TECHNISCHE HOCHSCHULE
WÜRZBURG-SCHWEINFURT

BACHELOR THESIS

Development of a Virtual interactive Laboratory Testbench for Socially-Aware Robot Navigation

---

Author:
Prince Sakariya

Supervisor:
Prof. Dr.-Ing Stefan Friedrich

A thesis submitted in fulfillment of the requirements for the degree of B.Eng. Mechatronics

May 20, 2025

thws
Technische Hochschule
Würzburg-Schweinfurt

Declaration of Authorship

I, Prince Sakariya, declare that this thesis titled, "Development of a Virtual interactive Laboratory Testbench for Socially-Aware Robot Navigation" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

TECHNISCHE HOCHSCHULE WÜRZBURG-SCHWEINFURT

Abstract

Faculty of Electrical and Mechanical Engineering
Department or School Name

B.Eng. Mechatronics

Development of a Virtual interactive Laboratory Testbench for Socially-Aware Robot Navigation

by Prince Sakariya

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

As robots increasingly operate in human-populated environments such as hospitals, shopping malls, and homes, traditional navigation approaches focused solely on collision avoidance and path optimization have proven insufficient. Socially-aware navigation represents a critical advancement that extends beyond obstacle avoidance to incorporate human comfort, cultural norms, and implicit social conventions. This evolution is essential for robots to gain acceptance in shared human spaces. Social navigation is inherently interdisciplinary, bridging robotics, social psychology, cultural anthropology, and human-computer interaction. At its core lies the concept of proxemics—the study of human use of space and its cultural variations—first introduced by anthropologist Edward T. Hall in 1966. Hall's delineation of intimate, personal, social, and public interaction zones provides a fundamental framework for robot navigation in human environments, as respecting these invisible boundaries is crucial for social acceptance. The development of socially-aware mobile robots faces several significant challenges:

- *Complexity of Social Rules* - Human spatial behavior is governed by implicit, context-dependent rules that vary across cultures and situations. These rules are typically learned intuitively by humans but must be explicitly encoded for robots.

- *Experimentation Challenges* - Real-world testing with humans is resource-intensive, potentially risky during early development stages, and difficult to reproduce consistently across different research groups.

- *Interdisciplinary Knowledge Requirements* - Developing effective social navigation systems demands expertise across multiple domains, creating both research and educational barriers.

- *Technical Implementation Hurdles* - Integrating social awareness into existing navigation frameworks requires sophisticated software architecture and computational efficiency to maintain real-time performance.

Recent advances in social robotics have produced numerous platforms and approaches for addressing these challenges (Kästner et al., 2021, Pérez-Higueras et al., 2023, Savva et al., 2019). However, these innovations often remain siloed within research laboratories, with limited accessibility for educational purposes. This accessibility gap represents a significant obstacle for preparing the next generation of roboticists to develop socially-aware systems.

1.2 Problem Statement

Often times, navigation systems treat humans as mere dynamic obstacles without considering social contexts, leading to behaviors that may be technically efficient but socially inappropriate. This research addresses the gap between technically sound and socially acceptable robot navigation in human-shared spaces.

Despite significant research advances in social navigation, three critical problems persist:

1. *Technical-Social Disconnect* - Most deployed robot navigation systems continue to treat humans as mere dynamic obstacles, disregarding social contexts and norms. This approach leads to behaviors that may be computationally efficient but socially inappropriate or discomforting to humans sharing the space.

2. *Educational Access Barriers* - Existing social navigation implementations typically demand extensive technical expertise and computational resources, making them inaccessible for educational purposes. Students face significant hurdles in learning about and experimenting with social navigation concepts due to complex installation requirements, dependencies, and hardware constraints.

3. *Lack of Standardized Learning Tools* - While several research platforms exist, there is a notable absence of standardized educational tools that demonstrate social navigation concepts in a structured, pedagogically sound manner. This gap impedes effective teaching and learning of this increasingly important aspect of robotics.

This thesis addresses these interconnected problems by developing an accessible, education-focused platform that bridges the gap between advanced social navigation research and practical robotics education.

1.3 Research objectives

This bachelor thesis aims to develop an educational system for teaching and experimenting with social robot navigation concepts. The primary goal is to create a virtual laboratory environment that lowers technical barriers and provides structured learning experiences. Specific objectives include:

1. **Create an accessible virtual testbench** that demonstrates various social navigation concepts without requiring extensive technical setup or specialized hardware. This environment will:
   - Integrate existing open-source social navigation implementations
   - Provide a virtualized environment that runs efficiently on standard student hardware
   - Offer a unified interface for interaction with different navigation approaches

2. **Develop structured educational experiments** that:
   - Demonstrate a variety of off-the-shelf social navigation algorithms
   - Illustrate the effects of different proxemic models and parameter configurations
   - Allow comparison between socially-aware and traditional navigation approaches

    - Showcase the impact of environmental context on navigation behavior

3. **Enable hands-on experimentation** through:
    - Real-time costmap parameter modification
    - Customizable evaluation metrics for navigation performance
    - Visualization tools for understanding algorithm decision-making
4. **Create comprehensive educational materials** including:
    - Laboratory exercises with clear learning objectives
    - Documentation explaining theoretical concepts and their implementation
    - Guided exploration activities with progressive complexity

The system is designed primarily for undergraduate and graduate robotics courses, enabling students to develop an intuitive understanding of social navigation principles through direct experimentation before potentially developing their own implementations.

1.4 Scope and Limitations

This thesis focuses on creating an educational platform for social navigation rather than developing novel navigation algorithms or proxemic models. The scope encompasses:

In Scope:

- Integration of existing open-source social navigation implementations
- Development of a virtualized environment for accessible deployment
- Creation of standardized test scenarios for comparative evaluation
- Design of structured educational experiments and supporting materials
- Implementation of visualization tools for algorithm behavior and decision processes
- Extension of existing platforms with missing components required for educational purposes

Out of Scope:

- Development of fundamentally new social navigation algorithms
- Large-scale human studies to validate navigation approaches
- Physical robot implementation and testing
- Cross-platform compatibility beyond the specified virtualization approach
- Comprehensive cultural adaptation of proxemic models

Limitations:

- The system will prioritize educational clarity over computational performance
- Simulated human behaviors will represent simplified models of actual human movement patterns
- The virtualized environment introduces some performance overhead compared to native installation
- The platform targets educational use cases rather than deployment-ready implementations

These scope boundaries ensure the project remains achievable within the constraints of a bachelor thesis while still delivering significant educational value through an accessible platform for teaching social navigation concepts.

Chapter 2

Literature Review

2.1 Social Navigation in Robotics

2.1.1 Definition and Importance

Rios-Martinez, Spalanzani, and Laugier, 2015 gave a compact description of socially-aware navigation: *"Socially-aware navigation is the strategy exhibited by a social robot that identifies and follows social conventions (in terms of management of space) to preserve a comfortable interaction with humans. The resulting behavior is predictable, adaptable, and easily understood by humans."* This definition implies that, from the robot's point of view, humans are no longer perceived only as dynamic obstacles but also as social entities.

The importance of social navigation is paramount for robots (Kruse et al., 2013) intended to operate in human-centric environments such as homes, hospitals, shopping malls, offices, and public spaces. As robots become increasingly integrated into our daily lives, their ability to interact seamlessly and naturally with humans is crucial for their acceptance and widespread adoption. Poor social navigation can lead to discomfort, anxiety, inefficiency, and even safety hazards for humans. Conversely, robots capable of navigating socially appropriately can enhance human productivity, provide assistance in various tasks, and improve overall quality of life. Furthermore, in applications like assistive robotics and healthcare, the ability of a robot to navigate in close proximity to individuals, while maintaining their comfort and safety, is fundamental (Möller et al., 2021).

2.1.2 Key Challenges

Developing robust social navigation capabilities in robots presents several key challenges:

- **Human Behavior Prediction:** Humans are inherently unpredictable. Their motion patterns are influenced by a multitude of factors including their goals, intentions, emotions, social context, and cultural background. Accurately predicting human trajectories and intentions is a significant challenge, requiring sophisticated models that can capture the nuances of human behavior.

- **Social Norms and Etiquette:** Navigating social environments requires adherence to a complex set of implicit social norms and etiquette. These norms can vary across cultures and situations. Robots need to understand and respect these norms, such as maintaining appropriate personal space, avoiding sudden or erratic movements, and yielding to pedestrians in certain situations.

- **Uncertainty and Dynamic Environments:** Human-populated environments are inherently dynamic and uncertain. People may change their direction or speed unexpectedly, form groups, or engage in interactions that affect robot navigation. Robots must be able to perceive and react to these dynamic changes in real-time while maintaining their navigation goals.

- **Computational Complexity:** Implementing sophisticated models for human behavior prediction, social norm understanding, and real-time adaptation can be computationally demanding. Developing efficient algorithms that can run on robot platforms with limited computational resources is a crucial challenge.

- **Evaluation Metrics and Benchmarking:** Defining appropriate metrics to evaluate the social acceptability and effectiveness of robot navigation is challenging. Establishing standardized benchmarks and simulation environments is necessary to facilitate the comparison and progress of different social navigation approaches.

2.2 Proxemics and Human-Robot Interaction

2.2.1 Proxemics Theory

Edward Hall's theory of proxemics Hall et al., 1968 suggests that people will maintain differing degrees of personal distance depending on the social setting and their cultural backgrounds.

- *Intimate space* - the clossest "bubble" of space surrounding a person. Entry into this space is acceptable only for the closest friends and intimates.

- *Social and consultative spaces* - the spaces in which people feel comfortable conducting routine social ineractions with acquaintances as well as strangers.

- *Public space* - the area of space beyond which people will perceive iteractions as impersonal and relatively anonymous.

The main contribution of Hall's Proxemics into path planning consists of providing a framework to build social maps, i.e. dynamic maps in which humans are perceived as obstacles following the definition of Hall's personal space. The work by Henkel et al., 2014 evaluates different distance strategies by how they affect the human's perception of the robot's likeability, intelligence and submissiveness.

2.2.2 Application in Robotics

Proxemics has been employed in robotics to guide the development of spatially aware path planning. Robots use proxemic principles to maintain comfortable distances from humans, avoid intrusions into personal zones, and adjust their behavior based on environmental context. Several implementations integrate proxemic rules into costmaps and behavior trees to ensure adherence to social comfort zones, increasing user satisfaction and perceived safety. Proxemic-aware navigation also supports differentiated behaviors depending on robot intent — for example, service robots vs. delivery robots — providing richer HRI experiences

2.3 Simulation Platforms for Social Navigation

Helbing and Molnár, 1995, show that pedestrian motion can be described by a simple social force model for individual pedestrian behavior. The social force model is an essential component in many platforms including Arena-rosnav Kastner et al., 2022, HuNavSim Pérez-Higueras et al., 2023 etc. to simulate the pedestrian movements. The navigation comprises of path planners and costmaps. There are two types of planners - global planner determines a path from the current location to the goal location, and a local planner follows the global path. Costmaps are created using static maps and real-time data from onboard sensors.

Sacco, Recchiuto, and Mårtensson, 2024, use A* global planner and an MPC, with a detailed cost function to achieve advanced social navigation capabilities with the help of SMPC (Social Model Predictive Control) software stack. They leverage the predicitvity of MPC and the reactivity of SFM, modelling the pedestrian motion.

Chen et al., 2018, presented SA-CADRL (Socially Aware Collision Avoidance with Deep Reinforcement Learning) to explain/induce socially aware behaviors in a RL framework. They generalized to multiagent ($n > 2$) scenarios through developing symmetrical neural network structure, and demonstrated on robotic hardware autonomous navigation at human walking speed in a pedestrian-rich environment.

There exist various other approaches based on traditional algoritms as well as novel Neural Network, Deep Reinforcement Learning etc. In the next section, several simulation platforms, their advantages and disadvantages are described.

2.3.1 Overview of Existing Platforms

This section focuses on capabilities, features, and limitations of each environment for making specific recommendations for implementation purposes.

Habitat-Sim is a flexible, high performance 3D simulator with a focus on embodied AI research including navigation tasks Savva et al., 2019, Szot et al., 2022. It is capable of running thousands of simulations in parallel, with photo-realistic 3D environments from real-world scans, semantic scene understanding and support for multiple sensors (RGB, depth, semantic segmentation). It is however limited by a lack of in-built social navigation features, human motion models can be integrated however it requires some technical understanding of AI and has a steep learning curve.

SEAN 2.0 is specifically designed for social navigation research with emphasis on human behavior modeling Tsoi et al., 2022. It is a high fidelity, extensible, and open-source simulation platform for fair evaluation of social navigation algorithms. Environments correspond to the physical, static elements in a scenario in Unity. *SEAN 2.0* [1] includes warehouss, lab, and outdoor environments from *SEAN 1.0* with annotations for new pedestrian behaviours. It also provides numerous evalutation metrics including path efficiency, path irregularity, completed, totol time etc. It is limited by the simulation backend options as it limited to Unity.

HuNavSim focuses specifically on realistic human navigation behavior modelling Pérez-Higueras et al., 2023. It is a new open-source software library used to simulate human navigation behaviors. The tool, programmed under the new ROS 2 framework, can be employed to control the human agents of different general robotics

[1] https://sean.interactive-machines.com/

simulators. It utilizes a Social Costmap Layer (a custom ROS 2 version of the social navigation layers implemented in ROS 1 [2]) It provides a wide range of metrics for various scenarious, however it is limited by the number of available planners and lacks documentation which makes it a difficult choice for educational applications.

Arena-rosnav is an open-source modular benchmark environment built on ROS that specifically targets socially aware navigation Kästner et al., 2021. It provides support a total of 15 navigation planners (see table: 2.1) which include classic, hybrid and learning-based planners. It (Arena Rosnav 3.0)[3] provides support for several both 2D and 3D simulators including Flatland, Rviz, Gazebo, Unity and provides an interface to integrate other simulation softwares, worlds like *Hospital, Canteen, Campus, Factory and Warehouse* are supported in Gazebo and *Hospital, Restaurant School, Japanese Garden and Warehouse* are supported in Unity, additionally users can add or create new worlds. Multiple robots including *tutlebot3-burger, jackal, ridgeback, agvota, tiago, robotino, youbot, turtlebot3_waffle_pi etc.* are supported and users can add more easily. It's limitations however are that it is a little above 50 GB in size and computationally intensive.
TODO: fix appearance of citations.

TABLE 2.1: Available Navigation Planners

Classic	Hybrid	Learning-based
TEB Rösmann, Hoffmann, and Bertram, 2015	Applr Xiao et al., 2020	ROSNavRL Kästner et al., 2021
DWA Khatib, 1985	LfLH Xiao et al., 2022b	RLCA Long et al., 2018
MPC Rösmann, 2019	Dragon Xiao et al., 2022a	Crowdnav Chen et al., 2019
Cohan Teja Singamaneni, Favier, and Alami, 2021	TRAIL Xiao et al., 2022a	SARL Li et al., 2019
		Arena Kästner, Marx, and Lambrecht, 2020
		CADRL Everett, Chen, and How, 2018
		Navrep Dugas et al., 2020

2.3.2 Comparison and Suitability for Education

Simulation tools differ in usability, fidelity, extensibility, and educational value:

- **Habitat-Sim** offers high visual fidelity but is difficult for beginners due to limited social behavior modules.
- **SEAN 2.0** provides tailored environments for social navigation and good evaluation metrics, making it suitable for advanced education.
- **HuNavSim** is lightweight and ROS 2 compatible but suffers from poor documentation.

[2] https://github.com/robotics-upo/nav2_social_costmap_plugin
[3] https://3.arena-rosnav.org/

- **Arena-Rosnav** is ideal for research and advanced robotics courses, providing a wide range of planners and robot models, but requires high computational resources.

This thesis utilizes Arena-Rosnav, the reasons for selecting it are discussed in section 3.1.1

2.4 Summary of Gaps and Research Opportunities

While considerable progress has been made, several gaps persist:

- Lack of universal benchmarks for social navigation evaluation.
- Insufficient modeling of nuanced social behaviors (e.g., group dynamics, cultural norms).
- Limited cross-simulator compatibility.
- High barrier to entry due to complex setups or resource demands.
- Sparse integration between proxemics theory and learning-based planners.

Opportunities exist in developing:

- Lightweight, user-friendly simulation tools for education.
- Integrative frameworks combining proxemics.

Chapter 3

System Design and Implemenation

This chapter presents a comprehensive overview of the system design and implementation details of our social navigation educational platform. We begin with a thorough examination of the simulation environment, including the rationale behind our platform selection and the architectural design choices. We then detail the specific customizations made to enhance the educational value of the system. Finally, we describe the implementation of various social navigation scenarios, focusing on their design, challenges, and parameterization options.

3.1 Overview of the Simulation Environment

The simulation environment forms the foundation of our educational platform for social robot navigation. Its design required careful consideration of educational objectives, technical capabilities, and accessibility for students with varying levels of robotics experience. After evaluating several options, we selected Arena-Rosnav as our primary simulation platform.

3.1.1 Platform Selection and Justification

The selection of Arena-Rosnav as our simulation environment was based on a systematic evaluation of requirements for both research-grade social navigation experiments and effective educational tools. We identified several critical requirements:

- **ROS Integration**: The platform needed to offer seamless integration with the Robot Operating System (ROS) ecosystem, which remains the predominant middleware in robotics research and education.
- **Social Navigation Support**: The environment needed native support for human agents and the ability to model social interactions.
- **Scalability**: The platform should support scenarios with varying complexity, from simple corridor interactions to complex multi-agent environments.
- **Reproducibility**: The environment needed to provide deterministic simulation capabilities for scientific reproducibility and consistent educational experiences.
- **Extensibility**: The platform should allow for straightforward customization and extension to incorporate new navigation algorithms and social behavior models.

- **Accessibility**: The simulation environment needed to be approachable for undergraduate students while still offering the depth required for graduate-level research.

Arena-Rosnav Kästner et al., 2024 satisfies these requirements with its comprehensive feature set. Specifically, Arena-Rosnav provides:

1. **Modular Architecture**: Arena-Rosnav offers a highly modular architecture that separates robot models, navigation algorithms, and environmental factors. This modularity allows students to focus on specific components without needing to understand the entire system at once.

2. **Multiple Robot Support**: The platform supports various robot models (including TurtleBot3, Jackal, and custom platforms), allowing for experimentation with different kinematic constraints and sensor configurations without changing the underlying navigation code.

3. **Advanced Human Simulation**: Arena-Rosnav implements the Social Force Model (SFM) and ORCA (Optimal Reciprocal Collision Avoidance) for human agent movement, creating realistic crowd behaviors. This allows for more authentic testing of social navigation algorithms compared to environments with simplistic human models.

4. **Scenario Generation**: The platform includes tools for procedural generation of environments and scenarios, which facilitates systematic testing across a broad spectrum of conditions. This feature is particularly valuable for our educational objectives, as it allows students to test navigation algorithms against consistent challenges.

5. **Benchmarking Tools**: Arena-Rosnav provides built-in performance metrics and visualization tools that align well with our requirement for educational clarity and research rigor.

6. **Active Community**: The platform has an active development community, ensuring longevity and continuous improvement of the codebase.

7. **Open-Source Accessibility**: As an open-source project, Arena-Rosnav allows for complete transparency and modification, which is essential for both research innovation and educational customization.

Other platforms considered included HuNavSim and Habitat-Sim. While these alternatives offered some advantages, Arena-Rosnav provided the most comprehensive feature set for our specific needs, particularly in human-robot interaction scenarios and educational flexibility. Gazebo standalone environments lacked standardized human models and would have required significant development to reach feature parity with Arena-Rosnav. Stage simulator, while lightweight, lacks the physical fidelity needed for realistic social navigation research. Custom environments would have required prohibitive development time without offering significant advantages over the already mature Arena-Rosnav platform.

Furthermore, Arena-Rosnav's recent updates have enhanced its crowd simulation capabilities, with implementations of the latest pedestrian modeling approaches and improved computational efficiency that makes it suitable for running on standard laboratory computers available to students.

3.1.2 System Architecture

The architecture of our simulation environment builds upon the core Arena-Rosnav structure while integrating additional components specific to social navigation education and research. The system architecture, illustrated in Figure 3.1, consists of five primary layers:

FIGURE 3.1: System architecture of the social navigation simulation environment

Core Simulation Layer

At the foundation of our architecture is the core simulation layer, which provides the physical simulation environment. This layer:

- Utilizes Gazebo as the underlying physics engine, providing realistic simulation of robot dynamics, sensor interactions, and environmental physics.
- Implements a temporal discretization model that balances simulation fidelity with computational efficiency, using a variable time-step approach that allocates more computational resources to complex interaction scenarios.
- Provides environmental rendering through both Gazebo's built-in visualization and custom RViz configurations optimized for observing social navigation behaviors.
- Manages simulation state, including robot poses, human positions and velocities, and environmental object states, all synchronized through ROS topic communication.

Human Simulation Layer

The human simulation layer is responsible for generating realistic pedestrian behaviors:

- Implements multiple pedestrian models including SFM (Social Force Model), which simulates humans as particles influenced by social and physical forces, and ORCA (Optimal Reciprocal Collision Avoidance), which provides more computationally efficient local navigation for large crowds.
- Models human intentions through goal-directed behavior with configurable parameters for walking speed, personal space preferences, and decision-making characteristics.
- Includes crowd formation models that can simulate common social groupings (pairs, families, tour groups) with appropriate inter-personal spacing and coordination behaviors.
- Provides a human behavior API that allows for scripted scenarios where human agents follow predetermined paths or respond to environmental triggers, essential for reproducible educational experiments.

Robot Navigation Layer

The robot navigation layer encapsulates the navigation stack components:

- Integrates the standard ROS navigation stack with customizable costmap layers, including traditional layers (static, obstacle, inflation) and social navigation layers (proxemics, passing).
- Implements multiple local planners, including the default Dynamic Window Approach (DWA) planner, Timed Elastic Band (TEB) planner, and custom social planners developed for this research.
- Provides standardized interfaces for global planners, including A*, Dijkstra, and RRT variants, with hooks for social cost integration at the global planning level.
- Exposes navigation parameters through both ROS parameter server and structured configuration files, facilitating systematic experimentation by students.

Social Interaction Layer

The social interaction layer, which is our primary contribution to the Arena-Rosnav architecture, implements social navigation concepts:

- Provides the proxemics costmap layer that dynamically generates cost fields around humans based on proxemic theories of personal, social, and public space.
- Implements the passing layer that models directional preferences when navigating around humans, incorporating cultural and contextual variables.
- Includes visualization components that render proxemic zones, anticipated human trajectories, and robot planning decisions in human-interpretable formats.
- Offers a social metrics collector that measures and records quantities such as minimum separation distance, personal space intrusions, and trajectory smoothness during navigation.

Educational Interface Layer

The educational interface layer makes the system accessible to students:

- Provides simplified launch files with graduated complexity, allowing students to start with basic scenarios and progressively engage with more complex system components.
- Implements parameter templates for different experimental configurations, reducing the learning curve for system customization.
- Includes data collection scripts that automatically store and process experimental results in formats suitable for analysis and report generation.
- Offers debug visualization through custom RViz configurations that highlight relevant aspects of social navigation for educational understanding.

Data flow through this architecture follows a consistent pattern. Environmental perception (simulated sensor data) flows from the core simulation layer to the robot navigation layer. The human simulation layer provides human position and velocity data to both the core simulation layer and the social interaction layer. The social interaction layer processes human data to generate social costs, which are then integrated into the navigation layer's planning processes. The educational interface layer primarily consumes data from other layers for visualization and analysis while providing configuration inputs that modify layer behaviors.

This multi-layered architecture provides both the technical depth required for research-grade experimentation and the conceptual clarity needed for educational applications. The clear separation of concerns between layers allows students to focus on specific aspects of social navigation without being overwhelmed by system complexity.

3.2 Customization for Educational Use

While Arena-Rosnav provides an excellent foundation for social navigation research, significant customization was necessary to optimize it for educational purposes. Our customizations focused on reducing complexity barriers, providing educational scaffolding, and enhancing the visualization of social navigation concepts.

3.2.1 Simplified Configuration System

We developed a tiered configuration system that allows students to engage with increasing levels of complexity:

- **Level 1: Parameter Templates** - Predefined parameter sets that allow students to run complete experiments without parameter tuning, focusing instead on observing and analyzing results.
- **Level 2: Guided Configuration** - Template files with clearly documented parameters that students can modify within safe ranges, encouraging experimentation while preventing system failures due to invalid configurations.
- **Level 3: Full Configuration** - Complete access to all system parameters for advanced students and research projects, with extensive documentation of parameter interactions and effects.

This tiered approach enables instructors to match system complexity to student preparation and learning objectives. For introductory robotics courses, Level 1 configurations allow students to focus on conceptual understanding. For advanced courses, Level 3 configurations promote deeper technical engagement.

3.2.2 Enhanced Visualization Tools

We developed specialized visualization components that make social navigation concepts visually explicit:

- **Proxemic Visualization** - Custom RViz plugins that render personal, social, and public spaces around simulated humans as color-coded regions with appropriate transparency to maintain visibility of the underlying environment.

- **Planning Visualization** - Visualization tools that show considered and rejected paths, highlighting how social costs influence navigation decisions.
- **Cost Function Visualization** - Heat map displays that show the distribution of navigation costs throughout the environment, making the abstract concept of cost functions tangible for students.
- **Metric Displays** - Real-time displays of key social navigation metrics (minimum separation distance, personal space intrusions, path efficiency) that allow students to immediately observe the effects of parameter changes.

These visualization enhancements are crucial for educational effectiveness, as they make abstract algorithmic concepts visually concrete. Students can directly observe how proxemic models influence robot behavior, building intuition that complements their theoretical understanding.

3.2.3 Structured Learning Scenarios

We developed a progression of learning scenarios that introduce social navigation concepts in a structured sequence:

- **Observation Scenarios** - Initial scenarios where students observe pre-configured robots navigating around humans, focusing on developing observational skills before implementation.
- **Parameter Exploration Scenarios** - Scenarios where students modify specific parameters and observe outcomes, building cause-effect understanding of navigation algorithms.
- **Comparative Analysis Scenarios** - Scenarios that run multiple navigation approaches simultaneously for side-by-side comparison, highlighting the advantages and limitations of different approaches.
- **Challenge Scenarios** - Complex environments that test student understanding through navigation problems requiring thoughtful parameter selection and algorithm choice.

These structured scenarios provide a pedagogical progression that aligns with educational best practices for skills development. Students move from guided observation to independent problem-solving, developing both technical skills and conceptual understanding.

3.2.4 Documentation and Tutorials

We developed comprehensive educational materials specifically designed for the learning progression:

- **Conceptual Tutorials** - Materials explaining the theoretical foundations of social navigation, proxemics theory, and human-aware planning.
- **Technical Walkthroughs** - Step-by-step guides for system installation, configuration, and operation, accommodating students with varying technical backgrounds.

- **Experimental Guides** - Structured procedures for conducting experiments, including data collection protocols, analysis methods, and interpretation guidelines.

- **Troubleshooting Resources** - Common error documentation and resolution strategies, reducing the frustration factor often associated with complex simulation environments.

Unlike general-purpose documentation, these materials are explicitly aligned with educational objectives and student developmental progression. They incorporate pedagogical best practices such as scaffolded learning, worked examples, and conceptual bridges between theory and implementation.

3.3 Implementation of Social Navigation Scenarios

The effectiveness of our simulation environment depends significantly on the quality of the implemented scenarios. We developed specific scenarios that highlight key aspects of social navigation while providing controlled experimental conditions for both research and education.

3.3.1 Corridor Scenario Implementation

The corridor scenario represents one of the most common human-robot interaction contexts in indoor environments. This scenario tests a robot's ability to navigate in confined spaces while respecting human proxemic preferences.

Physical Environment Design

The corridor environment is implemented with the following characteristics:

- **Dimensions** - The corridor measures 2.5 meters in width and 15 meters in length, representing a typical institutional hallway dimension that is narrow enough to force interaction decisions but wide enough to allow multiple navigation strategies.

- **Boundary Definition** - Corridor walls are defined as both collision objects in the physics engine and as occupied cells in the static costmap layer, ensuring consistent representation across the simulation stack.

- **Surface Properties** - The corridor floor uses a friction coefficient that accurately models typical indoor flooring, ensuring realistic robot motion dynamics, particularly for differential drive platforms that may experience wheel slip during rapid direction changes.

- **Contextual Elements** - The environment includes typical corridor features such as doorways, wall fixtures, and occasional side tables that add environmental complexity without fundamentally changing the navigation challenge.

The physical dimensions were carefully selected based on architectural standards and empirical studies of human comfort in corridor passing interactions. The width of 2.5 meters creates what proxemics researchers call a "decision zone" where robots must make explicit passing choices rather than simply maintaining maximum separation.

Human Agent Configuration

The human agents in the corridor scenario are configured to create realistic navigation challenges:

- **Movement Patterns** - Humans follow bidirectional paths along the corridor with randomized entry timing to create varied encounter scenarios (oncoming, overtaking, group passing).
- **Walking Characteristics** - Human walking speeds are drawn from a normal distribution with mean 1.4 m/s and standard deviation 0.2 m/s, based on empirical human walking speed studies.
- **Group Formations** - The scenario includes both individual pedestrians and small groups (pairs and triads) that maintain social formation during navigation, presenting more complex spatial challenges.
- **Attention Models** - Humans are configured with variable attention states, including attentive (aware of and responsive to the robot) and inattentive (minimally responsive), reflecting the range of human behaviors in real environments.

These configurations create a rich variety of interaction scenarios while maintaining enough consistency for experimental repeatability. The human models were calibrated against empirical observations of pedestrian behavior in corridor environments to ensure behavioral realism.

Scenario Variants

We implemented multiple variants of the corridor scenario to isolate specific navigation challenges:

- **Basic Passing** - A single oncoming human in an otherwise empty corridor, testing fundamental passing behavior.
- **Multiple Passing** - Multiple humans moving in both directions, requiring the robot to handle sequential interactions.
- **Group Navigation** - Scenarios featuring human groups that occupy more horizontal space, testing the robot's ability to navigate around collective social entities.
- **Crowded Corridor** - High-density human traffic that approaches the corridor's capacity limit, testing navigation in socially dense environments.

These variants allow for systematic investigation of specific social navigation challenges while maintaining the basic corridor context. They progress from simple to complex interactions, supporting both structured education and research depth.

3.3.2 Corner Scenario Implementation

The corner scenario tests a robot's ability to navigate blind corners safely and socially, a significant challenge in many indoor environments. This scenario is particularly valuable for evaluating how robots handle limited visibility combined with potential human encounters.

Physical Environment Design

The corner environment has the following characteristics:

- **Geometry** - A 90-degree corner with 2.5-meter wide corridors on each side, creating a navigation challenge where the robot cannot see around the corner until committed to the turn.
- **Visibility Constraints** - The corner is designed with solid walls that block both visual and laser sensor data, preventing the robot from detecting humans on the other side of the corner until physically approaching the intersection.
- **Approach Zones** - The corridors extend 8 meters from the corner in each direction, providing sufficient distance for the robot to adjust its approach velocity based on corner-handling strategies.
- **Environmental Markers** - Subtle environmental cues (floor coloration changes, wall textures) are included 2 meters before the corner, allowing for the development and testing of anticipatory algorithms that recognize corner contexts.

The corner geometry was specifically designed to create the partial observability problem that makes corner navigation socially challenging. The 90-degree angle represents the most common architectural corner configuration while creating a genuine blind spot that cannot be fully resolved with current sensor technologies.

Human Agent Configuration

Human agents in the corner scenario are configured to create challenging interaction situations:

- **Trajectory Patterns** - Humans approach the corner from multiple directions with paths that create potential collision trajectories if the corner is navigated without social awareness.
- **Variable Speeds** - Human walking speeds are more variable in this scenario (0.8 m/s to 1.8 m/s) to test robot adaptation to different human approach velocities.
- **Corner Behavior Models** - Human agents implement realistic corner navigation behaviors, including trajectory adjustment, speed modulation when approaching blind corners, and recovery behaviors when near-collisions occur.
- **Attention Distribution** - The scenario includes a higher proportion of distracted humans (e.g., looking at mobile devices) who may not actively avoid the robot, creating a more realistic and challenging navigation environment.

These human configurations create the challenging corner interactions observed in real environments where humans must negotiate shared space with limited advance information. The behavior models were validated against observational studies of human corner navigation in public buildings.

Scenario Variants

We implemented several corner scenario variants to isolate specific navigation challenges:

- **Basic Corner Encounter** - Single humans approaching from around the corner, testing fundamental corner navigation safety.
- **Concurrent Approaches** - Multiple humans approaching the corner simultaneously from different directions, requiring more complex negotiation.
- **Stationary Human** - A stationary human positioned just around the corner, testing the robot's ability to handle sudden static obstacles in socially appropriate ways.
- **Dynamic Speed Adjustment** - Humans who change speed as they approach the corner, testing the robot's adaptation to changing human intentions.

These variants systematically explore the corner navigation challenge space, allowing for controlled experiments that isolate specific aspects of the problem. The progression from basic to complex scenarios supports both educational sequencing and systematic research.

3.3.3 Parameterization and Modifiability

A key requirement for both educational use and research flexibility is comprehensive parameterization of the simulation environment. We implemented a multi-level parameter system that balances flexibility with usability.

Physical Environment Parameters

The physical environments are parameterized through:

- **Dimensional Parameters** - Corridor width, length, corner angles, and approach distances can be modified through configuration files, allowing for systematic exploration of spatial constraints.
- **Object Placement** - Environmental objects (side tables, plants, chairs) can be procedurally placed with controllable density and distribution parameters, enabling systematic variation of environmental complexity.
- **Sensor Noise Models** - Realistic sensor noise can be applied with configurable intensity, allowing students to explore robustness to perceptual uncertainty.
- **Surface Properties** - Floor friction, material reflectivity, and other physical properties can be adjusted to test navigation robustness across different environmental conditions.

These parameters allow for systematic manipulation of the physical challenges facing the navigation system. The parameterization uses a hierarchical YAML structure that organizes related parameters into logical groups, enhancing usability.

Human Behavior Parameters

Human agent behavior is highly parameterized:

- **Motion Parameters** - Walking speed distributions, acceleration profiles, and directional variance can be adjusted to model different human movement patterns.

- **Social Parameters** - Personal space preferences, group cohesion forces, and collision avoidance aggressiveness can be modified to represent different cultural and individual behavioral norms.
- **Attentional Parameters** - Human attention to the robot, responsiveness to potential collisions, and path predictability can be adjusted to test robot adaptation to different human awareness levels.
- **Trajectory Generation** - Goal positions, waypoint density, and path constraints can be modified to create different human movement patterns through the environment.

These parameters allow for modeling a wide range of human behaviors, from highly predictable and robot-aware to unpredictable and inattentive. The parametric models are based on established pedestrian simulation research, ensuring behavioral realism.

Navigation Algorithm Parameters

The robot navigation stack is comprehensively parameterized:

- **Costmap Parameters** - Each costmap layer (static, obstacle, inflation, proxemics, passing) has adjustable parameters controlling its influence on navigation decisions. Key parameters include:
  - Static layer: update frequency, obstacle inflation
  - Obstacle layer: obstacle detection thresholds, clearing thresholds
  - Inflation layer: inflation radius, cost scaling factor
  - Proxemics layer: personal/social/public space radii, cost decay functions
  - Passing layer: preferred passing side, minimum passing distance
- **Local Planner Parameters** - The DWA and TEB planners expose parameters controlling:
  - Path scoring weights (path distance, goal alignment, obstacle clearance)
  - Kinematic constraints (maximum velocities, accelerations)
  - Lookahead distance and planning horizon
  - Social navigation specific weights and constraints
- **Global Planner Parameters** - A*, Dijkstra, and RRT planners expose parameters for:
  - Heuristic weighting and behavior
  - Sampling strategy and density
  - Path optimization and smoothing
  - Social cost integration methods

These navigation parameters allow for systematic exploration of algorithm behavior across a wide range of configurations. The parameters are exposed through both ROS parameter server mechanisms and configuration files, supporting both runtime adjustment and systematic experimental design.

Parameter Configuration Interface

To make this extensive parameterization accessible, we developed a layered configuration interface:

- **Scenario Configuration Files** - Top-level files that define complete scenarios, including environmental setup, human behavior patterns, and robot configurations.
- **Component Configuration Files** - Modular files for specific system components (human models, robot navigation, sensors) that can be combined into complete scenarios.
- **Parameter Templates** - Predefined parameter sets representing specific navigation strategies or experimental conditions that can be applied to any scenario.
- **Educational Parameter Guides** - Documentation that explains parameter effects, reasonable ranges, and expected behavioral impacts for student experimentation.

This layered approach makes the complex parameter space accessible to users with different expertise levels. Students can begin with scenario-level configurations and progressively engage with component-level and individual parameters as their understanding develops.

3.4 Summary

This chapter has detailed our system design and implementation choices for creating an effective social navigation simulation environment. We selected Arena-Rosnav as our foundation due to its comprehensive feature set, particularly its support for realistic human simulation and integration with the ROS ecosystem. Our implementation extends this platform with specialized components for social navigation research and education.

The system architecture provides clear separation between simulation, human modeling, robot navigation, social interaction, and educational interface layers. This modular design supports both research flexibility and educational clarity. Our customizations for educational use, including simplified configuration systems, enhanced visualizations, and structured learning scenarios, make complex social navigation concepts accessible to students.

The implemented scenarios—corridor and corner navigation—provide controlled environments for investigating key social navigation challenges. These scenarios have been carefully designed to balance realism with experimental control, supporting both educational objectives and research rigor. The comprehensive parameterization of the environment enables systematic exploration of navigation behaviors across a wide range of conditions.

This implementation provides the technical foundation for the student experiments and research contributions described in subsequent chapters. By combining research-grade simulation capabilities with educational accessibility, our system supports both the advancement of social navigation algorithms and the effective teaching of social navigation concepts.

Chapter 4

Methodology

Building upon the social navigation simulation environment described in Chapter 3, this chapter outlines the methodological framework used to evaluate both the technical efficacy of social navigation algorithms and their educational value. Our methodology addresses the dual research objectives of advancing social navigation algorithm performance and developing effective educational approaches for teaching social navigation concepts.

4.1 Experiment Design

The experimental design follows a systematic approach that isolates specific aspects of social navigation while maintaining sufficient environmental complexity to yield meaningful results. We employed both controlled comparative experiments for algorithm evaluation and structured learning sequences for educational assessment.

4.1.1 Scenario Configuration

To ensure experimental rigor, we developed a standardized approach to scenario configuration that maintains consistency across experimental trials while allowing for systematic parameter variation:

- **Scenario Definition Protocol** – Each experimental scenario is defined through a hierarchical YAML configuration that specifies environment geometry, human agent parameters, robot configuration, and navigation algorithm settings. This approach enables precise replication of experimental conditions.

- **Initialization Procedures** – Robot and human agent positions are initialized using consistent starting configurations with controlled randomization where appropriate. For each scenario variant, we defined standard starting regions rather than exact positions, allowing natural variation while maintaining comparable interaction conditions.

- **Trial Duration and Termination Criteria** – Experimental trials run until one of several termination conditions is met: successful navigation to goal, collision event, timeout (120 seconds), or deadlock detection (no progress for 15 seconds). This approach ensures comparable data collection across algorithm variants.

- **Scenario Progression** – Experiments follow a progression from baseline scenarios (single human, simple environment) to complex scenarios (multiple humans, challenging environmental features). This structured progression supports both incremental algorithm testing and pedagogical sequencing.

For each of the core scenarios described in Chapter 3 (corridor and corner navigation), we developed specific configuration variants that systematically increase in complexity: Through Arena-Rosnav's scenario generation capabilities, we programmatically created these scenarios with consistent parameters while allowing controlled variation in human movement patterns. This approach yields statistically meaningful results while capturing the natural variability of human-robot interactions.

TABLE 4.1: Scenario Progression for Experimental Evaluation

Complexity Level	Corridor Scenario	Corner Scenario
Basic	Single oncoming human, centered approach	Single human around corner, moderate speed
Intermediate	Multiple humans, bidirectional flow	Multiple humans with varied approach speeds
Advanced	Group formations with social structures	Blind corner with distracted humans
Complex	High-density crowd with bottlenecks	Multiple corners with crossing patterns

4.1.2 Variables and Parameters

The experimental design incorporates independent variables (controlled by the experimenter), dependent variables (measured outcomes), and controlled variables (held constant across experimental conditions).

Independent Variables

We systematically manipulated the following independent variables across experimental trials:

- **Navigation Algorithm Configuration**
  - Social cost function parameters (personal space radius: 0.45m-1.2m)
  - Proxemic layer weighting (0.3-1.0 relative to obstacle costs)
  - Planning horizon duration (1.0s-3.5s for local planning)
  - Goal-directed vs. social compliance behavior weighting
- **Human Density and Distribution**
  - Number of humans (1-12)
  - Spatial distribution patterns (uniform, clustered, directional)
  - Group formation configurations (individuals, dyads, larger groups)
- **Human Behavior Models**
  - Attention levels (attentive, distracted, unaware)
  - Cooperation levels (cooperative, neutral, uncooperative)
  - Movement predictability (highly predictable to erratic)
- **Environmental Constraints**

- Corridor width (1.8m-3.5m)
- Corner angle (70°-110°)
- Obstacle density and placement

These variables were manipulated using Arena-Rosnav's parameter configuration system, which allows systematic parameter sweeps across predefined ranges. For educational experiments, a subset of these variables was exposed to students through simplified interfaces appropriate to their learning stage.

Dependent Variables

We measured several categories of dependent variables to evaluate both technical performance and social compliance:

- **Navigation Efficiency Metrics**
  - Path length ratio (actual/optimal)
  - Navigation time ratio (actual/optimal)
  - Smoothness of trajectory (jerk analysis)
  - Computational efficiency (planning cycle time)
- **Safety Metrics**
  - Minimum separation distance to humans
  - Collision count and near-miss frequency
  - Time spent in human personal/social spaces
  - Velocity modulation near humans
- **Social Compliance Metrics**
  - Adherence to social conventions (right/left passing, queuing)
  - Human trajectory disturbance (deviation from preferred paths)
  - Predictability of robot movement (human-reported comfort)
  - Social signal appropriateness (speed, direction changes)
- **Educational Metrics** (for student experiments)
  - Conceptual understanding assessment scores
  - Parameter tuning effectiveness
  - Algorithm selection appropriateness
  - Experimental design quality

These metrics were collected automatically through Arena-Rosnav's integrated data collection system, which records comprehensive trajectory, planning, and interaction data for each experimental trial.

Controlled Variables

To ensure experimental validity, we held the following variables constant across experimental conditions:

- **Robot Platform Configuration**
  - Physical dimensions (0.5m diameter circular footprint)
  - Sensor configuration (270° laser scanner, 4m range)
  - Maximum velocity constraints (0.5m/s linear, 1.2rad/s angular)
  - Actuator response characteristics (constant acceleration model)
- **Simulation Parameters**
  - Physics engine configuration (step size, solver iterations)
  - Sensor noise models (constant parameters)
  - Environmental conditions (lighting, floor texture)
  - Simulation time scaling (real-time execution)

These controlled variables ensure that observed differences in experimental outcomes can be attributed to the independent variables rather than uncontrolled factors in the simulation environment.

4.2 Data Collection Methods

Our data collection approach combines automated quantitative measurements, qualitative assessments, and educational evaluations to provide a comprehensive view of social navigation performance.

4.2.1 Automated Measurement System

We extended Arena-Rosnav's data collection capabilities with specialized components for social navigation metrics:

- **Trajectory Recording System** – Captures time-indexed position, velocity, and acceleration data for the robot and all human agents at 10Hz. This data forms the foundation for trajectory analysis, proximity calculations, and social compliance evaluation.
- **Planning Process Instrumentation** – Records internal planning data including considered paths, rejected trajectories, cost distribution maps, and computation time. This instrumentation provides insight into the navigation algorithm's decision-making process.
- **Proxemic Violation Detection** – Implements a zonal model of human personal space with concentric regions (intimate: 0-0.45m, personal: 0.45-1.2m, social: 1.2-3.6m) and continuously monitors for zone penetration events, recording duration and penetration depth.
- **Social Convention Monitoring** – Detects adherence to environment-specific social conventions including preferred passing side, yielding behavior, and queuing dynamics. This monitoring system uses geometric analysis of trajectories to identify convention adherence or violation.

- **Environmental Context Tracking** – Captures contextual information including location type (corridor, corner, open space), density conditions, and environmental constraints. This contextual data supports nuanced analysis of navigation behavior across different environmental conditions.

Data collection runs as a distributed system integrated with ROS, with dedicated nodes for different measurement aspects. All data is time-synchronized through the ROS clock and stored in both ROS bag format (for detailed analysis) and processed CSV files (for statistical analysis).

4.2.2 Qualitative Assessment

In addition to quantitative measurements, we conducted qualitative assessments of navigation behavior:

- **Expert Evaluation Protocol** – We developed a structured protocol for expert assessment of robot social navigation behavior. This protocol includes rating scales for naturalness, predictability, appropriateness, and overall social intelligence, applied to recorded navigation episodes.
- **Behavior Coding System** – A formal coding system for categorizing specific robot behaviors during human interactions, including avoidance initiation timing, trajectory communication clarity, and recovery behavior appropriateness.
- **Critical Incident Analysis** – Detailed examination of edge cases and failure modes, documenting the circumstances and contributing factors to navigation breakdowns or socially inappropriate behaviors.

Three domain experts with backgrounds in robotics, human-robot interaction, and social psychology independently evaluated a subset of experimental trials using these qualitative methods. Inter-rater reliability was assessed using Cohen's kappa coefficient, with a m inimum threshold of $K > 0.7$ required for inclusion in the analysis.

4.2.3 Educational Assessment

For the educational objectives of our research, we implemented a multi-faceted assessment approach:

- **Pre/Post Knowledge Assessments** – Validated instruments measuring student understanding of key social navigation concepts before and after using the simulation environment. These assessments include both technical knowledge (algorithm operation, parameter effects) and conceptual understanding (proxemics theory, social navigation principles).
- **Laboratory Task Performance** – Structured tasks requiring students to achieve specific navigation objectives by configuring and tuning social navigation algorithms. Task performance is measured through objective metrics (success rate, efficiency) and process measures (approach, testing strategy).
- **Design Challenge Evaluation** – Open-ended design challenges requiring students to develop novel solutions to specific social navigation problems. Evaluation criteria include technical correctness, creativity, effectiveness, and justification quality.

- **Reflective Analysis** – Guided reflection assignments where students analyze the behavior of their navigation solutions and connect observations to theoretical principles. These reflections are assessed for depth of analysis, connection to theory, and evidence-based reasoning.

Educational assessments were conducted with undergraduate robotics students (n=24) and graduate-level human-robot interaction students (n=17) using the simulation environment under controlled classroom conditions.

4.3 Evaluation Criteria

We established comprehensive evaluation criteria that address both technical performance and social dimensions of navigation, as well as educational effectiveness.

4.3.1 Technical Performance Criteria

Technical performance was evaluated against the following criteria:

- **Navigation Success Rate** – The percentage of trials where the robot successfully reaches its goal position without collisions or deadlocks. This primary measure of navigation competence is calculated across scenario categories with increasing difficulty thresholds for more complex environments.
- **Path Efficiency** – Measured as the ratio between actual path length and optimal path length (with no humans present). Efficiency thresholds were established for different environmental conditions, with the understanding that social navigation necessarily involves detours from the geometrically optimal path.
- **Computational Efficiency** – Planning cycle times must remain within real-time constraints (planning cycle < 100ms) to ensure responsive navigation. We also evaluated planning stability through analysis of planning cycle time variance.
- **Robustness to Uncertainty** – Performance degradation under sensor noise, human behavior unpredictability, and environmental variability. We systematically increased these uncertainty factors and measured the impact on navigation success and efficiency.

These criteria establish minimum thresholds for technical acceptability while recognizing the inherent trade-offs involved in social navigation. Rather than optimizing for a single technical metric, we seek balanced performance across the technical criteria while maintaining social compliance.

4.3.2 Social Compliance Criteria

Social compliance was evaluated against both general social navigation principles and scenario-specific social conventions:

- **Proxemic Compliance** – Adherence to established proxemic zones around humans, measured through zone penetration frequency, duration, and depth. We established graduated compliance thresholds based on environmental constraints, with more stringent requirements in unconstrained spaces.

- **Behavioral Legibility** – The predictability and understandability of robot motion from a human perspective. This is assessed through trajectory analysis (acceleration patterns, path consistency) and expert evaluation of motion naturalness.
- **Minimization of Human Disturbance** – The degree to which robot navigation avoids disrupting human trajectories or causing humans to change their preferred paths. This is measured through analysis of human trajectory deviations in the presence vs. absence of the robot.
- **Adherence to Social Conventions** – Compliance with contextual social norms including:
  - Corridor passing: Consistent side selection, appropriate speed modulation
  - Corner navigation: Cautious approach, clear intent signaling
  - Group navigation: Treating coherent groups as units, avoiding group separation

We recognize that social compliance criteria must be context-sensitive, with different expectations in different environmental and cultural contexts. Our evaluation framework accommodates these contextual factors through parameterized thresholds that can be adjusted to reflect specific social contexts.

4.3.3 Educational Effectiveness Criteria

Educational effectiveness was evaluated against learning objectives at multiple levels:

- **Knowledge Acquisition** – Students should demonstrate understanding of:
  - Proxemic theory and its application to robotics
  - Social cost function formulations and effects
  - Parameter-behavior relationships in social navigation
  - Evaluation methods for social navigation performance
- **Skill Development** – Students should develop abilities to:
  - Configure and tune social navigation algorithms for specific contexts
  - Diagnose and resolve navigation issues through systematic parameter adjustment
  - Implement and modify social cost functions
  - Design effective experiments to evaluate navigation performance
- **Conceptual Understanding** – Students should form mental models that:
  - Connect algorithm behavior to human social expectations
  - Recognize the inherent trade-offs in social navigation objectives
  - Anticipate how environmental factors influence appropriate social behavior
  - Apply social navigation principles across different robotic contexts

These educational criteria were assessed through a combination of direct measures (knowledge tests, performance tasks) and indirect measures (reflective assignments, experimental design quality).

4.4 Analysis Methods

Our analysis approach combines statistical methods, trajectory analysis techniques, and qualitative interpretation to derive meaningful insights from the experimental data.

4.4.1 Statistical Analysis Framework

We employed the following statistical methods to analyze experimental results:

- **Hypothesis Testing** – We formulated specific hypotheses about algorithm performance and social compliance, testing these hypotheses using appropriate statistical tests (t-tests for pairwise comparisons, ANOVA for multi-factor analysis). Significance threshold was set at $p < 0.05$ with Bonferroni correction for multiple comparisons.

- **Parameter Sensitivity Analysis** – To understand the relationship between algorithm parameters and navigation outcomes, we conducted sensitivity analysis using response surface methodology. This approach maps the parameter space to outcome metrics, identifying critical parameters and optimal operating regions.

- **Multi-objective Performance Analysis** – Given the inherent trade-offs between efficiency, safety, and social compliance, we employed multi-objective analysis techniques including Pareto frontier identification to characterize the performance envelope of different navigation approaches.

- **Learning Analytics** – For educational assessment, we applied learning analytics techniques including knowledge gain analysis (normalized pre/post differences), skill development trajectories, and correlation analysis between different assessment dimensions.

Statistical analysis was performed using R (v4.2.0) with specialized packages for robotics data analysis and visualization (TrajR, HRIstats).

4.4.2 Trajectory Analysis

We developed specialized trajectory analysis methods to extract meaningful social navigation metrics:

- **Social Interaction Identification** – Automated detection of interaction episodes within trajectory data, identifying when the robot and humans are engaged in navigation coordination. This detection uses proximity thresholds, relative velocity, and heading alignment to segment continuous trajectories into discrete interaction events.

- **Navigation Strategy Classification** – Machine learning based classification of robot navigation strategies (e.g., proactive avoidance, reactive adjustment, human-following) from trajectory features. This classification was trained on expert-labeled trajectory segments and achieves 87% agreement with human coders.

- **Quality of Interaction Metrics** – Derived measures that combine multiple trajectory features to assess interaction quality, including fluency (smoothness of interleaved trajectories), coordination (mutual adaptation patterns), and minimal jerk criteria.

Trajectory analysis was implemented through custom Python libraries built on NumPy, SciPy, and specialized robotics analysis tools, integrated with ROS through dedicated analysis nodes.

4.4.3 Qualitative Analysis

For qualitative data from expert evaluations and student reflections, we implemented a structured analysis approach:

- **Thematic Analysis** – Identification of recurring themes in qualitative data using an initial codebook developed from theoretical principles, expanded through inductive coding. Code application was performed by multiple coders with regular inter-coder reliability assessment.
- **Critical Instance Analysis** – Detailed examination of particularly successful or problematic navigation episodes, combining trajectory data, algorithm state information, and expert assessment to understand the factors contributing to the outcome.
- **Comparative Case Analysis** – Structured comparison of navigation behavior across different algorithms, environmental conditions, and human behavior models to identify patterns and principles that generalize across contexts.

Qualitative analysis was conducted using MAXQDA software with a team-based coding approach that combines robotics expertise and human-robot interaction knowledge.

4.5 Methodological Limitations

We acknowledge several limitations in our methodology that must be considered when interpreting results:

- **Simulation Fidelity Limitations** – While Arena-Rosnav provides high-fidelity simulation, it cannot perfectly reproduce the complexity of real-world human behavior, particularly the subtle social signals and cultural variations present in human navigation. Our results should be interpreted with this limitation in mind.
- **Human Model Simplifications** – The human agent models, though based on empirical human movement data, necessarily simplify human decision-making and responsiveness. In particular, the models may not fully capture the adaptive and strategic aspects of human navigation behavior in response to robots.
- **Scenario Coverage** – While we have designed scenarios to represent common navigation contexts, they cannot exhaustively cover the full range of social navigation challenges. Results may not generalize to substantially different environments or interaction contexts.
- **Educational Population Limitations** – The educational assessment was conducted with computer science and robotics students who may have different prior knowledge and learning approaches than other potential user groups for social navigation education.

We have attempted to mitigate these limitations through careful experimental design, validation of simulation components against real-world data where possible, and appropriate qualification of conclusions based on methodological constraints.

4.6 Summary

This chapter has outlined a comprehensive methodological framework for evaluating social navigation algorithms both technically and socially while assessing their educational value. Our approach extends the capabilities of the Arena-Rosnav simulation environment with specialized data collection, analysis, and evaluation methods tailored to social navigation research.

The experimental design provides systematic coverage of navigation scenarios with controlled progression of complexity, while the data collection system captures both quantitative metrics and qualitative aspects of navigation performance. Our evaluation criteria balance technical performance with social compliance considerations, recognizing that effective social navigation requires optimization across multiple, sometimes competing objectives.

The analysis methods combine statistical rigor, specialized trajectory analysis, and structured qualitative approaches to derive meaningful insights from complex navigation data. While acknowledging methodological limitations, this approach provides a solid foundation for the experimental results presented in subsequent chapters.

This methodology supports both the scientific objectives of advancing social navigation algorithms and the educational goals of developing effective teaching approaches for social navigation concepts. By integrating these dual purposes into a unified methodological framework, we contribute not only to the development of better navigation algorithms but also to the training of future roboticists who will implement these algorithms in real-world systems.

Chapter 5

Experiments and Results

5.1 Experiment Execution

5.1.1 Corridor Scenario Results

5.1.2 Corner Scenerio Results

5.2 Analysis of Results

5.2.1 Technical Evaluation

5.2.2 Educational Outcomes

5.2.3 Social Navigation Assessment

5.3 Discussion of Findings

Chapter 6

Discussion

6.1 Implications for Social Navigation Research

6.2 Implications for Robotics Education

6.3 Limitations of the Study

6.4 Recommendations for Future Work

Chapter 7

Conclusion

7.1 Summary of Contributions

7.2 Key Findings

7.3 Future Directions

Appendix A

Installation Instructions

This appendix provides instructions for setting up the development environment used in the experiments.

A.1 Virtual Machine Setup

1. Download and install **VirtualBox** from the official website:

   https://www.virtualbox.org/wiki/Downloads

2. Import the provided virtual machine image (`dev.ova`) into VirtualBox. A step-by-step guide for importing `.ova` files can be found here:

   https://chenweixiang.github.io/docs/How_to_Import_and_Export_OVA_Files_in_VirtualBox.pdf

Once the virtual machine is imported, you can launch the pre-configured development environment to reproduce experiments, build the workspace, and run simulations.

A.2 Virtual machine Launch

Once the virtual machine is imported successfully, go ahead and click the **Start** button (See Fig:A.1). This will launch a new virtual box window where the system is running Ubuntu 20.04. The login the details are as follows:

- *User name*: dev
- *Password*: thws

Arena-rosnav and Visual Studio Code are fully installed in this virtaul machine and are can be used now.

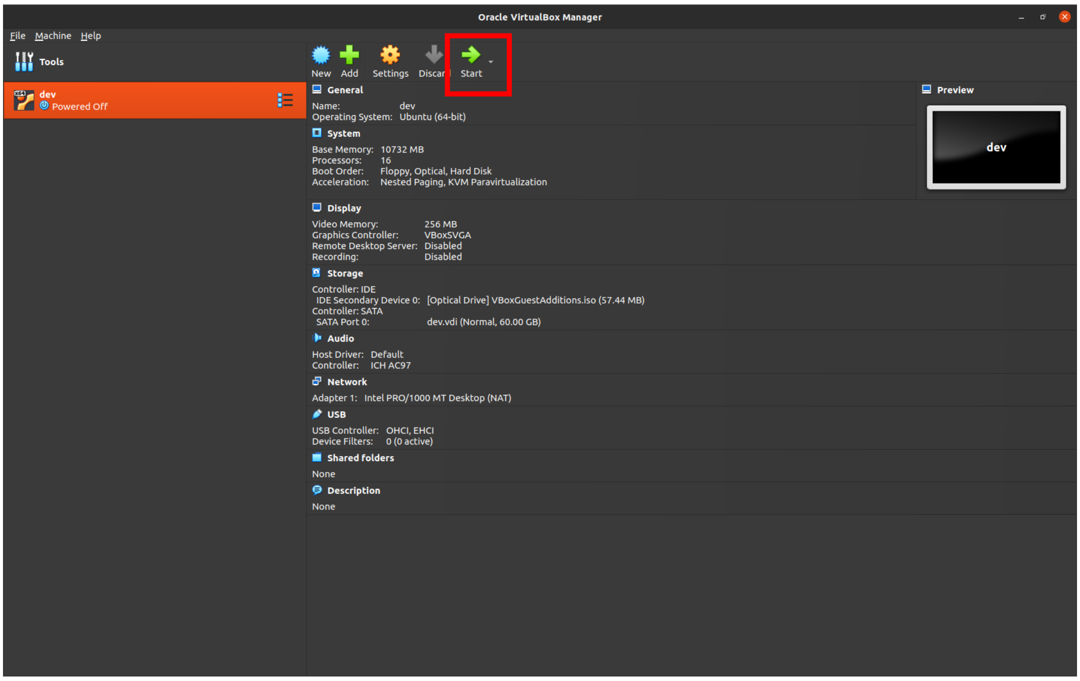

FIGURE A.1: Virtual Box main window

Appendix B

Implementation Details of Social Layer Integration in Arena-Rosnav

This appendix documents the step-by-step procedure for integrating the social navigation layer into the costmap using the Arena-Rosnav simulation framework. The configuration enables socially-aware robot navigation by leveraging proxemic behavior modeling.

B.1 Repository Setup

Step 1: Clone the `people` repository

Clone the `people` repository into the utilities folder:

```
cd ~/arena_ws/src/arena/utils
git clone -b noetic https://github.com/DLu/people.git
```

This repository provides the `people_msgs` message type which must be published on the `/people` topic. For documentation, see: https://docs.ros.org/en/api/people_msgs/html/msg/People.html.

Step 2: Clone the `navigation_layers` repository

Clone the `navigation_layers` repository:

```
cd ~/arena_ws/src/arena/utils/navigation
git clone -b noetic https://github.com/DLu/navigation_layers.git
```

This repository provides the `social_navigation_layers` package, which includes the Proxemic and Passing layers. See documentation: http://wiki.ros.org/social_navigation_layers.

B.2 Publishing `people_msgs` from Pedsim

Step 3: Modify the `pedsim_simulator`

Find the simulator:

```
rospack find pedsim_simulator
```

Modify the following files to publish `people_msgs`:

- `src/simulator.cpp`
- Corresponding header file

Ensure the message is published on the `/people` topic.

B.3 Rebuild Workspace

Step 4: Rebuild with `catkin build`

```
cd ~/arena_ws
catkin build
```

B.4 Costmap Configuration

Step 5: Add Social Layer Plugins

Edit the costmap parameters file for the robot (e.g., Jackal):
`/home/dev/arena_ws/src/arena/simulation-setup/entities/robots/jackal/configs/costmaps/global_costmap_params.yaml`

Add the following plugin:

```
- { name: proximic_layer, type:
"social_navigation_layers::ProxemicLayer" }
```

Parameters for each layer are configured in: `costmap_common_params.yaml`

B.5 Launching and Runtime Adjustment

Step 6: Launch the Simulation

First, source the workspace and activate the virtual environment if needed:

```
source ~/arena_ws/devel/setup.bash
roslaunch arena_bringup start_arena.launch
```

Step 7: Use `rqt_reconfigure` for Dynamic Parameters

Open a new terminal and run:

```
rosrun rqt_reconfigure rqt_reconfigure
```

This allows for real-time adjustment of social navigation parameters.

Appendix C

Evaluation Pipeline for Arena-Rosnav

This appendix outlines the procedure for evaluating simulation results using the tools provided in the Arena-Rosnav framework.

C.1 Environment Setup

1. Open the Arena workspace in VS Code:

```
cd /home/dev/arena_ws
code .
```

2. Open a new terminal in VS Code: `Ctrl + Shift + ``

3. Activate the poetry environment:

```
cd ~/arena_ws/src/arena/arena-rosnav
poetry shell
cd ~/arena_ws/src/arena/evaluation/arena_evaluation/
scripts
```

C.2 Generating Evaluation Metrics

1. Suppose your recorded data is located at:

```
/home/dev/arena_ws/src/arena/evaluation/arena_evaluation
/data/25-05-03_11-49-11/jackal
```

   This directory should contain the following files:

   - `episode.csv`
   - `odom.csv`
   - `params.yaml`
   - `pedsim_agents_data.csv`
   - `scan.csv`
   - `start_goal.csv`

2. Run the following command to compute evaluation metrics (replace path as necessary):

```
python get_metrics /home/dev/arena_ws/src/arena/
evaluation/arena_evaluation/data/25-05-03_11-49-11/
jackal --pedsim
```

3. A new file `metrics.csv` will be generated in the data directory.

C.3 Plotting Evaluation Results

1. Navigate to the `plot_declarations` directory and create a YAML configuration file:

```
cd ~/arena_ws/src/arena/evaluation/arena_evaluation
/plot_declarations
touch eval_teb_2025.yaml
```

2. Copy contents from `sample_schema.yaml` into your new file:

```
cp sample_schema.yaml eval_teb_2025.yaml
```

3. Edit the YAML file to match your evaluation data and desired plots.

4. Generate plots using the following command (adjust path as necessary):

```
python create_plots /home/dev/arena_ws/src/arena
/evaluation/arena_evaluation/plot_declarations/
eval_teb_2025.yaml
```

Bibliography

Kästner, Linh et al. (2021). *Arena-Rosnav: Towards Deployment of Deep-Reinforcement-Learning-Based Obstacle Avoidance into Conventional Autonomous Navigation Systems*. arXiv: 2104.03616 [[cs.RO](http://cs.ro/)]. URL: https://arxiv.org/abs/2104.03616.

Pérez-Higueras, Noé et al. (2023). *HuNavSim: A ROS 2 Human Navigation Simulator for Benchmarking Human-Aware Robot Navigation*. arXiv: 2305.01303 [[cs.RO](http://cs.ro/)]. URL: https://arxiv.org/abs/2305.01303.

Savva, Manolis et al. (2019). "Habitat: A Platform for Embodied AI Research". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346. DOI: 10.1109/ICCV.2019.00943.

Rios-Martinez, J., A. Spalanzani, and C. Laugier (2015). "From proxemics theory to socially-aware navigation: A survey". In: *International Journal of Social Robotics* 7.2, pp. 137–153. DOI: 10.1007/s12369-014-0251-1. URL: https://doi.org/10.1007/s12369-014-0251-1.

Kruse, Thibault et al. (2013). "Human-aware robot navigation: A survey". In: *Robotics and Autonomous Systems* 61.12, pp. 1726–1743. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2013.05.007. URL: https://www.sciencedirect.com/science/article/pii/S0921889013001048.

Möller, Ronja et al. (2021). "A survey on human-aware robot navigation". In: *Robotics and Autonomous Systems* 145, p. 103837. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2021.103837. URL: https://www.sciencedirect.com/science/article/pii/S0921889021001226.

Hall, Edward T. et al. (1968). "Proxemics [and Comments and Replies]". In: *Current Anthropology* 9.2/3, pp. 83–108. DOI: 10.1086/200975. eprint: https://doi.org/10.1086/200975. URL: https://doi.org/10.1086/200975.

Henkel, Zachary et al. (June 2014). "Evaluation of Proxemic Scaling Functions for Social Robotics". In: *Human-Machine Systems, IEEE Transactions on* 44, pp. 374–385. DOI: 10.1109/THMS.2014.2304075.

Helbing, Dirk and Péter Molnár (May 1995). "Social force model for pedestrian dynamics". In: *Physical Review E* 51.5, 4282–4286. ISSN: 1095-3787. DOI: 10.1103/physreve.51.4282. URL: http://dx.doi.org/10.1103/PhysRevE.51.4282.

Kastner, Linh et al. (Oct. 2022). "Arena-Bench: A Benchmarking Suite for Obstacle Avoidance Approaches in Highly Dynamic Environments". In: *IEEE Robotics and Automation Letters* 7.4, 9477–9484. ISSN: 2377-3774. DOI: 10.1109/lra.2022.3190086. URL: http://dx.doi.org/10.1109/LRA.2022.3190086.

Sacco, Federico, Carmine Recchiuto, and Jonas Mårtensson (2024). "A Novel Social Navigation Approach Based on Model Predictive Control and Social Force Model". In: *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, pp. 1705–1711. DOI: 10.1109/RO-MAN60168.2024.10731256.

Chen, Yu Fan et al. (2018). *Socially Aware Motion Planning with Deep Reinforcement Learning*. arXiv: 1703.08862 [[cs.RO](http://cs.ro/)]. URL: https://arxiv.org/abs/1703.08862.

Szot, Andrew et al. (2022). *Habitat 2.0: Training Home Assistants to Rearrange their Habitat*. arXiv: 2106.14405 [cs.LG]. URL: https://arxiv.org/abs/2106.14405.

Tsoi, Nathan et al. (2022). "SEAN 2.0: Formalizing and Generating Social Situations for Robot Navigation". In: *IEEE Robotics and Automation Letters*, pp. 1–8. DOI: 10.1109/LRA.2022.3196783.

Rösmann, Christoph, Frank Hoffmann, and Torsten Bertram (2015). "Timed-Elastic-Bands for time-optimal point-to-point nonlinear model predictive control". In: *2015 European Control Conference (ECC)*, pp. 3352–3357. DOI: 10.1109/ECC.2015.7331052.

Xiao, Xuesu et al. (2020). *APPLD: Adaptive Planner Parameter Learning from Demonstration*. arXiv: 2004.00116 [cs.RO]. URL: https://arxiv.org/abs/2004.00116.

Khatib, O. (1985). "Real-time obstacle avoidance for manipulators and mobile robots". In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. Vol. 2, pp. 500–505. DOI: 10.1109/ROBOT.1985.1087247.

Xiao, Xuesu et al. (2022b). *Motion Planning and Control for Mobile Robot Navigation Using Machine Learning: a Survey*. arXiv: 2011.13112 [cs.RO]. URL: https://arxiv.org/abs/2011.13112.

Long, Pinxin et al. (2018). *Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning*. arXiv: 1709.10082 [cs.RO]. URL: https://arxiv.org/abs/1709.10082.

Rösmann, Christoph (2019). "Time-optimal nonlinear model predictive control". In:

Xiao, Xuesu et al. (Aug. 2022a). *Autonomous ground navigation in highly constrained spaces: Lessons learned from the benchmark autonomous robot navigation challenge at icra 2022 [competitions]*. DOI: 10.48550/arXiv.2208.10473.

Chen, Changan et al. (2019). *Crowd-Robot Interaction: Crowd-aware Robot Navigation with Attention-based Deep Reinforcement Learning*. arXiv: 1809.08835 [cs.RO]. URL: https://arxiv.org/abs/1809.08835.

Teja Singamaneni, Phani, Anthony Favier, and Rachid Alami (2021). "Human-Aware Navigation Planner for Diverse Human-Robot Interaction Contexts". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5817–5824. DOI: 10.1109/IROS51168.2021.9636613.

Li, Keyu et al. (2019). "SARL: Deep Reinforcement Learning based Human-Aware Navigation for Mobile Robot in Indoor Environments". In: *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 688–694. DOI: 10.1109/ROBIO49542.2019.8961764.

Kästner, Linh, Cornelius Marx, and Jens Lambrecht (2020). *Deep-Reinforcement-Learning-Based Semantic Navigation of Mobile Robots in Dynamic Environments*. arXiv: 2008.00516 [cs.RO]. URL: https://arxiv.org/abs/2008.00516.

Everett, Michael, Yu Fan Chen, and Jonathan P. How (2018). *Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning*. arXiv: 1805.01956 [cs.RO]. URL: https://arxiv.org/abs/1805.01956.

Dugas, Daniel et al. (2020). *NavRep: Unsupervised Representations for Reinforcement Learning of Robot Navigation in Dynamic Human Environments*. arXiv: 2012.04406 [cs.RO]. URL: https://arxiv.org/abs/2012.04406.

Kästner, Linh et al. (2024). *Arena 3.0: Advancing Social Navigation in Collaborative and Highly Dynamic Environments*. arXiv: 2406.00837 [cs.RO]. URL: https://arxiv.org/abs/2406.00837.