr, and Martin Buss. Safety assessment of robot trajectories for navigation in uncertain and dynamic environments. *Autonomous Robots*, 32(3):285–302, 2012.

[62] Tomasz Kosiński, Mohammad Obaid, Paweł W Woźniak, Morten Fjeld, and Jacek Kucharski. A fuzzy data-based model for human-robot proxemics. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 335–340. IEEE, 2016.

[63] Konstantinos Charalampous, Ioannis Kostavelis, and Antonios Gasteratos. Recent trends in social aware robot navigation: A survey. *Robotics and Autonomous Systems*, 93:85–104, 2017.

[64] Mohammad Obaid, Eduardo B Sandoval, Jakub Złotowski, Elena Moltchanova, Christina A Basedow, and Christoph Bartneck. Stop! that is close enough. how body postures influence human-robot proximity. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 354–361. IEEE, 2016.

[65] Gonzalo Ferrer and Alberto Sanfeliu. Multi-objective cost-to-go functions on robot navigation in dynamic environments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3824–3829. IEEE, 2015.

[66] Fritz Heider and Marianne Simmel. An experimental study of apparent behavior. *The American Journal of Psychology*, 57(2):243–259, 1944.

[67] Jorge Rios-Martinez, Anne Spalanzani, and Christian Laugier. From proxemics theory to socially-aware navigation: A survey. *International Journal of Social Robotics*, 7(2):137–153, 2015.

[68] Caroline Pantofaru. *Leg Detector*. URL `http://wiki.ros.org/leg_detector`.

[69] L Yliniemi and K Tumer. Complete coverage in the multi-objective paccet framework. In *Genetic and Evolutionary Computation Conference*, 2015.

[70] Mitchell Colby, Logan Yliniemi, Paolo Pezzini, David Tucker, Kenneth Mark Bryden, and Kagan Tumer. Multiobjective neuroevolutionary control for a fuel

cell turbine hybrid energy system. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 877–884. ACM, 2016.

[71] V. Sarfi and H. Livani. A novel multi-objective security-constrained power management for isolated microgrids in all-electric ships. In *2017 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 148–155, Aug 2017. doi: 10.1109/ESTS.2017.8069273.

[72] Logan Yliniemi. Considerations for multiagent multi-objective systems. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1719–1720. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[73] Vahid Sarfi, Hanif Livani, and Logan Yliniemi. A novel multi-objective security-constrained power management for isolated microgrids in all-electric ships. In *Electric Ship Technologies Symposium (ESTS), 2017 IEEE*, pages 148–155. IEEE, 2017.

[74] Brian Gerkey, Richard T Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th international conference on advanced robotics*, volume 1, pages 317–323, Coimbra, Portugal, 2003. IEEE.

[75] Angus Leigh, Joelle Pineau, Nicolas Olmedo, and Hong Zhang. Person tracking and following with 2d laser scanners. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 726–733. IEEE, 2015.

[76] Felix Lindner and Carola Eschenbach. Towards a formalization of social spaces for socially aware robots. In Max Egenhofer, Nicholas Giudice, Reinhard Moratz, and Michael Worboys, editors, *Spatial Information Theory*, pages 283–303, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23196-4.

[77] Richard Kelley, Alireza Tavakkoli, Christopher King, Monica Nicolescu, Mircea Nicolescu, and George Bebis. Understanding human intentions via hidden markov models in autonomous mobile robots. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 367–374. ACM, 2008.

[78] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[79] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

[80] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[81] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller,

Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[82] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks*, pages 92–101. Springer, 2010.

[83] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[84] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[85] François Chollet et al. Keras. `https://keras.io`, 2015.

[86] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems*

Design and Implementation (OSDI 16), pages 265–283, 2016. URL `https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf`.

[87] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[88] Martin E Ford and Marie S Tisak. A further search for social intelligence. *Journal of Educational Psychology*, 75(2):196, 1983.

[89] Kerstin Dautenhahn. Socially intelligent robots: dimensions of human–robot interaction. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 362(1480):679–704, 2007.

[90] Lilia Moshkina. Reusable semantic differential scales for measuring social response to robots. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, pages 89–94. ACM, 2012.

[91] Chin-Chang Ho and Karl F MacDorman. Revisiting the uncanny valley theory: Developing and validating an alternative to the godspeed indices. *Computers in Human Behavior*, 26(6):1508–1518, 2010.

[92] Martin Agran, Carolyn Hughes, Colleen A Thoma, and LaRon A Scott. Employment social skills: What skills are really valued? *Career Development and Transition for Exceptional Individuals*, 39(2):111–120, 2016.

[93] Deniz Tahiroglu, Louis J Moses, Stephanie M Carlson, Caitlin EV Mahy, Eric L Olofson, and Mark A Sabbagh. The children's social understanding scale: Construction and validation of a parent-report measure for assessing individual differences in children's theories of mind. *Developmental psychology*, 50(11): 2485, 2014.

[94] Qualtrics survey platform. *Qualtrics*, 2018.

[95] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL `https://www.R-project.org`.

[96] Guy Hoffman. Evaluating fluency in human-robot collaboration. In *International conference on human-robot interaction (HRI), workshop on human robot collaboration*, volume 381, pages 1–8, 2013.

[97] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 301–308. IEEE Press, 2013.

[98] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)*, 51(6):1–36, 2019.

[99] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[100] Ashutosh Saxena, Ashesh Jain, Ozan Sener, Aditya Jami, Dipendra K Misra, and Hema S Koppula. Robobrain: Large-scale knowledge engine for robots. *arXiv preprint arXiv:1412.0691*, 2014.