WARSAW UNIVERSITY OF TECHNOLOGY

DISCIPLINE OF SCIENCE AUTOMATIC CONTROL, ELECTRONICS,
ELECTRICAL ENGINEERING AND SPACE TECHNOLOGIES/
FIELD OF SCIENCE ENGINEERING AND TECHNOLOGY

Ph.D. Thesis

Jarosław Karwowski, M.Sc.

Trajectory planning for a social robot using a hybrid trajectory candidates generation and spatiotemporal cost functions

Supervisor
Dr. hab. Eng. Wojciech Szynkiewicz, university professor

WARSAW 2024

Acknowledgements

I would like to extend my gratitude to my supervisor, Prof. Wojciech Szynkiewicz, for his valuable scientific and organisational support, as well as for preventing my discouragement throughout the journey in the PhD program.

I wish to acknowledge the contributions of my lovely wife, Marlena, who never let me give up and always willingly assisted in laboratory studies. I would also like to extend deep appreciation to my parents as well as my close friends for their continued support during challenging times.

Additionally, I would like to thank my previous supervisor, Dr. Witold Czajewski, and the lecturers at Warsaw University of Technology who inspired me to delve into robotics.

Abstract

Autonomous robot navigation is a key capability supporting various mobility-dependent tasks in robotics. As robots of diverse types increasingly operate in public spaces, developing a comprehensive social robot navigation system is one of the essential research tasks. However, this requires a prior understanding of the principles of social acceptance, encompassing factors that may cause human discomfort and rules for robot navigation in populated areas. Approaches existing in the literature often overlook many aspects of social navigation, which presents a multifaceted challenge. Furthermore, systems benchmarking navigation techniques lack essential mathematical indicators for assessing human discomfort levels, highlighting the need for further research in this field.

This thesis aims to develop a trajectory planning algorithm that enhances the navigation quality of robots operating in environments shared with people by reducing human discomfort resulting from robot movement while maintaining the navigation performance of traditional methods.

The dissertation contributes to several key aspects of social robot navigation. Firstly, the work defines the requirements for social robot navigation. A literature review was conducted to gather insights for grounding social robot navigation requirements, which should be implemented in comprehensive navigation systems. The requirements taxonomy distinguishes the following groups of necessities: requirements regarding humans' physical and perceived safety, requirements for assessing robot motion naturalness, and compliance with social conventions.

Secondly, the thesis addresses the challenge of quantitatively assessing social robot navigation. While various metrics for evaluating traditional robot navigation concepts have been implemented in different benchmark systems, indicators that assess human-aware robot navigation are lacking. Therefore, additional metrics were developed to evaluate the compliance of navigation algorithms with the grounded requirements. The substantially original aspect is that the social awareness indicators account for human tracking uncertainty, facilitating the evaluation using robot onboard perception. The novel metrics were introduced and integrated into the new benchmarking system, which can be used to test robots operating in simulated and real-world environments.

Thirdly, the dissertation presents a novel human-aware local trajectory planner that employs the hybrid trajectory candidates generation method and spatiotemporal cost functions. The algorithm developed to enhance social robot navigation is a geometric planner that addresses the issue of receding horizon trajectory planning for dynamic systems operating in unstructured environments. The proposed method is suitable for differential drive and holonomic mobile robots. The hybrid approach for producing various trajectory candidates employs two generation methods for online planning. The first strategy is based on a pedestrian motion model, whereas the second employs a technique of sampling feasible velocity control commands. The novel aspect of the first method lies in extending a pedestrian motion model to obtain emphasised collision avoidance behaviours and improved motion legibility compared to the baseline social force model-based formulation. Numerous admissible trajectory candidates are produced by exploiting the parameterisation of the deterministic motion model. The objective function used for assessing the quality of aggregated trajectories considers collision avoidance and soft constraints related to social acceptance encompassing robot motion naturalness, and human physical and perceived safety measures implemented as spatiotemporal cost functions. The planner operates based on several behaviours that implement various strategies, enabling compliance with social norms and enhancing reliability using environmental context information.

A multitude of experiments has been conducted to assess the performance and social appropriateness of the proposed trajectory planning method against various traditional and specialised methods for social robot navigation, including learning-based approaches. The evaluation criteria included a range of metrics verifying the compatibility of the algorithms with the requirements for social robot navigation. A controlled study-based, multi-scenario comparison implemented standardised protocols for evaluating robot navigation in human-populated environments. Analogous scenarios have been performed in both the real-world laboratory environment and its virtual equivalent to compare the outcomes obtained from simulations with those observed in the real world.

The experiments demonstrated improved navigation quality of the proposed local trajectory planner. According to standardised metrics derived from social robot navigation requirements, the developed algorithm outperforms state-of-the-art methods in reducing human discomfort but also ensures reliable and efficient navigation task execution across various dynamic scenarios.

Keywords: social robotics, social robot navigation, trajectory planning, quantitative evaluation

Streszczenie

Autonomiczna nawigacja robotów stanowi ich podstawową umiejętność, która jest wymagana do realizacji złożonych zadań wymagających mobilności. Roboty różnych typów coraz częściej pojawiają się w przestrzeni publicznej, zatem opracowanie kompleksowego systemu nawigacji robotów społecznych jest jednym z istotnych zadań badawczych. Wymaga to jednak wcześniejszego sformułowania kryteriów społecznej akceptacji, obejmujących zasady nawigacji robotów w otoczeniu ludzi oraz czynniki mogące powodować ich dyskomfort. Proponowane w literaturze algorytmy często pomijają wiele zagadnień nawigacji społecznej stanowiącej wieloaspektowy problem. Co więcej, narzędzia wykorzystywane do porównywania technik nawigacji nie zawierają ugruntowanych i powszechnie akceptowanych wskaźników, co podkreśla potrzebę dalszych badań w tej dziedzinie.

Podstawowym celem postawionym w niniejszej rozprawie było opracowanie algorytmu planowania trajektorii, który poprawi jakość nawigacji robotów działających w środowiskach współdzielonych z ludźmi poprzez zmniejszenie u nich dyskomfortu wywołanego przez ruch robota przy jednoczesnym zachowaniu wydajności nawigacji tradycyjnych metod planowania trajektorii.

Rozprawa wnosi autorski wkład w kilka kluczowych aspektów nawigacji robotów społecznych. Po pierwsze, definiuje wymagania wobec nawigacji robotów społecznych. Przeprowadzono przegląd literatury w celu zebrania informacji o wymaganiach dotyczących nawigacji robotów społecznych, które powinny zostać uwzględnione w kompleksowych systemach sterowania robotów społecznych. Opracowana taksonomia wymagań wyróżnia następujące grupy: wymagania dotyczące zapewnienia fizycznego oraz postrzeganego bezpieczeństwa ludzi, wymagania dotyczące oceny naturalności ruchu robota oraz zgodność z normami społecznymi.

Po drugie, praca koncentruje się na kryteriach ilościowej oceny nawigacji robotów społecznych. Różne wskaźniki do ewaluacji zagadnień związanych z tradycyjną nawigacją robotów zostały wdrożone w systemach wzorcowych (ang. benchmark), natomiast wskaźniki jakości do oceny społecznej nawigacji robotów są rzadkością. W związku z tym opracowano dodatkowe wskaźniki w celu oceny zgodności algorytmów nawigacji z wymaganiami. Oryginalnym aspektem proponowanych wskaźników do oceny dyskomfortu ludzi jest uwzględnienie niepewności śledzenia człowieka, co umożliwia efektywną jego ocenę przy wykorzystaniu modułów percepcji robota. Nowe wskaźniki zostały zintegrowane z opracowanym systemem wzorcowej oceny, przeznaczonym do ewaluacji jakości nawigacji robotów działających w rzeczywistych i symulowanych środowiskach.

Po trzecie, rozprawa przedstawia nowe podejście do lokalnego planowania trajektorii uwzględniające obecność człowieka, które wykorzystuje hybrydową metodę generowania kandydatów trajektorii i przestrzenno-czasowe funkcje kosztu. Algorytm opraco-

wany w celu zwiększenia społecznej akceptacji poruszających się robotów jest planistą geometrycznym, który rozwiązuje problem planowania trajektorii z przesuwanym horyzontem dla systemów dynamicznych działających w nieustrukturyzowanym środowisku. Proponowana metoda jest odpowiednia dla robotów mobilnych o napędach różnicowych i holonomicznych. Hybrydowe podejście do generowania różnych kandydatów trajektorii wykorzystuje dwie strategie. Pierwsza opiera się na modelu ruchu pieszych, podczas gdy druga wykorzystuje technikę próbkowania dopuszczalnych składowych wektora prędkości stanowiącego typowe polecenie sterujące baz jezdnych. Oryginalnym aspektem pierwszej strategii jest rozszerzenie modelu ruchu pieszych w celu uzyskania realistycznych zachowań unikania kolizji oraz zwiększonej czytelności ruchu w porównaniu z podstawową postacią opartą na modelu siły społecznej. Wykorzystując parametryzację deterministycznego modelu ruchu, tworzone są liczne dopuszczalne trajektorie.

Funkcja celu wykorzystywana do oceny jakości wygenerowanych trajektorii uwzględnia unikanie kolizji i miękkie ograniczenia związane z akceptacją społeczną obejmującą naturalność ruchu robota oraz fizyczne i postrzegane bezpieczeństwo ludzi wyrażone jako przestrzenno-czasowe funkcje kosztu. Schemat działania planisty oparty jest na kilku zachowaniach realizujących zróżnicowane strategie umożliwiające przestrzeganie norm społecznych oraz zwiększenie niezawodności poprzez wykorzystanie informacji o kontekście środowiskowym.

Przeprowadzono wiele eksperymentów w celu oceny wydajności i społecznej akceptowalności proponowanej metody planowania trajektorii w porównaniu z tradycyjnymi metodami nawigacji robotów i dedykowanymi do działania wśród ludzi, w tym podejściami opartymi na uczeniu ze wzmocnieniem. Kryteria ewaluacji obejmowały szereg wskaźników weryfikujących zgodność algorytmów z wymaganiami nawigacji robotów społecznych. W kontrolowanym badaniu zweryfikowano działanie różnych algorytmów w wielu scenariuszach zaprojektowanych na podstawie standardowych wytycznych wobec oceny społecznej nawigacji robotów. Analogiczne scenariusze zostały przeprowadzone zarówno w rzeczywistym środowisku laboratoryjnym, jak i jego wirtualnym odpowiedniku, aby porównać wyniki uzyskane w symulacji i w świecie rzeczywistym.

Badania wykazały, że proponowane rozwiązanie problemu lokalnego planowania trajektorii poprawia jakość nawigacji robotów pracujących w środowiskach współdzielonych z ludźmi. Zgodnie ze wskaźnikami wynikającymi z wymagań społecznej nawigacji robotów, opracowany algorytm przewyższa dotychczasowe metody w zmniejszeniu dyskomfortu u ludzi pod wpływem ruchu robota, jednocześnie zapewniając niezawodne i wydajne wykonywanie zadań nawigacyjnych w różnych dynamicznych scenariuszach.

Słowa kluczowe: robotyka społeczna, społeczna nawigacja robotów, planowanie trajektorii, ewaluacja ilościowa

Contents

Chapter 1

Introduction

Social robot navigation is a substantial branch of the mobile robotics field, as it fundamentally alters the dynamics of human-robot interactions. With the growing popularity and application areas of service and assistive robots, the interaction between humans and robots has become a vast field of study. People performing tasks in populated environments typically behave in a manner that avoids disrupting other humans' motions while trying to accomplish their tasks as effectively as possible [1]. The same guidelines apply to robots that assist workers in, e.g., restaurants and hospitals. Such activities are referred to as unfocused interactions [2], tackled by social robotics at the motion planning level with human-aware constraints.

1.1 Motivation

Autonomous robot navigation is a fundamental capability, upon which other complex tasks requiring robot mobility depend. Classical robot navigation entails environment sensing, map building, localisation, planning, and motion execution, focusing on objectives such as avoiding collisions with obstacles and reaching destinations quickly [3].

However, the social robot navigation concept introduces additional considerations to the classical formulation, regarding humans as special types of objects in the environment, with which interactions must be handled appropriately. As an interdisciplinary field, social robotics adapts expertise from different disciplines such as artificial intelligence, psychology, and natural language processing. This collaboration produces a wide range of results and corresponds to the complexity of human-robot interaction.

Over the years, the range of robots interacting with humans has been employed in diverse ways. In the late 2000s, Satake et al. [4] established a field study in a shopping mall where a robot recommended shops to people. A long-term validation of a robot operating in a crowded cafeteria was conducted by Trautman et al. [5]. Another extended

(a) Pudu Bellabot (b) Pudu Puductor (c) Pudu Swiftbot

deployment was accomplished by Biswas and Veloso [6], whose *CoBots* reached 1000 km of autonomous navigation. In contrast, Shiomi et al. [7] performed a short-term validation study of a robot operation in a shopping mall. Moreover, multiple challenges for academic robotic teams are organised, e.g., “Take The Elevator“ [8] or “Socially Acceptable Item Delivery”.[1] Recently, the popularity of robots for restaurant services (Fig. 1.1a)[2], hotel and hospital disinfection (Fig. 1.1b)[3], or transportation tasks (Fig. 1.1c)[4] is rapidly growing, as they have become commercially accessible. Other applications involve robots for home assistance and healthcare [9] or various delivery tasks, e.g., mail or packages [10].

Given the emergence of various types of robots performing diverse tasks in public spaces, it is justified to develop a comprehensive social robot navigation system. However, it is first necessary to identify the principles of social acceptance. These encompass factors that may cause discomfort to humans and rules of how robots should navigate in populated areas. User studies that precisely indicate how socially navigating robots should move, based on impressions of human participants, are substantial but often conducted in a fragmented manner. Therefore, one of the topics of this dissertation was to identify the **requirements** and classify them into a standard taxonomy.

Generally, the guidelines to be followed by socially navigating robots include ensuring the physical and psychological safety of humans, mimicking natural human movement, and adhering to social norms. Importantly, human-aware robot navigation also inherits the requirements of classical robot navigation, making socially navigating robots execute a task with (locally) conflicting goals. Designing and implementing a system that adheres to all these principles simultaneously enables seamless navigation and social acceptance of robots operating in populated environments. However, it poses significant challenges and existing social robot navigation approaches presented in the literature do not investigate

[1] `https://eu-robotics.net/2023-09-erl-mk-smart-city-challenge/` (accessed 23/04/2024)
[2] Figure source: `https://www.youtube.com/watch?v=DGajUN1icAs` (accessed 23/04/2024)
[3] Figure source: `https://www.youtube.com/watch?v=sQHeF4pP8yk` (accessed 23/04/2024)
[4] Figure source: `https://www.youtube.com/watch?v=gtUKUOzpZFc` (accessed 23/04/2024)

even half of the identified requirements in their objectives, overlooking many aspects of the multi-faceted problem of social navigation. Taking this into consideration, an attempt has been made to develop a new social robot **motion planning** framework with a priority on the mitigation of human discomfort by incorporating numerous requirements in its objective. Another focus lies in providing the high robustness of the approach for its suitability for real-world applications.

Furthermore, navigation methods are typically compared with each other either qualitatively or quantitatively. Qualitative assessment often involves visually comparing trajectories executed by a robot operating with different motion planning algorithms. Instead, quantitative evaluation of navigation methods is essential for objectively assessing their performance with invariant evaluation formulas. However, benchmarking systems for social robot navigation lack various mathematical indicators that allow for assessing the level of human discomfort, i.e., the degree of requirements fulfilment. Consequently, another issue addressed in this thesis is the **quantitative evaluation** of social robot navigation. The system developed for this purpose proved to be useful in the analysis of experiments' results.

1.2 Research problem and objectives

The breadth of research on the problem of social robot navigation is significant, as evidenced by the numerous literature reviews in the field, each discussing different aspects of the topic. For example, Gao and Huang [10] examined the evaluation techniques employed in prior studies, Francis et al. [11] proposed various guidelines for conducting social navigation studies, while Kruse et al. [12] attempted to identify the key features of human-aware navigation enhancing human comfort. Moreover, Charalampous et al. [13] and Möller et al. [9] reviewed the state-of-the-art focusing on perception aspects, whereas Chik et al. [14] highlighted different motion planning system structures. Rios-Martinez et al. [2] delved into sociological concepts regarding the challenges of human-aware navigation. Furthermore, Medina Sánchez et al. [15] verified modern algorithms for environment feature extraction, human trajectory prediction, and planning, while Guillén-Ruiz et al. [16] classified socially-aware navigation methods according to the techniques implemented in robots to handle interaction or cooperation with humans.

Topic relevance The recent survey [17] discusses relevant research directions in human-aware robot navigation. Referring to their paper, this thesis raises and contributes to two out of three core challenges constraining the seamless deployment of socially navigating autonomous robots in crowded environments, specifically **planning** and **evaluation**.

Planning is a broad topic, involving, i.a., strategic decision-making and symbolic planning for robot tasks. This thesis focuses on the geometric planning of robot trajectories. The second fundamental challenge in the field is the evaluation of social robot navigation systems. Various metrics for assessing human awareness and robot navigation performance are proposed to address this issue. The metrics calculation has been implemented in the benchmarking system developed as a part of this thesis.

Social robot navigation definition In our work, a **socially navigating robot** (or "robot navigating in a human-aware manner") is an autonomous machine designed to act and interact with humans in shared environments, mitigating potential human discomfort by mimicking social behaviours and adhering to norms. Robot navigation requirements are derived from user studies illustrating human preferences during an interaction, while robot decision-making autonomy relies on perception and planning capabilities.

In this work, the "social robot navigation" phrase is used interchangeably with "human-aware navigation"; however, an attempt to distinguish the meaning between those has been recently proposed in [18].

Objectives The aim of this thesis is the development of an algorithm that increases the efficacy of robots navigating in environments shared with humans. This is achieved by the mitigation of human discomfort induced by robot movement while preserving the navigation performance of traditional methods.

An extensive literature review has been performed to extract requirements for socially navigating robots reflecting the factors that cause human discomfort. Then, the quantitative criteria for assessing social navigation algorithms were designed and implemented in the benchmarking system that is suitable for evaluating aspects of both robot navigation performance and social awareness. Subsequently, a new local trajectory planning method has been developed, which considers the efficiency of navigation tasks, the physical and perceived safety of surrounding humans, and implements the adherence to social norms. Finally, a social robot control system has been integrated with the proposed planning algorithm and the benchmark to conduct comparative experiments against the state-of-the-art motion planning methods. The system has been extensively tested both in simulation and in real-world experiments.

1.3 Thesis statement

In this dissertation, theses are formulated as follows:

Thesis 1 *State-of-the-art human-aware local trajectory planners for mobile robots do not* *perform superior to traditional algorithms regarding the navigation task efficacy and mitigating discomfort among humans in the robot's environment.*

Thesis 1 suggests that quantitative evaluation of human discomfort and the robot's navigation task efficacy is feasible with metrics relevant for assessing social robot navigation. A thorough comparison of local trajectory planners relies on conducting the controlled study, in which each validated algorithm operates under the same environmental conditions and factors influencing the results of experiments are isolated.

The evaluation criteria are multifaceted and originate from the grounded requirements for social robot navigation. They include the robot's task performance, measured as the time required to reach the goal pose, as well as human discomfort ratings, assessed separately regarding the robot's motion naturalness, and the perceived safety of humans.

Thesis 1 could be proven if traditional algorithms achieve better or comparable quantitative indicators of robot performance and human comfort ratings.

Thesis 2 *A local trajectory planning method can be developed to enable robots to operate effectively in environments shared with humans, with the effectiveness quantified by surpassing performance compared to existing traditional and human-aware local trajectory planning algorithms regarding robustness, navigation task efficacy, and mitigating discomfort among humans in the robot's environment.*

Thesis 2 suggests that developing a method for adapting robots to operate in populated environments is feasible, and the degree of adaptation is measurable. To substantiate this thesis, performance verification of a new local trajectory planner against existing traditional and human-aware approaches must be conducted. A thorough comparison of algorithms requires the implementation of a controlled study, such as for proving thesis 1.

Thesis 2 states that an alternative planner could perform comparably to traditional trajectory planners regarding navigation task performance and could surpass state-of-the-art human-aware trajectory planners in terms of social indicators, assessed separately for the naturalness of robot movements and human perceived safety. Compared to the verification of thesis 1, the evaluation criteria also consider the robustness of examined algorithms, which should be advantageous in a novel method.

Thesis 2 could be proven if a method is established that achieves comparable or better results regarding quantitative metrics of robot navigation performance and better scores concerning human comfort indicators than existing approaches.

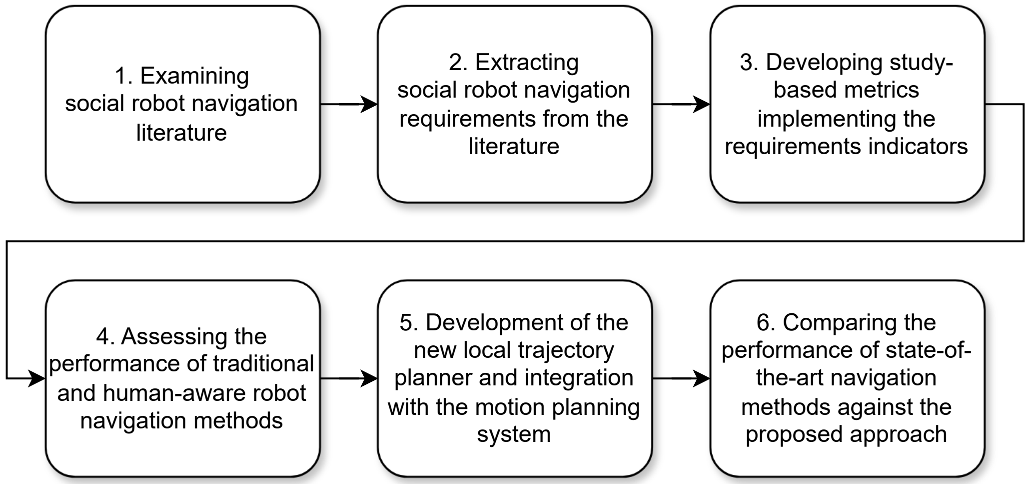


Figure 1.2: Basic workflow of the tasks performed as part of preparing the thesis.

1.4 Thesis contribution

This thesis investigates human-aware robot navigation from different perspectives and contributes to the advancements in the field. The significant additions to the state-of-the-art are outlined below, while the general workflow of tasks conducted to prepare this dissertation is outlined in Fig. 1.2. Parts of the thesis have already been published as journal articles and conference papers, which are identified in relevant contributions.

Contribution 1 *Review of the state-of-the-art literature to obtain study-based requirements for social robot navigation.*

The extensive literature review is a substantial contribution to the research topic discussed in this work. In contrast to previous review works, our survey aimed to explicitly demonstrate how the key concepts explored by robotics and social sciences researchers can be transferred into requirements for robot control systems implementing robot navigation tasks.

Contribution 1.1 *Grounding of social robot navigation requirements to form a taxonomy of elementary necessities.*

Our review reaches user studies to gather insights and perform the grounding of social robot navigation requirements (Chapter 2), which should be implemented in comprehensive navigation systems. Specifically, the taxonomy of requirements distinguishes the following groups of necessities: requirements regarding the physical and perceived safety of humans, requirements for assessing robot motion naturalness, and compliance with social conventions. After identifying those core principles, perception, motion planning and evaluation methods are reviewed in Chapter 3. A proper grounding of fundamental features helps to address the problem of researchers, who often try to implement different robot control strategies in an ad-hoc manner to mimic human behaviours.

Contribution 1.2 *Classification of existing social robot navigation approaches and state-of-the-art evaluation benchmarks according to the proposed requirements' taxonomy.*

As a part of the literature survey, the classification of state-of-the-art methods for adapting robots for operation among humans has been established based on the proposed requirements taxonomy. Specifically, the recent socially-aware robot navigation algorithms (Tab. 3.1 and 3.2), as well as benchmarks for the quantitative evaluation (Tab. 3.5) have been organised.

The classification of the social robot navigation requirements established in this study enables the identification of the gaps in motion planning algorithms, the drawbacks of state-of-the-art evaluation methods, and the proposal of relevant future work perspectives for researchers in the field.

The literature review is presented in Chapter 2 and 3, and those chapters constitute the extended version of our survey that has been published in [19].

Contribution 2 *Design and implementation of quantitative metrics for evaluating social robot navigation.*

The problem of the quantitative assessment of social robot navigation is one of the core challenges identified in [17]. Various metrics for the evaluation of traditional robot navigation concepts have already been implemented in different benchmark systems, but the indicators for assessing human-aware robot navigation are lacking. Therefore, we designed additional metrics to evaluate algorithms' compliance with the requirements regarding the physical and perceived safety of humans, as well as requirements for assessing robot motion naturalness. The metrics proposed for evaluating human-aware robot navigation are defined directly based on the findings from various user studies and the grounded requirements.

The novel metrics were embedded in our Social Robot Planner Benchmark (*SRPB*) system, which regards the assessment from the traditional and human-aware navigation perspectives. Compared to the state-of-the-art benchmark systems, our approach expands the diversity of metrics for evaluating navigation performance and introduces novel metrics focused on human awareness concepts. Another substantially original aspect is that the social-awareness indicators account for people tracking uncertainty, facilitating the evaluation using robot onboard perception. Our benchmark can be used to test robots operating in simulated and real-world environments. Moreover, metrics were formulated to allow the benchmark's usage with different robot types (either with nonholonomic or holonomic drives).

This contribution is thoroughly presented in Chapter 4, the majority of which has been published in the conference paper [20] and journal article [21].

Contribution 3 *Design and implementation of a human-aware local trajectory planner using the hybrid trajectory candidates generation method and spatiotemporal cost functions.*

This work aims to develop an algorithm that enhances the navigation quality of robots operating in environments shared with people. A key step in achieving that is to propose a new human-aware trajectory planning method that regards constraints arising from the presence of humans in the robot's environment while providing navigation task performance comparable to state-of-the-art traditional approaches.

The algorithm developed according to the thesis' objectives, named *HUman-Aware Trajectory Planner MApping the Pedestrians Motion Pattern* (*HUMAP*), is a geometric planner that solves the problem of receding horizon trajectory planning for dynamic systems operating in unstructured environments. The proposed approach is suitable for differential drive and holonomic[5] robots. Distinctive characteristics of the planner are the hybrid approach to trajectory candidates generation and the multifaceted objective function for scoring trajectory candidates.

The hybrid approach for producing various trajectory candidates employs two generation methods. The first relies on a pedestrian motion model, whereas the second samples the feasible velocity control commands [22]. The novelty of the approach lies in: extending the *Social Force Model*-based [1] pedestrian motion model to obtain emphasised collision avoidance behaviours and improved motion legibility, exploiting the parameterisation of the deterministic motion model to produce various trajectory candidates, and combining two trajectory candidate generation methods for online planning.

In contrast, the objective function regards both navigation performance as well as the physical and perceived safety of humans. As these objectives are contradictory, the planner searches for a Pareto-optimal solution in each planning step, successfully achieving real-time operation. Obedience to social norms is provided using a behaviour-based approach, as the planner operates in various modes, allowing for capturing customary spatiotemporal protocols of pedestrian motion.

[5] The support for holonomic robots is not yet implemented, but it is straightforward to integrate such robots with the proposed planning framework.

Contribution 3.1 *Extension of a Social Force Model-based pedestrian motion model with a Fuzzy Inference System to increase motion legibility and emphasise realistic collision avoidance.*

The baseline *Social Force Model*-based pedestrian motion model has been extended with a novel term emphasising proactivity in collision avoidance movements to enhance motion legibility [23], understood as intent expressiveness. The new component included in the model is based on a *Fuzzy Inference System* and affects the generated robot trajectories only in proximity to humans or other robots. The formulated model is called the *Fuzzy-Extended Social Force Model* (*FESFM*) and is used in the model-based trajectory candidates generator.

This contribution has only been mentioned in the conference paper [24] and is described in detail in Sec. 5.5.5.

Contribution 3.2 *Diversifying trajectory candidates of a deterministic pedestrian motion model.*

Numerous extensions of the original *Social Force Model* approach were proposed to develop models specialised for unfocused interactions or specific navigation tasks, e.g., accompanying (Sec. 3.2.2). Each formulation establishes a distinctive set of parameters to be adjusted to obtain a desired robot behaviour. Therefore, a significant part of the *SFM* research is related to the calibration of those models, i.e., the search for parameter values that provide the best approximation of the intended behaviour (Sec. 5.5.3).

The parameters of the baseline pedestrian motion model [25] employed by the model-based trajectory generator of the *HUMAP*, were calculated based on human movement trajectories from real-world pedestrian traffic video sequences. However, the model parameters exhibited significant standard deviations from the mean values. Therefore, the proposed approach evaluates model parameters across the range of meaningful values, so numerous trajectory candidates are obtained from the *FESFM* deterministic pedestrian motion model.

This contribution has been described in [24]. The thesis discusses that topic in Sec. 5.5.7.

Contribution 3.3 *Design of cost functions for assessing the social awareness of robot trajectories.*

Implementing social acceptance in robot navigation can be performed in different ways (Sec. 3.2.2). In the *HUMAP*, the objective function for scoring trajectory candidates captures constraints related to the physical and perceived safety of humans and the naturalness of robot motion. The social cost functions included in the objective are based on the quantitative metrics implemented in the *SRPB*, which are directly derived from the study-based social navigation requirements. Therefore, the transition of findings related to the offline assessment of robot trajectories has been performed to obtain the cost functions suitable for online trajectory candidate evaluation.

This contribution has only been briefly described in [24]. It is thoroughly discussed in Sec. 5.7.

Contribution 3.4 *Implementing the contextual awareness for efficient orchestration of the planner operation using the behaviour-based approach.*

A robot's intelligence is often regarded as utilising contextual information in its imperative [13, 26]. Including contextual data is often required for the robots' obedience to social conventions (Sec. 2.5).

The *HUMAP* planner considers environmental information to switch between different operational behaviours. Specifically, one of the behaviours implements a norm of yielding a way to a person crossing the robot's planned path, but additional norms can also be realised using the behaviour-based framework. Simultaneously, the potential occlusions of the planned global path are investigated, and, if necessary, the robot's behaviour is switched to perform an observatory action. These real-time adjustments enhance the robot's robustness in challenging practical scenarios.

This contribution was not previously discussed. The explanation is in Sec. 5.2.

Contribution 4 *Comparative experiments of various local trajectory planning algorithms in simulation and real-world environments.*

Experiments conducted for this thesis can be divided into two phases (Fig. 1.2). The first phase assessed how state-of-the-art methods for classical robot navigation perform against the recent human-aware algorithms. This part of the study also constitutes the validation of our *SRPB* benchmarking system. During the first part, 6 local trajectory planners have been evaluated in a controlled study, 2 of which are socially-aware methods. With the quantitative assessment, the proficiency of various planning algorithms has been compared in terms of navigation performance as well as human awareness in simulation and real-world environments using a TIAGo robot. The study aimed to identify whether socially-aware local trajectory planning is still an open problem, and the results are discussed in Sec. 7.3. The majority of findings in this scope have been published in the conference paper [20] and journal article [21].

The second phase of the experiments intended to validate the performance of the *HUMAP* planner against the state-of-the-art algorithms; hence, various tests were performed. Similarly, the controlled study was employed, and the *SRPB* benchmarking system was used to evaluate the robot's behaviour when operating with different planners.

Compared to the first phase of the experiments, different scenarios were designed, but still conforming to standardised guidelines [10, 11]. During the second part of the experiments, *HUMAP* was compared against 12 different state-of-the-art local trajectory planners (5 of which are learning-based methods) in 3 scenarios, each repeated 100 times, which, to the best of our knowledge, is the most extensive study in the field. The large-scale simulation study allowed for data collection for statistical analysis of the operation. The proposed planner was also validated in real-world tests in analogous 3 scenarios but only with planning methods ensuring the safe operation of the physical robot. Supplementing those experiments, the *HUMAP*'s performance across its various configurations and scenarios was examined. Parts of the results in that matter are included in [24], whereas a thorough description is provided in Sec. 7.4.

Notably, most of the implemented methods and metrics are available as **open-source software** to ease the further development of the field.

1.5 Problem scoping

The scope of the social robot navigation field is vast; hence, the range of topics included in the examination has to be limited due to practical and feasibility reasons. Defining the social robot navigation requirements from literature studies is among the main concentrations of this thesis. Based on the requirements, requirements-driven metrics related to the social acceptance of robots are developed, and the human-aware robot trajectory planning algorithm is proposed. Requirements, metrics, and the planning algorithm do not investigate the domains of, i.a., explicit communication or negotiation. Also, the range of interactions examined was limited to align with the scope of primary topics.

Communication Effective decision-making in socially-aware navigation requires communication between robots and humans, particularly when the robot's knowledge about the environment is limited. Specifically, explicit communication involves the auditory domain and written instructions, which robots should interpret and respond to. Robots also need to convey their intentions and decisions to humans, utilising verbal and visual techniques such as speech and gestures performed with onboard actuators. The topic of explicit communication is rather related to higher-level decision-making and has been explored to varying degrees in review works from the field [27, 28, 18]. In contrast, implicit communication is commonplace in human-robot interaction studies and is relevant to the topics investigated in this thesis.

Negotiation Negotiation in social robot navigation acts as a form of dynamic information exchange. This may involve collaborative decision-making processes, e.g., requesting permission to pass. While the scope of the negotiations field extends way beyond human-robot interaction, this concept has been briefly discussed in social robotics surveys [2, 18]; however, none of the primary concepts of this dissertation is examined with the focus on this matter.

Robot types What substantially affects the requirements and objectives of perception and human-aware robot motion planning is the type of robot. Variations in ground, aerial, or aquatic robots [14, 18] significantly impact possible scenarios; hence, also the range of human-robot interactions. Nevertheless, this thesis focuses on ground-wheeled robots. Additionally, although multi-robot systems are not investigated in this dissertation, the presence of other robots in the environment of the controlled robot is taken into account. Specifically, other robots are distinguishable from humans and treated as typical dynamic objects.

Interaction types The physical (contact-rich) interaction between robots and humans is a crucial topic in collaborative robotics and safety management. However, this study examines other types of interactions, namely, unfocused and focused [2], neither of which involve physical contact.

Cultural scope The requirements for social robot navigation were defined based on the findings resulting from user studies, which involved participants mainly from the Transatlantic cultural sphere, and less frequently from the Pan-Asian. On the other hand, the participants engaged in the experimental studies conducted as part of this thesis were only from the Central European cultural sphere.

1.6 Thesis outline

This thesis is organised as follows: firstly, an extensive literature review is allocated in Chapter 2 and Chapter 3, as the presented state-of-the-art analysis is segmented into two perspectives: *requirements* and *algorithmic*. Namely, Chapter 2 presents the definition of the study-based social robot navigation requirements and its content is organised according to the introduced taxonomy of the necessities. Then, Chapter 3 investigates the algorithmic solutions for adapting robots to operate in human environments. Notably, the majority of Chapter 2 and Chapter 3 have been published in our survey article [19].

Next, in Chapter 4, our method for quantitative evaluation of socially-aware robot navigation algorithms is proposed. Specifically, this chapter presents the mathematical formulation of metrics implemented in our benchmark. It contains an extended explanation of the proposed approach published in the conference paper [20] and the journal article [21].

In the following part of the thesis, Chapter 5, the problem of human-aware robot trajectory planning is formulated, and the contribution to the topic is described. The chapter describes our hybrid method of trajectory candidates generation and specifies the components of the objective function used for scoring candidates and selecting an optimal trajectory. Parts of this chapter have been included in [24], but a detailed description of the approach is provided.

Next, Chapter 6 discusses the implementation and integration aspects of the robot control system. The allocation of developed algorithms is also explained in the context of the entire system, which has been used during the experiments described in Chapter 7, where the tests validating the proposed methods are discussed. Namely, that part of the thesis presents the two phases of the experiments that have been conducted. In the first, the state-of-the-art traditional and human-aware methods for robot navigation were compared using the quantitative benchmarking method (Chapter 4). In the second phase, the proposed trajectory planning method was assessed against numerous planners using the same evaluation methodology. The design of the experiments and selection of algorithms for the comparative controlled studies of each section of this chapter are explained in detail. Most results from the first phase of the experiments (Sec. 7.3) have been encompassed in our previous works [20, 21], while elements of the second phase of the experiments (Sec. 7.4) have been included in [24].

Finally, the Chapter 8 constitutes conclusions drawn from the literature review and experimental studies to prove the theses stated in the introduction. The summary is complemented with future work perspectives regarding proposed approaches for social robot navigation.

The thesis is enhanced with two appendices, in which the commonly referred algorithms are presented.

Chapter 2

Requirements for socially-aware navigation

Social robots were introduced to make human-robot interaction more natural and intuitive [29]. Generic characteristics of social navigation are commonly recalled in review works. For example, Kruse et al. [12] classify the main features as safety, comfort, naturalness, and sociability. On the other hand, in [2], the authors indicate key factors as distinguishing obstacles from persons, considering the comfort of humans – their preferences and their needs, not being afraid of people and the legibility of motion intentions. More recently, Mavrogiannis et al. [17] proposed a classification that relies on proxemics, intentions, formations and social spaces, ordered according to the social signal richness. Furthermore, Francis et al. [11] stated that principles of social robot navigation include safety, comfort, legibility, politeness, social competency, agent understanding, proactivity, and contextual appropriateness.

While the aspects above schematically display the goals of social navigation, the authors of the surveys do not attempt to extract the straightforward requirements to follow in social robot navigation. Instead, these terms are loosely defined; hence, might refer to different means in different contexts or applications. As a consequence, it is tough to determine how to effectively gauge whether the robot behaves in a socially-compliant manner. Our survey aims to reduce those abstract terms describing social norms. This is contrary to other review works, where, although taxonomies are presented and articles are classified into those groups, the fundamental concepts persist as vague definitions.

Thus, we perform the grounding of social robot navigation requirements. The requirements must be known to properly design a socially-aware robot navigation system. Various techniques have been experimented with an assertive robot, revealing that using knowledge from psychology leads to increased user trust [30]. Incorporating a study-driven approach, we reached the human-robot interaction user studies to determine how humans perceive the robot navigating around them and how robots should behave around humans under certain controlled conditions. Such an approach aims to explicitly demonstrate how the key concepts explored by researchers in robotics and social sciences can be transferred into requirements for robot control systems [31] implementing robot navigation tasks.

Notably, we separated the study-based grounding of social robot navigation requirements (Chapter 2) from algorithmic approaches to resolving them (Chapter 3). Requirements are obtained from the results of user studies, whereas an algorithmic perspective is presented based on technical papers from the robotics field. Precise requirements grant implementation guidelines and straightforward evaluation of whether the robot behaves as expected.

2.1 Taxonomy of requirements for social robot navigation

Social robot navigation **extends** the requirements of classical navigation with capabilities to accommodate social interaction between robots and humans. Traditional robot navigation emphasises generating collision-free motions for a robot to move to the goal pose as fast as possible. This requires environment sensing for obstacle detection, efficient global pose estimation, and usually map building.

Traditional robot navigation requirements As the classical robot navigation requirements are not the main focus of the considerations, they will only be briefly explained with relevant resources from the literature. Specifically, the robot task performance maximisation aspects (**Req. 1**) are divided into five groups. The first requirement is avoiding collisions with the environment (**Req. 1.1**), which is straightforward, as it can cause damage to the environment or a mobile base. The second one is planning trajectories that are feasible for the mobile base (**Req. 1.2**), which is crucial since the planned trajectories might not be executable due to kinodynamic constraints of a robot [32]. Another requirement is reaching the goal as fast as possible (**Req. 1.3**), which decreases the time of a navigation task [33, 10]. The last two necessities are: reaching the goal by taking the shortest possible path (**Req. 1.4**) [33, 34, 35] and minimising path irregularity (**Req. 1.5**) [36, 37, 10], both helping to reduce the energy expenditure [38] of mobile robots. Graphical abstract of this taxonomy is presented in Fig. 2.1.

Social robot navigation requirements On the other hand, the main objective of social navigation is to mitigate human discomfort caused by robot movements (**Req. 2**). Our taxonomy of social robot navigation requirements (Fig. 2.2) involves the physical

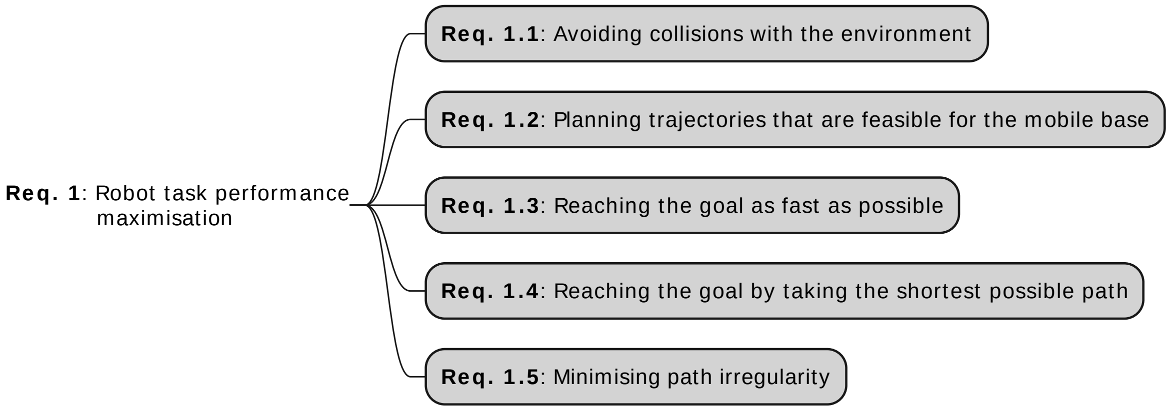


Figure 2.1: General taxonomy of traditional robot navigation requirements.

safety of humans (**Req. 2.1**), perceived safety of humans (**Req. 2.2**), the naturalness of robot motion (**Req. 2.3**) and the robot's compliance to social norms (**Req. 2.4**). Specifically, the perceived safety of humans mostly relies on proxemics theory and the prevention of scaring a human. In turn, robot motion naturalness does not affect the safety aspects of humans but regards the trustworthiness of the robot. Lastly, abiding by social conventions focuses on actions and sequences that require rich contextual information to mitigate human discomfort.

Our general taxonomy is designed to classify the essential concepts of social robot navigation clearly and unambiguously into one of the investigated groups to create a generic framework. We expect that the main characteristics selected for the taxonomy will stay pertinent in the future, with the possibility of incorporating additional attributes.

In the remaining part of this section, the social robot navigation requirements are discussed, while the algorithmic concepts describing how those socially-aware navigation responsibilities can be embedded into robot control systems are discussed in Sec. 3.1 and Sec. 3.2.

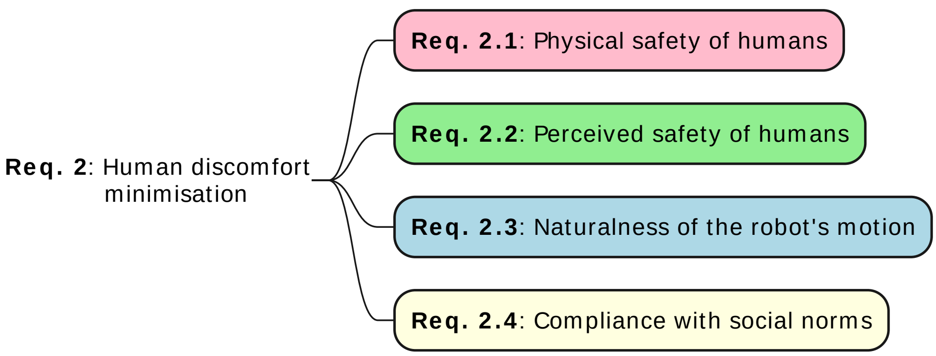


Figure 2.2: General taxonomy of social robot navigation requirements.

2.2 Physical safety of humans (Req. 2.1)

The physical safety of humans is closely related to the collision avoidance capabilities of robots (**Req. 1.1**). Social robot navigation inherits this skill from the classical robot navigation requirements.

Francis et al. [11] denote physical safety as the first principle of social navigation that intends to protect humans, other robots and their environments. Physical safety of humans during navigation is discussed in the newer literature [27, 39] but has already been addressed as a fundamental robotics challenge several decades ago [40]. Nonetheless, the physical safety of other robots or machines is also of great significance [41, 42, 43, 17].

For example, Guzzi et al. [36] conducted a study with multiple small-scale robots relying only on local sensing and employing proactive planning integrated with heuristic pedestrian motion model [44]. In real-world experiments, in a crossing scenario, they observed different frequencies of collisions depending on the sensors' field of view and safety margin; hence, the collision count was used as one of the metrics for assessing the safety margin parameter. Evaluating *time-to-collision* (TTC) is a proactive method to anticipate incoming collisions [45, 46] that was also embedded in some benchmarks [35].

2.3 Perceived safety of humans (Req. 2.2)

The comfort of humans around robots is crucial; however, the robot's behaviour can influence that, potentially causing annoyance or stress [12, 11]. Human discomfort during robot navigation often corresponds to a diminished perceived (or psychological) safety of humans. Perceived safety is the factor that might lead to physical safety violations (Sec. 2.2) if not addressed adequately beforehand. Stress-free and comfortable human-robot interaction is a broad topic [27] influenced by numerous features (Fig. 2.3), including adherence to spatial distancing [47, 2], performing natural movements [10], preventing of scaring or surprising a human [12]. The remaining part of this section discusses them in detail.

2.3.1 Regarding personal spaces of individuals (Req. 2.2.1)

Proxemics is the most prominent concept regarding social distancing rules [47, 48, 49]. Some fundamental studies connected to proxemics theory confirm that the psychological comfort of humans is affected by interpersonal distancing [48, 50, 51]. Butler and Agah [52] explored the influential factors of how humans perceive a service robot during unfocused interactions. One of them was the distance factor, which induced feelings of discomfort or stress in some configurations. A similar study was conducted by Althaus et al. [53], who

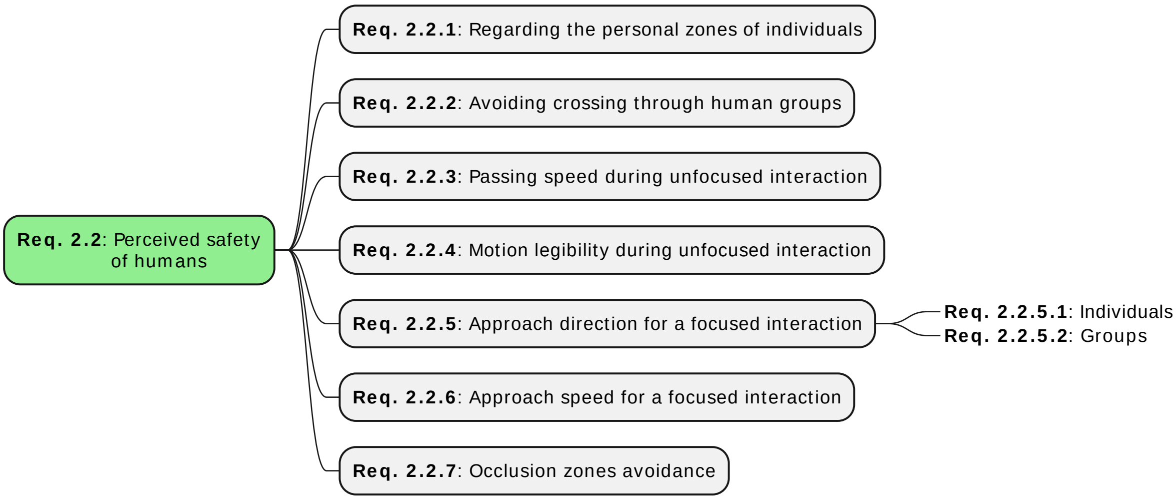


Figure 2.3: Taxonomy of social robot navigation requirements related to the perceived safety of humans.

validated a navigation system that respects the personal spaces of humans in a real-world study.

Shapes of a personal zone impact the comfortable passing distances. Hall originally specified four circular spaces [47], while the personal zone, reserved for friends, is usually regarded as a no-go zone during unfocused human-robot interaction. Entering the personal zone is counted as a violation of comfort and safety [54, 2, 7]. The classification of all proxemic zones was described in detail in prior surveys, e.g., [2].

The initially suggested circular shape of the personal space [47] might not appropriately capture the features of human perception and motion. Further empirical studies suggested extending that to an egg shape [55], ellipses [1, 56], asymmetrical shapes [57] (prolonged on the non-dominant side), or changing dynamically [58]. In [57], it is also reported that the size of personal space does not change while circumventing a static obstacle regardless of walking speed and that the personal space is asymmetrical. The natural asymmetry of personal spaces is also reported in [59], where authors found out that if the robot has to approach a human closely, it is preferred to not move behind a human, so they can see the robot.

Numerous works conducted human-involving experiments to gather empirical data and to model complex and realistic uses of space [60, 61, 62, 63, 64]. Participants of the study in [60] rated distances between 1.2–2.4 m as most comfortable for interaction situations. Experiments by Huettenrauch et al. [65] confirmed that in different spatial configurations, 73–85% of participants found Hall's personal distance range (0.46–1.22 m) as comfortable. Torta et al. [66], in their study involving human-robot interaction, examined the length of comfort zones as specific values of 1.82 m for a sitting person and 1.73 m for a standing person.

Pacchierotti et al. [61, 62] examined discomfort as a function of, e.g., lateral distance gap in a hallway scenario. The lateral gap was also examined by Yoda and Shiota [67] in terms of the safety of passing a human by a robot in a hallway scenario. Three types of encounters were anticipated as test cases for their control algorithm, including a standing, a walking, and a running person. They approximated human passing characteristics from real experiments, defining clear formulas to follow in a robot control system. The authors found that the average distance between the passing humans depends on their relative speed and varies from 0.57 to 0.76 m.

The authors of [63] found that the discomfort rates differ between intrusions and extrusions from personal spaces, and the distances of approximately 0.85–1.0 m are the most comfortable for a focused interaction with a stranger. On the other hand, Neggers et al. [64] conducted a study similar to [62] and compared their results. They obtained similar outcome and reported that the same function, inverted Gaussian linking distance and comfort, can be used to fit the results' data with only a small comfort amplitude shift between [62] and [64]. The authors of [64] also attempted to model an intrusion into personal space as a distance-dependent surface function.

However, there are also diverse exceptions to the mean shape of personal space. For example, Takayama et al. [68] indicated that during the study, participants with prior experience with pets or robots required less personal space near robots compared to people who do not possess such experience. Furthermore, a study presented in [69] endorses the concept that personal space is dynamic and depends on the situation. Velocity-dependent personal space shapes were also considered appropriate in [70, 71, 72].

Since various studies, even though conducted differently, yield similar results, they seem to approximate human impressions while interacting with robots and, as a consequence, allow modelling of the real-world phenomena of social distancing. The conclusions from the mentioned user studies give insights regarding the implementation of personal space phenomena in robot control systems.

2.3.2 Avoiding crossing through human groups (Req. 2.2.2)

Recent research revealed that pedestrians tend to travel in groups [73, 74]. Human groups create focused formations (F-formations) [75] – spatial arrangements that are intended to regulate social participation and the protection of the interaction against external circumstances [2]. F-formations might be static – consisting of people standing together engaged in a shared activity, or dynamic – consisting of people walking together, and might have different shapes [75, 2].

The necessity of avoiding crossing F-formations arises from the fact that they always contain an O-space which is the innermost space shared by group members and reserved for in-group interactions. The discomfort caused by a robot to a group might be assessed as the robot's intrusion into the O-space of the F-formation [76, 77]. Results of numerous studies confirm that humans involved in an F-formation keep more space around a group than the mere addition of single personal spaces [78, 79, 80]; thus, individuals stay away from social groups. Furthermore, the research by Rehm et al. [81] found that participants from high-contact cultures stand closer to a group of people compared to people from low-contact cultures.

A general guideline for robots navigating through populated environments is to avoid cutting through social groups [82], but if it is not possible, e.g., in a narrow corridor, to politely pass through the O-space [83, 11].

2.3.3 Passing speed during unfocused interaction (Req. 2.2.3)

Rios-Martinez et al. [2] define unfocused interactions as "interpersonal communications resulting solely by virtue of an individual being in another's presence". As already highlighted in Sec. 2.3.1, excessive or insufficient passing speed proved significant in terms of discomfort among humans involved in an unfocused interaction with a robot in numerous experimental studies [52, 61, 62, 72].

The most comprehensive study in that matter was recently proposed by Neggers et al. [72], who assessed human discomfort with a robot passing or overtaking them at different speeds at different distances. They have found that higher speeds are generally less comfortable for humans when a robot moves at smaller distances. The authors claimed the inverted Gaussians with variable parameters accurately approximate the experimental results for all combinations of scenarios and speeds. The approximation of their findings with a continuous multivariable function has already been implemented[1] and can be used for evaluating robot passing speed.

2.3.4 Motion legibility during unfocused interaction (Req. 2.2.4)

Studies conducted by Pacchierotti et al. [62] examined a mutually dynamic situation of passing each other. They assessed human discomfort as a function of the lateral distance gap in a hallway scenario. What they have found is that there was no significant impact of lateral gap size when a robot signalled its passing intentions early. This notion is often referred to as motion legibility, which is an intent-expressive way of performing actions [23]. It can be increased by explicit signalling and also enriching behaviour, so it can be used as a cue to the robot intention [84, 85].

[1] `https://github.com/rayvburn/social_nav_utils`

Lichtenthäler et al. [86] found a significant correlation between the perceived safety and legibility in their study. Gao and Huang [10] considered a flagship example of motion legibility as a scenario where a robot quickly moves towards a person, adjusting its trajectory just before an imminent collision. Despite avoiding direct physical contact, such behaviour is likely to produce notable discomfort by the robot heading direction [21] due to lack of early signalling.

2.3.5 Approach direction for a focused interaction (Req. 2.2.5)

Approaching direction to initiate a focused interaction is a broad field of social robot navigation studies. Rios-Martinez et al. [2] describe focused interaction as "occurring when individuals agree to sustain a single focus of cognitive and visual attention". In most experimental cases, focused interaction involves approaching to start a verbal communication or to hand over the transported goods. The taxonomy in this matter separates approaching guidelines between individuals and F-formations.

Individual humans (Req. 2.2.5.1) In studies conducted by Dautenhahn et al. [87] and Koay et al. [88], participants were seated and asked to gauge their discomfort levels during the handover of objects by a robot that approached from various directions. The subjects of the study preferred frontal approaches over diagonal approaches from the left or right. The contradictory results were found in a study by Butler and Agah [52], where standing participants preferred an indirect approach direction.

Multiple studies depict that human preference is to be approached from the front and within the human field of view. [89, 90, 91, 86, 92, 93, 94, 95]. Walters et al. [89] examined the robot's behaviour of approaching a human for a fetch-and-carry task. The authors reported that seating participants found the direct frontal approach uncomfortable. The general preference was to be approached from either side, with a preference biased slightly to a rightward approach by the robot. However, the study depicted that a frontal approach is considered acceptable for standing humans in an open area. Another conclusion derived from this study is that humans prefer to be approached from within their field of view; hence approaching from behind should be avoided.

Torta et al. [91] conducted a user study considering different robot approach directions with the final pose at the boundary of a personal space. Similarly, they found that experiment subjects (seated) assessed frontal approach directions (up to $\pm 35°$) as comfortable while perceived farthermost ($\pm 70°$) as uncomfortable. Comparable outcomes ensued from the study in [90]. Unlike the results of the user study performed by Dautenhahn et al. [87], in [91], no significant difference was found when the robot approached from the right side or the left side.

Furthermore, Koay et al. [92] researched robot approach distances and directions to a seated user for a handover task. The results show that the preferred approach direction is from either side at a distance of about 0.5 m from the subjects. An interesting fact is that this distance lies within an intimate space [47], but was preferred because prevented humans from reaching out longer with their arms or standing up to pick up the goods from the robot's tray.

Human groups (Req. 2.2.5.2) Approaching groups of humans requires slightly different strategies. Ball et al. [94] investigated the comfort levels of seated pairs of people engaged in a shared task when approached by a robot from eight directions. Participants rated robot approach behaviour for three spatial configurations of seats. Approaches from common (to all subjects involved) "front" directions were found to be more comfortable (group's average) than from a shared rear direction. When seated pairs were in a spatial configuration that did not exhibit the common "front" or "rear" direction, no significant statistical differences were found. However, another finding of this study is that the presence and location of another person influence the comfort levels of individuals within the group.

Joosse et al. [95] explored the optimal approach of an engagement-seeking robot towards groups from three distinct countries, employing Hall's proxemics model [47]. Their findings indicate that the most suitable approach distance seems to be approximately 0.8–1.0 m from the centre of the group.

Karreman et al. [93] investigated techniques for a robot to approach pairs of individuals. Their findings revealed a preference among people for frontal approaches (regardless of side), with a dislike for being approached from behind. They also noted that environmental factors appeared to influence the robot's approach behaviour.

2.3.6 Approach speed for a focused interaction (Req. 2.2.6)

Robot speeds are one of the factors impacting discomfort when approaching a human. Since the literature regarding approaching behaviour is rich, there are also guidelines to follow in social robot navigation.

Butler and Agah [52] assessed the navigation of a mobile base around a stationary human using various trajectories and equipment resembling the human body. They discovered that speeds ranging from approximately 0.25 to 0.4 m/s were most comfortable, while speeds exceeding 1 m/s were uncomfortable. They also claimed that there might be a speed between 0.4 and 1.0 m/s that produces the least discomfort.

Sardar et al. [96] conducted a user study in which a robot approached a standing individual engaged in another activity. Experiments revealed notable distinctions in acceptance of invading the participant's personal space by a robot and a human. In their study, only two speeds were evaluated, namely 0.4 and 1.0 m/s, while the robot's faster speeds were more trustworthy (opposite to human confederates).

In a more recent study, Rossi et al. [97] evaluated speeds of 0.2, 0.6 and 1.0 m/s that affected the robot's stopping distance while approaching. They have found different human preferences for stopping distance depending on the activity currently executed by humans. Sitting participants favoured shorter distances while walking subjects longer.

2.3.7 Occlusion zones avoidance (Req. 2.2.7)

Occlusion zones are related to areas not reached by the robot's sensory equipment. Despite the robot's most recent assumptions suggesting that these areas were previously unoccupied, such estimates may be inaccurate. Consequently, robots should avoid traversing near blind corners, as they may fail to detect individuals behind them, and vice versa. By going around the corner with a wider turn, the robot can explore the occluded space earlier, making it possible to react to humans sooner [12]. Proactivity in that matter prevents surprise or panic and generally positively impacts comfort and physical safety.

User studies generally confirm this issue, showing that humans tend to shorten their paths [98, 99] to minimise energy expenditure. Taking shortcuts in public spaces increases the risk of encounters around blind corners.

Francis et al. [11] suggested that a robot entering a blind corner should communicate intentions explicitly with voice or flashing lights. However, this seems slightly unnatural, as even humans avoid shouting in corridors. Enabling audio or flashing lights might also be annoying for surrounding workers in shopping aisles.

2.4 Naturalness of the robot's motion (Req. 2.3)

The naturalness of the robot's motion can be referred to as emerging robot behaviours that are not perceived as odd. This is often related to the avoidance of erratic movements and oscillations (Fig. 2.4). Keeping a smooth velocity profile also produces an impression of trust and legibility among observing humans [86].

2.4.1 Avoiding erratic motions (Req. 2.3.1)

Erratic motions involve sudden changes in velocity, making it difficult to anticipate the next actions. This term is often used to describe the behaviour of objects exhibiting chaotic movement patterns that make the robot look confused.

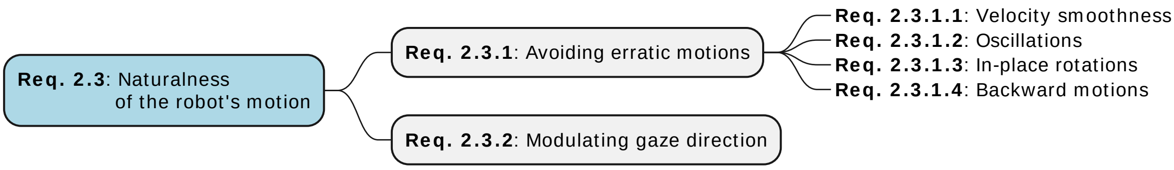


Figure 2.4: Taxonomy of social robot navigation requirements related to the naturalness of the robot's motion.

Erratic motions are often related to the lack of smoothness of the robot's velocity profile (**Req. 2.3.1.1**). Natural motions favour movements with a minimum jerk [100] with mostly stable linear velocity and the angular velocity of zero, i.e., adjusting orientation only when necessary [12, 10].

In contrast to the smooth velocities, oscillating motions (**Req. 2.3.1.2**) involve alternating forward and backward motions, where the robot effectively does not make any progress. They may be present in some navigation approaches that rely solely on *Artificial Potential Field* [101] or *Social Force Model* [1].

Additionally, in-place rotations (**Req. 2.3.1.3**) of a robot appear unnatural for human viewers; hence, it is preferred to avoid trajectories where a turning on spot [100, 102]. Also, significant backward movements (**Req. 2.3.1.4**) should also be avoided as individuals rarely move in reverse in public areas. Such actions can pose collision risks, particularly for mobile bases lacking range sensors at the back.

2.4.2 Modulating gaze direction (Req. 2.3.2)

A broad area of research regarding motion naturalness corresponds to modulating the robot gaze direction. Humanoid robots are typically equipped with a "head", inside which a camera is located (*RGB* or *RGB-D*), e.g., Nao, TIAGo, Pepper, Care-O-bot. Pan and tilt motions of the head joints can be used to modulate gaze direction.

Gaze direction is considered one of the social signals (cues) and a specific type of nonverbal communication between a robot and surrounding humans [28]. Among humans, it is closely related to their perception captured by the notion of *Information Process Space* [2]. Gaze is a general concept in which measurable aspects can be evaluated, such as fixation count and length [103], as well as gaze-movement angle [104]. Both provide valuable insights into human trajectory or behaviour prediction [28].

Unfocused interaction In a study by Kitazawa and Fujiyama [105], the authors investigated gaze patterns in a collision avoidance scenario with multiple pedestrians moving along a corridor. Results of the experiment show that humans pay significantly more attention to the ground surface, which they explain as a focus on detecting potential dynamic hazards than fixating on surrounding obstacles. In an experiment conducted by Hayashi et al. [106], they noticed that participants were more willing to speak to the robot when it modulated its gaze direction. Kuno et al. [107] also concluded that robot head movement encourages interaction with museum visitors.

Fiore et al. [108] analysed human interpretation of social cues in hallway navigation. They designed a study to examine different proxemics and gaze cues implemented by rotating the robot sensors. The results depict that the robot's gaze behaviour was not found to be significant, contrary to the robot's proxemics behaviour that affected participant impressions about the robot (Sec. 2.3.1). Similarly, a study by May et al. [109] showed an understanding of robot intentions while conveyed using different cues. It turned out that the robot was understood better when a mechanical signal was used compared to using the gaze direction cue. Also, Lynch et al. [110] conducted a study employing a virtual environment where virtual agents established a mutual gaze with real participants during path-crossing encounters in a virtual hallway. Subjects of a study found the gaze factor as not important to inferring about paths of the virtual agents.

Different strategies of gaze modulation were studied by Khambhaita et al. [111]. Their research indicates that the robot's head behaviour of looking at the planned path resulted in more accurate anticipation of the robot's motion by humans compared to when the head was fixed. The authors also found that the robot operating with the head behaviour of alternately looking at the path and glancing at surrounding humans gave the highest social presence measures among the subjects. Similarly, Lu et al. [112] discussed a strategy of a robot looking at the detected human followed by looking ahead in 5-second cycles.

Focused interaction Research has shown that gaze modulation of the robot's focused interactions should be treated differently than unfocused ones. Breazeal et al. [113] explored the impressions of humans participating in an experiment with a Kismet robot capable of conveying intentionality through facial expressions and behaviour. They identified the necessity of gaze direction control for regulating conversation rate, as the robot directs its gaze to a locus of attention.

In another study, Mutlu et al. [114] implemented a robot gaze behaviour based on previous studies [115, 116] and their observations that people use gaze cues to establish and maintain their conversational partner's roles as well as their own. The gaze behaviour strategy produced turn-yielding signals only for conversation addressees. In their experiment, they found that using only the gaze cues, the robot manipulated who participated in and attended to a conversation.

2.5 Compliance with social norms (Req. 2.4)

Navigating humans adhere to diverse social norms influenced by cultural, interactional, environmental, and individual factors such as gender and age. Therefore, the robot's compliance with social conventions is also a multifaceted concept (Fig. 2.5), in contrast to low-level motion conventions, such as approach velocity. The aforementioned factors shape high-level social conventions involving navigation-based interactions like queueing, elevator decorum, yielding way to others, and adhering to right-of-way protocols. Robots considered sociable abide by social conventions. Despite the existence of customary routines, they are often challenging to model precisely due to their abstract nature, as seen in the discussion by Barchard et al. [117].

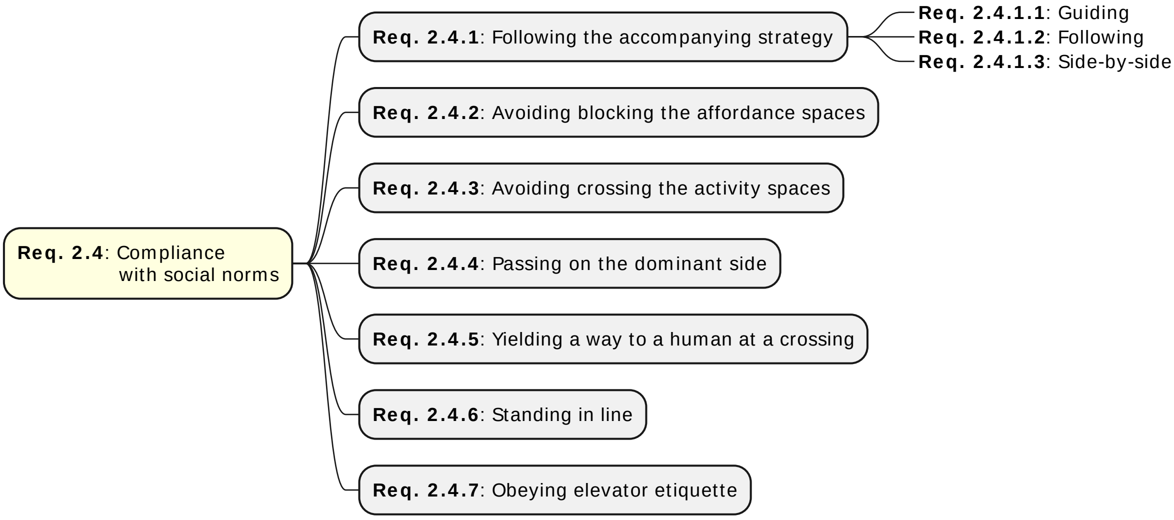


Figure 2.5: Taxonomy of social robot navigation requirements related to the robot's compliance with social norms.

The authors of surveys [12, 10] exemplify that even if the robot's movements may appear natural and unobtrusive (**Req. 2.3**), it can violate typical social conventions. For instance, entering a crowded elevator without allowing occupants to exit first breaches common expectations, thereby potentially causing discomfort. Also, in different user studies, it is reported that human discomfort can be caused due to violations of social norms even if the rules of perceived safety of humans are properly adhered to in the robot navigation [118, 119].

There are no predetermined sets of high-level social conventions, making compliance a dynamic and context-dependent aspect of robotic behaviour [10], that requires a diverse level of contextual awareness. The most common and meaningful social conventions that have been examined in the literature are illustrated below. The complementary discussion attempts to clarify how they should be addressed in robot control systems.

2.5.1 Following the accompanying strategy (Req. 2.4.1)

Strategies of executing the task of accompanying humans by the robot are dictated by the social conventions of how humans navigate in relation to other pedestrians. Customary human behaviours entail how robots should adjust their movements based on the relative position of the accompanying human (or humans), ensuring smooth and natural interactions.

Tracking humans from the front (Req. 2.4.1.1) Numerous studies reviewed the relative pose that the robot should maintain while tracking a human from the front. For example, Jung et al. [120] performed a study to evaluate how often humans look back at the robot that tracks the subject from behind. They found that participants often looked back as they were curious about the robot, whether it bumped into them or tracked them well. The authors concluded that tracking from the front might be more comfortable and designed a robot control strategy that involves moving 1 m ahead of the tracked human, whose local movement goal is inferred by the robot online.

On the other hand, Young et al. [121] compared various relative poses for a robot led on a leash by a participant. The results revealed that having the robot move in front of the person was the most comfortable approach for joint motion. In another study, Carton et al. [122] proposed a framework for analysing human trajectories. Their studies led to the conclusion that humans plan their navigation trajectories similarly whether they are walking past a robot or another human.

Person following (Req. 2.4.1.2) Gockley et al. [123] evaluated methods of avoiding rear-end collisions of a robot following a person. The first approach focuses on direction-following, where the robot follows the heading of a person, whereas the second method, path-following, relies on imitating the exact path that a person takes. The participants of the real-world experiments rated the direction-following robot's behaviour as substantially more human-like. However, the participants rated that the robot stayed too far away (1.2 ± 0.1 m) from them while moving.

Following an individual in populated environments is challenging as crowd behaviour often manifests as flows of social groups, with individuals typically following the flow [73]. Studies show that joining a flow with a similar heading direction is more socially acceptable, resulting in fewer disturbances to surrounding pedestrians [124]. Collision avoidance techniques for following one person through a populated environment are discussed in [125, 126].

Side by side (Req. 2.4.1.3) The tendency for people to walk side-by-side when walking together was discussed by Kahn et al. [127]. In situations with only two individuals walking, they typically adopt a side-by-side formation, while in crowded conditions or with three or more individuals, more complex formations such as “V” shapes are observed [128]. Spatial preferences of humans when being followed by a robot were reviewed in [129]. In the majority of studies, the robot’s relative position to the person typically remains constant, with any adjustments being made primarily in response to environmental factors.

Saiki et al. [130] discussed how robots can serve walking people. In their experiments, people trajectories were recorded to develop a histogram of relative distances. The conclusion is that people’s average distance while walking alongside was 0.75 m.

Karunarathne et al. [131] designed a spatial model for side-by-side accompanying without explicit communication about the goal of a human. During their study, they found that, e.g., a distance maintained in a robot-human pair (1.25 m) was larger than that from the human pair average (0.815 m).

2.5.2 Avoiding blocking the affordance spaces (Req. 2.4.2)

The concept of affordance space relates to the potential activities that the environment offers to agents [132]. Affordance spaces could be mapped as free regions or banned regions in a function of time [133]. They have no specific shape [2] as they depend on specific actions.

Affordance spaces are specific to the robot environment and can be exemplified by the area near a painting in a gallery or menu stands in restaurants. In general, an affordance space can be crossed without causing disturbance to a human (unlike activity spaces in Sec. 2.5.3), but blocking an affordance space could be socially not accepted [2]. Also, for the robot with a limited *field of view* (FOV), it is essential to utilise a predefined map of affordance spaces.

Raubal and Moratz [134] discussed a robot architecture incorporating a functional model for affordance-based agents. The crucial concept is to consider the information about locations of affordance spaces when selecting a coarsely defined (region-based) navigation goal or a goal on a topological map. The notion of affordance spaces was also discussed in the context of learning them online [135], as well as in gaining knowledge from the analysis of human trajectories [136].

2.5.3 Avoiding crossing the activity spaces (Req. 2.4.3)

The activity space is an affordance space linked to an ongoing action performed by an agent – human or another robot [2]. An activity space can be exemplified by the area between an observer and a painting in a gallery. Once the visitor initiates this space, the robot is obliged not to cross it [132]. Additionally, the robot's perception has to dynamically infer whether a certain agent has initiated an activity space, e.g., by observing an object [135]. Furthermore, the activity space should be conditionally constrained; for instance, it should be less restrictive for a shorter robot compared to a taller one that might fully occlude the painting when crossing through an activity space.

2.5.4 Passing on the dominant side (Req. 2.4.4)

Bitgood and Dukes [99] discussed that people tend to proactively move to the right half portion of a hallway or a narrow passage, which is tied to cultural traffic rules. Multiple existing social robot navigation approaches already implemented strategies to follow the right side of the corridor or to favour passing humans on the right [137, 71, 126, 84]. However, as Bitgood and Dukes suggest, this might not be a strict rule to follow in crowded spaces, as some people follow the other side as they have an incoming left-turn destination [99]. This is supported by the study conducted by Neggers et al. [72], who also examined the effect of the passing side and found that participants reported equal comfort levels for both sides. Nevertheless, Moussaïd et al. [138] conducted a set of controlled experiments and observed pedestrians' preference to perform evasive manoeuvres to the right, while passing each other.

2.5.5 Yielding a way to a human at crossing (Req. 2.4.5)

Moller et al. [9] posed the problem of who goes first at an impasse as one of the social conventions that are "less well-defined". As stated in a survey by Mirsky et al. [28], the term "social navigation" usually refers to a human-centric perspective; therefore, the robot is often obliged to yield a way to a human at the crossing.

The user study performed by Lichtenthäler et al. [86] showed that at the crossing scenario, the participants favoured the navigation method in which the robot stopped to let a person pass. Yielding a way to a human based on the predicted motion was also investigated in [77].

2.5.6 Standing in line (Req. 2.4.6)

Standing in line while forming a queue is one of the most common collective behaviours of humans. Nakauchi and Simmons [139] modelled how people stand in line by first collecting empirical data on that matter. Further, they utilised these data to model a range of behaviours for a robot tasked to get into a queue, wait and advance in the queue alongside other individuals awaiting service.

2.5.7 Obeying elevator etiquette (Req. 2.4.7)

"Elevator etiquette" refers to the customary rules of humans entering and exiting a bounded space through a doorway, specifically, letting people leave an elevator before attempting to enter. These rules are generalisable to numerous closed areas like rooms and corridors.

Gallo et al. [140] proposed the machine-like approach for the design of robot behaviour policies that effectively accomplish tasks in an indoor elevator-sharing scenario without being disruptive. Alternatively, Lin et al. [119] discussed the social appropriateness of lining up for an elevator in the context of deploying a mobile remote presence. Elevator-related conventions were tackled in a robotic competition – "take the elevator challenge" [8].

2.6 Summary

In this chapter, social robot navigation requirements were grounded based on the reviewed user studies regarding unfocused and focused human-robot interactions. This, in turn, highlighted objectives on how robots should behave in populated environments. The human-aware robot navigation requirements are organised into our taxonomy consisting of requirements for ensuring the physical and perceived safety of humans, as well as the requirements assuring the robot's motion naturalness and the robot's compliance with the social norms. This classification is the basis for the analysis of algorithmic topics (Chapter 3).

We acknowledge that the proposed set of primitive requirements is subject to extension as the social navigation studies advance and new issues or additional cases are found [11]. Not only some requirements mentioned above have not been sufficiently studied, but many other human conventions have not been considered at all in user studies with robots; hence, there are no clear guidelines on how they can be tackled properly in social robot navigation. As a consequence, the comprehensive method for assessing compliance with social norms remains unresolved, in contrast to the agreement on criteria for evaluating the physical and perceived safety, as well as most cases covered by naturalness aspects.

An example phenomenon that was not targeted by user studies to the extent that allows establishing specific principles is facial expressions. Petrak et al. [83] discussed a side note of their study that enhanced robot facial expressions and gestures could make the behaviour easier to anticipate for the experiment participants. Kruse et al. [12] pointed out additional navigation conventions such as: giving priority to elderly people at doorways, asking for permission to pass, and excusing oneself when one has to traverse a personal zone to reach a goal. Furthermore, Gao and Huang [10] indicated observing right-of-way at four-way intersections as another navigation-based interaction. On the other hand, despite overtaking on the non-dominant side has been implemented in some navigation methods [71, 141], there are no clear guidelines that such behaviour is common in environments other than narrow passages.

Nevertheless, implementing all requirements in a single robot control system is an enormous challenge, while integrating all constraints and norms requires rich contextual awareness of the robot.

Chapter 3

Related work

Our literature review can be segmented into two perspectives: *requirements* and *algorithmic*. The *requirements* perspective, explained in Chapter 2, involves exploring various user studies to identify the rules for social robots to adhere to. The primary focus of that part lies in examining factors that cause human discomfort, as confirmed in real-world experiments involving human participants. In addition to identifying these factors, we aim to extract methods for mitigating discomfort to obtain implementable guidelines for robot control systems.

Subsequently, the *algorithmic* perspective, discussed in this chapter, categorises existing research regarding the perception, motion planning, and evaluation approaches (Fig. 3.1) and maps state-of-the-art navigation methods onto the specified requirements taxonomy (Fig. 2.2).

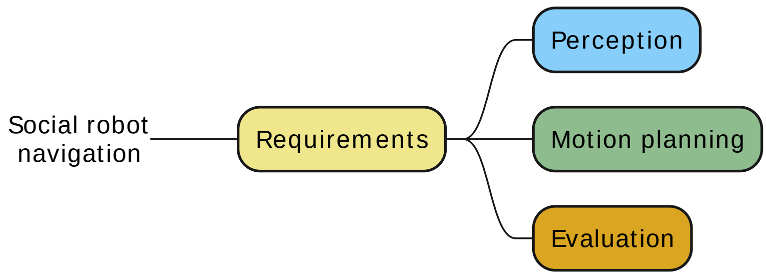


Figure 3.1: A taxonomy of main concepts in social robot navigation.

The following sections give an *algorithmic* overview of fundamental aspects of social robot navigation. The Sec. 3.1 discusses the key methods for addressing the main challenges of social robot perception, namely the detection and tracking of humans in the robot's environment. These considerations are complemented by the analysis of diverse environment representations and contextual awareness of robots. Then, in Sec. 3.2, which is the major part of this chapter, various methods employed for robot motion planning are discussed. The review involves both traditional methods and dedicated socially-aware approaches that take into account constraints arising from the presence of surrounding humans. Sec. 3.2 is summarised by tables mapping the state-of-the-art navigation algorithms onto the requirements taxonomy, based on the objectives addressed in each approach. Moreover, Sec. 3.3 explores the methods for evaluating social robot navigation as well as study types and tools relevant to the development of navigation techniques. The summary of the analyses from this chapter is provided in Sec. 3.4.

3.1 Perception

Robot perception plays a substantial role in safe navigation and expands the intelligence of a robot. Social robots must differentiate obstacles from humans to interact in a discomfort-mitigating manner.

In robotics, various types of exteroreceptors [31] are utilised to perceive the environment. Tactile sensors provide feedback about physical contact, enabling robots to detect and respond to touch [53, 61, 62, 142, 143]. They are crucial for tasks requiring object recognition that other sensor types can't capture. Sonar sensors utilise sound waves to detect the presence, distance, and velocity of objects, allowing robots to navigate and avoid obstacles in dynamic environments [144, 52, 53, 145, 146]. Laser range finders use laser beams to measure distances accurately, aiding in mapping and localisation tasks [61, 147, 148, 149, 150, 151, 152]. *RGB* cameras capture images in visible light, enabling robots to recognise objects, navigate environments, and interpret visual cues [53, 42, 153]. Finally, *RGB-D* cameras, equipped with depth sensors, provide both colour and depth information, enhancing object detection and enabling 3D mapping [154, 149, 155, 156]. These sensor types play essential roles in robotics research and development, enabling robots to perceive and interact with their surroundings effectively.

The remainder of this section follows the taxonomy illustrated in Fig. 3.2.

3.1.1 Environment representation

Besides detecting obstacles and tracking humans, robot perception is usually employed to collect subsequent observations of the surroundings to create an environment model, among which the most popular are dense, sparse, and dual representations.

A dense representation constitutes a discretised map of the robot environment. Classical maps contain all types of obstacles embedded into the environment model without a semantic distinction. The most common planar map types are occupancy grids [157] and costmaps [22]. In contrast, octomaps [158], representing occupancies in 3D space, and elevation grid maps [159] are less frequently integrated with social robot navigation systems.

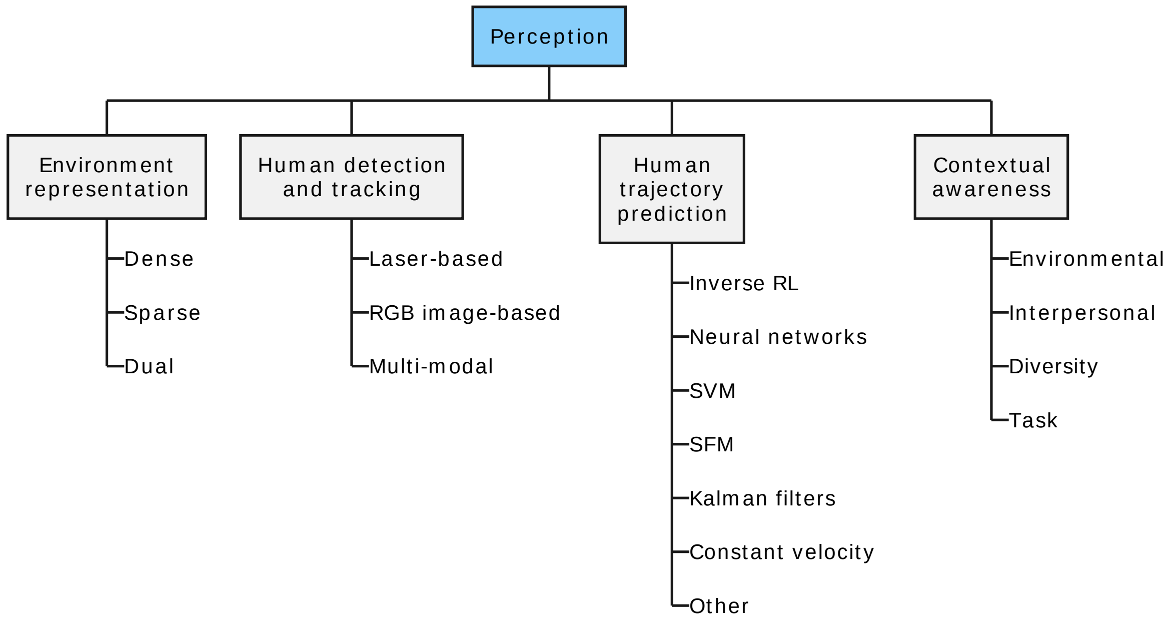


Figure 3.2: A taxonomy of perception for social robot navigation.

The pioneering dense model is an occupancy grid [157] that represents the environment as a binary grid (graph) where each cell is either occupied or free, and all occupied cells are treated as equal obstacles. Therefore, costmaps were proposed to extend the classical occupancy grids. Costmaps introduce intermediate states (between free and occupied) of a cell [22] and constitute a 2D traversability grid in which cells are given a cost of traversal reflecting the difficulty of navigating the respective area of the environment [160]. This allows robots to plan paths that optimise not just for avoiding collisions but also for factors like proxemics. The dense representation of an environment is often solely used in classical robot navigation approaches [161, 147, 158].

Sparse environment representations typically refer to representations where only certain key features or landmarks are represented explicitly, with the rest of the space left unstructured or minimally represented. Sparse representation usually provides a concise description of the objects detected in the environment, constituting their semantic information with geometric attributes [162, 163, 164, 43]. This method of storing environment objects also allows, e.g., applying linear algebra formulas to easily predict objects' motion.

Dual environment representations, combining dense and sparse ones, are commonly used in social robot navigation [165, 166, 167, 168]. While obstacle-filled costmaps are calculated, robot perception modules simultaneously detect and track humans in the environment. They provide sparse data about each human, e.g., a pose and velocity, or even spatial relationships [149, 169]. Such information allows for dynamic modelling of personal spaces of individuals (**Req. 2.2.1**) and O-spaces of F-formations (**Req. 2.2.2**), which can later be embedded onto layered costmaps [170]. Layered costmaps extend the notion of traditional costmaps to facilitate separate representations of different contextual cues as spatial constraints in the robot environment. The resultant costmap with enriched information is flattened for motion planning; therefore, classical algorithms can still be used.

3.1.2 Human detection and tracking

Social robot navigation encompasses the awareness of humans surrounding the robot, as they must be treated differently from typical obstacles. The awareness arises from detecting and tracking people by the robot perception system [125] as well as exhibiting behaviour that mitigates the discomfort of nearby humans (**Req. 2**). Various methods for human detection and tracking have been proposed in the literature [171, 172, 149, 173, 174, 175, 176].

Arras et al. [171] proposed a method utilising a supervised learning technique for creating a classifier for people detection. Specifically, *AdaBoost* was applied to train a classifier from simple features of groups of neighbouring beams corresponding to legs in the *LiDAR*'s range data. Similarly, Bozorgi et al. [176] focused on *LiDAR* data filtering to obtain robust human tracking in cluttered and populated environments. They integrated Hall's proxemics model [47] with the global nearest neighbour to improve the accuracy of scan-to-track data association of leg detection. Results of their experiments show that their method outperformed the state-of-the-art detector from [172].

In contrast, Linder et al. [149] proposed a multi-modal (*LiDAR* and *RGB-D*) people-tracking framework for mobile platforms in crowded environments. Their pipeline comprises different detection methods, multi-sensor fusion, tracking and filtering. Triebel et al. [169] extended multi-hypothesis tracker from [177] for detecting F-formation arrangements. Both works were integrated and implemented in the *SPENCER* robot [149, 169].

Redmon et al. [173] framed the object detection problem as a regression problem to spatially separated bounding boxes and associated class probabilities. They proposed a generic framework for detecting objects of various classes on 2D images. Alternatively, Cao et al. [175] proposed an *Open-Pose* system for human skeleton pose estimation from *RGB* images. In another work, Juel et al. [178] presented a multi-object tracking system that can be adapted to work with any detector and utilise streams from multiple cameras. They implemented a procedure of projecting *RGB-D*-based detections to the robot's base frame that are later transformed to the global frame using a localisation algorithm.

Theodoridou et al. [153] used *TinySSD* [174] for human detection in their robot with limited computational resources. *TinySSD* is a lightweight single-shot detection deep convolutional neural network for real-time object detection, which only finds people in the images; hence, the authors of [153] had to perform image and range-based data matching in their system.

In real-world studies, robot sensors are used to detect and track humans. The survey by Möller et al. [9] discusses, i.a., the active perception idea. The authors denoted that active vision systems can influence the input by controlling the camera. As an extension of active perception, they depict active learning [179], which also influences the input data, but during the training process. This enables the agent to intelligently choose what data points to exploit next.

To the best of our knowledge, currently, the most comprehensive human perception stack is *SPENCER* [149, 169], which is available as the open-source software[1] compatible with the *Robot Operating System* (*ROS*) [180, 181].

3.1.3 Human trajectory prediction

In social navigation, classical planning methods, e.g., *Artificial Potential Field* (*APF*) [101] or *DWA* [144] often exhibit limited efficacy as pedestrians are treated merely as uncooperative obstacles. This limitation is exemplified by the freezing robot problem [182], where a mobile robot may become immobilised in a narrow corridor when confronted with a crowd of people unless it can anticipate the collective collision avoidance actions [183]. Therefore, predicting human trajectories is one of the fundamental concepts in social robot navigation, in particular in unfocused human-robot interactions, where explicit communication between agents is not present. Understanding how agents move can reduce the potential for conflicts, i.e., sudden encounters in which humans and robots might collide (**Req. 2.1**) [28, 184]. Another particularly important aspect is that humans frequently undergo lengthy occlusion events; hence, their motion prediction prevents possible unexpected encounters.

In the social robot navigation literature, the prevailing method is the *Inverse Reinforcement Learning* (*IRL*) [185], which is based on the *Markov Decision Process* (*MDP*) [186]. The *IRL* identifies reward functions based on the observed behaviour, enabling robots to learn from human demonstrations. It can be classified as an offline inference and learning method [28]. Henry et al. [187] used *IRL* to learn human motion patterns in simulation to use them later for socially-aware motion planning. Rhinehart et al. [188] extended *IRL* for the task of continuously learning human behaviour models with first-person-view camera images. Their *Darko* algorithm jointly discovers states, transitions, goals, and the reward function of the underlying *MDP* model. In another work, Vasquez et al. [189] conducted experiments to compare the performance of different *IRL* approaches, namely, *Max-margin IRL* [190] and *Maximum Entropy IRL* [191], which were later applied

[1] `https://github.com/spencer-project/spencer_people_tracking`

for robot navigation in a densely populated environment. Also, Kretzschmar et al. [192] used *Maximum Entropy IRL* to deduce the parameters of the human motion model that imitates the learned behaviours. *IRL* seeks to extract the latent reward or cost function from expert demonstrations by considering the underlying *MDP*. It learns from entire trajectories, and its computational expense arises from running *RL* in an inner loop [193]. Another approach was proposed by Goldhammer et al. [194], who used an *Artificial Neural Network* (*ANN*) with the multilayer perceptron architecture to learn usual human motion patterns. A different method was presented by Gao et al. [195], who trained a *Reinforced Encoder-Decoder* network to predict possible activities.

Alternatively, *Long Short-Term Memory* (*LSTM*) networks are one of the sequential methods that learn conditional models over time and recursively apply learned transition functions for inference [196]. Unlike standard feed-forward neural networks, also known as recurrent neural networks, these networks include feedback connections. Following the work by Alahi et al. [197], who presented a human trajectory forecasting model based on *LSTM* networks, they have become widely popular for this purpose. For example, Furnari and Farinella [198] utilised *LSTM* to predict future human actions in a domestic setting. Chen et al. [199] also created an *LSTM*-based model predicting socially-aware trajectories learned from a dataset to later integrate this into a robot motion planning scheme. *Recurrent Neural Networks* (*RNN*) were also applied for sequence learning, e.g., by Vemula et al. [200] who proposed the *Social Attention* trajectory prediction model that captures the relative importance of each person when navigating in the crowd, irrespective of their proximity. Another work by Farha et al. [201] relies on training a *Convolutional Neural Network* (*CNN*) and a *RNN* to learn future sequences. They proved their method to be suited for long-term predictions of video sequences.

Another effective data-based method for learning from demonstrations is *Generative Adversarial Imitation Learning* (*GAIL*), applied by, e.g., Tai et al. [193] to learn continuous actions and desired force toward the target. Huang et al. [202] proposed a model-based interactive imitation framework combining the advantages of *GAIL*, interactive *RL* and model-based *RL*.

On the other hand, Kanda et al. [203] used the *Support Vector Machine* (*SVM*) to classify 2-second recordings of human trajectories in a shopping mall into four behaviour classes: fast-walking, idle-walking, wandering, and stopping. The classification relies on features of trajectory shapes and velocity. Coarse classification enables forecasting human trajectories [4]. Similarly, Xiao et al. [204] first pretrained the *SVM* to group activity classes, then predicted the trajectories based on those classes, and finally evaluated the system in a lab environment.

Alternatively, the *Social Force Model* (*SFM*) [1] with its numerous modifications [205, 165, 167] is also a popular method for human trajectory prediction; however, requires knowledge about environmental cues to infer the possible goals of humans. Luber et al. [206] combined *SFM* with a tracker based on the Kalman filter to produce a more realistic prediction model of human motion under the constant velocity assumption. Recently, multiple approaches integrating *SFM* into neural network schemes were proposed. For example, Yue et al. [207] integrated *SFM* and a deep neural network in their *Neural Social Physics* model with learnable parameters. Gil and Sanfeliu [208] presented *Social Force Generative Adversarial Network* (*SoFGAN*) that uses a *GAN* and *SFM* to generate different plausible people trajectories reducing collisions in a scene.

Numerous works across various application domains depend on kinematic models for their simplicity and satisfactory performance, particularly in scenarios with minimal motion uncertainty and short prediction horizons. Among others, Elnagar [209] proposed a method predicting future poses of dynamic obstacles using a Kalman filter under the assumption of using a constant acceleration model. Similarly, Lin et al. [210] proposed a forecasting strategy that employs a bimodal extended Kalman filter to capture the dual nature of pedestrian behaviour – either moving or remaining stationary. Also, Kim et al. [211] used a combination of ensemble Kalman filters and a maximum-likelihood estimation algorithm for human trajectory prediction.

In applications where performance is crucial, the constant velocity model, assuming piecewise constant velocity with white noise acceleration, can be applied. Despite its simplicity, it is commonly chosen as an ad-hoc method for motion prediction in numerous approaches [212, 213, 214, 215, 148, 216, 217] having lightweight and straightforward implementation and yielding satisfactory results with high-frequency updates. Recently, Schöller et al. [218] discussed that the constant velocity model might outperform state-of-the-art neural methods in some scenarios.

Diverse methods were also evaluated for usage in human trajectory prediction; for example, belief distribution maps [219] that consider the obstacle situation in the robot's environment, *multi-goal Interacting Gaussian Processes* (*mgIGP*) [220] that can reason multiple goals of a human for cooperative navigation in dense crowds, or *Human Motion Behaviour Model* (*HMBM*) [221] allowing the robot to perform human-like decisions in various scenarios. Another method was proposed by Ferrer and Sanfeliu [222], who presented a geometric-based long-term *Bayesian Human Motion Intentionality Predictor* using a naive Bayes classifier that only requires training to obtain the set of salient destinations that configure a scene.

Our survey discusses the most common methods used in robotic applications, but various other methods for human trajectory prediction have evolved over the years. Rudenko et al. [196] presented a thorough review of the state-of-the-art human motion prediction methods, where they also discussed approaches that account for map information or environmental cues for predictions. An appropriate forecasting method has to be selected for a specific application based on multiple criteria, e.g., computational resources, prediction horizon, and detection uncertainty.

3.1.4 Contextual awareness

A robot is perceived as intelligent if it utilises the contextual information in its imperative [13, 26]. The proper socially-aware activity of a robot performing a single task might differ depending on the situation defined by a contextual arrangement. It is connected to adjusting the robot's behaviour, knowing what environment it is in (gallery or shopping mall), what task it performs (transporting a glass full of hot tea or packed goods), whom it interacts with (young person or elderly), and what social norms are expected in the environment (may differ between cultures).

Francis et al. [11] in their survey identified the following forms of context: cultural context [47, 95, 223, 41, 224, 225], environmental context, individuals diversity, task context, and interpersonal context, but their literature review in this area is narrow. The notion of context is usually regarded in the deliberative layer of the robot's planning and embedded as spatial or spatiotemporal constraints in the motion planning [226, 227, 17].

Environmental context The environmental context is constituted by various characteristics of the robot's surroundings. This information is particularly important for robots that act in different types of rooms, e.g., corridors and libraries of the university. While the robot might be sociable and lively in corridors, it is not necessarily appropriate to distract students in the library, where the robot should move slowly and be quiet. Therefore, researchers investigate different environmental concepts to embed them into robot navigation schemes.

Banisetty et al. [228] proposed a model-based context classifier integrated with a high-level decision-making system for socially-aware navigation. Their *CNN* model distinguishes between different environmental contexts such as an art gallery, hallway, vending machine, and others. Additionally, based on the *LiDAR* observations and using the *SVM*, they classified social contexts, namely people forming a queue and F-formations. In continuation of this article, Salek Shahrezaie et al. [229] introduced classification and detection information into a knowledge base they queried to extract applicable social rules associated with the context at hand. This approach has been further extended in [151] for using environmental context, object information, and more realistic interaction rules for complex social spaces. On the other hand, Jia et al. [230] proposed a deep-learning-based method for detecting hazardous objects in the environment of an autonomous cleaning robot to maintain safe distances from them on the motion planning level. Recognising human activity spaces is a part of environmental context awareness, as presented in the work by Vega et al. [231], who exploited the detection of specific objects for this purpose.

A leading approach to enable the robot's contextual awareness is semantic mapping [232, 233, 234]. For example, Zhang et al. [235] used an object semantic grid map along with a topological map for the automatic selection of roughly defined navigation goals in a multiroom scenario. Alternatively, Núñez et al. [236] proposed a navigation paradigm where the semantic knowledge of the robot's surroundings and different social rules are used in conjunction with the geometric representation of the environment's semantic solutions. Their approach aims to integrate semantic knowledge and geometrical information. A promising method for the interactive building of semantic maps for robot navigation has been illustrated in [237].

Interpersonal context Interpersonal cues are mainly related to social relationships between tracked humans in the robot environment. This knowledge can be embedded in control systems to enhance robot navigation skills. For example, Li et al. [238] proposed a dual-glance *CNN*-based model for visual recognition of social relationships. The first glance fixates on the person of interest, and the second glance deploys an attention mechanism to exploit contextual cues. Lu et al. [170] proposed an approach for context-sensitive navigation, mainly focusing on human-aware robot navigation and embedded spatial constraints into environment models in the form of costmaps.

The algorithm by Luber and Arras [177] was extended in [169] for detecting and learning socio-spatial relations, which are used for creating a social network graph to track groups of humans. Patompak et al. [239] developed a *Reinforcement Learning* method of estimating a social interaction model for assisting the navigation algorithm regarding social relations between humans in the robot's environment model. Similarly, Okal and Arras [240] employed *Bayesian Inverse Reinforcement Learning* for learning the cost function of traversing in the area with a group of humans.

Haarslev et al. [241] introduced contextual information into robot motion planning, namely, F-formation spatial constraints in the costmaps used for planning. The F-formation arrangement is inferred from participants' speed, line of sight and potential focus points. Similarly, Schwörer et al. [242] detected people and their interactions to create spatial constraints in the environment model used for motion planning.

Diversity context Diversity-related contexts facilitate leveraging human diversity in social robot navigation. Researchers presented multiple studies regarding gender [243, 244, 245], age [243, 246, 244] personality [145, 247], and diverse human groups representations [248]. All these traits affect how people interact with and perceive robots. Furthermore, Bera et al. [41] attempted to classify the personality of each pedestrian in the crowd to differentiate the sizes of personal spaces of individuals. Subsequently, the emotional state of the pedestrians was also inferred and embedded for socially-aware navigation [249, 250, 42].

Task context The robot's behaviour differs based on the task to perform. If the robot is delegated to execute a task of a high priority, e.g., urgent transportation in a hospital, it will interact with humans only in an unfocused manner committing to collision avoidance and respecting personal spaces. However, if the robot's task is to start sociably interacting with customers in a shopping mall to present products to them, it has to mildly start focused interactions with pedestrians. Therefore, the objectives of robot navigation differ between tasks, affecting the socially correct behaviour scheme that should be followed.

Popular tasks delegated to social and assistive robots are transportation [89], guiding [169, 251], or accompanying [252, 166]. For example, accompanying objectives differ even between the tasks of attending individuals [253, 252] and groups [254, 166] or even between different strategies for accompanying individuals (Sec. 2.5.1). Similarly, a guiding robot, e.g., proposed in [251], mainly focuses on leader-follower tasks, but once it finishes the guided tour, it may drop the constraints specific to the guiding behaviour (speed etc.) and switch to socially-aware collision avoidance and back to the reception area.

A significant challenge lies in integrating the contradictory objectives of treating humans as social obstacles during tasks requiring only unfocused interactions and regarding them as interaction partners when needed. As a result, methods introducing human awareness and social acceptance must be carefully selected to avoid interfering with contradictory modes of operation, as some constraints may need to be disabled in focused interaction mode while enabled in unfocused interaction mode [30].

3.2 Motion planning

This section discusses various motion planning approaches and methods of incorporating social awareness into robot control systems. The motion planning module is crucial for safely guiding the robot through dynamic environments. Motion planning for mobile robots is understood as a pose control scheme aimed at moving the robot from its initial pose to the target pose while considering the kinematic and dynamic (kinodynamic) constraints of the mobile base.

From the perspective of motion planning, requirements for social awareness presented in Chapter 2 might entail the necessity of specific enhancements compared to classical robot navigation. Namely, those can be classified into three specific groups. Firstly, modifications of the intermediate trajectory to the fixed goal. This might involve adjustments originating from respecting personal spaces (**Req. 2.2.1**), O-spaces of F-formations (**Req. 2.2.2**), and modulating speed (**Req. 2.2.3**) to mitigate the discomfort of surrounding humans. Secondly, the extended selection of the final poses for navigation tasks with coarsely defined goals. In particular, selecting such a pose that, e.g., does not block any affordance space (**Req. 2.4.2**), minimises the discomfort of the approach to a human (**Req. 2.2.5.1**), or provides joining a queue in a socially compliant manner (**Req. 2.4.6**). Thirdly, dynamically inferring and following virtual goals in real time depending on the poses of cooperating humans, which enables efficient execution of accompanying tasks (**Req. 2.4.1**).

The predominant motion planning architecture for mobile robots relies on hierarchical planning with two asynchronously running modules, specifically, a global path planner and a local trajectory planner [255, 147]. Global path planning involves finding a feasible path from a start configuration to a goal configuration while avoiding environmental obstacles. Algorithms generating global paths typically operate in a configuration space and consider the entire environment [256]. In contrast, local trajectory planning aims to generate trajectories for the robot to follow within a short time horizon that navigate the robot safely and efficiently through the environment while reacting to dynamic obstacles and perturbations. Algorithms producing local trajectories typically operate in the robot's control space or velocity space and consider immediate sensor feedback and environmental information [161, 147]. Usually, local trajectory planners operate at a higher frequency than global path planners to adjust the robot's motion in real-time, accounting for dynamic changes in the environment and ensuring safe and efficient navigation.

Our taxonomy of the algorithmic perspective of social robot navigation follows the hierarchical motion planning scheme, differentiating approaches for global path planning and local trajectory planning (Fig. 3.3).

Numerous surveys regarding social robot navigation thoroughly discussed motion planning [12, 2, 14]. However, our review aims not only to investigate the variety of methods of implementing human awareness in robot control systems but also to classify those approaches according to the requirements they fulfil. The classification of requirements regarded in objectives of different navigation algorithms is presented in Sec. 3.2.3.

3.2.1 Global path planning

In the context of global path planning for social navigation for surface robots, various methodologies are employed for the research. Recently, multiple surveys regarding path planning for mobile robots have been proposed [257, 258, 259, 260, 261]. State-of-the-

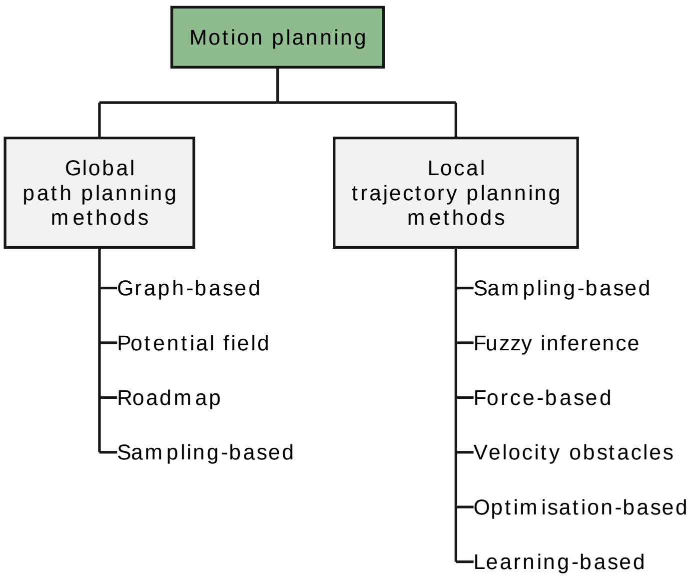


Figure 3.3: A taxonomy of motion planning for social robot navigation.

art techniques can be classified into distinct groups. These include graph-based methods, potential field methods, roadmap methods, and sampling-based methods. Each class of approaches offers unique advantages and challenges, contributing to the broader landscape of mobile robot path planning [262].

Although in classical path planning metaheuristic methods like genetic algorithms or particle swarm optimisation are commonly discussed [263], to the best of our knowledge, they were not applied for human-aware navigation.

Graph-based methods Graph-based methods for path finding fall into the category of approximate cell decomposition approach in which cells of predefined shape (usually rectangles) do not exactly cover the free space (in contrast to exact cell decomposition) but the cell connectivity in a graph is encoded [264].

Algorithms The earliest graph (or grid) search methods in the context of computer science and algorithmic development can be traced back to the 1950s. One significant development was Dijkstra's algorithm [265], which laid the foundation for many subsequent graph search and pathfinding algorithms. This algorithm was primarily focused on finding the shortest path in a graph. Later, Hart et al. [266] presented the A^* algorithm, which builds upon Dijkstra's algorithm by incorporating heuristic information to guide the search more efficiently, making it particularly useful for pathfinding in large graphs. The heuristic utilises the distance between the current processing node and the goal node on the solution space. Globally shortest paths are obtained using both heuristic estimates and actual costs in a weighted graph. Other variants of the A^* planning algorithm include *D** [267], *Focused D** [268], *LPA** [269], *D* Lite* [270], *E** [271], *Field D** [160], and *Theta** [272]. A brief description of each variant is depicted below.

Graph-based planners usually require replanning if the underlying environment model changes. This drawback is addressed by the *D** [267], which is an incremental search algorithm for finding the shortest paths designated particularly for graphs that may dynamically change once the search begins as it possesses the procedure for updating paths if changes occur. *Focused D** [268] adapts the *D** to prioritise the exploration of areas closer to the goal. *Lifelong Planning A** (*LPA**) [269] is an incremental heuristic search algorithm that continuously improves its estimates of the shortest path while adapting to changes in the environment, providing efficient planning in dynamic environments. *D* Lite* [270] is a simplified version of the *D** algorithm, focusing on efficient replanning for real-time performance in static or partially unknown environments. The wavefront expansion procedure (known as *NF1* in [264]) is a simple global planner that expands the search to all adjacent nodes until the start node and goal node are covered. It was employed in [221] for path planning in human-populated environments. Another method is *E** [271] algorithm capable of dynamic replanning and user-configurable path cost interpolation. It calculates a navigation function as a sampling of an underlying smooth goal distance that takes into account a continuous notion of risk that can be controlled in a fine-grained manner.

The authors of *Field D** [160] addressed the problem of using discrete state transitions that constrain an agent's motion to a narrow set of possible headings, which often occurs in classical grid-based path planners. Instead, they proposed the linear interpolation approach during planning to produce paths with a continuous range of headings. Alternatively, the *Theta** [272] method propagates information along grid edges (to achieve a short runtime) but without constraining the paths to the grid edges. Instead, *any-angle* paths are found by performing line-of-sight checks between nodes. When a direct line of sight is feasible between two adjacent nodes without intersecting obstacles, *Theta** considers the straight-line path, reducing the number of nodes expanded, compared to A^*. Also, *Theta** retains the optimality guarantees of A^* while producing smoother, more natural paths, especially in environments with narrow passages or obstacles.

Notably, Dijkstra's algorithm does not account for the robot's kinodynamic constraints, which may generate paths not admissible to robots with, e.g., Ackermann kinematics. However, Dolgov et al. [273] addressed this issue in their *Hybrid A** algorithm that extends the traditional A^* to handle continuous state spaces by discretising them into a grid. It incorporates vehicle kinematic constraints, such as maximum velocity and steering angle, to generate feasible paths for vehicles navigating through complex environments. Recently, Macenski et al. [256] presented a search-based planning framework with multiple algorithm implementations, including *Cost-Aware Hybrid-A** planner that provides feasible paths using a Dubins or Reeds-Shepp motion model constrained by a minimum turning radius for Ackermann vehicles.

Human-aware constraints The classical path-finding algorithms focus on calculating the shortest, collision-free path and do not explicitly regard humans in the environment; hence, they also do not consider social constraints. However, in graph-based methods, the planning procedure is separated from the definition of planning constraints incorporated into the environment representation [215]. Hence, researchers started to modify the environment models, e.g., costmaps, to embed human-aware constraints into the motion planning scheme while employing classical path-finding algorithms. Most approaches that extend environment representations focus on introducing spatial or spatiotemporal soft constraints representing proxemics [274] or social conventions [71, 170].

For example, Sisbot et al. [274] presented a *Human Aware Motion Planner* (*HAMP*) that exploits algorithms for reasoning on humans' positions, fields of view, and postures. They integrated different social constraints into their highly configurable planning scheme, including Gaussian-modelled personal spaces or hidden zones behind obstacles (visibility constraints).

Kirby et al. [71] proposed a *Constraint-Optimising Method for Person-Acceptable NavigatION* (*COMPANION*) framework in which multiple human social conventions, such as personal spaces and tending to one side of hallways, are represented as spatial cost functions. The authors emphasised the importance of accounting for social aspects at the global path-planning level. Their extended environment representation, including human awareness constraints, is utilised by the customised A^* algorithm to produce socially acceptable global paths for robots.

Lu et al. [84] presented a costmap-based system capable of creating more efficient corridor navigation behaviours by manipulating existing navigation algorithms and introducing social cues. They extended robot environment models with socially-aware spatial constraints to navigate in a more human-friendly manner. Similarly, the authors of [85] attempted to provide socially legible robot motions using proxemics-based spatial compatibility model and directional compatibility preventing frontal collisions of human and robot. Their concepts have been integrated with *HANP* global path planning method [274].

Kollmitz et al. [215] presented a planning-based approach that uses predicted human trajectories and a social cost function to plan collision-free paths taking human comfort into account. They employed search-based, time-dependent path planning to reason about human motion over time, while simultaneously accounting for the kinematic and dynamic constraints of a robot. The authors extended the layered costmap architecture [170] proposing multiple layers, each related to the state of the robot environment in subsequent prediction step. Their framework aimed to include the spatial nature of human proxemics (Gaussian social cost model used) and temporal aspects of human motion. However, the authors noted that addressing spatiotemporal intricacies of human-aware navigation in a global path planning scheme (instead of local trajectory planning) is computationally expensive [215].

Okal et al. [240] proposed a method that uses *IRL* to learn features of a populated environment to model socially normative behaviours [189]. Once the reward function for a navigation task is obtained, it is used to define spatial costs of social normativeness that can be injected into a costmap used by a motion planner (either global or local).

Extending classical obstacle-filled costmaps with social constraints is a common and straightforward practice to include basic human awareness in robot motion behaviour. Ginés et al. [8], for example, attempted to adjust spatial cost functions representing proxemics zones by resizing the personal spaces according to the mood of a human. Some other works also embedded dynamically recalculated personal zones into costmaps to account for dynamics of individual humans [71, 275, 276, 252] or groups [277].

Potential field methods Purely graph-based planners have limitations originating from their discontinuous representation of configuration space. On the other hand, potential field methods offer smoother path generation and can be directly related to sensor data, yet they suffer from the presence of local minima [271]. Path planning utilising a potential field creates a gradient across the robot's map that directs the robot to the goal position from multiple prior positions [264].

One of the pioneering works that introduced the concept of *Artificial Potential Field* (*APF*) for obstacle avoidance and navigation in robotics is [101]. The potential field methods treat the robot as a point in the configuration space under the influence of an *APF*. The goal, acting as a minimum in this space, exerts an attractive force on the robot, while obstacles act as repulsive forces. The superposition of all forces is applied to the robot. Such an *APF* smoothly guides the robot toward the goal while simultaneously avoiding known obstacles, just as a ball would roll downhill [3].

Later, Borenstein and Koren [278] developed a *Virtual Force Field* method that relies on two basic concepts: certainty grids for obstacle representation and potential fields for navigation. Their method enables continuous motion of the robot without stopping in front of obstacles with a speed of 0.78 m/s. However, the approach was abandoned due to the method's instability and inability to pass through narrow passages [3]. The extended potential field method has been proposed by Khatib and Chatila [279] with two additions to the basic potential field, in particular, the rotation potential field and the task potential field.

More recently, Iizuka et al. [280] proposed a modified *APF* approach resistant to the local minimum issue in multi-obstacle environments, while Weerakoon et al. [281] presented a deadlock-free *APF*-based path planning algorithm. Similarly, Azzabi and Nouri [282] developed an approach that addresses the common issues of the original *APF*, namely local minima and the goal being non-reachable with obstacles nearby. Szczepanski [283] also proposed a path planning method for mobile robots that uses the attractive potential for goal reaching as the original *APF*, but the repulsive potential is replaced by a general obstacle potential, equal to repulsive potential, vortex potential, or their superposition.

Roadmap methods Roadmap strategies capture the connectivity of the robot's unobstructed space through a network of 1D curves or lines, denoted as roadmaps. Subsequently, the roadmap serves as a network of path segments for planning robot movement. Consequently, path planning is reduced to connecting the robot's initial and goal positions to the road network, followed by identifying a sequence of routes from the initial robot position to its destination [3]. The most common approaches falling into the roadmap-based category are visibility graphs and Voronoi diagrams.

The visibility graph method is one of the earliest path planning methods [264]. For a polygonal configuration space, the graph consists of edges joining all pairs of vertices that can see each other (including both the initial and goal positions as vertices as well). The unobstructed straight lines (roads) joining those vertices are the shortest distances between them, guaranteeing optimality in terms of the length of the solution path. The main caveat of the visibility graph is that the solution paths tend to move the robot as close as possible to obstacles on the way to the goal [3]. In contrast, the Voronoi diagram is an approach that maximises the distance between the robot and obstacles in the map [3].

Our research regarding the applications of classical roadmap methods shows that they are rarely used in social robot navigation as they only consider binary environment models (obstacle or free space); hence, human awareness cannot be properly tackled. However, Voronoi diagrams might be used as reference path planning approaches [284, 285, 213, 286] for capturing the skeleton of the environment along with human-aware trajectory planners as in [141].

Sampling-based methods The main idea of sampling-based motion planning is to avoid the explicit construction of obstacle regions but instead conduct a search that probes the configuration space with a sampling scheme [287]. The most prevalent methods falling into the category of sampling-based path planners are the *Probabilistic Roadmap* (*PRM*) [288] and the *Rapidly-exploring Random Trees* (*RRT*) [289], both being probabilistically complete [287].

Algorithms *PRM* [288] constructs a roadmap, a graph representation of the configuration space, by sampling random points and connecting them with collision-free paths. It focuses on building a network of feasible paths between different regions of the configuration space and is effective for multi-query scenarios or environments with complex obstacles.

RRT [289] builds a tree structure by iteratively selecting random points in the configuration space and extending the tree towards those points. It explores the configuration space rapidly and is particularly effective for high-dimensional spaces. Different variants of *RRT* has been developed including *RRT-Connect* [290], *RRT** [291] or dual tree version – *DT-RRT* [292].

Both *PRM* and *RRT* have different characteristics. *PRM* requires a two-phase process: first, constructing the roadmap offline and then querying the roadmap online to find a path between a start and goal configuration. In contrast, *RRT* performs exploration and path planning simultaneously, gradually growing towards the goal configuration during the search process. *PRM* is a well-suited method for scenarios where the environment is relatively static, and the planner has sufficient computational resources to construct the roadmap offline, while *RRT* is often favoured for real-time or dynamic environments, as it can adaptively explore the space and find feasible paths in a run-time. A notable feature of sampling-based methods is that these planners can regard the kinodynamic limits of the robot to generate feasible and safe motion plans in continuous state and action spaces.

Human-aware constraints Some works focus on including constraints related to social conventions in sampling-based path-planning schemes. For example, Svenstrup et al. [293] modified the original *RRT* for navigation in human environments assuming access to full state information. Their modifications include adding the potential model designed for moving humans, so the customised *RRT* planner plans with a potential field representation of the world. Similarly, Rios-Martinez et al. [294] proposed *Risk-RRT* for global path planning. Their algorithm includes the knowledge of the personal spaces of pedestrians and the possible interactions between the F-formation's participants. *Risk-RRT* penalises the robot's crossing through personal spaces and O-spaces of F-formations by assigning additional costs to those areas. Furthermore, Shrestha et al. [295] used *RRT* for global path planning in the environment with a stationary human. Vega et al. [231] attempted to integrate proxemics theory with their path planner incorporating *PRM* [296] and *RRT* [289] methods by defining personal spaces and activity spaces as forbidden areas for robot navigation. Alternatively, Pérez-Higueras et al. [297] developed a cost function for the *RRT*-based path planner employing *Inverse Reinforcement Learning* from demonstrations.

3.2.2 Local trajectory planning

The most common architecture for robot motion planning separates global path planning and local trajectory planning [255, 147]. This separation of concerns allows for modular and flexible robotic systems, where different strategies can be applied at each level of abstraction to address specific requirements.

Local trajectory planners generate trajectories for the robot to follow within a short time horizon. Short time horizons allow operating with a higher frequency to instantly react to environmental changes and possible encounters. Trajectory planners operate in the robot's control space or velocity space and regard not only spatial aspects of motion planning but also temporal ones. In the following part of this survey, various trajectory planning methods and approaches to incorporating human awareness into robot behaviour are reviewed.

Sampling-based methods Besides global path planning (Sec. 3.2.1), sampling-based methods can also be applied to local trajectory planning. An extended *RRT* with a notion of time included – spatiotemporal *RRT*, was proposed by Sakahara et al. [213]. Their method integrates ideas of the *RRT* and the Voronoi diagram. Although motion prediction of dynamic objects is regarded, they do not explicitly capture social conventions. Nishitani et al. [214] extended this approach presenting a human-centered X–Y–T space motion planning method. The authors included human personal space and directional area as well as the robot's dynamic constraints in the planning scheme.

Pérez-Higueras et al. pointed out in [298] the future work perspective of using *RRT* as a local trajectory planner due to real-time capability, but their further work leaned towards learning-based approaches.

Fuzzy inference methods Fuzzy inference systems (*FIS*) form another well-established paradigm for control systems, specifically useful to model imprecise or non-numerical information and decisions. *FIS* are applied for traditional robot navigation [299, 300, 301, 302, 303] and social robot navigation tasks [304, 305, 306, 307]. They can also be integrated with other approaches, e.g., *Q-learning* [308] or *Reinforcement Learning* [309].

An example of *FIS* method adapted for human-aware robot navigation is the work by Palm et al. [304], who derived fuzzy control rules for the robot's actions based on expected human movements relative to the robot. They investigated the movement of humans in a shared space with a robot to determine lane preference and agent classification for collision avoidance. Another method was proposed by Obo and Yasuda [305], who developed a framework for robot navigation in crowds employing multi-objective behaviour coordination for collision avoidance. Rifqi et al. [306] used *FIS* to dynamically change parameters of the *SFM*, which has been applied for controlling the movement of a healthcare robot. Rules that they designed switch the robot's motion behaviour based on its distance to human proxemics zones. Recently, Sampathkumar et al. [307] proposed a framework integrating an *Artificial Potential Field* and *FIS* for navigation that prioritises safety and human comfort.

Force-based methods Force-based approaches model the motion of individuals (humans or robots) in the environment considering the forces acting on them. These include a force attracting the agent to the goal and forces arising from interactions with other agents and environment objects such as obstacles. Typically, they are purely reactive methods that decide the next movement based on the environment arrangement at hand, i.e., obstacles and human locations. The resultant force can be directly transformed into a velocity command for a robot. The predominant methodologies within this category are *Elastic Bands* [310] and *Social Force Model* [1].

Elastic Bands [310] is a method that aims to close the gap between global path planning and reactive control, as it performs local path deformation based on internal and external forces. Internal forces contract the path, favouring the shortest path to the goal, while external forces repel the path from obstacles. The authors of the algorithm proposed a reference implementation based on bubbles that represent discrete path points and free space. Later, this method was extended by Brock et al. [311] mainly for motion generation in manipulation tasks performed in human environments. More recently, a socially-aware specialisation focusing on improving motion legibility of the *Elastic Bands* local trajectory planner has been developed for the *SPENCER* project [169]. The notion of human awareness has also been implemented into the *Elastic Bands* approach by Vega et al. [231].

On the other hand, *Social Force Model* (*SFM*) [1] has been one of the prevalent methods for crowd behaviour simulation [312, 313], human trajectory prediction (Sec. 3.1.3), and human-like motion generation in robotics. It constitutes a model inspired by fluid dynamics that illustrates an agent's motion using a set of attractive and repulsive forces. Its flexible formulation allows for capturing additional models of social phenomena to obtain more realistic motion behaviours. Therefore, the original approach has undergone multiple extensions and over the years numerous successful real-world robotic applications have emerged [7, 314, 315, 165, 253, 166, 167].

Researchers expanded the basic *SFM* with explicit collision prediction [316, 205], making the behaviour more proactive and anticipatory. Kivrak et al. [167] also introduced collision prediction into *SFM*-based model which they integrated with a robot operating in an unknown environment with no a priori map. Similarly, Shiomi et al. [7] evaluated *SFM* with collision prediction [205] in a real-world shopping mall. Collective motion conventions were also integrated into the model formulation [317] as well as group formations [73, 318, 319]. Some works also focused on improving the realism of generated trajectories [320].

Truong and Ngo [314] proposed a proactive social motion model for safe and socially-aware navigation in crowded environments. Their formulation takes into account the socio-spatiotemporal characteristics of humans, including human body pose, field of view, hand poses, and social interactions, which consist of human-object interaction and human group interaction.

Ferrer et al. [315] presented another model extending the original formulation to effectively accompany a person. They implemented human behaviour prediction to estimate the destination of the person the robot is walking with. Additionally, the authors exploited the parameterisation of the *SFM* and applied a method of interactively learning the parameters of their Extended Social Force Model (*ESFM*) using multimodal human feedback.

Moreover, Repiso et al. presented studies regarding the robot accompanying single humans [253] and human groups [166]. In [253], they implemented three stages of focused interaction between the robot and a human: accompanying, approaching, and positioning. They inferred the human's final destination (among all destinations marked in the environment beforehand) and predicted the human motion with the *SFM*. The *SFM* was also employed for the robot's local trajectory planning, and spatial cost functions were used for trajectory scoring. In the following work, Repiso et al. [166] proposed an extended method that allows the robot to break the ideal side-by-side formation to avoid other people and obstacles, implementing the human-aware robot navigation strategy for accompanying groups of multiple humans.

Alternatively, Ferrer and Sanfeliu [165] developed a *SFM*-based *Anticipative Kinodynamic Planning* method for unfocused interactions between a robot and humans. They implemented a scalarised multi-objective cost function to choose the best trajectory amid the generated ones. On the other hand, We et al. [321] proposed a pedestrian's heterogeneity-based social force model that captures the physiology and psychology attributes of pedestrians introducing physique and mentality coefficients into the *SFM*. Recently, *SFM* has also been involved in approaches integrating machine learning techniques with motion models [322, 208].

Velocity obstacles methods The *Velocity Obstacle* (*VO*) [323] concept is a foundation for a broad class of proactive methods for a robot's local navigation. *VO* methods are based on a persistent effort to keep a robot collision-free, requiring only: a radius, a position, and a speed of each robot [324]. They generate avoidance manoeuvres by selecting the robot velocities outside the collision cone, which consists of velocities that in the future would result in close encounters with obstacles moving at known velocities. A practical application of *VO* was introduced by Lin et al. [325]. They adapted the concept by assuming that each agent is a decision-making entity capable of selecting the appropriate velocity that responds to the other agents' movements and replanning its path. Moreover, an extension of *VO*, called *Reciprocal Velocity Obstacle* (*RVO*), was developed by van den Berg et al. [326]. They exploited the fact that humans in the environment cooperate [327] and the approach guarantees to generate safe and oscillation-free motions under an assumption that all dynamic agents make a similar collision-avoidance reasoning [14]. Furthermore, a related, reactive and rule-based method called *Optimal Reciprocal Collision Avoidance* (*ORCA*) [328] does not require implicit communication between agents and optimises global objectives when finding collision-free velocities. However, agents' observations must be accurate, therefore this approach is predominantly used in the simulation.

VO-based methods are rarely enhanced with socially-aware concepts. Martinez-Baselga et al. [152] presented a *Strategy-based Dynamic Object Velocity Space* trajectory planner that explicitly regards the presence of dynamic obstacles but does not take any social conventions into account. Similarly, Zhang et al. [148] proposed a local trajectory planning scheme using *ORCA* that includes uncertainties of states of surrounding humans when selecting collision-free velocities.

Optimisation-based methods Another class of approaches for human-aware trajectory planning formulates the problem as an optimisation task, which relies on finding control inputs that optimise (minimise or maximise) an objective function while satisfying kinodynamic and collision-free motion constraints. Those hard constraints, inherited from classical robot navigation, restrict control inputs to those feasible for the specific mobile base at a given time and ensure the absence of collisions within the prediction horizon. The presence of collisions with the surrounding objects is assessed using the environment model and forward simulation of applying the computed controls. In contrast, soft constraints are embedded in the optimised objective function that takes into account, e.g., intrusions into the personal spaces of humans.

Most state-of-the-art methods planning optimal socially-aware local trajectories extend the classical robot navigation algorithms – the *Dynamic Window Approach* [144] and the *Timed Elastic Bands* [162], referred to as *DWA* and *TEB*, respectively.

***DWA*-based methods** The *DWA* is one of the most common algorithms for collision avoidance. The main characteristic of the approach is that commands, controlling the translational and rotational velocities of the robot, are searched directly in the space of velocities. The search space is reduced to velocity pairs fulfilling kinodynamic constraints. Typically, for each velocity pair, the effect of applying those controls to the robot is simulated over a short time horizon, e.g., 1.5–3.0 s, which produces multiple circular trajectories. The optimal trajectory is the one maximising the objective function consisting of three weighted components. In particular, the components evaluate the progress toward the goal, the distance to the closest obstacle, and the forward velocity of the robot. Numerous modifications of *DWA* have been proposed, as the objective function is expandable [329, 330]. However, the method does not explicitly capture the dynamics of the obstacles taking into account only their current position.

Another method, *Trajectory Rollout* [161] is similar to *DWA* but exhibits one essential difference – in each forward simulation step, a set of feasible velocity pairs is updated as the kinematic constraints are recalculated according to the current velocity and dynamic constraints.

Constraints related to social conventions are usually embedded in the environment representation used by trajectory planners [219] or by extending the objective function [331, 221]. For example, Weinrich et al. [219] applied the E^* algorithm as a global path planner along with an extended *DWA* method as a local trajectory planner. They extended *DWA* with an additional objective rating that considers spatiotemporal occupation probabilities of the tracked humans. In particular, they assigned personal spaces to humans using *Gaussian Mixtures*. The method provided successful collision avoidance by the robot in a passing scenario of a narrow hallway. A similar extension of *DWA* was proposed in [332].

Seder et al. [331] and Oli et al. [221] proposed navigation approaches that employed a modified *DWA* for human-aware local trajectory planning. They introduced human awareness by modifying the objective component related to clearance from obstacles, in particular, including predicted poses of tracked humans as future obstacle positions. The difference between those methods is that in [331] the authors assumed human motion predictions driven by the constant velocity model, while in [221] the *SFM* has been implemented. Also, the method from [331] used *Focused* D^* as a global path planner, whereas in [221] – the *NF1* [264] was integrated.

***TEB*-based methods** The *TEB* is a traditional local trajectory planner that laid a foundation for multiple methods that enhanced this approach to capture human-awareness constraints [216, 168, 333]. The basic *TEB* deforms local trajectories according to the locations of obstacles in the environment, but, in contrast to *Elastic Bands*, with temporal information. Instead of forces from *Elastic Bands*, *TEB* uses an optimisation objective to follow the global path regarding kinodynamic constraints, forming the optimisation problem of non-linear least-squares.

Human-aware specialisation of *TEB*, named *HaTEB*, was proposed by Khambhaita and Alami [216]. They extended the original optimisation constraints with safety (minimum safety distance), time to collision, and directional constraints, including the predicted human trajectories in the problem formulation. Singamaneni et al. [217, 168] developed the *CoHAN* planner – the *HaTEB* extension that handles large numbers of people and focuses on motion legibility improvements. The *CoHAN* has different tunable planning modes that can handle various indoor and crowded scenarios. Recently, Hoang et al. [333] studied the topic of the robot approaching people in dynamic social environments and presented *GTEB* model – a goal-oriented specialisation of *TEB* planner. Their approach takes into account the robot's current state, robot dynamics, dynamic social zones [275], regular obstacles, and potential approaching poses to generate the socially optimal robot trajectory.

Other methods Alternatively to *DWA*- and *TEB*-based methods, Forer et al. [334] proposed the *Pareto Concavity Elimination Transformation* (*PaCcET*) local trajectory planner. It aims to capture the non-linear human navigation behaviour, scoring trajectories with multiple objectives. The first relies on path distance, goal distance, heading difference and distance to obstacles, while the second is based on the interpersonal distance between the robot and humans. Later, Banisetty et al. [228] extended *PaCcET* with social awareness objectives, specifically, maintaining appropriate distances to F-formations (groups) and distance to a scenario-dependent social goal. In contrast, the authors of [335] proposed a planner that aims to exaggerate motions to increase intent expressiveness over passing sides for legible robot navigation [23]. They implemented a decision-making strategy, constructing the *Social Momentum* objective that takes pairwise momentum between robot and human into consideration. Another method was presented by Mehta et al. [336] who applied *Multi-Policy Decision Making* to navigate in dynamic environments with different policies, namely, *Go-Solo*, *Follow-other*, and *Stop*. The values of utility functions, which compromise between the distance travelled to the goal and the disturbance to surrounding agents caused by the robot, are predicted through forward simulation.

Optimal control techniques have also been employed to maintain the formation integrity [337, 338]. For instance, in [337], formation control in a leader-follower arrangement was discussed. The authors developed a method that, under mild assumptions, guarantees the stabilisation of the formation to the desired shape and scale. Similarly, an optimal control algorithm, but for sustaining formations of various structures, was proposed in [338]. On the other hand, Truc et al. [339] developed a 3D reactive planner for human-aware drone navigation in populated environments that is based on a stochastic optimisation of discomfort caused by the drone's proximity to pedestrians and the visibility of the drone.

Learning-based methods In recent years rapid growth in the machine learning field has been observed, and numerous planning approaches have evolved to capture the intricacies of human behaviours and transfer them into robot control strategies. The broadest attention in robot control applications gained *Reinforcement Learning* (*RL*) and *Deep Reinforcement Learning* (*DRL*). Specialised surveys on the applications of *RL* methods for robot navigation [340] and particularly on social robot navigation were already published [341].

Inverse Reinforcement Learning A distinctively useful method for learning from demonstration is *Inverse Reinforcement Learning* (*IRL*) [190], as it allows to model the factors that motivate people's actions instead of the actions themselves [189]. Example applications of *IRL* methods for human motion prediction were already presented in Sec. 3.1.3, but they might also be used for control purposes. For example, Kim and Pineau [342] learned a cost function involving social cues from features extracted from *RGB-D* camera. Their *IRL* module uses a set of demonstration trajectories to learn the reference behaviour when faced with different state features. Their approach is implemented as a trajectory planner with *IRL*-based cost function operating along with a global path planner. Similarly, Kuderer et al. [343] also use *IRL* with human demonstrations, but they extract features from the human trajectories and then use entropy maximisation to determine the robot's behaviour during navigation in human environments. Pérez-Higueras et al. [298] also used *IRL* to transfer human motion behaviour to a mobile robot. They evaluated different *Markov Decision Process* models and compared them with the baseline implementation of a global path planner and local trajectory planner without social costs. More recently, Karnan et al. [344] collected a large-scale dataset of socially compliant navigation demonstrations. They used it to perform behaviour cloning [345] for a global path planner and local trajectory planner agents that aimed to mimic human navigation behaviours. The authors also performed an evaluation study for the learned approach, comparing it with a baseline *ROS* implementation.

Reinforcement Learning In contrast to *IRL*, the *RL* is used when the reward function is known or can be easily defined, and the goal is to find the best policy for achieving cumulative rewards. Recent works present the *DRL* as a framework to model complex interactions and cooperation, e.g., in social robot navigation.

In a study by Olivier et al. [327], the authors found that walking people mutually adjust their trajectories to avoid collision. This concept was exploited by Silva and Fraichard [346], whose approach relies on sharing motion effort between a robot and a human to avoid collisions. They learned a robot behaviour using the *RL* to solve the reciprocal collision avoidance problem during simulated trials.

Li et al. [183] presented a *Role Playing Learning* formulated under a *RL* framework for purely local navigation of a robot accompanying a pedestrian. In their approach, the robot takes into account the motion of its companion to maintain a sense of affinity when they are travelling together towards a certain goal. A navigation policy is trained by *Trust Region Policy Optimisation* with the use of features extracted from a *LiDAR* along with the goal as an input to output continuous velocity commands for navigation.

A series of works by Chen et al. [347, 348] developed *Collision Avoidance with Deep Reinforcement Learning* (*CADRL*) approaches. Specifically, in a *Socially-Aware CADRL* (*SA-CADRL*) [348], they designed a hand-crafted reward function that incorporates the social convention of passing side and enables a robot to move at human walking speed in a real-world populated environment. Everett et al. [163] proposed a *GPU/CPU Asynchronous Advantage Actor-Critic CADRL* (*GA3C-CADRL*) strategy that employs *LSTM* to use observations of arbitrary number or surrounding agents, while previous methods had this size fixed. A distinctive characteristic is that their algorithm learns collision avoidance among various types of dynamic agents without assuming they follow any particular behaviour rules.

Jin et al. [349] presented another *DRL* method but for mapless collision avoidance navigation where humans are detected using *LiDAR* scans. The reward function regards ego-safety, assessed from the robot's perspective, and social-safety, evaluating the impact of the robot's actions on nearby humans. The ego-safety zone maintains 0.4 m of separation between the robot and other objects, while social safety aims to prevent intrusions into approximated human personal space. Liang et al. [155] developed a *RL*-based collision-avoidance algorithm, named *CrowdSteer*, for navigation in crowded environments. The authors trained the algorithm using *Proximal Policy Optimization* (*PPO*) in high-fidelity simulation and deployed the approach on two differential drive robots.

Chen et al. [350] discussed extending pairwise interactions between the robot and individual humans to a robot interacting with a crowd. The authors developed *Socially Attentive Reinforcement Learning* (*SARL*) that jointly models human-robot as well as human-human interactions in an attention-based *DRL* framework by learning the collective importance of neighbouring humans with respect to their future states. Their work was further enhanced by Li et al. [351] who addressed the problems of learned policies being limited to certain distances associated with the training procedure and the simplified environment representation that neglects obstacles different from humans. In their *SARL** method, they introduced a dynamic local goal-setting mechanism and a map-based safe action space, addressing the problem of multiple detours of *SARL*. Nevertheless, social constraints, other than keeping at least some separation distance between the robot and humans, were not included.

Guldenring et al. [352] proposed another *DRL*-based system to train neural-network policies for local trajectory planning explicitly taking nearby humans into consideration. The approach uses *Proximal Policy Optimization* (*PPO*) as the main learning method while *DRL* agents are trained in randomised virtual 2D environments interacting with humans in an unfocused manner for plain collision avoidance. In addition, the method relies on raw data readings, in contrast to, e.g., [163].

Recently, Xie and Dames [156] proposed *DRL* policy for robot navigation through obstacle-filled and populated areas that intend to be generalisable to new environments. In particular, the *DRL-VO* reward function contains a novel term based on *VO* (Sec. 3.2.2) to guide the robot to actively avoid pedestrians and move toward the goal. In turn, Qin et al. [353] introduced a socially-aware robot mapless navigation algorithm employing *RL* to learn strategies that conform to social customs and obey specific traffic rules.

Miscellaneous approaches Besides the aforementioned methods, learning-based applications include employing *Hidden Markov Model* (*HMM*) in a higher hierarchy system to learn the choice between the *RL*-based collision avoidance and target pursuing [354].

On the other hand, Tai et al. [193] attempted to apply *Generative Adversarial Imitation Learning* (*GAIL*) strategy to navigate in populated dynamic environments in a socially compliant manner via only raw depth inputs from *RGB-D* camera. Their approach learns continuous actions and desired force toward the target and outperformed pure behaviour cloning policy regarding safety and efficiency.

In the approach by Lu et al. [355], the crowd's density is dynamically quantified and incorporated into a reward function deciding the robot's distance from pedestrians. The authors extended the *DRL*-based work from [350], so the best action is inferred from a reward function that regards the uncomfortable distance between the robot and a human. Alternatively, a system proposed by Yao et al. [124] incorporates a *Generative Adversarial Network* to track and follow social groups.

Physical safety (Req. 2.1)	
[53, 356, 228, 133, 102, 348, 350, 135, 237, 324, 163, 315, 165, 334, 144, 322, 8, 252, 352, 36, 241, 106, 143, 333, 349, 120, 131, 111, 216, 211, 71, 167, 330, 215, 85, 343, 183, 351, 155, 84, 355, 147, 152, 357, 335, 336, 277, 292, 255, 126, 139, 251, 338, 214, 305, 240, 221, 61, 146, 141, 164, 358, 297, 353, 253, 254, 166, 306, 294, 77, 162, 130, 213, 307, 4, 242, 331, 7, 346, 217, 168, 150, 274, 90, 293, 283, 337, 153, 125, 91, 66, 220, 169, 339, 359, 275, 314, 328, 154, 189, 231, 276, 281, 219, 156, 124, 67, 121, 136, 148, 235]	
Perceived safety (Req. 2.2)	
Personal spaces	[356, 228, 133, 350, 135, 324, 165, 334, 322, 8, 252, 352, 36, 241, 143, 333, 349, 111, 216, 71, 167, 215, 85, 183, 351, 155, 84, 355, 152, 336, 277, 139, 214, 240, 221, 61, 146, 141, 297, 353, 253, 254, 166, 306, 294, 77, 130, 307, 242, 7, 168, 150, 274, 360, 90, 293, 153, 91, 66, 169, 359, 275, 314, 154, 231, 276, 219, 156]
O-spaces of F-formations	[53, 228, 324, 241, 277, 240, 254, 166, 294, 77, 242, 360, 169, 359, 275, 314, 154, 231, 276, 124]
Passing speed	[106, 61, 146, 217, 168, 150, 339, 154, 189, 67]
Motion legibility	[324, 111, 216, 211, 215, 85, 343, 357, 335, 353, 217, 168, 150, 169, 328, 189, 156, 67, 148]
Approach direction	[53, 102, 237, 252, 333, 277, 253, 254, 166, 4, 90, 293, 91, 66, 339, 359, 275, 314]
Approach speed	[53, 102, 253, 254, 166, 91, 66]
Occlusion zones	[141, 150, 274]

Table 3.1: Classification of robot navigation methods implementing human safety requirements from the presented taxonomy.

3.2.3 Discussion

A summary of discussed navigation methods according to the requirements they implement is presented in Tab. 3.1 and 3.2. The approaches listed in most cases employ the hierarchical structure in the motion planning system composed of a global path planner and a local trajectory planner. However, not all works explicitly reveal the planning algorithms used; thus, we do not show the details in that matter.

Each reviewed navigation method is classified based on the objectives addressed in the approach. However, the consequence of this methodology is that behaviour cloning or imitation learning (Sec. 3.2.2) are excluded from this classification, as without investigating

Motion naturalness (Req. 2.3)	
Velocity smoothness	[135, 165, 144, 36, 71, 156]
Oscillations	[155, 152]
In-place rotations	—
Backward movements	—
Gaze modulation	[106, 111, 84]
Social conventions (Req. 2.4)	
Accompanying	[53, 237, 315, 252, 120, 131, 183, 336, 126, 251, 338, 141, 253, 254, 166, 130, 337, 125, 124, 121, 136]
Affordance spaces	[133, 135, 359, 275, 314, 231, 276, 235]
Activity spaces	[133, 135, 359, 275, 314, 231, 276]
Passing side	[348, 71, 343, 84, 61, 146, 141, 229]
Yielding way	—
Standing in line	[228, 135, 139]
Elevator etiquette	—

Table 3.2: Classification of robot navigation methods implementing the requirements of robot's motion naturalness and obedience to social conventions from the presented taxonomy.

the dataset, it is not clear which features were captured; hence, which requirements were targeted. On the other hand, *VO*-based methods (Sec. 3.2.2), which proactively adjust motion direction to avoid collisions, are always denoted as respecting *motion legibility* (**Req. 2.2.4**) (Sec. 2.3.4).

The requirements group most covered is by far the *physical safety* (**Req. 2.1**) inherited by social robot navigation from traditional navigation. It regards collision avoidance; hence, even approaches that do not explicitly regard humans in the environment (but rather moving obstacles) fall into this category. The most popular objective among social robot navigation algorithms is respecting personal spaces. However, in most methods, they are modelled as a circular shape, while many studies revealed their asymmetry (Sec. 2.3.1). In contrast, *motion naturalness* and, importantly, *social conventions* aspects are less frequently discussed. The latter is rarely considered, as in research, robots are usually designated for specific tasks, which influences a fragmentary approach to design and implementation.

3.3 Evaluation

Evaluating social robot navigation systems is essential for gathering insights on comfort among users and optimising their performance in real-world environments. This section discusses different evaluation methods, classifies types of studies conducted to explore or verify designed navigation algorithms, and identifies tools facilitating efficient assessment, namely, datasets, simulators, and benchmarks (Fig. 3.4).

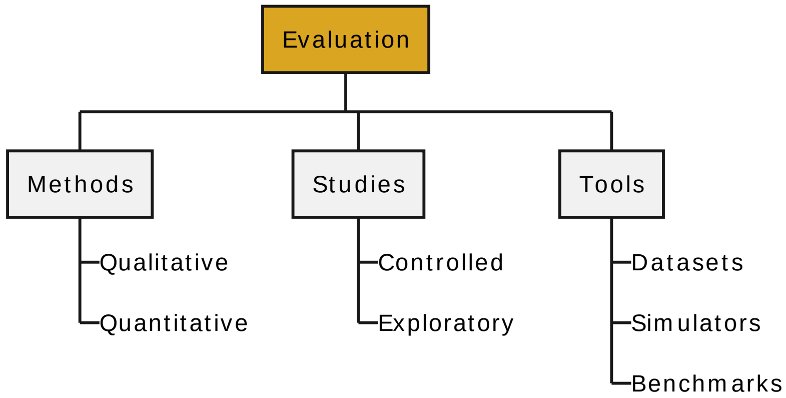


Figure 3.4: A taxonomy of evaluation for social robot navigation.

3.3.1 Methods

In general, evaluation methods encompass qualitative and quantitative approaches. Qualitative methods often involve subjective assessments, such as questionnaires conducted during user studies, which gauge users' preferences and comfort levels while interacting with the robot (e.g. [53, 7, 97]). These subjective evaluations provide valuable insights into the social acceptability of robot navigation.

On the other hand, quantitative methods utilise objective metrics formulated mathematically to assess various aspects of robot performance and social awareness (e.g. [342, 336, 357, 8, 330]). These metrics enable precise assessment and, thus, evidence-based comparison of different navigation algorithms. Researchers employing a combination of qualitative and quantitative evaluation methods [95, 8, 335] can comprehensively gauge both the performance and suitability of human-aware navigation systems in meeting the expectations of users.

In recent work, Biswas et al. [35] stated that an ideal method of evaluating social robot navigation is a large-scale, costly, and time-consuming qualitative user study. However, due to the indicated drawbacks, automated methods that provide a quantitative approximation of facts are required. Quantitative assessment methods are particularly useful for learning-based approaches, where the reward of action must be numeric. Similarly, the authors of planners that employ heuristics or optimise a single criterion benefit from benchmarking their methods against various strategies. Since automated quantitative methods produce invariable indicators of the algorithm's performance, they are particularly relevant for usage, e.g., during the new algorithm development stage. Nevertheless, grounding the social robot navigation requirements and approximating the social phenomena as quantitative metrics would be impossible without user studies yielding qualitative results.

3.3.2 Studies

Social robotics experiments often involve user studies to gather subjective human impressions about the robot's behaviour, which is crucial for social robot navigation as they provide valuable insights that can be directly transferred onto navigation system requirements (Chapter 2). Experiments conducted for collecting such data can be differentiated between controlled and exploratory.

Controlled studies provide the possibility to conduct tests under configurable conditions. Hence, researchers can control variables and conditions to isolate specific factors, e.g., robot speed [72], passing distances [61], and observe their effects. This, in turn, allows for gathering more precise measures of robot behaviour when operating with different navigation algorithms. This type of study might include both questionnaires and laboratory studies. In contrast, exploratory studies are conducted in natural conditions with minimum or no preparation. They might take the form of, e.g., a case study [361] to gain insights or field studies [362, 363] connected with observing and gathering data (qualitative and/or quantitative) regarding a robot deployed in the target environment. The principles of human-robot interaction studies design were identified by Bartneck et al. in [364].

Controlled studies facilitate the systematic evaluation of the robot's human awareness across different motion planning algorithms. However, direct comparison necessitates adherence to two crucial rules. Firstly, environmental conditions must be reproducible in subsequent trials. Secondly, a specific baseline motion planning setup (e.g., relying on classical navigation objectives), against which the examined navigation system will be compared, must remain unchanged in the following trials. In the literature, customised navigation approaches are contrasted against other algorithms [217] or a teleoperated agent [166], depending on the study design and goals.

Controlled laboratory studies intend to simplify complex interactions into prescribed scenarios of agents' movements under constant environmental conditions, so the number of varying factors in subsequent trials is limited. Gao and Huang [10] identified standard scenarios investigated in social robot navigation works that include passing [327, 72, 365], crossing [215, 83], overtaking [319, 348, 72], approaching [359, 275, 333], accompanying [129, 253, 166], or combined.

3.3.3 Tools

Multiple tools facilitate the evaluation of social robot navigation approaches. They are particularly useful for performing preliminary tests before arranging real-world experiments, which may pose a significant organisational effort [99, 87, 4, 7].

Datasets The datasets can be employed to train models for human trajectory prediction and learning robot movements in populated environments. They are irreplaceable for neural approaches that optimise policy learning from data [329, 355, 277].

The pioneering datasets in the field are *ETH* [366] and *UCY* [367], suitable for tracking and prediction. They provide pedestrian trajectories from a top-view, fixed, outdoor-located camera. Later, Rudenko et al. [368] developed *THÖR* indoor dataset with human trajectory and eye gaze data with accurate ground truth information. The data was collected using motion capture hardware with 3D *LiDAR* recordings and a mobile robot in the scene. Another dataset, named *SCAND*, was proposed by Karnan et al. [344] and contains indoor and outdoor data from multiple sensors of a mobile robot teleoperated in a socially compliant manner.

Alternatively, *SocNav1* [369] and *SocNav2* [356] datasets were designed to learn and benchmark functions estimating social conventions in robot navigation by using human feedback in simulated environments. Wang et al. [370] developed *TBD* dataset containing human-verified labels, a combination of top-down and egocentric views, and naturalistic human behaviour in the presence of a mobile capturing system moving in a socially acceptable way. Another dataset was used as a part of the *CrowdBot* project and is applicable for crowd detection and tracking, as well as learning navigation in populated, dynamic environments [371].

Recently, new datasets have emerged, for example, *SiT* [372], which contains indoor and outdoor recordings collected while the robot navigated in a crowded environment, capturing dense human-robot interactive dynamic scenarios with annotated pedestrian information. Nguyen et al. [373] developed *MuSoHu* dataset gathering recordings of sensors placed on human participants walking in human-occupied spaces; thus, interactions between robots and humans have not been captured. Hirose et al. [143] presented *HuRoN* dataset collected with multi-modal sensory data from a robot operating with an autonomous policy interacting with humans in real-world scenes.

The publications relying on some of these datasets were identified in [10] and partially in [17], while in [9] the authors separated datasets for activity recognition, human pose estimation, and trajectory prediction.

Simulators In recent years, simulation experiments have been more often chosen due to the growth of the field of *RL* [348, 183, 163, 352, 156] and other data-driven approaches [193]. Simulators are particularly useful tools for the systematic evaluation of social robot navigation algorithms as they can provide identical initial conditions of experiments in the following trials, which is not always possible in user studies. Simulators also facilitate the agile development of algorithms and provide flexibility, which datasets often lack. Furthermore, as opposed to real-world tests, in terms of time and resources, they are easily reconfigurable and cost-effective in repeating trials.

Numerous simulation ecosystems have been developed for robotics [374]. The majority is directly applicable to social robotics as they provide movable human-like postures, and several are suitable for rich human-robot interaction. The main characteristics of state-of-the-art approaches for conducting virtual social robot navigation experiments are presented in Tab. 3.3, whereas Tab. 3.4 illustrates their methods for simulating human motion behaviours.

The comparison in Tab. 3.3 includes 2D and 3D simulators, as well as frameworks that have *ROS* integration (the most popular robotic framework), are actively maintained, and are open-source. Architectures of software for human simulation can be distinguished on standalone simulators and frameworks. The latter are usually designed for controlling simulated humans and they abstract from a specific simulator; therefore, interfacing components are necessary for integration. The proposed classification regards the fidelity of the replication of virtual robots, i.e., whether dynamic intricacies (friction, etc.) are included or only the ideal kinematic model is considered. Additionally, the comparison identifies the variety of tasks that can be performed by simulated humans and the methods for controlling humans. The capability of setting dynamic goals for virtual humans is crucial for rich human-robot interactions, which usually require an orchestrator. For example, handover tasks can be simulated only with the synchronisation of human and robot activities. Specifically, the human receives an object after the robot approaches them (which in high-fidelity simulation always takes varying amounts of time); hence, the reception must be triggered at different timestamps.

On the other hand, Tab. 3.4 presents the characteristics of the virtual humans' navigation in each simulation ecosystem. The comparison points out the algorithms used for motion planning and whether the motion of each agent can be configured differently. The classification also includes information on whether the simulation ecosystem allows the formation-like motion of virtual humans, which is restricted by the capabilities of motion planning algorithms available.

Notably, more advanced simulators facilitate transferring the algorithms from virtual to real-world hardware. All listed simulators except *flatland*[2] [352] provide the kinodynamic fidelity of robots, whereas the exactness of frameworks depends on the simulators they are integrated with. Simplified, lightweight simulators with the possibility to simulate dynamic agents, such as *SocialGym 2.0*, are well-suited for learning applications requiring multiple repetitions, whereas high-fidelity simulators, like *Gazebo* (*Ignition*) or *iGibson*, target the rich interaction scenarios. Nevertheless, transferring navigation methods from the simulation into real-world experiments is essential to demonstrate that developed algorithmic approaches work not only in simulated setups but are also reliable and prospective for wider applications.

Benchmarks Due to a growing set of navigation algorithms available, the importance of quantitative evaluation has increased. Lately, various automated quantitative assessment systems, called benchmarks, have been developed to ease the evaluation of traditional and social robot navigation. The appropriate benchmark design requires the knowledge of the requirements for robot navigation system (Chapter 2), concurrently from the classical and human-aware points of view [21].

Several works have recently proposed benchmarking frameworks for evaluating robot motion planning algorithms from the classical navigation perspective [34, 385, 386, 387, 388, 389, 33, 390, 391, 392], i.e., without considering human awareness constraints. Those works mainly focus on performance metrics like navigation success rate, path length, or time required to reach the goal.

Heiden et al. [34], for example, have introduced *Bench-MR* – a benchmark concerning sampling-based motion planners for nonholonomic, wheeled mobile robots. *Bench-MR* consists of two main components: motion planning algorithms and evaluation components. These latter indicate diverse navigation scenarios in static environments along with basic performance metrics assessing planning efficiency and path quality.

Another framework for comparing path planning algorithms is *PathBench* proposed by Toma et al. [385]. It provides implementations of classical and learned-based techniques allowing evaluation using typical metrics, e.g., path length, path deviation, success rate, and computational time. *PathBench* is relevant for simulated and real-world applications.

[2] `https://github.com/avidbots/flatland`

Approach	Software architecture	Robot fidelity	Human task variety	Human control		
				scripted scenarios	dynamic goals	teleop
Webots [375]	standalone	kinodynamic	*MG*	✓	—	—
Gazebo [376] (*Ignition*)	standalone	kinodynamic	*MG*, *PG*	✓	—	—
PedsimROS [149]	framework (*Gazebo* interface)	—	*MG*	✓	—	—
flatland	standalone	kinematic	*MG*	—	✓	—
HuBeRo [377]	framework (*Gazebo* interface)	—	*MG*, *PG*, *FO*, *ST*, *CO*, *MO*	✓	✓	✓
SEAN 2.0 [378]	*Unity*	kinodynamic	*MG*, *JG*	✓	✓	✓
Crowdbot [379]	*Unity*	kinodynamic	*MG*	✓	—	—
iGibson 2.0 [380]	standalone	kinodynamic	*MG*	✓	—	—
InHUS [381]	framework (*Stage/Morse* interfaces)	—	*MG*	✓	✓	✓
IMHuS [382]	framework (*Gazebo* interface)	—	*MG*	✓	✓	—
SocialGym 2.0 [383]	framework (*UTMRS* interface)	kinodynamic	*MG*	✓	✓	—
HuNavSim [384]	framework (*Gazebo* interface)	—	*MG*	✓	✓	—

Table 3.3: Classification of robotic simulation systems with capabilities for replicating human motion behaviour. Abbreviations used in the table: *MG* stands for moving to a goal, *PG* – performing gestures, *FO* – following an object, *ST* – sitting, *CO* – conversating, *JG* – joining groups, and *MO* – moving to an object.

Similarly, Rocha and Vivaldini [386] have proposed *Plannie* framework for developing, testing, and benchmarking various motion planning techniques in real-world 2D and 3D environments. The authors reimplemented classical, meta-heuristics, and machine learning planning algorithms that can be scored with common metrics such as a success rate, path length, time to produce a trajectory, and time to complete the mission.

Tani et al. [387] have introduced a robotics research platform focused on providing reproducibility of experiments. Their framework integrates development and benchmark-

Approach	Human motion planning	Human motion diversity	Human groups
Webots [375]	naive trajectory following	configurable speed in a scripted trajectory	—
Gazebo [376] (*Ignition*)	*APF*-like	configurable weights of potentials	—
PedsimROS [149]	*SFM*	configurable motion model's properties and group assignment	✓
flatland	any *ROS* plugin for motion planning	possible individual parameters for each planning agent	—
HuBeRo [377]	any *ROS* plugin for motion planning	possible individual parameters for each planning agent	—
SEAN 2.0 [378]	*Unity*'s built-in path planner with *SFM*	configurable behaviours (randomised, handcrafted or graph-based control of pedestrians), variable posture	✓
Crowdbot [379]	*DWA*, *RVO*, *SFM*	configurable speed in a scripted trajectory	—
iGibson 2.0 [380]	A^* with *ORCA*	configurable object radius of *ORCA*	—
InHUS [381]	any *ROS* plugin for motion planning	possible individual parameters for each planning agent	—
IMHuS [382]	any *ROS* plugin for motion planning	possible individual parameters for each planning agent	—
SocialGym 2.0 [383]	*SFM*	configurable motion model's properties and group assignment	—
HuNavSim [384]	*APF*-like/*SFM*	configurable behaviours (regular, impassive, surprised, curious, scared, threatening)	✓

Table 3.4: Classification of robotic simulation systems from the perspective of methods to replicate human motion behaviour.

ing, enabling users to create, test, and evaluate various motion planning algorithms in simulation and real robots. They mainly concentrated on autonomous vehicles operating in exemplary urban environments, validating the reproducibility of experiments across different robots using basic spatial metrics.

Mishkin et al. [388] proposed a method of evaluating classical and learning-based approaches to navigation. They tested different navigation algorithms only in simulation environments using basic metrics regarding the success rate, path length, and time required to reach the goal. Perille et al. [389] proposed *BARN* method to examine mobile robot navigation systems in standardised test environments. To evaluate the environment's difficulty, they used *Dynamic Window Approach* (*DWA*) [144] and *Elastic Bands* [310] algorithms scored with simple metrics – traversal time and navigation failures.

Wen et al. [33] proposed *MRPB* framework for evaluating the general performance of mobile robot navigation. Although their approach is suitable for simulated and real-world tests, they did not incorporate any social metrics. Similar features characterise *Arena-Bench* [390], whose authors proposed a complete suite for benchmarking different navigation algorithms but without any human awareness metrics.

MotionBenchMaker [391] is one more open-source tool for benchmarking motion planning datasets. Their approach is intended to ease the evaluation of motion planning algorithms in typical manipulation tasks performed in a simulation. The authors compared different planners using only the average planning time metric. Another mainly performance-focused benchmark was proposed by Tafnakaji et al. [392], who assessed the navigation of mobile manipulators. They evaluated, e.g., the robot's accuracy of following the global path or final pose accuracy.

On the other hand, benchmarks for socially-aware robot navigation are the minority, but there are several works in that matter [35, 378], as well as ours [21, 20], extensively discussed in Chapter 4.

One of the examples, *SocNavBench* [35], is intended to regard social aspects in robot navigation, but implements only two basic indicators – distance to the closest pedestrian and time to collision. Moreover, integrating navigation algorithms other than those provided by the authors is considered tricky; therefore, the approach is not yet for practical use.

Another approach, proposed by Tsoi et al. [378], is *SEAN 2.0* – a framework for evaluating robot navigation using different metrics concerning motion efficiency and human awareness. However, despite the variety of tools provided and integration with the most popular robotic framework, ROS, their approach is not applicable for evaluating real-world experiments, as the metrics calculation is integrated into the simulator.

Mavrogiannis et al. [335] have also quantified human awareness of robot navigation in the work presenting their *Social Momentum* planning framework. The authors used known metrics – the topological complexity [393] and the path irregularity index [36], to compare their *Social Momentum* with other methods.

The primary features of state-of-the-art approaches for benchmarking robot navigation are presented in Tab. 3.5. The comparison includes only actively maintained and open-source benchmarks. Our Social Robot Planner Benchmark (*SRPB*) system, described in

Name	Metrics					Suitable env.	Analysis tools
	Classical navigation performance	Physical safety	Perceived safety	Motion naturalness	Social norms		
iGibson Benchmark [394]	✓	—	✓	—	—	*S*	–
MRPB [33]	✓✓✓✓	✓	—	✓	—	*S/R*	–
BenchMR [34]	✓✓✓✓✓ ✓	✓	—	✓	—	*S*	scenario rendering, metrics plots
CrowdBot Benchmark [379]	✓✓	✓✓	—	✓✓✓✓	—	*S*	scenario rendering, metrics plots
SocNavBench [35]	✓✓✓✓✓ ✓✓✓✓✓	✓✓	✓✓	✓✓	—	*S*	scenario rendering, metrics plots
Arena-Bench [390]	✓✓✓✓✓ ✓✓✓	✓	—	✓✓✓	—	*S*	scenario rendering, metrics plots
SEAN 2.0 [378]	✓✓✓✓✓ ✓✓✓	✓	✓✓	✓	—	*S*	–
InHuS [381]	✓	✓✓	✓	—	—	*S/R*	scenario and metrics rendering
Tafnakaji et al. [392]	✓✓✓✓✓	—	—	✓	—	*S/R*	scenario rendering
SRPB [21]	✓✓✓✓✓ ✓	✓✓✓✓	✓✓✓✓✓ ✓✓✓✓✓ ✓✓✓✓✓ ✓	✓✓✓✓✓	—	*S/R*	scenario rendering, metrics plots, exporting results to a LaTeX table or a spreadsheet
HuNavSim [384]	✓✓✓✓✓ ✓✓✓	✓✓✓✓	✓✓✓✓	✓✓	—	*S*	—

Table 3.5: A classification of state-of-the-art methods for quantitative evaluation of robot navigation requirements. The number of ticks (✓) reflects the number of metrics implemented in each benchmark. Abbreviations used: *S* stands for simulation environments, *R* – real-world environments, and *S/R* reflects simulation and real-world environments.

detail in Chapter 4, is also included in the listing. The classification of methods focuses on the variety of metrics implemented (according to the taxonomy identified in Chapter 2), as well as determining suitable test environments (simulation/real world) and a set of analysis tools provided, e.g., for results presentation. In some cases, simulators linked with social robotics are coupled with internally calculated metrics for assessing navigation [378, 383].

Quantitative metrics are the inherent parts of benchmark systems as they aim to implement objective criteria approximating subjective assessments. Therefore, the quantitative metrics should reflect mathematical formulas of requirements discussed in Chapter 2. Metrics covering most of the perceived safety principles for social robot navigation (Tab. 3.5) are developed in our *SRPB* benchmark, where human-awareness indicators also account for people tracking uncertainty, facilitating the evaluation with the robot's onboard perception [21]. Besides the listed benchmark systems, several complementary indicators for assessing the perceived safety of humans in the context of social robot navigation also appear in [395]. The survey by Gao and Hoang [10] discusses in detail metrics presented in the literature.

3.4 Summary

The study presented in Chapter 3 examines the key methods for addressing the perception, motion planning, and evaluation problems in the context of social robot navigation.

The fundamental challenges of social robot perception, identified as the detection and tracking of humans in the robot's environment, were analysed, and state-of-the-art algorithms implementing such were highlighted. Diverse environment representations utilised in different motion planning approaches were also discussed, as well as various methods for human trajectory prediction which is crucial in real robots equipped with sensors with a limited *field of view*. The survey also highlights the topic of contextual awareness and how it was tackled in state-of-the-art navigation approaches.

The major part of our review encompasses various methods employed for robot motion planning that take into account constraints arising from the presence of surrounding humans. Approaches present in the literature were classified into global path planning and local trajectory planning algorithms according to the common hierarchical structure of motion planning systems. Both global path planners and local trajectory planners were organised into groups sharing common algorithmic characteristics. Besides a thorough description of various navigation methods, those approaches are classified according to the established requirements taxonomy, based on the objectives addressed.

Our literature review also explores the methods for evaluating social robot navigation as well as study types and tools relevant to the agile development of navigation techniques. The tools for the assessment were discussed distinguishing datasets, simulators, and benchmarks. An extensive comparison of actively maintained simulators for social robotics was proposed. Moreover, benchmarks suitable for quantitative evaluation of social robot navigation were classified utilising the proposed requirements taxonomy, according to the implemented metrics.

Chapter 4

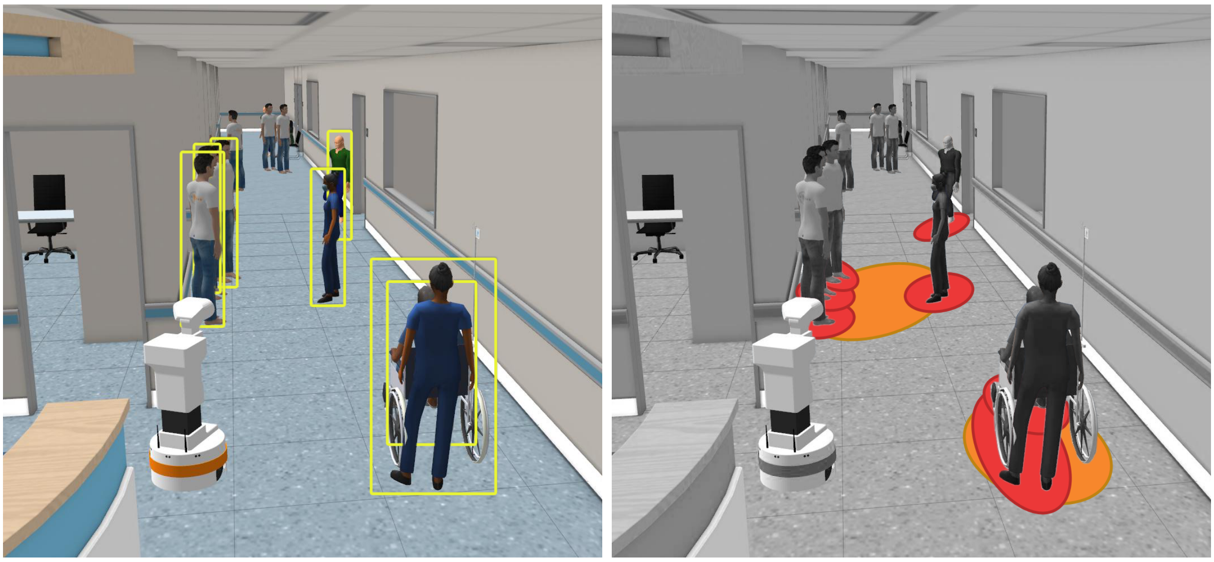
Quantitative evaluation of human-aware motion planning algorithms

Navigation is the fundamental skill of mobile robots that is widely integrated into most complex tasks. Since the 1960s, many approaches to robot navigation have been proposed [396]. The main objective of classical navigation algorithms is collision avoidance, considering all objects as generic obstacles. Social robot navigation, instead, relies on principles from social sciences. Based on research from that domain, robot systems designers try to deal with the presence of humans in the environment considering multiple objectives to react in a socially acceptable manner. Recently, due to the growing popularity of social robots, many researchers focused on creating human-aware navigation approaches [12, 10].

Since various navigation approaches are available, system designers must choose the best algorithm for a specific robotic application. Selecting the optimal method requires conducting comparative experiments that allow confronting investigated methods. Such experiments also benefit developers of new human-aware navigation algorithms, as they can reveal areas for potential improvement.

Robot navigation evaluation is difficult as demonstrating the overall advantage of one method over another is challenging. However, different algorithms can be compared regarding a single aspect, e.g., the undertaken path length or the time required to reach a goal. The evaluation complexity grows with the number of navigation objectives, as in human-aware navigation.

Biswas et al. [35] discussed an ideal method of evaluating social navigation. They state that qualitative methods providing a good approximation of facts are large-scale, costly, and time-consuming. We agree that automated quantitative methods are more appropriate



(a) Online stage (b) Offline stage

Figure 4.1: The two-stage procedure of *SRPB* benchmark for assessing the quality of the robot navigation. (a) Online stage: a navigating robot tracks obstacle locations, humans (marked as bounding boxes in the figure), F-formations and its own state, e.g., a pose and velocity. All the data is recorded and saved to a file. (b) Offline stage: after a finished experiment recordings are used to evaluate the quantitative results of the navigation using multiple metrics. In (b), personal spaces are schematically depicted with red ellipses, whereas F-formations' O-spaces with orange ones.

for the iterative evaluation, e.g., during the new algorithm development stage, since they produce invariable indicators of the algorithm's performance.

Quantitative assessment methods are useful for learning-based approaches, where the reward of action must be numeric. Similarly, benchmarking against other methods may benefit planners that employ heuristics or optimise a single criterion.

Designing the appropriate benchmark requires knowledge of the requirements for navigation systems from both classical and human-aware perspectives. Navigation systems exhibiting socially acceptable robot behaviours cannot remarkably sacrifice the general effectiveness of robot motions in favour of social metrics maximisation.

To address the problem of the quantitative assessment of social robot navigation, we developed *SRPB* – the benchmark that evaluates both social and task performance aspects of robot navigation (Fig. 4.1). Novel metrics proposed in *SRPB* evaluate robot compliance with proxemics rules regarding single humans [47] and F-formations [75], as well as other social norms, e.g., avoidance of heading in the direction of a human [12]. Another original aspect is that our metrics are designed to account for the reduced tracking quality of humans since robot perception systems are imperfect. Our benchmark can be used to test robots operating in simulated and real-world environments. Moreover, metrics were formulated so as to allow benchmark usage with different robot types (either with nonholonomic or holonomic drives). We provide an open-source implementation of our benchmark system[1] that is compatible with the Robot Operating System (ROS) [180].

The metrics proposed for evaluating human-aware robot navigation are defined on the basis of the findings from various user studies. We reached the results from the literature to perform the grounding of social robot navigation requirements (Chapter 2) and extracted the guidelines to formulate relevant metrics. Notably, some proposed *SRPB* indicators directly model the discrete findings of the user studies, e.g., [72, 64]. On the other hand, the metrics for the assessment of general robot navigation performance are mostly derived from state-of-the-art benchmarks, but the extension of several metrics was also proposed. The new indicators are mainly dictated with practical reasons, as explained in the further part of the chapter.

We state that the closest to our work is *MRPB* [33]; however, we extended that method concerning metrics diversity, focusing on human awareness indices. Furthermore, our benchmark allows evaluating different methods during on-site tests (simulated or real-world); robot operation in a preprepared environment is not required as in [389, 387]. Since robot navigation behaviours can be evaluated in target environments, our benchmark allows a more accurate algorithm selection for a specific application. Also, *SRPB* aims not to reimplement state-of-the-art navigation methods (as in [34, 386]) but relies on ROS-integrated, easily swappable planning algorithms that are under constant development. Furthermore, such an approach does not restrict the usage of the *SRPB* with any specific class of planners. Our benchmark allows comparing path planners [34, 385] and trajectory planners in separation or as combined motion planning methods [386].

The remainder of this chapter consists of the definition of the notation (Sec. 4.1), which is then used to present mathematical formulations of metrics implemented in the *SRPB* benchmark. The metrics are organised according to the requirements taxonomy defined in Chapter 2. Specifically, the following groups of indicators were distinguished: metrics for evaluation of robot navigation performance Sec. 4.2, metrics for evaluation of robot motion naturalness Sec. 4.3, and metrics for evaluation of perceived safety among humans Sec. 4.4. In the last section of the chapter, Sec. 4.5, the presented benchmark system is critically analysed.

This chapter constitutes an extended description of the proposed benchmarking method based on our conference paper [20] and the journal article [21].

[1] `https://github.com/rayvburn/srpb`

4.1 Mathematical notation

To describe the metrics for the robot navigation evaluation, we developed a mathematical notation used in equations (Fig. 4.2). The top-left index (b) of the symbol corresponds to a specific entity (from those listed in the ontology), whose state influences the value of the entire symbol. Some symbols may depend on states of multiple entities, in which case the b is represented by a set of entities' identifiers. A value of any entity at time t^n is referred to as $(\cdot)^n$. Common symbols are presented in Tab. 4.1.

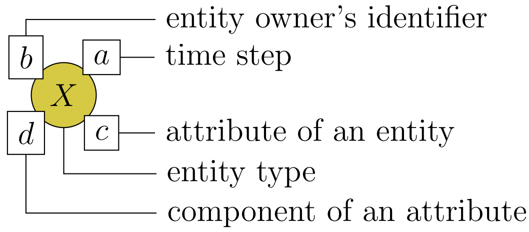


Figure 4.2: A general description of symbol composition method used in the notation.

The ontology that we propose for social robot navigation is organised as follows: the world configuration at each time t^n consists of the state of a single robot, r^n, and the state of its environment. The latter, recalculated at each time step, aggregates: a set of obstacles, $\mathbb{O}^n$, and a set of humans, $\mathbb{H}^n$, that may be arranged into F-formations, $\mathbb{G}^n$. Therefore, at time t^n, the association of h-th human into g-th F-formation can be expressed as ${}^{h}H^n \in {}^{g}G^n$, whereas ${}^{g}G^n \in \mathbb{G}^n$. The ${}^{h}H^n$, being prone to collisions with the robot, can also be involved in calculations related to generic obstacles, $\mathbb{O}^n$.

Experiment time stamps, t^n, where $n = \{1, \ldots, N\}$, are shared among the robot, humans and human groups. We commonly refer to the summation of time differences between subsequent time steps to consider that they may not be equal-sampled in non-real-time systems, affecting average values. Conditional summation is represented with the Iverson bracket operator [397].

The following sections present metrics calculation methods focusing on social navigation metrics derived from the requirements (Chapter 2). Nevertheless, general navigation performance aspects are also briefly discussed.

4.2 Metrics for evaluation of robot navigation performance

Socially acceptable robot behaviours should not significantly degrade the general performance of the navigation task (**Req. 1**). The problem of robot performance during

Symbol	Description
r	identifier of a robot
t^n	n-th time stamp
${}^{i}H$	human identified as i
$\mathbb{H}$	set of humans
${}^{i}G$	humans F-formation, i.e., a group, identified as i
$\mathbb{G}$	set of humans F-formations
${}^{i,j}d$	Euclidean distance between i and j
$\mathbf{p}$	pose vector in a form $[x, y, \theta]^T$
$\mathbf{v}$	velocity vector in a form $\left[v_x, v_y, \omega\right]^T$
$\mathbf{a}$	acceleration vector in a form $\left[a_x, a_y, \alpha\right]^T$
${}^{i,j}\phi$	direction of a vector connecting centres of i and j
${}^{i,j}\delta$	relative location of i regarding the heading of j
$\mathfrak{R}\left(\mathrm{Z}, \theta\right)$	rotation matrix around Z axis by the θ angle
var	variance
$\mathbf{\Sigma}$	covariance matrix
$\mathcal{N}$	normal distribution
f	function
m	metric

Table 4.1: A dictionary of common symbols used for formulating quantitative metrics.

navigation was already discussed in multiple works, as shown in Sec. 3.3.3. However, we propose several metrics that, we argue, are also crucial for a robot behaviour assessment.

4.2.1 Obstacle safety

Robot navigation benchmarks usually report the number of collisions along the path to the goal [378, 390] but this type of assessment is not anticipatory. Hence, we argue that for robust navigation approaches, assessing the percentage of time the robot has spent in the dangerous area around obstacles (nearer than the configurable distance of ${}^{r,\mathbb{O}}d_{\text{min}}$) is more appropriate. The relevant metric was presented in [33, 390], which we refer to as m_{obs} (4.3) (further explained by (4.1) and (4.2)).

$$ {}^{r,o}d^n = \min_{{}^{o}O^n \in \mathbb{O}^n} \sqrt{\sum_{j \in \{x,y\}} \left({}^{r}j^n - {}^{o}j^n\right)^2} \tag{4.1} $$

The distance ${}^{r,o}d^n$ is calculated from the centre of the robot to a border point of the o-th obstacle. Therefore, the Euclidean distance representing the actual gap between the objects is less by the robot's circumradius, d_{cr}.

$$t_o = \sum_{n=1}^{N-1} \left(t^{n+1} - t^n \right) \left[{}^{r,o}d^n < {}^{r,\mathbb{O}}d_{\text{min}} \right] \tag{4.2}$$

$$m_{\text{obs}} = \frac{t_o}{t^N - t^1} \cdot 100\% \tag{4.3}$$

We argue that m_{obs} metric is sufficient to assess the compliance with **Req. 1.1**; however, a complementary metric, ${}_{\text{min}}m_{\text{obs}}$, representing the minimum distance between the robot's centre and the closest obstacle point throughout the experiment (4.4), is also proposed.

$${}_{\text{min}}m_{\text{obs}} = \min_{n \in \{1,\dots,N\}} {}^{r,o}d^n \tag{4.4}$$

4.2.2 Motion efficiency

A metric expressing the time required to reach the goal pose (ours m_{mef} (4.5)) was proposed in [388, 33, 35, 390, 378, 392] and is appropriate for verification of the goal-reaching requirement (**Req. 1.3**).

$$m_{\text{mef}} = t^N - t^1 \tag{4.5}$$

4.2.3 Path length

Classical navigation is often focused on minimising of robot's path length while traversing to the goal (**Req. 1.4**). The path is determined by a sequence of poses. To evaluate the path length, m_{plin} (4.6), the sum of Euclidean displacements of the mobile base during the scenario is computed [390, 35, 378, 392].

$$m_{\text{plin}} = \sum_{n=1}^{N-1} \sqrt{\sum_{j \in \{x,y\}} \left({}^{r}j^{n+1} - {}^{r}j^{n} \right)^2} \tag{4.6}$$

4.2.4 Cumulative heading change

A metric complementary to the m_{plin} represents robot orientation change along the path (**Req. 1.5**). For example, the path irregularity metric was discussed in [36], providing a normalised score of unnecessary turning per unit path length. However, since it requires knowing the perfect path to the goal, we argue that it applies only to very small or perfectly known environments. Therefore, in our benchmark, we use the cumulative heading change metric, m_{chc} (4.7), as in [240, 398, 37].

$$m_{\text{chc}} = \sum_{n=1}^{N-1} |{}^{r}\theta^{n+1} - {}^{r}\theta^{n}| \tag{4.7}$$

4.2.5 Computational efficiency

Trajectory planners for mobile base navigation have different degrees of complexity. Therefore, it is adequate to verify the average computation time the planner takes to accomplish a new velocity command (**Req. 1.2**). Such a metric was proposed in [33], which we refer to as m_{cef} (4.8).

$$m_{\text{cef}} = \frac{1}{N} \sum_{n=1}^{N} c^{n} \tag{4.8}$$

4.2.6 Computational time repeatability

Evaluating how much computation times differ from the mean value, $\bar{c}$, is also important. It shows how likely the planner will violate requested computation times and, thus, whether it can be safely applied in robots operating in highly dynamic environments. Therefore, we proposed the m_{cre} metric, constituting a standard deviation of all computational times (n-th denoted as c^{n}) during the scenario (4.9).

$$m_{\text{cre}} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(c^{n} - \bar{c}\right)^{2}} \tag{4.9}$$

4.3 Metrics for evaluation of robot motion naturalness

Social metrics are essential for robots operating in dynamic and populated environments. This section discusses metrics related to robot motion naturalness (**Req. 2.3**).

4.3.1 Velocity smoothness

The velocity smoothness metric, m_{vsm}, defines how much robot's linear velocities, ${}^{r}v_{x}^{n}$ and ${}^{r}v_{y}^{n}$, differed in subsequent time steps, which indicates a presence of erratic motions (**Req. 2.3.1.1**). A similar metric was proposed in [33]; however, their formulation lacks the holonomic drive support. Instead, in our approach, both linear velocity components (along the x and y axes) are taken into consideration (4.10). Investigated robot velocities are expressed in the mobile base's coordinate system.

$$m_{\text{vsm}} = \frac{1}{N-1} \sum_{n=1}^{N-1} \frac{\sqrt{\sum_{j \in \{x,y\}} \left({}^{r}v_j^{n+1} - {}^{r}v_j^{n} \right)^2}}{t^{n+1} - t^{n}} \tag{4.10}$$

4.3.2 Heading change smoothness

Another indicator of erratic motions (**Req. 2.3.1.1**), defines an average rate of robot heading changes [240] during the scenario. The m_{hsm} metric is computed by comparing differences of robot angular velocity, ${}^{r}\omega$, in subsequent steps, as in (4.11).

$$m_{\text{hsm}} = \frac{1}{N-1} \sum_{n=1}^{N-1} \frac{\left| {}^{r}\omega^{n+1} - {}^{r}\omega^{n} \right|}{t^{n+1} - t^{n}} \tag{4.11}$$

Similar metrics regarding robot motion naturalness (m_{vsm} and m_{hsm}) were also discussed in [390, 378, 35], yet the authors did not show their calculation methods.

4.3.3 Oscillations

The oscillations metric[2], m_{osc}, defines the percentage of the total time that the robot has spent oscillating, i.e., has not developed significant linear and angular velocities (**Req. 2.3.1.2**). The oscillating behaviour in a given time step occurs when robot velocities, ${}^{r}\mathbf{v}^{n}$, and oscillation threshold velocities, ${}^{r}\mathbf{v}_{\text{osc}}$, fulfil conditions shown in (4.12). The linear speed of the robot at time t^n is represented as ${}^{r}v_{\text{lin}}^{n} = \sqrt{\left({}^{r}v_x^n\right)^2 + \left({}^{r}v_y^n\right)^2}$.

$$m_{\text{osc}} = \frac{100\%}{t^N - t^1} \sum_{n=1}^{N-1} \left(t^{n+1} - t^{n} \right) \begin{bmatrix} {}^{r}v_{\text{lin}}^{n} < {}^{r}_{\text{lin}}v_{\text{osc}} \\ \wedge \left|{}^{r}v_x^n\right| < {}^{r}_{x}v_{\text{osc}} \\ \wedge \left|{}^{r}v_y^n\right| < {}^{r}_{y}v_{\text{osc}} \\ \wedge \left|{}^{r}\omega^n\right| < {}^{r}\omega_{\text{osc}} \end{bmatrix} \tag{4.12}$$

4.3.4 Backward movements

The backward movements metric, m_{bwd}, defines the percentage of the total execution time that the robot has been advancing in the backward direction (**Req. 2.3.1.4**) with a speed of at least ${}^{r}_{x}v_{\text{osc}}$ (4.13).

$$m_{\text{bwd}} = \frac{100\%}{t^N - t^1} \sum_{n=1}^{N-1} \left(t^{n+1} - t^{n} \right) \left[{}^{r}v_x^n \leqslant -{}^{r}_{x}v_{\text{osc}} \right] \tag{4.13}$$

[2]A similar metric in [11], which has been published later than our [21], is named "stalled time".

4.3.5 In-place rotations

The in-place rotations metric, m_{iprot}, defines the percentage of the total time that the robot has spent rotating in place (**Req. 2.3.1.3**). In-place rotation is an action of the robot when its linear velocities are kept at 0, but the angular velocity is maintained above the threshold value of ${}^{r}\omega_{\text{osc}}$ (4.14).

$$m_{\text{iprot}} = \frac{100\%}{t^N - t^1} \sum_{n=1}^{N-1} \left(t^{n+1} - t^n\right) \left[\substack{{}^{r}v_x^n=0 \\ \wedge\, {}^{r}v_y^n=0 \\ \wedge |{}^{r}\omega^n| \geqslant {}^{r}\omega_{\text{osc}}}\right] \tag{4.14}$$

It is crucial that m_{osc}, m_{bwd} and m_{iprot} metrics are orthogonal to each other, i.e., in each time step robot's action can be qualified as fulfilling conditions of only one of these metrics.

4.4 Metrics for evaluation of perceived safety among humans

In this section, our metrics for the evaluation of the robot's intrusiveness and disturbance to adjacent people are discussed.

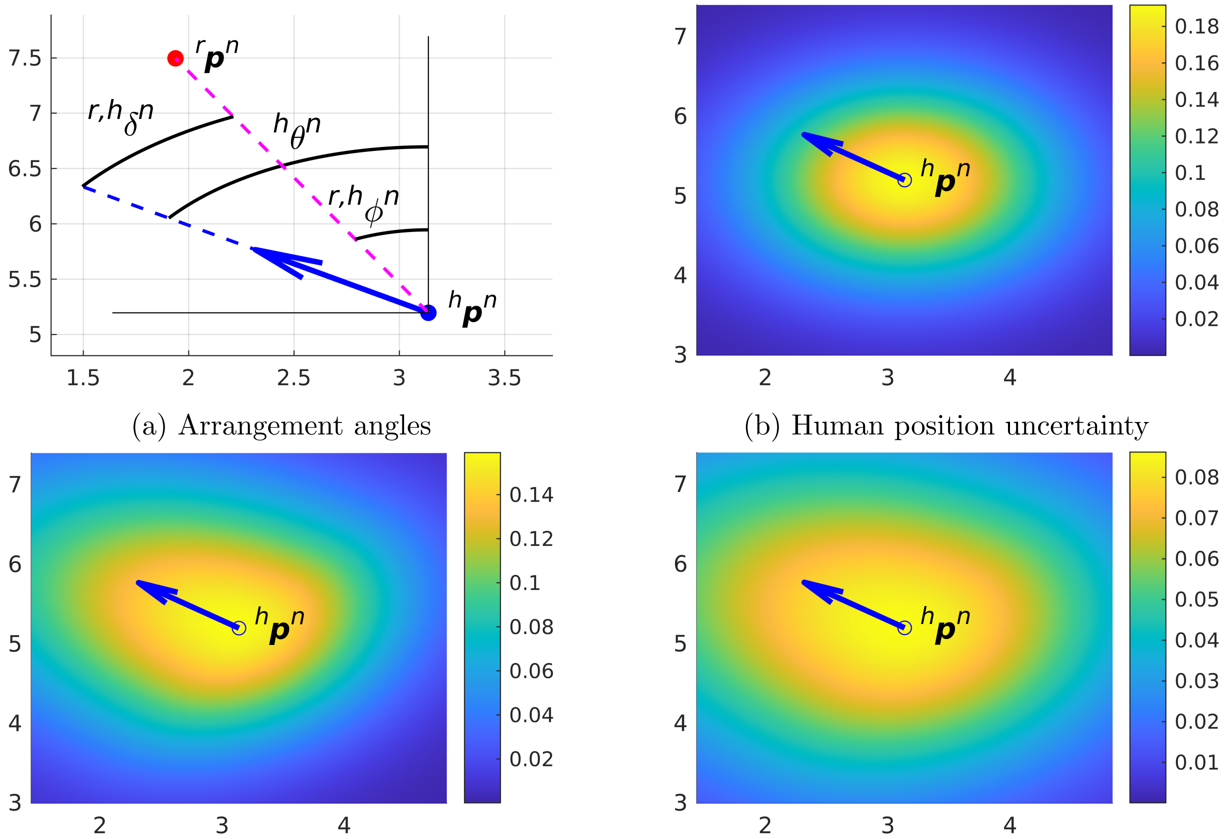
4.4.1 Personal spaces intrusion

The personal space concept was adopted in social robotics from the proxemics theory [47]. Our personal space intrusion metric, m_{psi}, defines the scale of robot intrusions into any human's personal space [12] throughout the scenario execution (**Req. 2.2.1**).

Recent studies show that Gaussian functions are legitimate for modelling personal spaces [399, 64]. Therefore, we represent the human's personal space as a multivariate asymmetric Gaussian function, f_{mag} (explained in A), centred at the h-th human's position, ${}^{h}x^n$ and ${}^{h}y^n$, oriented according to the human's heading ${}^{h}\theta^n$. Variances along the front (${}^{h}\text{var}_{\text{fr}}^n$), side (${}^{h}\text{var}_{\text{sd}}^n$), and rear (${}^{h}\text{var}_{\text{rr}}^n$) directions of the human pose were estimated in [399].

The variance along the heading axis, ${}^{r,h}\text{var}_{\text{hd}}^n$, is selected (${}^{h}\text{var}_{\text{fr}}^n$ or ${}^{h}\text{var}_{\text{rr}}^n$) in a three-step procedure, so the symmetrical variant of the multivariate Gaussian, f_{mg}, can be used in calculations. Firstly, to evolve, where the robot is located compared to the human's heading direction, the angle of the vector connecting the centres of the human and the robot, ${}^{r,h}\phi^n$, is computed (4.15). Then, the relative location ${}^{r,h}\delta^n$ of the robot r, compared to the human's h heading direction, is calculated as in (4.16) and presented in Fig. 4.3a. Finally, using the ${}^{r,h}\delta^n$ indicator, the variance is selected as in (4.17).

$${}^{r,h}\phi^n = \arctan 2\left({}^{r}y^n - {}^{h}y^n, {}^{r}x^n - {}^{h}x^n\right) \tag{4.15}$$



(a) Arrangement angles

(b) Human position uncertainty

(c) Personal space

(d) Personal space with position uncertainty

Figure 4.3: Processing of the h-th human data. Angles of an example arrangement along with lines reflecting the orientation of the global coordinate system (the x-axis pointing upwards) are presented in (a). Gaussians of: position uncertainty (b), personal space (c), and resultant distribution (d) are shown with the mean of the estimated h-th human pose. The personal space model was created using ${}^{h}\mathrm{var}^{n}_{\mathrm{fr}} = 3.0$, ${}^{h}\mathrm{var}^{n}_{\mathrm{rr}} = 0.75$, ${}^{h}\mathrm{var}^{n}_{\mathrm{sd}} = 1.33$.

$$ {}^{r,h}\delta^{n} = {}^{r,h}\phi^{n} - {}^{h}\theta^{n} \tag{4.16} $$

$$ {}^{r,h}\mathrm{var}^{n}_{\mathrm{hd}} = \begin{cases} {}^{h}\mathrm{var}^{n}_{\mathrm{fr}}, & \text{if} \left| {}^{r,h}\delta^{n} \right| \leqslant \frac{\pi}{2} \\ {}^{h}\mathrm{var}^{n}_{\mathrm{rr}}, & \text{otherwise} \end{cases} \tag{4.17} $$

To compute a value of f_{mg}, the h-th human's personal space covariance matrix needs to be created. Variances defining the personal space are expressed in the human's coordinate system; therefore, the personal space covariance matrix, ${}^{r,h}\boldsymbol{\Sigma}^{n}_{\mathrm{psi}}$, must be rotated according to the h-th human's orientation, ${}^{h}\theta^{n}$, as in (4.18).

$$ {}^{r,h}\boldsymbol{\Sigma}^{n}_{\mathrm{psi}} = \mathfrak{R}\left(\mathrm{Z}, {}^{h}\theta^{n}\right) \begin{bmatrix} {}^{r,h}\mathrm{var}^{n}_{\mathrm{hd}} & 0 \\ 0 & {}^{h}\mathrm{var}^{n}_{\mathrm{sd}} \end{bmatrix} \mathfrak{R}^{T}\left(\mathrm{Z}, {}^{h}\theta^{n}\right) \tag{4.18} $$

In the evaluation process, we also account for human tracking reliability. It aims to prevent excessive penalisation of robot states when, e.g., a tracked human becomes occluded. The covariance matrix of the estimated human position, ${}^{h}\mathbf{\Sigma}_{p}^{n}$, is obtained from the robot perception system. The sum of independent normal random variables is applied to compute the resultant covariance matrix, ${}^{r,h}_{\Delta}\mathbf{\Sigma}_{\text{psi}}^{n}$ (4.19). It accounts for position estimation uncertainty and the personal space model (Fig. 4.3).

$$ {}^{r,h}_{\Delta}\mathbf{\Sigma}_{\text{psi}}^{n} = {}^{h}\mathbf{\Sigma}_{p}^{n} + {}^{r,h}\mathbf{\Sigma}_{\text{psi}}^{n} \tag{4.19} $$

The scale of r robot intrusion into the personal space of h-th human in time t^n is referred to as ${}^{r,h}\text{psi}^n$ (4.20). It represents a value of the f_{mg} function (modelling the h-th human's personal space) at the robot's pose at that time, ${}^{r}\mathbf{p}^n$. Equation (4.20) presents arguments that the f_{mg} function takes – a pose and a multivariate normal distribution, the value of which will be computed at the given pose. The multivariate normal distribution, described by a mean of, e.g., $\mathbf{p}$, and covariance matrix of $\mathbf{\Sigma}$, is denoted as $\mathcal{N}(\mathbf{p}, \mathbf{\Sigma})$.

The final formulation of the personal space intrusions metric, m_{psi}, is shown in (4.21). Our method investigates the maximum intrusion in a given time step t^n, provided that some human was detected. The scale of the robot intrusion is normalised to the Gaussian value at the h-th human's centre, ${}^{h}\text{psi}^n$, so the metric value in each time step corresponds to a percentage of the maximum intrusion. If no human was observed during the scenario, $m_{\text{psi}} = 0$.

$$ {}^{r,h}\text{psi}^n = f_{\text{mg}}\left({}^{r}\mathbf{p}^n, \mathcal{N}\left({}^{h}\mathbf{p}^n, {}^{r,h}_{\Delta}\mathbf{\Sigma}_{\text{psi}}^{n}\right)\right) \tag{4.20} $$

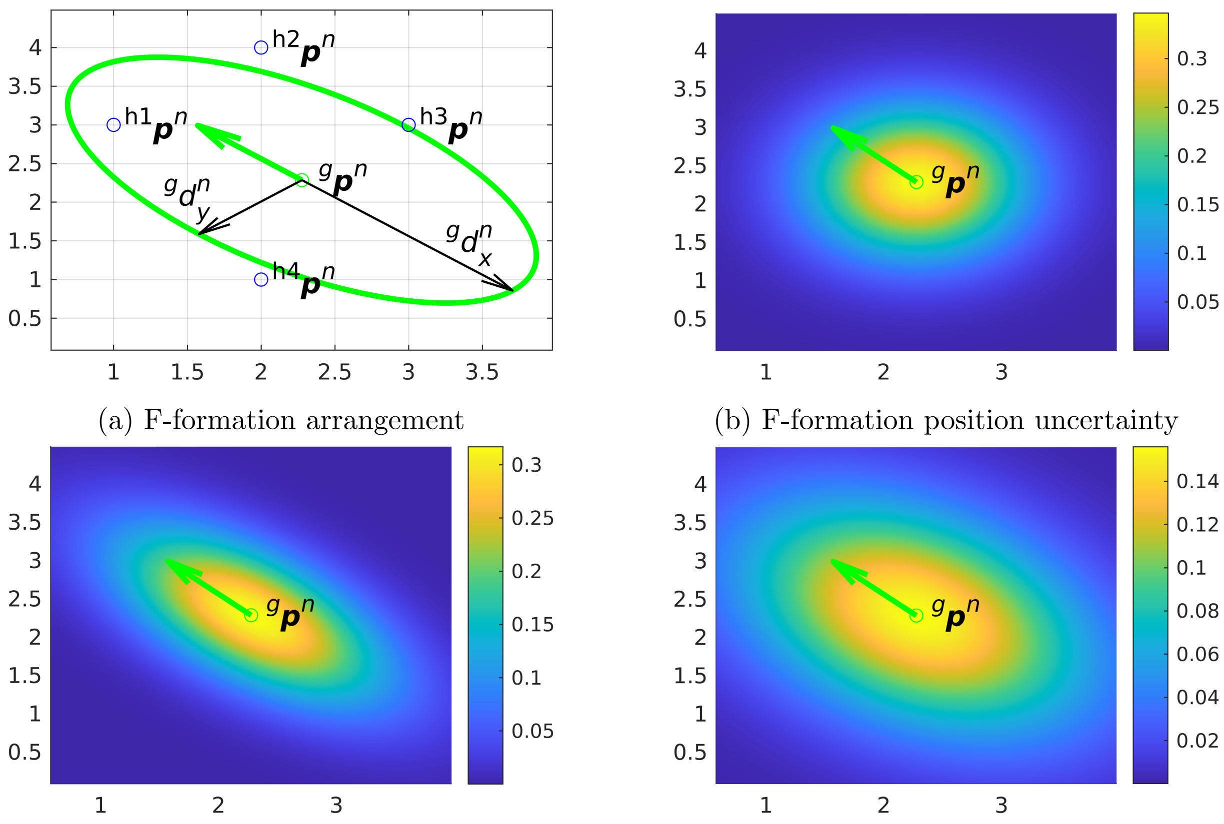
$$ m_{\text{psi}} = \frac{\sum\limits_{n=1}^{N-1}\left(\left(t^{n+1} - t^n\right) \max\limits_{{}^{h}H^n \in \mathbb{H}^n} \frac{{}^{r,h}\text{psi}^n}{{}^{h}\text{psi}^n}\right)}{\sum\limits_{n=1}^{N-1}\left(t^{n+1} - t^n\right)\left[\mathbb{H}^n = \varnothing\right]} \tag{4.21} $$

While the m_{psi} reflects the mean normalised value of the metric throughout the scenario execution, the benchmark implementation also provides the minimum and maximum of all normalised values collected in a single trial, i.e., ${}_{\min}m_{\text{psi}}$ and ${}_{\max}m_{\text{psi}}$, accordingly.

4.4.2 F-formations' O-spaces intrusion

The O-spaces of F-formations were proposed in [75] to reflect the elliptical spaces created by a group of humans involved in a focused interaction [2]. Our m_{fsi} metric aims to penalise a robot for traversing through O-spaces (**Req. 2.2.2**).

Firstly, to find the pose of the g-th O-space's ellipse, ${}^{g}\mathbf{p}^n$, we employ Taubin's algebraic method of ellipse fitting [400], supplied with mean positions of g-th F-formation members. Then, to assess the cost of robot movement in terms of human groups' presence, we model O-spaces as bivariate Gaussians (Fig. 4.4). The span of the 2-dimensional



(a) F-formation arrangement

(b) F-formation position uncertainty

(c) O-space of the F-formation

(d) O-space with position uncertainty

Figure 4.4: Processing of an exemplary F-formation consisting of 4 members. The mean of an estimated pose obtained from ellipse fitting is shown in (a). The remaining figures present corresponding Gaussians of position uncertainty (b), O-space (c), and resultant distribution (d).

O-space's Gaussian model is derived from the lengths of semi-axes ($^gd^n_x$ and $^gd^n_y$) of the F-formation's ellipse (Fig. 4.4a). Using the 2σ rule, the variances along the direction of the semi-major and semi-minor axes are derived, ${}^g_x\mathrm{var}^n_d$ and ${}^g_y\mathrm{var}^n_d$, accordingly (4.22). The O-space model's covariance matrix, ${}^g\boldsymbol{\Sigma}^n_{\mathrm{fsi}}$, expressed in the global coordinate system, is computed by applying a rotation (by the angle of the F-formation's ellipse orientation, ${}^g\theta^n$) to a matrix composed of variances as in (4.23).

$$\forall j \in \{x, y\}\,, {}^g_j\mathrm{var}^n_d = \left(\frac{{}^gd^n_j}{2}\right)^2 \tag{4.22}$$

$${}^g\boldsymbol{\Sigma}^n_{\mathrm{fsi}} = \mathfrak{R}\left(\mathrm{Z}, {}^g\theta^n\right) \begin{bmatrix} {}^g_x\mathrm{var}^n_d & 0 \\ 0 & {}^g_y\mathrm{var}^n_d \end{bmatrix} \mathfrak{R}^T\left(\mathrm{Z}, {}^g\theta^n\right) \tag{4.23}$$

In the spatial model of an F-formation, we also incorporate the uncertainty of the g-th F-formation's position estimation (Fig. 4.4b), arising from position uncertainties of members, ${}^g\mathbb{H}^n$. The uncertainty is represented by the variances: ${}^g_x\mathrm{var}^n_p$, ${}^g_y\mathrm{var}^n_p$, ${}^g_{\mathrm{xy}}\mathrm{var}^n_p$, and

${}_{\mathrm{yx}}^{g}\mathrm{var}_{p}^{n}$, computed as in (4.24) and (4.25). The composition of the F-formation's position covariance matrix, ${}^{g}\mathbf{\Sigma}_{p}^{n}$, is shown in (4.26).

$$\forall j \in \{x, y\}, {}_{j}^{g}\mathrm{var}_{p}^{n} = \max_{{}^{h}H^{n} \in {}^{g}G^{n}} {}_{j}^{h}\mathbf{\Sigma}_{p}^{n} \tag{4.24}$$

$${}_{\mathrm{xy}}^{g}\mathrm{var}_{p}^{n} = {}_{\mathrm{yx}}^{g}\mathrm{var}_{p}^{n} = \max_{{}^{h}H^{n} \in {}^{g}G^{n}} \left(\max_{j \in \{\mathrm{xy}, \mathrm{yx}\}} {}_{j}^{h}\mathbf{\Sigma}_{p}^{n} \right) \tag{4.25}$$

$${}^{g}\mathbf{\Sigma}_{p}^{n} = \begin{bmatrix} {}_{x}^{g}\mathrm{var}_{p}^{n} & {}_{\mathrm{xy}}^{g}\mathrm{var}_{p}^{n} \\ {}_{\mathrm{yx}}^{g}\mathrm{var}_{p}^{n} & {}_{y}^{g}\mathrm{var}_{p}^{n} \end{bmatrix} \tag{4.26}$$

The covariance matrix that accounts for F-formation's O-space and members' position estimation uncertainties, ${}_{\Delta}^{g}\mathbf{\Sigma}_{\mathrm{fsi}}^{n}$, is formulated as a sum of normally distributed random variables (4.27). The computation method of the intrusion, ${}^{r,g}\mathrm{fsi}^{n}$, of r robot into the O-space of g-th F-formation in time t^n, along with arguments that the Gaussian function takes, is presented in (4.28). The final formulation of the O-spaces intrusions metric, m_{fsi}, is shown in (4.29). The scale of the robot intrusion is normalised to the value of Gaussian at the g-th group's centre, ${}^{g}\mathrm{fsi}^{n}$, so the metric value in each time step corresponds to a percentage of the maximum intrusion. If no F-formation was observed during the scenario, $m_{\mathrm{fsi}} = 0$.

$${}_{\Delta}^{g}\mathbf{\Sigma}_{\mathrm{fsi}}^{n} = {}^{g}\mathbf{\Sigma}_{p}^{n} + {}^{g}\mathbf{\Sigma}_{\mathrm{fsi}}^{n} \tag{4.27}$$

$${}^{r,g}\mathrm{fsi}^{n} = f_{\mathrm{mg}}\left({}^{r}\mathbf{p}^{n}, \mathcal{N}\left({}^{g}\mathbf{p}^{n}, {}_{\Delta}^{g}\mathbf{\Sigma}_{\mathrm{fsi}}^{n}\right)\right) \tag{4.28}$$

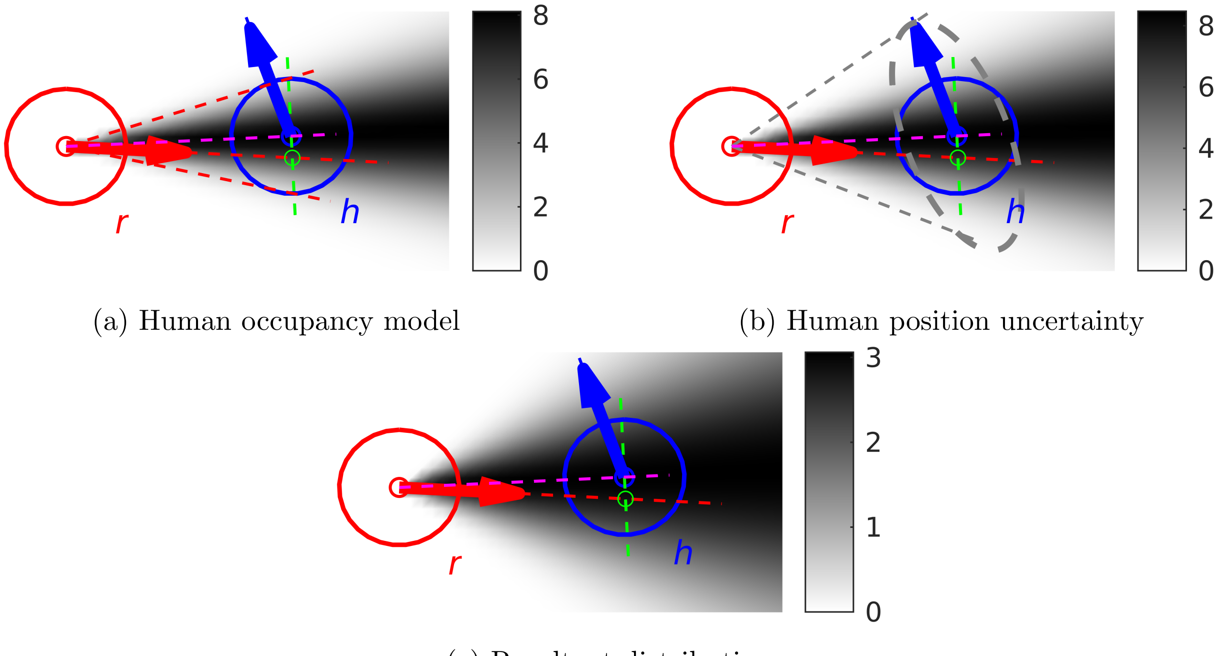
$$m_{\mathrm{fsi}} = \frac{\sum_{n=1}^{N-1} \left(\left(t^{n+1} - t^{n}\right) \max_{{}^{g}G^{n} \in \mathbb{G}^{n}} \frac{{}^{r,g}\mathrm{fsi}^{n}}{{}^{g}\mathrm{fsi}^{n}} \right)}{\sum_{n=1}^{N-1} \left(t^{n+1} - t^{n}\right) \left[\mathbb{G}^{n} = \varnothing\right]} \tag{4.29}$$

The m_{fsi} reflects the mean normalised value of the metric throughout the scenario execution, but the benchmark implementation also provides the minimum and maximum of all normalised values collected in a single trial, i.e., ${}_{\mathrm{min}}m_{\mathrm{fsi}}$ and ${}_{\mathrm{max}}m_{\mathrm{fsi}}$, accordingly.

4.4.3 Heading straight into a human

Reactive approaches to robot navigation usually suffer from late trajectory adjustment in dynamic environments causing the robot to turn just before the imminent collision with, e.g., a human, diminishing their perceived safety (**Req. 2.2.4**). The problem was initially investigated by Truong and Ngo [275], who tried to assess the robot's approach direction to the humans. However, their approach does not account for human position estimation uncertainty and a robot's dynamics.

Thus, we propose a new metric, m_{dir}, to evaluate the scale of the problem in different algorithms. The metric penalises a robot for undertaking motion directions leading straight into humans, especially when the robot moves with a decent speed near a human. It is directly related to the notion of motion legibility [23].



(a) Human occupancy model

(b) Human position uncertainty

(c) Resultant distribution

Figure 4.5: An exemplary human-robot arrangement and corresponding: distribution of the human physical space occupancy model (a), position estimation uncertainty (b), and resultant distribution (c). The robot's and human's heading directions are represented by arrows – red and blue, accordingly. Green dashed lines, constituting ${}^{r,h}l^{n}_{\text{cc}}$, are defined to find the intersection point, ${}^{r,h}\mathbf{p}^{n}_{\text{isc}}$, represented by green circles. Magenta dashed lines indicate the robot's direction with the maximum likelihood of heading straight into the human. Blue circles with a radius of d_{ocp} represent the human occupancy model, whereas in (b), the grey ellipse represents human position estimation uncertainty (cut-off determined by the 2σ rule).

To compute the metric, we investigate a geometrical arrangement of the human h and the robot r. Namely, we compare the robot's current heading to directions leading into the centre of the human. The span of cross-human robot heading angles arises from the space physically occupied by the human (inflated with a circular model[3] with a configurable radius of d_{ocp}) and the human position estimation uncertainty (represented by a covariance matrix, ${}^{h}\mathbf{\Sigma}^{n}_{p}$). Both effects are visualised in Fig. 4.5.

The variance of the bivariate Gaussian representing the circular occupancy model,

[3]This procedure addresses the typical simplification of perception systems representing a human as a pose in space, without estimating the area it occupies.

$\mathrm{var}_{\mathrm{ocp}}$, is computed by applying the 2σ rule to the d_{ocp} (4.30). The resultant covariance matrix, ${}^{h}_{\Delta}\boldsymbol{\Sigma}^{n}_{\mathrm{dir}}$, aggregating the occupancy model and the position uncertainty, is defined as in (4.31).

$$\mathrm{var}_{\mathrm{ocp}} = \left(\frac{d_{\mathrm{ocp}}}{2}\right)^{2} \tag{4.30}$$

$${}^{h}_{\Delta}\boldsymbol{\Sigma}^{n}_{\mathrm{dir}} = {}^{h}\boldsymbol{\Sigma}^{n}_{p} + \begin{bmatrix} \mathrm{var}_{\mathrm{ocp}} & 0 \\ 0 & \mathrm{var}_{\mathrm{ocp}} \end{bmatrix} \tag{4.31}$$

The value of the Gaussian at the ${}^{r,h}\mathbf{p}^{n}_{\mathrm{isc}}$ point, ${}^{r,h}\mathrm{dir}^{n}_{\mathrm{cc}}$, represents how much the robot's direction leads into the centre of the human (4.33). The ${}^{r,h}\mathbf{p}^{n}_{\mathrm{isc}}$ is an intersection point of the robot's direction axis (ray), $\overrightarrow{{}^{r}\mathbf{p}^{n}}$, and the line, ${}^{r,h}l^{n}_{\mathrm{cc}}$, defined by the crossed point and the direction angle in (4.32). The geometrical representation of finding the ${}^{r,h}\mathbf{p}^{n}_{\mathrm{isc}}$ is depicted in Fig. 4.5.

$${}^{r,h}l^{n}_{\mathrm{cc}} = \overleftrightarrow{{}^{h}\mathbf{p}^{n}, \measuredangle\left({}^{r,h}\phi^{n} + \frac{\pi}{2}\right)} \tag{4.32}$$

$${}^{r,h}\mathrm{dir}^{n}_{\mathrm{cc}} = f_{\mathrm{mg}}\left({}^{r,h}\mathbf{p}^{n}_{\mathrm{isc}}, \mathcal{N}\left({}^{h}\mathbf{p}^{n}, {}^{h}_{\Delta}\boldsymbol{\Sigma}^{n}_{\mathrm{dir}}\right)\right) \tag{4.33}$$

We also investigate how much the human can notice the robot's movement (potentially disturbing), the scale of which is represented by ${}^{r,h}\mathrm{fov}^{n}$. Applying the 2σ rule to the configurable field of view angle, φ_{fov}, the variance, $\mathrm{var}_{\mathrm{fov}}$, is computed. The relative location indicator, ${}^{r,h}\delta^{n}$ (4.16), determines directly how far the robot is situated from the centre axis of human's sight. Then, the value of the Gaussian appointed in the normalised angle domain, f_{ang} (explained in B), is computed for the current arrangement of the human and the robot (4.34).

$${}^{r,h}\mathrm{fov}^{n} = f_{\mathrm{ang}}\left({}^{r,h}\delta^{n}, \mathcal{N}\left(0, \mathrm{var}_{\mathrm{fov}}\right)\right) \tag{4.34}$$

The m_{dir} metric also accounts for the speed of the robot, ${}^{r}v^{n}_{\mathrm{lin}}$, and the distance between the human and the robot, ${}^{r,h}d^{n}$. The final formulation of the robot heading direction penalty, ${}^{r,h}\mathrm{dir}^{n}$, defined for a single human-robot pair, is presented in (4.35).

The normalisation of the metric value relies on comparing the current arrangement to the worst possible case. To accomplish that, platform-specific features must be determined, namely the circumradius of the mobile base, d_{cr}, and the maximum linear speed of the robot, ${}^{r}_{\max}v_{\mathrm{lin}}$. Moreover, it is assumed that the robot's heading points straight into the human position (${}^{r,h}\mathbf{p}^{n}_{\mathrm{isc}} = {}^{h}\mathbf{p}^{n}$, computed in ${}^{r,h}_{\max}\mathrm{dir}^{n}_{\mathrm{cc}}$) and the robot is located along the human's sight axis (${}^{r,h}\delta^{n} = 0$, calculated in ${}^{r,h}\mathrm{fov}^{n}_{\max}$). The formula for the normalisation factor, ${}^{r,h}\mathrm{dir}^{n}_{\mathrm{nrm}}$, is shown in (4.36). The metric for the whole scenario, m_{dir}, is calculated as in (4.37) and corresponds to the average percentage of heading disturbance generated by the robot. If no human was observed during the scenario, $m_{\mathrm{dir}} = 0$.

$$ {}^{r,h}\mathrm{dir}^{n} = \frac{{}^{r,h}\mathrm{dir}^{n}_{\mathrm{cc}} \cdot {}^{r,h}\mathrm{fov}^{n} \cdot {}^{r}v^{n}_{\mathrm{lin}}}{{}^{r,h}d^{n}} \tag{4.35} $$

$$ {}^{r,h}\mathrm{dir}^{n}_{\mathrm{nrm}} = \frac{{}^{r,h}_{\max}\mathrm{dir}^{n}_{\mathrm{cc}} \cdot {}^{r,h}\mathrm{fov}^{n}_{\max} \cdot {}^{r}_{\max}v_{\mathrm{lin}}}{d_{\mathrm{cr}} + d_{\mathrm{ocp}}} \tag{4.36} $$

$$ m_{\mathrm{dir}} = \frac{\sum\limits_{n=1}^{N-1} \left(\left(t^{n+1} - t^{n}\right) \max\limits_{{}^{h}H^{n} \in \mathbb{H}^{n}} \frac{{}^{r,h}\mathrm{dir}^{n}}{{}^{r,h}\mathrm{dir}^{n}_{\mathrm{nrm}}} \right)}{\sum\limits_{n=1}^{N-1} \left(t^{n+1} - t^{n}\right) \left[\mathbb{H}^{n} = \varnothing\right]} \tag{4.37} $$

The m_{dir} reflects the mean normalised value of the metric throughout the scenario execution, but the benchmark implementation also provides the minimum and maximum of all normalised values collected in a single trial, i.e., ${}_{\min}m_{\mathrm{dir}}$ and ${}_{\max}m_{\mathrm{dir}}$, accordingly.

4.5 Summary

In this chapter, *SRPB* has been presented – the social robot navigation benchmark that evaluates both the performance and the human-awareness aspects. It was designed to verify the fulfilment of the robot navigation requirements and assist system designers in selecting the best method for the application. Our approach allows comparing different navigation algorithms rapidly in both simulated and real-world environments. It also ensures easy integration with popular ROS-driven robots (differential drive and holonomic).

We focused on implementing quantitative metrics to evaluate common robot behaviour patterns. Most of the metrics in our benchmark allow confronting navigation algorithms, provided that the initial and final conditions of the evaluated scenario are the same in each trial. Therefore, path and trajectory similarities must be guaranteed in subsequent tests for a given scenario.

Our method investigates only unfocused interactions [2], so only the movement behaviours of humans and the robot in a shared space are evaluated. Extending our benchmark for evaluating focused human-robot interactions would be another significant contribution to social robotics. Initial research on this topic has already started and relates to, e.g., the approach pose of a robot that initiates an interaction with a human [275].

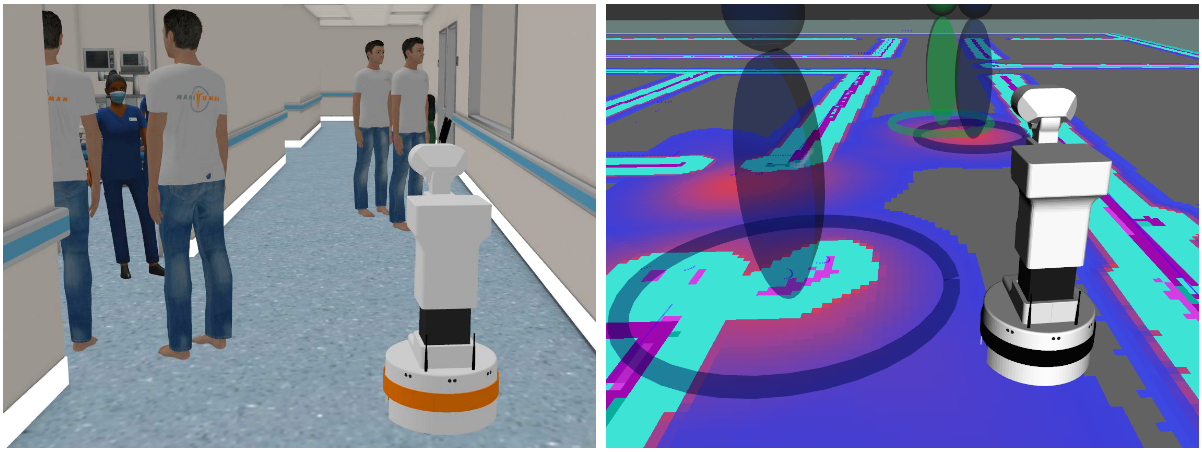
Chapter 5

Human-aware local trajectory planning for mobile robots using a hybrid trajectory generation and spatiotemporal cost functions

Implementing social acceptance in robot navigation can be performed in different ways. Firstly, adding spatial cost functions to the environment representation used for global path planning and local trajectory planning allows for the inclusion of, e.g., proxemics rules [47]. Secondly, adding spatiotemporal cost functions for local trajectory planning can produce natural robot motions, avoiding erratic movements [10]. Thirdly, the axiomatic way to create a socially acceptable robot motion is to apply pedestrian motion models. Generating trajectories using pedestrian motion models provides realistic and effective robot motions, mainly when using models calibrated with real-world data [25]. Finally, a combination of these methods can be applied to generate a comprehensive approach that includes social cues at every level of planning robot motions.

This chapter presents the local trajectory planning framework for mobile robots named *HUMAP – HUman-Aware Trajectory Planner MApping the Pedestrians Motion Pattern*. The planner produces trajectories regarding the human presence in the robot environment (Fig. 5.1). Our method accounts for spatial constraints arising from the proxemics theory [47], regards the naturalness of the robot's motion to reduce erratic movements, and explicitly incorporates human-like motion behaviours into the robot motion pattern. Namely, our planner implements yielding a way for a person crossing the robot's planned path, slowing down when a collision is predicted, and stopping when a collision is imminent.

Complementary to describing the procedure of generating human-aware trajectories,



(a) Simulation (b) Environment model

Figure 5.1: An overview of the robot motion planning system; (a) shows the robot operating in a populated hospital environment, whereas (b) represents the robot's model of this environment. In (b), the high-cost areas (ranging from blue to red) around humans (marked as simplified figures) correspond to proxemics-based spatial constraints, whereas obstacles are indicated as non-traversable costmap cells (light blue).

we also explain our spatial and spatiotemporal cost functions that allow selecting a robot trajectory that introduces the least disruption to human comfort [12, 10]. It is assessed with quantitative metrics that approximate people's impressions of association with a robot [64, 21].

The main contributions of our planning approach are:

- the hybrid method of generating local trajectory candidates utilising a pedestrian motion model and a velocity sampling approach, both regarding the robot's kinodynamic constraints,

- the extension of the pedestrian motion model with a component based on a *Fuzzy Inference System* for reasoning about the mutual spatial arrangement of the human and the robot; the enhancement aims to emphasise proactive movements that increase intent expressiveness and comply with social cues, e.g., passing on the dominant side,

- multifaceted spatiotemporal cost functions for evaluating trajectory candidates to mitigate human discomfort during efficient robot navigation,

- contextual awareness for efficient orchestration of the planner operation using the behaviour-based approach; specialised behaviours were implemented to yield a way for a human to cross the robot's path or to recover from a global path occlusion in dynamic and populated environments.

We state that the closest to our research topic are [315, 165, 253, 166]; however, our work stands out from the referenced algorithms. Ferrer et al. [315] modified the *SFM*-based approach for a specific navigation task – accompanying. In contrast, we directed our attention towards unfocused human-robot interactions that are more commonplace for assistive robots navigating in populated environments. In the following work [165], they employed the probabilistic *RRT* method with a *SFM*-based heuristic for generating feasible trajectory candidates for kinodynamic planning. However, they did not implement any proxemics-based cost functions to evaluate generated trajectories. Also, to diversify the obtained trajectories, they introduced randomness into the steering function. This makes their trajectory generation not systematic, whereas we directly employed a hybrid and deterministic approach to trajectory generation. Moreover, applying the velocity sampling trajectory generator in our hybrid approach makes our method immune to local minima and oscillations of the underlying motion model while still employing its human-like collision avoidance behaviour. On the other hand, we argue that offline pedestrian motion model parameter tuning, e.g., in [253], is unnecessary for legible robot navigation among humans. Instead, separating trajectory generation from trajectory evaluation with spatiotemporal cost functions generalises social navigation to being robust in various scenarios discussed in Sec. 7.4. In terms of implemented human-aware cost functions, the closest to our approach is [333], where dynamic social zones and F-formations' O-spaces were also investigated. In our approach, as in other model-based trajectory generation methods, searching for possible velocities in each planning step is not limited to a small set of motion primitives [144, 215]. Instead, velocities are allowed to vary along the prediction horizon, provided that they meet the kinodynamic constraints of the robot.

Our *HUMAP* planning framework copes with most of the limitations of the referred works, providing a comprehensive solution for social robot navigation in both structured and populated areas. The remainder of this chapter constitutes a detailed explanation of the method described in [24].

5.1 Basic concepts

In our previous work [21] (outcomes are also presented in Sec. 7.3), we quantitatively evaluated state-of-the-art traditional and human-aware trajectory planners using Social Robot Planner Benchmark (*SRPB*) and the results have shown that state-of-the-art human-aware trajectory planners do not significantly improve social navigation over classical approaches; hence, human-aware navigation is still an open problem. Conclusions drawn from the previous study prompted the investigation of that topic, and an alternative planner – *HUMAP*, has been developed.

The *SRPB*'s metrics are the preliminaries for this work, as they stand for quantitative indicators of human discomfort. Since trajectories regard both spatial and temporal dimensions, and the *SRPB*'s metrics capture spatiotemporal intricacies of human-robot interaction, those indicators are directly applicable to the trajectory planning's objective.

Notation The notation to describe the novel local trajectory planning scheme generally follows the rules explained in Chapter 4, in particular, Fig. 4.2 and Tab. 4.1. However, the notation in Chapter 4 is human-centric, while the notation in Chapter 5 is robot-centric. The difference can be exemplified using the relative location symbol, δ, that appears in both chapters. Namely, if the *upper-left* symbols of r, h (indicating the entities that the base symbol is related to) appear in Chapter 4, the ${}^{r,h}\delta^n$ denotes the relative location of the robot r, compared to the human's h heading direction. In contrast, in Chapter 5, the symbol of ${}^{r,j}\delta^n$ describes the angle of the j-th object's location in relation to the robot's r heading direction. For clarity in notation, several equations will be re-introduced in Chapter 5, if necessary. Additional symbols, appearing in the following sections, are explained in relevant tables, i.e., Tab. 5.4 and 5.5.

Trajectory definition In *HUMAP*, an individual trajectory is considered as a tuple storing subsequent poses, $\boldsymbol{p} = [x, y, \theta]^T$, and velocities, $\boldsymbol{v} = \left[v_x, v_y, \omega\right]^T$, achieved at the end of subsequent time steps (5.1). The duration of trajectories, t_{hor}, is derived from the plan's sampling period, t_Δ, and the number of samples within the planning horizon, q_{hor}, specifically $t_{\text{hor}} = q_{\text{hor}} \cdot t_\Delta$. For brevity, we also define the time stamp of the end of the planning horizon, t^{pl}, starting from the current t^n, computed as $t^{\text{pl}} = t^n + q_{\text{hor}} \cdot t_\Delta$.

$$ {}^{r}\text{traj}^n = \left[\left({}^{r}\boldsymbol{p}^n, {}^{r}\boldsymbol{v}^n\right), \ldots, \left({}^{r}\boldsymbol{p}^{\text{pl}}, {}^{r}\boldsymbol{v}^{\text{pl}}\right)\right] \tag{5.1} $$

Ontology The ontology to formalise our human-aware navigation system is partially derived from the entities specified in the pedestrian motion model we employ for trajectory generation. The ontology is organised as follows: the world configuration at each time step t^n consists of the state of a single robot, r^n, and the state of its environment, E^n. The latter, recalculated at each time step, aggregates: a set of obstacles, $\mathbb{O}^n$, either static, $\mathbb{K}^n$, or dynamic, $\mathbb{J}^n$. Dynamic obstacles represent moving humans and robots different from the ego-robot (the controlled one), as they are prone to collisions with the ego-robot. For clarity, we specified a set of social agents, $\mathbb{H}^n$, representing static or dynamic humans that are taken into consideration when, e.g., evaluating the cost functions, as well as F-formations [75] (social groups), reflected by $\mathbb{G}^n$, in which social agents can be arranged. At time t^n, the association of h-th human into g-th F-formation can be expressed as ${}^{h}H^n \in {}^{g}G^n$, whereas ${}^{g}G^n \in \mathbb{G}^n$ and $\mathbb{G}^n$ is a set of groups detected at time t^n. The final

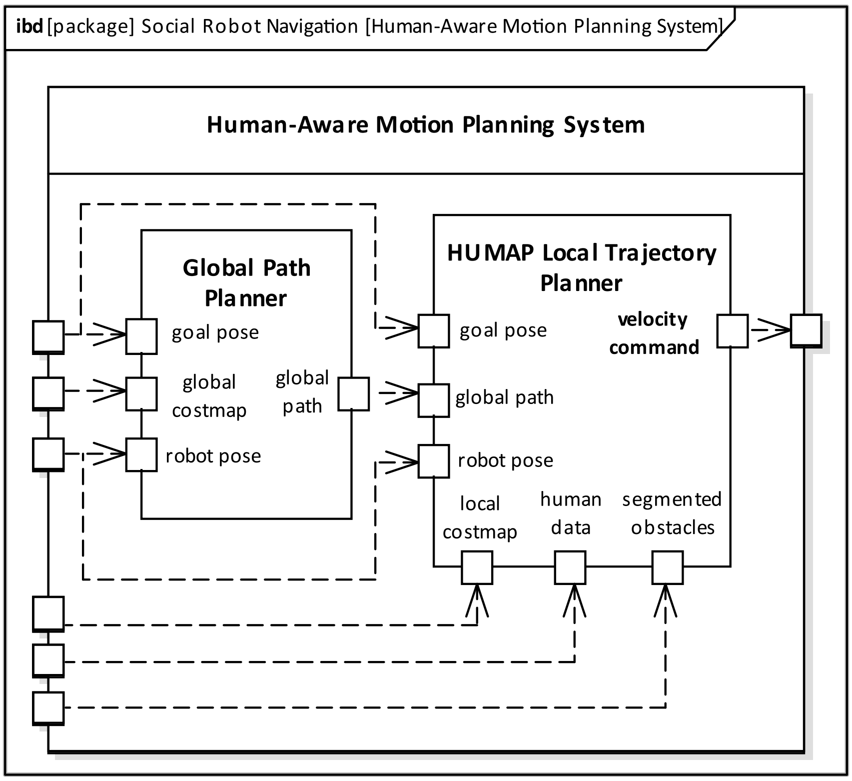


Figure 5.2: Internal block diagram of the motion planning system for social mobile robots. The system aggregates the presented human-aware local trajectory planner.

composition of the environment is expressed as $E^n = \{\mathbb{O}^n, \mathbb{H}^n, \mathbb{G}^n\} = \{\mathbb{K}^n, \mathbb{J}^n, \mathbb{H}^n, \mathbb{G}^n\}$.

Subsequent time stamps, t^n, are shared between all object types, while bold symbols indicate vectors or matrices.

5.2 Motion planning framework architecture

The overall structure of the presented motion planning framework is presented as the *SysML* internal block diagram, shown in Fig. 5.2. The detailed structure of the internal blocks of the *HUMAP* local trajectory planner is depicted in Sec. 6.9.

It is expected that the global path planner periodically recomputes the global path as the robot advances towards the goal [147, 401], which is de facto a standard approach for designing motion planning systems for mobile robots. A path and a trajectory are planned in separate coordinate systems [255], but the relation between them is known. This is further explained among implementation details in Sec. 6.1.

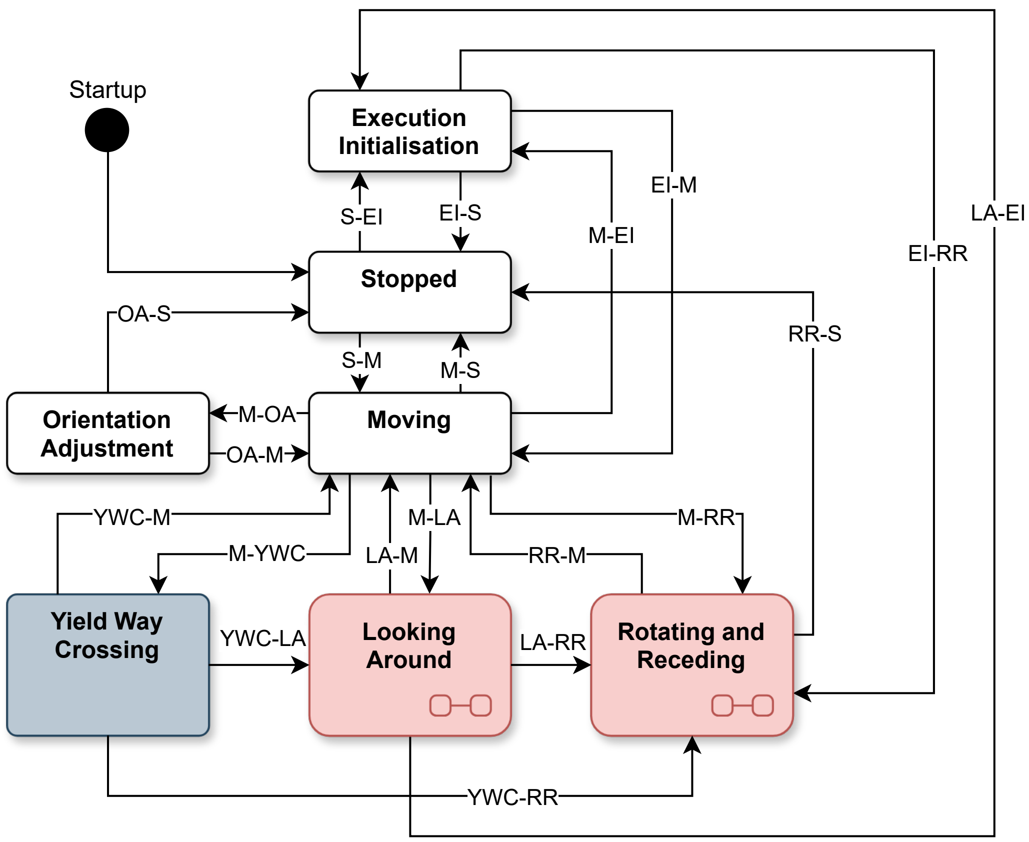


Figure 5.3: The finite state machine diagram of the *HUMAP*'s local trajectory planning scheme. Nested finite state machines are marked red, whereas the state indicated with blue implements a social convention. Identifiers of transition conditions are shown on the edges between states.

5.2.1 Finite State Machine

The fundamental behavioural aspects of the system are illustrated in Fig. 5.3, which shows the *Finite State Machine* (*FSM*) diagram.[1] The following states were distinguished in the planning scheme: *Stopped*, *Execution Initialisation*, *Moving*, *Orientation Adjustment*, *Yield Way Crossing*, as well as superstates *Look Around* and *Rotating and Receding*.

A brief description of each (super)state is presented in Tab. 5.1, along with typical situations, when a certain (super)state gets activated. However, the robot's activity in the *Moving* state will be thoroughly discussed. This is because the robot's behaviour associated with that state is the main novelty of the *HUMAP*, as it implements the extensive human-robot unfocused interaction strategy using the hybrid method for trajectory generation and spatiotemporal cost functions. Moreover, another contribution lies in the *Yield*

[1] For a concise description, the presented *FSM* is slightly simplified compared to the implemented version but illustrates most of the operational principles. The planner's implementation details are described in Sec. 6.9.

State	Robot behaviour description	Typical occurrence
Stopped	the algorithm assures that the robot is fully stopped	before or after the robot finishes or aborts a navigation task
Execution Initialisation	the algorithm rotates the robot in place until its orientation aligns with the direction towards the initialisation goal (simultaneously checking for potential collisions with basic cost functions)	the robot is oriented differently compared to the global path
Moving	the algorithm plans the robot's motion using the hybrid trajectory generation and trajectory scoring with the entire set of cost functions	regular operation in an empty environment or for unfocused interactions with dynamic objects (e.g., humans)
Orientation Adjustment	the algorithm rotates the robot in place until its orientation aligns with the global goal (simultaneously checking for potential collisions with basic cost functions)	the robot achieved the goal position, but not the goal orientation
Yield Way Crossing	based on predicted trajectories of moving objects (robots or humans), the algorithm finds a safe pose to approach (to perform yielding way to dynamic objects) and generates velocity commands to reach that point (instead of simply stopping)	the robot's path is expected to be crossed by a moving object(s)
Look Around	the algorithm performs a slow 25 cm backing up action (if the collision-free pose is found), then slowly rotates the mobile base: 60° to one side, then 120° to the other side, and 120° to the first side (all angles are relative to the reference orientation after the translation stage); if a valid global path is meanwhile received from the global planner, the translation or rotation procedure is terminated	the global path cannot be calculated, so the robot performs some actions to update the environment model
Rotating and Receding	the algorithm tries to find and approach a "recovery" position. If such a point is found, the strategy creates slow rotational movement first until the mobile base faces that position. Then, the algorithm creates commands allowing it to slowly and safely approach the "recovery" position (constantly checking for collision)	the robot is very close to an obstacle (e.g., a dynamic object that has approached) but not in collision

Table 5.1: Description of the *HUMAP*'s behaviour in each state and their typical occurrences.

Way Crossing state, whose corresponding behaviour realises a social norm, namely, yielding a way to a human at the crossing (**Req. 2.4.5**), discussed in Sec. 2.5.5. The robot's activity in other states will only be schematically presented.

Basic states as *Stopped*, *Execution Initialisation*, *Moving*, and *Orientation Adjustment* are sufficient for robot operation in structured and semi-structured but mainly static environments; however, additional states were added, such as *Rotating and Receding* or *Look Around*, which help to make navigation in dynamic environments robust against path occlusions and near-collision configurations.

Fig. 5.3 identifies the states and the state transitions in the system. Insights on the conditions causing those transitions are found in the predicates table shown in Tab. 5.2, and complementary transition conditions are illustrated in Tab. 5.3. The *FSM*'s update cycle occurs at each computation cycle of the planner; hence, each transition occurs no more frequently than the planner's operational period.

In any state, the *HUMAP* uses the global path, provided by the global path planner, to find specific "goal" poses to orchestrate the planner's operation. In particular, the initiation goal is placed along the global path approximately 0.2 m from the centre of the mobile base. It is used for limiting excessive motions at the start of the navigation execution. Next, the local goal is placed along the global path approximately at 110% of a distance that the robot can traverse along the whole planning horizon (based on the current state and kinodynamic limits) and is used as an intermediate goal supplied to the pedestrian motion model as well as to compute the `localGoalBehind` predicate. Finally, the global goal is placed at the end of the global path and reflects the task-level goal pose. While the global goal is updated externally (Sec. 6.1), the *HUMAP*'s planning system periodically refreshes the initiation goal and the local goal.

5.2.2 Behaviours implemented in the FSM's states

Each state presented in Fig. 5.3 has a corresponding one behaviour (superstates have multiple states), whose transition function is cyclically executed during operation in a state [31]. This section schematically describes most of the behaviours, but the calculations performed in the socially-aware activity associated with the *Moving* state are extensively explained in the following parts of this chapter. Internal parts of the superstates are not discussed for brevity, but some non-obvious transitions of the *FSM* are clarified. Nonetheless, the algorithmic description of the *HUMAP*'s activity shown in Alg. 1 and Alg. 3 applies to all behaviours, while Alg. 2 is specific to the transition function corresponding to the *Moving* state's behaviour.

Predicate	Predicate definition
`newGoal`	a new goal pose has been received, which activates a navigation task
`directedToInitGoal`	determines if the robot is facing towards the initialisation goal (with a 30° tolerance)
`localGoalBehind`	specifies whether the local goal is behind the robot (relative to its orientation; with a 30° tolerance)
`posReached`	determines if the goal position has been reached (with a 0.2 m tolerance)
`goalReached`	determines if the goal pose has been reached (with a 0.2 m tolerance on position and 0.2 rad tolerance on orientation)
`oscillating`	the absolute mean values of velocity components collected for 5 seconds are less than threshold values (0.02 m/s for linear components, and 0.06 rad/s for the angular component) and the "zero crossing" of the angular component has occurred at least once
`stuck`	the absolute mean values of velocity components collected for 5 seconds are negligible (less than 0.001 m/s for linear components, and 0.001 rad/s for the angular component), and "zero crossing" of the angular component did not occur
`nearCollision`	specifies if the inflated robot's footprint (2.5 cm extension) is in a collision according to the costmap
`canRecover`	the non-inflated robot's footprint is not in a collision according to the costmap (the inflated footprint might overlap with collision cells)
`globalPathOccluded`	a path to the goal expected from the global path planner was not updated for 1.5 s
`crossingDetected`	determines whether any human in proximity to the robot is expected to cross the robot's planned path or trajectory within the planning horizon (which equals to human trajectory prediction horizon)
`tempPosReached`	indicates whether the intermediate position found for a routine has been reached
`tempOrientReached`	specifies whether the target direction defined for a routine has been reached
`travelDistExceeded`	determines whether the maximum distance (0.75 m) has been travelled since the start of the "yielding way" routine
`closestHumanFarAway`	indicates that the closest human is further than 0.6 m
`ywRoutineEnded`	the "*Yield Way Crossing*" ends when: `tempPosReached` ∨ `tempOrientReached` ∨ `travelDistExceeded` ∨ `closestHumanFarAway`
`laRoutineEnded`	a sequence of "*Look Around*" rotations has ended
`rrRoutineEnded`	a timeout of 30 seconds has elapsed during "*Rotating and Receding*" and a non-collision pose was not reached

Table 5.2: Description of the predicates used for describing the *HUMAP*'s *Finite State Machine*'s transitions.

Execution Initialisation The transition function associated with the *Execution Initialisation* state's behaviour performs the in-place rotation to align the mobile base with

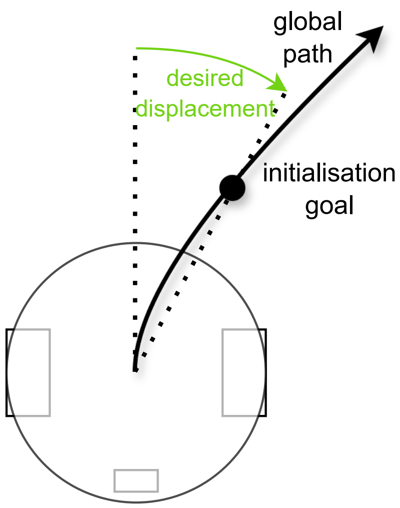
Transition	Condition
S–EI	¬goalReached ∧ ¬nearCollision ∧ newGoal ∧ ¬directedToInitGoal
S–M	¬goalReached ∧ ¬nearCollision ∧ newGoal ∧ directedToInitGoal
EI–S	goalReached ∨ (nearCollision ∧ ¬canRecover)
EI–M	¬goalReached ∧ ¬nearCollision ∧ directedToInitGoal
EI–RR	¬goalReached ∧ nearCollision ∧ canRecover
M–S	goalReached ∨ (nearCollision ∧ ¬canRecover)
M–EI	¬posReached ∧ ¬nearCollision ∧ localGoalBehind
M–OA	¬goalReached ∧ posReached ∧ ¬nearCollision
M–YWC	¬posReached ∧ ¬nearCollision ∧ crossingDetected
M–LA	¬posReached ∧ ¬nearCollision ∧ (oscillating ∨ stuck ∨ globalPathOccluded)
M–RR	¬goalReached ∧ nearCollision ∧ canRecover
OA–S	goalReached ∨ (nearCollision ∧ ¬canRecover)
OA–M	¬posReached
YWC–M	¬nearCollision ∧ ywRoutineEnded
YWC–LA	¬nearCollision ∧ (oscillating ∨ stuck ∨ ywRoutineEnded ∧ globalPathOccluded)
YWC–RR	nearCollision
LA–M	¬nearCollision ∧ ¬oscillating ∧ ¬stuck ∧ ¬globalPathOccluded
LA–RR	nearCollision
LA–EI	laRoutineEnded
RR–M	¬nearCollision
RR–S	nearCollision ∧ rrRoutineEnded

Table 5.3: Description of the state transition conditions. The naming pattern of transition identifiers reflects the initial letters of the current state and the next state. Names of states are mapped as follows: *S* – *Stopped*, *EI* – *Execution Initialisation*, *M* – *Moving*, *OA* – *Orientation Adjustment*, *YWC* – *Yield Way Crossing*, *LA* – *Look Around*, and *RR* – *Rotating and Receding*.

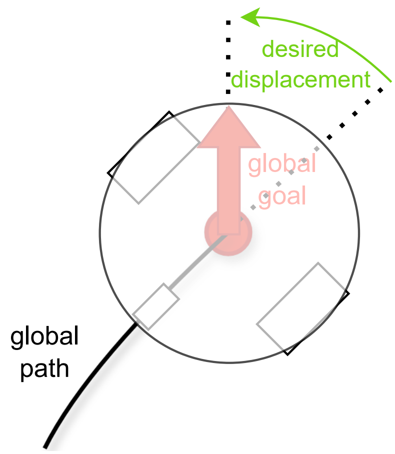
the initial part of the newly computed global path (Fig. 5.4a). Specifically, the vector connecting the robot's position with the position of the initiation goal is determined, and the implemented strategy computes the velocity commands with angular velocities (regarding kinodynamic limits) to align the robot's orientation with the direction of the defined vector (with a tolerance for the target angle identified in Tab. 5.1).

Notably, the *FSM* diagram (Fig. 5.3) identifies the *EI–S* transition, which is applicable when a new goal is at the initial position of the robot, but oriented differently. In such a situation, switching between *Stopped–Execution Initialisation–Stopped* is the indented sequence.

Moving In the behaviour associated with the *Moving* state, the transition function computes velocity commands that perform regular robot's movement toward the global



(a) *Execution Initialisation*



(b) *Orientation Adjustment*

Figure 5.4: Typical scenarios when the robot starts operation in the *Execution Initialisation* and *Orientation Adjustment* states.

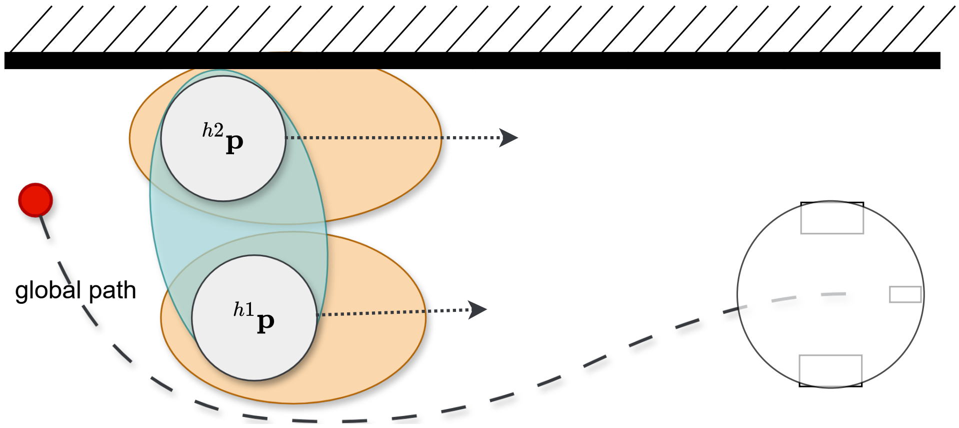


Figure 5.5: Typical scenario of the robot operating in the *Moving* state, where the robot interacts with humans in an unfocused way. Orange ellipses reflect the personal spaces of humans *h1* and *h2*, whereas the green ellipse indicates the O-space of their F-formation.

goal, while avoiding collisions and interacting with the dynamic objects, e.g., humans, in an unfocused way. Such an interaction is related to respecting, i.a., personal zones of individual humans and O-spaces of F-formations while moving through the environment according to the requested navigation task (Fig. 5.5).

The *M–EI* transition (Fig. 5.3) is worth noting, as it is applicable when the robot, during typical task execution receives another request (of a higher priority) with the new goal behind its current facing direction. In such a situation, the transition from the *Moving* to *Execution Initialisation* can be activated to reduce the excessive translational motions when turning back.

The robot's behaviour when operating in the *Moving* state will be further discussed in this chapter.

Orientation Adjustment The *Orientation Adjustment* is active once the robot achieves the goal position, but not orientation; which requires in-place rotation. Therefore, the transition function associated with the behaviour of the *Orientation Adjustment* state performs a similar activity to the one associated with the *Execution Initialisation* state. The only difference is that in the *Orientation Adjustment* the desired orientation is defined by the orientation of the goal pose (Fig. 5.4b). The necessity to perform orientation adjustment before completing the navigation task occurs when the final part of the path to the global goal ends with poses oriented at a significant angle compared to the goal pose's orientation.

Notably, the *OA–M* transition is relevant in a situation when, during the orientation adjustment, the localisation module estimates that the global pose has shifted from the goal position beyond the tolerance distance (the position tolerance is noted in Tab. 5.1), which requires performing an additional translational movement in the *Moving* state.

Yield Way Crossing The transition function associated with the behaviour of the *Yield Way Crossing* state performs a manoeuvre that effectively makes the robot grant the right of way to a human (or another dynamic object) at a junction (**Req. 2.4.5**), which is identified as one of the social norms (Sec. 2.5.5).

The action of allowing priority at a crossing, instead of stopping the robot, relies on finding a "safe pose" behind the human (along its movement direction), which serves as an intermediate goal of the described routine (Fig. 5.6). That intermediate goal is displaced from the human centre according to the radius of the human circular occupancy model, d_{ocp}, multiplied by a parameter (exposed for the user's adjustment).

The *FSM* operation in the *Yield Way Crossing* can start once any human in proximity to the robot is predicted to cross the robot's path. The transition from the *Moving* state to *Yield Way Crossing* is mainly dictated by the `crossingDetected` predicate, whose calculation method is schematically illustrated in Fig. 5.7.

Both the planned trajectory and the global path plan are utilised to determine the predicate value. Using a planned trajectory for calculating the predicate is straightforward, as it constitutes the path the robot will most certainly perform in incoming control steps. However, the robot's path is also considered for detecting human crossing since the entire path to the goal is more suitable for proactively detecting the upcoming situation than a few-second trajectory. On the other hand, the subsequent poses of the trajectory might significantly differ from subsequent poses of the global plan; thus, both sources are evaluated in terms of the human crossing.

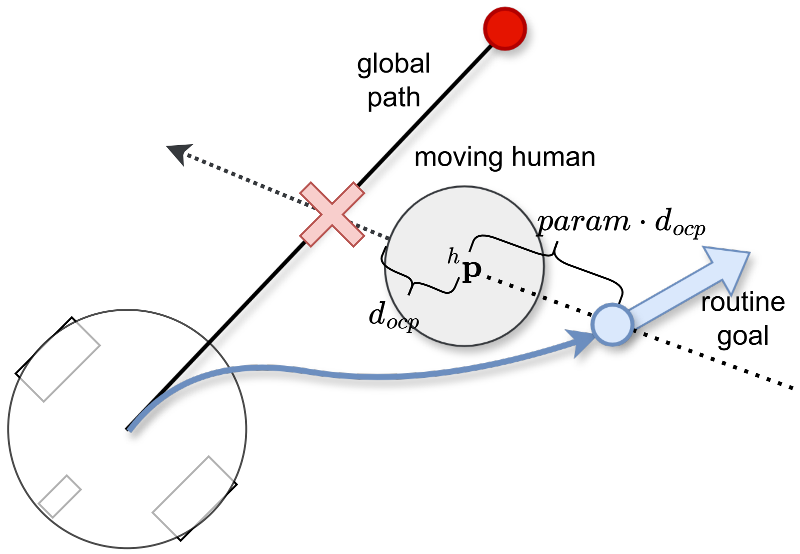


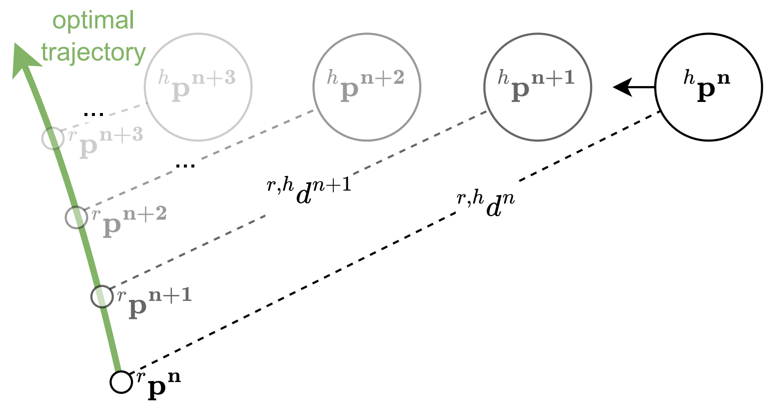
Figure 5.6: Typical scenario when the robot starts operation in the *Yield Way Crossing* state.

Specifically, since the temporal occurrence of trajectory poses is known, the distances, ${}^{r,(\cdot\cdot)}d^{(\cdot)}$, between the subsequent poses of the predicted human trajectories and the corresponding poses of the optimal robot trajectory are computed (Fig. 5.7a). In contrast, due to the lack of temporal knowledge of the robot's path, the distances between predicted human poses and path poses must be evaluated with each other (Cartesian product), as shown in Fig. 5.7b. Then, if the h-th human's occupancy model (assuming that h-th human is the closest in the robot's proximity) overlaps with the robot's footprint at any pose of the trajectory or path, the timestamp of the crossing detection is saved as ${}^{r,h}t^{\mathrm{crs}}$ and further calculations are performed. The space occupied by a human is modelled as a circle with a radius d_{ocp}, whereas the circumradius of the robot's footprint equals d_{cr}. Therefore, the overlap at time t^n occurs when $({}^{r,h}d^n - d_{\mathrm{ocp}} - d_{\mathrm{cr}}) < 0$. The predicate computation method includes the timing factor, ${}^{r}\mathrm{crs}_t^n$, to reflect decreasing confidence of state estimation with successive prediction steps (5.2).[2] Furthermore, a directional factor, ${}^{r,h}\mathrm{crs}_\theta^n$, is calculated to react only to close-to-perpendicular crossings (5.3).[3] Moreover, the frontal location factor is calculated, as the importance of crossing events behind the robot is marginal (5.4).[4] Finally, the logical value of the predicate is computed as in (5.5).

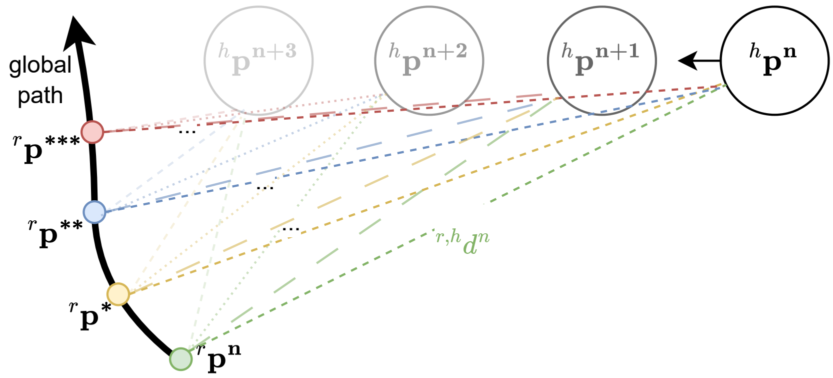
[2] The ${}^{r}\mathrm{crs}_{\mathrm{exp}}$ parameter has been found experimentally and is set to -0.34, which gives approximately 50% confidence of the crossing expected in 2 s prediction.

[3] The ${}^{r}_{\theta}\mathrm{var}_{\mathrm{crs}}$ parameter is determined using the 2-sigma rule applied to the standard deviation of $\pi/4$ rad.

[4] The ${}^{r}_{\mathrm{fr}}\mathrm{var}_{\mathrm{crs}}$ parameter is determined using the 2-sigma rule applied to the standard deviation of $\pi/4$ rad.



(a) Trajectory crossing



(b) Path crossing

Figure 5.7: The principle of determining distances between a human and the robot for the `crossingDetected` predicate calculation. The world state is presented at t^n with the robot's planned optimal trajectory, the global path, and predicted human trajectories. The subsequent predicted states are less certain; hence, the transparency of all poses and the circles representing humans is gradually reduced.

$$^{r,h}\mathrm{crs}_t^n = \exp\left({}^{r}\mathrm{crs}_{\mathrm{exp}} \cdot \left({}^{r,h}t^{\mathrm{crs}} - t^n\right)\right) \tag{5.2}$$

$$^{r,h}\mathrm{crs}_\theta^n = \mathrm{fun}_{\mathrm{ang}}\left({}^{r}\theta^{\mathrm{crs}} - {}^{h}\theta^{\mathrm{crs}},\ \mathcal{N}\left(\begin{cases}\frac{\pi}{2} & \text{if } {}^{r,h}\delta^{\mathrm{crs}} \geqslant 0\\ -\frac{\pi}{2} & \text{otherwise}\end{cases}, {}^{r}_{\theta}\mathrm{var}_{\mathrm{crs}}\right)\right) \tag{5.3}$$

$$^{r,h}\mathrm{crs}_{\mathrm{fr}}^n = \mathrm{fun}_{\mathrm{ang}}\left({}^{r,h}\delta^{\mathrm{crs}},\ \mathcal{N}\left(0, {}^{r}_{\mathrm{fr}}\mathrm{var}_{\mathrm{crs}}\right)\right) \tag{5.4}$$

$$\texttt{crossingDetected} = \begin{cases}1 & \text{if } {}^{r,h}\mathrm{crs}_t^n \cdot {}^{r,h}\mathrm{crs}_\theta^n \cdot {}^{r,h}\mathrm{crs}_{\mathrm{fr}}^n \geqslant 0.55\\ 0 & \text{otherwise}\end{cases} \tag{5.5}$$

Look Around The *Look Around* is activated once the global path has not been received for a few planning cycles (duration parameterised by the user) or the mobile base has not

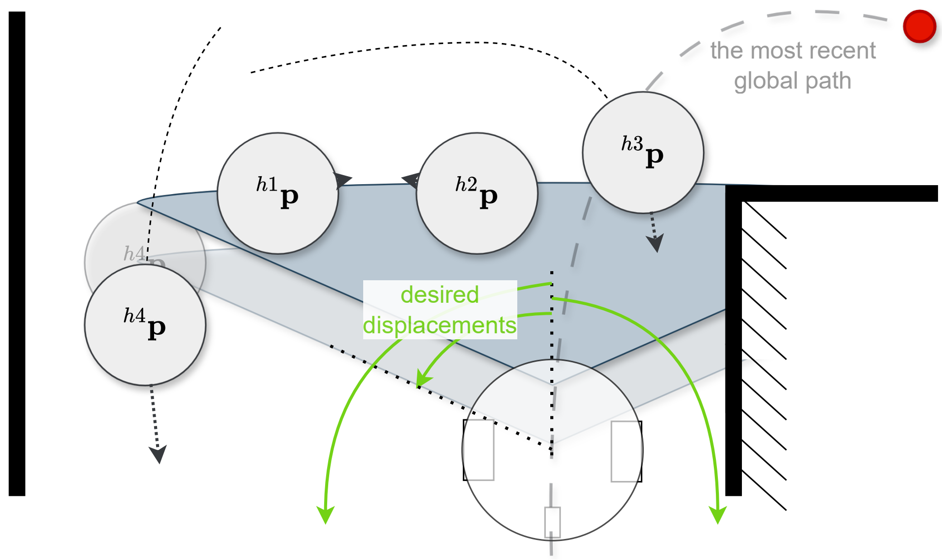


Figure 5.8: Typical scenario when the robot starts operation in the *Look Around* superstate. The cone in front of the robot reflects the *field of view* of its sensor, whereas the "desired displacements" reflect the sequence of rotations to be executed.

developed significant velocities (either linear or angular) for a considerable time (which usually indicates the inability to move towards the global goal). Hence, the transition functions associated with the behaviours of the *Look Around* superstate perform a sequence of actions to update the environment model, which aims to facilitate finding the global path. A typical scenario, when the global path cannot be found is when the robot operates among dynamic objects and its sensors have a limited *field of view*; hence, the environment representation is not fully known and might be partially outdated, which is common in practice.

An example situation when the *Look Around* superstate is active is presented in Fig. 5.8. The scenario shows that the area near the corner in a corridor was not occupied during the last global path planning, but due to the detection of a dynamic object *h3*, the most recent global path of the robot is no longer traversable. Additionally, a valid alternative cannot be found, due to the persisting observation of a dynamic object *h4*, which occupied the space on the robot's left once it approached the corner. Without additional verification, it cannot be determined, whether previously occluded space is still occupied. Therefore, the routine implemented in the *Look Around* performs a short backward movement and a sequence of rotations (Tab. 5.1), while checking for potential collisions. Once the conditions of transition to another state are met (i.e., the global path has been successfully planned), the routine might end earlier.

Rotating and Receding The behaviours implemented in the *Rotating and Receding* state intend to safely escape space configurations, where the robot's inflated footprint (the

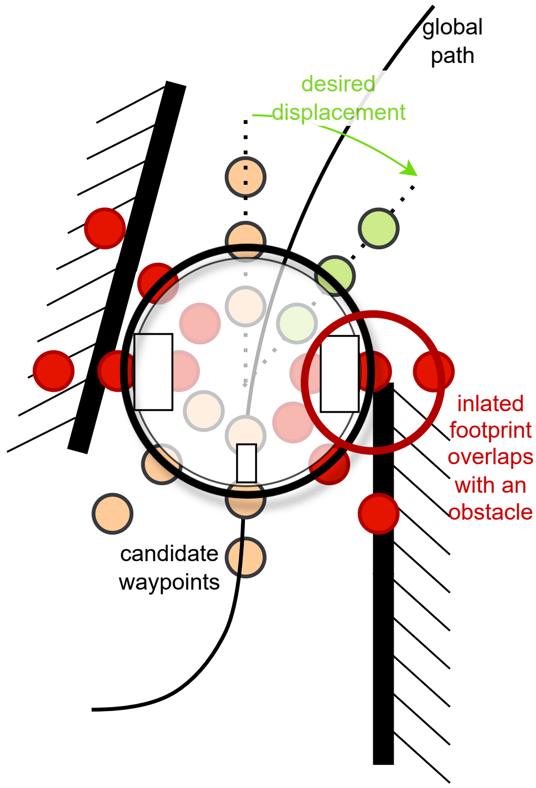


Figure 5.9: Typical scenario when the robot starts operation in the *Rotating and Receding* superstate. The colours of local path candidates correspond to their preference from selecting (green) to rejecting (red).

footprint's inflation emphasises maintaining additional gap from environment objects) is in a collision, but the non-inflated footprint is non-collision. Such a situation might happen when a dynamic object approaches the robot in close proximity or if the robot traverses a narrowing passage. Therefore, the behaviours implemented in the *Rotating and Receding* superstate generate velocity commands that slowly rotate and approach the safe configurations until the inflated footprint is not in collision.

An example of the robot's arrangement in the environment applicable for the *Rotating and Receding* operation is shown in Fig. 5.9, where the robot needs to escape a narrow passage. To accomplish that, a transition function creates local path candidates (0.3 m long) around the robot, which are assessed as entire lines (waypoints separated by distances of the footprint's inflation) regarding free space and lack of collisions along that way. The Fig. 5.9 shows 8 candidate waypoints, whereas during the scenarios performed in the experiments (Chapter 7) 16 potential direction lines were evaluated. Starting from the current direction of the robot's motion, the frontal local paths are the most favourable, the side ones being less, and the rear directions being the least preferable.

Both *Rotating and Receding* and *Look Around* superstates have heuristics implemented to provide more robust operation of the robot operating in dynamic environments. In the remaining part of this chapter, the transition function executed in the behaviour of the *Moving* state is extensively explained.

5.3 Outline of the trajectory planning approach

The *HUMAP* is a geometric planner that solves the problem of receding horizon trajectory planning for dynamical systems operating in unstructured environments. Our planner formulates the objective function regarding navigation requirements from classical and human-aware perspectives. The traditional navigation requirements are implemented as hard constraints, e.g., collision avoidance and the adherence to kinematic and dynamic constraints of a mobile base. In contrast, socially-aware navigation requirements are implemented as soft constraints, e.g., the avoidance of intrusions into personal spaces of surrounding humans [47, 2] or the avoidance of crossing the F-formations' O-spaces [75, 2].

The general idea behind the *HUMAP*'s planning procedure is to develop various feasible (regarding kinodynamic constraints) robot trajectories using the hybrid approach to trajectory generation (Sec. 5.5 and 5.6), then score trajectories based on cost functions assuring collision-free motions, while also considering the robot's performance, motion naturalness, and human discomfort (Sec. 5.7), to finally select a trajectory with the lowest cost, as formalised in (5.6). The symbols commonly used for describing our method are presented in Tab. 5.4.

$$
\begin{aligned}
\min_{i} \quad & \left\{ {}^{r,E}_{i}\mathrm{cost}^{n}_{\mathrm{all}}, \ldots, {}^{r,E}_{\chi+\zeta}\mathrm{cost}^{n}_{\mathrm{all}} \right\} \\
\text{subject to} \quad & {}^{r,E}_{i}\mathrm{cost}^{n}_{\mathrm{all}} = \textsc{scoreTraj}\left({}^{r}\mathrm{traj}^{n}_{i}, {}^{r,E}\mathbf{cfun}^{n}_{\mathrm{all}} \right) \quad \forall i \in \{1, \ldots, \chi+\zeta\} \\
& {}^{r}\mathrm{traj}^{n}_{j} = \textsc{generateTrajectory}\left({}^{r,\mathbb{O}}\mathrm{gen}^{n}_{\mathrm{soc}},\ j \right) \quad \forall j \in \{1, \ldots, \chi\} \\
& {}^{r}\mathrm{traj}^{n}_{k} = \textsc{generateTrajectory}\left({}^{r}\mathrm{gen}^{n}_{\mathrm{smp}},\ k \right) \quad \forall k \in \{1, \ldots, \zeta\}
\end{aligned}
\tag{5.6}
$$

The problem definition in (5.6) explains that for a robot r at a given time step t^n, the $\textsc{generateTrajectory}$ function takes an individual $(\cdot)$-th trajectory generator ${}^{r,(\cdot\cdot)}\mathrm{gen}^{n}_{(\cdot)}$ as an argument and produces a single trajectory, e.g., i-th is denoted by ${}^{r}\mathrm{traj}^{n}_{i}$, from a set of trajectories to create, ${}^{r}\mathbf{traj}^{n}_{(\cdot)}$. Furthermore, the $\textsc{scoreTraj}$ function quantitatively evaluates a given trajectory ${}^{r}\mathrm{traj}^{n}_{(\cdot\cdot)}$ with a set of all implemented cost functions given by ${}^{r,E}\mathbf{cfun}^{n}_{\mathrm{all}}$, that results in a scalarised cost of that trajectory, denoted by ${}^{r,E}_{(\cdot\cdot)}\mathrm{cost}^{n}_{\mathrm{all}}$.

Our novel hybrid method of creating trajectory candidates relies on two independent trajectory generators. The first uses an extended pedestrian motion model that provides prospects with human-like collision avoidance, ${}^{\mathbb{O}}\mathrm{gen}^{n}_{\mathrm{soc}}$ (Sec. 5.5), whereas the second one samples the space of feasible robot velocities[5], $\mathrm{gen}^{n}_{\mathrm{smp}}$ (Sec. 5.6). Two trajectory generators

[5] The sampling is performed directly in the space of admissible controls (feasible velocities), instead of

Symbol	Description
${}^{r}\mathrm{traj}_{(\cdot)}$	$(\cdot)$-th trajectory candidate generated for the r agent
${}^{r,(\cdot\cdot)}\mathrm{gen}_{(\cdot)}$	individual trajectory generator identified as $(\cdot)$ that creates trajectories based on the states of the r agent and entities included in the $(\cdot\cdot)$ set
${}^{r,\mathbb{O}}\mathbf{gen}_{\mathrm{all}}$	list of all trajectory generators of the r agent, consisting of $\left[{}^{r,\mathbb{O}}\mathrm{gen}_{\mathrm{soc}}, {}^{r}\mathrm{gen}_{\mathrm{smp}}\right]$
${}^{r,(\cdot\cdot)}\mathrm{cfun}_{(\cdot)}$	individual cost function identified as $(\cdot)$ which evaluates the r agent's interaction with the $(\cdot\cdot)$ agents along a trajectory that is provided as an argument to the cost function
${}^{r,E}\mathbf{cfun}_{\mathrm{all}}$	list of all cost functions regarded during the trajectory planning for the r agent; cost functions evaluate the state and actions of the r agent, taking into account its interaction with the environment E
${}^{r}\rho_{(\cdot)}$	weight of a cost function identified as $(\cdot)$, related to the r agent
${}^{r,(\cdot\cdot)}_{(\cdot\cdot\cdot)}\mathrm{cost}_{(\cdot)}$	value of a cost function(s) identified as $(\cdot)$, assessing the r agent's interaction with the $(\cdot\cdot)$ agents along the $(\cdot\cdot\cdot)$-th trajectory
$\mathrm{COST}_{\mathrm{infeas}}$	cost value assigned to a trajectory that was considered as leading to a collision or infeasible in terms of kinodynamic constraints of the robot

Table 5.4: Description of symbols used to present the activity of our local trajectory planner.

are used, as the sole pedestrian motion model, based on the force fields with similar features as potential fields, is susceptible to local minima and oscillations, whereas the velocity sampling-based method produces curved trajectory candidates from a discretised control space.

The cardinality of a trajectory set produced by the model-based trajectory generator, ${}^{\mathbb{O}}\mathrm{gen}^{n}_{\mathrm{soc}}$, is χ, while the velocity sampling generator, $\mathrm{gen}^{n}_{\mathrm{smp}}$, develops ζ trajectories. The substantial difference between the employed trajectory generators is that the model-based trajectory generator considers a sparse environment model (the $\mathbb{O}$ symbol indicates that the obstacle set valid at time t^n is considered) to produce trajectories. In contrast, the velocity sampling trajectory generator provides curved trajectories by sampling directly in the space of admissible velocities without regarding environmental constraints.

The activity of our local trajectory planning approach is described in detail with pseudocodes Alg. 1, 2, and 3 using symbols from Tab. 5.4. Specifically, Alg. 1 depicts

the sampling in the configuration space as discussed in (Sec. 3.2.2).

the general receding horizon control scheme. On the other hand, Alg. 2 presents the planning procedure with the search for a trajectory candidate with the lowest cost. The GENERATETRAJECTORIES functions used therein produce sets of trajectories to consider at the given time step. They are defined per each individual trajectory generator, i.e., ${}^{\mathbb{O}}\mathrm{gen}^{n}_{\mathrm{soc}}$ and $\mathrm{gen}^{n}_{\mathrm{smp}}$, as described in Sec. 5.5 and 5.6, accordingly. Furthermore, the scoring procedure implemented in the SCORETRAJ function is thoroughly described in Alg. 3. The criteria for selecting the lowest cost trajectory among the candidates are discussed in detail in Sec. 5.7.

In our approach, trajectory generation and relevant environment state predictions are computed deterministically. Similarly, trajectory scoring uses deterministic cost functions, but human-related ones incorporate Gaussian covariances as perception uncertainties (Sec. 4.4).

Algorithm 1 General robot motion control scheme of the robot r at time t^n

```
function COMPUTECONTROLCOMMAND( )
    ▷ Generate traj. candidates, score them and select
    ▷ a trajectory with the lowest cost
    traj^n_best ← PLANTRAJECTORY( ^𝕆gen^n_all, ^E cfun^n_all )
    ▷ Robotic mobile platforms are velocity-controlled
    ▷ Get the vel. that initialises driving the robot along the traj.
    v^{n+1} ← GETTRAJVELOCITY(traj^n_best)
    return v^{n+1}
end function
```

5.4 Environment representation

A robot operating in populated environments must be equipped with onboard distance and vision sensors to obtain accurate information about surrounding obstacles and humans [402, 149]. An alternative solution might integrate the robot with an external perception system, which, however, requires a reliable communication channel. The social robot's perception modules must provide human pose, velocity, and their covariances, so the reliability of the human track can be estimated.

The *HUMAP*'s planning approach uses a dual environment representation – dense and sparse, each applied for different purposes. The dense representation constitutes a discretised costmap [22, 170] of the robot's environment, which is an extension of the traditional occupancy grid [157] (Sec. 3.1.1). In contrast, the sparse representation provides a concise description of the objects detected in the environment, containing their semantic inform-

Algorithm 2 Finding the lowest cost trajectory among the candidates acquired from generators ${}^{r,\mathbb{O}}\mathbf{gen}^{n}_{\text{all}}$, scored with cost functions ${}^{r,E}\mathbf{cfun}^{n}_{\text{all}}$ of the robot r at time t^n

function planTrajectory$({}^{\mathbb{O}}\mathbf{gen}_{\text{all}}, {}^{E}\mathbf{cfun}_{\text{all}})$
  $\text{cost}_{\text{best}} \leftarrow \infty$ ▷ Expecting costs lower than ∞
  $\text{traj}_{\text{best}} \leftarrow \emptyset$ ▷ Storage for the lowest cost traj.
  for each ${}^{(\cdot)}\text{gen}_i \in {}^{\mathbb{O}}\mathbf{gen}_{\text{all}}$ **do** ▷ Investig. all tr. gen.
    $\mathbf{traj}_i \leftarrow$ generateTrajectories$\left({}^{(\cdot)}\text{gen}_i\right)$
    for each $\text{traj}_j \in \mathbf{traj}_i$ **do** ▷ Investigate each traj.
      ${}^{E}_{j}\text{cost}_{\text{all}} \leftarrow$ scoreTraj$\left(\text{traj}_j, {}^{E}\mathbf{cfun}_{\text{all}}\right)$
      if ${}^{E}_{j}\text{cost}_{\text{all}} == \text{COST}_{\text{infeas}}$ **then**
        continue ▷ Skip invalid trajectory
      end if
      if $\text{cost}_{\text{best}} \leqslant {}^{E}_{j}\text{cost}_{\text{all}}$ **then**
        ▷ Better or equal trajectory already found
        continue
      end if
      $\text{cost}_{\text{best}} \leftarrow {}^{E}_{j}\text{cost}_{\text{all}}$ ▷ Update the lowest cost
      $\text{traj}_{\text{best}} \leftarrow \text{traj}_j$ ▷ Update the best traj.
    end for
  end for
  return $\text{traj}_{\text{best}}$ ▷ Return the lowest cost trajectory
end function

Algorithm 3 Scoring a trajectory candidate ${}^{r}\mathrm{traj}^{n}$ using cost functions ${}^{r,E}\mathbf{cfun}^{n}_{\mathrm{all}}$ of the robot r at time t^n

function $\textsc{scoreTraj}(\mathrm{traj}_i, {}^{E}\mathbf{cfun}_{\mathrm{all}})$
    ${}^{E}_{i}\mathrm{cost}_{\mathrm{all}} \leftarrow 0$ ▷ Initialise value of traj. cost
    for each ${}^{(\cdot)}\mathrm{cfun}_j \in {}^{E}\mathbf{cfun}_{\mathrm{all}}$ **do** ▷ Iter. over cost funs.
        if $\rho_j \leqslant 0$ **then** ▷ Ensure positive cost fun. weight
            continue
        end if
        ${}^{(\cdot)}_{i}\mathrm{cost}_j \leftarrow {}^{(\cdot)}\mathrm{cfun}_j\,(\mathrm{traj}_i)$ ▷ Eval. the cost fun.
        if ${}^{(\cdot)}_{i}\mathrm{cost}_j == \mathrm{COST}_{\mathrm{infeas}}$ **then**
            return $\mathrm{COST}_{\mathrm{infeas}}$ ▷ Skip invalid trajectory
        end if
        ${}^{(\cdot)}_{i}\mathrm{cost}_j \leftarrow \rho_j \cdot {}^{(\cdot)}_{i}\mathrm{cost}_j$ ▷ Factor in the weight
        ${}^{E}_{i}\mathrm{cost}_{\mathrm{all}} \leftarrow {}^{E}_{i}\mathrm{cost}_{\mathrm{all}} + {}^{(\cdot)}_{i}\mathrm{cost}_j$ ▷ Sum up
    end for
    return ${}^{E}_{i}\mathrm{cost}_{\mathrm{all}}$ ▷ Return the total cost of the trajectory arising from individual cost functions
end function

ation with geometric attributes. Types of objects extracted from the robot's environment are listed in the ontology in Sec. 5.1.

There are direct reasons why a dual environment representation is implemented. Firstly, the dense representation, inherited from the classical robot navigation, aggregates information about the obstacles in the robot's environment over time and provides such data even if the current *field of view* of onboard sensors does not allow observing those obstacles. Costmaps typically embody the environment in a 2D plane; however, projecting sensor readings, mainly from *RGB-D* cameras, onto the robot's base plane allows for the representation of the environment in so-called 2.5D [403]. Additionally, the layered architecture of the costmap [170] allows embedding contextual information, e.g., proxemics [47], into the environment model used for navigation. The resultant costmap with enriched information is flattened for motion planning, so it can be used for calculating the cost of the robot's traversal through the costmap cells. On the other hand, the sparse representation is required to compute the controls according to the pedestrian motion model governing the trajectories produced by the ${}^{\mathbb{O}}\mathrm{gen}^{n}_{\mathrm{soc}}$ generator. The model computes pairwise interactions between the robot and other objects; hence, all environment objects must be segmented and their spatial attributes estimated.

While the human data are obtained directly from the perception modules [149], the segmented obstacle data are extracted from the flattened costmap[6] (as in [162]), which contains all types of obstacles in the environment without semantic distinction. Therefore, a procedure of excluding social agents, $\mathbb{H}^n$, from all other obstacle types, $\mathbb{O}^n$, was developed.

Unlike other applications of the pedestrian motion model [404], the implementation of *HUMAP* does not assume that there are multiple point obstacles in the environment, but the real forms of objects are estimated by processing obstacles marked in the costmap. Overall, environment objects in the sparse representation are modelled with: circles, lines, and polygons, ensuring that the algorithm applies to a real-world operation. Also, most approaches integrating the pedestrian motion model treat humans as points representing the centre of a body [138, 193], but our sparse representation treats humans as objects that physically occupy some space. Hence, for obstacle avoidance in the model-based trajectory generator, ${}^{\mathbb{O}}\mathrm{gen}^{n}_{\mathrm{soc}}$, humans, represented by static or dynamic objects, are modelled as circles with a radius of d_{ocp}, and the closest points between each human and the robot are determined in each time step.

Social robots operate in highly dynamic environments; therefore, motion anticipation of surrounding objects is crucial for efficient navigation. In the recent work, Schöller et al. [218] compared sophisticated state-of-the-art human trajectory prediction methods against the constant velocity model, and they found that this simple approach can yield similar results. As a consequence of their inspiring outcomes, the *HUMAP* has the constant velocity assumption implemented to forecast trajectories of all entities distinguished in the ontology. Finally, a joint state space of the robot and all objects is developed, as the object's trajectory prediction step is equal to the robot's trajectory planning step, t_{Δ}, and the object's motion prediction horizon is equal to the robot's trajectory planning horizon, t_{hor}.

5.5 Trajectory generation using the pedestrian motion model

The recent successful real-world applications of the *Social Force Model* (*SFM*) [275, 314, 315, 165, 253, 166], which is the prevalent pedestrian motion model, have inspired us to incorporate this method for robot trajectory generation. The baseline *SFM* model was extended with an additional component based on *Fuzzy Inference System* (*FIS*) to engage social rules of pedestrian motions and to enhance realistic collision avoidance behaviours. The reactive baseline *SFM* approach, integrated with the proactive *FIS*-based compon-

[6] `https://wiki.ros.org/costmap_converter`

ent, creates the *Fuzzy-Extended Social Force Model* (*FESFM*) proposed in this work. The *FESFM* has been employed in the pedestrian-motion-model-based *social trajectory generator*, denoted as $^{\mathbb{O}}\text{gen}^{n}_{\text{soc}}$, providing valid trajectory candidates for the *HUMAP*'s trajectory planning scheme. The trajectory candidates produced by the $^{\mathbb{O}}\text{gen}^{n}_{\text{soc}}$ are later scored (along with the candidates from the second generator explained in Sec. 5.6) by cost functions to obtain the optimal solution according to the objective function described in Sec. 5.7.

The remaining part of this section discusses the most common pedestrian motion models and the calibration of the *SFM* parameters. Furthermore, the formulation of the baseline model is disclosed in Sec. 5.5.4, and the proposed *SFM* extension, introduced in the *FESFM*, is described in Sec. 5.5.5.

5.5.1 Pedestrian motion models

The modelling of pedestrian dynamics has been an active field since the 1970s. Studies focus either on a macroscopic approach that investigates the movement pattern of a whole collective (a crowd) or on a microscopic approach, which examines the behaviour of individuals [1]. Employing pedestrian dynamics models to reproduce typical human movement behaviours in robot navigation systems primarily focuses on microscopic methods.

Schadschneider et al. [405] reviewed classical models describing pedestrian motion. Multiple algorithms employed stimulus-based approaches to model human dynamics, e.g., *Cellular Automata* (*CA*) [406, 407] or *Social Force Model* (*SFM*) [1]. Both *CA* and *SFM* are microscopic approaches, but the substantial difference between these two methods lies in the background of interactions. Interactions in *CA* are implemented as rules – often motivated by arguments from psychology [405], whereas *SFM*-based models define interactions directly on a level of motion equations, similarly to the classical mechanics. Another difference is related to the continuity – *CA* is a discrete method, while *SFM*-based approaches expose a continuous formulation.

The *SFM* is one of the most prominent models describing pedestrians' motion due to its easily extendable, parameterised method of capturing a mutual influence of individual pedestrians. The *SFM*'s flexible formulation allows the development of specialised methods that include additional factors into the model, as discussed in (Sec. 3.2.2). The general idea behind the *SFM* is to define social analogues of physical forces, e.g., attractive or repulsive interactions, frictional forces, dissipation, and fluctuations [317], and embed them into motion equations. For many years the *SFM* has been commonly used for simulating the evacuation in mass events [313] or pedestrian crossing analyses [408].

The *SFM*'s concept is similar to the *Artificial Potential Field* approach [101] representing the environment as a potential field with attractive and repulsive potentials that guide the agent's movement, albeit without focusing on social interactions.

5.5.2 Social Force Model formulation

This work synthesises the research on the topic of *SFM*, which exhibits a broad naming and symbol diversity among numerous works [1, 56, 138, 408, 25]. Therefore, the unification of symbols used in mathematical formulations has been developed. Tab. 5.5,[7] complementary to the Tab. 4.1, contains common symbols, while others are explained at their respective occurrences.

Symbol	Description
r	ego-agent for which the force is calculated; here, ego-agent is the controlled robot
$\mathbb{O}$	set of generic obstacles, with o-th obstacle identified as ${}^{o}O$
$\mathbb{K}$	set of static obstacles, with k-th static obstacle identified as ${}^{k}K$
$\mathbb{J}$	set of dynamic obstacles, with j-th dynamic obstacle identified as ${}^{j}J$
${}^{r,(\cdot)}\boldsymbol{f}$	resultant force calculated for the robot r; obtained from the model that takes into account entities given by the list $(\cdot)$, e.g., "$\mathbb{O},\mathbb{L}$"
${}^{r}\boldsymbol{f}_{\text{des}}$	acceleration force that attracts the robot r straight towards its goal position
${}^{r,o}\boldsymbol{f}_{\text{rep}}$	repulsive force exerted by the o-th obstacle on the robot r
${}^{r,j}\boldsymbol{f}_{\text{dyn}}$	repulsive force exerted by the j-th dynamic obstacle on the robot r
${}^{r,k}\boldsymbol{f}_{\text{stat}}$	repulsive force exerted by the k-th static obstacle on the robot r
${}^{r,j}\boldsymbol{f}_{\text{beh}}$	*social behaviour* force exerted by the j-th social agent on the agent r
${}^{r,(\cdot)}\breve{\boldsymbol{f}}_{(\cdot\cdot)}$	$(\cdot\cdot)$-type force exerted by the $(\cdot)$-th object on the robot r, corrected with the *field of view* factor, indicated by ${}^{r,(\cdot)}\text{fov}$
${}^{r}m$	mass of the object r
${}^{r,(\cdot)}\boldsymbol{d}$	vector connecting the closest points of r and $(\cdot)$ objects, directed towards the position of $(\cdot)$
${}^{r,(\cdot)}\phi$	direction of the ${}^{r,(\cdot)}\boldsymbol{d}$ vector (defined for the description's conciseness)
${}^{r,(\cdot)}\delta$	relative location of $(\cdot)$ compared to the heading of r
${}^{r}\boldsymbol{e}$	unit vector directed from the current position of the robot r towards its goal position
v_{des}	baseline model's parameter reflecting the desired speed of the ego-agent

[7] As stated in Sec. 5.1, the robot-centric notation is used in the Chapter 5, while the notation in the Chapter 4 investigates the human-centric perspective. Therefore, for clarity and conciseness, some symbols influenced by the state of multiple entities have been reintroduced in the table describing symbols appearing in Chapter 5.

Symbol	Description
A_n	baseline model's parameter affecting the strength of ego-agent's deceleration caused by the repulsive forces exerted by dynamic obstacles
B_n, C_n	baseline model's parameters affecting the range of deceleration components of the force exerted by dynamic obstacles on the ego-agent
A_p	baseline model's parameter affecting the strength of ego-agent's evasive movement caused by the repulsive forces exerted by dynamic obstacles
B_p, C_p	baseline model's parameters affecting the range of evasive components of the force exerted by dynamic obstacles on the ego-agent
A_w	baseline model's parameter affecting the strength of repulsive forces exerted by static obstacles affecting the ego-agent's movement
B_w	baseline model's parameter affecting the range of repulsive forces exerted by static obstacles affecting the ego-agent's movement
A_s	extended model's parameter affecting the strength of *social behaviour* forces caused by social agents'

Table 5.5: Common symbols used for describing the pedestrian motion model.

The original *SFM* has its distinctive ontology consisting of the agent of interest, i.e., ego-agent denoted by r, obstacles $\mathbb{O}$ (static, $\mathbb{K}$, and dynamic $\mathbb{J}$ can be distinguished), and attractive objects, $\mathbb{L}$, not being a movement goal for the ego-agent. When the *SFM* is applied for robot navigation (as in the *HUMAP*'s case), the ego-agent is represented by the robot for which the *social force* is calculated. All *SFM* formulas indicated in this work regard a 2D problem without taking the orientation of objects into account, and all data are expressed in the global frame.

The *social force* originates from Newton's second law (5.7). The method exploits a vector field idea assigning a force vector, ${}^{r}\boldsymbol{f}^{n}$, to an investigated object r that affects its acceleration, ${}^{r}\boldsymbol{a}^{n}$ [1].

$$ {}^{r}\boldsymbol{f}^{n} = {}^{r}m \cdot {}^{r}\boldsymbol{a}^{n} \tag{5.7} $$

Moreover, the original *SFM* formulation also includes a nondeterministic component, ${}^{\text{noise}}\boldsymbol{f}^{n}$, that can be associated with a process noise, which also influences the velocity of the robot r at time t^n (5.8).

$$ \frac{d\,{}^{r}\boldsymbol{v}^{n}}{d\,t}\,{}^{r}m = {}^{r}\boldsymbol{f}^{n} + {}^{\text{noise}}\boldsymbol{f}^{n} \tag{5.8} $$

The mass of the ego-agent, ${}^{r}m$, is known, but the masses and encountered obstacles $\mathbb{O}$ are usually elided, i.e., a unit value is presupposed, ${}^{o}m = 1$ kg, e.g., in [1, 408, 25]. In most *SFM* approaches, all objects are modelled as single points in a two-dimensional space.

The general formulation of the *social force* vector is defined in (5.9). The resultant force vector, ${}^{r,\mathbb{O},\mathbb{L}}\boldsymbol{f}^n$, is a sum of the acceleration term and interaction components – repulsive and attractive. The resultant interaction component for the r robot at time t^n is a sum of interaction forces generated by all obstacles, $\mathbb{O}^n$, and attractive objects, $\mathbb{L}^n$, located within the space of interest.

$$
{}^{r,\mathbb{O},\mathbb{L}}\boldsymbol{f}^n = \underbrace{{}^{r}\boldsymbol{f}^n_{\text{des}}}_{\text{acceleration term}} + \underbrace{\sum_{o\,\in\,\mathbb{O}^n} {}^{r,o}\boldsymbol{f}^n_{\text{rep}}}_{\text{repulsive term}} + \underbrace{\sum_{l\,\in\,\mathbb{L}^n} {}^{r,l}\boldsymbol{f}^n_{\text{attr}}}_{\text{attractive term}} \tag{5.9}
$$

The acceleration term (5.10) describes the ideal force driving the ego-agent r towards its target point [1, 56, 25]. The goal position determines the magnitude and the direction of the ideal velocity vector, ${}^{r}\boldsymbol{e}^n$. Besides that, the acceleration force is also affected by the current velocity of the ego-agent, ${}^{r}\boldsymbol{v}^n$, and the relaxation time, τ, indicating the duration required for the ego-agent to adjust its velocity to match the desired velocity or to react to changes in the environment. Since the *SFM* is designed for pedestrian motion simulation, the formulation parameters assigned to the robot agent, e.g., relaxation time or desired speed, correspond to pedestrian attributes to mimic their motion [138, 408].

$$
{}^{r}\boldsymbol{f}^n_{\text{des}} = \frac{1}{\tau}\left({}^{r}v^n_{\text{des}} \cdot {}^{r}\boldsymbol{e}^n - {}^{r}\boldsymbol{v}^n\right){}^{r}m \tag{5.10}
$$

People generally treat static objects differently, compared to dynamic ones, e.g., other humans [12]; hence, the agent's interaction with the static and dynamic objects should differ. Therefore, researchers divide the repulsive component into two separate parts [56, 25]. The mathematical formulation of the resultant *SFM* structure is presented in (5.11).

$$
{}^{r,\mathbb{O},\mathbb{L}}\boldsymbol{f}^n = {}^{r}\boldsymbol{f}^n_{\text{des}} + \sum_{j\,\in\,\mathbb{J}^n} {}^{r,j}\boldsymbol{f}^n_{\text{dyn}} + \sum_{k\,\in\,\mathbb{K}^n} {}^{r,k}\boldsymbol{f}^n_{\text{stat}} + \sum_{l\,\in\,\mathbb{L}^n} {}^{r,l}\boldsymbol{f}^n_{\text{attr}} \tag{5.11}
$$

Due to the diversity of interaction force formulations [1, 56, 409, 138, 408], in this work, only the equations of the baseline model employed in the *HUMAP* will be further discussed in Sec. 5.5.4.

5.5.3 Social Force Model parameter calibration

The *SFM* method stands for a parameterised mathematical formulation of processes observed in the real world. Numerous representations have been proposed to differentiate the original model [1], which used an elliptical formulation for repulsive forces produced by dynamic obstacles and a circular one for the forces exerted by static obstacles. Since multiple specialised models have been proposed, a significant part of the *SFM* research is related to the calibration of those models, i.e., the search for parameter values that provide the best approximation of pedestrian motion captured during real-world experiments, usually in video tracking footages [56, 25] or robot sensor data recordings [253].

For example, Johansson et al. [56] verified the circular and different elliptical formulations for repulsive forces. Their calibration was performed based on video tracking data and then experimentally validated in simulation scenarios. In contrast, Moussaïd et al. [138] conducted a set of controlled experiments with pedestrians performing simple avoidance tasks. They calibrated a static obstacle collision avoidance and interaction behaviours modelled by the *SFM* formulation.

Seer et al. [25] obtained human movement trajectories based on real-world pedestrian traffic data. They used the video sequences to calibrate parameters of 3 different *SFM* formulations and determined the accuracy of each model. In another work, Taherifar et al. [410] proposed a macroscopic framework for calibration and validation of the *SFM* for bidirectional pedestrian streams. They managed to reproduce desired macroscopic features while still generating microscopic emergent self-organisation and lane formation phenomena. Their framework was benchmarked with the use of the pedestrian macroscopic fundamental diagram [411] that aims to define an ideal relation between pedestrian density and their spatial flow [405]. Nevertheless, the authors of [25] stated that treating the fundamental diagram as a reference to find microscopic model parameters can lead to unexpected results.

Repiso et al. [253] attempted to tune the *SFM* offline based on the recordings of people walking in a side-by-side formation. However, the results of their optimisation scheme, minimising squared distances between the subsequent real-world human poses and poses obtained from the model, still required some manual fine-tuning. In contrast, Ferrer et al. [315] exploited online feedback from experiment participants to dynamically tune the *SFM* parameters for a specific scenario.

The *HUMAP*'s approach to deal with the parameter estimation uncertainties and exploiting that fact to produce multiple trajectory candidates is detailed in Sec. 5.5.7.

5.5.4 Pedestrian motion baseline model

Our approach focuses on implementing *SFM* for generating feasible trajectories for a robot. The *SFM* is a deterministic method providing collision avoidance and emerging the agent's behaviour to realistic motions. Multiple extensions of the original *SFM* have been proposed focusing on the microscopic perspective [205, 412, 44, 413, 167].

As the baseline pedestrian motion model, we use the formulation proposed in Seer et al. [25], who obtained human movement trajectories from video sequences of real-world pedestrian traffic and used these data to calibrate parameters of different *SFM* formulations. Specifically, we rely on their *Model C*, as it discriminates the influence of interaction with static and dynamic objects, in contrast to the original formulation [1]. However, the substantial advantage of the pedestrian motion model selected as the baseline (5.12) is that its parameter values were estimated to exhibit the best fit to the real-world calibration data.

$$ {}^{r,\mathbb{O}}\breve{\boldsymbol{f}}^{n}_{\text{bsl}} = {}^{r}\boldsymbol{f}^{n}_{\text{des}} + \sum_{j \in \mathbb{J}^n} {}^{r,j}\breve{\boldsymbol{f}}^{n}_{\text{dyn}} + \sum_{k \in \mathbb{K}^n} {}^{r,k}\breve{\boldsymbol{f}}^{n}_{\text{stat}} \tag{5.12} $$

The model designated as the baseline (5.12) neglects the impact of attractive objects in the environment, as only task-focused robot operation (without distractors) is investigated. Also, the noise component appearing in the original formulation (5.8) [1] is not included, since the non-determinism is regarded by differentiating model parameters, as explained in Sec. 5.5.7.

Breves above symbols in (5.12) indicate that the *field of view* (*FOV*) factor, computed for each object that the robot interacts with, is already included. Different *FOV* factor forms were proposed in the literature [56, 205], but *HUMAP* implements a customised one presented in Sec. 5.5.5.

The formulation denoted by *Model C* in [25] used the standardised acceleration term, ${}^{r}\boldsymbol{f}_{\text{des}}$, pointed out in (5.10), but takes into account different methods for generating repulsive forces depending on the type of obstacle. Namely, an interaction force exerted by a static obstacle k onto the robot r, indicated by ${}^{r,k}\boldsymbol{f}^{n}_{\text{stat}}$, is developed according to the elliptical specification from [409], presented in (5.13).

$$ {}^{r,k}\boldsymbol{f}^{n}_{\text{stat}} = A_w \cdot e^{-\frac{{}^{r,k}w^n}{B_w}} \cdot \frac{\left\| {}^{r,k}\boldsymbol{d}^n \right\| + \left\| {}^{r,k}\boldsymbol{d}^n - {}^{r,k}\mathbf{ds}^n \right\|}{2\,{}^{r,k}w^n} \cdot \frac{1}{2} \left(\frac{{}^{r,k}\boldsymbol{d}^n}{\left\| {}^{r,k}\boldsymbol{d}^n \right\|} + \frac{{}^{r,k}\boldsymbol{d}^n - {}^{r,k}\mathbf{ds}^n}{\left\| {}^{r,k}\boldsymbol{d}^n - {}^{r,k}\mathbf{ds}^n \right\|} \right) {}^{k}m \tag{5.13} $$

The semi-minor axis ${}^{r,k}w^n$ of the elliptical formulation is given by (5.14), whereas the dynamics of objects (originally, the step size of pedestrians [409]) is taken into account by the ${}^{r,k}\mathbf{ds}^n$, computed as in (5.15).

$$ {}^{r,k}w^n = \frac{1}{2}\sqrt{\left(\left\| {}^{r,k}\boldsymbol{d}^n \right\| + \left\| {}^{r,k}\boldsymbol{d}^n - {}^{r,k}\mathbf{ds}^n \right\| \right)^2 - \left\| {}^{r,k}\mathbf{ds}^n \right\|^2} \tag{5.14} $$

$$ {}^{r,k}\mathbf{ds}^n = \left({}^{k}\boldsymbol{v}^n - {}^{r}\boldsymbol{v}^n \right) \cdot \left(t^n - t^{n-1} \right) \tag{5.15} $$

In contrast, a force generated by a dynamic obstacle j, represented by ${}^{r,j}\boldsymbol{f}^{n}_{\text{dyn}}$, is calculated based on the findings from [408], where a variant combining two separate, distinctively scaled forces was proposed, as described in (5.16).

$$
{}^{r,j}\boldsymbol{f}^{n}_{\text{dyn}} = \left(- {}^{r}\hat{\boldsymbol{x}}^{n} \underbrace{A_n \exp\left(\frac{-B_n \left({}^{r,j}\delta^{n}\right)^2}{{}^{r,j}v^{n} - C_n \left\| {}^{r,j}\boldsymbol{d}^{n} \right\|} \right)}_{\text{deceleration scale}} + {}^{r,j}\hat{\boldsymbol{y}}^{n} \underbrace{A_p \exp\left(\frac{-B_p \left|{}^{r,j}\delta^{n}\right|}{{}^{r,j}v^{n} - C_p \left\| {}^{r,j}\boldsymbol{d}^{n} \right\|} \right)}_{\text{evasion scale}} \right) \cdot {}^{j}m \tag{5.16}
$$

The ${}^{r,j}\boldsymbol{f}^{n}_{\text{dyn}}$ depends, i.a., on the relative location, ${}^{r,(\cdot)}\delta$, of the j-th dynamic obstacle, compared to the heading of r. In turn, the relative location depends on the ${}^{r,j}\phi$, i.e., the direction of the vector connecting the positions of the robot r and the investigated object j (5.17).[8] The spatial attributes, namely ${}^{r,j}\delta$ and ${}^{r,j}\phi$, are exemplified and visualised in Sec. 5.5.5. Overall, the relative location is the angular difference between the ${}^{r,j}\phi$, and the robot orientation angle, ${}^{r}\theta$, calculated as in (5.18).[9]

$$
{}^{r,j}\phi = \arctan 2 \left({}^{r,j}d_y, {}^{r,j}d_x\right) \tag{5.17}
$$

$$
{}^{r,j}\delta = {}^{r,j}\phi - {}^{r}\theta \tag{5.18}
$$

The first force in (5.16) points in the direction opposite to the unit vector aligned with the motion direction of r, i.e., ${}^{r}\hat{\boldsymbol{x}}$ (5.19). On the other hand, the second force, ${}^{r,j}\hat{\boldsymbol{y}}$, is perpendicular to the first one and points away from the dynamic obstacle j, as revealed in (5.20). The resultant repulsive force ${}^{r,j}\boldsymbol{f}_{\text{dyn}}$ exerted by the dynamic obstacle j on the robot r is the superposition of the two described forces.

$$
{}^{r}\hat{\boldsymbol{x}}^{n} = \left[\cos\left({}^{r}\theta^{n}\right), \sin\left({}^{r}\theta^{n}\right)\right]^T \tag{5.19}
$$

$$
{}^{r,j}\hat{\boldsymbol{y}}^{n} = \begin{cases} \left[\cos\left({}^{r}\theta^{n} + \frac{\pi}{2}\right),\ \sin\left({}^{r}\theta^{n} + \frac{\pi}{2}\right)\right]^T, & \text{if } {}^{r,j}\delta^{n} < 0 \\ \left[\cos\left({}^{r}\theta^{n} - \frac{\pi}{2}\right),\ \sin\left({}^{r}\theta^{n} - \frac{\pi}{2}\right)\right]^T, & \text{otherwise} \end{cases} \tag{5.20}
$$

Notably, the *SFM* scheme involves the computation of the distance vector between objects, e.g., the robot r and a static obstacle k, denoted by ${}^{r,k}\boldsymbol{d}^{n}$. In the *HUMAP* implementation, the spatial attributes (shapes) of objects are estimated; thus, the closest points between the robot and each environment object (static or dynamic obstacles, e.g., humans) are determined in each time step, in contrast to [404], where only body centres are considered.

[8] The robot-centric notation is used in (5.17), while the analogous equation appears in the Sec. 4.4.1, specifically, (4.15), investigating the human-centric perspective.

[9] The robot-centric notation is used in (5.18), while the analogous equation appears in the Sec. 4.4.1, specifically, (4.16), investigating the human-centric perspective.

The equations describing the baseline pedestrian model, i.e., (5.10) and (5.13)–(5.16), contain the calibrated parameter set, $\boldsymbol{\rho}_{\text{bsl}}$, pointed out in (5.21). In this work, the symbols of individual parameters are in coherence with the original notation from [25]. Symbols commonly used in the considerations are identified in Tab. 5.5.

$$\boldsymbol{\rho}_{\text{bsl}} \in \Big\{ v_{\text{des}}, A_n, B_n, C_n, A_p, B_p, C_p, A_w, B_w \Big\} \tag{5.21}$$

5.5.5 Customisation of the baseline pedestrian motion model

The *HUMAP* aims to introduce customary conflict avoidance behaviours among humans [28] into a robot navigation scheme. For this purpose, the baseline *SFM*-based pedestrian motion model (5.12) was extended with the new component. The novel term emphasises motions that increase intent expressiveness inspired by pedestrian cues and enhances the motion legibility [23]. Another contribution to the baseline model is proposing a customised *FOV* factor, which is simpler and more realistic than the forms proposed in the literature [56, 205], allowing us to consider the actual *FOV* present in humans [414].

The new component of the extended pedestrian motion model, *FESFM*, implements a decision-making strategy for the robot interacting with dynamic entities $\mathbb{J}$, e.g., humans and other robots. The scheme integrates motion behaviours, such as passing on the right into the robot's movement pattern. A crucial feature of reproducing human-like conflict avoidance behaviours relies on taking customary rules of pedestrian motion into account to develop a heuristic that mimics customary behaviours. A common approach to solving rule-based problems is the use of *FIS*, which has already been successfully implemented in complex robot navigation strategies [305, 306, 307].

The Mamdani model [415] has been employed in the novel term of the *FESFM* to incorporate the decision-making strategy in the robot motion pattern explicitly. The main goal is to detect specific situations that involve human-robot unfocused interaction and proactively react in a socially compatible manner to prevent conflicts. The *FIS* module takes two inputs to produce the output used to compute an additional term of the pedestrian motion model.

The formulation of the new *FESFM* motion model, extending the baseline pedestrian motion model (5.12) with the new fuzzy-inference-based social behaviour term, ${}^{r,j}\boldsymbol{f}_{\text{beh}}$, is presented in (5.22) (the *FOV* factor included).

$${}^{r,\mathbb{O}}\breve{\boldsymbol{f}}^{n}_{FESFM} = {}^{r,\mathbb{O}}\breve{\boldsymbol{f}}^{n}_{\text{bsl}} + \sum_{j \in \mathbb{J}^n} {}^{r,j}\breve{\boldsymbol{f}}^{n}_{\text{beh}} \tag{5.22}$$

Customised FOV factor The *field of view* (*FOV*) factor in the original formulation represents a limited range of human perception that causes objects behind a human agent to have less impact on his movement than objects in front. In *HUMAP*, this feature is transferred to a robot's behaviour.

The *FOV* factor, ${}^{r,(\cdot)}\text{fov}^{n}$, by which the model components are multiplied, is computed as in (5.23)[10] and indicates the scale of $(\cdot)$-th object's influence on r robot's motion. The ${}^{r}\text{fun}_{\text{ang}}$ function (explained in B) computes the value of the univariate Gaussian distribution at the of point of the relative location ${}^{r,j}\delta$ (5.18), where the Gaussian distribution is defined as $\mathcal{N}\left(0, {}^{r}\text{var}_{\text{fov}}\right)$ and is appointed in the angle domain. Specifically, the mean of the Gaussian is the angle of the robot's view axis (in the local coordinate system), which equals 0, while the shape of the distribution is defined by the robot's *FOV* variance, computed based on the ${}^{r}\text{fov}$ parameter with, e.g., the 2-sigma rule applied, that replicates the limited *FOV* of humans.

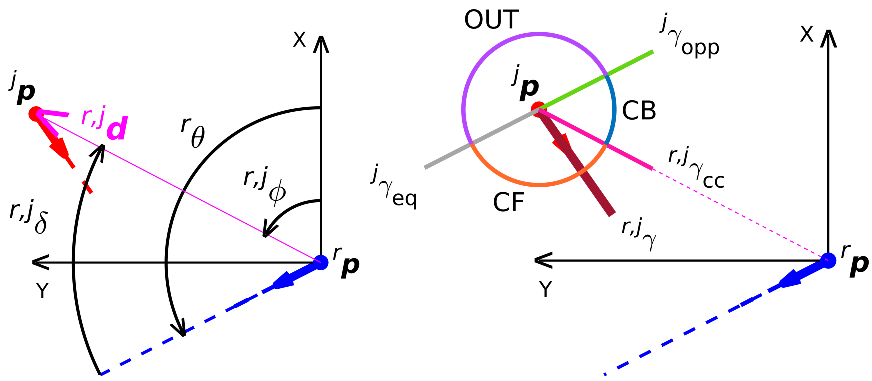
$$ {}^{r,(\cdot)}\text{fov}^{n} = \text{fun}^{n}_{\text{ang}}\left({}^{r,(\cdot)}\delta^{n},\ \mathcal{N}\left(0, {}^{r}\text{var}_{\text{fov}}\right)\right) \tag{5.23} $$

FIS input variables Defining a proper set of variables for a *FIS* is crucial to differentiate environment states to detect specific situations involving an interaction. Although social rules of pedestrian motion are driven by many causes, we state that two environmental factors are crucial for a moving human in their decision process of selecting the movement actions. Both developed input variables are expressed in the angle domain, which is illustrated by corresponding membership functions shown in Fig. 5.11 and Fig. 5.12. In the following considerations, all symbols are expressed for time t^{n}, and the j-th dynamic object can be identified as a moving human or a moving robot (different from the controlled one r).

The first *FIS* factor, indicated by ${}^{r,j}\delta$ (5.18), is a location of the j-th dynamic object in relation to the heading direction of the robot r, which is denoted as ${}^{r}\theta$ (Fig. 5.10a). The following regions for the relative location input variable were distinguished: *front* (F), *front-left* (FL), *back-left* (BL), *back* (B), *back-right* (BR), *front-right* (FR). The membership function with each region is presented in Fig. 5.11.

The second *FIS* factor, ${}^{r,j}\gamma$, determines the location of the intersection point of r's and j's direction rays relative to the r's centre, as illustrated in Fig. 5.10b. The intersection point is determined assuming that the robot r is stationary, and the dynamic object j moves with a constant velocity. Therefore, the fuzzification regions describing the possible ${}^{r,j}\gamma$ values are distinguished as: *cross-centre* (CC), *cross-behind* (CB), *opposite* (OPP), *outwards* (OUT), *equal* (EQ), *cross-in-front-of* (CF), as depicted in Fig. 5.10b.

[10] The robot-centric notation is used in (5.23), while the analogous equation appears in the Sec. 4.4.3, specifically, (4.34), investigating the human-centric perspective.



(a) Spatial arrangement (b) Regions and border angles of ${}^{r,j}\gamma$

Figure 5.10: (a) Geometric attributes of an example spatial arrangement of r and j agents along with lines reflecting the orientation of the global coordinate system. (b) Fuzzification regions and the angles associated with region borders defined for the second input variable of the *FIS*. Note that (a) and (b) illustrate the same spatial arrangement of r and j.

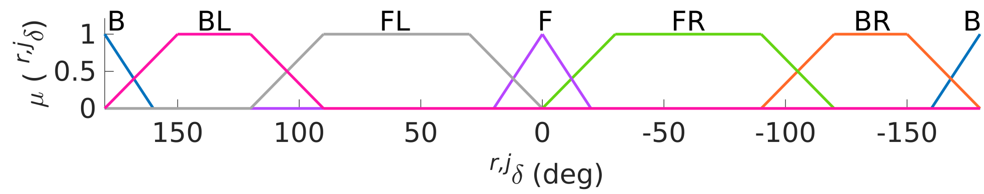


Figure 5.11: A membership function of the first *FIS* input variable ${}^{r,j}\delta^n$

Regions of the second input variable, in contrast to the first one, are dynamically arranged (5.25); hence, must be computed in the context of the observed spatial arrangement of r and j. For a straightforward geometrical interpretation, the value of ${}^{r,j}\gamma$ is referenced to the orientation of a dynamic agent j, i.e., ${}^{r,j}\gamma = {}^{j}\theta$ (see the clarification in [11]). Three specific values of ${}^{r,j}\gamma$, standing for division points for the second input's regions (Fig. 5.10b), were distinguished, as shown in the set of equations (5.24). The following cases of the ${}^{r,j}\gamma$ value were specified to develop the boundaries of *FIS* regions:

- ${}^{r}\gamma_{\mathrm{eq}}$ indicates that j moves in the same direction as r,
- ${}^{r}\gamma_{\mathrm{opp}}$ indicates that j moves in a direction opposite to r,
- ${}^{r,j}\gamma_{\mathrm{cc}}$ indicates that a ray created from a centre point and a heading of j crosses the centre point of r.

[11]. Although the ${}^{r,j}\gamma$ is calculated as ${}^{r,j}\gamma = {}^{j}\theta$, i.e., it only depends on the state of the object j, the symbol of ${}^{r,j}\gamma$ is indicated as describing a value connected to the entities of r and j. This mathematical procedure was performed to reflect the fact that a pose of a dynamic object j directly influences the value of ${}^{r,j}\gamma$, which is calculated for the robot r.

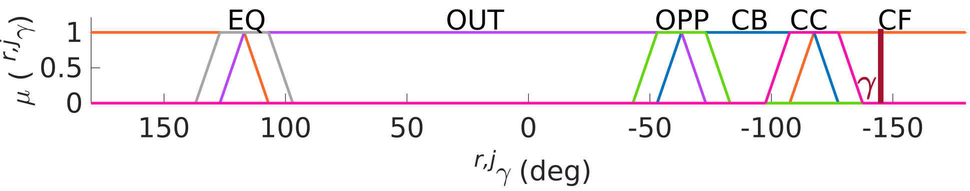


Figure 5.12: A membership function of the second *FIS* input variable with the ${}^{r,j}\gamma$ angle value marked with the vertical bar. The function is presented for the specific spatial arrangement of r and j shown in Fig. 5.10, where j's direction ray crosses r's direction ray in front of r ("CF" case).

$$\begin{aligned} {}^{r}\gamma_{\mathrm{eq}} &= {}^{r}\theta \\ {}^{r}\gamma_{\mathrm{opp}} &= {}^{r}\gamma_{\mathrm{eq}} + 180° \\ {}^{r,j}\gamma_{\mathrm{cc}} &= {}^{r,j}\phi + 180° \end{aligned} \tag{5.24}$$

Regions of the second input evolve between the values defined in (5.24). The arrangement of regions, ${}^{r,j}\Gamma$, can be formulated in the normalised angle domain regarding the relative location (indicating the right or left side), as in (5.25). The fuzzification regions are generated with a 10° extension (experimentally determined) beyond the region boundary values (Fig. 5.12). All configurations of input variables considered in the rule bases are presented in Fig. 5.13.

$${}^{r,j}\Gamma = \begin{cases} \begin{cases} {}^{r}\gamma_{\mathrm{opp}} < {}^{r,j}\gamma_{\mathrm{out}} < {}^{r}\gamma_{\mathrm{eq}} \\ {}^{r,j}\gamma_{\mathrm{cc}} < {}^{r,j}\gamma_{\mathrm{cb}} < {}^{r}\gamma_{\mathrm{opp}} \\ {}^{r}\gamma_{\mathrm{eq}} < {}^{r,j}\gamma_{\mathrm{cf}} < {}^{r,j}\gamma_{\mathrm{cc}} \end{cases} & \text{if } {}^{r,j}\delta < 0 \\ \begin{cases} {}^{r}\gamma_{\mathrm{opp}} > {}^{r,j}\gamma_{\mathrm{out}} > {}^{r}\gamma_{\mathrm{eq}} \\ {}^{r,j}\gamma_{\mathrm{cc}} > {}^{r,j}\gamma_{\mathrm{cb}} > {}^{r}\gamma_{\mathrm{opp}} \\ {}^{r}\gamma_{\mathrm{eq}} > {}^{r,j}\gamma_{\mathrm{cf}} > {}^{r,j}\gamma_{\mathrm{cc}} \end{cases} & \text{if } {}^{r,j}\delta \geqslant 0 \end{cases} \tag{5.25}$$

FIS rules The rationales for the rule design in the proposed *FIS* are social conflict avoidance behaviours [28] and customary rules of pedestrian motion. For instance, in most countries, pedestrians try to pass others on the right [137, 71, 126, 84, 348, 72], overtake on the left [71, 141], and give way to a human on the right when directions of both pedestrians nearly cross [86, 77, 416]. Additionally, enhancing robot motions that increase intent expressiveness improves its motion legibility perceived by humans [23]. Furthermore, the *FIS* rules emerge to recreate the two-lane formation social phenomena [138, 44] and enable the robot to slow down when a collision is predicted and stop when a collision is imminent.

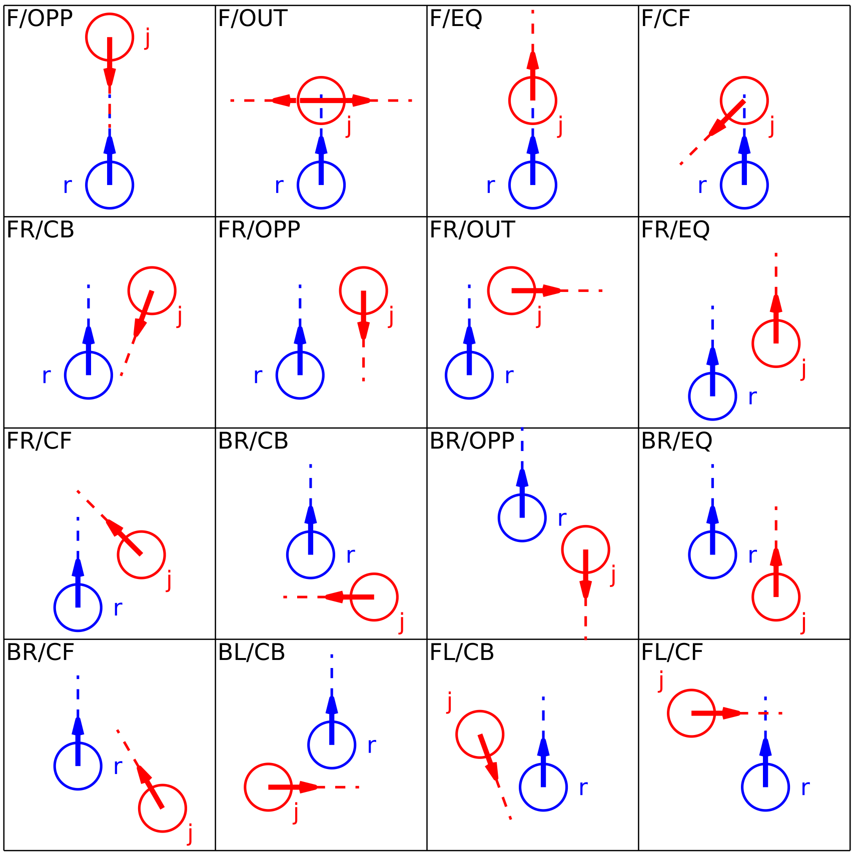


Figure 5.13: Visualisation of the reciprocal r and j pose configurations. The specified cases are accommodated in the fuzzy inference system to implement social behaviours in a robot navigation system.

In *FESFM*, reproducing social behaviours is accomplished with short-term actions, namely: accelerate (*ACC*), turn right accelerating (*TRA*), turn right (*TR*), turn right decelerating (*TRD*), decelerate (*DEC*), stop (*S*), turn left decelerating (*TLD*), turn left (*TL*), turn left accelerating (*TLA*). The actions are induced from the robot's r and dynamic object's j reciprocal location (reflected by ${}^{r,j}\delta$) as well as their motion directions (reflected by ${}^{r,j}\gamma$). Based on these features, the spatiotemporal arrangement of the robot and the dynamic object is classified into one of the specific cases described in the reasoning block in Tab. 5.6, which contains the set of fuzzy rules of the presented *FIS*.

FIS output variable The defuzzified output variable, ${}^{r,j}\nu^{n}$, represents the membership function's argument (Fig. 5.14) identified as the angle defining the direction of the action that implements the human-like decision-making strategy. However, movement actions in the *SFM*-based pedestrian motion model must be represented by the force; therefore, the magnitude of the new force component, ${}^{r,j}\boldsymbol{f}_{\text{beh}}$, must be separately calculated.

To find the magnitude of the force implementing social behaviour for the situation at hand, a novel heuristic has been developed based on the findings from the literature.

$^{r,j}\delta$ \ $^{r,j}\gamma$	***CC***	***CB***	***OPP***	***OUT***	***EQ***	***CF***
F	–	–	*TR*	*DEC*	*DEC*	*TR*
FR	–	*TL*	*TL*	*TL*	*TLA*	*TR*
BR	–	*TLA*	*TR*	–	*TRA*	*ACC*
B	–	–	–	–	–	–
BL	–	*ACC*	–	–	–	–
FL	–	*TR*	–	–	–	*TRA*

Table 5.6: Fuzzy rule bases. Values of the first input are denoted under $^{r,j}\delta$, whereas the values of the second input correspond to $^{r,j}\gamma$.

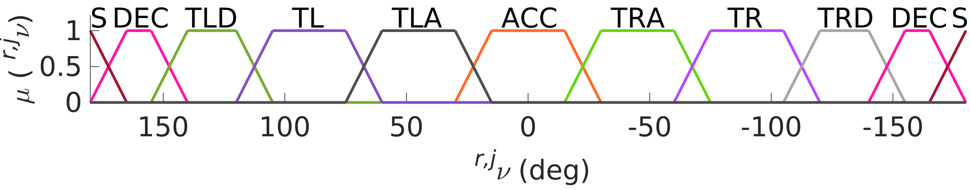


Figure 5.14: A membership function for the *FIS* output variable.

Specifically, the scale of the force depends exponentially on the distance $^{r,j}d$ between the robot r and the object j [1, 314, 275], but is also affected by their relative speed, $^{r,j}v$ [408, 25]. Additionally, the force strength is influenced by the value of the output's membership to the best matching *FIS* case, $^{r,j}\mu_{\nu}$, which directly assesses the certainty of classification of the inferred situation (arrangement). Notably, the action angle, $^{r,j}\nu$, is defined in the robot's r local coordinate system; thus, the direction of the force needs to be transformed from the robot's r local coordinate system to the global one. Finally, the configurable amplitude factor, A_s, is introduced for, e.g., levelling the scale with other components. The overall formulation of the *FIS*-based component of the *FESFM* pedestrian motion model is shown in (5.26).

$$^{r,j}\boldsymbol{f}_{\mathrm{beh}} = A_s \cdot e^{-\,^{r,j}d} \cdot \left(e^{\,^{r,j}v} - 1\right) \cdot \,^{r,j}\mu_{\nu} \cdot \begin{bmatrix} \cos\left(^{r}\theta + \,^{r,j}\nu\right) \\ \sin\left(^{r}\theta + \,^{r,j}\nu\right) \end{bmatrix} \tag{5.26}$$

The *FIS* must handle the presence of multiple dynamic objects, identified as $\mathbb{J}^n$, around a robot. The vector addition principle has been used as an aggregation method when the cardinality of the set of dynamic objects at time t^n conforms to $|\mathbb{J}^n| > 1$.

Nevertheless, when none of the implemented actions is activated, the robot's candidate trajectories produced by the $^{\mathbb{O}}\mathrm{gen}_{\mathrm{soc}}$ generator are not influenced by the $^{r,j}\boldsymbol{f}_{\mathrm{beh}}$ component.

5.5.6 Conforming pedestrian motion model to velocity control of mobile bases

Mobile robots are commonly velocity-controlled. Since the *SFM*-based motion model that was employed in the social trajectory generator, $\mathrm{gen}_{\mathrm{soc}}$, provides control commands expressed in the force domain, they need to be transformed to the velocity domain. Furthermore, the virtual forces in the *SFM* are generated without taking the mobility constraints of agents into account.

Holonomic robot platforms characterise performing lateral motions contrary to the heading direction, which is considered unnatural and not goal-directed for people [215]. Consequently, most social robot mobile bases are equipped with simple differential drives. Nonholonomic constraints of the platform selected for tests (Sec. 7.1) make the robot not compatible with the raw *SFM* driving vector. This problem was already addressed in [165], where the authors proposed a function transforming a force, ${}^{r}\boldsymbol{f}$, to a velocity vector ${}^{r}\boldsymbol{v}$ for a nonholonomic robot r. The method is referred to as forceToVelocity in Alg. 4.

5.5.7 Generating numerous pedestrian model-based trajectory candidates

The *HUMAP* local trajectory planning algorithm employs the *FESFM* pedestrian motion model (Sec. 5.5.5), based on the *SFM* (Sec. 5.5.4), for producing trajectory candidates. The planning approach implemented in the *HUMAP* relies on searching for the lowest cost trajectory amid the candidate trajectories. Increasing the number of candidates produced enhances the likelihood of finding a solution closer to the optimal. In this section, the approach of generating multiple trajectory candidates from a deterministic pedestrian motion model is demonstrated.

Limitations of static parameter values in SFM-based models As stated in Sec. 5.5.3, the authors of multiple works have already attempted to assess the parameter values of pedestrian motion models applied for social robot navigation. The parameters of the baseline *SFM* formulation were estimated in [25], where the authors found that after the calibration driven by the real-world data, each parameter still displays a significant standard deviation from the mean value. This can be explained by the non-deterministic nature of real-world processes that involve humans. However, researchers acknowledge that a static set of *SFM* parameters might produce satisfactory navigation results for a specific scenario, as in [314, 321], and still be valid in scenarios meeting similar conditions [165], but lacks generalisation to different environments. Even parameter calibration via online learning is the same in that matter [165]. On the other hand, dynamically changing parameters may produce valid trajectories for different scenarios [315], but defining a versatile relationship between *SFM* parameter values and the environment state is challenging. Taking into account the aspects of the deviation of calibrated parameter values and the necessity to dynamically select them across various scenarios, the method of generating multiple trajectory candidates has been developed in the *HUMAP*.

Diversifying trajectory candidates The *HUMAP*'s model-based generator of socially acceptable trajectory candidates, $\mathrm{gen}_{\mathrm{soc}}$, exploits that each parameter of the baseline pedestrian motion model should undergo validation with values spanning throughout the range defined by their standard deviations. Hence, our method introduced the multipliers of pedestrian motion model parameters. Manipulating the value of each parameter has a direct impact on robot dynamics and emphasises distinctive behaviours in the robot's motion, e.g., a bigger keepout distance from static obstacles or earlier evasive manoeuvres in front of a dynamic obstacle. Furthermore, applying numerous coefficients to each parameter ensures that the deterministic *FESFM* model can generate diverse pedestrian-like trajectory candidates. Those candidates are later scored with socially-aware cost functions to select the most relevant trajectory for a human-aware robot at the current state, as detailed in Sec. 5.7.

The span of values of model parameters' multipliers is bounded by the minimum, ${}^{r}_{\min}\kappa_{(\cdot)}$, and the maximum, ${}^{r}_{\max}\kappa_{(\cdot)}$, values, which can either be derived from the standard deviation of the baseline model parameters (e.g., [25]) or determined experimentally, by evaluating the range that significantly impacts the search space. Then, the multiplier's discretisation step (granularity), ${}^{r}_{\mathrm{stp}}\kappa_{(\cdot)}$, must be established by striking a balance between computational complexity and search resolution.[12] The composition of multipliers for a given parameter i, denoted by ${}^{r}\boldsymbol{\kappa}_i$, is described in (5.27).

$$\begin{aligned}
{}^{r}\boldsymbol{\kappa}_i &= \left\{ {}^{r}_{j}\kappa_i \mid j \in \mathbb{N}_0,\ 0 \leqslant j \leqslant \left\lceil \frac{{}^{r}_{\max}\kappa_i - {}^{r}_{\min}\kappa_i}{{}^{r}_{\mathrm{stp}}\kappa_i} \right\rceil \right\} \\
{}^{r}_{j}\kappa_i &= \begin{cases} \kappa = {}^{r}_{\min}\kappa_i + j \cdot {}^{r}_{\mathrm{stp}}\kappa_i & \text{if } \kappa \leqslant {}^{r}_{\max}\kappa_i \\ {}^{r}_{\max}\kappa_i & \text{otherwise} \end{cases}
\end{aligned} \tag{5.27}$$

Then, the multipliers are applied to each baseline parameter from the $\boldsymbol{\rho}_{\mathrm{bsl}}$ set, augmented by the *FIS*-related A_s parameter of the ${}^{r,j}\breve{\boldsymbol{f}}^{n}_{\mathrm{beh}}$, to obtain the set of resultant parameter values, ${}^{r}\boldsymbol{\rho}_i$ (for a specific parameter i), that are used for generating diversified trajector-

[12] The smaller the discretisation step is, the closer the solution will be to a model-constrained-optimal one; however, for real-world, time-constrained systems, it is desirable to keep computation times reasonable and to recalculate with a higher frequency at the cost of acquiring a suboptimal solution.

ies (5.28).

$$ {}^{r}\boldsymbol{\rho}_i = {}^{r}\boldsymbol{\kappa}_i \cdot i, \text{ where } i \in \boldsymbol{\rho}_{\text{bsl}} \cup A_s \tag{5.28} $$

Our trajectory generation involves searching for all possible parameter combinations, ${}^{r}\boldsymbol{\rho}_{\text{all}}$; hence, the parameter tuples are the results of the Cartesian product of uniformly spaced parameters (5.29).[13] The i-th tuple with parameters, ${}^{r}_{i}\boldsymbol{\rho}_{\text{all}}$, is shown in (5.30).[14]

$$ \begin{aligned} {}^{r}\boldsymbol{\rho}_{\text{all}} = {}^{r}\boldsymbol{\rho}_{v_{\text{des}}} \times {}^{r}\boldsymbol{\rho}_{A_n} \times {}^{r}\boldsymbol{\rho}_{B_n} \times {}^{r}\boldsymbol{\rho}_{C_n} \times {}^{r}\boldsymbol{\rho}_{A_p} \times \\ \times {}^{r}\boldsymbol{\rho}_{B_p} \times {}^{r}\boldsymbol{\rho}_{C_p} \times {}^{r}\boldsymbol{\rho}_{A_w} \times {}^{r}\boldsymbol{\rho}_{B_w} \times {}^{r}\boldsymbol{\rho}_{A_s} \end{aligned} \tag{5.29} $$

$$ \begin{aligned} {}^{r}_{i}\boldsymbol{\rho}_{\text{all}} = \Big({}^{r}_{(\cdot)}\rho_{v_{\text{des}}}, {}^{r}_{(\cdot)}\rho_{A_n}, {}^{r}_{(\cdot)}\rho_{B_n}, {}^{r}_{(\cdot)}\rho_{C_n}, {}^{r}_{(\cdot)}\rho_{A_p}, \\ {}^{r}_{(\cdot)}\rho_{B_p}, {}^{r}_{(\cdot)}\rho_{C_p}, {}^{r}_{(\cdot)}\rho_{A_w}, {}^{r}_{(\cdot)}\rho_{B_w}, {}^{r}_{(\cdot)}\rho_{A_s} \Big) \end{aligned} \tag{5.30} $$

Since we applied multipliers to 10 parameters of our *FESFM* pedestrian motion model, and each combination of multiplier values is assessed, the search space of a model-constrained-optimal trajectory[15] is at most 10-dimensional. However, dimensionality reduces by 1 with each multiplier set's cardinality conforming $\left|{}^{r}\boldsymbol{\kappa}_{(\cdot)}\right| = 1$.

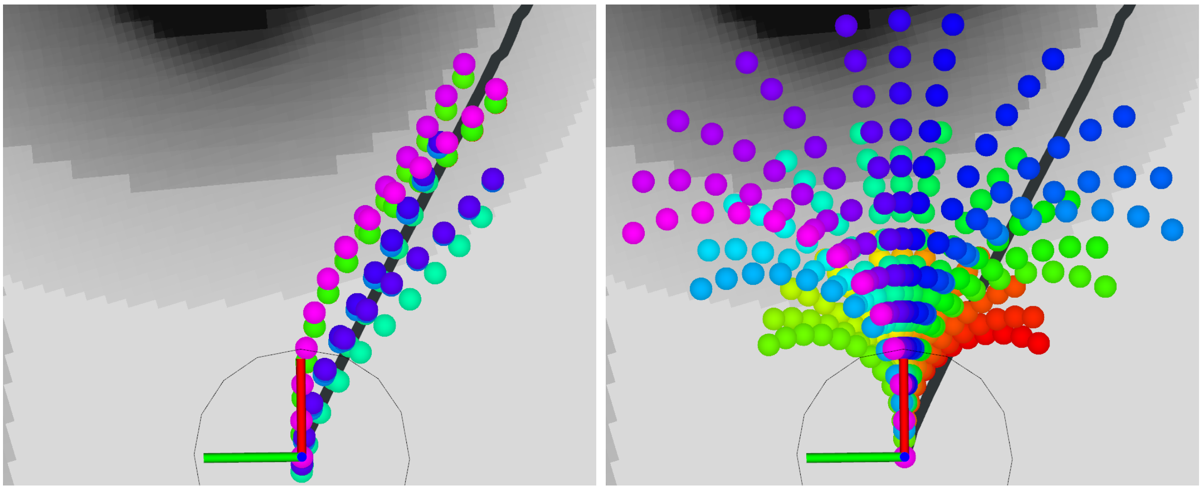
Trajectory generation Using the deterministic pedestrian motion model to generate numerous trajectory candidates exploits the model's parameterisation. The candidates are produced by replacing the baseline parameters, $\boldsymbol{\rho}_{\text{bsl}}$, with the subsequent parameter tuples from ${}^{r}\boldsymbol{\rho}_{\text{all}}$, i.e., ${}^{r}_{1}\boldsymbol{\rho}_{\text{all}}$, ${}^{r}_{2}\boldsymbol{\rho}_{\text{all}}$, ..., ${}^{r}_{\chi}\boldsymbol{\rho}_{\text{all}}$. Specifically, with the $(\cdot)$-th tuple, the A_n parameter appearing in the *FESFM* formulation is swapped with the ${}^{r}_{(\cdot)}\rho_{A_n}$, then B_n with ${}^{r}_{(\cdot)}\rho_{B_n}$, and so forth.

A valid trajectory generation in time t^n reveals a tuple of parameters, denoted as ${}^{r}_{(\cdot)}\boldsymbol{\rho}_{\text{all}}$, which produces a trajectory with the lowest cost among the set of model-based trajectories generated by applying various parameter tuples to our pedestrian motion

[13] An exception to the "uniform spacing" may hold for at most 1 (the biggest) value of each parameter. It might occur if the second case of the second part of the (5.27) equation applies for a given parameter, i.e., there does not exist an integer number, that multiplied by the discretisation step ${}^{r}_{\text{stp}}\kappa_{(\cdot)}$, equals the range given by ${}^{r}_{\max}\kappa_{(\cdot)} - {}^{r}_{\min}\kappa_{(\cdot)}$.

[14] Generally, with the variable number of multipliers for each parameter, it is not viable to identify the index of each parameter in the i-th tuple, ${}^{r}_{i}\boldsymbol{\rho}_{\text{all}}$, therefore, a generic placeholder, $(\cdot)$, has been used. Each $(\cdot)$ may indicate a different index in (5.30).

[15] The phrase "model-constrained-optimal trajectory" means that: 1) there is a parameterised "model" from which trajectories are generated, and 2) numerous trajectories can be generated, but none of them might be globally optimal due to a discretised search space. However, amid the trajectories produced by the model (thus "model-constrained"), there is 1 trajectory with the lowest cost (thus "optimal" in terms of the solutions generated by the model).



(a) Model-based generator (b) Sampling-based generator

Figure 5.15: Candidate trajectories created using the pedestrian motion model $\mathrm{gen}_{\mathrm{soc}}$ (a) and the sampling-based generator $\mathrm{gen}_{\mathrm{smp}}$ (b). The global path is indicated by the black line. The robot's footprint is marked with a grey circle and its orientation is depicted with a coordinate system marker. Path points resulting from an individual trajectory are of the same colour.

model. Moreover, a trajectory is considered a valid candidate only when it conforms to the kinodynamic constraints of the mobile base throughout the planning horizon, i.e., the entire trajectory must constitute only feasible velocities (hence isFeasible expression in Alg. 4). Trajectories failing to meet this requirement are rejected from further investigation. The complete procedure for human-aware trajectories generation using the *FESFM* pedestrian motion model is shown in Alg. 4. The implementation used during experimental studies produces $\chi = 72$ trajectories using the model-based trajectory generator, $\mathrm{gen}_{\mathrm{soc}}$. Nevertheless, investigating more trajectories is justified, if the real-time performance of the trajectory planning is not an issue. The visual representation of the produced trajectories is illustrated in Fig. 5.15a.

5.6 Velocity sampling-based trajectory generation

We argue that a sole social trajectory generator utilising the *SFM*-based motion model is not sufficient for robust robot navigation in dynamic, populated or cluttered environments due to being vulnerable to the local minima or oscillations [1, 165]. Therefore, we also employed a velocity sampling trajectory generator, $\mathrm{gen}_{\mathrm{smp}}$, that creates trajectories from feasible motion primitives, i.e., velocity tuples ($[v_x, \omega]$ pairs for nonholonomic or $\left[v_x, v_y, \omega\right]$ triplets for holonomic drives) regarding kinodynamic constraints [161], but without taking environment model into account (collision checking is performed by scor-

Algorithm 4 Creating trajectory candidates, ${}^{r}\mathbf{traj}^{n}_{\mathrm{soc}}$, for the robot r at time t^n using the generator based on the pedestrian motion model, ${}^{r,\mathbb{O}}\mathrm{gen}^{n}_{\mathrm{soc}}$

function GENERATETRAJECTORIES$({}^{\mathbb{O}}\mathrm{gen}_{\mathrm{soc}})$
  $\mathbf{traj}_{\mathrm{soc}} \leftarrow \emptyset$ ▷ List containing generated trajectories
  for each ${}_{i}\boldsymbol{\rho}_{\mathrm{all}} \in \boldsymbol{\rho}_{\mathrm{all}}$ **do** ▷ Iterate over param. tuples
    $t^{\mathrm{sim}} \leftarrow t^{n}$ ▷ Save the initial time stamp
    $\mathbb{O}^{\mathrm{sim}} \leftarrow \mathbb{O}^{n}$ ▷ Save the initial environment state
    traj $\leftarrow \emptyset$ ▷ Uninitialised trajectory candidate
    invalid $\leftarrow$ False ▷ For detecting traj. gen. failure
    ▷ Iterate over time stamps along the plan. horizon
    while $t^{\mathrm{sim}} \leqslant (t^{n} + t_{\mathrm{hor}})$ **do**
      ▷ Compute force according to the motion model
      ${}^{\mathbb{O}}\breve{\boldsymbol{f}}^{\mathrm{sim}}_{FESFM} \leftarrow \mathrm{FESFM}\left({}_{i}\boldsymbol{\rho}_{\mathrm{all}}, \mathbb{O}^{\mathrm{sim}}\right)$
      $\boldsymbol{v}^{\mathrm{sim}} \leftarrow$ FORCETOVELOCITY$\left({}^{\mathbb{O}}\breve{\boldsymbol{f}}^{\mathrm{sim}}_{FESFM}\right)$
      ▷ Evaluate the feasibility of the velocity
      if not ISFEASIBLE$\left(\boldsymbol{v}^{\mathrm{sim}}\right)$ **then**
        invalid $\leftarrow$ True
        break ▷ Traj. violates kinodyn. constraints
      end if
      ▷ Initialise traj. with the first planned velocity
      if traj $== \emptyset$ **then**
        traj $\leftarrow$ INITTRAJ$(\boldsymbol{p}^{n}, \boldsymbol{v}^{n})$
      end if
      ▷ Extend the robot trajectory applying $\boldsymbol{v}^{\mathrm{sim}}$
      traj $\leftarrow$ PREDICT$\left(\mathrm{traj}, \boldsymbol{v}^{\mathrm{sim}}\right)$
      ▷ Predict the state of the environment
      $\mathbb{O}^{\mathrm{sim}+1} \leftarrow$ PREDICT$\left(\mathbb{O}^{\mathrm{sim}}\right)$
      $t^{\mathrm{sim}} \leftarrow t^{\mathrm{sim}} + t_{\Delta}$ ▷ For the next traj.
    end while
    if invalid $==$ True **then**
      continue ▷ Curr. par. tuple produces infeas. traj.
    end if
    APPEND$(\mathbf{traj}_{\mathrm{soc}}, \mathrm{traj})$ ▷ Extend the traj. list
  end for
  return $\mathbf{traj}_{\mathrm{soc}}$ ▷ Return the list of generated traj.
end function

ing functions). We finally selected ${}^{r}\mathrm{smp}_x = 3$ linear (${}^{r}\mathrm{smp}_y = 1$, equal to 0, as our test platform is nonholonomic) and ${}^{r}\mathrm{smp}_\omega = 11$ angular velocities that produce additional $\zeta = 33$ trajectories to be scored in each time step. The algorithm for producing candidate trajectories by the velocity sampling generator is illustrated in Alg. 5.

Algorithm 5 Creating trajectory candidates, ${}^{r}\mathbf{traj}^{n}_{\mathrm{smp}}$, for the robot r at time t^n using the trajectory generator based on velocity sampling ${}^{r}\mathrm{gen}^{n}_{\mathrm{smp}}$

---

function GENERATETRAJECTORIES($\mathrm{gen}_{\mathrm{smp}}$)
    ▷ Compute boundaries of feasible vel. at the end of
    ▷ the plan. horizon, e.g., for the lower boundary:
    ▷ ${}_{\min}\bar{\boldsymbol{v}} = \left[{}_{\min}\bar{v}_x, {}_{\min}\bar{v}_y, {}_{\min}\bar{\omega}\right]$
    ${}_{\min}\bar{\boldsymbol{v}} \leftarrow \boldsymbol{v} - \boldsymbol{a} \cdot t_{\mathrm{hor}}$ ▷ $\boldsymbol{a}$ robot's accel. limits
    ${}_{\max}\bar{\boldsymbol{v}} \leftarrow \boldsymbol{v} + \boldsymbol{a} \cdot t_{\mathrm{hor}}$
    ▷ Create lists of feasible velocities, taking the cardinality
    ▷ of each velocity component into account;
    ▷ LINSPACE returns evenly spaced num. over an interval;
    ▷ if ${}_{\min}\bar{v}_{(\cdot)} < 0$ and ${}_{\max}\bar{v}_{(\cdot)} > 0$, then "0" sample is incl.
    $\tilde{\boldsymbol{v}}_x \leftarrow$ LINSPACE$({}_{\min}\bar{v}_x, {}_{\max}\bar{v}_x, \mathrm{smp}_x)$
    $\tilde{\boldsymbol{v}}_y \leftarrow$ LINSPACE$\left({}_{\min}\bar{v}_y, {}_{\max}\bar{v}_y, \mathrm{smp}_y\right)$
    $\tilde{\boldsymbol{\omega}} \leftarrow$ LINSPACE$({}_{\min}\bar{\omega}, {}_{\max}\bar{\omega}, \mathrm{smp}_\omega)$
    $\tilde{\boldsymbol{v}}_{\mathrm{smp}} \leftarrow \tilde{\boldsymbol{v}}_x \times \tilde{\boldsymbol{v}}_y \times \tilde{\boldsymbol{\omega}}$ ▷ List of feas. vel. triplets
    $\mathbf{traj}_{\mathrm{smp}} \leftarrow \emptyset$ ▷ List containing generated trajectories
    for each $\tilde{\boldsymbol{v}} \in \tilde{\boldsymbol{v}}_{\mathrm{smp}}$ **do** ▷ Iter. over feas. vel. triplets
        $t^{\mathrm{sim}} \leftarrow t^n$ ▷ Save the initial time stamp
        traj $\leftarrow$ INITTRAJ$(\boldsymbol{p}^n, \boldsymbol{v}^n)$ ▷ Initialise traj.
        ▷ Iterate over time stamps along the plan. horizon
        while $t^{\mathrm{sim}} \leqslant (t^n + t_{\mathrm{hor}})$ **do**
            traj $\leftarrow$ PREDICT(traj, $\tilde{\boldsymbol{v}}$) ▷ Const. vel.
            $t^{\mathrm{sim}} \leftarrow t^{\mathrm{sim}} + t_\Delta$ ▷ For the next traj.
        end while
        APPEND$\left(\mathbf{traj}_{\mathrm{smp}}, \mathrm{traj}\right)$ ▷ Extend the list
    end for
    return $\mathbf{traj}_{\mathrm{smp}}$ ▷ Return the list of generated traj.
end function

Examples of trajectory candidates obtained for the same environment state with the model-based social trajectory generator, $\mathrm{gen}_{\mathrm{soc}}$, and the velocity sampling-based generator, $\mathrm{gen}_{\mathrm{smp}}$, are shown in Fig. 5.15a and Fig. 5.15b, accordingly. The latter produces curved trajectories that cover a discretised space of feasible velocities (nonholonomic robot example). On the other hand, the model-based generator creates concentrated trajectories (95 candidates in Fig. 5.15a) that avoid collisions and follow the local goal located along the global path. Using both generators enables covering most of the viable configuration space with diverse candidates.

5.7 Trajectory scoring

Creating multiple trajectories requires scoring the candidates produced by each generator, $\mathrm{gen}_{\mathrm{soc}}$ and $\mathrm{gen}_{\mathrm{smp}}$, to select the one with the lowest cost (5.6). Trajectory evaluation has an essential impact on which candidate will be selected; therefore, cost functions for human-aware navigation should map the social robot navigation requirements (Chapter 2).

A distinctive characteristic of the *HUMAP* planner is that the cost functions for online trajectory scoring are adapted from our metrics that were originally proposed for the offline benchmarking (regarding the whole experiment) of social robot navigation [21], described in (Chapter 4). The metrics quantitatively evaluate both robot navigation performance and its social acceptance during navigation, taking into account the uncertainty of robot perception in terms of human tracking.

In *HUMAP*, cost functions for scoring candidate trajectories can be classified into two main groups: evaluating robot navigation performance and assessing human discomfort. Furthermore, human discomfort cost functions can be further divided into those quantifying robot motion naturalness, and those evaluating the physical and perceived safety among humans (following the taxonomy identified in Sec. 2.1). Since the objective function for candidate scoring implements opposite criteria, i.e., includes both performance-focused cost functions as well as cost functions mitigating human discomfort, a Pareto-optimal solution is being searched for.

Our approach to local trajectory planning uses spatial, spatiotemporal and temporal cost functions, all stored in the ${}^{r,E}\mathbf{cfun}_{\mathrm{all}}$ vector, to evaluate candidate trajectories. Spatial cost functions are commonly embedded into a discretised costmap representation of the robot environment [170] to penalise the robot for traversing through certain positions. Spatiotemporal cost functions, on the other hand, also penalise the robot for moving through certain areas but evaluate the actions that happen in time and affect the pose of the robot, usually in a significant horizon. They might require environment state predictions as well, e.g., forecasting human trajectories. In contrast, temporal cost functions penalise the robot's dynamics within a given trajectory, without considering spatial aspects.

In our method, only trajectories conforming to the kinodynamic constraints are treated as valid and those are evaluated by cost functions. Assessing the i-th trajectory is equivalent to calculating its total cost, ${}^{r,E}_{i}\mathrm{cost}^{n}_{\mathrm{all}}$. The cost of a trajectory is computed using the scalarised multi-objective cost function, presented in (5.31), with the weighted sum method, as illustrated in Alg. 3. The $(\cdot)$-th cost function, which only relies on the state of the robot r and does not take any environment objects into account, is denoted by ${}^{r}\mathrm{cfun}^{n}_{(\cdot)}$, whereas its weight as ${}^{r}\rho_{(\cdot)}$. Individual cost functions are described later in this section.

$$\begin{aligned}
{}^{r,E}_{i}\mathrm{cost}^{n}_{\mathrm{all}} = {} & {}^{r}\rho_{\mathrm{trav}} \cdot {}^{r,E}\mathrm{cfun}^{n}_{\mathrm{trav}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + {}^{r}\rho_{\mathrm{pth}} \cdot {}^{r,E}\mathrm{cfun}^{n}_{\mathrm{pth}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + \\
& + {}^{r}\rho_{\mathrm{goal}} \cdot {}^{r,E}\mathrm{cfun}^{n}_{\mathrm{goal}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + {}^{r}\rho_{\mathrm{glfr}} \cdot {}^{r,E}\mathrm{cfun}^{n}_{\mathrm{glfr}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + \\
& + {}^{r}\rho_{\mathrm{bwd}} \cdot {}^{r}\mathrm{cfun}^{n}_{\mathrm{bwd}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + {}^{r}\rho_{\mathrm{osc}} \cdot {}^{r}\mathrm{cfun}^{n}_{\mathrm{osc}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + \\
& + {}^{r}\rho_{\mathrm{ttc}} \cdot {}^{r,E}\mathrm{cfun}^{n}_{\mathrm{ttc}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + {}^{r}\rho_{\mathrm{hsm}} \cdot {}^{r}\mathrm{cfun}^{n}_{\mathrm{hsm}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + \\
& + {}^{r}\rho_{\mathrm{vsm}} \cdot {}^{r}\mathrm{cfun}^{n}_{\mathrm{vsm}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + {}^{r}\rho_{\mathrm{dir}} \cdot {}^{r,\mathbb{H}}\mathrm{cfun}^{n}_{\mathrm{dir}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + \\
& + {}^{r}\rho_{\mathrm{psi}} \cdot {}^{r,\mathbb{H}}\mathrm{cfun}^{n}_{\mathrm{psi}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + {}^{r}\rho_{\mathrm{fsi}} \cdot {}^{r,\mathbb{G}}\mathrm{cfun}^{n}_{\mathrm{fsi}}\left({}^{r}\mathrm{traj}^{n}_{i}\right) + \\
& + {}^{r}\rho_{\mathrm{psd}} \cdot {}^{r,\mathbb{H}}\mathrm{cfun}^{n}_{\mathrm{psd}}\left({}^{r}\mathrm{traj}^{n}_{i}\right)
\end{aligned} \tag{5.31}$$

Changing the weights of cost functions influences, which trajectory candidate will be selected as the best in a given time step, for a given planning horizon. Manipulating the weights enables the system designer to select, e.g., whether the robot is intended to sacrifice time performance in favour of social compliance in a populated environment. The trajectory selection procedure treats socially acceptable robot motions as soft constraints, meaning they are permissible unless they result in a collision, which is interpreted as a hard constraint. Therefore, the robot could still apply a trajectory that deviates from the globally planned and shortest path, when the cost functions assess the trajectory's cost as the lowest (minimising the objective amid the candidates).

As the *HUMAP* is mainly developed for robots performing unfocused interactions with humans, in our test setup, the weights were tuned towards the human-aware motion behaviour. This means that the weights of cost functions penalising human discomfort and unnatural robot motions were increased, but only at a minimal degradation in overall robot navigation performance (Pareto optimality).

The remaining part of this section describes all cost functions that are regarded in the objective function, i.e., they are also embedded into the ${}^{r,E}\mathbf{cfun}^{n}_{\mathrm{all}}$ vector.

5.7.1 Assessing robot navigation performance

The aspects of robot navigation performance in local trajectory planning mainly regard the global path following, while avoiding collisions. The operational scheme of cost functions described in this section is illustrated in Alg. 3: the investigated trajectory is passed as an argument to each cost function, resulting in the cost of that trajectory being returned from the function. All cost functions discussed below are aggregated in the objective introduced in (5.31).

Traversal costs A fundamental cost function, ${}^{r,E}\mathrm{cfun}_{\mathrm{trav}}$, discards trajectories moving the robot into obstacles and penalises traversing through areas in obstacles' proximity [147]. This spatial cost function uses a dense representation of the environment to quantitatively evaluate the robot's footprint traversal through the predicted trajectory poses (associated with the local costmap's cells), providing the investigated trajectory was applied (Alg. 6).

The local costmap, representing the robot's surroundings, is created in real-time based on the recent perception data and determines the cost of traversal through certain environment positions, e.g., through locations occupied by obstacles (Fig. 5.16a). Additionally, the areas close to obstacles have exponentially increased costs assigned (in a procedure usually called "obstacle inflation") [147]. However, with the layered costmap architecture [170], various high-cost areas can also be embedded. The contextualised local environment representation might contain different special areas that are not treated as empty spaces but also not as impassable locations. The authors of existing approaches prepare costmaps to capture the information specific to social navigation, e.g., left sides of corridors [170] (for right-sided motion pattern) or human activity spaces [2] to discourage planners to traverse such areas. The resultant costmap used for trajectory scoring is flattened, so the number of layers considered only affects the system performance and does not influence the planning procedure.

Other performance-focused costs Our approach also adapts other commonly integrated spatial cost functions for evaluating robot navigation performance. For example, the ${}^{r,E}\mathrm{cfun}_{\mathrm{pth}}$ cost function favours trajectories that overlap with the global path (Fig. 5.16b), which is received by the local trajectory planner from the global path planner. Additionally, the ${}^{r,E}\mathrm{cfun}_{\mathrm{goal}}$ cost function prioritises trajectories that drive the robot towards the local goal (Fig. 5.16c), as proposed in [144, 147]. Furthermore, the ${}^{r,E}\mathrm{cfun}_{\mathrm{glfr}}$ cost function attracts the robot towards a virtual goal placed in front of the robot to prevent deadlocks, i.e., being stuck at local minima (Fig. 5.16d).

The implementation of all these cost functions is inherited from the original *ROS* navigation system.[16] Compared to the original work [147], weights of the cost functions were decoupled to provide a more versatile platform for configuration.

[16] `https://github.com/ros-planning/navigation`

Algorithm 6 Computing the value of the ${}^{r,E}\mathrm{cfun}^{n}_{\mathrm{trav}}$ cost function for a trajectory ${}^{r}\mathrm{traj}^{n}$ generated for the robot r at time t^{n}

```
function ^E cfun_trav(traj)
    ^E cost_trav ← −∞                                   ▷ Stores the highest cost along the traj.
    ▷ Env. dense representation containing obstacles 𝕆
    cm ← GETCOSTMAP()
    for each p^sim ∈ traj do                            ▷ Rewind predicted states
        ▷ Costmap from time t is used in each step
        cost ← GETCOST(cm, p^sim, ftprint)
        ▷ Highest cost of the pred. pose of the r's footprint
        ^E cost_trav ← MAX(^E cost_trav, cost)
    end for
    return ^E cost_trav
end function
```

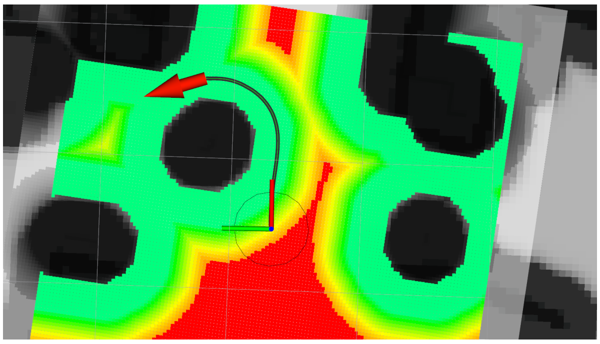
5.7.2 Assessing robot motion naturalness

The robot's motion can be described as natural when it exhibits behaviours that are not perceived as unusual, which typically involves avoiding erratic movements and oscillations. Cost functions evaluating robot motion naturalness in the presence of humans are in most cases temporal and regard robot dynamics or particular movement types. Here, robot velocities are expressed in the mobile base's coordinate system. All cost functions that assess the robot's motion naturalness in the *HUMAP* local trajectory planner were adapted from *SRPB* (Sec. 4.3).

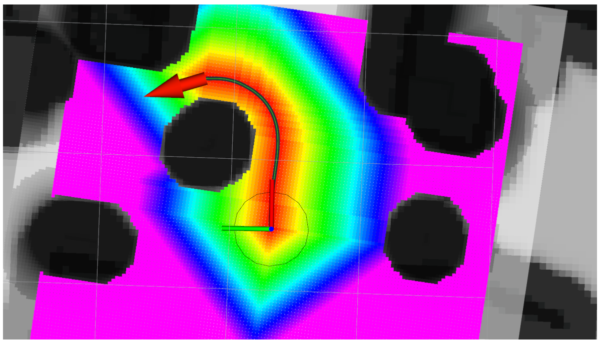
Oscillating motions The ${}^{r}\mathrm{cfun}^{n}_{\mathrm{osc}}$ cost function discards robot trajectories exhibiting oscillating motions, i.e., not developing significant linear and angular velocities. The corresponding cost value, ${}^{r}\mathrm{cost}^{n}_{\mathrm{osc}}$, is computed as in (5.32), where ${}^{r}_{(\cdot)}v_{\mathrm{osc}}$ represent configurable threshold values for relevant robot velocity components.

$$
{}^{r}\mathrm{cost}^{n}_{\mathrm{osc}} = \begin{cases} \mathrm{COST}_{\mathrm{infeas}}, & \text{if } \begin{array}{r} {}^{r}v^{n}_{\mathrm{lin}} < {}^{r}_{\mathrm{lin}}v_{\mathrm{osc}} \\ \wedge \left|{}^{r}v^{n}_{x}\right| < {}^{r}_{x}v_{\mathrm{osc}} \\ \wedge \left|{}^{r}v^{n}_{y}\right| < {}^{r}_{y}v_{\mathrm{osc}} \\ \wedge \left|{}^{r}\omega^{n}\right| < {}^{r}_{\theta}v_{\mathrm{osc}} \end{array} \\ 0, & \text{otherwise} \end{cases} \tag{5.32}
$$

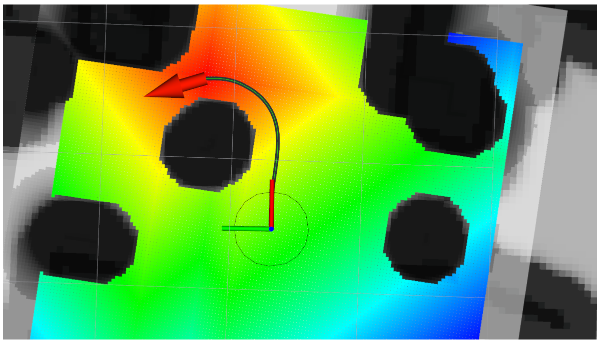
Backward movements The ${}^{r}\mathrm{cfun}^{n}_{\mathrm{bwd}}$ cost function penalises trajectories constituting backward motions. The respective cost value, ${}^{r}\mathrm{cost}^{n}_{\mathrm{bwd}}$, is calculated as in (5.33). The configurable velocity threshold value, ${}^{r}_{x}v_{\mathrm{osc}}$, is shared with the analogous parameter used in ${}^{r}\mathrm{cfun}^{n}_{\mathrm{osc}}$ cost function, so a trajectory is not classified as having oscillating and backward



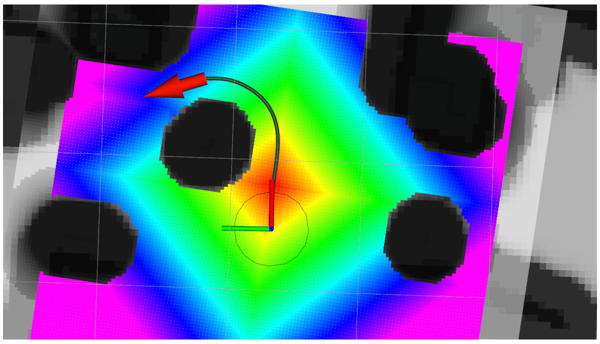
(a) Traversal in obstacles' proximity



(b) Exact path following



(c) Attraction towards the global goal



(d) Attraction towards a local goal

Figure 5.16: A visualisation of cost functions aggregated by the objective function for evaluating the robot navigation performance: (a) obstacle avoidance, (b) global path following, (c) goal-reaching capabilities, and (d) deadlock prevention. In all figures, the cost of traversal through costmap cells is mapped onto the colour scale. Red-coloured areas represent a minimal cost, whereas black colour spaces indicate lethal obstacles. Local costmap bounds are also presented in each figure. The global path is indicated by the thick black line, while a thin black circle represents the robot's footprint.

movements simultaneously.

$$
{}^{r}\mathrm{cost}^{n}_{\mathrm{bwd}} = \begin{cases} 1, & \text{if } {}^{r}v^{n}_{x} \leqslant -{}^{r}_{x}v_{\mathrm{osc}} \\ 0, & \text{otherwise} \end{cases} \tag{5.33}
$$

Velocity smoothness Another cost function, ${}^{r}\mathrm{cfun}^{n}_{\mathrm{vsm}}$, is used for avoiding erratic motions, i.e., trajectories that exhibit a significant change in subsequent linear velocities [12]. The linked cost value, ${}^{r}\mathrm{cost}^{n}_{\mathrm{vsm}}$, is computed as in (5.34), where $\check{n}$ is the index of the time stamp indicating the initiation of the latest planning procedure.

$$
{}^{r}\mathrm{cost}^{n}_{\mathrm{vsm}} = \frac{1}{q_{\mathrm{hor}} - 1} \sum_{n=\check{n}}^{\check{n}+q_{\mathrm{hor}}-1} \frac{\sqrt{\sum_{i\in\{x,y\}} \left({}^{r}v^{n+1}_{i} - {}^{r}v^{n}_{i}\right)^{2}}}{t^{n+1} - t^{n}} \tag{5.34}
$$

Heading change smoothness The ${}^{r}\mathrm{cfun}^{n}_{\mathrm{hsm}}$ cost function penalises robot angular velocity changes [10]. The associated cost value, ${}^{r}\mathrm{cost}^{n}_{\mathrm{hsm}}$, is calculated as in (5.35).

$$ {}^{r}\mathrm{cost}^{n}_{\mathrm{hsm}} = \frac{1}{q_{\mathrm{hor}} - 1} \sum_{n=\check{n}}^{\check{n}+q_{\mathrm{hor}}-1} \frac{\left| {}^{r}\omega^{n+1} - {}^{r}\omega^{n} \right|}{t^{n+1} - t^{n}} \tag{5.35} $$

5.7.3 Assessing humans' physical and perceived safety

The discomfort experienced by humans during robot navigation is often linked with a decreased perceived safety among humans [10]. Neglecting the perceived safety may result in breaches of physical safety protocols. Achieving stress-free and comfortable human-robot interaction is a multifaceted issue, influenced by factors such as adherence to spatial distancing [47, 2], execution of natural movements [10], and avoidance of frightening humans [12]. In the *HUMAP*, several cost functions assessing the perceived safety among humans were adapted from *SRPB* (Sec. 4.4).

Time To Collision costs The ${}^{r,E}\mathrm{cfun}^{n}_{\mathrm{ttc}}$ spatiotemporal cost function penalises trajectories that lead to a collision with dynamic agents within the planning horizon, t_{hor}. It is typically associated with the group of physical safety requirements of social robot navigation (Sec. 2.2). The cost function relies on the *TTC* concept [45, 46, 35], and its calculation method is illustrated in Alg. 7. It exploits the motion prediction of environment objects against the subsequent entries in the investigated trajectory.

Heading straight into a human The concept of motion legibility [23] explores the movement patterns that aim to increase the intent expressiveness. Unfocused human-robot interactions often require passing each other, e.g., in narrow passages. Therefore, the robot should signal its intention early to avoid a collision. In *SRPB*, the m_{dir} metric allows assessing whether the robot moves straight into a human, including the uncertainty of the human pose estimation (Sec. 4.4.3); therefore, implements a practical motion legibility measure. A cost function that utilises the analogous scheme, ${}^{r,\mathbb{H}}\mathrm{cfun}^{n}_{\mathrm{dir}}$, is developed in the *HUMAP*. It employs the normalised value of the m_{dir} metric to evaluate a trajectory (Alg. 8).

Personal spaces intrusion The personal space concept originates from the proxemics theory [47] and stands for a fundamental idea in social robotics. The personal space intrusion metric, m_{psi}, assessing the scale of robot intrusions into any human's personal space [12], was proposed in *SRPB*. The variances modelling a personal space along front, side, and rear directions are calculated based on a human velocity, according to the rules

Algorithm 7 Computing the value of the Time-To-Collision cost function, ${}^{r,E}\mathrm{cfun}^{n}_{\mathrm{ttc}}$, for a trajectory ${}^{r}\mathrm{traj}^{n}$ generated for the robot r at time t^n

function ${}^{E}\mathrm{cfun}_{\mathrm{ttc}}(\mathrm{traj})$
  $E^{\mathrm{sim}} \leftarrow E$ ▷ Save initial env. sparse representation
  for each $\boldsymbol{p}^{\mathrm{sim}} \in \mathrm{traj}$ **do** ▷ Rewind predicted states
    ▷ Compute the shortest dist. vector between the r
    ▷ and any object from the environment at time t^{sim}
    ${}^{E}\boldsymbol{d}^{\mathrm{sim}} \leftarrow \textsc{closestDistance}\left(\boldsymbol{p}^{\mathrm{sim}}, E^{\mathrm{sim}}\right)$
    if $\left\|{}^{E}\boldsymbol{d}^{\mathrm{sim}}\right\| \leqslant d_{\mathrm{ttc}}$ **then** ▷ Dist. below threshold
      $t_{\mathrm{ttc}} \leftarrow t^{\mathrm{sim}} - t^{n}$ ▷ Time proceeded forward
      ▷ The rational function represents an increasing
      ▷ pred. uncertainty for the longer pred. horizon
      return $(t_{\mathrm{hor}}/\, t_{\mathrm{ttc}})$
    end if
    ▷ Prepare prediction of the env. sparse representation
    $E^{\mathrm{sim}+1} \leftarrow \textsc{predict}\left(E^{\mathrm{sim}}\right)$
  end for
  return 0
end function

Algorithm 8 Computing the value of the ${}^{r,\mathbb{H}}\mathrm{cfun}^{n}_{\mathrm{dir}}$ cost function for a trajectory ${}^{r}\mathrm{traj}^{n}$ generated for the robot r at time t^{n}

function ${}^{r,\mathbb{H}}\mathrm{cfun}_{\mathrm{dir}}({}^{r}\mathrm{traj})$
    ▷ Container with discomfort values of each human throughout the planning horizon
    ${}^{\mathbb{H}}\mathbf{dir} \leftarrow \emptyset$
    ▷ Retrieve predicted human trajectories with pose covariances (thus ˜ accent)
    ${}^{\mathbb{H}}\widetilde{\mathbf{traj}} \leftarrow \textsc{getPredTrajs}(\mathbb{H})$
    for each ${}^{h}\widetilde{\mathrm{traj}} \in {}^{\mathbb{H}}\widetilde{\mathbf{traj}}$ **do** ▷ Iterate over predicted human trajectories
        for $\mathrm{sim} \leftarrow n$ to $n + q_{\mathrm{hor}}$ **do** ▷ Iterate over IDs of planning horizon timestamps
            ▷ Retrieve poses (with a covariance) and a velocity at a given step
            ${}^{h}\boldsymbol{p}^{\mathrm{sim}}, {}^{h}\boldsymbol{\Sigma}^{\mathrm{sim}}_{p} \leftarrow \textsc{unpack}({}^{h}\widetilde{\mathrm{traj}}, \mathrm{sim})$
            ${}^{r}\boldsymbol{p}^{\mathrm{sim}}, {}^{r}\boldsymbol{v}^{\mathrm{sim}} \leftarrow \textsc{unpack}({}^{r}\mathrm{traj}, \mathrm{sim})$
            ▷ Calculate the value of the indicator
            ${}^{h,r}\mathrm{dir}^{\mathrm{sim}} \leftarrow \textsc{discomfortDir}({}^{h}\boldsymbol{p}^{\mathrm{sim}}, {}^{h}\boldsymbol{\Sigma}^{\mathrm{sim}}_{p}, {}^{h}d_{\mathrm{ocp}}, {}^{h}\varphi_{\mathrm{fov}}, {}^{r}\boldsymbol{p}^{\mathrm{sim}}, {}^{r}\boldsymbol{v}^{\mathrm{sim}})$
            ▷ Calculate the value of the normalisation factor
            ${}^{h,r}\mathrm{dir}^{\mathrm{sim}}_{\mathrm{nrm}} \leftarrow \textsc{normalisationDir}({}^{h}\boldsymbol{p}^{\mathrm{sim}}, {}^{h}d_{\mathrm{ocp}}, {}^{r}\boldsymbol{p}^{\mathrm{sim}}, {}^{r}d_{\mathrm{cr}}, {}^{r}_{\max}v_{\mathrm{lin}})$
            $\textsc{append}\left({}^{\mathbb{H}}\mathbf{dir}, \frac{{}^{h,r}\mathrm{dir}^{\mathrm{sim}}}{{}^{h,r}\mathrm{dir}^{\mathrm{sim}}_{\mathrm{nrm}}}\right)$ ▷ Extend the list with normalised discomfort values
        end for
    end for
    return $\textsc{max}({}^{\mathbb{H}}\mathbf{dir})$ ▷ Return the maximum discomfort value
end function

proposed in [399]. The corresponding cost function, ${}^{r,\mathbb{H}}\mathrm{cfun}^{n}_{\mathrm{psi}}$, exploits the normalised value of the metric and calculates the rating of a trajectory as in Alg. 9.

Algorithm 9 Computing the value of the ${}^{r,\mathbb{H}}\mathrm{cfun}^{n}_{\mathrm{psi}}$ cost function for a trajectory ${}^{r}\mathrm{traj}^{n}$ generated for the robot r at time t^{n}

function ${}^{r,\mathbb{H}}\mathrm{cfun}_{\mathrm{psi}}({}^{r}\mathrm{traj})$
    ▷ Container with discomfort values of each human throughout the planning horizon
    ${}^{\mathbb{H}}\mathbf{psi} \leftarrow \emptyset$
    ▷ Retrieve predicted human trajectories with pose covariances (thus~accent)
    ${}^{\mathbb{H}}\widetilde{\mathbf{traj}} \leftarrow \textsc{getPredTrajs}(\mathbb{H})$
    for each ${}^{h}\widetilde{\mathrm{traj}} \in {}^{\mathbb{H}}\widetilde{\mathbf{traj}}$ **do** ▷ Iterate over predicted human trajectories
        for $\mathrm{sim} \leftarrow n$ **to** $n + q_{\mathrm{hor}}$ **do** ▷ Iterate over IDs of planning horizon timestamps
            ▷ Retrieve poses (with a covariance) and a velocity at a given step
            ${}^{h}\boldsymbol{p}^{\mathrm{sim}}, {}^{h}\boldsymbol{\Sigma}_{p}^{\mathrm{sim}}, {}^{h}\boldsymbol{v}^{\mathrm{sim}} \leftarrow \textsc{unpack}({}^{h}\widetilde{\mathrm{traj}}, \mathrm{sim})$
            ${}^{r}\boldsymbol{p}^{\mathrm{sim}} \leftarrow \textsc{unpack}({}^{r}\mathrm{traj}, \mathrm{sim})$
            ▷ Calculate the value of the indicator
            ${}^{h,r}\mathrm{psi}^{\mathrm{sim}} \leftarrow \textsc{discomfortPsi}({}^{h}\boldsymbol{p}^{\mathrm{sim}}, {}^{h}\boldsymbol{\Sigma}_{p}^{\mathrm{sim}}, {}^{h}\boldsymbol{v}^{\mathrm{sim}}, {}^{r}\boldsymbol{p}^{\mathrm{sim}})$
            ▷ Calculate the value of the normalisation factor
            ${}^{h,r}\mathrm{psi}^{\mathrm{sim}}_{\mathrm{nrm}} \leftarrow \textsc{normalisationPsi}({}^{h}\boldsymbol{p}^{\mathrm{sim}}, {}^{r}\boldsymbol{p}^{\mathrm{sim}})$
            $\textsc{append}\left({}^{\mathbb{H}}\mathbf{psi}, \frac{{}^{h,r}\mathrm{psi}^{\mathrm{sim}}}{{}^{h,r}\mathrm{psi}^{\mathrm{sim}}_{\mathrm{nrm}}}\right)$ ▷ Extend the list with normalised discomfort values
        end for
    end for
    return $\textsc{max}({}^{\mathbb{H}}\mathbf{psi})$ ▷ Return the maximum discomfort value
end function

F-formations' O-spaces intrusion The spatial patterns of human F-formations were examined in [75], where certain areas of human group arrangements were specified, with O-spaces being one of them. The O-spaces are areas reserved for the participants of a focused interaction, so a robot interacting in an unfocused way ought not to cross the O-spaces. In the *SRPB*, we proposed the m_{fsi} metric that intends to penalise a robot for traversing through O-spaces. The *HUMAP* implements the ${}^{r,\mathbb{G}}\mathrm{cfun}^{n}_{\mathrm{fsi}}$ cost function that aims to replicate the behaviour of m_{fsi} metric but for the online trajectory planning. The employed approach is described in Alg. 10. Notably, the weight of the cost function might be zeroed out, once the global goal is detected to be located within an O-space.

Passing speed Maintaining appropriate robot speeds when passing humans is also of substantial importance in unfocused interactions (Sec. 2.3.3). The recent user study

Algorithm 10 Computing the value of the ${}^{r,\mathbb{G}}\text{cfun}^{n}_{\text{fsi}}$ cost function for a trajectory ${}^{r}\text{traj}^{n}$ generated for the robot r at time t^{n}

function ${}^{r,\mathbb{G}}\text{cfun}_{\text{fsi}}({}^{r}\text{traj})$
  $\triangleright$ Container with discomfort values of each F-formation throughout the planning horizon
  ${}^{\mathbb{G}}\mathbf{fsi} \leftarrow \emptyset$
  $\triangleright$ Retrieve predicted F-formation trajectories with supplementary data (thus $\tilde{}$ accent)
  ${}^{\mathbb{G}}\widetilde{\mathbf{traj}} \leftarrow \textsc{getPredTrajs}(\mathbb{G})$
  for each ${}^{g}\widetilde{\text{traj}} \in {}^{\mathbb{G}}\widetilde{\mathbf{traj}}$ **do** $\triangleright$ Iterate over predicted F-formation trajectories
    for $\text{sim} \leftarrow n$ **to** $n + q_{\text{hor}}$ **do** $\triangleright$ Iterate over IDs of planning horizon timestamps
      $\triangleright$ Retrieve poses (with a covariance) and F-formation data at a given step
      ${}^{g}\boldsymbol{p}^{\text{sim}}, {}^{g}\boldsymbol{\Sigma}^{\text{sim}}_{p}, {}^{g}d^{\text{sim}}_{x}, {}^{g}d^{\text{sim}}_{y} \leftarrow \textsc{unpack}({}^{g}\widetilde{\text{traj}}, \text{sim})$
      ${}^{r}\boldsymbol{p}^{\text{sim}} \leftarrow \textsc{unpack}({}^{r}\text{traj}, \text{sim})$
      $\triangleright$ Calculate the value of the indicator
      ${}^{g,r}\text{fsi}^{\text{sim}} \leftarrow \textsc{discomfortFsi}({}^{g}\boldsymbol{p}^{\text{sim}}, {}^{g}\boldsymbol{\Sigma}^{\text{sim}}_{p}, {}^{g}d^{\text{sim}}_{x}, {}^{g}d^{\text{sim}}_{y}, {}^{r}\boldsymbol{p}^{\text{sim}})$
      $\triangleright$ Calculate the value of the normalisation factor
      ${}^{g,r}\text{fsi}^{\text{sim}}_{\text{nrm}} \leftarrow \textsc{normalisationFsi}({}^{g}\boldsymbol{p}^{\text{sim}}, {}^{r}\boldsymbol{p}^{\text{sim}})$
      $\textsc{append}\left({}^{\mathbb{G}}\mathbf{fsi}, \frac{{}^{g,r}\text{fsi}^{\text{sim}}}{{}^{g,r}\text{fsi}^{\text{sim}}_{\text{nrm}}}\right)$ $\triangleright$ Extend the list with normalised discomfort values
    end for
  end for
  return $\textsc{max}({}^{\mathbb{G}}\mathbf{fsi})$ $\triangleright$ Return the maximum discomfort value
end function

presented in [72] examines the effect of robot speed on comfortable human passing distances. Their discrete findings were approximated[17] with a bicubic spline with fourth-order continuity along both the speed and distance dimensions (Fig. 5.17)[18] and added (as m_{psd}) to the set of metrics[19] evaluated by *SRPB*.

The normalised value of the metric evaluating human discomfort induced by the robot's passing speed and distance is used in the ${}^{r,\mathbb{H}}\text{cfun}^{n}_{\text{psd}}$ spatiotemporal cost function to penalise the robot for not adhering to the least obtrusive passing speeds. The employed approach is described in Alg. 11.

Algorithm 11 Computing the value of the ${}^{r,\mathbb{H}}\text{cfun}^{n}_{\text{psd}}$ cost function for a trajectory ${}^{r}\text{traj}^{n}$ generated for the robot r at time t^{n}

function ${}^{r,\mathbb{H}}\text{cfun}_{\text{psd}}({}^{r}\text{traj})$
    ▷ Container with discomfort values of each human throughout the planning horizon
    ${}^{\mathbb{H}}\mathbf{psd} \leftarrow \emptyset$
    ▷ Retrieve predicted human trajectories
    ${}^{\mathbb{H}}\mathbf{traj} \leftarrow \text{GETPREDTRAJS}(\mathbb{H})$
    for each ${}^{h}\text{traj} \in {}^{\mathbb{H}}\mathbf{traj}$ **do** ▷ Iterate over predicted human trajectories
        for sim $\leftarrow n$ to $n + q_{\text{hor}}$ **do** ▷ Iterate over IDs of planning horizon timestamps
            ▷ Retrieve poses and a velocity at a given step
            ${}^{h}\boldsymbol{p}^{\text{sim}} \leftarrow \text{UNPACK}({}^{h}\text{traj}, \text{sim})$
            ${}^{r}\boldsymbol{p}^{\text{sim}}, {}^{r}\boldsymbol{v}^{\text{sim}} \leftarrow \text{UNPACK}({}^{r}\text{traj}, \text{sim})$
            ▷ Calculate the value of the indicator
            ${}^{h,r}\text{psi}^{\text{sim}} \leftarrow \text{DISCOMFORTPSD}({}^{h}\boldsymbol{p}^{\text{sim}}, {}^{r}\boldsymbol{p}^{\text{sim}}, {}^{r}\boldsymbol{v}^{\text{sim}})$
            ▷ Calculate the value of the normalisation factor
            ${}^{h,r}\text{psi}^{\text{sim}}_{\text{nrm}} \leftarrow \text{NORMALISATIONPSD}({}^{h}\boldsymbol{p}^{\text{sim}}, {}^{r}\boldsymbol{p}^{\text{sim}}, {}^{r}d_{\text{cr}}, {}^{r}_{\max}v_{\text{lin}})$
            $\text{APPEND}\left({}^{\mathbb{H}}\mathbf{psd}, \frac{{}^{h,r}\text{psi}^{\text{sim}}}{{}^{h,r}\text{psi}^{\text{sim}}_{\text{nrm}}}\right)$ ▷ Extend the list with normalised discomfort values
        end for
    end for
    return $\text{MAX}({}^{\mathbb{H}}\mathbf{psd})$ ▷ Return the maximum discomfort value
end function

[17] `https://github.com/rayvburn/social_nav_utils`

[18] The Matlab's cubic spline interpolation method was used: `https://www.mathworks.com/help/curvefit/csapi.html`

[19] Since the initial release, the software package with *SRPB* has been developed in terms of metrics supported and user tools.

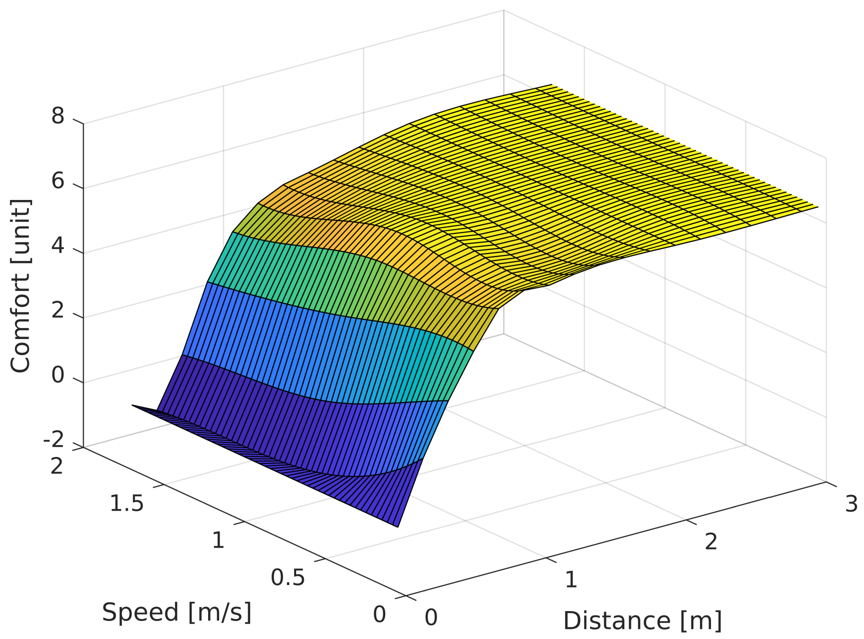


Figure 5.17: A visualisation of the function approximating human discomfort based on the robot's speed and the distance between the robot and a human.

5.8 Summary

In this chapter, *HUMAP* – the system that solves the problem of receding horizon trajectory planning for holonomic and differential drive robots operating in unstructured environments has been presented. The *HUMAP* is a geometric planner, whose objective function regards navigation requirements from both classical and human-aware perspectives.

Although the functioning of the *HUMAP*'s trajectory planning scheme involves multiple behaviours orchestrated with the *FSM*, the typical operational behaviour, designated for unfocused human-robot interactions, employs a hybrid approach of generating kinodynamically feasible trajectory candidates and scoring them with spatiotemporal cost functions evaluating the robot performance, robot motion naturalness and human discomfort.

The first trajectory generation method relies on the *SFM*-based pedestrian motion model [1], which allows for incorporating realistic collision avoidance, as its parameter values were calibrated on the basis of real-world data. The employed baseline model's formulation was extended with the *Fuzzy Inference System* (*FIS*) component that emphasises anticipative collision-avoidance actions while reproducing customary pedestrian motion rule of passing on the right (**Req. 2.4.4**). Since the employed pedestrian motion model is formulated deterministically, multiple trajectory candidates are produced from the model by supplying it with various parameter sets. The spread of each parameter value is determined based on their uncertainties assessed during the calibration process [25].

The second trajectory generation approach is well-established and samples the set of feasible velocities to produce uniformly curved trajectories [22]. While the velocity sampling generator covers a discretised space of admissible controls, the model-based generator creates concentrated trajectories that avoid collisions and follow the local goal located along the global path. The procedure of producing model-based trajectories as well as candidates generated with the velocity sampling method is explained using pseudocodes (Alg. 4 and 5).

In the *HUMAP*'s planning scheme, all trajectory candidates obtained from two generators are quantitatively assessed. The cost functions employed for the evaluation of trajectories regard various aspects of robot navigation – from task execution performance, through the robot's motion naturalness, to the humans' physical and perceived safety. Scoring of the human awareness of the robot trajectories is performed using study-based indicators of human discomfort relevant for social robot navigation discussed in Chapter 4.

The planner is context-aware and numerous predicates are used to orchestrate the multi-behaviour operation of its *Finite State Machine*. In each calculation step, the analysis of the environment state is performed to compute the predicate values, that directly influence the state in which the planner operates. An example of environmental context awareness is adjusting the robot's behaviour once a human is expected to cross the robot's planned path. Another case is detecting the inability to plan the global path, which might be caused by the sudden free space occlusion as well as an outdated environment model. In such a situation, the *HUMAP* undertakes special actions to obtain updated observations of the robot's surroundings. Additionally, due to the planner's awareness of interpersonal contexts, the weight of the cost function penalising the intrusions into O-spaces of F-formations is dynamically changed once the global goal is detected to be placed within the bounds of an O-space.

The proposed planner fulfils numerous requirements from the taxonomy discussed in Sec. 2.1. The navigation performance necessities are explicitly included in the problem formulation (5.6), including collision avoidance (**Req. 1.1**), generating feasible trajectories (considering kinodynamic constraints, **Req. 1.2**), and capability to reach goal poses (**Req. 1.3**). Additionally, the objective function takes into account requirements related to the social perspective of navigation, e.g., the physical safety of humans (**Req. 2.1** realised by the *TTC* cost function), and perceived safety of humans, which include: avoiding personal space intrusions (**Req. 2.2.1**), avoiding crossing O-spaces of F-formations (**Req. 2.2.2**), modulating speed when passing humans (**Req. 2.2.3**), and the avoidance of heading straight into humans (**Req. 2.2.4**, motion legibility). Moreover, the robot's motion naturalness concepts included in the objective function are the smoothness of the robot's velocity profile (**Req. 2.3.1.1**) and the avoidance of oscillating (**Req. 2.3.1.2**) and backward (**Req. 2.3.1.4**) motions. Notably, a social convention is also implemented on the behavioural level of the planner (orchestrated by the *FSM*), namely, yielding a way to a human is performed once an individual crosses the robot's path (**Req. 2.4.5**). To the best of our knowledge, the *HUMAP* planner covers social robot navigation requirements the most extensively amid the state-of-the-art frameworks.

The collision-free motions of the *HUMAP* are guaranteed by the ${}^{r,E}\mathrm{cfun}_{\mathrm{trav}}$ cost function, as it rejects candidate trajectories leading to the collisions according to the dense environment model. This property applies to the operation in static environments even if only partial observability of an environment is provided, but requires the full observability of the environment in the local context. The algorithm can generate collision-free motions as long as it is aware of all obstacles in its proximity; therefore, possessing an accurate environment model is crucial. Although there are no guarantees regarding the completeness of the algorithm, as it uses gradient-based cost functions for scoring movements towards the global goal (Sec. 5.7.1 and Fig. 5.16), the planner obtained the highest robustness rates in diverse test scenarios described in Sec. 7.4.

Chapter 6

Implementation

The implementation of a robot control system is crucial for the preparation of this thesis, as it enables the execution of experiments and the evaluation of the effectiveness of various motion planning algorithms. The goal was to create a flexible and modular system to easily exchange system components for testing different methods and effectively comparing their results. Since the comparison of various strategies was meant to be conducted in both simulated and real-world environments, additional constraints were taken into consideration when designing the system. This chapter focuses on describing the implementation of the robot control system, which served as the platform for experiments.

6.1 System structure

The implementation concepts primarily regard the structure of the system, as the behavioural aspects are related to a specific planner configuration selected for the operation, which is instantiated during the system startup. The only task that the described system must be capable of is navigation, but it can be extended to accomplish complex tasks relying on navigation.

The basic organisation of the system structure is presented in Fig. 6.1, which illustrates components arranged into three main groups: *Robot Hardware Platform*, *Perception*, and *Motion Planning*.

Robot Hardware Platform The *Robot Hardware Platform* group is related to the direct management of robot onboard resources that observe the robot's state and the environment, and act on it. Namely, the *Mobile Base Controller* concerns the low-level motion control that is executed by the robot hardware controllers (usually *PID*). In contrast, *Sensor Drivers* perform communication with the robot sensors to obtain the most recent readings, convert them into a unified format for a certain sensor type, and send the

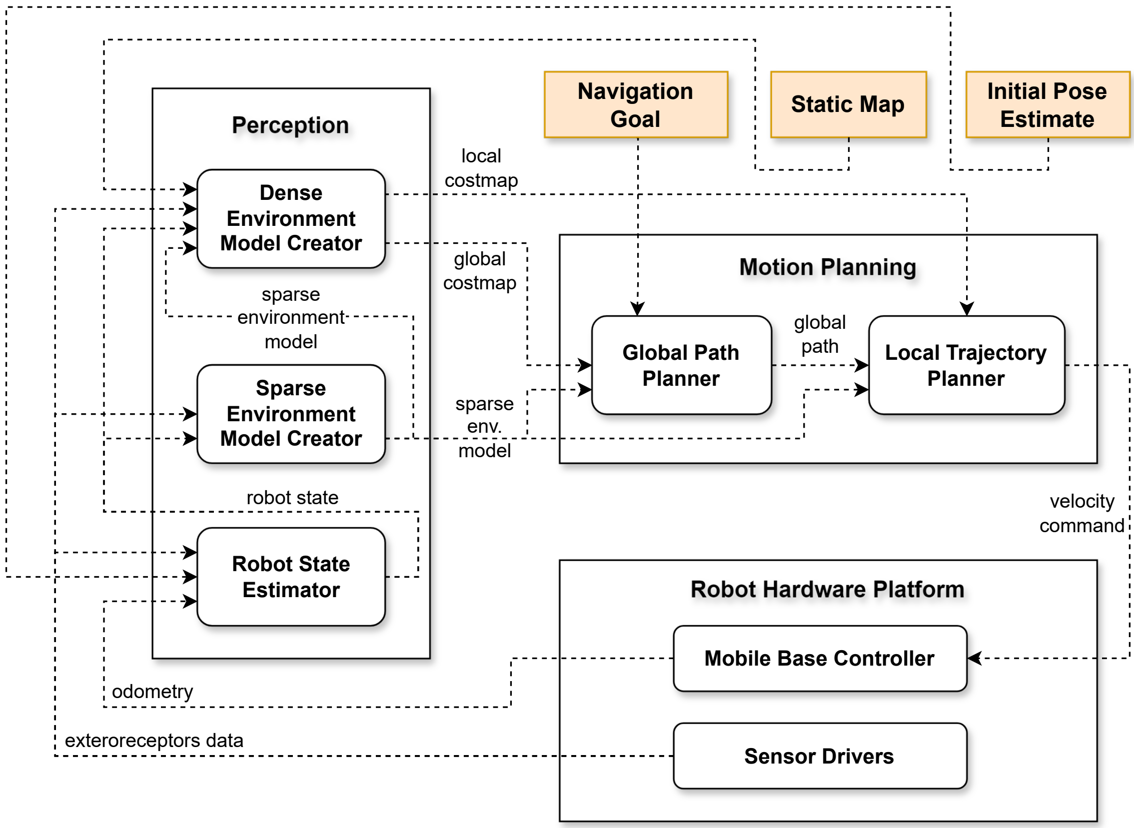


Figure 6.1: General block diagram of the robot control system. Inputs to the system are marked with colour, whereas communication channels between functional blocks are indicated with dashed lines.

prepared messages to the rest of the system. The produced *exteroreceptors data* (readings from sensors observing the environment [31]) is specific to a robot and in the implemented system consists of the *LiDAR* and *RGB-D* camera data. In contrast, the *odometry data* constitutes the result of a dead reckoning procedure according to the proprioceptors (sensors assessing the robot's state without observing the environment).

Perception Another organisational group of the system is *Perception*, which aggregates *Robot State Estimator*, *Sparse Environment Model Creator*, and *Dense Environment Model Creator*. Specifically, the *Robot State Estimator* is related to the global and local pose estimation – performs data fusion for obtaining accurate localisation estimates frequently. Separate coordinate systems (local and global) were distinguished for localisation (and also motion planning) according to [255]. Typically, the origin of the global coordinate system is set to the centre of the static map of the environment, whereas the origin of the local coordinate system is located at the pose of the robot at the control system's startup. Therefore, the system input, *Initial Pose Estimate*, defines the accurate homogenous transformation between the origins of the local and global coordinate systems, which facilitates reliable operation from the beginning of the scenario execution.

Another functional block within the *Perception* group is the *Sparse Environment Model Creator*, which uses *exteroreceptors data* to extract features from the robot environment in order to prepare a sparse environment representation. In the described system, the *sparse environment model* contains, e.g., human and F-formation data; hence, this block involves human detection and tracking modules (Sec. 6.7). The internal block diagram that is applicable for operation with most investigated trajectory planners (Sec. 6.4) is schematically presented in Fig. 6.2, while its modified representation, shown in Fig. 6.3, is used by *TEB*-based trajectory planners and *HUMAP*. Namely, the extended version contains the *Costmap Converter*, which processes a *local costmap*[1] to obtain environment obstacles in a segmented form, i.e., sparse representation of robot obstacles identified as geometric primitives. The data aggregated by the *sparse environment model* is shown in the output ports in Fig. 6.2 and 6.3. Moreover, the *robot state* is used by the *Human Detection and Tracking* as a reference (e.g., pose) when calculating the humans and F-formations information.

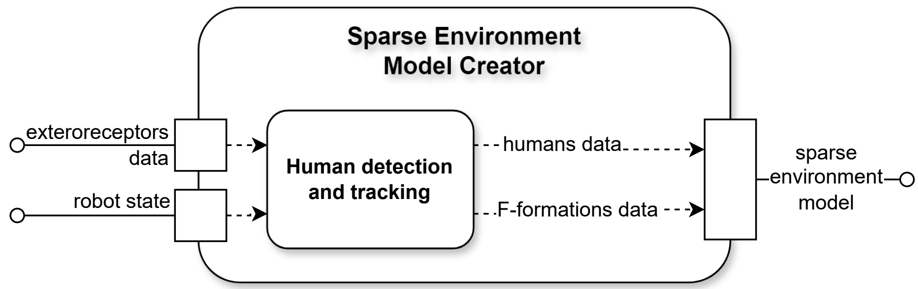


Figure 6.2: Schematic representation of the internal block diagram of the *Sparse Environment Model Creator* applicable for most trajectory planners.

The last functional block from the *Perception* group is *Dense Environment Model Creator*, which aims to create metric maps of the environment. Firstly, the *Static Map* of the environment, which serves as a system input, is used to create a global environment representation (in the form of an occupancy grid), and real-time *exteroreceptors data* are used to mark obstacles in a resultant occupancy grid (new obstacles might be added, but obstacles from the *Static Map* cannot be cleared). Similarly, the *exteroreceptors data* is also used to update a local occupancy grid, which is developed only from the sensor observations (without the *Static Map*). The occupancy grids are then transformed into costmaps, which might also include information encoded in the *sparse environment model*

[1] The general block diagram (Fig. 6.1) does not contain a communication channel between the *Dense Environment Model Creator* and *Sparse Environment Model Creator*, while in the Fig. 6.3 a port expecting a *local costmap* (generated by the *Dense Environment Model Creator*) appears. This is intentionally presented in this manner to enhance the clarity of the main diagram.

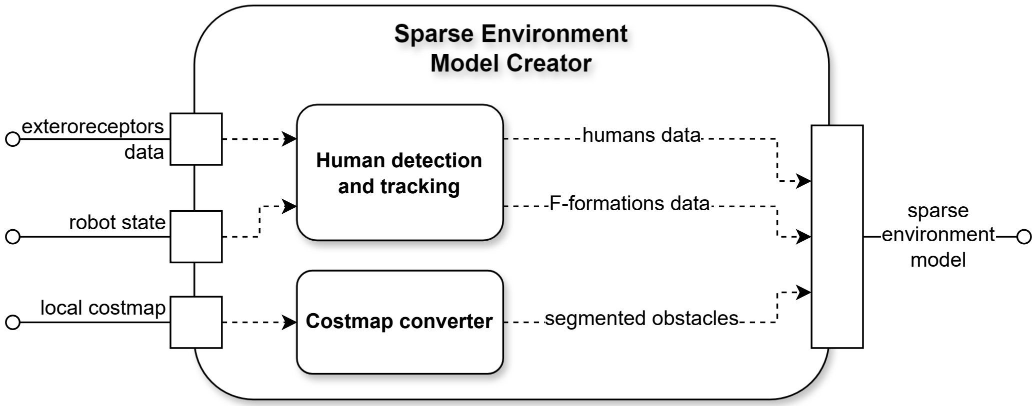


Figure 6.3: Schematic representation of the internal block diagram of the *Sparse Environment Model Creator* applicable for *TEB*-based and *HUMAP* trajectory planners.

(explained in detail in Sec. 6.3). For brevity, it is assumed that both the *global costmap* and *local costmap* contain the robot's state in the form of a current pose (expressed in the coordinate system relevant to the planner) and the velocity expressed in the mobile base frame.

Motion Planning The third group included in the system structure is *Motion Planning*, which utilises environment models developed by the *Perception* modules to plan a robot action in the form of a *velocity command*. The group consists of the *Global Path Planner* and *Local Trajectory Planner*. The *Global Path Planner* computes a *global path* according to the current robot state and the *Navigation Goal*, which is the system input represented by a pose. The calculated *global path* is passed to the *Local Trajectory Planner*, which solves the problem of receding horizon trajectory planning – computes the entire trajectory for a given horizon, but only the first *velocity command* is applied.

For brevity, it is assumed that traditional local trajectory planners do not make use of the communication channel providing *sparse environment model*, which contains *humans data* and *F-formations data*.

Adapter components The system was designed with a focus on compatibility with various motion planning algorithms, leading to a structure that includes additional elements utilised only in specific launch configurations (with certain trajectory planners). These variations in the system's structure are primarily driven by the utilisation of learning-based algorithms, which are usually implemented using dedicated libraries and frameworks; hence, their usage with typical robotic frameworks requires the development of additional interfaces.

Notably, some planners that have been integrated with the system (Sec. 6.4) require specialised representation of data; hence, adapter components need to be implemented.

However, those were not shown in the schematic block diagram (Fig. 6.1), as they often rely on converting, e.g., a *LiDAR* scan to the form that has smaller resolution compared to the robot's sensor (employed with the *RG's DRL* trajectory planner), or merging *Point Clouds* representing obstacles detected by *LiDAR* and *RGB-D* camera. All additional system components developed according to the *adapter* design pattern are published as open-source software.

6.2 System implementation tools

The implementation of the proposed system exploits the *Robot Operating System* (*ROS*) – version 1 [180][2], which is the most popular robotic framework nowadays. *ROS*, being a framework, provides a collection of libraries and development tools for building robotic systems, as well as ready-to-use algorithms in a modular form. *ROS* facilitates integrating robotic systems, as it offers a distributed computing environment, allowing nodes (basic organisational entities of *ROS* systems) to communicate with each other over a network. Furthermore, the integrated visualisation tool, *Rviz* aids in testing and debugging complex systems by providing an interactive 3D environment representing the robot's perception. *ROS* contributed to the extensive usage of component-based systems in robotics, since its modular architecture enables the seamless integration of sensors or algorithms and using different implementations interchangeably. Additionally, *ROS* advocates community collaboration due to its open-source nature.

In the implemented system, each functional block presented (Fig. 6.1) is realised by one or multiple *ROS* nodes. Moreover, the development of the navigation system for the experiments described in Chapter 7 led to the implementation of numerous open-source packages for *ROS* that are utilised by the robot launch system.

The *ROS* is closely integrated with numerous high-fidelity simulators (Sec. 3.3.3), but the *Gazebo* (*Classic* version) was selected as the main simulation platform for testing and experiments. The main reason is that the manufacturer of the robot, which has been extensively used during experiments, integrated a well-developed simulation model with *Gazebo*. The flexible architecture of the selected simulator also allowed for integrating the human behaviour control framework (discussed in Sec. 6.8), which facilitated the conducting of simulation experiments.

The paramount feature of *ROS* is the `launch` tool that allows automating the system instantiation ("bringup") with different configurations of parameters or algorithms used.

[2] *ROS*1 was selected instead of *ROS*2 since the target robot for experiments has factory-installed *ROS*1 Melodic. Hence, the usage of *ROS*2 would require a significant amount of additional integration work when porting the system from simulation to real hardware.

Its nesting characteristic (ability to include other `launch` files in the main `launch` file) has been widely used for developing our system, which is intended to be started in various configurations.

6.3 Navigation ecosystem

The control system developed for the studies integrates the *ROS*1 navigation system, whose structure consists of a global and a local [147, 12, 170] planners. In *ROS*, the navigation system is implemented as a monolithic structure with the orchestrating `move_base` node that aggregates configurable planners. The reference system was designed with the idea of modularity; hence, allows developing customised planning components and loading them as plugins in the system runtime. The usage of *ROS* navigation facilitated the implementation of our system, whose primary goal is the ability to be launched in various configurations, i.e., with different global path planners, local trajectory planners, or costmap layers.

Namely, the navigation system integrates path and trajectory planners with multilayer global and local costmaps [170] that take into account real-time obstacle detection, and robot size (gradient cost around obstacles). Human proxemics, modelled as spatial costs [47], can also be embedded in costmaps. Additionally, the global costmap includes a preprepared static map of the environment [147].

In all experiments with the explained setup (Sec. 7), the wavefront Dijkstra's algorithm[3] has been used as the robot's global path planner (operating at 2 Hz). Only trajectory planners were swapped, utilising public *ROS*-interfaced implementations of the examined algorithms. Moreover, in all experimental scenarios, the robot operated with the same preprepared map.[4] Nevertheless, environment obstacles were detected in real-time by the robot sensors and added to the costmaps (of global and local planners), making the robot resistant to the changes not captured in the map. For the global pose estimation, the *AMCL*[5] [417] algorithm was used.

The `move_base` node detects in real-time whether the global path planner and local trajectory planner find solutions to the problem at the current state of the robot in the environment. To increase the robustness of the navigation ecosystem, the orchestrator investigates whether the timeouts for finding the solutions have been exceeded. Namely, once the timeout expires, the space-clearing operations on costmaps are performed, so

[3] `https://wiki.ros.org/navfn`

[4] The layout of the laboratory equipment had changed between the first (Sec. 7.3) and the second phase of the experiments (Sec. 7.4); therefore, the static map had to be adjusted. The same map was used in the corresponding real-world and simulation scenarios.

[5] `https://wiki.ros.org/amcl`

any persisting outdated environment observations do not influence the planning process. If a valid solution is not found during the next planning iteration, the navigation task is aborted. The same scheme applies to both global path planner and local trajectory planner, but their timeouts are different, namely – 0.5 s and 3.0 s, respectively. Additionally, an oscillation timeout is defined as 10 s; hence, if the robot does not perform any significant progressive or rotational movements, the navigation task might also be aborted.

6.4 Planners integration

The comprehensive conduction of comparative experiments of the proposed *HUMAP* with the state-of-the-art trajectory planners requires integrating numerous approaches with the robot control system. While the structure of the proposed system is planner-dependent (Sec. 6.1), most *ROS*-based planners, acting as direct plugins for the *ROS* navigation system, can operate without additional components such as adapters or converters. However, learning-based planners usually need to be tailored for a specific robot, e.g., when transferring the policy learned with a different one. Therefore, the system instantiation procedure accounts for certain planner-specific adjustments that have to be performed with some planner implementations to prepare them for integration with the rest of the system.

Research algorithms are often adapted for a specific mobile base; therefore, their parameterisation is often required at the first stage to ensure valid operation with other robots. Those parameters are related to the kinematic or kinodynamic constraints of the mobile base, but also to the interfaces with the remaining part of the system. Most of the implementations of examined algorithms underwent minor changes regarding build or execution issues, as some packages (or their dependencies) were incompatible with the desired operating system or *ROS* distribution (*ROS* Melodic used). Nevertheless, no functional changes were applied to the algorithms to degrade or enhance their navigation performance.

The testbed system was integrated with several learning-based approaches. Such planners nominally operate outside of *ROS*, using popular frameworks such as Stable Baselines (*RG's DRL* and *DRL-VO*), TensorFlow (*GA3C-CADRL*), and specialised simulation environments like CrowdNav (*SARL* and *SARL**). Nevertheless, all planners selected for integration with our system had *ROS* interfaces implemented.

The *RL*-based methods are claimed to be prone to the lack of generalisation to different environments [418]. Not all implementations of the *RL* planners were prepared by the authors to be fine-tuned for adapting the algorithm to new environments. On the other hand, training algorithms from scratch is impractical and easily undermined by factors such as insufficient training duration or training on inadequately fast hardware, as the authors do not always explicitly declare the training time and setup. Therefore, the overarching goal was to use policies provided by the algorithms' designers to prevent the influence of underfitting or overfitting of the algorithms. Due to the specificities of the mobile bases on which the methods were trained, to use the original policies with another robot, it was necessary, for example, to adjust sensor characteristics (e.g., resolution of the *LiDAR* data), correct robot dimensions, or adapt kinematic constraints (all algorithms). Additionally, for the planners using *LiDAR* data to describe the environment, the 2.5D representation of the world was prepared, i.e., the scans from the robot's *LiDAR* and *RGB-D* camera were projected onto the base plane to provide a comprehensive understanding of nearby obstacles. However, the conducted experiments are easily reproducible as the modified implementations are publicly available.[6]

The learning-based navigation approaches are usually developed using different libraries and interfaced with *ROS* in isolation from the standard, plugin-based navigation framework. Therefore, the `external_local_planner`[7] trajectory planning plugin was developed for easier interfacing with externally-operating approaches. The plugin is configurable in terms of providing a local goal (located along the globally planned path) or enabling the "in-place rotation" behaviour once the robot reaches the goal position. Delivering local goals facilitates the navigation with planners, like *SARL**, that expect a periodically provided local goal, as they do not rely on the global path explicitly. Moreover, the "in-place rotation" behaviour (preceded by the nominal behaviour of moving towards the target) allows the robot to reach the goal pose operating with algorithms that only investigate the goal position, disregarding the orientation, which applies to most learning-based approaches. Therefore, this extension enables objectively comparing the overall navigation performance of all algorithms integrated with the target robot.

6.5 Environment models used by trajectory planners

As human proxemics [47] can be modelled as soft, spatial constraints around detected individuals in costmaps, the specialised environment models have been prepared to be used by all planners. Specifically, layered costmap architecture enables tracked humans to be embedded as bivariate Gaussians into the costmaps representing the robot's human-aware

[6] The source code for the *GA3C-CADRL* planner is available at `https://github.com/rayvburn/cadrl_ros`, for *SARL* and *SARL** at `https://github.com/rayvburn/sarl_star`, for *RG's DRL* at `https://github.com/rayvburn/drl_local_planner_ros_stable_baselines`, and for *DRL-VO* at `https://github.com/rayvburn/drl_vo_nav`.

[7] `https://github.com/rayvburn/external_local_planner`

environment model used for planning [84].[8] Additionally, the reference *ROS* implementation package has been modified and extended.[9] The substantial contribution is that the expanded package accounts for the human tracking uncertainty and introduces a new F-formation layer.

The consequence of utilising extended environment representation (with, e.g., spatial costs reflecting proxemics) is that the global path planner and both classical and human-aware local trajectory planners use the environment model that captures personal spaces and O-spaces of F-formations embedded as spatial costs in the costmaps [71]. Such system configuration enables effective evaluation of the trajectory planners' isolating the actual planning scheme from the underlying environment model.

However, among the evaluated navigation approaches, there are *RL*-based trajectory planners that do not make a (full) use of costmap extensions. For example, *RG's DRL* does not employ either a map or global path planning, similar to *GA3C-CADRL*, which uses a fully sparse environment model. In contrast, *SARL* and *SARL** make use of the global costmap extensions, but only *SARL** relies on a globally planned path (implicitly, as it expects poses of subsequent local goals). On the other hand, *DRL-VO* navigates using a context-aware global path plan, but the policy calculation is supported by a mere occupancy grid.

Notably, we aimed to use the same *SRL-EBand* planner configuration as originally evaluated in [240]. Therefore, to avoid interfering with the integration of the global path planner and *SRL-EBand* planner, this method operated with the different global costmap configuration (proposed in the referenced work) compared to other trajectory planners.

6.6 Managing different system configurations

The proposed robot control system is prepared for being instantiated in different configurations, which is achieved by the usage of the *ROS* `launch` tool and a proper file organisation.

Simulation and real-world setup The robot control system has been prepared to switch from the simulation environment to the real-world environment without any source code modifications. The reason behind this decision is that the analogous scenarios for experiments conducted for this thesis aimed to be performed in simulation and real-world environments,

Therefore, the component-based system has been "virtually" divided into parts that

[8] `https://wiki.ros.org/social_navigation_layers`
[9] `https://github.com/rayvburn/navigation_layers`

operate in unchanged form both in the simulation and in the real world, as well as into simulation-specific and real-world-specific components. Naturally, the source code for the *ROS*-based navigation launch system (tailored for the TIAGo robot but easily generalisable to other platforms) has also been divided into parts. Namely, the *common* part constituting the configuration and interfaces for motion planning and perception,[10] and domain-specific extensions: *sim* for starting the simulated environments with a virtual robot[11] and *real* for preparing and launching components on the real hardware[12]. Once the *sim* and *real* parts of the control systems are properly integrated with the rest of the system instantiated on the basis of the *common* part, the system does not require any source code changes when launched in *sim* or *real* configuration.

Parameterisation of launched components The configuration of the system that needs to be instantiated is dictated by the various arguments selected by the user (or a script). Namely, the highest level arguments, e.g., which local planner (`local_planner`) and global planner (`global_planner`) to use, or which costmap configuration should be applied (`costmap_contexts`), are specific to the evaluated case; hence, must be externally provided to the main `launch` file. Those crucial arguments are further injected into the nested parts of the main `launch` file, so the final structure of the instantiated system is resolved in runtime (Fig. 6.4).

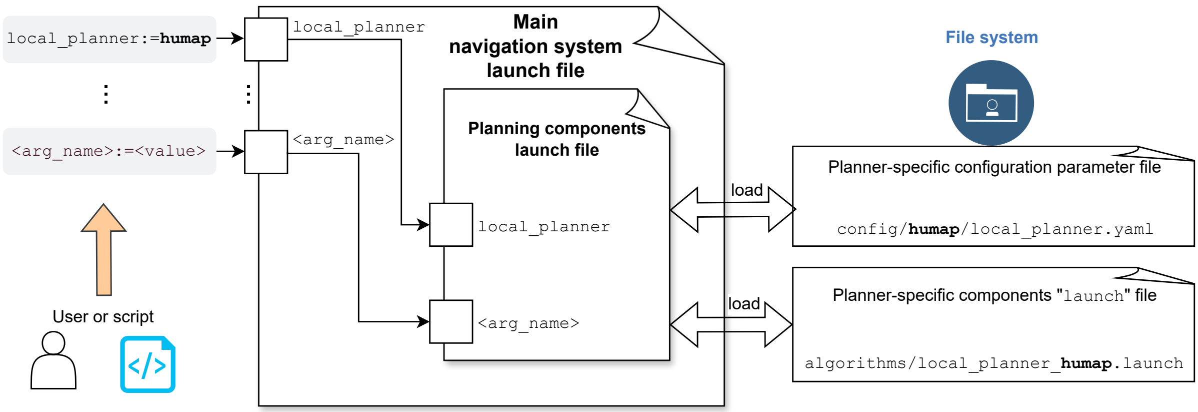


Figure 6.4: Procedure of external argument injection into the hierarchical `launch` setup using the *HUMAP* trajectory planner as example. Note that `<arg_name>` represents any argument and *Navigation components launch file* provides that all *ROS* nodes required for a given navigation system setup are started.

The main `launch` file of the navigation system launches the perception and motion planning components. As some software components might require special interfaces (ad-

[10] `https://github.com/rayvburn/tiago_social_robot`
[11] `https://github.com/rayvburn/tiago_social_robot_sim`
[12] `https://github.com/rayvburn/tiago_social_robot_real`

ditional adapting components) and different parameters, performing some adjustments, or even optional activities during the system instantiation must be available. Therefore, the mechanism allowing such behaviour is that the planning-related `launch` file includes another abstract `launch` file, with the name conforming to the `local_planner` argument value. Similarly, the planning components' `launch` prepares the path to the trajectory planner parameters file, filling up the predefined naming pattern with the `local_planner` argument value. For example: launching the system with `local_planner:=humap` additionally runs a nested `algorithms/local_planner_humap.launch`, in which some activities of system components can be started on demand, and loads the trajectory planner parameters that are located at `config/humap/local_planner.yaml`. Provided the relevant package files are systematically organised, the instantiation procedure will select the required files and launch the system in the desired form.

The simple rule to adhere to is to structure the source code of the system, so both the `launch` and trajectory planning parameter files are separated between individual algorithms. Nonetheless, parameterisation of the local trajectory planner that is intended to be started within the system is only an example, as similarly, costmap configurations (enabled layers [170]), initial pose estimates in different scenarios, and other parameters are organised in the source code according to a specific naming pattern. The appropriate file selection is resolved at runtime, ensuring adaptability to the requirements of different planners. The feature of launching additional interfacing components is also widely used in learning-based planning approaches that require, e.g., sensor data conversion or transforming human information to a different format. Default values of arguments are provided to disallow starting a malfunctioning system.

Overall, the fundamental aspect of the system usage (and extension) is the adherence to the file naming convention according to the expected argument names. The system prepared according to these rules is easily extendable and can be integrated with more algorithms for trajectory planning, global path planning, or human tracking.

6.7 Human detection and tracking

Detecting and tracking humans in the robot environment is one of the fundamental requirements of a socially navigating robot. Namely, the *SPENCER* human perception stack[13] was employed in the proposed system as the *Human Detection and Tracking* functional block. It provides information about human poses along with estimation uncertainties represented by covariances of Gaussian distributions, as well as human velocities. Moreover, human relations are also estimated, and on this basis, F-formation membership

[13] `https://github.com/spencer-project/spencer_people_tracking`

data are assigned to each human.

The *SPENCER* human perception stack is capable of utilising multimodal sensor data [149]. The usage of only *LiDAR*-based human detections was tested and the results were often false positives. Therefore, detection modules were configured to primarily use *RGB-D* vision data supported by *LiDAR*-based detections once the tracked human is no longer visible to the camera. In all experiments, human tracking was performed by the robot's onboard sensors. Employing ground truth human poses in simulation was not an option, since the results from simulation experiments were intended to be directly compared with those from real-world tests (Sec. 7.2).

To facilitate the usage of the *SPENCER* detections with software packages relying on *ROS*' standardised `people_msgs`[14] (applies to several integrated trajectory planning methods), the component aggregating current detections into the *SPENCER*-specific format has been extended.[15] Namely, for broader compatibility, the original messages are converted to the standardised *ROS* `people_msgs`. Then, messages in such a form are processed with the usage of libraries from the `people_msgs_utils`,[16] package, as at the stage of developing `people_msgs` messages some additional information is encoded in a serialised form.

6.8 Simulating human behaviour

Conducting social robotics experiments in simulation assists the development of robot control systems considerably. For this purpose, the *Gazebo* simulation platform, which is closely integrated with *ROS*, has been used throughout the development stage but has also been employed for conducting experiments in virtual scenarios. Moreover, for controlling humans in virtual scenarios the *HuBeRo* framework[17] [377] has been used.

HuBeRo is a framework that simulates human behaviours typical for social robotics research tasks by providing navigation skills and realistic animation management for simulated human characters (actors). Additionally, given that the simulator provides a realistic 3D model of a person, the framework allows a more detailed examination of robot perception in the simulation. *HuBeRo* proved to be helpful in conducting various virtual tests.

The framework is simulator-agnostic, but the *Gazebo* interface has been developed. Namely, the *Gazebo*'s Actor plugin[18] is used as the provider of movable human postures.

[14]`http://wiki.ros.org/people_msgs`

[15]`https://github.com/rayvburn/spencer_people_tracking`

[16]`https://github.com/rayvburn/people_msgs_utils`

[17]`https://github.com/rayvburn/hubero`

[18]`http://classic.gazebosim.org/tutorials?tut=actor`

However, due to the plugin's implementation intricacies, the simulated actors are non-collision but detectable by robot sensors.

In all virtual experiments, *HuBeRo*-controlled dynamic human agents used online motion planning modules employing the same path planner as the robot and the *TEB* trajectory planner (due to the limited computational burden). The actors operated according to the scenario-specific initial and goal poses via defined waypoints. Their activities are explained in detail in Chapter 7.

6.9 HUMAP implementation

The proposed planning approach – *HUMAP*, is implemented in C++ programming language as a direct plugin to the *ROS*1 navigation ecosystem. It is organised in the open-source software package named `humap_local_planner`.[19]

Planner's structure Internal structure of the *HUMAP* is shown in Fig. 6.5 that illustrates which parts of the *HUMAP*'s planner implementation were developed from scratch, which were modified, and which are directly inherited from the `base_local_planner` package – an element of the *ROS*1 navigation ecosystem (Sec. 6.3). The activity of the `base_local_planner:: SimpleScoredSamplingPlanner` class, which aggregates trajectory generators and cost functions, is described in Alg. 2.

In the *HUMAP*, the hybrid method for producing trajectory candidates is used. The velocity sampling trajectory generator is used in an unchanged form implemented as the `base_local_planner::SimpleTrajectoryGenerator`, whereas the model-based candidates' generator is dedicated to the *HUMAP*. Notably, the original trajectory selection strategy implemented in the `base_local_planner::SimpleScoredSamplingPlanner` allows the use of multiple trajectory generation methods, but once the primary generator finds a valid trajectory, candidates from other generators are not considered. Therefore, the customisation[20] had to be implemented and relies on enabling the usage of multiple generators in an unconditional sequence.

Furthermore, the set of cost functions regarded during the trajectory scoring scheme has also undergone a substantial extension. Novel cost functions related mainly to the robot's human awareness (Sec. 5.7) are applied for scoring trajectory candidates. Moreover, several cost functions, mainly performance-related (Fig. 5.16), are directly inherited from the original *ROS* resources also implemented in the `base_local_planner` package.

The enhancements of the *ROS* navigation ecosystem involve all cost functions assessing

[19] `https://github.com/rayvburn/humap_local_planner`

[20] `https://github.com/ros-planning/navigation/pull/1201`

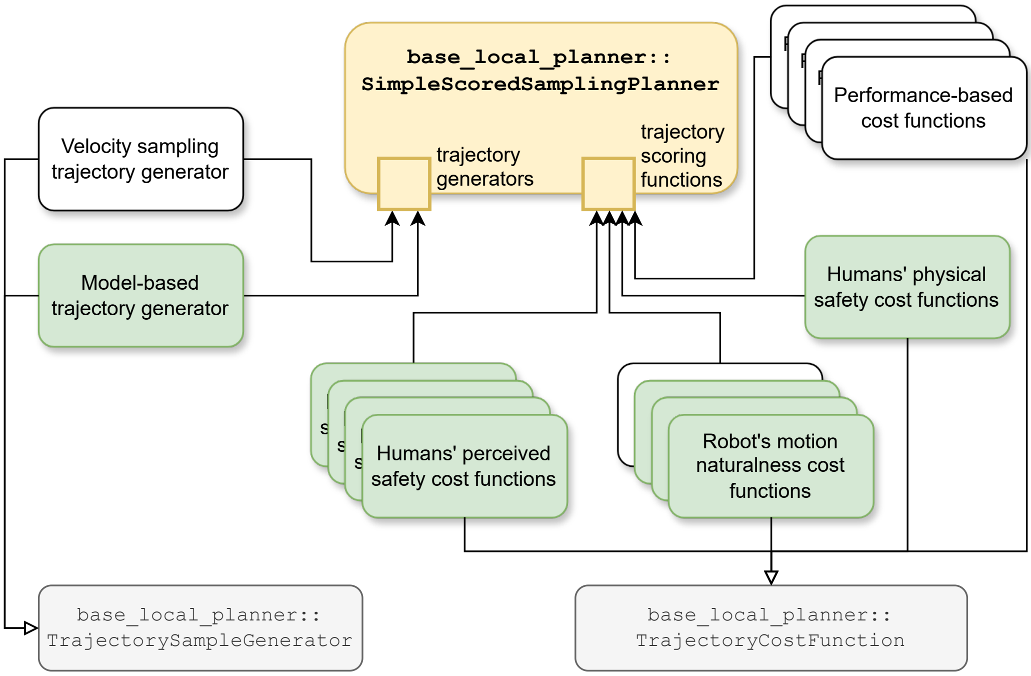


Figure 6.5: A schematic structural representation of the implemented planning framework. The elements added in the *HUMAP* are marked green, whereas white blocks indicate parts inherited from the *ROS*1 navigation ecosystem. Customised modules are highlighted yellow, and grey blocks identify type interfaces (ecosystem's base classes).

the trajectory concerning the humans' perceived safety, i.e., evaluating the robot's heading direction, ${}^{\mathbb{H}}\mathrm{cfun}_{\mathrm{dir}}$, personal zones and F-formation's O-spaces intrusions – ${}^{\mathbb{H}}\mathrm{cfun}_{\mathrm{psi}}$ and ${}^{\mathbb{G}}\mathrm{cfun}_{\mathrm{fsi}}$, accordingly, and the discomfort induced to humans by the robot's passing speed – ${}^{\mathbb{H}}\mathrm{cfun}_{\mathrm{psd}}$. Furthermore, dedicated cost functions evaluating the robot's motion naturalness are also implemented, particularly, $\mathrm{cfun}_{\mathrm{osc}}$ – penalising oscillating motions, as well as $\mathrm{cfun}_{\mathrm{vsm}}$ and $\mathrm{cfun}_{\mathrm{hsm}}$ – penalising velocity changes, linear and angular, accordingly. Complementary *ROS* cost function penalises backward motions of the robot, $\mathrm{cfun}_{\mathrm{bwd}}$. Additionally, the cost function assessing the physical safety of humans constitutes a novel implementation of the ${}^{E}\mathrm{cfun}_{\mathrm{ttc}}$.

The dedicated trajectory generator along with cost functions are implemented using the base class interfaces, which is crucial to utilise new modules with the original planning scheme (implemented in the `base_local_planner::SimpleScoredSamplingPlanner`) employing the polymorphism principle of Object-Oriented Programming.

Planner's behaviours While Fig. 6.5 shows the structural elements of the planning framework, the high-level behavioural aspects of the trajectory planner are described in Sec. 5.2.2 and constitute a dedicated implementation utilising finite state machines. The orchestration of the *HUMAP*'s states is implemented in the `PlannerState` class.

Notably, the robot's behaviour in the *Execution Initialisation* and *Orientation Adjustment* states relies on the slightly modified version of the `base_local_planner::LatchedStopRotateController` – modified class has been directly incorporated into the *HUMAP* planner's package.

Parameters Various parameters regarding planning scheme and system configuration were introduced in the *HUMAP*. However, embedding all those parameters in this thesis is impractical;[21] therefore, a snapshot of the public repository has been created[22] to easily reproduce the results of the experiments (Chapter 7).

Nonetheless, the crucial parameters influencing the *HUMAP*'s operation are (in the order of importance): the cost function weights (Sec. 5.7), number and values of pedestrian motion model's multipliers (Sec. 5.5.7), and number of trajectories generated with the method of sampling feasible robot velocities (Sec. 5.6). Identified parameters affect the form of the scoring function and the solution space, whose influence is examined in Sec. 7.5.1.

Furthermore, the fundamental parameters influencing the computational complexity of the planner are the number of generated trajectories (all trajectories are scored in each step; hence, the duration of the scoring stage is affected) and the planning/prediction horizon (in the experiments, $t_{\text{hor}} = 2$ s). The reference parameters of the *HUMAP* have been selected by experimentally achieving a compromise between the emphasised respect to social aspects of robot navigation and the task performance, as well as computational complexity.

Numerous parameters, that are less important or rarely changed, serve to enable or disable the system modules (useful in the development stage) or are related to the configuration of the planner's visualisation (for debugging purposes). Additionally, most parameters are integrated with the *ROS* `dynamic_reconfigure` tool that enables the capability to change the parameters in a runtime.

SFM* and *FIS The trajectory generator that employs the pedestrian motion model formulated as *Fuzzy-Extended Social Force Model* utilises implementations of the *Social*

[21] The *HUMAP* exposes approximately a total of 130 parameters to the user (as a comparison – *TEB* approximately 160).

[22] In the source code (the URL is placed in footmark 19), the dynamically adjustable parameters are available in the `cfg/HumapPlanner.cfg` file, whereas complementary parameters (mainly a robot's kinodynamic specification) loaded statically at the system startup, are placed in the `src/humap_config_ros.cpp`. Relevant documentation regarding planner parameters is prepared in the repository's information file.

Force Model and *Fuzzy Inference System*. The underlying *SFM* is implemented as a dedicated C++ library, whereas the *FIS* part is designed using the *fuzzylite* C++ library [419] (version 6.0). Both *SFM* and *FIS* are aggregated in the same software package along with the rest of the source code.

6.10 SRPB implementation

The proposed Social Robot Planner Benchmark (Chapter 4) has been widely used for conducting the quantitative evaluation of robot performance while executing navigation tasks for the experiments discussed in this thesis (Chapter 7). The benchmark has been published as open-source software.

The *SRPB*'s operational procedure consists of two main stages – online and offline. During the online stage (Fig. 6.6), the logging modules are instantiated and periodically update the text files stored in the filesystem. Specifically, in the *ROS*1 implementation, the logging components are aggregated into the modified `move_base` node, i.e., `srpb_move_base`,[23] which acts as the original navigation orchestrator but also collects data for further evaluation. The data sampling is performed at the frequency of the local planning scheme to avoid (possibly) inaccurate interpolation of data (if the logging would be performed at the rate of the system's module operating with the highest frequency).

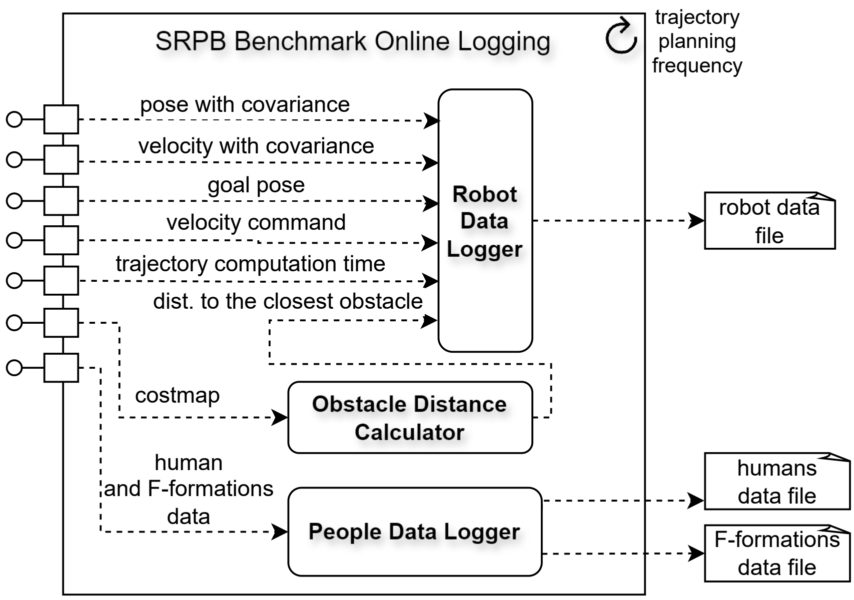


Figure 6.6: Schematic presentation of the *SRPB*'s logging scheme.

In Fig. 6.6, the *Obstacle Distance Calculator* uses bicubic interpolation to calculate the closest distance between the robot and any obstacle. The calculations are performed using

[23] `https://github.com/rayvburn/srpb_move_base`

costmap, which aggregates sensor observations gathered throughout the scenario progression. The implementation of this module is inherited from the *MRPB* benchmark [33].

In contrast, during the offline stage, the standalone programs process the content of the files updated during the previous stage and compute the metrics. A typical workflow for obtaining data with the *SRPB* benchmark from large-scale experiments conducted for this thesis is presented in Fig. 6.7. The figure shows that 100 trials need to be repeated in 3 scenarios with a certain system configuration, i.e., with a specific local trajectory planning method. As stated, data logging is performed during the online stage of the *SRPB*'s operation. Then, metric results are computed based on each collected set of logging artefacts (output files in Fig. 6.6). After that, using the provided *SRPB* tooling, the metric results are loaded in a batch and inserted into a new spreadsheet file that contains both raw metric values as well as filtered ones (medians). The spreadsheet files are generated to facilitate the diagnostics of the results, as the collected dataset might be significant. Usually, the metric batch loading is performed on results related to numerous trajectory planning algorithms.

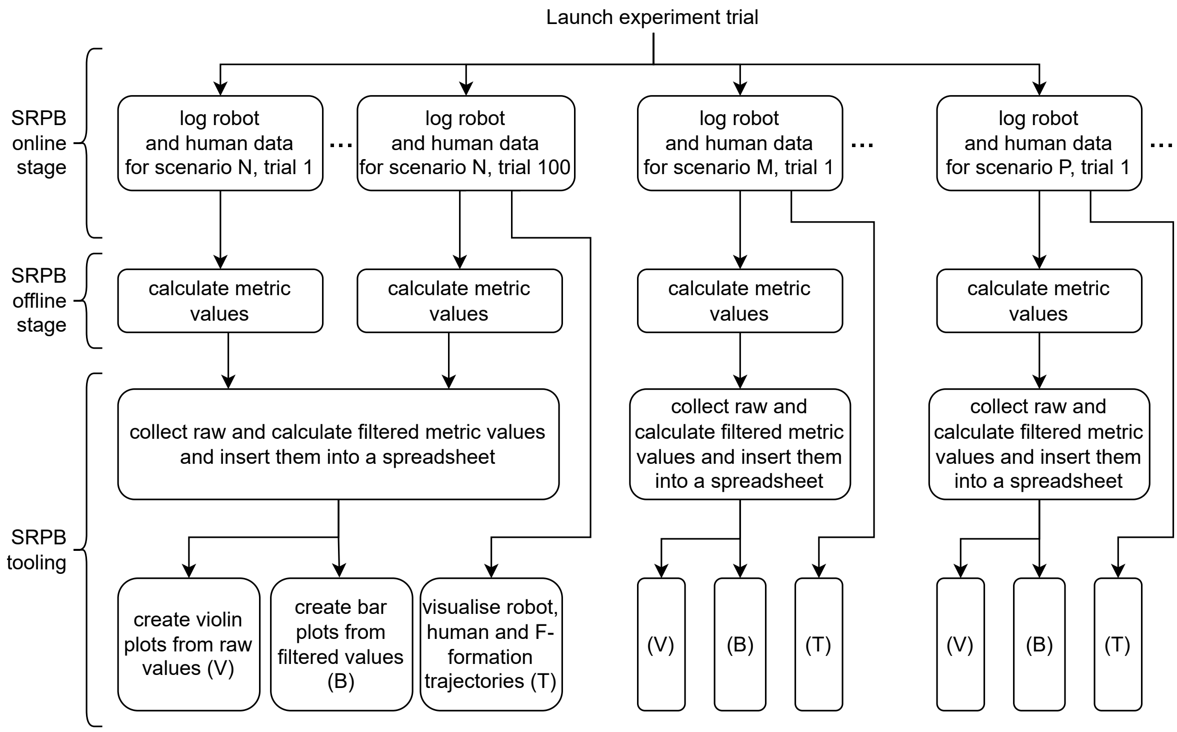


Figure 6.7: A typical workflow of the *SRPB* benchmark usage.

After collecting data in the spreadsheets, results can be used for visualisation purposes. Specifically, the tooling of the `srpb_evaluation` software package supplies the user with scripts for creating violin plots, bar plots, or visualising trajectories executed by a robot, humans or F-formations during experiments. Additionally, LaTeX tables can also be generated using the provided script. All those data representations except trajectory visualisation are created on the basis of the spreadsheets' contents. Typically, separate spreadsheets are created with data related to different scenarios.

Chapter 7

Experiments

This thesis proposes study-based metrics for the quantitative evaluation of human-aware motion planning algorithms, as well as the novel approach for the socially-aware trajectory planning method. Therefore, numerous experiments have been conducted to assess the performance of the state-of-the-art approaches focused on traditional navigation against the methods that are focused on human awareness concepts.

This chapter is divided into two major parts. The first part of the experiments aimed to determine, whether motion planning for human-aware navigation is still an open problem. It can be assessed by exploring if state-of-the-art human-aware local trajectory planners significantly outperform classical algorithms in terms of social metrics (Sec. 7.3). On the other hand, the second phase of the experiments relies on the multi-scenario comparison of the performance and social appropriateness of the proposed trajectory planning method – *HUMAP*, against various methods – classical and specialised for social robot navigation, including learning-based approaches (Sec. 7.4).

In both parts of the experiments, the same assessment methodology was implemented – the controlled studies were designed and conducted to isolate the factors that might influence the results. The second common aspect of both parts of the conducted experiments is the quantitative evaluation approach, relying on the metrics implemented in the *SRPB*. The application of the same method ensures a systematic and independent comparison, which is paramount when inspecting state-of-the-art methods against the novel algorithm. Nevertheless, each phase of the experiments differs in terms of the scenarios selected for the study.

The metrics embedded into the proposed *SRPB* benchmark approximate the human impressions directly on the basis of the examinations published in the literature. Namely, the proposed metrics implement continuous models that serve as indicators of human discomfort, conforming to the multiple social robot navigation requirements developed based on the extensive literature review (Chapter 2). In particular, the metrics assessing

Parameter	${}^{r}_{\min}v_{\text{lin}}$	${}^{r}_{\max}v_{\text{lin}}$	${}^{r}_{\min}\omega$	${}^{r}_{\max}\omega$
Value	-0.1 $\frac{\text{m}}{\text{s}}$	0.5 $\frac{\text{m}}{\text{s}}$	-1.05 $\frac{\text{rad}}{\text{s}}$	1.05 $\frac{\text{rad}}{\text{s}}$
Parameter	${}^{r}_{\min}a_{\text{lin}}$	${}^{r}_{\max}a_{\text{lin}}$	${}^{r}_{\min}\alpha$	${}^{r}_{\max}\alpha$
Value	-1.0 $\frac{\text{m}}{\text{s}^2}$	1.0 $\frac{\text{m}}{\text{s}^2}$	-1.05 $\frac{\text{rad}}{\text{s}^2}$	1.05 $\frac{\text{rad}}{\text{s}^2}$

Table 7.1: Trajectory planners' parameters that were constant throughout the experiments.

the robot's motion naturalness (Sec. 2.4) and the impact of the robot's trajectory on the perceived safety of humans (Sec. 2.3) are the main indicators in the evaluation of the social acceptance of a certain algorithm. Taking into account the above premises, primarily that the *SRPB* metrics were designed based on results from different studies, we did not attempt to revalidate the human impressions of interacting with the robot, as it has already been a broad topic of numerous user studies and surveys, which we take advantage of.

The majority of results from the first phase of the experiments (Sec. 7.3) have been included in our previous works [20, 21], while elements of the second phase of the experiments (Sec. 7.4) have been encompassed in the conference paper [24].

7.1 Hardware setup

Real-world experiments were conducted with PAL's TIAGo Iron robot, and simulation results were obtained with the robot's digital twin provided by PAL. Although the robotic platform is under constant development by the laboratory team,[1] during the experiments only its factory equipment has been used. The main sensors of the robot are: a Sick TIM571 *LiDAR* (0.05 – 25 m scan range, 180° field of view, 0.33° step angle) and an Orbbec Astra *RGB-D* camera (depth stream with a resolution of 640 x 480 pixels and a 0.6 – 8 m depth sensor range). Parameters related to kinematic and dynamic constraints of the mobile base, shown in Tab. 7.1, were common for all examined trajectory planners.

The factory control interface of the mobile base is implemented with *ROS* and follows the standards, expecting velocity commands to be sent at least each 500 ms. Once the new command is not obtained in time, the safety layer of the robot control system stops the mobile base until the new velocity command is received. Therefore, only trajectory planners capable of real-time operation were suitable for real-world tests.

Since the target robot has factory-installed *ROS* Melodic, we performed simulated trials with the same framework version. We have chosen Gazebo (version 9) as the simu-

[1] `https://www.robotyka.ia.pw.edu.pl/`

lation platform due to its integration with *ROS*. Simulation experiments were performed on a laptop with an Intel Core i7-4720HQ CPU and 16 GB RAM.

In our experiments, social metrics were computed based on data gathered by the robot's onboard sensors during the run to a goal pose. That approach is appropriate for rapid prototyping and often sufficient to obtain representative results; however, still prone to poor performance of the limited-range robot sensors, e.g., *RGB-D* cameras. Thus, integrating a robot with an external, e.g., vision-based system, can increase the evaluation robustness, decreasing metric deviations between subsequent trials. We argue that external systems for human tracking can be used for benchmarking once the robot control system is integrated with them. Otherwise, planners may be penalised for actions disregarding surrounding humans that the planners are unaware of. Nevertheless, both the *SRPB* benchmark, as well as *HUMAP* trajectory planner, can be interfaced with any source of aggregated information about humans surrounding the robot.

7.2 Experiments design methodology

The experiments described in this thesis aim to inspect the state-of-the-art social robot navigation methods and compare them in different scenarios. The previous approaches are evaluated against the novel *HUMAP* planner; hence, selecting a proper experiment design methodology and applying it systematically ensures that results are accurate and unbiased, and allows viable comparisons of outcomes obtained in different scenarios.

The experiments described in this chapter were conducted as controlled studies. This type of assessment has been selected to isolate the factors that might appear during the experiments and might influence the results. Specifically, in human-aware robot navigation experiments, the crucial factor is the unintended presence of humans (or other dynamic agents) who were not supposed to participate in scenarios. Detecting them by the robot causes unintentional modification of its environment model. While it is straightforward and easy to achieve in the simulation experiments, it might not always be feasible in real-world trials. Therefore, the experiments were conducted during off-peak hours and on weekends.

This chapter discusses the experiments performed in two phases, each differs in terms of the scenarios selected for the study. In contrast, a common aspect of both parts of the conducted experiments (after the applied methodology) is the quantitative evaluation approach, relying on the metrics implemented in the *SRPB* which were computed based on data gathered during the real-world and simulation trials. The *SRPB*'s parameters used for the evaluation were static and are shown in Tab. 7.2.

Although several tests were conducted to verify the *HUMAP*'s performance across

Parameter	${}^{r,\mathbb{O}}d_{\min}$	${}^{r}_{x}v_{\text{osc}}$	${}^{r}_{y}v_{\text{osc}}$	${}^{r}_{\text{lin}}v_{\text{osc}}$
Value	0.55 m	0.025 $\frac{\text{m}}{\text{s}}$	0.025 $\frac{\text{m}}{\text{s}}$	0.025 $\frac{\text{m}}{\text{s}}$
Parameter	${}^{r}\omega_{\text{osc}}$	φ_{fov}	d_{ocp}	d_{cr}
Value	0.05 $\frac{\text{rad}}{\text{s}}$	3.3 rad	0.28 m	0.275 m

Table 7.2: Configurable parameters of metrics that were used in the experiments.

its various configurations and scenarios (Sec. 7.5), the majority of results were obtained from extensive comparisons of different local trajectory planners operating under the same environmental conditions but in various scenarios. The goal was to design scenarios that enabled the robot to reach a goal pose operating with each examined trajectory planner, navigating collision-free from the shared start pose. Thus, we started with evaluating the capabilities of various planners integrated into the robot's control strategy while the robot operated in partially unknown,[2] static or dynamic environments. While selecting the scenario configuration admissible for all trajectory planners was possible during the first phase of the experiments (Sec. 7.3), it was not viable in the second phase (Sec. 7.4), in which more planners were tested. This topic is detailed in relevant sections of the chapter.

The test environment for the real-world experiments was a robotics laboratory at Warsaw University of Technology (Fig. 7.3b, 7.4b, 7.7b, 7.8b, and 7.9b) and it was only minimally prepared for the study, remaining cluttered, which poses an additional challenge for the navigating robot. One of the main goals was to compare real results with outcomes obtained from the simulation; therefore, analogous scenarios have been performed in a virtual equivalent of the environment.[3]

During the comparative experiments, the environment configuration in the following trials had to be replicated to evaluate different trajectory planners under the same conditions (fundamental principle of a controlled study). In real-world tests, ensuring that human participants move similarly in each trial is a challenging task, particularly in dynamic scenarios. Hence, to maximise the path similarity, the entire paths (not only the starting and ending points) of dynamic actors were indicated with a tape glued to the floor. Additionally, for trajectory similarity, paths were equipped with subsequent pose markers (Fig. 7.4b, 7.7b, 7.8b, 7.9b) and the participants were asked to finish each step with a tick of a metronome that was programmed to 60 beats per minute.

Conducting experiments with simulated humans poses another substantial challenge. The behaviour of virtual dynamic agents, namely their movements, can be scripted or

[2] "Partially unknown" in this context means that despite the provided map of the environment, some differences in the operating area might be present, but those were common to all investigated planners.

[3] `https://github.com/rayvburn/tiago_sim_integration`

based on planning. Scripting human trajectories provides that they will behave exactly the same in each simulation trial. However, experiments rely on testing different trajectory planning methods, that directly affect where the robot is located in subsequent time steps. Therefore, when the scripted movements method is applied, virtual humans will not interact in the same way with a planner that needs 30 seconds to reach the goal, as the planner that requires 60 seconds. Hence, as stated in Sec. 6.8, virtual dynamic agents utilise motion planning modules for online replanning, As a consequence, the simulated humans might not necessarily exactly reproduce their trajectories in subsequent trials, but rather naturally interact with the environment in an unfocused manner. Human operation in all scenarios is defined by fixed waypoints in key places (near corners or walls) and such an approach is still suitable for the controlled studies.

A fundamental necessity of conducting social robot navigation experiments is providing the robot's capability to detect humans. As denoted in Sec. 6.7, the robot's raw sensor data are used to detect and track people in the environment. This is contrary to a more common approach of using the perfect data about humans, e.g., [168, 156], that is provided by a simulator; however, using the same perception methods in the simulation and in the real-world experiments (where ground truth data are not available) is crucial to allow direct comparisons of the results obtained in different domains (simulation and real world).

While for real-world experiments only the most reliable planners were designated, the simulation tests were conducted using various trajectory planning methods that performed differently in the designed scenarios. A practical assumption of the experiments' conduction is that a timeout value of 120 s has been set for the execution of each simulated trial. Nonetheless, most of the planners that consequently completed the navigation task were able to finish designed runs within 60 s.

7.3 Evaluation of the state-of-the-art trajectory planners

The experiments described in this section intend to determine whether state-of-the-art human-aware trajectory planners perform superior to traditional ones regarding the mitigation of discomfort among humans in the robot's environment. To gather insights on this topic, different robot navigation methods were validated utilising the proposed *SRPB* benchmark (Sec. 4.1) for the quantitative evaluation. The assessment criteria of robot operation involve the metrics regarding the robot's motion naturalness and the perceived safety of humans that are influenced by the movements of the robot. Additionally, the overall robot navigation performance was examined and the qualitative assessment of each planner's characteristics has been outlined.

7.3.1 Scenarios description

We conducted experiments in which humans participate as static or dynamic elements of the robot environment; therefore, tests are identified as *static* in a simulation (Fig. 7.3a), *static* in the real world (Fig. 7.3b), *dynamic* in a simulation (Fig. 7.4a), and *dynamic* in the real world (Fig. 7.4b). These scenarios are later referred to as *1-S*, *1-R*, *2-S*, and *2-R*, accordingly.

In the *static* scenario (Fig. 7.3), an F-formation of two humans stays near the robot's goal. Reaching the goal by the robot requires passing the humans, so when approaching the final pose, the robot is foreseen not to distract the humans involved in a focused interaction and take an outside path. Instead, in the *dynamic* scenario (Fig. 7.4), the robot moving to the goal pose along a narrow corridor encounters a moving human followed by another moving human, both going opposite to the robot.

The sequence diagrams illustrating the scenario progression are shown in Fig. 7.1 and 7.2 for *static* and *dynamic* scenarios, accordingly. The diagrams describe the task interaction between the robot and humans, which applies to both simulation and real-world experiments.

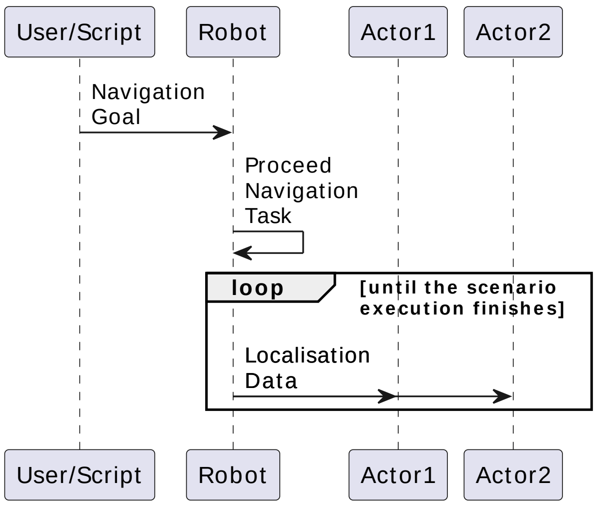


Figure 7.1: The sequence diagram of the robot and humans' activity during the *static* scenario. Humans are stationary in this scenario.

In both scenarios, the robot interacts with humans in an unfocused way; therefore, the robot is expected to avoid collisions and maximise the perceived safety of humans.[4] Each evaluated trial involved one-shot navigation ("PoseGoal" in the nomenclature from [420]) from a start pose to a goal pose – both fixed but scenario-specific.

[4]A video presenting test scenarios is available at `https://vimeo.com/805337193`.

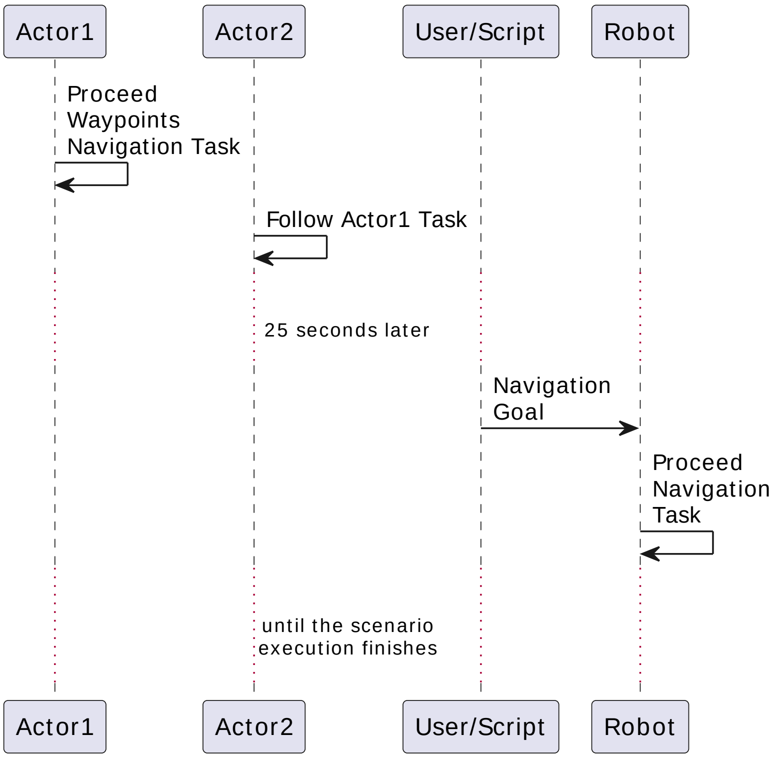


Figure 7.2: The sequence diagram of the robot and humans' activity during the *dynamic* scenario. In this scenario, one actor explicitly executes the *following* task, trying to maintain 1.5 m distance from the other.

7.3.2 Evaluation principles

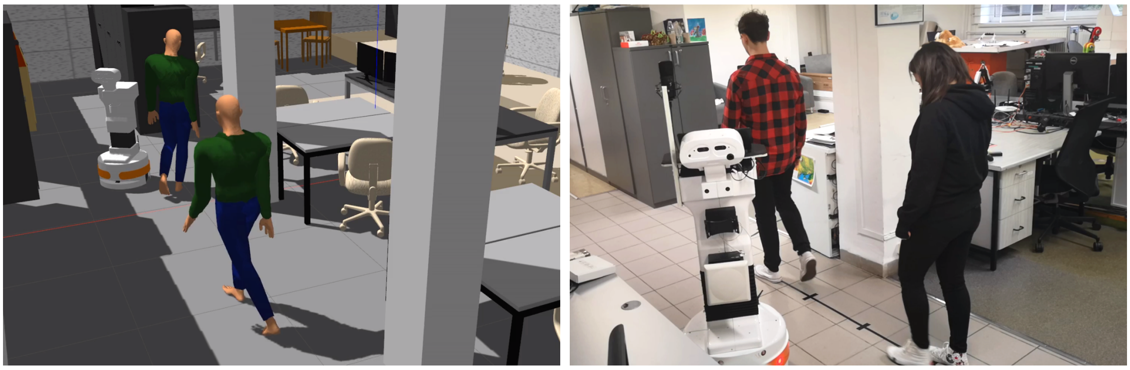
Multiple planning approaches were integrated with the TIAGo robot and their operation under the same environmental conditions have been evaluated. In each scenario, traditional trajectory planners for mobile robots were tested, namely: *Elastic Bands* [310], *DWA* [144], *Trajectory Rollout* [161, 147], *TEB* [162], as well as human-aware trajectory planners: *Human-aware TEB* (*HaTEB*) [216] and *Co-operative Human-Aware Navigation* (*CoHAN*) [168]. The public *ROS* implementations of the evaluated algorithms were utilised. The remainder of the navigation ecosystem's configuration (global path planner, localisation algorithm) is described in Sec. 6.3, whereas the framework for simulating humans in virtual trials is explained in Sec. 6.8. The distinctive feature of this study is that all trajectory planners (also the traditional ones) utilised costmap environment representations with human proxemics modelled as spatial costs around detected individuals [47, 84], without regarding F-formations.[5]

During the study, each trajectory planner operated with the maximum possible frequency which ensured that the designated trajectory planning period was not surpassed,

[5]The F-formation costmap layer was not developed at the time of the first phase of the experiments. The discussion on extensions of the navigation ecosystem is in Sec. 6.5.

(a) Simulation (b) Real world

Figure 7.3: An overview of the *static* scenario.



(a) Simulation (b) Real world

Figure 7.4: An overview of the *dynamic* scenario.

but the frequencies did not exceed 10 Hz. Specifically, the *Elastic Bands* operated at 10 Hz, *DWA* at 4 Hz, *Trajectory Rollout* at 8 Hz, and *TEB*, *HaTEB*, *CoHAN* – at 10 Hz.

Each trajectory planner was tested in each scenario's simulated and real-world variants. At least five representative trials were benchmarked for each case, and then, the median of each metric was computed to score a trajectory planner. The results of our experiments, shown in Tab. 7.3, are discussed in the following sections. Examples of trajectories performed by each planner are shown in Fig. 7.5 (*static*) and Fig. 7.6 (*dynamic* scenario).

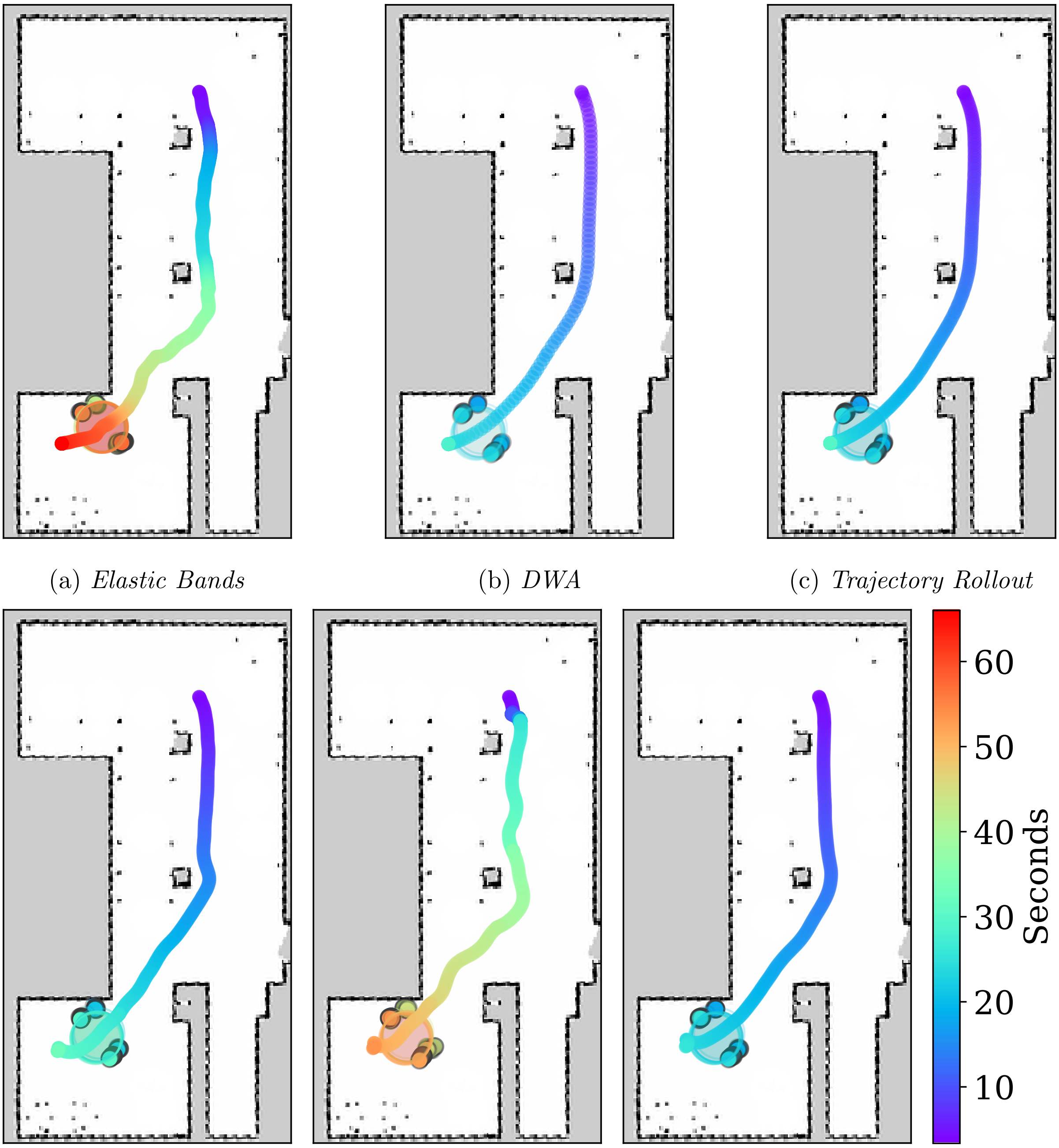
Metric	Scenario	*Elastic Bands*	*DWA*	*Trajectory Rollout*	*TEB*	*HaTEB*	*CoHAN*
m_{obs} [%]	*1-S*	58.51	33.98	39.15	36.10	**22.83**	44.17
	1-R	50.44	42.25	43.39	36.77	**21.88**	41.53
	2-S	24.03	35.44	28.13	28.39	**19.16**	53.55
	2-R	23.46	52.93	71.65	38.67	**19.81**	46.02
m_{mef} [s]	*1-S*	73.30	**25.50**	26.50	29.75	55.70	27.90
	1-R	85.84	**28.63**	29.00	31.15	38.70	39.70
	2-S	52.80	29.50	**28.00**	32.90	38.80	55.10
	2-R	59.19	**37.70**	47.50	38.70	57.50	40.10
m_{plin} [m]	*1-S*	10.27	9.80	**9.78**	10.18	10.90	10.00
	1-R	11.78	**10.59**	10.65	11.14	12.37	12.30
	2-S	12.90	11.98	**11.66**	12.03	13.10	15.71
	2-R	13.96	13.08	**11.93**	12.78	15.09	12.63
m_{chc} [rad]	*1-S*	32.62	**10.68**	10.72	14.09	28.35	12.09
	1-R	50.61	11.55	**8.11**	12.84	25.10	12.37
	2-S	38.11	4.55	**3.91**	15.14	30.77	13.14
	2-R	66.96	**7.11**	35.48	16.68	55.31	13.54
m_{cef} $[10^{-3} \cdot \text{s}]$	*1-S*	3.33	66.02	33.45	**2.50**	4.42	3.80
	1-R	5.12	160.15	78.88	**4.99**	10.30	6.61
	2-S	**1.92**	65.23	36.13	2.20	3.31	2.63
	2-R	**2.42**	123.13	64.31	4.54	8.80	7.88
m_{cre} $[10^{-3} \cdot \text{s}]$	*1-S*	3.89	34.57	9.99	**2.35**	3.44	3.27
	1-R	3.91	81.93	15.95	**2.05**	4.42	4.29
	2-S	2.47	28.44	11.56	**2.21**	3.28	3.88
	2-R	2.53	75.80	28.46	**1.64**	3.89	7.15
m_{vsm} $\left[\frac{\text{m}}{\text{s}^2}\right]$	*1-S*	0.07	0.05	**0.04**	0.11	0.32	0.14
	1-R	0.22	**0.11**	0.12	0.14	0.39	0.22
	2-S	0.15	0.09	**0.05**	0.12	0.34	0.16
	2-R	0.28	0.18	**0.17**	0.19	0.35	0.25
m_{hsm} $\left[\frac{\text{rad}}{\text{s}^2}\right]$	*1-S*	1.44	**0.18**	0.21	0.93	0.89	0.85
	1-R	1.54	**0.22**	0.37	0.86	1.16	0.78
	2-S	1.44	**0.18**	0.19	1.10	0.92	0.56
	2-R	1.68	**0.33**	0.57	0.96	1.01	0.78
m_{osc} [%]	*1-S*	3.79	1.95	**0.95**	1.18	3.67	1.69
	1-R	9.70	3.89	2.62	**1.30**	3.27	2.12
	2-S	**0.73**	1.78	0.90	0.91	2.11	7.05
	2-R	6.01	**2.15**	2.15	4.57	5.68	2.50
m_{bwd} [%]	*1-S*	0.00	0.00	0.00	1.50	3.44	6.04
	1-R	0.00	0.00	0.00	1.58	5.35	7.79
	2-S	0.00	0.00	0.00	0.00	1.74	0.00
	2-R	**0.00**	10.82	0.26	0.26	2.92	2.50

Metric \ Scenario \ Method		*Elastic Bands*	*DWA*	*Trajectory Rollout*	*TEB*	*HaTEB*	*CoHAN*
m_{iprot} [%]	*1-S*	5.18	13.72	13.69	**0.00**	1.63	0.40
	1-R	5.74	9.79	10.19	0.94	1.48	**0.12**
	2-S	6.41	3.51	3.63	0.30	0.31	**0.00**
	2-R	4.07	0.40	24.06	0.28	3.24	**0.00**
m_{psi} [%]	*1-S*	22.35	15.09	16.47	19.32	**14.30**	19.78
	1-R	**15.88**	20.94	31.37	17.88	24.29	24.88
	2-S	**20.24**	31.53	38.07	40.70	35.35	40.44
	2-R	**19.88**	42.60	30.00	25.40	34.48	39.74
m_{fsi} [%]	*1-S*	38.29	42.07	53.25	**32.11**	36.60	45.32
	1-R	**3.33**	31.85	41.05	13.30	8.25	35.73
	2-S	0.00	0.00	0.00	0.00	0.00	0.00
	2-R	0.00	0.00	0.00	0.00	0.00	0.00
m_{dir} [%]	*1-S*	**0.29**	0.76	1.04	1.31	0.77	0.87
	1-R	**0.23**	2.38	0.69	0.46	1.09	0.62
	2-S	**0.22**	1.32	0.39	0.56	1.34	0.55
	2-R	**0.31**	2.13	0.74	0.62	0.97	1.46

Table 7.3: Results of simulation and real-world experiments conducted in the WUT laboratory environments. Tests are identified as: *1-S* – *static* scenario in a simulation, *1-R* – *static* scenario in the real-world, *2-S* – *dynamic* scenario in a simulation, and *2-R* – *dynamic* scenario in the real-world.

7.3.3 Robot navigation task performance

The *HaTEB* planner was the safest in all scenarios, as it maintained the greatest distances from obstacles (m_{obs}, **Req. 1.1**), which is related to its characteristic of taking wide paths at corners (similar to going along the centre line of the available space). However, it came at the cost of the time required to reach the goal (m_{mef}). By contrast, the robot operating with *DWA* planner traversed closer to obstacles, i.e, the robot has spent a higher percentage of time within the ${}^{r,\mathbb{O}}d_{\text{min}}$ distance from obstacles along the path – reflected by the m_{obs}. Still, it reached the goal significantly faster (**Req. 1.3**). The reason behind such timing performance is that weights of cost functions in the objectives of *DWA* and *Trajectory Rollout* planners were tuned with a focus on approaching the goal with the shortest possible path avoiding high-cost areas (obstacles or humans embedded into the costmaps, detailed in Sec. 6.3) along the way. The best timing results (m_{mef}) are confirmed by the values of path-related metrics, m_{plin} (Euclidean length of the path) and m_{chc}



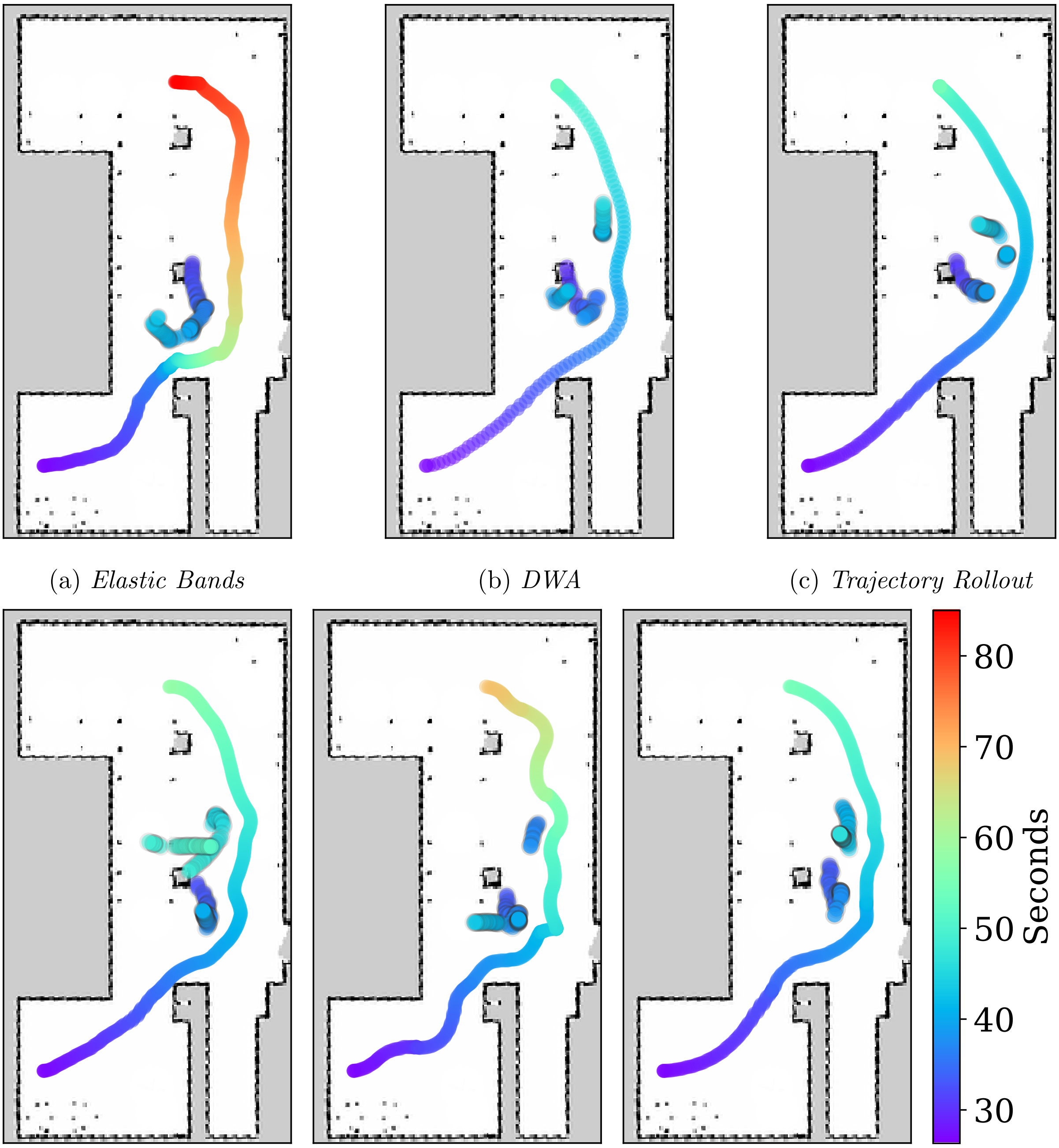
(a) *Elastic Bands* (b) *DWA* (c) *Trajectory Rollout*

(d) *TEB* (e) *HaTEB* (f) *CoHAN* (g) Time

Figure 7.5: Robot trajectories generated by different planners in the *static* scenario. The colour of a symbol represents its occurrence in time (g). Solid circles with dark edges represent humans, whereas partially transparent circles indicate F-formations. Due to the perception inaccuracy, human positions float over time, especially, after robot rotation at the very end of the scenario.

(cumulative rotations along the path), which *DWA* and *Trajectory Rollout* planners have the lowest.

The results of the computation time metrics, namely, m_{cef} and m_{cre}, must be ana-



(a) *Elastic Bands* (b) *DWA* (c) *Trajectory Rollout*

(d) *TEB* (e) *HaTEB* (f) *CoHAN* (g) Time

Figure 7.6: Robot trajectories generated by different planners in the *dynamic* scenario. The colour of a symbol represents its occurrence in time (g). Solid circles with dark edges represent humans. Due to the robot's perception inaccuracies, human positions may fluctuate over time, especially when humans become occluded after being passed by the robot.

lysed, remembering that the simulated scenarios were performed on a different computer than real-world experiments. Nevertheless, data show that optimal velocity search methods (*DWA* and *Trajectory Rollout*) exhibit a higher computational burden (m_{cef}) than force-based (*Elastic Bands*) and graph optimisation-based (*TEB*, *HaTEB*, *CoHAN*) approaches. These latter, mainly *TEB*, have much more stable computation times in different scenarios (m_{cre}).

An interesting observation concerns the values of m_{cef} and m_{cre} metrics between *TEB* and its human-aware variants. Namely, *HaTEB* and *CoHAN* have longer computation times due to the increased number of constraints captured by the optimisation objective regarding humans.

7.3.4 Robot motion naturalness

Optimal velocity searching planners provide smoother trajectories (smallest m_{vsm} and m_{hsm}), increasing motion naturalness of the robot (**Req. 2.3**). In terms of oscillations (m_{osc}) and backward movements (m_{bwd}), the traditional trajectory planners performed the best in most cases, avoiding unnatural motions. However, in the real-world *dynamic* scenario (*2-R*), *DWA* planner occasionally performed recoveries moving backwards (Fig. 7.6b presents more representative trial from a simulation). As shown in Tab. 7.1, we allowed planners to command the mobile base backwards to verify how they will behave against an unexpected human agent.

As for in-place rotations (m_{iprot} metric), *TEB* and derived planners – *HaTEB* and *CoHAN*, generally outperform others due to the feature of this class of planners that adjust the final part of the trajectory to reach the goal position and orientation simultaneously. Instead, *DWA* and *Trajectory Rollout* rotate the mobile base according to the goal orientation after reaching the goal position. Another situation when the robot executes in-place rotations is when it encounters a dynamic obstacle along the path. This issue may be addressed with human trajectory prediction that has been employed in human-aware planners – *HaTEB* and *CoHAN*.

7.3.5 Perceived safety of humans

Our human-perceived safety metrics (**Req. 2.2**) recreating study-based human preferences [64] are novelties amid robot planning benchmarks. We evaluated the planners against personal space intrusion (m_{psi}), F-formation space intrusion (specifically, O-spaces, m_{fsi}), and robot heading direction relative to the human's centre (m_{dir}).

Surprising results are related to m_{psi} metric, where in half of the test cases the traditional *TEB* planner outperformed its human-aware specialisations – *HaTEB* and *CoHAN*. Only in the simulated *static* scenario (*1-S*), *HaTEB* performed better than the traditional *DWA* planner only by 1 p.p. In 3 out of 4 scenarios, the least personal space intrusions have been noted with the *Elastic Bands*, but at the cost of overall performance.

The *static* scenario was designed to provide insights into whether any trajectory planner would favour taking a longer path to avoid crossing the F-formation. Although none of the planners explicitly accommodate human formations, we expected that in the *static* scenario (*1-S* and *1-R*), the robot's behaviour will emerge to F-formation avoidance due to respecting personal spaces of single humans. It did not happen with any planner, as the robot has crossed through the O-space of the F-formation in each trial (Fig. 7.3b). The phenomenon is reflected in m_{fsi} metric, remembering from (4.29) that the robot escaping from the O-space sooner obtains a smaller metric value. In the *static* scenarios, the human-aware *CoHAN* planner stopped and often oscillated when crossing through the O-space of the F-formation (therefore increased m_{fsi} values). In the *dynamic* scenarios (*2-S* and *2-R*), the robot's perception did not qualify moving humans as an F-formation (as expected); thus values of m_{fsi} are 0.0.

The metric representing human disturbance induced by a robot heading direction (m_{dir}) is useful for evaluating whether a planner is capable of adjusting the trajectory heading towards a human as soon as it detects such an agent. Exhibiting such intent-expressive behaviour is often identified as motion legibility [23]. Again, human-aware trajectory planners performed similarly to traditional ones across all scenarios. Here, *Elastic Bands* accomplished the best metric scores, which are caused by its frequent heading changes (m_{chc}).

In our tests, the examined state-of-the-art human-aware trajectory planners do not significantly improve robot navigation regarding social metrics over the traditional approaches that treat humans as generic obstacles.

7.3.6 Robustness

The results presented in Tab. 7.3 are based on successful trials of the robot navigating from the initial pose to the goal pose. However, we performed multiple test runs beyond the benchmarked trials to find the start and goal poses accessible for all planners (Sec. 7.2). During these tests, we could observe the robustness of each planner. The *TEB* planner outperforms others with 0% of failed runs, whereas *HaTEB* commonly aborted further navigation being stuck, e.g., before an F-formation in the *static* scenario. By contrast, although the *Elastic Bands* planner appeared to be as robust as *TEB*, it generates multiple erratic motions (reflected by m_{chc} and m_{hsm} metrics), these, in turn, cause much longer times needed to reach the goal (m_{mef} metric), making *Elastic Bands* impractical.

The experiments were conducted with a differential drive robot (Sec. 7.1); hence, in-place rotations, reflected by the m_{iprot} metric, were sometimes necessary.

7.3.7 Characteristics of trajectory planners

The analysis of the combination of quantitative (Tab. 7.3) and qualitative experiment results (Fig. 7.5 and 7.6) provide insights on the characteristics of each investigated planner.

Elastic Bands While the robustness of this force-based planner is at the level of *TEB*, its practicality is limited due to numerous rotational movements along the path and slow progressive movements towards the goal, which affects the time required to finish navigation tasks.

DWA* and *Trajectory Rollout Both planners explore the space of feasible velocities from which curved trajectories are created and select the optimal candidate according to the objective function. The difference is that *DWA* maintains the admissible velocity selected at the first planning step throughout the whole planning horizon, whereas *Trajectory Rollout* repeats the search considering feasible controls (**Req. 1.2**) in each subsequent step.

However, the dynamics of obstacles are not explicitly considered in the robot trajectory planning, which, in turn, might lead to late trajectory adjustment in dynamic scenarios, especially when the robot moves at high speeds and when dynamic obstacles are not captured adequately early in the robot's environment model.

TEB*, *CoHAN* and *HaTEB *TEB*-derived optimisation-based trajectory planners consider the goal pose's orientation information in the planning procedure. Such an approach provides a simultaneous transition of the robot's position and orientation when approaching the goal pose; hence, the robot rarely needs to rotate in place (**Req. 2.3.1.3**) once the goal position is reached.

The human-aware specialisations of the traditional *TEB* planners extend the set of social constraints regarded during the optimisation. Namely, the *HaTEB* considers minimum safety distance, time to collision, and directional constraints, including the predicted human trajectories in the problem formulation. Furthermore, *CoHAN* extends *HaTEB* with motion legibility improvements. Although the planners use human motion predictions with a constant velocity model, during the experiments both planners exhibited decreased velocity smoothness in proximity to humans (both in real-world and in simulation). It indicates that the formulation of human proximity constraints might need to be reformulated in the optimisation framework. Additionally, the visualisation of trajectories planned by the *HaTEB* presents that the planner's outcome sometimes passes through untraversable regions.

The robot operating with *HaTEB* traverses mainly through the centre of available space between obstacles, undertaking the safest path available. The feature may be relevant for robots traversing along corridors with blind corners, as it favours the exploration of the robot's side spaces. In contrast, the *CoHAN* has been found sensitive to optimisation weights but generally generates smoother trajectories near humans compared to *HaTEB*.

7.3.8 Summary

We have found that comparing different planners in a simulation generally allows finding the one that will also perform best in a similar real-world scenario regarding a particular metric. We observed the robot operating with different planners during our simulation and real-world experiments, and some distinctive behaviours of certain planners are visible in both cases. These include, e.g., wide turns of *HaTEB*, mostly straight trajectories of *DWA*, and smooth stopping of *Trajectory Rollout*.

The results of the *static* scenario indicate that considering the F-formations explicitly is necessary. Since the navigation ecosystem configuration, which treated humans as single entities, did not enable handling social cues properly, before the second phase of experiments the novel F-formation costmap layer plugin was developed to utilise extended environment model representation on both path planning and trajectory planning levels (Sec. 6.5).

The outcomes of the conducted experiments have to be studied holistically, as social robot navigation is usually posed as a problem of contrary criteria. Namely, socially navigating robots might naturally lose performance in favour of adapting their behaviour for seamless interaction with surrounding humans. This is particularly visible in the case of *Elastic Bands*, which controlled the mobile base so the personal spaces were least intruded; however, due to the slow progress toward the goal this planner is rather impractical by today's standards.

Our quantitative findings indicate that planners treating humans as typical obstacles obtained comparable or better results in terms of the personal space intrusion metric (m_{psi}), which was also observed qualitatively during the study. Overall, the first part of the study provided the baseline assessment of the current state of the research in the field and the practical aspects of modern methods.

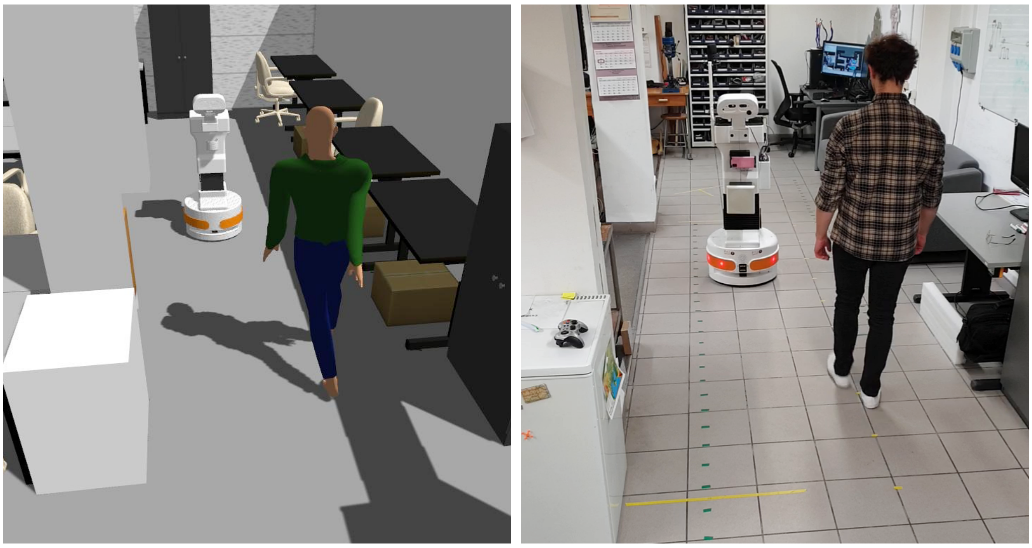
7.4 Comparative study of the proposed trajectory planning method

The conclusions drawn from the first phase of the study confirmed the relevance of the development of a custom algorithm for human-aware robot navigation. Therefore, the second stage of the experiments was designed and conducted to validate the performance of the proposed planner (Chapter 5) in various scenarios within demanding real-world and simulation environments including the presence of humans.

Numerous tests were carried out with state-of-the-art methods to evaluate the efficacy of the developed *HUMAP* algorithm compared to other approaches. Assessment criteria were divided into the robot navigation performance, robot motion naturalness, and the perceived safety of humans while the robot is moving. Quantitative evaluation of real-world and simulation trials was performed using the *SRPB* benchmark that assesses most aspects considered in social navigation studies.

7.4.1 Scenarios description

We designed test scenarios to examine the robot's capabilities of passing the humans that approach in narrow corridors (Fig. 7.7), overtaking the slowly moving people (Fig. 7.8), and yielding a way to a human that crosses the robot's path (Fig. 7.9). These scenarios are commonly referenced as standards in human-aware navigation evaluation [10, 11], while reflecting the real-world challenges for assistive robots operating in, e.g., hospitals or restaurants.



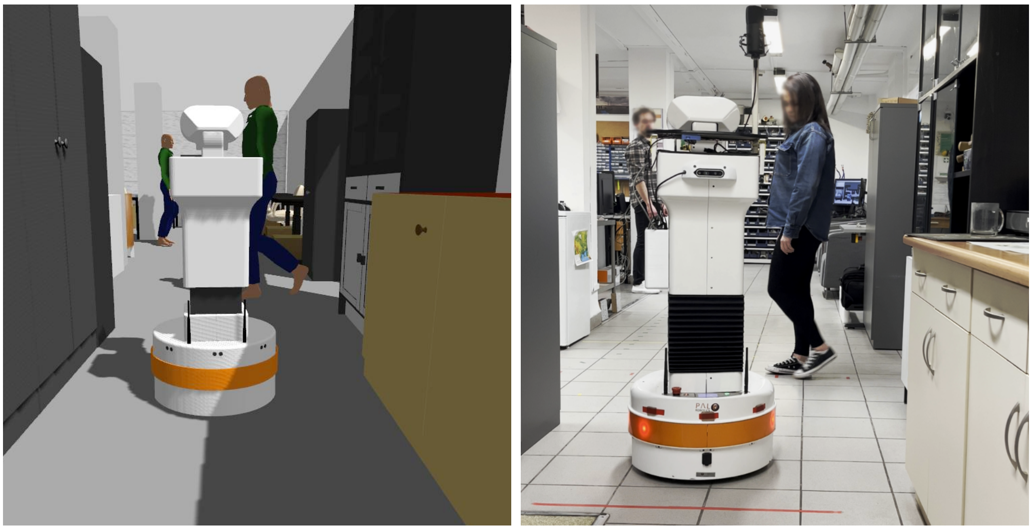
(a) Simulation (b) Real world

Figure 7.7: An overview of the *passing* scenario used for the local trajectory planners' comparison.

In all experiments of our controlled study, humans participate as dynamic objects of the robot environment, but the type of encounters and the domain of the experiment differ. Therefore, the following tests are identified as *passing* in a simulation (Fig. 7.7a), *passing* in the real world (Fig. 7.7b), *overtaking* in a simulation (Fig. 7.8a), *overtaking* in

(a) Simulation (b) Real world

Figure 7.8: An overview of the *overtaking* scenario used for the local trajectory planners' comparison.



(a) Simulation (b) Real world

Figure 7.9: An overview of the *crossing* scenario used for the local trajectory planners' comparison.

the real world (Fig. 7.8b), *crossing* in a simulation (Fig. 7.9a), and *crossing* in the real world (Fig. 7.9b).[6]

In the *passing* scenario, the robot begins its movement toward the goal in a narrow aisle, and after a few metres encounters the first human walking in the opposite direction (Fig. 7.7b). The reference path to the goal contains an S-turn to avoid obstacles. The

[6] A video presenting test scenarios with the robot operating under the *HUMAP* local trajectory planner is available at `https://vimeo.com/934800693`.

second human waits in the S-turn area and when the robot approaches the turn entry, the human starts moving in the direction opposite to the robot. Therefore, the robot must effectively perform two safe passes in the *passing* scenario, where the first human is observable right from the start, but the second is detected at a closer distance and the area for passing is straitened. Simultaneously, the robot should mitigate the discomfort induced in humans.

On the other hand, in the *overtaking* trials the robot begins the run from the opposite corner of the laboratory. When the robot approaches the S-turn (from the opposite side compared to the *passing* scenario), it is intended to overtake the first human moving slowly on the left side of the aisle. The speed of a human was fixed at 0.15 m/s, while the robot's maximum speed was limited to 0.5 m/s during the experiments. Efficient overtaking requires motion prediction of a dynamic object and undertaking a longer path to account for space needed by a human to take further steps. However, once the robot passes the S-turn and starts moving further along the aisle (straight toward the goal) another human appears, occupying the right side of the aisle and moving even slower (0.1 m/s) than the first one. Therefore, the robot finds itself in a situation, where the overtaken human is slightly behind the robot's left side and the second human is in front on the right (Fig. 7.8b). The substantial challenge is that the spatial gap to overtake the slower human decreases due to the speed difference between both humans. The *overtaking* scenario not only validates the motion prediction of the planning approaches but also evaluates how certain planners weigh social distancing against navigation performance.

The last scenario type, *crossing*, evaluates the robot's capabilities to comply with the social norm of yielding a way to a human that crosses its reference path. Namely, the robot begins operation in a laboratory corner (different than in *passing* and *overtaking*) and needs to move straight through the main aisle to reach the goal pose. However, two humans are located at the sides of the aisle, ready to intrude ahead of the robot (Fig. 7.9b). The first human starts its movement, perpendicular to the robot's initial heading, when its distance from the robot's centre is 2.7 m and moves at the speed of 0.3 m/s. In contrast, the second human, whose speed is limited to 0.25 m/s, initiates crossing the robot's path when the distance between them is 2.3 m. This scenario also verifies the robot's behaviour in dynamic environments, where the area, through which the reference path passes, might become occluded. This, in turn, may require additional movement actions for updating the environment model to find a new traversable path.

A sequence diagram corresponding to the progression of each scenario is shown in Fig. 7.10. The diagram describes the task interaction between the robot and humans, which applies to each scenario for both simulation and real-world experiments.

In all scenarios, the robot interacts with humans in an unfocused way; therefore,

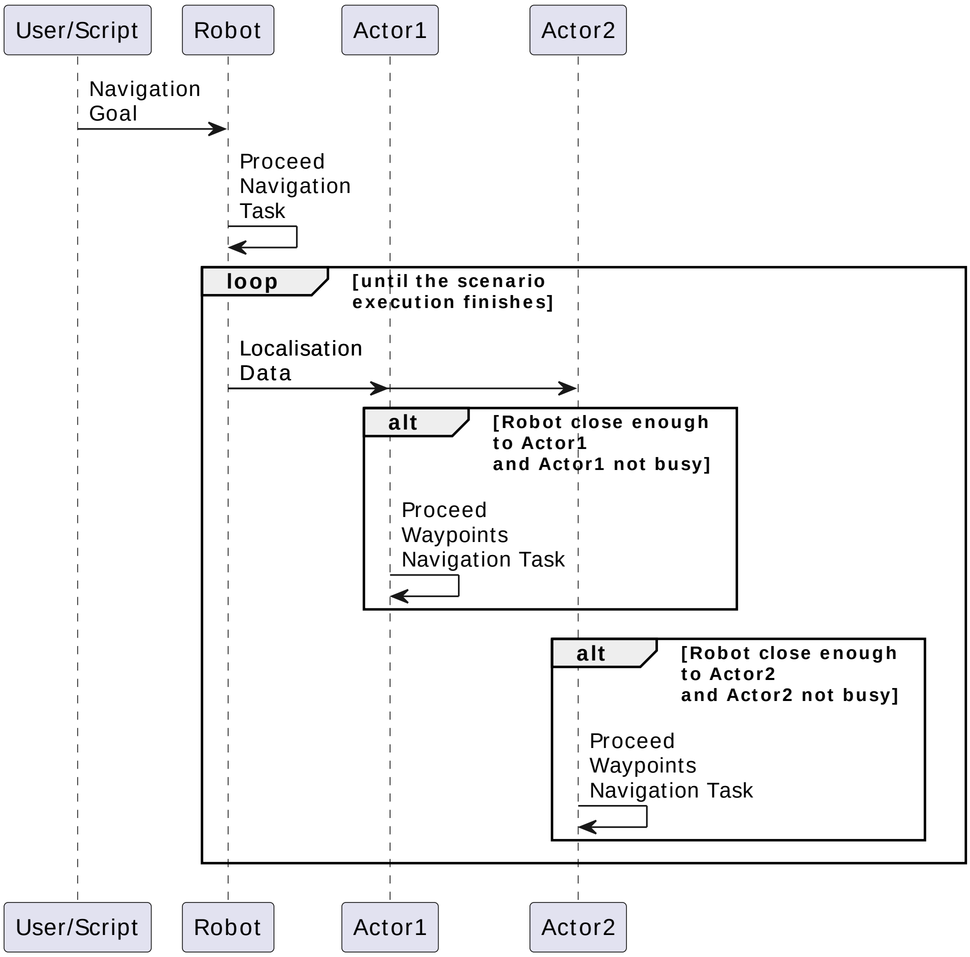


Figure 7.10: The sequence diagram of the robot and humans' activity during the designed scenarios.

the robot is expected to avoid collisions and maximise the perceived safety of humans. Each trial involved one-shot navigation ("PoseGoal" in the nomenclature from [420]) from a start pose to a goal pose – both fixed but scenario-specific.

7.4.2 Evaluation principles

The evaluation involved both traditional trajectory planners for mobile robots, namely: *Elastic Bands* [310], *DWA* [144], *Trajectory Rollout* [161], *TEB* [162], as well as state-of-the-art human-aware trajectory planners: *SRL-EBand* [169], *HaTEB* [216], *GA3C-CADRL* [163], *SARL* [350], *SARL** [351], *RG's DRL* [352], *CoHAN* [168], *DRL-VO* [156], which were compared with the proposed *HUMAP*. The public *ROS* implementations of the evaluated algorithms were utilised.

Each trajectory planner was aimed to be tested in each scenario's simulated and real-world variants. The planners selected for the study were first validated in a simulation to determine whether they were suitable for safe real-world operation. However, not all planners proved to be applicable for real-world trials due to prolonged in-place rotations or collisions, while others turned out impractical due to excessive rotational accelerations along the run (causing wheel slipping and localisation errors). Each investigated trajectory planner was benchmarked with 100 trials in a simulation, whereas at least 5 representative trials were recorded with prospective methods selected for the real-world experiments.

The principles for choosing the control frequency of each trajectory planner are the same as in the first phase of the experiments. Namely, trajectory planners operated with the maximum possible frequency which ensured that the designated trajectory planning period was not surpassed, but frequencies were limited to 10 Hz. Exceptionally, if the algorithm was designated by the authors for faster replanning, the original frequencies were maintained, if possible. Some planners were involved in the first phase of the experiments and have frequencies denoted in Sec. 7.3.2. Other planners operated at: *SRL-EBand* – 10 Hz, *GA3C-CADRL* – with a policy action update frequency of 10 Hz and control frequency of 100 Hz. Moreover, *SARL* and *SARL** operated at 4 Hz, *RG's DRL* – 10 Hz, *DRL-VO* – 20 Hz, and *HUMAP* at 4 Hz.

During this study, all trajectory planners (also the traditional ones) utilised costmap environment representations with human proxemics modelled as spatial costs around detected individuals [47, 84] – the effects of this approach are discussed in Sec. 6.5.

As the study type and the quantitative evaluation method are the same as in the first phase of the experiments (Sec. 7.3), some outcomes are similar, as several local trajectory planning methods appear in both studies. Nonetheless, the evaluation scenarios differ; hence, more insightful information on reused algorithms can be provided.

7.4.3 Qualitative evaluation

Examples of trajectories performed by each planner are shown in Fig. 7.11 (*crossing* scenario), Fig. 7.12 (*overtaking*), and Fig. 7.13 (*passing*). The colour of a symbol represents its occurrence in time (Fig. 7.11n, Fig. 7.12n, and Fig. 7.13n). Solid circles represent subsequent positions of the robot, whereas circles with dark edges represent human positions over time. Where available, successful trials were presented in the timing plots. Notably, subsequent trials of the same scenario might slightly differ, as indicated by a deviation of metric values discussed in the further part of the chapter.

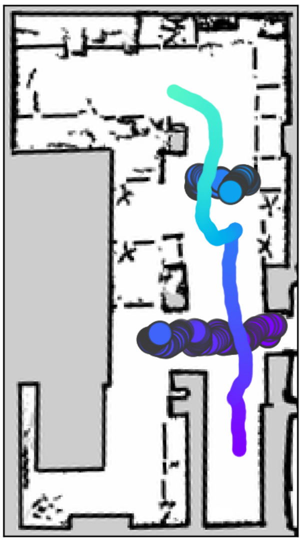
As stated in Sec. 7.1, information about humans in the environment is gathered by the robot's onboard sensors during the navigation task execution. Due to the limited *FOV* of robot sensors and inaccuracies in human tracking (performed on noisy raw sensor data), the human pose estimation is prone to errors. Therefore, in most identified visualisations of robot and human trajectories (with respect to the environment), humans' poses slightly jitter once the robot stops observing them. If ground truth human data were used, inaccuracies visible in the figures would not be present. Additionally, the trajectories of simulated humans are not scripted but follow waypoints using a planning algorithm, as the robot does. Therefore, slight differences in human trajectories in subsequent trials might occur.

Crossing scenario Major differences in the *crossing* scenario's execution are related to the way trajectory planners handled the situations of the robot's path being crossed by humans. The proposed *HUMAP* progresses smoothly while yielding a way to intruding individuals (due to the dedicated behaviour for handling that social convention, Fig. 7.11h), whereas other methods often perform full rotations in place to recover from sudden encounters. The *CoHAN* planner fluently passes the first person, but before the second intrusion, which is more aggressive (occurs at a closer distance between the robot and the human), performs a backing up movement before progressing further (Fig. 7.11g). A similar trajectory has been executed when operating with the *DWA* (Fig. 7.11b). In contrast, highly evasive movements are performed by the *Elastic Bands* (Fig. 7.11a) and *TEB* (Fig. 7.11d), but the latter reaches the goal considerably faster. Moreover, the *HaTEB* planner performs notable erratic turns along the way to the goal compared to the *TEB* (Fig. 7.11f).

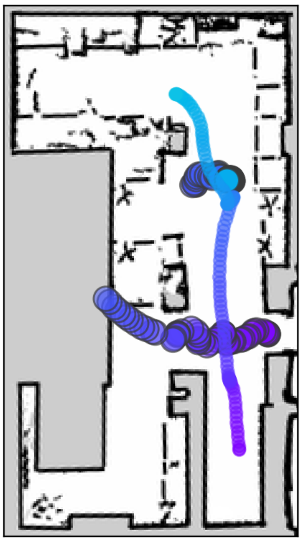
The example trajectories show that the *SRL-EBand* (Fig. 7.11e) and *Trajectory Rollout* planners (Fig. 7.11c) tend to pass the crossing humans at very close distances. Importantly, most *DRL*-based planners (*GA3C-CADRL*, *SARL*, *SARL**, and *RG's DRL*) did not exhibit any significant progression since the beginning of the scenario, performing mostly in-place rotations until the oscillation or experiment trial timeout expires. The observed behaviour might indicate that environments with narrow corridors were not involved in their training process. Nonetheless, among *DRL*-based planners, only *DRL-VO* successfully finished multiple trials.

Overtaking scenario The *overtaking* scenario aims to implement a standardised protocol for evaluating the robot's overtaking capabilities [10]. Depending on the performance of the planner, overtaking of the first human might occur in slightly different places. The presented examples of trajectories obtained with different planners (Fig. 7.12) illustrate typical gaps kept from humans and obstacles by the robot, as well as the smoothness of trajectories created by various methods.

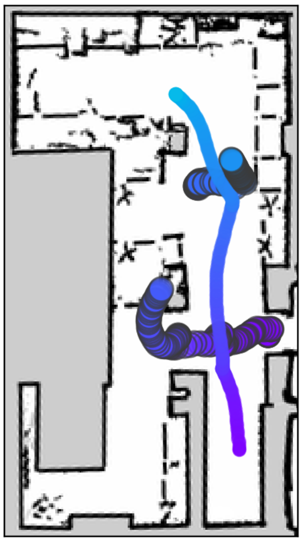
The most atypical trajectory has been recorded when the robot operated with the *Elastic Bands* planner (Fig. 7.12a). Specifically, instead of overtaking, the algorithm produced commands that provided following the humans throughout the narrow passage and once a substantial gap was unveiled (the humans' scenario of operation had ended), the



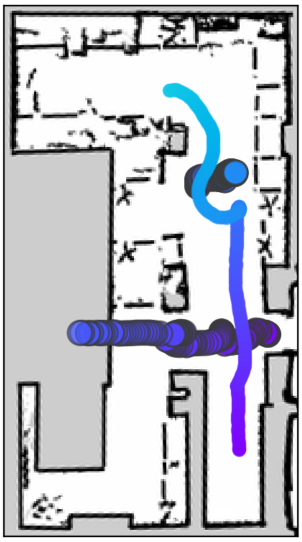
(a) *Elastic Bands*



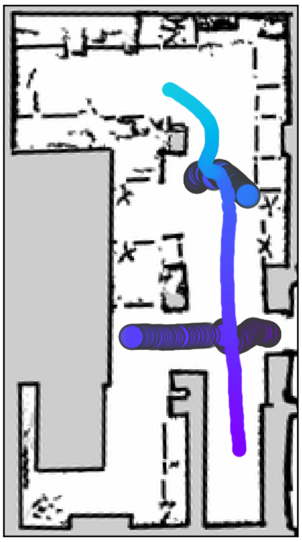
(b) *DWA*



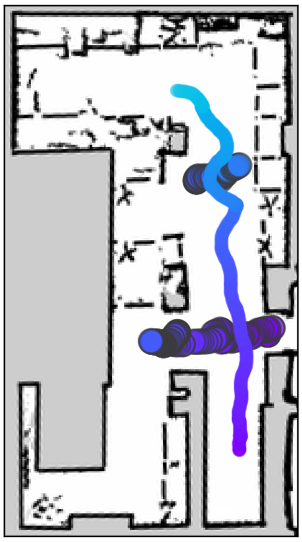
(c) *Trajectory Rollout*



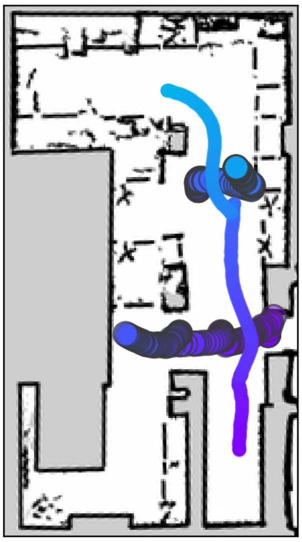
(d) *TEB*



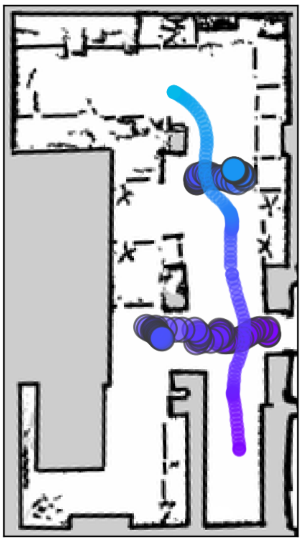
(e) *SRL-EBand*



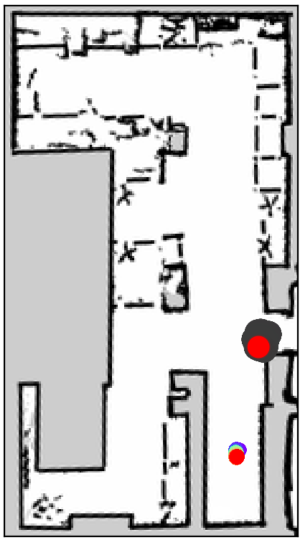
(f) *HaTEB*



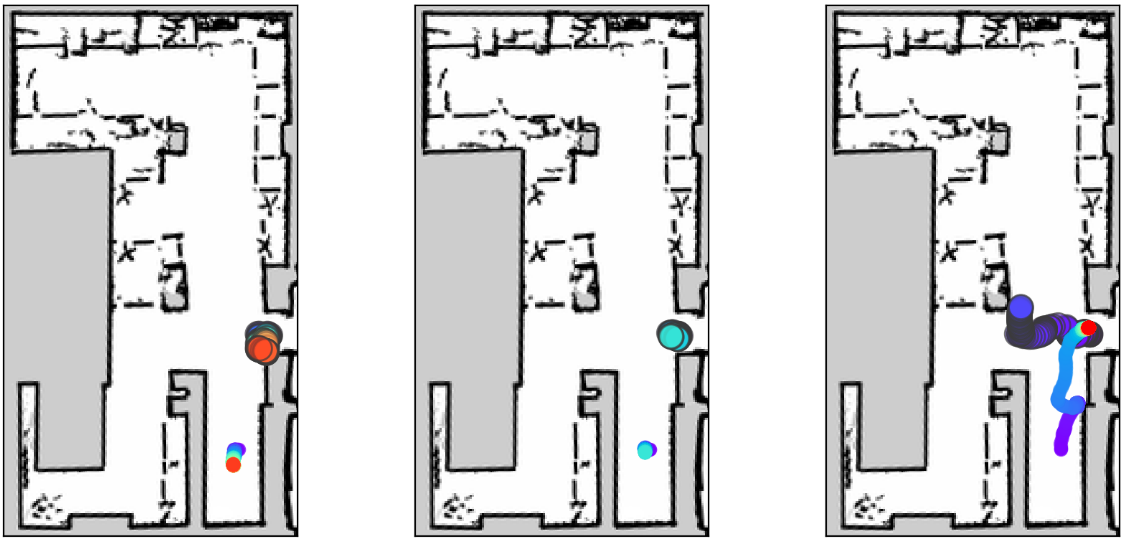
(g) *CoHAN*



(h) *HUMAP*



(i) *GA3C-CADRL*



(j) *SARL* (k) *SARL** (l) *RG's DRL*

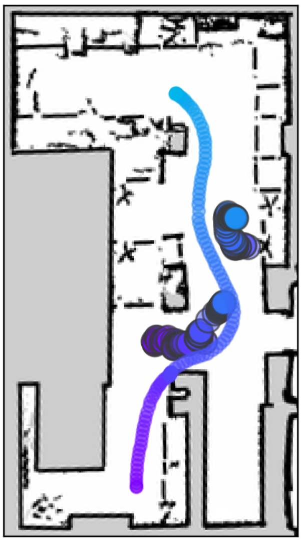
(m) *DRL-VO* (n) Time

Figure 7.11: Example robot trajectories generated with various planners along with observed human trajectories in the *crossing* scenario performed in the simulation environment.

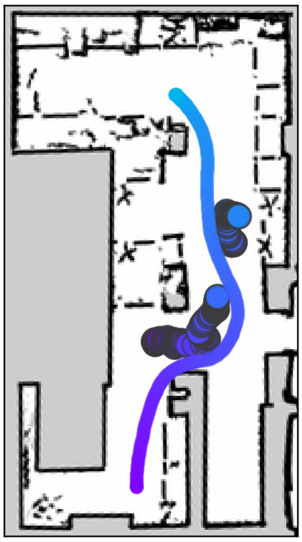
robot traversed that space. Therefore, numerous human poses are marked in the figure, as the robot observed them during the trial. While the robot has achieved a navigation goal in the given test, the time required to reach the target pose was considerably longer compared to if the robot had overtaken slowly moving humans. A distinctive characteristic of the *HaTEB* method, which generates trajectories with numerous heading changes is also visible in this scenario (Fig. 7.12f).

Remarkably, none of the *DRL*-based planners progressed substantially from the beginning configurations (Fig. 7.12i, 7.12j, 7.12k, 7.12l, 7.12m). Also, in the trials of *SARL* and *RG's DRL* planners, the first human to be overtaken is only visible at the beginning of

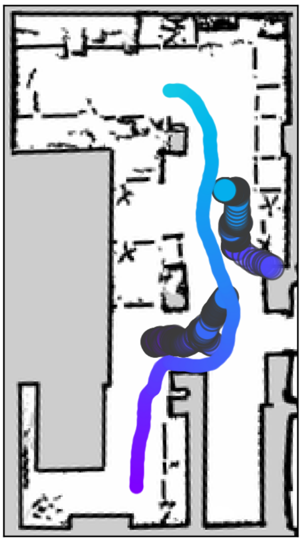
(a) *Elastic Bands*



(b) *DWA*



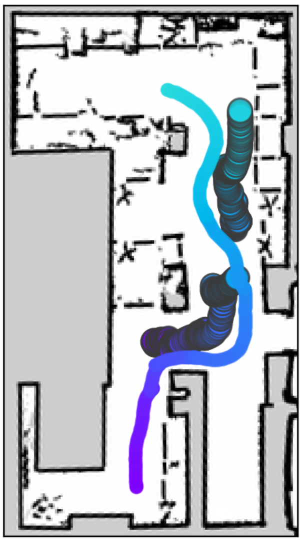
(c) *Trajectory Rollout*



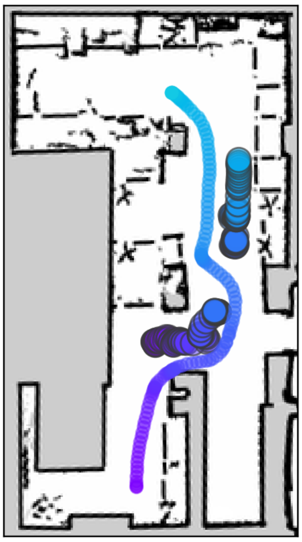
(d) *TEB*

(e) *SRL-EBand*

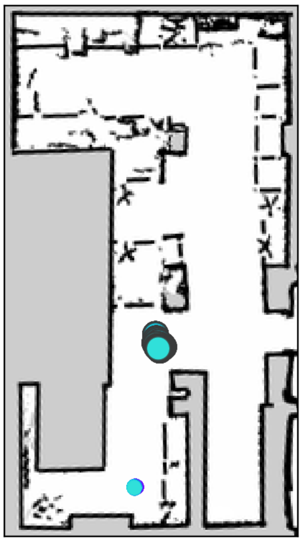
(f) *HaTEB*



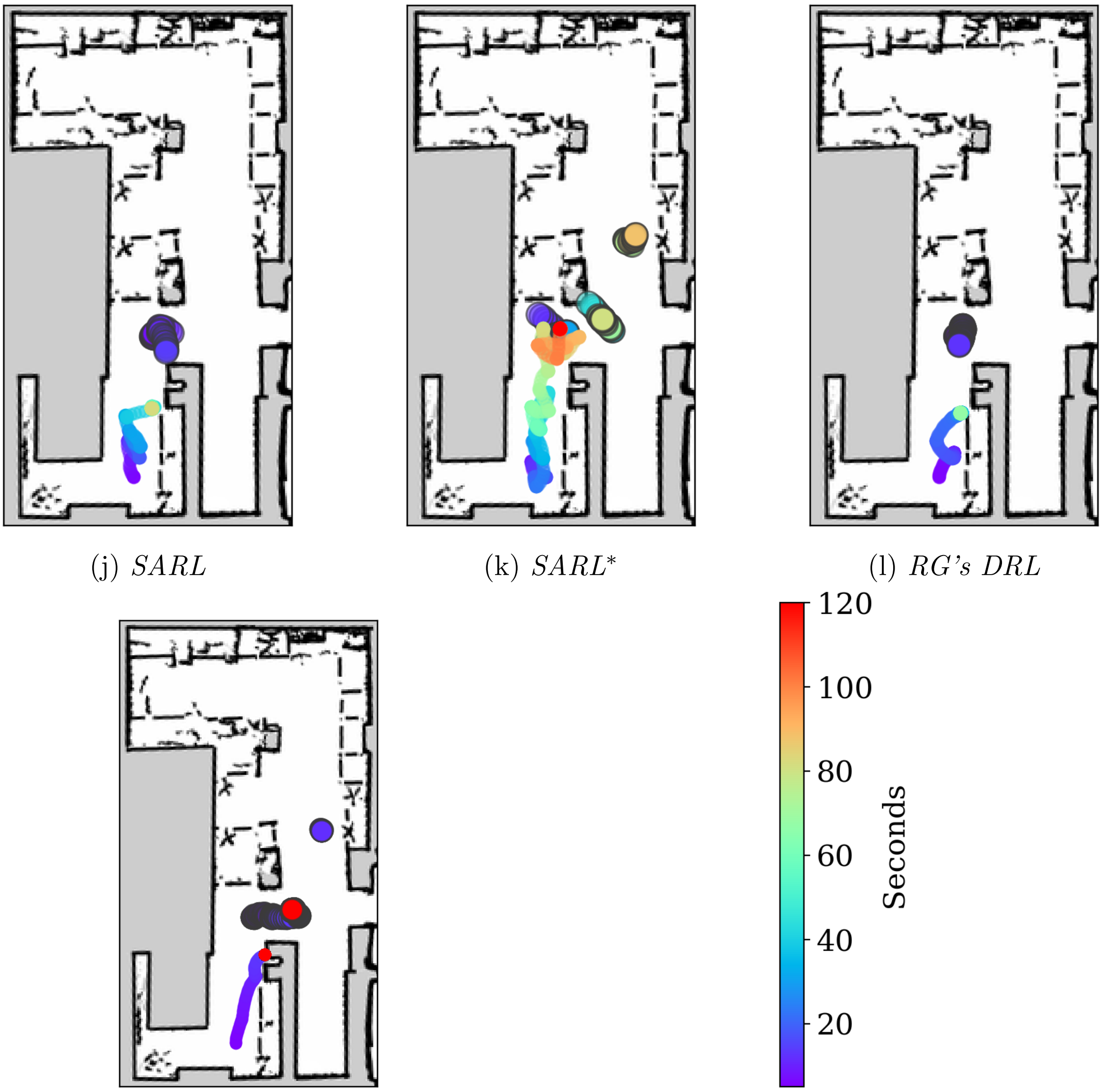
(g) *CoHAN*



(h) *HUMAP*



(i) *GA3C-CADRL*



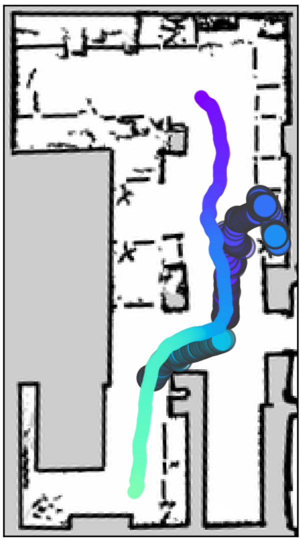
(j) *SARL* (k) *SARL** (l) *RG's DRL*

(m) *DRL-VO* (n) Time

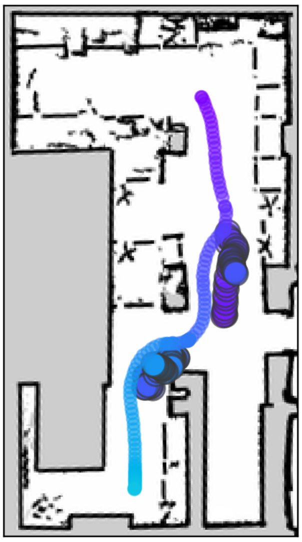
Figure 7.12: Example robot trajectories generated with various planners along with observed human trajectories in the *overtaking* scenario performed in the simulation environment.

the experiment. Humans start their waypoint following tasks once the distance between them and the robot is smaller than the threshold; therefore, once the robot approaches them, the humans continue their navigation regardless of prolonged robot oscillations.

Passing scenario The designed *passing* scenario mainly intends to evaluate the robot's motion legibility [23] but also its capability to operate among humans approaching from the opposite direction in narrow passages. The concept of motion legibility can be assessed in such a scenario by identifying the moment when a trajectory planner generates commands that tend the robot to the right side for a safe pass and early signalling of

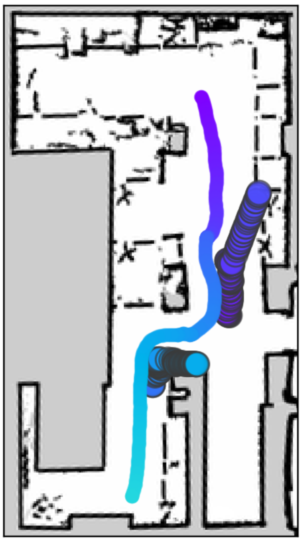


(a) *Elastic Bands*

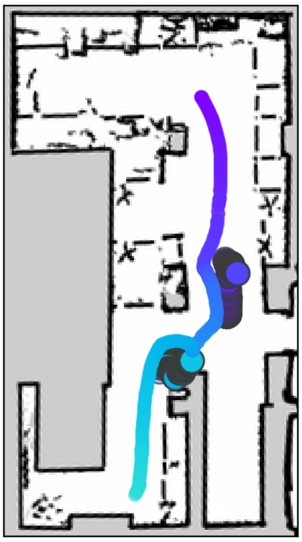


(b) *DWA*

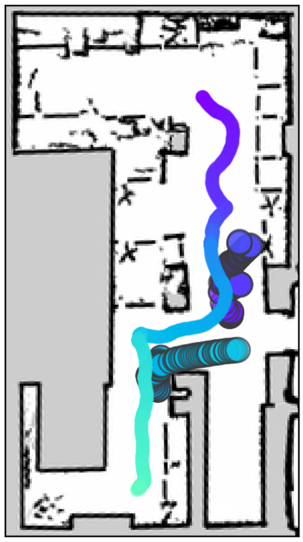
(c) *Trajectory Rollout*



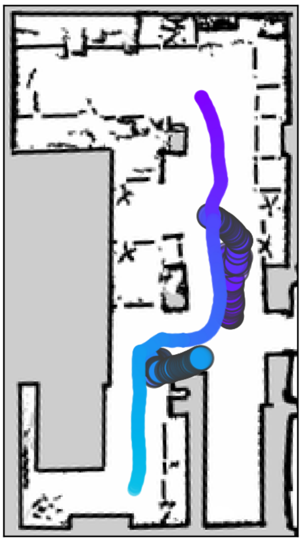
(d) *TEB*



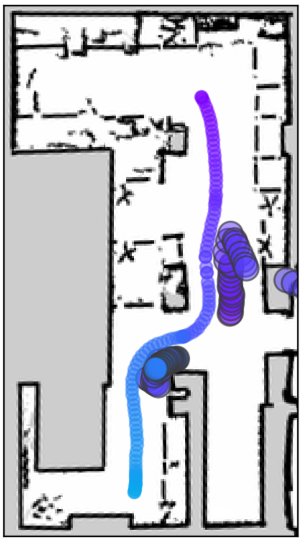
(e) *SRL-EBand*



(f) *HaTEB*

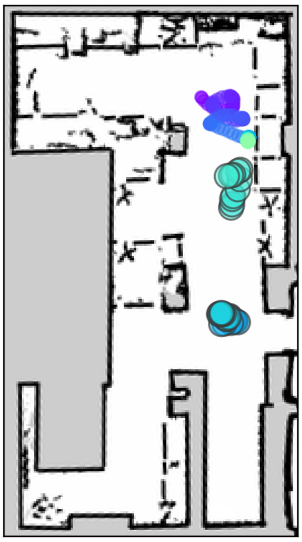


(g) *CoHAN*

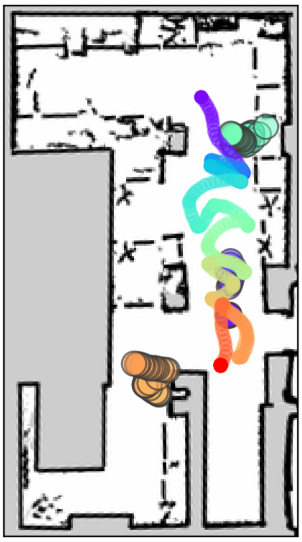


(h) *HUMAP*

(i) *GA3C-CADRL*

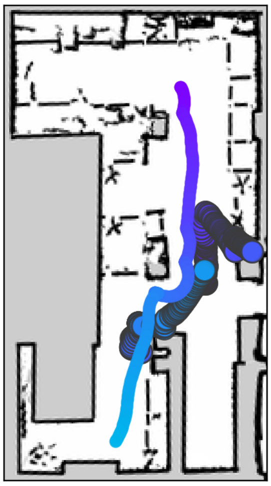


(j) *SARL*

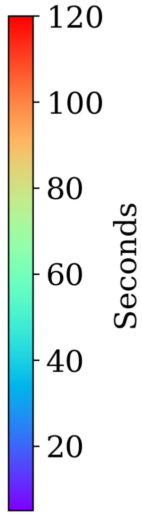


(k) *SARL**

(l) *RG's DRL*



(m) *DRL-VO*



(n) Time

Figure 7.13: Example robot trajectories generated with various planners along with observed human trajectories in the *passing* scenario performed in the simulation environment.

intentions.

The example trajectories show that *DWA* adjusts the robot heading just before the imminent collision with the first human (Fig. 7.13b). In contrast, *Trajectory Rollout* signals the intention of passing on the right early but keeps a minimum distance from the first human (Fig. 7.13c). Additionally, after the encounter with the second human, the planner generates in-place rotations until the second human moves away and then moves on the inside of the S-turn. Alternatively, while the *HaTEB* maintains a considerable gap between the robot and a human, the generated commands along the way produce erratic heading changes (Fig. 7.13f). Surprisingly, the human-aware *CoHAN* adjusts the robot's heading marginally when passing the first human; hence, the gap kept is minimal (Fig. 7.13g). In contrast, the proposed *HUMAP* traverses smoothly passing both humans at a noticeable distance and avoiding erratic motions (Fig. 7.13h).

Notably, in the reference trial (Fig. 7.13i), the robot operating with the *GA3C-CADRL* method successfully passed the first human (with several in-place rotations throughout the path), but missed the desired S-turn and got stuck at the extension of the narrow passage. Although the *SARL* barely moved from the initial pose (Fig. 7.13j), the *SARL** reached a similar stage of the scenario as the *GA3C-CADRL* but executed more evasive motions (Fig. 7.13k). Also, human detections with the *SARL* planner are first captured at the beginning of the experiment, and the following human poses are observed once the first dynamic human is close to the robot. This is caused by numerous in-place rotations produced by the *SARL* method that lead to prolonged time without human observation.

Importantly, in the presented trial, the *RG's DRL* approach almost reached the goal pose (performing numerous in-place rotations along the way), but the target pose was not achieved due to the experiment timeout (the same duration for all planners, Fig. 7.13l). Moreover, in the presented trial, the *RG's DRL* approach omitted to interact with humans, as they moved along the predefined waypoints, while the robot rotated in place nearby. The example trajectories show that the *DRL-VO* considerably outperformed other *DRL*-based planners in terms of the progress towards the goal and time required to reach the target pose (Fig. 7.13m).

7.4.4 Quantitative evaluation

Quantitative metrics of the Social Robot Planner Benchmark, which has been used for assessing the performance of different trajectory planning algorithms, can be divided into two categories. The first concerns the robot navigation performance reflected by, e.g., the time required to reach the goal or path lengths, whereas the second group regards the human discomfort measures. The latter can be further divided into subcategories, as discussed further in this section.

Numerous tests have been performed to obtain accurate quantitative assessment and statistical measures of the metrics. Extensive simulation experiments have been performed, which involved 100 trials executed with each planner in each scenario. The large-scale study facilitates showing explicitly the distribution of metric values obtained throughout the tests. Several metrics are presented in such detail using violin plots, which are similar to box plots, but can illustrate not only the minimum, maximum, and mean, but also the aforementioned distribution of a metric (e.g., Fig. 7.16). Generally, for all *SRPB* metrics except m_{goal}, m_{obs}, ${}_{\text{dist}}m_{\text{obs}}$, and ${}_{\text{min}}m_{\text{obs}}$, lower values indicate better performance.

The quantitative results of the experiments, condensed into medians of each metric assessing individual trajectory planners, are presented in Tab. 7.4 and 7.5. Metric results of the unfinished attempts (failures caused by, e.g., a collision or exceeding the experiment timeout due to infinite oscillations) are excluded from all calculations. The results were divided into two tables, as several local trajectory planners obtained low navigation success goals (Tab. 7.5); therefore, their other metric values will not be discussed. In Tab. 7.4 and 7.5, simulation trials are marked as S, while real-world ones as R; therefore, *passing* scenarios are marked as P-S and P-R, *overtaking* trials – as O-S and O-R, whereas *crossing* tests – as C-S and C-R. The results presented in the tables are discussed below.

Metric	Scenario \ Method	*Elastic Bands*	*DWA*	*Trajectory Rollout*	*TEB*	*SRL-EBand*	*HaTEB*	*CoHAN*	*HUMAP*
m_{goal} [%]	*C-S*	99.00	84.00	52.00	96.00	82.00	81.00	91.00	**100.00**
	C-R	—	**100.00**	80.00	**100.00**	—	—	80.00	**100.00**
	O-S	30.00	92.00	95.00	**100.00**	84.00	50.00	78.00	**100.00**
	O-R	—	100.00	100.00	100.00	—	—	100.00	100.00
	P-S	89.00	98.00	91.00	**100.00**	95.00	89.00	78.00	**100.00**
	P-R	—	**100.00**	80.00	80.00	—	—	80.00	**100.00**
$_{\text{dist}}m_{\text{obs}}$ [m]	*C-S*	**0.69**	0.54	0.53	0.60	0.59	0.65	0.67	0.63
	C-R	—	0.54	0.56	0.62	—	—	**0.67**	0.61
	O-S	**0.64**	0.57	0.53	0.58	0.59	0.54	0.63	0.59
	O-R	—	0.56	0.57	0.56	—	—	**0.60**	0.56
	P-S	**0.66**	0.54	0.50	0.57	0.59	0.62	0.63	0.56
	P-R	—	0.54	0.52	0.54	—	—	**0.63**	0.54
$_{\text{min}}m_{\text{obs}}$ [m]	*C-S*	0.51	0.29	0.28	0.43	0.29	0.39	**0.53**	0.34
	C-R	—	0.28	0.30	**0.38**	—	—	0.38	0.30
	O-S	0.30	0.31	0.29	0.34	0.33	0.28	**0.36**	0.33
	O-R	—	0.31	0.29	**0.34**	—	—	0.33	0.33
	P-S	**0.45**	0.29	0.29	0.34	0.31	0.30	0.30	0.31
	P-R	—	0.30	0.30	0.33	—	—	**0.34**	0.29
m_{mef} [s]	*C-S*	50.30	34.00	39.25	29.11	38.58	36.05	**27.20**	31.25
	C-R	—	33.00	32.13	34.10	—	—	36.05	**29.75**
	O-S	107.75	**27.50**	36.00	34.40	51.58	43.10	52.60	32.50
	O-R	—	**26.50**	35.54	33.20	—	—	46.60	36.75
	P-S	70.55	**27.50**	36.06	31.30	39.33	43.55	43.80	29.50
	P-R	—	**25.13**	35.00	31.50	—	—	57.90	30.25

Metric \ Scenario	Method	*Elastic Bands*	*DWA*	*Trajectory Rollout*	*TEB*	*SRL-EBand*	*HaTEB*	*CoHAN*	*HUMAP*
m_{plin} [m]	*C-S*	10.43	10.36	10.39	10.09	9.98	10.86	10.44	**9.84**
	C-R	—	9.61	9.65	10.23	—	—	11.88	**9.46**
	O-S	13.40	**11.38**	11.70	12.35	12.20	12.62	14.67	12.05
	O-R	—	**10.96**	11.72	11.94	—	—	13.73	12.10
	P-S	11.63	10.89	**10.65**	11.32	11.26	12.63	12.70	11.10
	P-R	—	**10.63**	11.35	11.23	—	—	14.13	11.15
m_{chc} [rad]	*C-S*	21.03	6.02	17.43	8.41	7.94	15.70	8.60	**5.81**
	C-R	—	**4.99**	10.42	10.12	—	—	8.72	5.17
	O-S	52.65	**5.85**	9.79	11.15	17.04	16.86	12.86	7.03
	O-R	—	**4.63**	6.57	8.92	—	—	10.13	7.20
	P-S	71.70	16.90	**16.05**	20.86	23.81	42.96	21.05	18.53
	P-R	—	**4.19**	20.30	21.43	—	—	32.71	17.46
m_{cef} $[10^{-3} \cdot \text{s}]$	*C-S*	4.66	76.14	36.74	**3.66**	4.52	5.19	5.08	122.46
	C-R	—	119.63	74.70	**6.05**	—	—	8.98	117.11
	O-S	8.14	90.15	37.00	**3.82**	8.15	6.10	7.00	153.31
	O-R	—	155.51	82.56	**6.19**	—	—	10.46	121.12
	P-S	4.91	85.39	35.11	**3.16**	4.36	5.31	5.54	130.95
	P-R	—	158.11	74.61	**4.78**	—	—	8.61	110.66
m_{cre} $[10^{-3} \cdot \text{s}]$	*C-S*	5.61	36.19	13.54	**3.38**	15.93	3.49	3.90	67.30
	C-R	—	67.83	22.63	**4.07**	—	—	5.63	69.98
	O-S	8.75	31.81	13.76	**3.19**	21.21	3.91	4.87	59.70
	O-R	—	56.35	20.60	**4.05**	—	—	5.60	62.59
	P-S	7.13	30.99	15.17	**2.53**	15.13	3.40	4.24	47.84
	P-R	—	50.69	27.77	**2.18**	—	—	5.79	53.05
m_{vsm} $\left[\frac{\text{m}}{\text{s}^2}\right]$	*C-S*	0.16	0.14	0.12	**0.09**	0.24	0.38	0.21	0.14
	C-R	—	0.22	**0.16**	0.17	—	—	0.28	0.19
	O-S	0.19	**0.10**	0.11	0.15	0.31	0.40	0.35	0.15
	O-R	—	**0.12**	0.15	0.20	—	—	0.31	0.17
	P-S	0.12	0.12	**0.09**	0.14	0.27	0.37	0.27	0.13
	P-R	—	**0.15**	0.16	0.22	—	—	0.32	0.17
m_{hsm} $\left[\frac{\text{rad}}{\text{s}^2}\right]$	*C-S*	1.43	**0.28**	0.37	0.99	0.77	1.00	1.00	0.34
	C-R	—	**0.29**	0.38	0.86	—	—	0.84	0.39
	O-S	1.38	**0.24**	0.28	1.13	0.96	1.08	0.99	0.31
	O-R	—	**0.25**	0.33	0.91	—	—	0.78	0.35
	P-S	1.46	0.26	**0.24**	1.08	0.79	0.91	0.84	0.33
	P-R	—	**0.23**	0.37	0.88	—	—	0.88	0.35

Metric \ Scenario	Method	*Elastic Bands*	*DWA*	*Trajectory Rollout*	*TEB*	*SRL-EBand*	*HaTEB*	*CoHAN*	*HUMAP*
m_{osc} [%]	*C-S*	1.73	2.43	2.04	**1.08**	1.96	3.97	1.38	3.89
	C-R	—	10.20	6.76	**4.55**	—	—	7.30	8.77
	O-S	7.06	1.85	1.50	**0.90**	4.78	9.59	5.42	1.68
	O-R	—	3.92	3.47	**2.41**	—	—	9.57	6.70
	P-S	4.67	2.66	1.46	**1.26**	2.15	5.48	11.13	1.74
	P-R	—	4.09	3.94	**2.64**	—	—	14.16	5.59
m_{bwd} [%]	*C-S*	**0.00**	9.14	0.94	0.35	**0.00**	3.32	11.78	3.94
	C-R	—	4.93	**0.00**	0.62	—	—	20.43	**0.00**
	O-S	**0.00**	**0.00**	**0.00**	0.28	**0.00**	3.16	19.60	**0.00**
	O-R	—	**0.00**	**0.00**	**0.00**	—	—	14.53	**0.00**
	P-S	**0.00**	0.91	0.50	0.96	**0.00**	3.99	15.39	**0.00**
	P-R	—	**0.00**	0.14	**0.00**	—	—	16.09	**0.00**
m_{iprot} [%]	*C-S*	7.28	3.65	15.13	3.18	5.85	1.08	**0.00**	4.80
	C-R	—	2.38	7.52	2.25	—	—	**0.28**	2.15
	O-S	11.12	3.67	14.27	0.61	17.92	0.64	**0.41**	3.97
	O-R	—	1.87	0.91	0.30	—	—	**0.30**	2.33
	P-S	5.27	0.93	29.39	0.66	12.23	0.80	**0.29**	1.55
	P-R	—	**0.00**	16.38	0.44	—	—	0.84	0.62
$_{\max}m_{\mathrm{psi}}$ [%]	*C-S*	**66.86**	89.17	90.22	76.47	86.12	80.83	69.34	70.93
	C-R	—	89.71	70.35	76.79	—	—	78.01	**65.06**
	O-S	96.86	87.75	87.74	90.34	85.83	93.09	87.59	**81.04**
	O-R	—	94.39	83.81	83.61	—	—	**72.47**	72.99
	P-S	91.22	95.42	91.61	88.13	95.02	**84.69**	94.12	86.85
	P-R	—	82.48	87.95	74.68	—	—	74.90	**73.32**
$_{\max}m_{\mathrm{dir}}$ [%]	*C-S*	12.88	15.07	12.57	12.67	17.20	24.25	19.41	**10.03**
	C-R	—	5.27	3.39	6.55	—	—	4.19	**2.97**
	O-S	**13.31**	32.79	28.48	24.21	28.27	35.54	25.43	22.91
	O-R	—	23.71	17.43	14.36	—	—	12.48	**4.77**
	P-S	**14.60**	62.85	17.90	26.34	43.81	26.16	29.25	23.24
	P-R	—	34.77	28.13	23.53	—	—	43.62	**18.72**
$_{\max}m_{\mathrm{psd}}$ [%]	*C-S*	**11.05**	57.11	66.08	25.59	57.33	29.57	22.12	24.67
	C-R	—	50.96	21.80	17.24	—	—	19.74	**14.92**
	O-S	57.08	50.18	57.14	64.55	53.44	88.42	45.62	**37.26**
	O-R	—	75.62	60.08	61.40	—	—	33.78	**20.88**
	P-S	41.63	86.90	55.59	**32.86**	76.09	41.29	63.44	34.95
	P-R	—	39.00	60.45	**20.87**	—	—	25.27	23.36

Table 7.4: Quantitative results of the performance of different local trajectory planners during the conducted experiments. Real-world trials are marked as R, whereas simulated as S. The scenario naming convention is prepared according to the following scheme: C – *crossing*, O – *overtaking*, and P – *passing*.

Metric \ Scenario	Method	*GA3C-CADRL*	*SARL*	*SARL**	*RG's DRL*	*DRL-VO*
m_{goal} [%]	*C*	0.00	0.00	0.00	0.00	**38.00**
	O	0.00	0.00	0.00	0.00	0.00
	P	0.00	0.00	0.00	0.00	**2.00**
${}_{\text{dist}}m_{\text{obs}}$ (*) [m]	*C*	**0.63**	0.58	0.58	0.42	0.49
	O	**0.72**	0.60	0.59	0.43	0.34
	P	0.75	0.64	**0.83**	0.52	0.38
${}_{\text{min}}m_{\text{obs}}$ (*) [m]	*C*	**0.58**	0.57	0.57	0.29	0.28
	O	**0.64**	0.39	0.38	0.30	0.28
	P	0.43	0.28	**0.51**	0.29	0.28

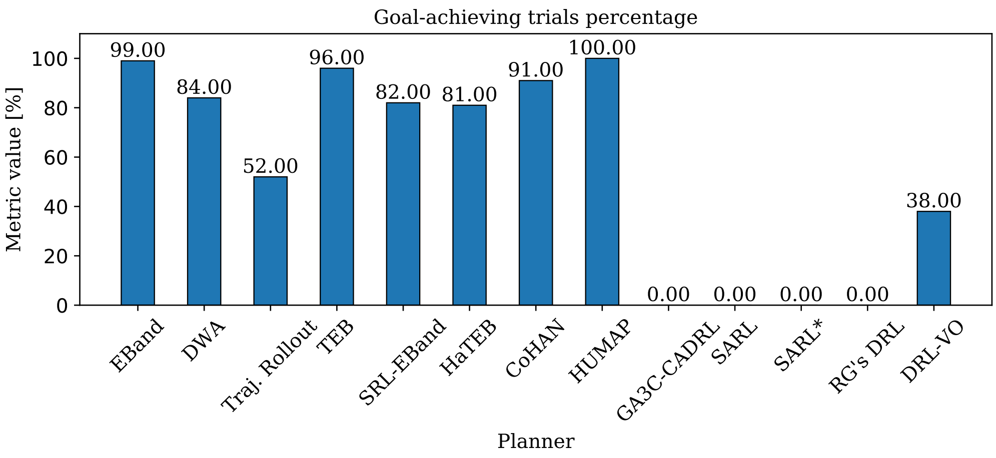
Table 7.5: Quantitative results of the performance of *DRL*-based local trajectory planners during the simulation experiments. The naming convention of scenarios follows the scheme: *C* – *crossing*, *O* – *overtaking*, and *P* – *passing*. (*): Exceptionally, values of the ${}_{\text{dist}}m_{\text{obs}}$ and ${}_{\text{min}}m_{\text{obs}}$ metrics are collected from all trials (instead of only successful ones) since planners included in this table obtained 0% of navigation success rate (m_{goal}) in most cases.

7.4.5 Robustness of the examined local trajectory planners

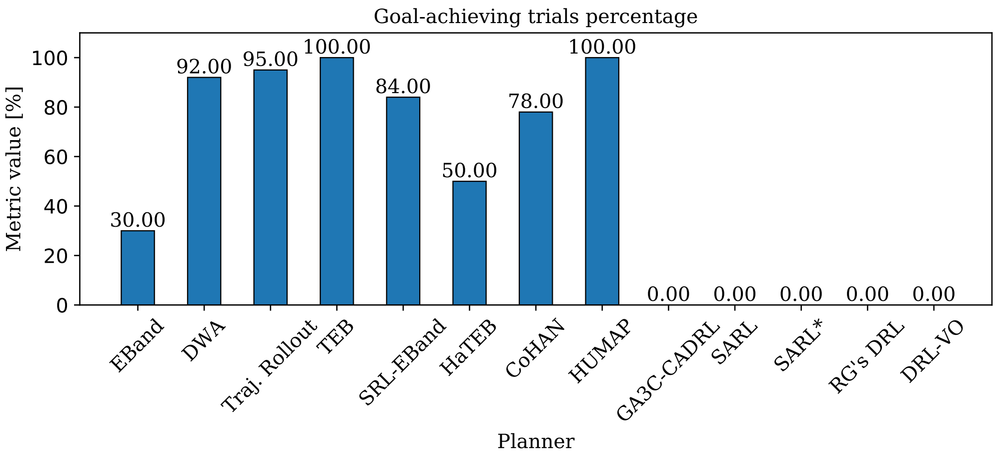
The robustness of local trajectory planners (**Req. 1.3**) is associated with the success rate of designated navigation tasks. The relevant metric, m_{goal}, indicates the percentage of successfully completed trials from the total number of started tests. A "successfully completed trial" represents a test in which the goal has been reached with the desired tolerance thresholds for target position and orientation. Trials can finish unsuccessfully due to violating those tolerance thresholds or prolonged oscillations (exceeded timeout, as explained in Sec. 6.3). The score of the m_{goal} metric obtained by each examined method in simulation experiments (Fig. 7.14) was the key factor when selecting the planners for the real-world trials (Fig. 7.15 and Tab. 7.4).

Analytical local trajectory planners The method with the highest success rate, m_{goal}, is the *HUMAP*, effectively finishing 100% of trials both in a simulation (Fig. 7.14) and in the real-world experiments (Fig. 7.15). The main factor providing such a robust operation is its environmental awareness implemented as the *FSM*'s predicates that orchestrate executed behaviour. Notably, one of the *HUMAP*'s behaviours performs additional actions to gather observations about the robot's surroundings as a response to the occlusion of the globally planned path.

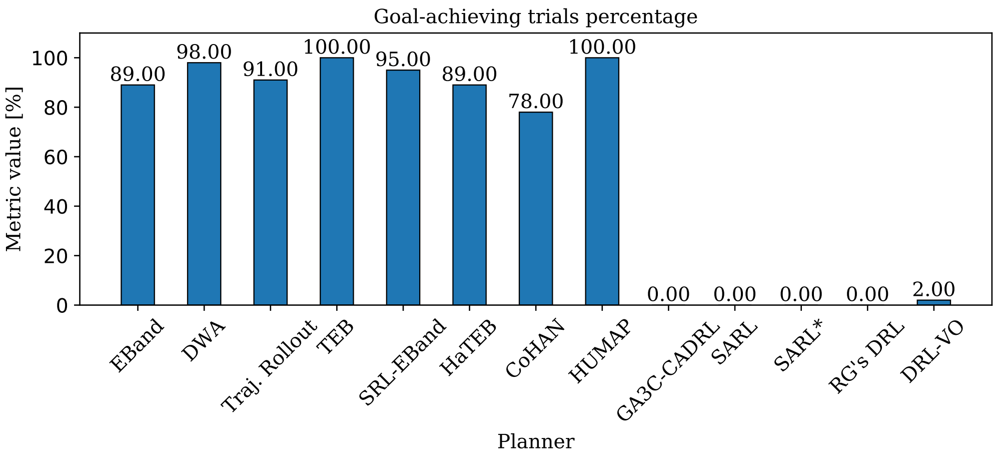
The *TEB* planner was nearly as efficient in task finishing as the *HUMAP*, obtaining m_{goal} =100% in *overtaking* and *passing* scenarios, but 96% in the *crossing*. For example,



(a) *crossing* scenario



(b) *overtaking* scenario



(c) *passing* scenario

Figure 7.14: Navigation success rates, m_{goal}, obtained with various trajectory planners in each scenario of the simulation experiments.

the *Trajectory Rollout* has a significantly smaller success rate in the *crossing* scenario; hence, it might not be suitable for environments when sudden intrusions and encounters are possible. Alike phenomenon occurs with the *DWA* planner, which is algorithmically similar to *Trajectory Rollout*, but the degradation is not as remarkable. Alternatively, *Elastic Bands* has also a considerably lower score of finished trials in the *overtaking* scenario compared to *crossing* and *passing*, which might indicate that it is not adequate for highly dynamic scenarios in challenging environments.

Besides the *HUMAP*, the most reliable among the human-aware trajectory planners are the *SRL-EBand* and *CoHAN*. The robot operating under *HaTEB* performed comparably, except for the *overtaking* scenario, where the planner completed only 50% of trials.

Local trajectory planners based on Deep Reinforcement Learning In simulation experiments, *GA3C-CADRL*, *SARL*, *SARL**, and *RG's DRL* planners exhibited 0% success rate in simulation trials of all scenarios; therefore, they were disqualified from real-world tests. Similarly, *DRL-VO*, with 38% success rate in the simulated *crossing* scenario, 0% in *overtaking*, and 2% in *passing*. In fact, most *DRL*-based planners frequently performed multiple in-place rotations but managed to traverse various segments of the reference paths. Even some *DRL*-based planners that have 0% navigation success rates were able to reach up to approximately 80% of the reference path of the passing scenario in some trials (Fig. 7.13l) but the tasks were aborted due to prolonged oscillations or the experiment timeout expiration (120 s).

The most problematic circumstances for *DRL*-based planners were narrow passages either in static or dynamic situations. The reason behind inadequate outcomes, as stated in Sec. 6.4, is that environment configurations similar to those encountered during the presented studies were not included during the *DRL* algorithm training process. This is related to the limited generalisability of this class of methods, which is their substantial drawback. The *DRL* policies used during the conduction of the experiments were provided by the authors of the algorithms. Notably, not all algorithm implementations were prepared to be fine-tuned for adapting policies to new environments. Instead of learning from scratch, the original policies were used to avoid the influence of underfitting or overfitting the algorithms. The approach for integrating *DRL* policies relies on adjusting robot sensor data for compatibility with specific algorithms (e.g., resolution of *LiDAR* scans, Sec. 6.4). Another cause of underperformance of *DRL*-based approaches is that the high-fidelity *Gazebo* simulator has been used during simulation experiments. It provides realistic raw sensor data with a small noise included (which is also present when using the real hardware), while most of the *DRL*-based planners were trained using simplified simulators (e.g., *flatland* (Sec. 3.3.3) has been used for learning the policy of the *RG's DRL*).

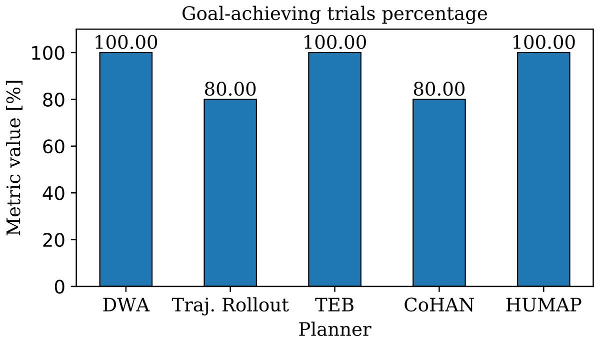
In recent work, Kästner et al. [418] stated that *DRL* approaches are not suitable for long-range navigation due to their proneness to local minima and lack of long-term memory. The results presented in Fig. 7.14 and 7.15 confirm their findings, especially considering that some *DRL*-based planners operated along with paths obtained from the global planner (detailed in Sec. 6.5). Although those local planners usually perform well in the tests conducted by the authors, it seems that the richness of social robot navigation intricacies makes the generalisation to different environments problematic for *DRL*-based planners. It is possible that in simpler environments (wider passages), easier scenarios (no path crossing or overtaking), or with low-fidelity simulators, the learning-based algorithms might perform more reliably. Another significant factor is that the policies were not adjusted for the test environment and specific scenarios.

Overall, the *HUMAP* planner outperforms other methods in terms of robustness reflected by the m_{goal} metric. The success rates of other approaches vary across both scenarios as well as environments, since in the simulation each scenario has been repeated with every trajectory planner 100 times, whereas in the real world – 5 times.

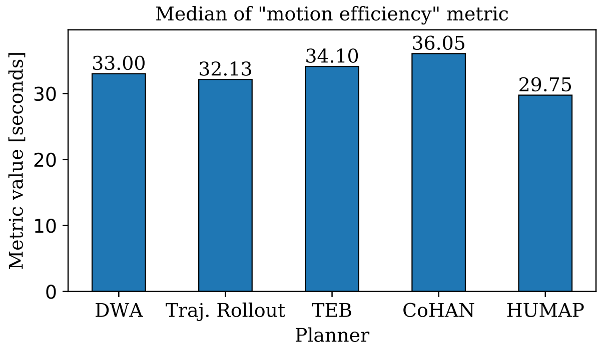
7.4.6 Robot navigation task performance

The performance results are associated with the verification of traditional robot navigation requirements fulfilment. Specifically, the requirements include: collision avoidance (**Req. 1.1**) – reflected by the ${}_{\text{dist}}m_{\text{obs}}$ and ${}_{\text{min}}m_{\text{obs}}$ metrics (mean and minimum distances, accordingly), goal reaching capabilities (**Req. 1.3**) – reflected by the m_{mef} metric (complementary to m_{goal}), taking the shortest possible path (**Req. 1.4**) – reflected by m_{plin}, and minimising path irregularity (**Req. 1.5**) – reflected by m_{chc}. Additionally, assessment criteria include the evaluation of computation times of a single trajectory planning procedure, m_{cef}, which is relevant for assessing whether an algorithm is capable of real-time operation on real hardware, and corresponding computation times repeatability, m_{cre}. The latter determines how likely the planner will violate requested computation times and, thus, whether it can be safely applied in robots operating in highly dynamic environments.

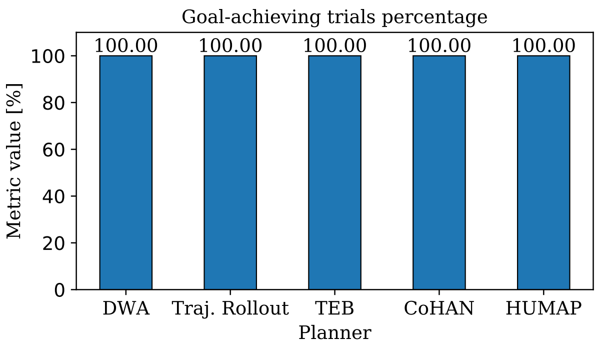
Collision avoidance The collision avoidance capabilities are discussed based on the values of the ${}_{\text{min}}m_{\text{obs}}$ metric calculated for each examined trajectory planner (Tab. 7.4 and 7.5). The metric represents the minimum distance between the robot's centre and edges of all obstacles (i.e., including the robot's circumradius) throughout a certain trial. The footprint of the robot employed for the experiments (both simulated and real-world) is circular with a (circum)radius of $d_{\text{cr}} = 0.275$ m; therefore, ${}_{\text{min}}m_{\text{obs}} = 0.28$ m (results are rounded to 2 decimal places) indicates that a collision might have occurred during a trial. The ${}_{\text{min}}m_{\text{obs}}$ is related to the distances from all types of obstacles in the robot's environment. A corresponding "collision avoidance" indicator regarding only humans is the ${}_{\text{max}}m_{\text{psi}}$ metric that assesses the scale of the robot's intrusions into the personal spaces



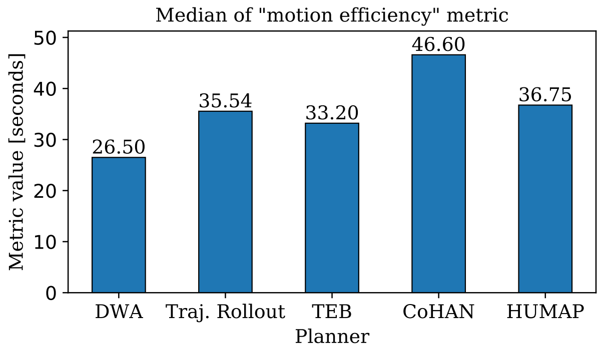
(a) m_{goal} in the *crossing* scenario



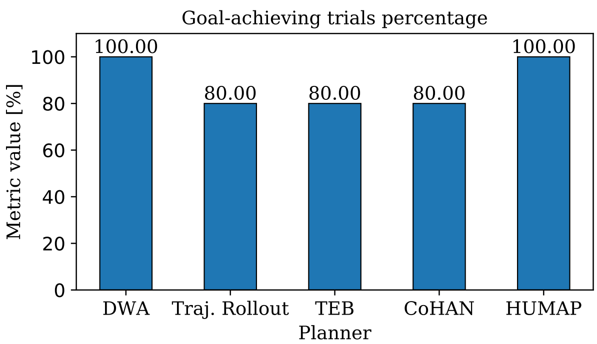
(b) m_{mef} in the *crossing* scenario



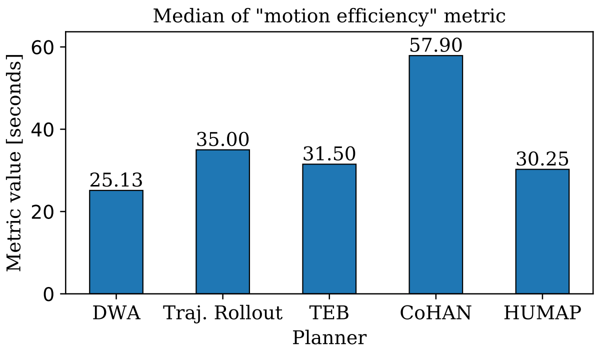
(c) m_{goal} in the *overtaking* scenario



(d) m_{mef} in the *overtaking* scenario



(e) m_{goal} in the *passing* scenario



(f) m_{mef} in the *passing* scenario

Figure 7.15: Navigation success rates, m_{goal} (a, c, e), and motion efficiencies, m_{mef} (b, d, f), obtained with various trajectory planners in each scenario of the real-world experiments.

of individuals (discussed further in this section).

In terms of maintaining the gap from environment objects (either typical obstacles or humans) the *CoHAN* and *TEB* planners keep the clearance the biggest in most cases. On the other hand, *DWA* and *Trajectory Rollout*, which, due to the similarities, perform comparably in terms of the ${}_{\text{min}}m_{\text{obs}}$ metric and keep the least clearance. In particular, the mobile base operating under *DWA* tends to pass obstacles at very close distances but avoids collisions effectively. However, this is not necessarily the characteristic of the *DWA* method itself, but a standard tuning of its scoring function, which is focused on task performance.

The differences between other analytical planners are rather subtle, and all keep the safe margins except the *HaTEB* in the *overtaking* scenario, where collisions occurred. This is also the main reason why *HaTEB* was disqualified from real-world trials despite reasonable performance in other scenarios. Since the ${}_{\min}m_{\text{obs}}$ metric focuses on distances from all environment objects, the proposed *HUMAP* planner, concentrating on social aspects of navigation, should be mainly assessed from the ${}_{\max}m_{\text{psi}}$ metric's perspective. Nevertheless, the *HUMAP*'s fulfilment of the collision avoidance requirement has been verified both in simulation and real-world studies.

Due to the robustness issues of *DRL*-based algorithms, they are analysed in separation from other methods, as the Tab. 7.5 contains metric values calculated on the basis of all trials (instead of only successful). Notably, the *GA3C-CADRL* algorithm avoided collisions achieving ${}_{\min}m_{\text{obs}}$ of 0.58 m in *crossing*, 0.64 m in *overtaking*, and 0.43 m in the *passing* scenario. Another method that kept respectable clearance from obstacles in the *SARL** approach. On the contrary, *DRL-VO* had difficulty maintaining enough clearance in narrow corridors and sharp corners of all scenarios achieving ${}_{\min}m_{\text{obs}}$ of 0.28 m.

Due to multiple navigation failures of *DRL*-based algorithms in the designed scenarios, and since they rarely reached the stages where human-robot interaction occurs (Fig. 7.11, 7.12, and 7.13), other metrics of the *RL*-based planners will not be discussed.

Motion efficiency and path characteristics The motion efficiency metric, m_{mef}, depicts how much time the planners need to command the robot until the goal pose is reached (Fig. 7.15b, 7.15d, and 7.15f). It is correlated with the characteristics of paths undertaken during the subsequent trials, e.g., path lengths (Euclidean), m_{plin}, and path irregularities (cumulative heading change), m_{chc}; therefore, those are jointly discussed.

The traditional *DWA* outperformed other planners in the *overtaking* and *passing* scenarios (in real world and simulation) in terms of time needed to reach the goal pose – represented by m_{mef}. Especially in the *overtaking* scenario, the gap between the fastest and the second fastest planner (*HUMAP* in the simulation and *TEB* in the real world) is considerable. The highest motion efficiency corresponds with shortest path lengths, m_{plin}, executed by the *DWA* (best results in the *overtaking* and *passing* scenarios). While the *Trajectory Rollout* obtained path lengths comparable to *DWA*, the times needed to reach the goal pose in the *overtaking* and *passing* scenarios are approximately 30% longer than *DWA*'s, which is related to increased path irregularity, m_{chc}, and in-place rotations, m_{iprot}. Moreover, *DWA*'s scores of m_{mef} and m_{plin} are highly correlated between different scenarios, which indicates that the planner performed minimal heading changes throughout experimental scenarios.

An important aspect is related to the traditional *Elastic Bands* planner, which in all scenarios reached the goals with path lengths (m_{plin}) that were not the longest but required substantially more time to complete the tasks compared to other planners (its m_{mef} scores are 28-104% higher across scenarios than any other planner's). This is mainly related to the distinctive property of the *Elastic Bands* planner, which commands the robot with limited translational actions but numerous erratic heading changes along the way. The latter is reflected by the path irregularity values, m_{chc}. The issue is particularly visible in the *overtaking* scenario, where *Elastic Bands*'s m_{chc} score is approximately 200% higher than the second worst rating (of the *SRL-EBand* approach). Another algorithm that commands the mobile base with excessive heading changes is the human-aware *HaTEB* planner and this attribute has also been identified in the qualitative analysis (Sec. 7.4.3).

Furthermore, a general feature observed in the planners originating from the *Elastic Bands* concept is that they perform more heading changes than the others, i.e., traditional *DWA*, *Trajectory Rollout*, and human-aware *HUMAP*. Among the *Elastic Bands*-based methods, the traditional *TEB* and human-aware *CoHAN* obtain the smallest m_{chc} rates. Despite reasonable scores in that matter, the motion efficiency of the *CoHAN* planner is drastically degraded in the *overtaking* and *passing* scenarios compared to the *TEB*.

Specifically, in many trials, the *CoHAN*'s generation of the trajectories produced numerous forward-backwards motions (reflected by the m_{bwd}) near the corners or when trying to reach target poses. It is most likely caused by the *CoHAN*'s human-aware optimisation constraints that extend the ones implemented in the *TEB* planner (from which the *CoHAN* is derived). The additional parameter tuning has already been performed before the first phase of the experiments (Sec. 7.3) according to the authors' guidelines[7] but helped to the limited number of scenarios. In some attempts of the *passing* scenario, the *CoHAN* planner produced oscillating motions trying to rotate to the goal orientation just before approaching the goal pose that was located along the corner; thus, its prolonged m_{mef} times. Another typical situation when *CoHAN* did not work as intended is navigating through narrow passages that are created in the costmap (the environment model used by the algorithm) during the *overtaking* scenario. In this case, the robot aims to overtake slowly moving humans, whose postures limit the space available for passing through.

Overall, it is understandable that human-aware trajectory planners (*SRL-EBand*, *HaTEB*, *CoHAN*, and *HUMAP*) might require longer times to reach goals (m_{mef}) than traditional methods, as human-aware algorithms need to perform extra manoeuvres to avoid humans in a socially-acceptable manner. Extended times needed to reach the goal are the trade-off for introducing human-awareness constraints into trajectory planning and tuning objective functions to favour less disruption to surrounding humans. However,

[7] `https://github.com/sphanit/CoHAN_Planner/issues/2`

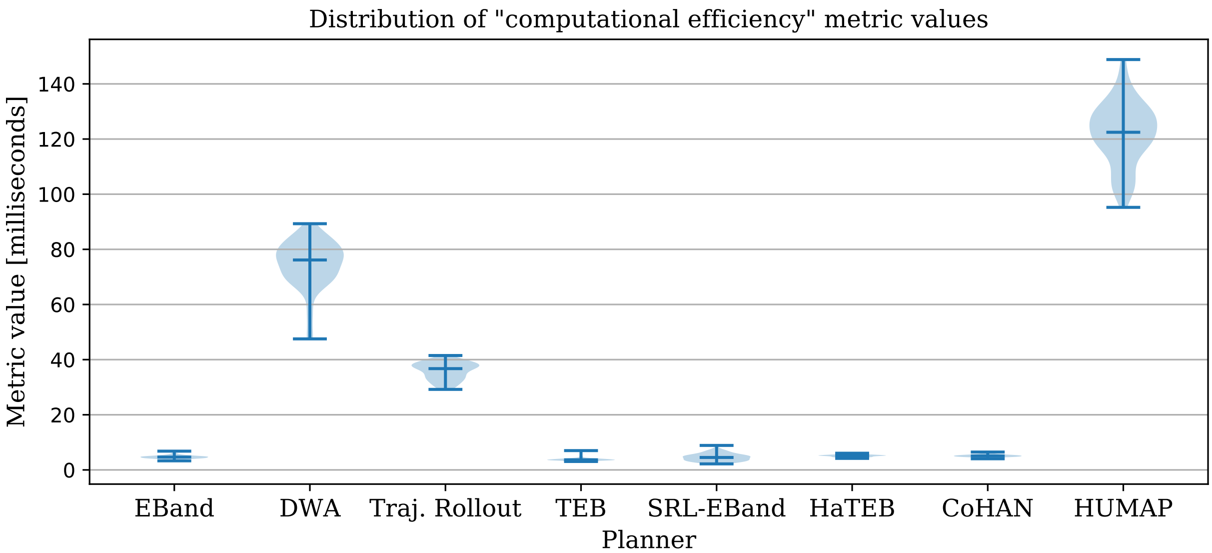


Figure 7.16: Mean computation times, m_{cef}, obtained throughout the simulated *crossing* scenario with various trajectory planners.

in the *crossing* scenario, the *HUMAP* planner achieved results comparable or better to other methods in terms of m_{mef}. This arises from the *HUMAP*'s environmental awareness and its capability to anticipate and handle "crossing" situations using a dedicated behaviour of its *FSM*. Moreover, in the *passing* scenario, only the *DWA* outperformed the *HUMAP*, whereas, in the *overtaking* scenario, *HUMAP* completed navigation tasks significantly faster compared to other human-aware trajectory planners.

The shortest paths (m_{plin}) of the *HUMAP* in the *crossing* scenario correspond to its motion efficiency scores, m_{mef}, which are comparable to the best-performing planners in this matter.

Computational complexity The computational complexity of various planning algorithms can be assessed with the *SRPB*'s m_{cef} metric, which reflects the average computation time the planner takes to develop a new velocity command. Fig. 7.16 illustrates the distribution of metric values for extensive simulation experiments (*crossing* scenario example). A complementary metric – computational time repeatability, m_{cre}, determines how much computation times differ from the mean value throughout the experiment. The latter aims to identify whether certain algorithms exhibit increased computational complexity, e.g., when humans are present in the environment.

The important aspect of the m_{cef} metric analysis is that the simulation experiments were conducted on different hardware than the real-world tests, as noted in Sec. 7.1. Therefore, the values of m_{cef} and m_{cre} metric presented in the Tab. 7.4 cannot be directly compared between simulated and real-world experiments, exceptionally.

In terms of computational complexity, the examined planners can be classified into three groups. The first group constitutes the *Elastic Bands*-based methods, i.e., *Elastic Bands*, *TEB*, *SRL-EBand*, *HaTEB*, and *CoHAN*. Those approaches exhibit the smallest computational complexity, needing up to 10 ms to compute a new velocity command. Notably, in all scenarios, the *TEB* outperforms other approaches from this category, but its human-aware specialisations are also computationally efficient. The highest rating comes from the fact that the optimisation framework implemented in *TEB* is capable of parallelising the calculations. Additionally, the *SRL-EBand*'s scores are at the level of the classical *Elastic Bands*.

The second group of planners includes algorithms constituting considerably higher computational burden compared to the first group, i.e., *DWA* and *Trajectory Rollout*. For example, *DWA* requires 76-90 ms with the simulated and 119-158 ms with the real-world setup to compute a new trajectory. On the other hand, *Trajectory Rollout* – approximately 37 ms in simulation and 74-82 ms in real-world.

The *HUMAP* achieved by far the longest computation times among all tested planners; hence, it can be classified into a separate "computation times group". Specifically, the *HUMAP*'s single computation cycle in a simulation reaches approximately $m_{\text{cef}} = 150$ ms. The fundamental parameters influencing the computational complexity of the proposed planner are the number of produced trajectories (all trajectories are scored in each step; hence, the duration of the scoring stage is also affected), the planning/prediction horizon (t_{hor} set to 2 s for the experiments), and the number of environment objects considered in the model-based trajectory generator (pairwise interaction forces between the robot and environment objects must be computed; limited to five closest obstacles and four closest humans). However, the m_{cef} metric reflects the mean value of a computation cycle, while the *HUMAP* operates according to the multiple behaviours associated with *FSM* states, which exhibit various computation burdens. The most extensive calculations, i.e., producing trajectory candidates from two generators and scoring all of them with all cost functions, are performed in the behaviour implemented in the *Moving* state. In contrast, the behaviour corresponding to *Orientation Adjustment* state scores only a single feasible trajectory that rotates the mobile base. This computation times diversity also influences the scores of m_{cre} metric.

Notably, the *HUMAP* operated in real-world experiments (on the robot's onboard computer) with a reduced number of trajectory candidates; therefore, has smaller m_{cef} values associated with the real-world experiments compared to the simulated trials. Knowing that the computation loop takes significantly longer on a real robot (based on the *DWA* and *Trajectory Rollout* scores), the planner might have exceeded the desired calculation period if the simulation setup was fully reproduced. Nonetheless, the *HUMAP* is fully capable of real-time operation at 4 Hz on mid-range computers but the parallelisation of

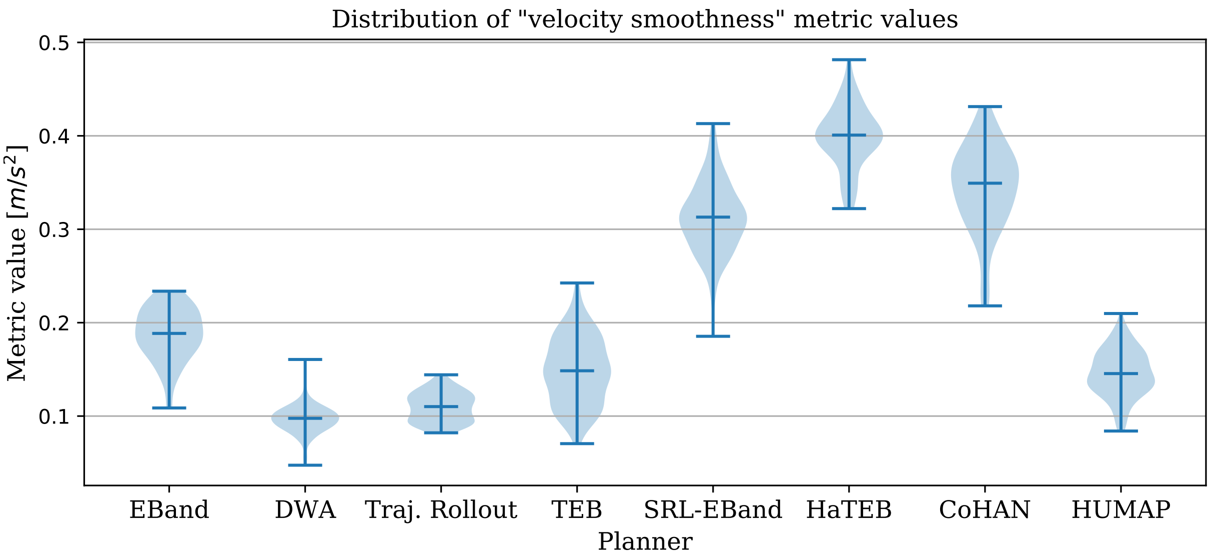


Figure 7.17: Velocity smoothness values, m_{vsm}, obtained throughout the simulated *overtaking* scenario with various trajectory planners.

calculations poses an interesting future development perspective.

In general, the computation times of each examined planner did not dramatically differ across scenarios, as well as their computational time repeatability scores, except for the *overtaking* tests where increased computational burden is observed in simulation results amid all tested methods. Data used for assessing the m_{cef} is not affected by the performance of the simulator (so-called "simulation time" is used in virtual experiments); therefore, it can be explained by more demanding calculations of other routines functioning within the same process of an operating system as a global path planner and a local trajectory planner (the ecosystem explained in Sec. 6.3).

7.4.7 Robot motion naturalness

The naturalness of the robot's motion is usually related to maintaining a smooth velocity profile and avoiding erratic movements and oscillations (Sec. 2.4.1). The concepts of the robot's motion naturalness correspond to multiple social robot navigation requirements that can be verified using the *SRPB*'s metrics.

The main indicator of the presence of erratic motions is a lack of smoothness in the robot's velocity profile (**Req. 2.3.1.1**), which can be assessed with metrics expressing the scale of robot accelerations – translational, m_{vsm}, and rotational, m_{hsm}. A high correlation between the distributions of the m_{vsm} as well as m_{hsm} values is observed across scenarios; therefore, metric values for only a single scenario are presented with the violin plots (Fig. 7.17 and 7.18). Nonetheless, median values of each scenario are identified in Tab. 7.4.

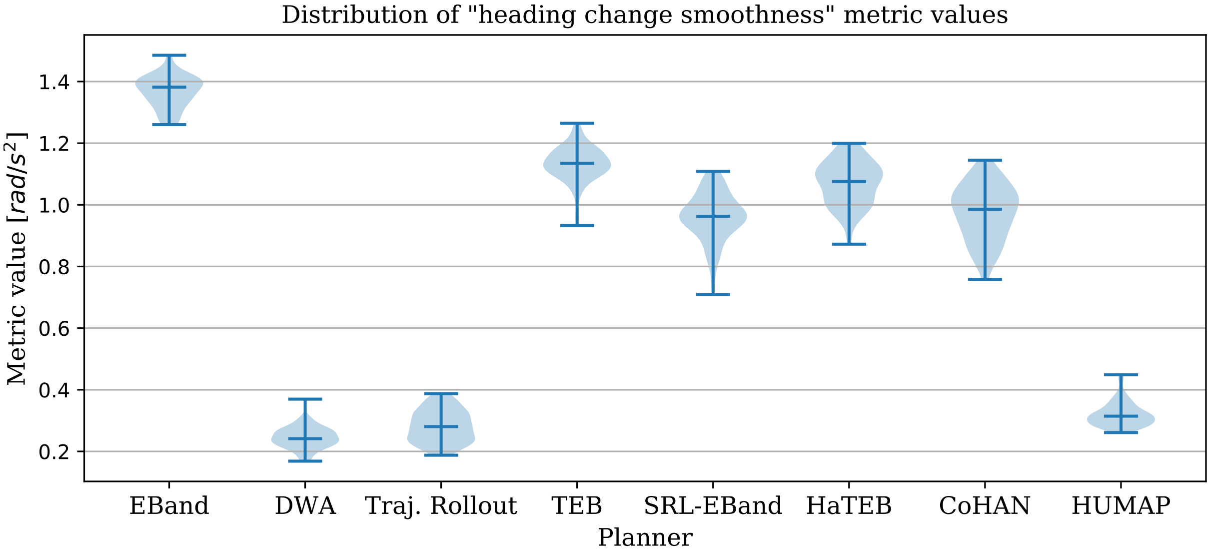


Figure 7.18: Heading change smoothness values, m_{hsm}, obtained throughout the simulated *overtaking* scenario with various trajectory planners.

The *DWA*, *Trajectory Rollout* and *TEB* planners performed the best in terms of the translational velocities' smoothness (m_{vsm}), with *HUMAP*'s rating being very close to them, in several cases outperforming *TEB* and by far outdoing other human-aware algorithms. Nevertheless, *TEB*-based planners obtained significantly more rough rotational velocity profiles compared to *DWA* and *HUMAP*. The outcomes of the smoothness of rotational velocity changes (represented by m_{hsm}) are correlated with the results of m_{chc}, as both metrics regard the change in the robot's orientation throughout the scenario. Again, traditional *DWA* and *Trajectory Rollout* planners achieved the best m_{hsm} results in all scenarios; however, the scores of *HUMAP* are 20-50% higher, while the second best-performing human-aware trajectory planner (*SRL-EBand*) has those measures 175-300% higher. Excessive heading changes of *Elastic Bands* (discussed earlier) are also visible in the m_{hsm} metric ratings.

The *HUMAP* outperforms other human-aware planners in terms of velocity smoothness scores but obtains slightly inferior results compared to the traditional *DWA* and *Trajectory Rollout*. This is a consequence of its passing speed discomfort soft constraint (${}^{\mathbb{H}}\text{cfun}_{\text{psd}}$), which requires the algorithm to adjust the commanded velocities in proximity to humans to reduce their discomfort that, in turn, causes additional accelerations captured by the metrics.

Other requirements in the category of the robot's motion naturalness assess in-place rotations (**Req. 2.3.1.3**) (m_{iprot}), backward movements (**Req. 2.3.1.4**) (m_{bwd}), and oscillating motions (**Req. 2.3.1.2**), which manifest as alternating forward and backward motions (m_{osc}).

The *HUMAP* has increased m_{iprot} metric scores by design, as it relies on in-place rotation to reach the goal orientation if it does not align with the direction of the final part of the path. The same applies to the implementations of *DWA* and *Trajectory Rollout* that were used during the experiments. On the other hand, *TEB*-based planners (traditional *TEB* and human-aware *HaTEB* and *CoHAN*) smoothly adjust the trajectories near the goal pose to reach the goal position and orientation at the same time (if feasible). Nevertheless, increased m_{iprot} scores indicate that certain algorithms could not progress further throughout the scenario and performed in-place rotations as a recovery action, which happens in *Trajectory Rollout*, *SRL-EBand*, and *Elastic Bands*.

Other indicators of a lack of progress throughout certain parts of a scenario are the m_{osc} and m_{bwd} metrics. The highest measure of m_{bwd} has been achieved by the *CoHAN* planner, which identifies the cause of its longer times required to reach the goal (m_{mef}). This is related to the problem of traversing narrow passages and near the corners, as explained while describing the motion efficiency of various algorithms. In contrast, the *HUMAP* obtained a non-zero score regarding backward movements only in the *crossing* scenario, which indicates that the planner operated in the *Look Around* state (slow backing up motion to update the environment model).

On the other hand, all examined planners (those mentioned in the Tab. 7.4) have non-zero measures concerning oscillating motions in all scenarios. While the *TEB* planner has those scores the lowest, the *HUMAP* outperforms other human-aware trajectory planners in that matter.

7.4.8 Perceived safety of humans

Metrics implemented in the *SRPB* allow assessing human perceived safety (**Req. 2.2**) while the robot operates with different planning algorithms. Specifically, indicators relevant to the performed experiments evaluate the scale of the robot's intrusions into personal spaces, its motion legibility, and discomfort caused to humans by the robot's passing speed during unfocused interaction. Notably, the maximum values of metrics are selected for the analysis (${}_{\max}m_{\text{psi}}$, ${}_{\max}m_{\text{dir}}$, and ${}_{\max}m_{\text{psd}}$), as they turned out to better replicate the impressions of participants during the real-world experiments compared to the mean values of metrics computed based on entire trials (m_{psi}, m_{dir}, and m_{psd}).

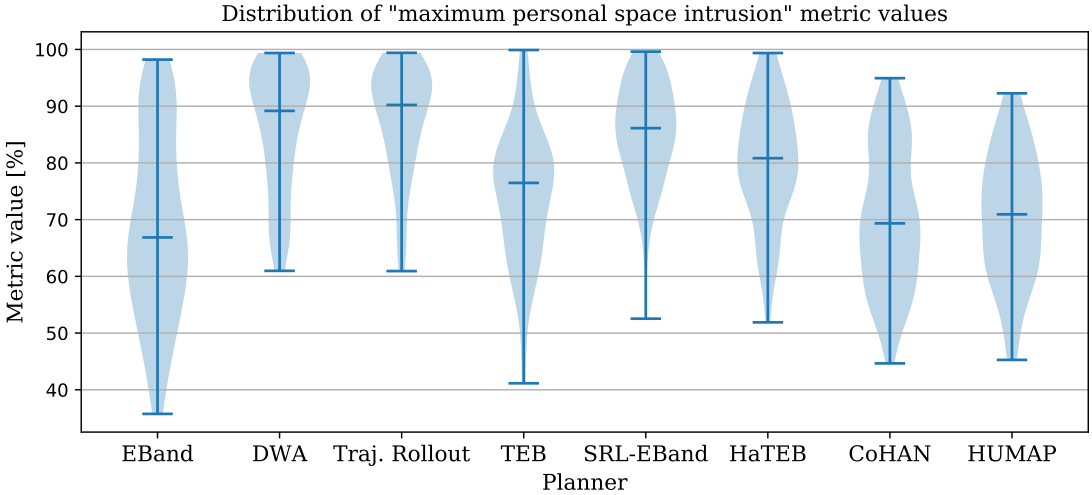
Respecting proxemics zones (**Req. 2.2.1**) is the fundamental concept regarded in algorithms designated for social robot navigation. In the simulated *crossing* scenario of the conducted experiments, the smallest maximum personal space intrusion score, ${}_{\max}m_{\text{psi}}$, was achieved by the *Elastic Bands* method. However, the *Elastic Bands*'s seemingly good performance is caused by the inadequate motion efficiency of the planner. In particular, in the *crossing* scenario of the presented controlled study, humans aim to move similarly in all trials with diverse planners. To accomplish that, a threshold distance between the robot and a human, at which a human starts the task of waypoints following, has been fixed. Therefore, slow progressive movements of the *Elastic Bands* (m_{mef}) enable the commanded robot to avoid the "crossing interaction". This phenomenon is visible in Fig. 7.11a. As a consequence, deceptively promising ${}_{\mathrm{max}}m_{\mathrm{psi}}$ measure of the *Elastic Bands* planner should not be considered. Instead, this method is low-rated in the *overtaking* and *passing* scenarios, where unavoidable frontal encounters occur.

In contrast, the proposed *HUMAP* planner achieved the best or the second-best results in all scenarios (concerning ${}_{\mathrm{max}}m_{\mathrm{psi}}$), being outperformed only by the *CoHAN* approach in the real-world *overtaking* scenario by 0.52 pp. and by the *HaTEB* in the simulated *passing* scenario by 2.16 pp. The distributions of the maximum personal space intrusions metric (calculated per entire trial of the extensive virtual study) obtained from 100 tests of each scenario (all planners) are presented in Fig. 7.19. Importantly, the *HUMAP* also acquired the best results of the ${}_{\mathrm{max}}m_{\mathrm{psi}}$ metric in most scenarios conducted in the real world, as illustrated in Fig. 7.20a (*crossing*), Fig. 7.20c (*overtaking*), and Fig. 7.20e (*passing*).

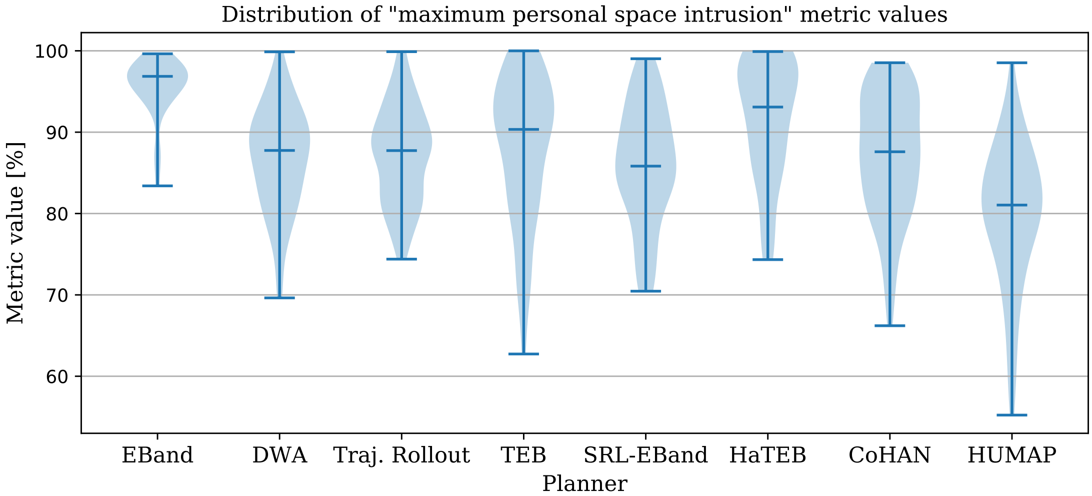
Another crucial aspect of the safety perceived by humans interacting with robots is the motion legibility of autonomous mobile bases. Namely, it can be quantitatively assessed with the metric that evaluates the certainty of heading straight into a human by a robot, especially with decent speed (**Req. 2.2.4**). A relevant metric has been implemented in the *SRPB*, identified as ${}_{\mathrm{max}}m_{\mathrm{dir}}$, which is useful for evaluating whether a planning approach is capable of adjusting the robot's heading as soon as it detects such an agent.

In two cases of the simulation study, the *Elastic Bands* algorithm acquired the finest metric results regarding ${}_{\mathrm{max}}m_{\mathrm{dir}}$. However, the indicator captures the *FOV* of humans and the planner commanded the robot so it moved mainly behind both humans. Hence, again, seemingly good results were obtained by that traditional method. Excluding the outcomes of *Elastic Bands* from the results, the *HUMAP* planner performed the best in terms of ${}_{\mathrm{max}}m_{\mathrm{dir}}$ in all scenarios both in simulated (Fig. 7.21) and in the real-world trials (Fig. 7.20b – *crossing*, 7.20d – *overtaking*, and 7.20f – *passing*). Among traditional local trajectory planners, the best results in most cases were obtained by the *TEB*, which outperformed its human-aware specialisations – *HaTEB* and *CoHAN*, but mainly in the virtual experiments. In contrast, the robot operating with the *DWA* planner had difficulties adjusting the heading early once a human was detected (${}_{\mathrm{max}}m_{\mathrm{dir}}$), which might have induced considerable fear in surrounding pedestrians.

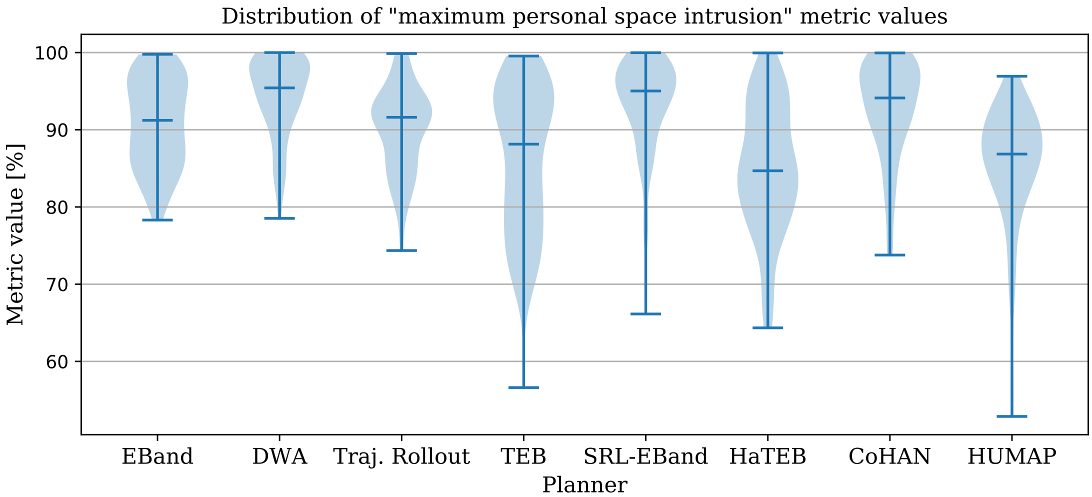
A different concept considered within the perceived safety of humans relies on adjusting the robot's speed once passing an individual (**Req. 2.2.3**). In the *crossing* scenario seemingly good ${}_{\mathrm{max}}m_{\mathrm{psd}}$ rating has been achieved by the *Elastic Bands*, similarly as with the previously discussed human awareness metrics. Besides the *Elastic Bands*, the best results have been acquired by the *HUMAP* in the *overtaking* scenario, and by the *HUMAP*



(a) *crossing* scenario

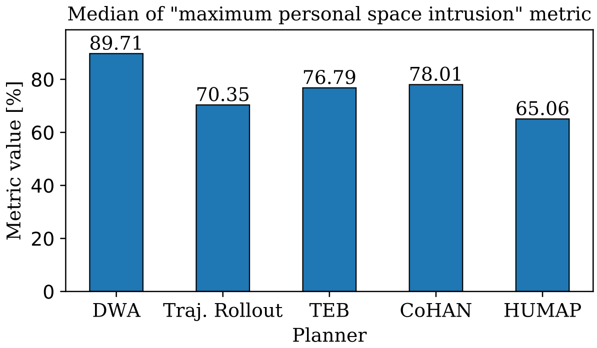


(b) *overtaking* scenario

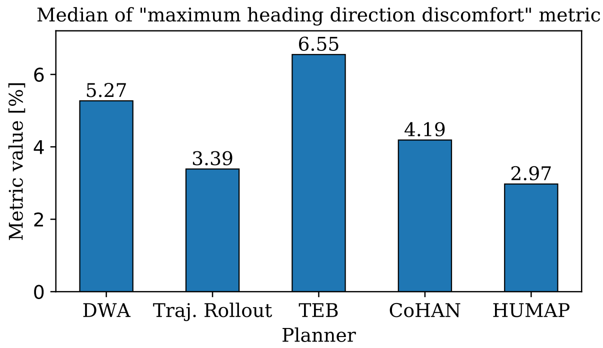


(c) *passing* scenario

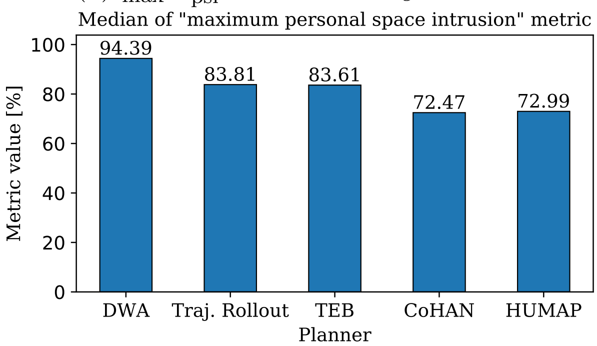
Figure 7.19: Distribution of maximum personal space intrusion (${}_{\max}m_{\text{psi}}$) metric values in various scenarios in the simulation environment.



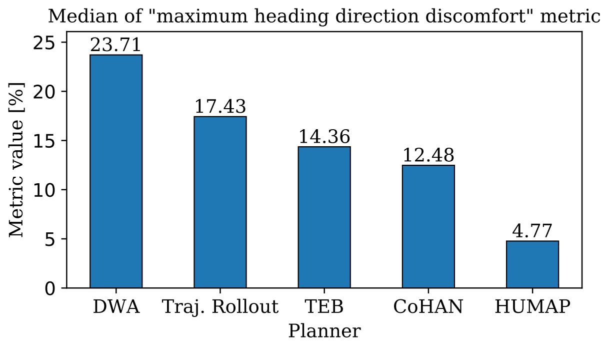
(a) ${}_{\max}m_{\mathrm{psi}}$ in the *crossing* scenario



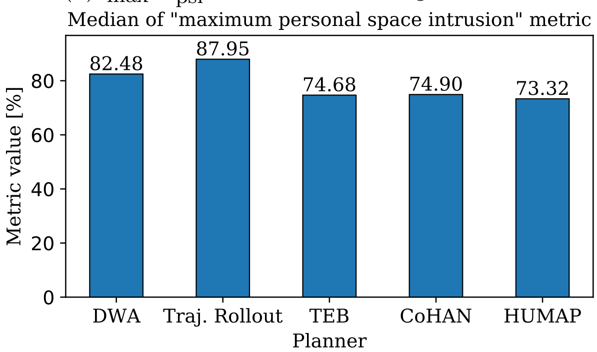
(b) ${}_{\max}m_{\mathrm{dir}}$ in the *crossing* scenario



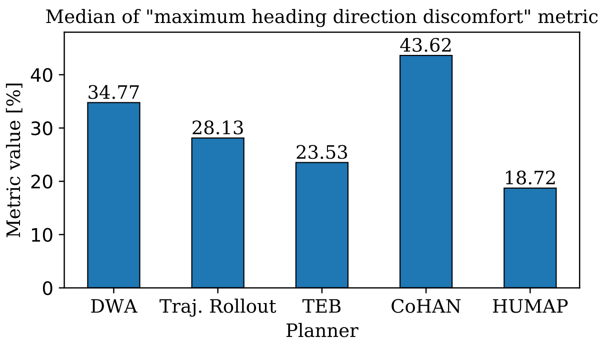
(c) ${}_{\max}m_{\mathrm{psi}}$ in the *overtaking* scenario



(d) ${}_{\max}m_{\mathrm{dir}}$ in the *overtaking* scenario



(e) ${}_{\max}m_{\mathrm{psi}}$ in the *passing* scenario



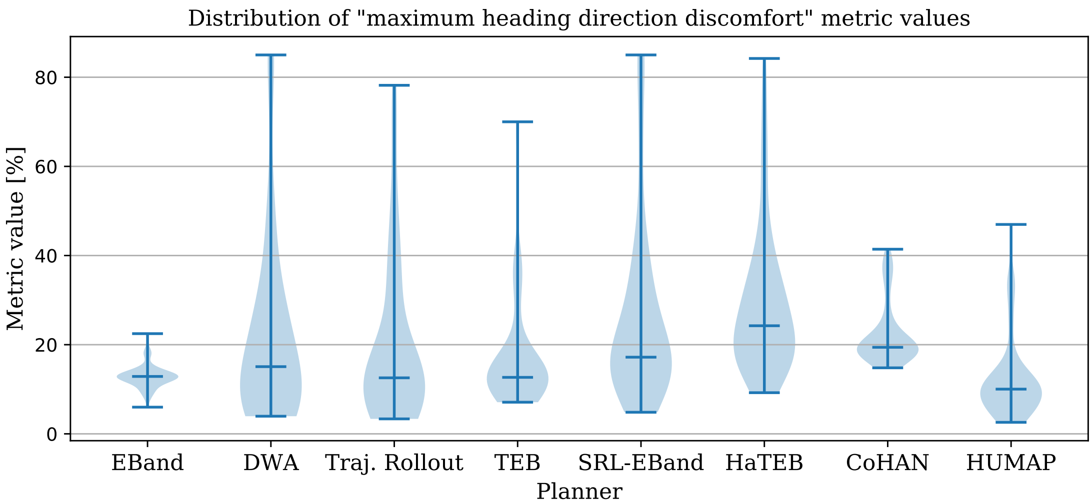
(f) ${}_{\max}m_{\mathrm{dir}}$ in the *passing* scenario

Figure 7.20: Maximum personal space intrusions, ${}_{\max}m_{\mathrm{psi}}$ (b, d, f), and maximum heading direction disturbances, ${}_{\max}m_{\mathrm{dir}}$ (a, c, e), obtained with various trajectory planners in each scenario of the real-world experiments.

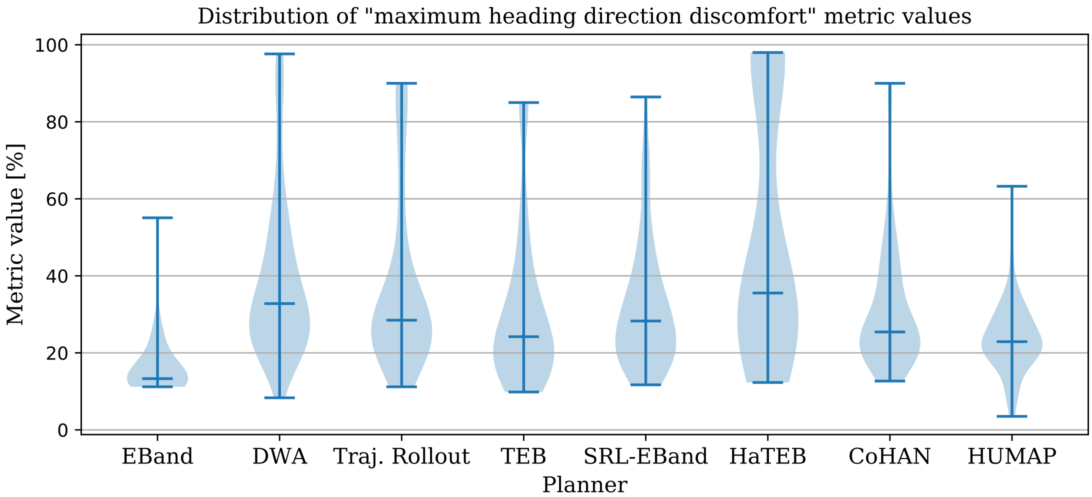
and *CoHAN* in the *crossing* setup. Interestingly, in the *passing* scenario, the *TEB* outperformed other planners both in the simulation and the real world, achieving approximately 2-2.5 pp. better scores compared to the *HUMAP*'s outcomes. Distribution of the ${}_{\max}m_{\mathrm{psd}}$ metric values obtained by various planners is presented in Fig. 7.22.

7.4.9 Summary

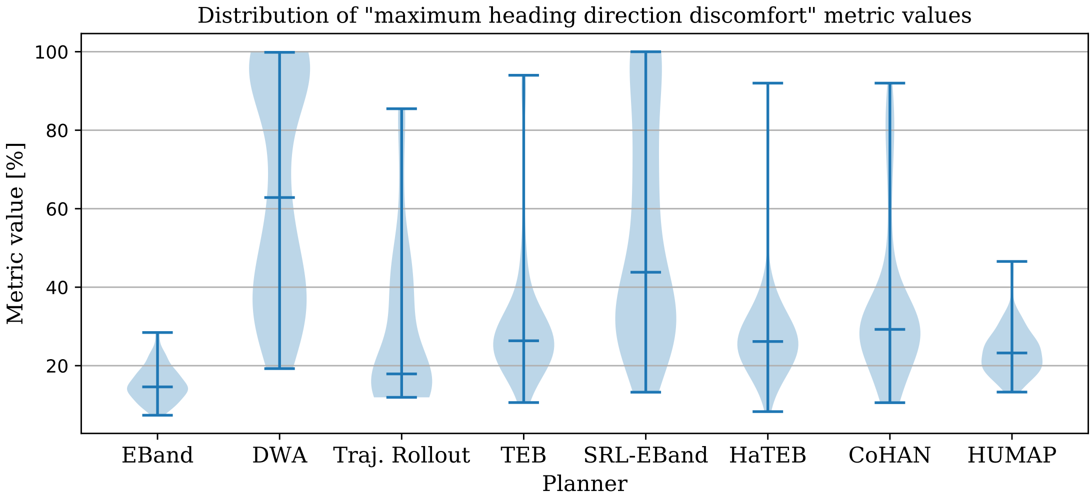
The comparative study described in this section aims to evaluate the proposed *HUMAP* local trajectory planner with traditional methods as well as state-of-the-art human-aware



(a) *crossing* scenario

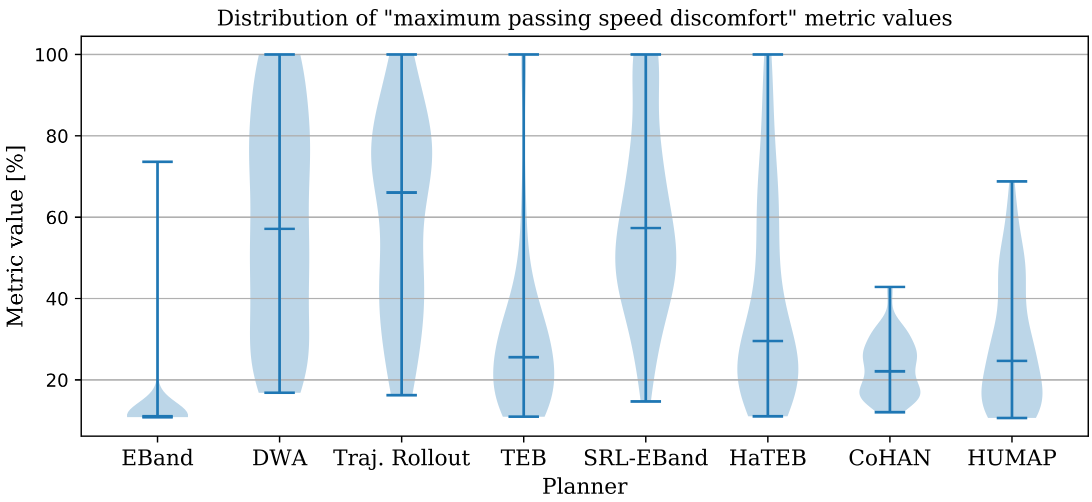


(b) *overtaking* scenario

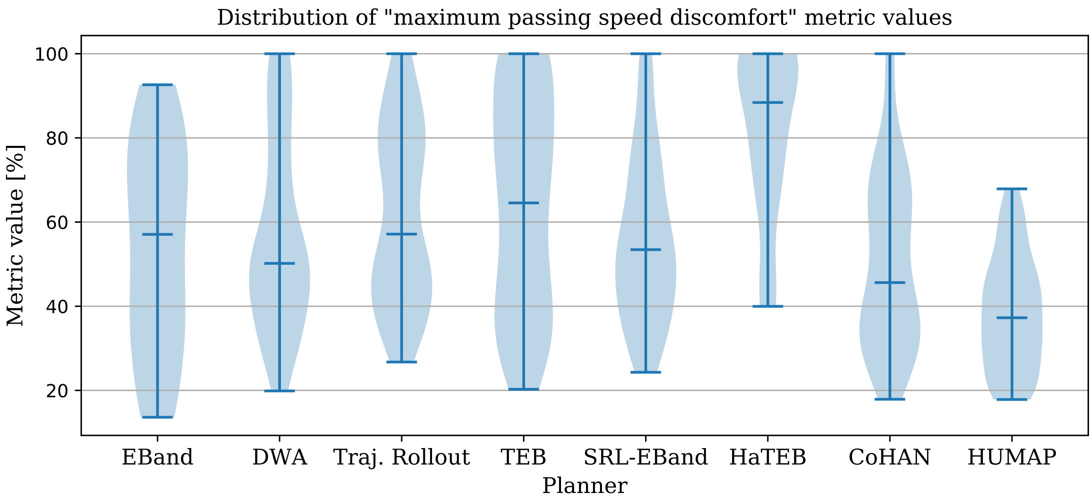


(c) *passing* scenario

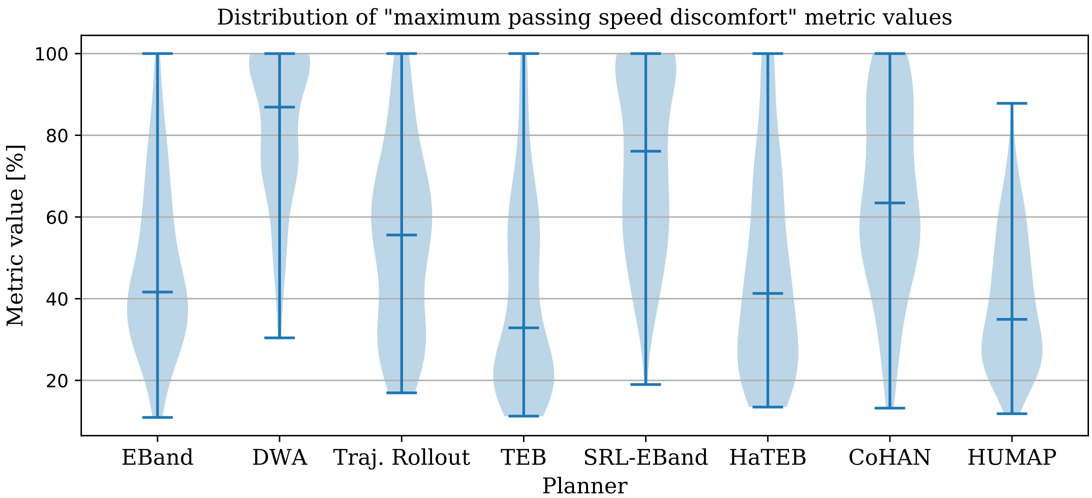
Figure 7.21: Distribution of maximum heading direction disturbance $({}_{\max}m_{\mathrm{dir}})$ metric values in various scenarios in the simulation environment.



(a) *crossing* scenario



(b) *overtaking* scenario



(c) *passing* scenario

Figure 7.22: Distribution of maximum passing speed discomfort ($_{\max}m_{\text{psd}}$) metric values in various scenarios in the simulation environment.

planners.

To assess the versatility of the examined approaches, various scenarios have been designed based on the standardised protocols [10] for robots operating in populated dynamic environments and subjected to unfocused interactions with humans. The three developed scenarios evaluate different types of encounters, i.e., *crossing*, *overtaking*, and *passing*. Each scenario has been conducted both in the real world and in the virtual equivalent of the target environment according to the controlled study principles to enable direct comparison of results. The trajectory similarity of two human participants has been ensured in each trial.

The assessment involved both traditional local trajectory planners, i.e., the *Elastic Bands*, *DWA*, *Trajectory Rollout*, and *TEB*, as well as human-aware algorithms – *SRL-EBand*, *HaTEB*, *GA3C-CADRL*, *SARL*, *SARL**, *RG's DRL*, *CoHAN*, *DRL-VO*. All these were compared with the proposed socially-aware *HUMAP* in the extensive simulation tests (100 trials have been performed in each scenario with every planner) and in the real-world study involving the most robust methods. Several planners have been also tested in the first phase of the experiments (described in Sec. 7.3) and, overall, their qualitative characteristics and quantitative performance have been confirmed in additional scenarios.

Both qualitative (Sec. 7.4.3) and quantitative evaluation (Sec. 7.4.4) of the examined trajectory planning algorithms have been extensively analysed based on the data collected during both real-world and simulation tests. Qualitative assessment regards mainly the distinctive characteristics of planning algorithms' behaviours and their trajectories obtained in different trials. In contrast, the quantitative analysis has been performed based on the metrics developed in the Social Robot Planner Benchmark (Chapter 4), which implements various navigation task performance indicators, but importantly, metrics evaluating human discomfort caused by a navigating robot. The usage of *SRPB* has been crucial in the comprehensive quantitative evaluation of multifaceted concepts of social robot navigation.

The analysis of the robustness of the planners revealed that local trajectory planners formulated analytically substantially outperform state-of-the-art *Deep Reinforcement Learning* methods. The most robust among the *DRL*-based approaches was the *DRL-VO*. The robot commanded by this algorithm completed approximately 40% of trials in the *crossing* scenario. In contrast, other approaches failed to succeed in any attempt (Fig. 7.14). Since investigated *DRL*-based methods have been successfully tested by the authors, the main reason for the limited robustness is the lack of generalisability to various environments. All learning-based algorithms were functioning with the robot using the original policies provided by the authors of the methods to avoid the influence of underfitting or overfitting the algorithms when learning from scratch.

On the other hand, the robustness of the most reliable analytical methods reaches 90-100% across all scenarios (Fig. 7.14). Nonetheless, amid this class of planners, several approaches are of limited practicality due to the slow progressive movements and erratic heading changes (*Elastic Bands*), or excessive angular accelerations (*HaTEB*). The extensive analysis of *SRPB* metrics enabled to comprehensively assess the performance of examined algorithms in terms of both classical (**Req. 1**) as well as social robot navigation requirements (**Req. 2**).

The results of the comparative study depict that the proposed *HUMAP* planner achieved the best robustness scores among all tested approaches, completing 100% of all trials, both in the simulation as well as in the real world. The robustness of the *HUMAP* is provided by the analysis of the current environmental context and supplying this information to the internal *Finite State Machine* that orchestrates the planner's state transitions. Multiple behaviours are implemented in the *FSM*, which adapts the robot for operating in partially observable (due to the limited *FOV* and range of robot sensors) dynamic environments. Particularly, the special *HUMAP* behaviours perform reactive navigation (with collision checking) until planning under typical circumstances, e.g., with a valid global path and the robot's footprint far from obstacles can be continued.

Furthermore, in most "scenario–environment" (real world or simulated) configurations, the *HUMAP* planner performed better than other methods in terms of the study-based social navigation criteria approximating the human discomfort metrics (implemented in the *SRPB* benchmark) without significant degradation in the navigation task-oriented performance (e.g., motion efficiency, path length, path irregularity).

The *HUMAP* planner integrates a hybrid trajectory candidates generation method with various cost functions forming the objective function and exposes numerous parameters to the user for system configuration. The *HUMAP* parameters (Sec. 6.9) have been selected by experimentally achieving a compromise between task performance, social aspects of robot navigation, and computational complexity. Specifically, the number of produced trajectory candidates substantially affects the computational burden of the method. The computation times of the proposed method have been by far the longest among all tested algorithms (Fig. 7.16), but still allowed stable operation at the frequency of 4 Hz.

The extensive analysis of the quantitative results of the *SRPB* metrics emphasises the need for analysing different algorithms holistically. This means that any human-aware trajectory planner should not be examined concerning only the maximum personal space intrusions metric regardless of, e.g., motion efficiency and heading changes along the path. For example, the human-robot interactions that occur in trials with planners that develop noticeably smaller translational motions than the others, do not exactly reflect the cases observed with more reliable planners. Particularly, the environmental layout along with an unintended scenario progression may favour generating slowly-moving velocity commands. Such a situation occurred, for instance, in the *crossing* scenario, where humans begin enacting their experimental scenarios based on the distance between them and a robot. Therefore, human intrusions have often ended before the robot operating with the *Elastic Bands* planner reached the challenging stage ("crossing interaction") of a scenario (Fig. 7.11).

Developing human-aware robot navigation algorithms is an enormous challenge. Due to the contradictory nature of traditional and social robot navigation requirements, adding human-aware constraints to optimisation-based strategies is not straightforward, as visible in the varying outcomes related to human awareness (reflected by the *SRPB* metrics) of, e.g, *HaTEB* and *CoHAN*, which are in several cases substantially outperformed by the traditional *TEB*, which they originate from.

Overall, the conducted experiments, designed without unrealistic assumptions, confirmed the practical aspects of the proposed *HUMAP* approach that effectively handles the partial observability of the environment, as well as human perception uncertainties, which are accounted for in the trajectory scoring procedure. The results of both stages of the experiments provided insights that the previous socially-aware algorithms might not cause substantially less disruption to the surrounding humans compared to the traditional planners. The presented *HUMAP* local trajectory planner mitigates human discomfort during interaction with robots according to the relevant study-based metrics. The outcomes of the experiments show the surpassing performance of the algorithm in terms of human awareness aspects which is consistent across scenarios.

7.5 In-depth analysis of the HUMAP system

Sec. 7.4 discusses the *HUMAP*'s performance against state-of-the-art traditional and human-aware trajectory planners; however, the proposed navigation system can operate in different configurations. Its performance under various parameters is analysed in this section.

7.5.1 Performance with different scoring functions

A set of simulation tests has been performed to assess the sensitivity of the *HUMAP*'s trajectory scoring function (5.31) to the changes of cost functions weights. The "sensitivity" is understood as the impact of manipulating the values of crucial parameters influencing the operation of the *HUMAP*. Therefore, the experiments described in this section aim to

Weights	ρ_{pth}	ρ_{goal}	ρ_{ttc}	ρ_{vsm}	ρ_{hsm}	ρ_{psi}	ρ_{fsi}	ρ_{dir}	ρ_{psd}
baseline	15.0	25.5	3.0	17.0	10.0	30.0	7.5	20.0	10.0
perf.	23.0	33.5	3.0	17.0	10.0	30.0	7.5	20.0	10.0
soc.	15.0	25.5	3.0	20.0	13.0	60.0	7.5	20.0	15.0
no-soc.	15.0	25.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 7.6: Weights of *HUMAP*'s cost functions, $\rho_{(\cdot)}$, applied during the evaluation of different configurations of the planner. The naming scheme is as follows: "baseline" – indicates the reference tuning, "perf." – indicates weights emphasising navigation task performance, "soc." – weights emphasising social aspects, and "no-soc." – weights with negligible social aspects.

depict whether the planner performs reasonably at a considerable range of the objective function's parameters. Those tests have been conducted in the same three scenarios as examined in Sec. 7.4 (the scenarios are described in Sec. 7.4.1), but additional configurations of the *HUMAP* have been tested ten times for each "scenario–weights" arrangement (while the reference *HUMAP* setting has been evaluated a hundred times).

Specifically, the four setups have been tested: reference ("baseline") scoring function tuning (used in the study described in Sec. 7.4), "perf." – modified "baseline" with a focus on task performance (e.g., path length, motion efficiency), "soc." – adjusted "baseline" to emphasise social aspects, and "no-soc." – the "baseline" setup but with zeroed-out weights of cost functions assessing the robot's human awareness.[8] The weights used in each case are identified in Tab. 7.6.

Results of the study are presented in Tab. 7.7. Metrics indicating the performance ("perf.") are the motion efficiency, m_{mef}, or path lengths, m_{plin}, while the social awareness metrics ("soc."/"no-soc.") are reflected by the velocity smoothness, m_{vsm} and m_{hsm}, maximum intrusions into personal spaces, ${}_{\max}m_{\text{psi}}$, heading direction discomfort metric ${}_{\max}m_{\text{dir}}$, and speeds maintained by the robot while passing humans ${}_{\max}m_{\text{psd}}$. Notably, the *HUMAP* planner maintained the 100% success rate (m_{goal}) despite the considerable changes in weights of cost functions. The biggest average clearance from environment objects (m_{obs}) is kept in the configuration emphasising the impact of social cost functions, but the collision avoidance capabilities (${}_{\min}m_{\text{obs}}$) are not affected by the changes in weights. The highest motion efficiency scores (m_{mef}) were obtained in the configuration with zeroed-out weights of human awareness cost functions. Importantly, the robot's

[8]In the implementation of the *HUMAP*, setting the cost function weight of "0" prevents some calculations from being performed (as illustrated in Alg. 3), i.e., the computational burden becomes smaller (improves computational efficiency m_{cef}). However, temporary changes have been applied so all calculations related to cost functions are performed in all circumstances.

speed has been upper-bounded throughout the experiments (${}_{\max}^{r}v_{\mathrm{lin}}$ in Tab. 7.1), but if there were no limits, the m_{mef} could have been enhanced even more in the "no-soc." configuration. The best scores regarding the social awareness metrics were achieved in the majority of cases with the "soc." setup emphasising the social acceptance compared to the reference tuning; however, at the cost of motion efficiency (m_{mef}) degradation.

Metric \ Scenario	Method	*HUMAP*	*HUMAP-(perf.)*	*HUMAP-(soc.)*	*HUMAP-(no-soc.)*
m_{goal} [%]	*C*	100.00	100.00	100.00	100.00
	O	100.00	100.00	100.00	100.00
	P	100.00	100.00	100.00	100.00
${}_{\mathrm{dist}}m_{\mathrm{obs}}$ [m]	*C*	0.63	0.61	**0.64**	0.57
	O	0.59	**0.59**	0.58	0.55
	P	0.56	0.56	**0.59**	0.54
${}_{\min}m_{\mathrm{obs}}$ [m]	*C*	0.34	**0.34**	0.33	0.33
	O	0.33	**0.34**	0.32	0.32
	P	0.31	0.30	**0.32**	0.30
m_{mef} [s]	*C*	31.25	36.37	34.25	**30.75**
	O	32.50	29.12	43.25	**28.62**
	P	29.50	**27.75**	30.87	28.13
m_{plin} [m]	*C*	9.84	10.24	**9.64**	10.29
	O	12.05	**11.77**	12.26	11.96
	P	11.10	11.05	**11.03**	11.28
m_{cef} $[10^{-3}\cdot\mathrm{s}]$	*C*	122.46	**99.82**	132.98	107.46
	O	153.31	146.28	177.30	**124.98**
	P	130.95	127.03	131.07	**108.26**
m_{vsm} $\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$	*C*	0.14	**0.14**	0.14	0.18
	O	0.15	**0.11**	0.18	0.12
	P	0.13	**0.11**	0.13	0.13
m_{hsm} $\left[\frac{\mathrm{rad}}{\mathrm{s}^2}\right]$	*C*	0.34	0.34	**0.32**	0.50
	O	0.31	0.32	**0.31**	0.44
	P	0.33	0.31	**0.30**	0.44
m_{osc} [%]	*C*	3.89	4.31	**3.40**	4.61
	O	**1.68**	2.08	2.08	2.74
	P	1.74	3.20	**1.26**	4.62
${}_{\max}m_{\mathrm{psi}}$ [%]	*C*	70.93	87.43	**62.92**	81.86
	O	**81.04**	84.96	87.17	93.42
	P	86.85	92.38	**84.05**	90.54
${}_{\max}m_{\mathrm{dir}}$ [%]	*C*	10.03	10.10	**9.41**	16.51
	O	22.91	25.27	**18.00**	34.89
	P	23.24	26.23	**19.92**	20.76

Metric \ Scenario	Method	*HUMAP*	*HUMAP*-(*perf.*)	*HUMAP*-(*soc.*)	*HUMAP*-(*no-soc.*)
${}_{\max}m_{\text{psd}}$ [%]	C	24.67	38.88	**11.45**	37.69
	O	**37.26**	52.11	54.83	70.92
	P	34.95	55.75	**28.35**	57.00

Table 7.7: Quantitative results of the performance of the *HUMAP* local trajectory planner operating in various scenarios with different weights of cost functions. The naming convention of the scenarios (performed in simulated environments) is prepared according to the following scheme: C – *crossing*, O – *overtaking*, and P – *passing*. On the other hand, the *HUMAP* planner's configuration naming is as follows: no suffix – indicates standard tuning, "perf." – indicates weights emphasising navigation task performance, "soc." – weights emphasising social aspects, and "no-soc." – weights with negligible social aspects.

The results of these experiments only confirm that a Pareto-optimal solution is being searched for when planning human-aware robot trajectories, as the contradictory objectives must be leveraged in the scoring function (Sec. 5.7). Specifically, the objective function for assessing a candidate trajectory implements opposite criteria, i.e., includes both performance-focused cost functions as well as cost functions mitigating human discomfort. In most cases, using specialised tuning parameters led to obtaining better scores in particular metrics, compared to the standard "baseline" configuration. However, the most interesting comparison involves the "baseline" setup and the human-aware but performance-focused "perf." tuning. The latter relies on enhancing the importance of performance-related components of the objective function while maintaining the weights of social cost functions at the level of the "baseline" setup. Differences in outcomes between those scoring function configurations illustrate that although overall navigation performance has been improved, the scores of social aspects (${}_{\max}m_{\text{psi}}$, ${}_{\max}m_{\text{dir}}$, ${}_{\max}m_{\text{psd}}$) have been degraded. Therefore, the principle of Pareto-optimality is visible, as increasing weight for a certain cost function will likely cause degradation in another aspect.

7.5.2 Analysis of the optimal trajectory selection

The *HUMAP*'s trajectory generation scheme involves producing candidates with two distinctive strategies. Namely, the velocity sampling trajectory generator, gen_{smp}, is used with the focus on exploring the space of feasible controls with uniformly curved trajectories, whereas the model-based generator, gen_{soc}, aims to construct human-like trajectories according to the pedestrian motion model. Based on the previously conducted experiments, the analysis regarding the selection of optimal trajectories can be conducted.

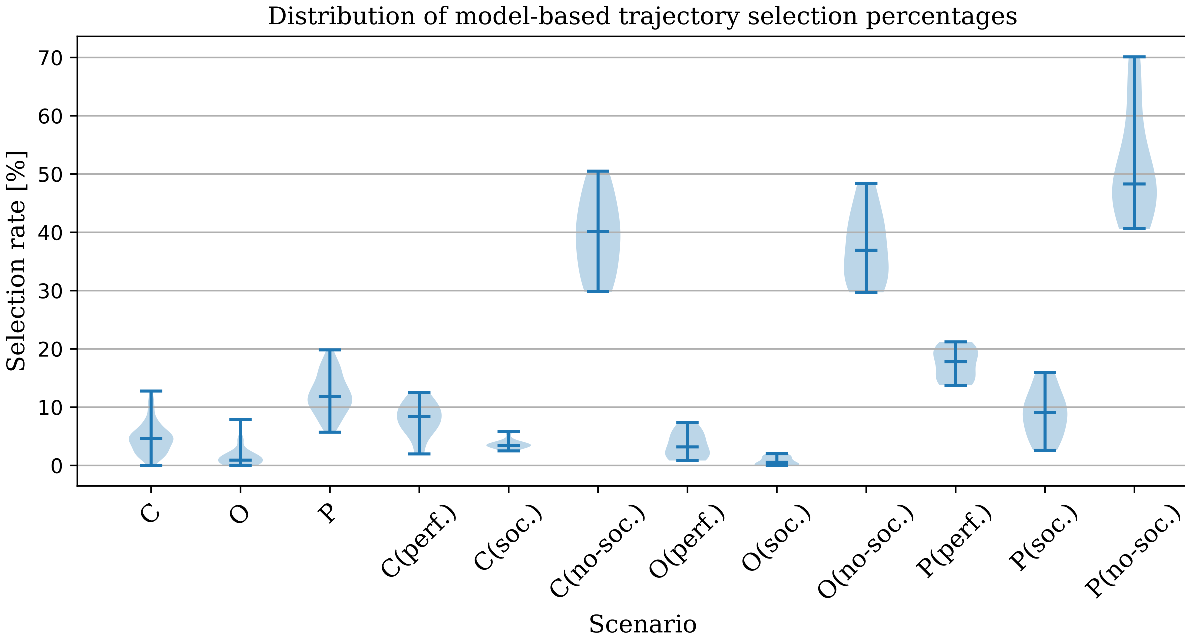


Figure 7.23: Distribution of percentages of optimal trajectories selected from the model-based generator when operating with the *HUMAP* planner. The scenario name mapping is as follows: *C* – *crossing*, *O* – *overtaking*, and *P* – *passing*. Furthermore, the naming convention of the scoring function configurations follows the scheme: no suffix represents the "baseline" tuning, "perf." indicates weights emphasising navigation task performance, "soc." – weights emphasising social aspects, and "no-soc." – weights with negligible social aspects.

The results, shown in Fig. 7.23, describe how frequently a model-based trajectory candidate has been found optimal throughout the entire trial. It is represented by the rate of selecting a model-based trajectory in relation to all planning routines within a single test. Each test of the conducted simulation experiments has been considered to obtain the distribution of the percentage values (rate values). The experiments involve the *HUMAP* planner functioning with the "baseline" scoring function setup as well as with the specialised cost function weights described in Sec. 7.5.1. Therefore, the selection rates can be examined depending on the scenario but also the objective function's weights.

In general, the trajectory selection tendency significantly differs across scenarios in which the robot operates. Specifically, during the extensive simulation tests with the "baseline" weights, the model-based trajectories were the most commonly selected in the *passing* scenario. This is understandable since the parameters of the baseline pedestrian motion model (*SFM*) were calibrated based on people passing situations [25], as described in Sec. 5.5.4. In contrast, in the same *HUMAP* setup, a marginal number of model-based candidates were selected in the *overtaking* scenario (up to 8%, but the mean is approximately 2%).

Another factor influencing the percentage of trajectories chosen from the model-based trajectory generator is the cost functions' weights configuration. Notably, once the social cost functions have a negligible impact on trajectory selection ("no-soc." variant), the model-based candidates are substantially more often selected as the optimal ones. This seems to be contradictory to the expected outcomes, as the model-based trajectory generator relies on the pedestrian motion model has been calibrated to replicate the human motion in the passing scenario. However, the reason behind the more favourable model-based trajectories might originate from the density of the crowd in the calibration data [421], which has been bigger than the density in the examined scenarios, involving two human participants interacting with the robot.

Chapter 8

Conclusions

This thesis discusses the multifaceted topic of social robot navigation. The main objective of this dissertation is to develop an algorithm that enhances the navigation quality of robots operating in environments shared with people. This is associated with mitigating human discomfort caused by the motion of a robot that executes designated service tasks, i.e., interacts with people in an unfocused way.

This dissertation contributes to several key aspects of social robot navigation. Firstly, it defines the requirements that should be implemented in comprehensive social robot navigation systems. Secondly, it addresses the challenge of the quantitative assessment by proposing additional metrics to evaluate the compliance of navigation algorithms with the grounded requirements. Thirdly, it presents a local trajectory planning approach that adapts robots to navigate in environments shared with humans.

8.1 Research outcomes

Review of the state-of-the-art literature to obtain study-based requirements for social robot navigation In Chapter 2, the study-based requirements, relevant to socially navigating robots, have been identified based on the extensive literature review. Gathering insights on how participants of user studies perceive robot behaviours enabled the grounding of social robot navigation necessities that can be further transferred into requirements for robot control systems implementing socially-aware robot navigation. The developed taxonomy of necessities classifies them as: requirements regarding the physical safety of humans, perceived safety of humans, requirements for assessing the naturalness of robot motion, and compliance with social conventions. The taxonomy of social robot navigation requirements is complemented by the objectives of traditional robot navigation.

The core principles identified as groups of human-aware robot navigation requirements are the foundation for reviewing various algorithms that solve perception (Sec. 3.1), motion planning (Sec. 3.2), and evaluation (Sec. 3.3) challenges. Then, state-of-the-art motion planning methods for social robot navigation, as well as evaluation benchmarks have been classified according to the proposed requirements' taxonomy. Those summaries guided the integration of the robotic system used during the experimental studies, which employs the most comprehensive methods available.

Design and implementation of quantitative metrics for evaluating social robot navigation The substantial topic of this thesis is the quantitative assessment of human discomfort, which serves as a measure of robot social appropriateness. The insights from the reviewed user studies have been used to develop various metrics relevant for evaluating the fulfilment of the grounded social robot navigation requirements across various motion planning methods (Chapter 4). The novel indicators regard the perceived safety of humans and assess robot motion naturalness. Their substantially original aspect is that the human discomfort metrics account for people tracking uncertainty, facilitating the evaluation using robot onboard sensors and perception modules.

Human discomfort indicators have been complemented by the metrics for evaluating algorithms' adherence to classical robot navigation requirements. The entire set of metrics is implemented in the open source benchmark system, named Social Robot Planner Benchmark (*SRPB*), applicable for testing robots operating in simulated and real-world environments. A distinctive characteristic of the proposed benchmark is the diversity of metrics (Tab. 3.5), which were formulated to allow the system usage with different robot types (either with nonholonomic or holonomic drives).

Design and implementation of a human-aware local trajectory planner using the hybrid trajectory candidates generation method and spatiotemporal cost functions An essential step in attaining the thesis' objective is the development of a novel socially-aware trajectory planning method for mobile robots that takes into account the constraints that mitigate human discomfort while ensuring navigation task performance comparable to state-of-the-art traditional approaches. These guidelines have been implemented in the algorithm named *Human-Aware Trajectory Planner Mapping the Pedestrians Motion Pattern* (*HUMAP*), developed for differential drive and holonomic robots functioning in partially observable unstructured environments.

The planner operates according to the behaviour-based paradigm, implementing various behaviours in the internal *Finite State Machine* (*FSM*). The main behaviour of the planner, which is described in detail in this thesis, focuses on enabling the robot to seamlessly interact with humans in an unfocused manner, which is a typical challenge for service robots intended to operate in populated environments. The behaviour-based approach allows for capturing customary spatiotemporal protocols of pedestrian motion, as well as enables the high robustness of the algorithm in realistic challenging scenarios. The *HUMAP* exhibits the contextual awareness at the environmental level (Sec. 3.1.4), as the predicates that dictate the *FSM*'s operation are obtained from, i.a., environment observations.

The unique attributes of the proposed human-aware trajectory planning method include the hybrid approach for generating trajectory candidates, as well as the multifaceted objective function that assesses trajectories from the perspective of navigation task performance and human discomfort. As the proposed *HUMAP* algorithm solves the problem of receding horizon trajectory planning for dynamic systems, the trajectory candidate with the lowest cost is selected as the valid trajectory, whose first velocity command is applied by the mobile base controller.

The hybrid method for the trajectory candidate generation integrates the pedestrian motion model with the well-established velocity sampling in the space of admissible controls [144, 161] producing uniformly curved trajectories.

The first trajectory generator employs the *Social Force Model*-based [1] pedestrian motion model extended with an additional component based on the *Fuzzy Inference System* to obtain emphasised collision avoidance behaviours and motion legibility in dynamic scenarios. The model-based generator produces concentrated trajectories avoiding collisions and following a local goal. The parameters of the baseline pedestrian motion model have been calibrated in [25] based on video sequences of real-world pedestrians passing each other. However, the parameters of the model formulation, that best fitted the reference data, displayed considerable standard deviations from the mean values. Therefore, the model-based trajectory generator of the *HUMAP* supplies the model with parameters across the spectrum of their meaningful values, resulting in the generation of numerous trajectory candidates using the deterministic pedestrian motion model.

The multifaceted objective function assesses the cost of trajectories from the perspective of navigation task performance (collision avoidance, motion efficiency, path length), as well as human discomfort (i.a., executing natural motions, respecting personal spaces, enhancing motion legibility). The objective function is composed of numerous cost functions, each reflecting an individual aspect involved in the grounded requirements of social robot navigation (Sec. 5.8). As the objectives of classical and human-aware perspectives are contradictory, the planner searches for a Pareto-optimal solution in each planning step, successfully achieving real-time operation.

8.2 Results discussion

As a part of this thesis, comparative experiments of various local trajectory planning algorithms have been conducted in simulation and real-world environments. The tests aimed to evaluate the multifaceted performance of the traditional algorithms, that treat humans as typical obstacles, against the state-of-the-art local trajectory planners specialised for human-aware robot navigation, including the proposed *HUMAP* method. In all studies, the TIAGo robot has been used (Sec. 7.1).

8.2.1 Thesis 1

Thesis 1 suggests that state-of-the-art human-aware trajectory planners do not significantly outperform traditional algorithms in terms of discomfort mitigation among humans in proximity to the robot.

The validation of the thesis relied on two experimental scenarios – *static* and *dynamic*, both involving a robot and two human participants but interacting in different ways. Tests were conducted as a controlled study to isolate the factors influential to the experiments' results (Sec. 7.3). Each tested trajectory planner (four traditional and two socially-aware algorithms) has been using the same environment model (both the static map, as well as the equal configuration of costmaps generated in real-time), and the same initial and goal poses for each trial of a scenario. The evaluation criteria, originating from the grounded requirements for social robot navigation, include the assessment of the robot's task performance, motion naturalness, and the perceived safety of humans that is affected by the unfocused interaction with the robot executing navigation tasks in test scenarios. The degree to which the evaluation criteria have been met can be verified using the quantitative indicators of the *SRPB*.

Perceived safety of humans The perceived safety of humans surrounding the navigating robot has been assessed with, i.a., the metric reflecting the scale of intrusions into humans' personal spaces. This indicator reflects the fundamental aspect of proxemics theory [47] and is a well-established objective of human-aware motion planning methods.

The results of the study (Tab. 7.3) show that across all test scenarios, the traditional *TEB* algorithm intruded personal zones at a similar or lower scale as its human-aware specialisations – *HaTEB* and *CoHAN* while reaching the goal in substantially shorter time. In contrast, the *HaTEB* planner, performed better than the traditional *DWA* planner only by 1 p.p. in the *static* scenario, but at the cost of task execution efficiency, requiring approximately 115% more time to achieve the target pose. Considerable results in terms of respecting the personal spaces were achieved by other reliable traditional trajectory planners, i.e., *DWA* and *Trajectory Rollout*.

Another indicator, against which the examined methods were tested, evaluates the legibility of the robot's motion, as reflected by its heading directly into a human. Again, the traditional *TEB* outperformed its both socially-aware specialisations in 75% of test cases, while another classical planner, *Trajectory Rollout*, performed comparably or better than *HaTEB* or *CoHAN* across scenarios.

Robot motion naturalness The robot's motion naturalness, selected as the second criterion for the comparison of algorithms, examines, i.a., the smoothness of the robot's velocity profile. In that matter, the traditional velocity sampling planners – *DWA* and *Trajectory Rollout* substantially outperformed other algorithms, providing smoother trajectories in terms of linear and angular velocities.

Navigation task performance The overall performance of the task execution of the robot is also included in the evaluation criteria. In each scenario, the classical *DWA* and *Trajectory Rollout* approaches required the least time to reach the goal poses, while also traversing the shortest paths with the least heading changes along the way. Notably, in most cases, the human-aware trajectory planners maintained bigger gaps from surrounding obstacles.

Summary Since social robot navigation is posed as a problem of contrary criteria (task performance against social awareness), the quantitative outcomes of the study have to be investigated holistically. Three criteria were selected for the comprehensive assessment of examined methods and validating the thesis 1. In all of them, traditional trajectory planners selected for the study performed better or similarly to the algorithms specialised for obtaining social acceptance; therefore, the thesis has been proved.

Overall, the study described in Sec. 7.3 involved several well-established planners and provided the baseline assessment of the social acceptance (reflected by the *SRPB*'s quantitative metrics) of the traditional and human-aware robot navigation algorithms. It aimed to illustrate whether socially-aware local trajectory planning is still an open problem and whether the development of a novel algorithm is justified.

8.2.2 Thesis 2

Thesis 2 suggests that it is possible to develop an algorithm that demonstrates navigation performance comparable to traditional trajectory planners and surpasses state-of-the-art human-aware trajectory planners in alleviating human discomfort. The developed *HUMAP* planner is employed to demonstrate the thesis.

The performance of the proposed approach was compared against other planners in three challenging scenarios that are typical for robots subjected to unfocused interactions with humans in dynamic environments. Test scenarios constitute various types of situations, identified as standard evaluation protocols [10], namely – *crossing*, *overtaking*, and *passing*. Each scenario involved two human subjects and a robot. To validate the thesis 2, large-scale simulation study as well as real-world experiments have been conducted (Sec. 7.4), providing trajectory similarity of human participants in each trial.

The setup of the navigation ecosystem was common among all (13) tested planners including 4 traditional approaches, and 8 human-aware planners (5 of which are learning-based), complemented with the proposed socially-aware *HUMAP* algorithm.

The evaluation criteria are similar to those for validating thesis 1, encompassing the robot's task performance, motion naturalness, and the perceived safety of humans, but also include robustness of navigation task execution. Demonstrating the extent to which the evaluation criteria have been met utilises the quantitative indicators implemented in the *SRPB* benchmark system.

Robustness The first criterion for the comparison of various algorithms focuses on their robustness, which is understood as the ability to complete the designated navigation tasks. The assessment relies on a success rate calculated for different planners based on their 100 simulated and 5 real-world tests (Tab. 7.4 and 7.5).

The highest robustness was achieved by the *HUMAP*, which effectively finished 100% of simulated and real-world experimental trials (Fig. 7.15). The robustness of the planner is primarily caused by the environmental awareness integrated into the *FSM*'s predicates, which orchestrate the *HUMAP*'s behaviours. Particularly, one of them performs additional actions to, e.g., gather environment observations in case of an occluded global path.

Nonetheless, the *TEB* planner achieved a success rate of 100% in *overtaking* and *passing* scenarios, but 96% in *crossing*. The *DWA* exhibited a lower success rate in *crossing* scenario, similarly as the *Trajectory Rollout*, whose degradation in this setup indicates potential unsuitability for highly dynamic environments. On the other hand, *Elastic Bands* displayed notably lower success rates in *overtaking* scenarios, which suggests its limited applicability for environments with narrow passages.

Besides the *HUMAP*, the *SRL-EBand* and *CoHAN* algorithms emerged as the most reliable human-aware trajectory planners with 78–95% success rates. Although the *HaTEB* completed 80–90% trials in *crossing* and *passing* tests, it obtained only 50% success rate in the *overtaking* scenario.

While analytical trajectory planners obtained various but relatively high success rates, the *DRL*-based planners, adapted for operating in populated environments performed substantially less reliably. In the simulation trials, the *GA3C-CADRL*, *SARL*, *SARL**, and *RG's DRL* algorithms exhibited a 0% success rate across all scenarios, whereas the *DRL-VO* demonstrated 38% success rate in the simulated *crossing* scenario but 2% in *passing*, and 0% in *overtaking*.

Several *DRL*-based planners managed to traverse only segments of reference paths. However, in some trials, approaches with 0% success rates reached up to 80% of the reference paths before the task terminated due to the prolonged oscillations of the mobile base or experiment timeouts (Sec. 7.4.3). Inadequate success rates of *DRL*-based planners disqualified them from real-world tests. Although algorithms from that category typically perform well in tests conducted by the authors, limited generalisation to diverse environments and the complexity of scenarios selected for evaluation led to their poor performance. Additionally, the policies of *DRL*-based planners were not tailored for the target environments, but rather sensor data provided by the robot have been adjusted for those algorithms.

Navigation task performance The second evaluation criterion takes into account the performance of navigation task execution with different trajectory planning algorithms.

In real-world and simulated environments, the traditional *DWA* outperformed other planners in terms of motion efficiency, i.e., the time required to reach the goal pose. In contrast, human-aware trajectory planners with considerable robustness (*SRL-EBand*, *HaTEB*, *CoHAN*, and *HUMAP*) required longer times to reach goals due to the need for additional manoeuvres to navigate around humans. However, this trade-off is justified by the reduced disruption to surrounding humans.

Notably, in the *crossing* scenario, the *HUMAP* achieved comparable or better results in terms of motion efficiency compared to other planners. This is due to its environmental awareness and dedicated behaviour for handling "crossing" situations. Furthermore, in the *passing* scenario, only the *DWA* outperformed the *HUMAP*, while in the *overtaking* setup, the *HUMAP* completed tasks significantly faster than other human-aware trajectory planners.

Robot motion naturalness Another aspect of the comparison criteria relates to the naturalness of the robot's movements, where the most important factor is the smoothness of linear and angular velocities.

In that matter, *TEB*-based planners exhibited notably more rough angular velocity profiles than the *DWA* and *HUMAP* approaches. While the traditional *DWA* and *Trajectory Rollout* planners consistently performed best across all scenarios, the *HUMAP* achieved scores 20-50% higher, but the second best-performing human-aware planner

(*SRL-EBand*) displayed measures 175-300% higher.

Although the *HUMAP* outperformed other human-aware planners in velocity smoothness scores, its performance was slightly inferior compared to traditional algorithms. However, it is explainable by the necessity to perform additional manoeuvres to enhance the perceived safety of humans.

Perceived safety of humans The crucial criterion of the human-aware trajectory planners' comparison regards the perceived safety of humans, which might be affected by the robot's motions.

One of the relevant aspects concerning humans' perceived safety relates to the intrusions into personal spaces, where the *HUMAP* consistently achieved the best or the second-best results in all scenarios, being outperformed only by the *CoHAN* approach in the real-world *overtaking* scenario by 0.52 pp. and by the *HaTEB* in the simulated *passing* scenario by 2.16 pp. Notably, the *HUMAP* planner demonstrated superior performance in most real-world scenarios.

Another concept of perceived safety is the motion legibility assessed as human discomfort caused by the robot's heading direction. The *HUMAP* achieved the best results in all scenarios, excluding the apparent outcomes of the *Elastic Bands* approach (detailed in Sec. 7.4.8) in the simulated *overtaking* and *passing* scenarios.

The assessment of human discomfort caused by the robot's inappropriate passing speed [72] is also taken into consideration in the comparison of planners. The *HUMAP* achieved the best results in the *overtaking* scenario and performed better or comparably to the *CoHAN* in the *crossing* interaction. Interestingly, the classical *TEB* planner displayed superior performance in the *passing* scenario, outperforming other planners in both simulation and real-world settings by approximately 2-2.5 pp, while the *HUMAP* significantly outperformed other human-aware planners in that case.

Summary The multifaceted assessment criteria have been considered to quantify the performance of the developed *HUMAP* method. The proposed algorithm aims to mitigate human discomfort during unfocused interactions with the robot while providing task performance efficiency comparable to traditional methods and high robustness in challenging scenarios.

The results of the large-scale simulation experiments as well as real-world tests demonstrated the surpassing performance of the *HUMAP* local trajectory planner. According to the standardised study-based metrics implemented in the *SRPB*, the planner not only outperforms state-of-the-art methods in terms of reducing human discomfort but also ensures reliable and efficient navigation task execution across various dynamic scenarios;

hence, thesis 2 has been proved.

What is more, the experiments performed to prove the thesis 2 involved several trajectory planners examined during the tests in support of thesis 1. The outcomes of the reused algorithms identify that the differences between investigated traditional and human-aware methods were consistent across the studies despite the variances in evaluation scenarios. Therefore, the evidence for thesis 1 has been expanded.

8.3 Future work

This dissertation examines the state-of-the-art in the social robot navigation field, as well as compares the proposed contributions against various methods frequently applied for practical uses. The thorough consideration allows for defining future work perspectives regarding the developed methods.

Validation of the proposed metrics Metrics implemented in the *SRPB* were used throughout the experiments for assessing the degree of fulfilment of the grounded requirements by different robot navigation algorithms. However, we did not attempt to revalidate the correlation between metric values and the impressions of humans interacting with the robot, as the implemented metrics constitute continuous models of findings from prior user studies or are directly derived from principles discussed in the literature (Sec. 4.1).

The accuracy of the implemented metrics could be validated with a multi-scenario user study involving human participants interacting with a robot operating with different motion planning algorithms. Then, the correlation of metric scores, obtained for different navigation setups, could be quantified against human ratings. Such a study; however, poses a significant organisational effort [99, 87, 4, 7], and is close to the "ideal" evaluation method defined in [35], requiring the involvement of numerous subjects in a large-scale study, which contradicts the primary idea of developing automatic benchmarks for quantitative evaluation, e.g., *SRPB*. A more cost-efficient alternative is a video-based evaluation, employed in numerous works [111, 335, 86], that relies on presenting the videos of a robot operating with different planning algorithms to subjects. The study participants could then be asked to score the robot's behaviour in terms of performance and acceptance of interaction with humans.

Moreover, conducting a user study that aims to gather first-hand human insights against the values of metrics implemented in various benchmarks (Sec. 3.3.3) could be a significant contribution to the social robotics field and a promising future work perspective. Additionally, the importance of heuristic-based human-perceived safety indicators, e.g., [395], might also be validated in such an experiment.

Composite metrics for the SRPB system During the experiments, it has been observed that methods like the *Elastic Bands* proved to outperform others in terms of some *SRPB* metrics, e.g., intrusions into personal spaces, but primarily because the human-robot interactions intended for a specific scenario, e.g., "crossing" (Sec. 7.4.3), were hindered due to slow progressive movements that emerged when the robot was commanded by an ineffective trajectory planner. Therefore, the analysis of benchmark scores achieved by planners designated for human-robot interaction should not only consider human discomfort metrics but also performance indicators related to the robot navigation task.

Currently, the *SRPB* benchmark system provides users with a variety of metrics that must be analysed holistically. Thus, it would be appropriate to develop two unified indicators separately calculated for metrics regarding the classical navigation requirements and metrics concerning requirements for social robot navigation, as this could significantly ease the initial overview of results obtained with the benchmark.

Extensive examination of the HUMAP's performance Most experiments involving *HUMAP* were performed in a specific laboratory environment in standardised scenarios posing realistic challenges for service robots operating in human surroundings. More studies in different environments could provide more insights regarding the performance of the *HUMAP*. Although the proposed planner has been evaluated in crowded but static scenarios in the development stage, an extensive evaluation within dense dynamic crowds could provide more insights regarding its suitability for populated environments. Additionally, the heuristic implemented in the behaviour associated with the *HUMAP*'s *Yield Way Crossing* state, provided reliable solutions for one or two humans nearby, but was not tested in dense crowds.

Future work perspectives involve comparing other human-aware navigation techniques with our approach. Particularly interesting would be to evaluate the *HUMAP* against the *GTEB* [333] method, which accounts for spatiotemporal constraints and focused interaction aspects. Such a study would be a good benchmark of the proposed algorithm and might identify practical drawbacks of the *HUMAP* in focused interaction scenarios. However, even though implementation of the *TEB*, which *GTEB* originates from, is open-source, its human-aware specialisation is not publicly available. An extensive comparison of *HUMAP* with *DRL*-based planners with policies tailored for specific environments is also planned.

Enhancements of the HUMAP local trajectory planner The *HUMAP* exhibited the highest computational burden among all examined methods while still being capable of real-time operation at a frequency of 4 Hz. However, the trajectory generation and scoring strategies are suitable for parallelising, which could lead to reducing the computational burden of the algorithm. Alternatively, the user might favour expanding the number of trajectory candidates investigated in each replanning procedure and maintaining the reference frequency of operation.

One of the analyses shows that model-based trajectories are the minority among the candidates selected as optimal in each time step (Sec. 7.5.2). However, a visible correlation between the scenario type and the percentages is found. Therefore, tests with different baseline pedestrian motion model parameters might be useful to explain the observed tendency. Since the employed pedestrian model-based trajectory generator contributed at a limited scale to the *HUMAP*'s outcomes, an interesting future work perspective would be to compare the results of the presented hybrid trajectory generation scheme in a setup consisting of the velocity sampling generator and one of the investigated *DRL*-based approaches. We argue that such an integration might be an interesting alternative for end-to-end learning methods.

The implemented human trajectory forecasting scheme assumes the constant velocity model; however, social actors in the environment may change their behaviour influenced by the robot's movement. Thus applying a more sophisticated method for predictions, e.g., proposed in [197], would be relevant in future studies.

The implemented formulations of human-aware spatial cost functions are designated for unfocused human-robot interactions, i.e., depend on human velocities [399] regardless of the robot task context. Instead, introducing dynamically modelled proxemics zones (of individual humans and F-formations) will provide a comprehensive framework also for the focused interactions. This problem has already been addressed in [359, 8].

Currently, the operation of the robot relies on in-place adjustment of the final orientation to align with the desired orientation. While the final in-place rotation occurs only when the goal orientation significantly differs from the final part of the path, to enhance the robot's motion naturalness, a smoother transition to the goal orientation would be desirable. Such a robot behaviour emerges with the *TEB*-based planners.

Recent trends show that the substantial development perspectives for social robot navigation go well beyond the motion generation scheme with trajectory planners, but rather focus on the orchestration of navigation tasks to comply with social norms [11]. Therefore, extending the *HUMAP*'s behaviour-based imperative, or integrating with a higher level orchestrator [19] for enriched contextual awareness, could be the most influential topic for future works.

Bibliography

[1] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[2] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *International Journal of Social Robotics*, vol. 7, pp. 137–153, Apr 2015.

[3] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. The MIT Press, 2nd ed., 2011.

[4] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, "How to approach humans? strategies for social robots to initiate interaction," in *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, HRI '09, (New York, NY, USA), pp. 109–116, Association for Computing Machinery, 2009.

[5] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human-robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.

[6] J. Biswas and M. Veloso, "The 1,000-km challenge: Insights and quantitative and qualitative results," *IEEE Intelligent Systems*, vol. 31, no. 3, pp. 86–96, 2016.

[7] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model," *International Journal of Social Robotics*, vol. 6, pp. 443–455, Aug 2014.

[8] J. Ginés, F. Martín, D. Vargas, F. J. Rodríguez, and V. Matellán, "Social navigation in a cognitive architecture using dynamic proxemic zones," *Sensors*, vol. 19, no. 23, 2019.

[9] R. Möller, A. Furnari, S. Battiato, A. Härmä, and G. M. Farinella, "A survey on human-aware robot navigation," *Robotics and Autonomous Systems*, vol. 145, p. 103837, 2021.

[10] Y. Gao and C.-M. Huang, "Evaluation of socially-aware robot navigation," *Frontiers in Robotics and AI*, vol. 8, 01 2022.

[11] A. Francis, C. Pérez-d'Arpino, C. Li, F. Xia, A. Alahi, A. Bera, A. Biswas, J. Biswas, R. Chandra, H.-T. Lewis Chiang, M. Everett, S. Ha, J. Hart, J. P. How, H. Karnan, T.-W. E. Lee, L. J. Manso, R. Mirksy, S. Pirk, P. Stone, A. V. Taylor, P. Trautman, N. Tsoi, M. Vázquez, X. Xiao, P. Xu, N. Yokoyama, A. Toshev, R. Martín-Martín, R. Alami, and P.-T. Singamaneni, "Principles and Guidelines for Evaluating Social Robot Navigation Algorithms." working paper or preprint, Jul 2023.

[12] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, pp. 1726–1743, 12 2013.

[13] K. Charalampous, I. Kostavelis, and A. Gasteratos, "Recent trends in social aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 93, pp. 85–104, 2017.

[14] S. F. Chik, C. F. Yeong, E. L. M. Su, T. Y. Lim, Y. Subramaniam, and P. J. H. Chin, "A review of social-aware navigation frameworks for service robot in dynamic human environments," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 8, pp. 41–50, 01 2016.

[15] C. Medina Sánchez, M. Zella, J. Capitán, and P. J. Marrón, "From perception to navigation in environments with persons: An indoor evaluation of the state of the art," *Sensors*, vol. 22, no. 3, 2022.

[16] S. Guillén-Ruiz, J. P. Bandera, A. Hidalgo-Paniagua, and A. Bandera, "Evolution of socially-aware robot navigation," *Electronics*, vol. 12, no. 7, 2023.

[17] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, pp. 1–39, Feb. 2023.

[18] P. T. Singamaneni, P. Bachiller-Burgos, L. J. Manso, A. Garrell, A. Sanfeliu, A. Spalanzani, and R. Alami, "A survey on socially aware robot navigation: Taxonomy and future challenges," *The International Journal of Robotics Research*, 2024.

[19] J. Karwowski, W. Szynkiewicz, and E. Niewiadomska-Szynkiewicz, "Bridging requirements, planning, and evaluation: A review of social robot navigation," *Sensors*, vol. 24, no. 9, 2024.

[20] J. Karwowski and W. Szynkiewicz, "SRPB: a benchmark for the quantitative evaluation of a social robot navigation," in *2023 27th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 411–416, 2023.

[21] J. Karwowski and W. Szynkiewicz, "Quantitative metrics for benchmarking human-aware robot navigation," *IEEE Access*, vol. 11, pp. 79941–79953, 2023.

[22] D. Ferguson and M. Likhachev, "Efficiently using cost maps for planning complex maneuvers," *Lab Papers (GRASP)*, 05 2008.

[23] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 301–308, 2013.

[24] J. Karwowski and W. Szynkiewicz, "Human-aware robot trajectory planning with hybrid candidate generation: leveraging a pedestrian motion model for diverse trajectories," in *2024 13th International Workshop on Robot Motion and Control (RoMoCo) [accepted]*, 2024.

[25] S. Seer, C. Rudloff, T. Matyus, and N. Brändle, "Validating social force based models with comprehensive real world motion data," *Transportation Research Procedia*, vol. 2, pp. 724–732, 2014.

[26] K. E. Schaefer, J. Oh, D. Aksaray, and D. Barber, "Integrating context into artificial intelligence: Research from the robotics collaborative technology alliance," *AI Magazine*, vol. 40, pp. 28–40, Sep 2019.

[27] P. A. Lasota, T. Fong, and J. A. Shah, "A survey of methods for safe human-robot interaction," *Foundations and Trends® in Robotics*, vol. 5, no. 4, pp. 261–349, 2017.

[28] R. Mirsky, X. Xiao, J. Hart, and P. Stone, "Conflict avoidance in social navigation - a survey," *ACM Transactions on Human-Robot Interaction*, vol. 13, Mar. 2024.

[29] C. Breazeal, *Designing Sociable Machines*, pp. 149–156. Boston, MA: Springer US, 2002.

[30] F. Babel, J. M. Kraus, and M. Baumann, "Development and testing of psychological conflict resolution strategies for assertive robots to resolve human-robot goal conflict," *Frontiers in Robotics and AI*, vol. 7, 2021.

[31] C. Zieliński, T. Kornuta, and T. Winiarski, "A systematic method of designing control systems for service and field robots," in *19-th IEEE International Conference on Methods and Models in Automation and Robotics, MMAR'2014*, pp. 1–14, IEEE, 2014.

[32] S.-H. Lin and F.-L. Lian, "Study of feasible trajectory generation algorithms for control of planar mobile robots," in *2005 IEEE International Conference on Robotics and Biomimetics - ROBIO*, pp. 121–126, 2005.

[33] J. Wen, X. Zhang, Q. Bi, Z. Pan, Y. Feng, J. Yuan, and Y. Fang, "MRPB 1.0: A unified benchmark for the evaluation of mobile robot local planning approaches," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8238–8244, 2021.

[34] E. Heiden, L. Palmieri, L. Bruns, K. O. Arras, G. S. Sukhatme, and S. Koenig, "Bench-mr: A motion planning benchmark for wheeled mobile robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4536–4543, 2021.

[35] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni, "Socnavbench: A grounded simulation testing framework for evaluating social navigation," *J. Hum.-Robot Interact.*, vol. 11, jul 2022.

[36] J. Guzzi, A. Giusti, L. M. Gambardella, G. Theraulaz, and G. A. Di Caro, "Human-friendly robot navigation in dynamic environments," in *2013 IEEE International Conference on Robotics and Automation*, pp. 423–430, 2013.

[37] S. B. Banisetty, *Multi-Context Socially-Aware Navigation Using Non-Linear Optimization.* PhD dissertation, Department of Computer Science and Engineering, University of Nevada, Reno, NV, USA, May 2020.

[38] I. Dulęba and J. Sasiadek, "Nonholonomic motion planning based on newton algorithm with energy optimization," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 3, pp. 355–363, 2003.

[39] P. Boddington, "EPSRC principles of robotics: commentary on safety, robots as products, and responsibility," *Connection Science*, vol. 29, no. 2, pp. 170–176, 2017.

[40] R. Clarke, "Asimov's laws of robotics: Implications for information technology-part i," *Computer*, vol. 26, pp. 53–61, dec 1993.

[41] A. Bera, T. Randhavane, R. Prinja, and D. Manocha, "SocioSense: Robot navigation amongst pedestrians with social and psychological constraints," in *2017 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), pp. 7018–7025, 2017.

[42] V. Narayanan, B. M. Manoghar, V. S. Dorbala, D. Manocha, and A. Bera, "ProxEmo: Gait-based emotion learning and multi-view proxemic fusion for socially-aware robot navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[43] R. M. Bena, C. Zhao, and Q. Nguyen, "Safety-aware perception for autonomous collision avoidance in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 7962–7969, 2023.

[44] M. Moussaïd, D. Helbing, and G. Theraulaz, "How simple rules determine pedestrian behavior and crowd disasters," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, pp. 6884–8, 04 2011.

[45] Z. Forootaninia, I. Karamouzas, and R. Narain, "Uncertainty models for ttc-based collision-avoidance," in *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017* (N. M. Amato, S. S. Srinivasa, N. Ayanian, and S. Kuindersma, eds.), 07 2017.

[46] I. Karamouzas, N. Sohre, R. Narain, and S. J. Guy, "Implicit crowds: Optimization integrator for robust crowd simulation," *ACM Trans. Graph.*, vol. 36, Jul 2017.

[47] E. T. Hall, *The hidden dimension / by Edward T. Hall.* A Doubleday Anchor book, Garden City, N.Y: Anchor Books, 1969.

[48] J. R. Aiello, "A further look at equilibrium theory: Visual interaction as a function of interpersonal distance," *Environmental psychology and nonverbal behavior*, vol. 1, pp. 122–140, Mar 1977.

[49] N. L. Ashton and M. E. Shaw, "Empirical investigations of a reconceptualized personal space," *Bulletin of the Psychonomic Society*, vol. 15, pp. 309–312, May 1980.

[50] M. Baldassare, "Human spatial behavior," *Annual Review of Sociology*, vol. 4, no. 1, pp. 29–56, 1978.

[51] C. I. Greenberg, M. J. Strube, and R. A. Myers, "A multitrait-multimethod investigation of interpersonal distance," *Journal of Nonverbal Behavior*, vol. 5, pp. 104–114, Dec 1980.

[52] J. T. Butler and A. Agah, "Psychological effects of behavior patterns of a mobile personal robot," *Autonomous Robots*, vol. 10, pp. 185–202, Mar 2001.

[53] P. Althaus, H. Ishiguro, T. Kanda, T. Miyashita, and H. Christensen, "Navigation for human-robot interaction tasks," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 2, pp. 1894–1900 Vol.2, 2004.

[54] L. Hayduk, "Personal space: An evaluative and orienting overview.," *Psychological Bulletin*, vol. 85, pp. 117–134, 1978.

[55] L. Hayduk, "The shape of personal space: An experimental investigation.," *Canadian Journal of Behavioural Science*, vol. 13, pp. 87–93, 1981.

[56] A. Johansson, D. Helbing, and P. K. Shukla, "Specification of the social force pedestrian model by evolutionary adjustment to video tracking data.," *Advances in Complex Systems*, vol. 10, pp. 271–288, 12 2007.

[57] M. Gérin-Lajoie, C. L. Richards, J. Fung, and B. J. McFadyen, "Characteristics of personal space during obstacle circumvention in physical and virtual environments," *Gait & Posture*, vol. 27, no. 2, pp. 239–247, 2008.

[58] J. C. Baxter, "Interpersonal spacing in natural settings," *Sociometry*, vol. 33, pp. 444–456, 2024/02/27/ 1970. Full publication date: Dec., 1970.

[59] J. Kessler, C. Schroeter, and H.-M. Gross, "Approaching a person in a socially acceptable manner using a fast marching planner," in *Intelligent Robotics and Applications* (S. Jeschke, H. Liu, and D. Schilberg, eds.), (Berlin, Heidelberg), pp. 368–377, Springer Berlin Heidelberg, 2011.

[60] D. E. Thompson, J. R. Aiello, and Y. M. Epstein, "Interpersonal distance preferences," *Journal of Nonverbal Behavior*, vol. 4, pp. 113–118, Dec 1979.

[61] E. Pacchierotti, H. Christensen, and P. 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.*, pp. 164–171, 2005.

[62] E. Pacchierotti, H. I. Christensen, and P. Jensfelt, "Evaluation of passing distance for social robots," in *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 315–320, 2006.

[63] R. Welsch, C. von Castell, and H. Hecht, "The anisotropy of personal space," *PLoS ONE*, vol. 14, 06 2019.

[64] M. Neggers, R. Cuijpers, P. Ruijten, and W. Ijsselsteijn, "Determining shape and size of personal space of a human when passed by a robot," *International Journal of Social Robotics*, vol. 14, pp. 1–12, 03 2022.

[65] H. Huettenrauch, K. S. Eklundh, A. Green, and E. A. Topp, "Investigating spatial relationships in human-robot interaction," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5052–5059, 2006.

[66] E. Torta, R. H. Cuijpers, and J. F. Juola, "Design of a parametric model of personal space for robotic social navigation," *International Journal of Social Robotics*, vol. 5, pp. 357–365, Aug 2013.

[67] M. Yoda and Y. Shiota, "The mobile robot which passes a man," in *Proceedings 6th IEEE International Workshop on Robot and Human Communication. RO-MAN'97 SENDAI*, pp. 112–117, 1997.

[68] L. Takayama and C. Pantofaru, "Influences on proxemic behaviors in human-robot interaction," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, vol. 2009, pp. 5495 – 5502, 11 2009.

[69] L. A. Hayduk, "Personal space: Understanding the simplex model," *Journal of Nonverbal Behavior*, vol. 18, pp. 245–260, Sep 1994.

[70] S. Park and M. Trivedi, "Multi-person interaction and activity analysis: A synergistic track- and body-level analysis framework," *Mach. Vis. Appl.*, vol. 18, pp. 151–166, 05 2007.

[71] R. Kirby, R. Simmons, and J. Forlizzi, "Companion: A constraint-optimizing method for person-acceptable navigation," in *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 607–612, sep. 2009.

[72] M. M. E. Neggers, R. H. Cuijpers, P. A. M. Ruijten, and W. A. IJsselsteijn, "The effect of robot speed on comfortable passing distances," *Frontiers in Robotics and AI*, vol. 9, 2022.

[73] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PLOS ONE*, vol. 5, pp. 1–7, 04 2010.

[74] M. L. Federici, A. Gorrini, L. Manenti, and G. Vizzari, "Data collection for modeling and simulation: Case study at the university of milan-bicocca," in *Cellular Automata*

(G. C. Sirakoulis and S. Bandini, eds.), (Berlin, Heidelberg), pp. 699–708, Springer Berlin Heidelberg, 2012.

[75] A. Kendon, *Spacing and Orientation in Co-present Interaction*, pp. 1–15. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[76] R. Mead, A. Atrash, and M. J. Matarić, "Proxemic feature recognition for interactive robots: Automating metrics from the social sciences," in *Social Robotics* (B. Mutlu, C. Bartneck, J. Ham, V. Evers, and T. Kanda, eds.), (Berlin, Heidelberg), pp. 52–61, Springer Berlin Heidelberg, 2011.

[77] J. Rios-Martinez, A. Renzaglia, A. Spalanzani, A. Martinelli, and C. Laugier, "Navigating between people: A stochastic optimization approach," in *2012 IEEE International Conference on Robotics and Automation*, pp. 2880–2885, 2012.

[78] M. G. Efran and J. A. Cheyne, "Shared space: The co-operative control of spatial areas by two interacting individuals.," *Canadian Journal of Behavioural Science / Revue canadienne des sciences du comportement*, vol. 5, no. 3, pp. 201–210, 1973.

[79] E. S. Knowles, B. Kreuser, S. Haas, M. Hyde, and G. E. Schuchart, "Group size and the extension of social space boundaries.," *Journal of Personality and Social Psychology*, vol. 33, no. 5, pp. 647–654, 1976.

[80] J. Krueger, "Extended cognition and the space of social interaction," *Consciousness and Cognition*, vol. 20, no. 3, pp. 643–657, 2011.

[81] M. Rehm, E. André, and M. Nischt, "Lets come together – social navigation behaviors of virtual and real humans," in *Intelligent Technologies for Interactive Entertainment* (M. Maybury, O. Stock, and W. Wahlster, eds.), (Berlin, Heidelberg), pp. 336–336, Springer Berlin Heidelberg, 2005.

[82] K. Katyal, Y. Gao, J. Markowitz, S. Pohland, C. Rivera, I.-J. Wang, and C.-M. Huang, "Learning a group-aware policy for robot navigation," 2020.

[83] B. Petrak, G. Sopper, K. Weitz, and E. André, "Do you mind if i pass through? studying the appropriate robot behavior when traversing two conversing people in a hallway setting," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pp. 369–375, 2021.

[84] D. V. Lu and W. D. Smart, "Towards more efficient navigation for robots and humans," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1707–1713, 2013.

[85] T. Kruse, A. Kirsch, H. Khambhaita, and R. Alami, "Evaluating directional cost models in navigation," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '14, (New York, NY, USA), pp. 350–357, Association for Computing Machinery, 2014.

[86] C. Lichtenthäler, T. Lorenzy, and A. Kirsch, "Influence of legibility on perceived safety in a virtual human-robot path crossing task," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 676–681, 2012.

[87] K. Dautenhahn, M. Walters, S. Woods, K. Koay, C. Nehaniv, E. Sisbot, R. Alami, and T. Siméon, "How may i serve you? a robot companion approaching a seated person in a helping context," in *HRI 2006: Proceedings of the 2006 ACM Conference on Human-Robot Interaction*, vol. 2006, pp. 172–179, 03 2006.

[88] K. Koay, E. Sisbot, D. S. Syrdal, M. Walters, K. Dautenhahn, and R. Alami, "Exploratory study of a robot approaching a person in the context of handing over an object.," in *AAAI Spring Symposium - Technical Report*, pp. 18–24, 01 2007.

[89] M. L. Walters, K. Dautenhahn, S. N. Woods, and K. L. Koay, "Robotic etiquette: results from user studies involving a fetch and carry task," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, HRI '07, (New York, NY, USA), p. 317–324, Association for Computing Machinery, 2007.

[90] M. Svenstrup, S. Tranberg, H. J. Andersen, and T. Bak, "Pose estimation and adaptive robot behaviour for human-robot interaction," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, ICRA'09, pp. 3222–3227, IEEE Press, 2009.

[91] E. Torta, R. H. Cuijpers, J. F. Juola, and D. van der Pol, "Design of robust robotic proxemic behaviour," in *Social Robotics* (B. Mutlu, C. Bartneck, J. Ham, V. Evers, and T. Kanda, eds.), (Berlin, Heidelberg), pp. 21–30, Springer Berlin Heidelberg, 2011.

[92] K. L. Koay, D. S. Syrdal, M. Ashgari-Oskoei, M. L. Walters, and K. Dautenhahn, "Social roles and baseline proxemic preferences for a domestic service robot," *International Journal of Social Robotics*, vol. 6, pp. 469–488, Nov 2014.

[93] D. Karreman, L. Utama, M. Joosse, M. Lohse, B. van Dijk, and V. Evers, "Robot etiquette: how to approach a pair of people?," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '14, (New York, NY, USA), p. 196–197, Association for Computing Machinery, 2014.

[94] A. Ball, D. Silvera-Tawil, D. Rye, and M. Velonaki, “Group comfortability when a robot approaches,” in *Social Robotics* (M. Beetz, B. Johnston, and M.-A. Williams, eds.), (Cham), pp. 44–53, Springer International Publishing, 2014.

[95] M. Joosse, R. Poppe, M. Lohse, and V. Evers, “Cultural differences in how an engagement-seeking robot should approach a group of people,” in *Proceedings of the 5th ACM international conference on Collaboration across boundaries: Culture, Distance & Technology (CABS 2014)*, pp. 121–130, Association for Computing Machinery, Aug 2014.

[96] A. Sardar, M. Joosse, A. Weiss, and V. Evers, “Don’t stand so close to me: Users’ attitudinal and behavioral responses to personal space invasion by robots,” in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 229–230, 2012.

[97] S. Rossi, M. Staffa, L. Bove, R. Capasso, and G. Ercolano, “User’s personality and activity influence on hri comfortable distances,” in *Social Robotics* (A. Kheddar, E. Yoshida, S. S. Ge, K. Suzuki, J.-J. Cabibihan, F. Eyssel, and H. He, eds.), (Cham), pp. 167–177, Springer International Publishing, 2017.

[98] W. A. Sparrow and K. M. Newell, “Metabolic energy expenditure and the regulation of movement economy,” *Psychonomic Bulletin & Review*, vol. 5, pp. 173–196, Jun 1998.

[99] S. Bitgood and S. Dukes, “Not another step! economy of movement and pedestrian choice point behavior in shopping malls,” *Environment and Behavior*, vol. 38, no. 3, pp. 394–405, 2006.

[100] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz, “The nonholonomic nature of human locomotion: a modeling study,” in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, pp. 158–163, 2006.

[101] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 500–505, 1985.

[102] D. Carton, A. Turnwald, D. Wollherr, and M. Buss, *Proactively Approaching Pedestrians with an Autonomous Mobile Robot in Urban Environments*, pp. 199–214. Heidelberg: Springer International Publishing, 2013.

[103] L. Nummenmaa, J. Hyönä, and J. K. Hietanen, "I'll walk this way: Eyes reveal the direction of locomotion and make passersby look and go the other way," *Psychological Science*, vol. 20, no. 12, pp. 1454–1458, 2009. PMID: 19883491.

[104] J. Cutting, P. Vishton, and P. Braren, "How we avoid collisions with stationary and moving objects," *Psychological Review*, vol. 102, pp. 627–651, 10 1995.

[105] K. Kitazawa and T. Fujiyama, "Pedestrian vision and collision avoidance behavior: Investigation of the information process space of pedestrians using an eye tracker," in *Pedestrian and Evacuation Dynamics 2008* (W. W. F. Klingsch, C. Rogsch, A. Schadschneider, and M. Schreckenberg, eds.), (Berlin, Heidelberg), pp. 95–108, Springer Berlin Heidelberg, 2010.

[106] K. Hayashi, M. Shiomi, T. Kanda, and N. Hagita, "Friendly patrolling: A model of natural encounters," in *Proceedings of Robotics: Science and Systems VII, University of Southern California, Los Angeles, CA, USA, June 27-30, 2011*, 06 2011.

[107] Y. Kuno, K. Sadazuka, M. Kawashima, K. Yamazaki, A. Yamazaki, and H. Kuzuoka, "Museum guide robot based on sociological interaction analysis," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, (New York, NY, USA), p. 1191–1194, Association for Computing Machinery, 2007.

[108] S. M. Fiore, T. J. Wiltshire, E. J. C. Lobato, F. G. Jentsch, W. H. Huang, and B. Axelrod, "Toward understanding social cues and signals in human-robot interaction: effects of robot gaze and proxemic behavior," *Front Psychol*, vol. 4, p. 859, Nov 2013.

[109] A. D. May, C. Dondrup, and M. Hanheide, "Show me your moves! conveying navigation intention of a mobile robot to humans," in *2015 European Conference on Mobile Robots (ECMR)*, pp. 1–6, 2015.

[110] S. D. Lynch, J. Pettré, J. Bruneau, R. Kulpa, A. Crétual, and A.-H. Olivier, "Effect of virtual human gaze behaviour during an orthogonal collision avoidance walking task," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 136–142, 2018.

[111] H. Khambhaita, J. Rios-Martinez, and R. Alami, "Head-Body Motion Coordination for Human Aware Robot Navigation," in *9th International workshop on Human-Friendlly Robotics (HFR 2016)*, (Gênes, Italy), p. 8p., Sep 2016.

[112] D. V. Lu, *Contextualized Robot Navigation*. PhD thesis, Washington University in St. Louis, December 2014.

[113] C. Breazeal, A. Edsinger, P. Fitzpatrick, and B. Scassellati, "Active vision for sociable robots," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 5, pp. 443–453, 2001.

[114] B. Mutlu, T. Shiwa, T. Kanda, H. Ishiguro, and N. Hagita, "Footing in human-robot conversations: How robots might shape participant roles using gaze cues," in *2009 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 61–68, 2009.

[115] A. Kendon, "Some functions of gaze-direction in social interaction," *Acta Psychologica*, vol. 26, pp. 22–63, 1967.

[116] S. Duncan, "Some signals and rules for taking speaking turns in conversations.," *Journal of Personality and Social Psychology*, vol. 23, no. 2, pp. 283–292, 1972.

[117] K. A. Barchard, L. Lapping-Carr, R. S. Westfall, A. Fink-Armold, S. B. Banisetty, and D. Feil-Seifer, "Measuring the perceived social intelligence of robots," *J. Hum.-Robot Interact.*, vol. 9, sep 2020.

[118] J. Mumm and B. Mutlu, "Human-robot proxemics: Physical and psychological distancing in human-robot interaction," in *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 331–338, 2011.

[119] C. Lin, J. Rhim, and A. J. Moon, "Less than human: How different users of telepresence robots expect different social norms," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3976–3982, 2023.

[120] E. Jung, B. Yi, and S. Yuta, "Control algorithms for a mobile robot tracking a human in front," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012*, IEEE International Conference on Intelligent Robots and Systems, pp. 2411–2416, Dec 2012. 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems, IROS 2012 ; Conference date: 07-10-2012 Through 12-10-2012.

[121] J. E. Young, Y. Kamiyama, J. Reichenbach, T. Igarashi, and E. Sharlin, "How to walk a robot: A dog-leash human-robot interface," in *RO-MAN, 2011 IEEE*, pp. 376–382, 31 2011-Aug. 3.

[122] D. Carton, W. Olszowy, and D. Wollherr, "Measuring the effectiveness of readability for mobile robot locomotion," *International Journal of Social Robotics*, vol. 8, 11 2016.

[123] R. Gockley, J. Forlizzi, and R. Simmons, "Natural person-following behavior for social robots," in *2007 2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 17–24, 2007.

[124] X. Yao, J. Zhang, and J. Oh, "Following social groups: Socially-compliant autonomous navigation in dense crowds," in *Proceedings of IROS '19 Cognitive Vehicles Workshop*, November 2019.

[125] E. A. Topp and H. I. Christensen, "Tracking for following and passing persons," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada, August 2-6, 2005*, pp. 2321–2327, IEEE, 2005.

[126] J. Müller, C. Stachniss, K. Arras, and W. Burgard, "Socially inspired motion planning for mobile robots in populated environments," in *Proceedings of the International Conference on Cognitive Systems (CogSys)*, (Karlsruhe, Germany), pp. 85–90, April 2008.

[127] P. H. Kahn, N. G. Freier, T. Kanda, H. Ishiguro, J. H. Ruckert, R. L. Severson, and S. K. Kane, "Design patterns for sociality in human-robot interaction," in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 97–104, 2008.

[128] M. Costa, "Interpersonal distances in group walking," *Journal of Nonverbal Behavior*, vol. 34, pp. 15–26, Mar 2010.

[129] S. S. Honig, T. Oron-Gilad, H. Zaichyk, V. Sarne-Fleischmann, S. Olatunji, and Y. Edan, "Toward socially aware person-following robots," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 4, pp. 936–954, 2018.

[130] L. Y. M. Saiki, S. Satake, R. Huq, D. F. Glas, T. Kanda, and N. Hagita, "How do people walk side-by-side? — using a computational model of human behavior for a social robot," *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 301–308, 2012.

[131] D. Karunarathne, Y. Morales, T. Kanda, and H. Ishiguro, "Model of side-by-side walking without the robot knowing the goal," *International Journal of Social Robotics*, vol. 10, 09 2018.

[132] F. Lindner and C. Eschenbach, "Towards a formalization of social spaces for socially aware robots," in *Spatial Information Theory* (M. Egenhofer, N. Giudice, R. Moratz, and M. Worboys, eds.), (Berlin, Heidelberg), pp. 283–303, Springer Berlin Heidelberg, 2011.

[133] L. Calderita, A. Vega, P. Bustos, and P. Núñez, "Social robot navigation adapted to time-dependent affordance spaces: a use case for caregiving centers," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 944–949, 2020.

[134] M. Raubal and R. Moratz, "A functional model for affordance-based agents," in *Towards Affordance-Based Robot Control* (E. Rome, J. Hertzberg, and G. Dorffner, eds.), (Berlin, Heidelberg), pp. 91–105, Springer Berlin Heidelberg, 2008.

[135] S. Y. Chung and H. Huang, "Incremental learning of human social behaviors with feature-based spatial effects," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pp. 2417–2422, IEEE, 2012.

[136] F. Yuan, L. Twardon, and M. Hanheide, "Dynamic path planning adopting human navigation strategies for a domestic mobile robot," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 3275 – 3281, 11 2010.

[137] E. Pacchierotti, H. I. Christensen, and P. Jensfelt, "Embodied social interaction for service robots in hallway environments," in *Field and Service Robotics* (P. Corke and S. Sukkariah, eds.), (Berlin, Heidelberg), pp. 293–304, Springer Berlin Heidelberg, 2006.

[138] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental study of the behavioural mechanisms underlying self-organization in human crowds," *Proceedings of the Royal Society B*, vol. 276, pp. 2755–2762, 2009.

[139] Y. Nakauchi and R. Simmons, "A social robot that stands in line," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, vol. 1, pp. 357–364 vol.1, 2000.

[140] D. Gallo, S. Gonzalez-Jimenez, M. A. Grasso, C. Boulard, and T. Colombino, "Exploring machine-like behaviors for socially acceptable robot navigation in elevators," in *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 130–138, 2022.

[141] A. K. Pandey and R. Alami, "A framework towards a socially aware mobile robot motion in human-centered dynamic environment," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5855–5860, 2010.

[142] C. Dondrup and M. Hanheide, "Qualitative constraints for human-aware robot navigation using velocity costmaps," in *25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, IEEE International Symposium on Robot and Human Interactive Communication, (United States), pp. 586–592, IEEE, nov 2016. 25th IEEE International Symposium on Robot and Human Interactive Communication 2016, RO-MAN 2016 ; Conference date: 26-08-2016 Through 31-08-2016.

[143] N. Hirose, D. Shah, A. Sridhar, and S. Levine, "SACSoN: Scalable autonomous control for social navigation," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 49–56, 2024.

[144] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[145] M. Walters, K. Dautenhahn, R. te Boekhorst, K. L. Koay, C. Kaouri, S. Woods, C. Nehaniv, D. Lee, and I. Werry, "The influence of subjects' personality traits on personal spatial zones in a human-robot interaction experiment," in *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pp. 347–352, 2005.

[146] E. Pacchierotti, H. I. Christensen, and P. Jensfelt, "Design of an office-guide robot for social interaction studies," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4965–4970, 2006.

[147] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *2010 IEEE International Conference on Robotics and Automation*, pp. 300–307, 2010.

[148] D. Zhang, Z. Xie, P. Li, J. Yu, and X. Chen, "Real-time navigation in dynamic human environments using optimal reciprocal collision avoidance," in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 2232–2237, 2015.

[149] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5512–5519, 2016.

[150] P. T. Singamaneni, A. Favier, and R. Alami, "Watch out! there may be a human. addressing invisible humans in social navigation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11344–11351, 2022.

[151] R. Salek Shahrezaie, B. N. Manalo, A. G. Brantley, C. R. Lynch, and D. Feil-Seifer, "Advancing socially-aware navigation for public spaces," in *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1015–1022, 2022.

[152] D. Martinez-Baselga, L. Riazuelo, and L. Montano, "Long-range navigation in complex and dynamic environments with full-stack s-dovs," *Applied Sciences*, vol. 13, no. 15, 2023.

[153] C. Theodoridou, D. Antonopoulos, A. Kargakos, I. Kostavelis, D. Giakoumis, and D. Tzovaras, "Robot navigation in human populated unknown environments based on visual-laser sensor fusion," in *Proceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '22, (New York, NY, USA), pp. 336–342, Association for Computing Machinery, 2022.

[154] D. Vasquez, P. Stein, J. Rios-Martinez, A. Escobedo, A. Spalanzani, and C. Laugier, "Human Aware Navigation for Assistive Robotics," in *ISER - 13th International Symposium on Experimental Robotics - 2012*, (Québec, Canada), Jun 2012. The original publication is available at www.springerlink.com.

[155] J. Liang, U. Patel, A. J. Sathyamoorthy, and D. Manocha, "Crowd-Steer: realtime smooth and collision-free robot navigation in densely crowded scenarios trained using high-fidelity simulation," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021.

[156] Z. Xie and P. Dames, "DRL-VO: Learning to navigate through crowded dynamic scenes using velocity obstacles," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2700–2719, 2023.

[157] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 116–121, 1985.

[158] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013. Software available at `https://octomap.github.io`.

[159] D. Belter, P. Łabcki, and P. Skrzypczyński, "Estimating terrain elevation maps from sparse and uncertain multi-sensor data," in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 715–722, 2012.

[160] D. Ferguson and A. Stentz, "Field d*: An interpolation-based path planner and replanner," in *Robotics Research* (S. Thrun, R. Brooks, and H. Durrant-Whyte, eds.), (Berlin, Heidelberg), pp. 239–253, Springer Berlin Heidelberg, 2007.

[161] B. Gerkey and K. Konolige, "Planning and Control in Unstructured Terrain," in *ICRA Workshop on Path Planning on Costmaps*, (Pasadena, CA, USA), may 2008.

[162] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, 11 2016.

[163] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3052–3059, 2018.

[164] U. Patel, N. K. S. Kumar, A. J. Sathyamoorthy, and D. Manocha, "DWA-RL: Dynamically feasible deep reinforcement learning policy for robot navigation among mobile obstacles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6057–6063, 2021.

[165] G. Ferrer and A. Sanfeliu, "Anticipative kinodynamic planning: multi-objective robot navigation in urban and dynamic environments," *Autonomous Robots*, vol. 43, pp. 1473–1488, aug 2019.

[166] E. Repiso, A. Garrell, and A. Sanfeliu, "People's adaptive side-by-side model evolved to accompany groups of people by social robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2387–2394, 2020.

[167] H. Kivrak, F. Cakmak, H. Kose, and S. Yavuz, "Social navigation framework for assistive robots in human inhabited unknown environments," *Engineering Science and Technology, an International Journal*, vol. 24, no. 2, pp. 284–298, 2021.

[168] P. Teja Singamaneni, A. Favier, and R. Alami, "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, 2021.

[169] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, H. Hung, O. A. I. Ramírez, M. Joosse, H. Khambhaita, T. Kucner, B. Leibe, A. J. Lilienthal, T. Linder, M. Lohse, M. Magnusson, B. Okal, L. Palmieri, U. Rafi, M. van Rooij, and L. Zhang, *SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports*, pp. 607–622. Cham: Springer International Publishing, 2016.

[170] D. V. Lu, D. Hershberger, and W. D. Smart, “Layered costmaps for context-sensitive navigation,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 709–715, 2014.

[171] K. Arras, O. Mozos, and W. Burgard, “Using boosted features for the detection of people in 2d range data,” in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3402–3407, 04 2007.

[172] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, “Person tracking and following with 2d laser scanners,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 726–733, 2015.

[173] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 779–788, IEEE Computer Society, jun 2016.

[174] A. Wong, M. J. Shafiee, F. Li, and B. Chwyl, “Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection,” *2018 15th Conference on Computer and Robot Vision (CRV)*, pp. 95–101, 2018.

[175] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, “OpenPose: Real-time multi-person 2d pose estimation using part affinity fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[176] H. Bozorgi, X. T. Truong, and T. D. Ngo, “Reliable, robust, accurate and real-time 2d lidar human tracking in cluttered environment: A social dynamic filtering approach,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11689–11696, 2022.

[177] M. Luber and K. O. Arras, “Multi-hypothesis social grouping and tracking for mobile robots,” in *Robotics: Science and Systems (RSS’13)*, (Berlin, Germany), 2013.

[178] W. K. Juel., F. Haarslev., N. Krüger., and L. Bodenhagen., “An integrated object detection and tracking framework for mobile robots,” in *Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics - ICINCO*, pp. 513–520, INSTICC, SciTePress, 2020.

[179] B. Settles, “Active learning literature survey,” Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[180] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, vol. 3, (Kobe, Japan), 2009.

[181] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[182] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 797–803, 2010.

[183] M. Li, R. Jiang, S. S. Ge, and T. H. Lee, "Role playing learning for socially concomitant mobile robot navigation," *CAAI Transactions on Intelligence Technology*, vol. 3, p. 49–58, apr 2018.

[184] R. Chandra, R. Maligi, A. Anantula, and J. Biswas, "Socialmapf: Optimal and efficient multi-agent path finding with strategic agents for social navigation," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3214–3221, 2023.

[185] S. Russell, "Learning agents for uncertain environments (extended abstract)," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, (New York, NY, USA), pp. 101–103, Association for Computing Machinery, 1998.

[186] R. Bellman, "A markovian decision process," *Indiana University Mathematics Journal*, vol. 6, pp. 679–684, 1957.

[187] P. Henry, C. Vollmer, B. Ferris, and D. Fox, "Learning to navigate through crowded environments," in *2010 IEEE International Conference on Robotics and Automation*, pp. 981–986, 2010.

[188] N. Rhinehart and K. M. Kitani, "First-person activity forecasting with online inverse reinforcement learning," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3716–3725, 2017.

[189] D. Vasquez, B. Okal, and K. O. Arras, "Inverse reinforcement learning algorithms and features for robot navigation in crowds: An experimental comparison," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1341–1346, 2014.

[190] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, (New York, NY, USA), p. 1, Association for Computing Machinery, 2004.

[191] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pp. 1433–1438, AAAI Press, 2008.

[192] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.

[193] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially compliant navigation through raw depth inputs with generative adversarial imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1111–1117, 2018.

[194] M. Goldhammer, K. Doll, U. Brunsmann, A. Gensler, and B. Sick, "Pedestrian's trajectory forecast in public traffic with artificial neural networks," in *2014 22nd International Conference on Pattern Recognition*, pp. 4110–4115, 2014.

[195] J. Gao, Z. Yang, and R. Nevatia, "RED: Reinforced encoder-decoder networks for action anticipation," in *British Machine Vision Conference (BMVC)*, 01 2017.

[196] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: a survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[197] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–971, 2016.

[198] A. Furnari and G. Farinella, "What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Los Alamitos, CA, USA), pp. 6251–6260, IEEE Computer Society, nov 2019.

[199] Z. Chen, C. Song, Y. Yang, B. Zhao, Y. Hu, S. Liu, and J. Zhang, "Robot navigation based on human trajectory prediction and multiple travel modes," *Applied Sciences*, vol. 8, no. 11, 2018.

[200] A. Vemula, K. Muelling, and J. Oh, “Social attention: Modeling attention in human crowds,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7, IEEE Press, 2018.

[201] Y. Farha, A. Richard, and J. Gall, “When will you do what? - anticipating temporal occurrences of activities,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 5343–5352, IEEE Computer Society, jun 2018.

[202] J. Huang, J. Hao, R. Juan, R. Gomez, K. Nakarnura, and G. Li, “Model-based adversarial imitation learning from demonstrations and human reward,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1683–1690, 2023.

[203] T. Kanda, D. F. Glas, M. Shiomi, H. Ishiguro, and N. Hagita, “Who will be the customer? a social robot that anticipates people’s behavior from their trajectories,” in *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp ’08, (New York, NY, USA), pp. 380–389, Association for Computing Machinery, 2008.

[204] S. Xiao, Z. Wang, and J. Folkesson, “Unsupervised robot learning to predict person motion,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 691–696, 2015.

[205] F. Zanlungo, T. Ikeda, and T. Kanda, “Social force model with explicit collision prediction,” *EPL (Europhysics Letters)*, vol. 93, p. 68005, 03 2011.

[206] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, “People tracking with human motion predictions from social forces,” in *2010 IEEE International Conference on Robotics and Automation*, pp. 464–469, 2010.

[207] J. Yue, D. Manocha, and H. Wang, “Human trajectory prediction via neural social physics,” in *Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIV*, (Berlin, Heidelberg), pp. 376–394, Springer-Verlag, 2022.

[208] O. Gil and A. Sanfeliu, “Human motion trajectory prediction using the social force model for real-time and low computational cost applications,” in *6th Iberian Robotics Conference, Portugal*, pp. 1–12, 2023.

[209] A. Elnagar, "Prediction of moving objects in dynamic environments using kalman filters," in *Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No.01EX515)*, pp. 414–419, 2001.

[210] C.-Y. Lin, L.-J. Kau, and C.-Y. Chan, "Bimodal extended kalman filter-based pedestrian trajectory prediction," *Sensors*, vol. 22, no. 21, 2022.

[211] S. Kim, S. J. Guy, W. Liu, D. Wilkie, R. W. Lau, M. C. Lin, and D. Manocha, "Brvo: Predicting pedestrian trajectories using velocity-space reasoning," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 201–217, 2015.

[212] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[213] H. Sakahara, Y. Masutani, and F. Miyazaki, "Safe navigation in unknown dynamic environments with voronoi based strrt," in *Proc. of the IEEE/SICE Int. Symp. on System Integration*, pp. 60–65, 01 2008.

[214] I. Nishitani, T. Matsumura, M. Ozawa, A. Yorozu, and M. Takahashi, "Human-centered x-y-t space path planning for mobile robot in dynamic environments," *Robotics and Autonomous Systems*, vol. 66, pp. 18–26, 2015.

[215] M. Kollmitz, K. Hsiao, J. Gaa, and W. Burgard, "Time dependent planning on a layered social cost map for human-aware robot navigation," in *2015 European Conference on Mobile Robots (ECMR)*, pp. 1–6, 2015.

[216] H. Khambhaita and R. Alami, "A human-robot cooperative navigation planner," in *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '17, (New York, NY, USA), pp. 161–162, Association for Computing Machinery, 2017.

[217] P.-T. Singamaneni and R. Alami, "HATEB-2: Reactive planning and decision making in human-robot co-navigation," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 179–186, 2020.

[218] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.

[219] C. Weinrich, M. Volkhardt, E. Einhorn, and H.-M. Gross, "Prediction of human collision avoidance behavior by lifelong learning for socially compliant robot naviga-

tion," in *2013 IEEE International Conference on Robotics and Automation*, pp. 376–381, 2013.

[220] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *2013 IEEE International Conference on Robotics and Automation*, pp. 2153–2160, 2013.

[221] S. Oli, B. L'Esperance, and K. Gupta, "Human motion behaviour aware planner (hmbap) for path planning in dynamic human environments," in *2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–7, 2013.

[222] G. Ferrer and A. Sanfeliu, "Bayesian human motion intentionality prediction in urban environments," *Pattern Recognition Letters*, vol. 44, pp. 134–140, 2014. Pattern Recognition and Crowd Analysis.

[223] A. Bera, S. Kim, T. Randhavane, S. Pratapa, and D. Manocha, "GLMP- realtime pedestrian path prediction using global and local movement patterns," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5528–5535, 2016.

[224] V. Lim, M. Rooksby, and E. S. Cross, "Social robots on a global stage: Establishing a role for culture during human–robot interaction," *International Journal of Social Robotics*, vol. 13, pp. 1307–1333, Sep 2021.

[225] C. Recchiuto and A. Sgorbissa, "Diversity-aware social robots meet people: beyond context-aware embodied ai," 2022.

[226] P. Bustos, L. Manso, A. Bandera, J. Bandera, I. García-Varea, and J. Martínez-Gómez, "The cortex cognitive robotics architecture: Use cases," *Cognitive Systems Research*, vol. 55, pp. 107–123, 2019.

[227] F. Martín, F. J. Rodríguez Lera, J. Ginés, and V. Matellán, "Evolution of a cognitive architecture for social robots: Integrating behaviors and symbolic knowledge," *Applied Sciences*, vol. 10, no. 17, 2020.

[228] S. B. Banisetty, S. Forer, L. Yliniemi, M. Nicolescu, and D. Feil-Seifer, "Socially aware navigation: A non-linear multi-objective optimization approach," *ACM Trans. Interact. Intell. Syst.*, vol. 11, jul 2021.

[229] R. Salek Shahrezaie, S. B. Banisetty, M. Mohammadi, and D. Feil-Seifer, "Towards deep reasoning on social rules for socially aware navigation," in *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '21

Companion, (New York, NY, USA), pp. 515–518, Association for Computing Machinery, 2021.

[230] Y. Jia, B. Ramalingam, R. E. Mohan, Z. Yang, Z. Zeng, and P. Veerajagadheswar, "Deep-learning-based context-aware multi-level information fusion systems for indoor mobile robots safe navigation," *Sensors*, vol. 23, no. 4, 2023.

[231] A. Vega, L. J. Manso, D. G. Macharet, P. Bustos, and P. Núñez, "Socially aware robot navigation system in human-populated and interactive environments based on an adaptive spatial density function and space affordances," *Pattern Recognition Letters*, vol. 118, pp. 72–84, 2019. Cooperative and Social Robots: Understanding Human Activities and Intentions.

[232] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, 2015.

[233] J. Crespo, J. C. Castillo, O. M. Mozos, and R. Barber, "Semantic information for robot navigation: A survey," *Applied Sciences*, vol. 10, no. 2, 2020.

[234] R. Alqobali, M. Alshmrani, R. Alnasser, A. Rashidi, T. Alhmiedat, and O. M. Alia, "A survey on robot semantic navigation systems for indoor environments," *Applied Sciences*, vol. 14, no. 1, 2024.

[235] J. Zhang, W. Wang, X. Qi, and Z. Liao, "Social and robust navigation for indoor robots based on object semantic grid and topological map," *Applied Sciences*, vol. 10, no. 24, 2020.

[236] P. Núñez, L. Manso, P. Bustos, P. Drews-Jr, and D. Macharet, "Towards a new semantic social navigation paradigm for autonomous robots using cortex," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2016) - BAILAR2016 Workshop*, Aug 2016.

[237] A. Cosgun and H. I. Christensen, "Context-aware robot navigation using interactively built semantic maps," *Paladyn, Journal of Behavioral Robotics*, vol. 9, no. 1, pp. 254–276, 2018.

[238] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Visual social relationship recognition," *International Journal of Computer Vision*, vol. 128, pp. 1750–1764, Jun 2020.

[239] P. Patompak, S. Jeong, I. Nilkhamhang, and N. Y. Chong, "Learning social relations for culture aware interaction," in *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 26–31, 2017.

[240] B. Okal and K. O. Arras, "Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2889–2895, 2016.

[241] F. Haarslev., W. K. Juel., A. Kollakidou., N. Krüger., and L. Bodenhagen., "Context-aware social robot navigation," in *Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics - ICINCO*, pp. 426–433, INSTICC, SciTePress, 2021.

[242] T. Schwörer, J. E. Schmidt, and D. Chrysostomou, "Nav2CAN: Achieving context aware navigation in ros2 using nav2 and rgb-d sensing," in *2023 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 1–6, 2023.

[243] T. Amaoka, H. Laga, and M. Nakajima, "Modeling the personal space of virtual agents for behavior simulation," in *2009 International Conference on CyberWorlds*, pp. 364–370, 2009.

[244] P. Flandorfer, "Population ageing and socially assistive robots for elderly persons: The importance of sociodemographic factors for user acceptance," *International Journal of Population Research*, vol. 2012, p. 829835, Apr 2012.

[245] M. Strait, P. Briggs, and M. Scheutz, "Gender, more so than age, modulates positive perceptions of language-based human-robot interaction," in *4th International Syposium on New Frontiers in Human-Robot Interaction, AISB*, 2015.

[246] T. Nomura, T. Kanda, T. Suzuki, and K. Kato, "Age differences and images of robots," *Interaction Studies*, vol. 10, no. 3, pp. 374–391, 2009.

[247] L. Robert, "Personality in the human robot interaction literature: A review and brief critique," in *Proceedings of the 24th Americas Conference on Information Systems, New Orleans, LA.*, 2018.

[248] J. V. Hurtado, L. Londoño, and A. Valada, "From learning to relearning: A framework for diminishing bias in social robot navigation," *Frontiers in Robotics and AI*, vol. 8, 2021.

[249] L. Chen, M. Wu, M. Zhou, J. She, F. Dong, and K. Hirota, "Information-driven multirobot behavior adaptation to emotional intention in human–robot interaction," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 647–658, 2018.

[250] A. Bera, T. Randhavane, and D. Manocha, "The emotionally intelligent robot: Improving socially-aware human prediction in crowded environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun 2019.

[251] A. Nanavati, X. Z. Tan, J. Connolly, and A. Steinfeld, "Follow the robot: Modeling coupled human-robot dyads during navigation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3836–3843, 2019.

[252] J. Ginés Clavero, F. Martín Rico, F. J. Rodríguez-Lera, J. M. Guerrero Hernández, and V. Matellán Olivera, "Defining adaptive proxemic zones for activity-aware navigation," in *Advances in Physical Agents II* (L. M. Bergasa, M. Ocaña, R. Barea, E. López-Guillén, and P. Revenga, eds.), (Cham), pp. 3–17, Springer International Publishing, 2021.

[253] E. Repiso, A. Garrell, and A. Sanfeliu, "Adaptive side-by-side social robot navigation to approach and interact with people," *International Journal of Social Robotics*, vol. 12, pp. 909–930, 2020.

[254] E. Repiso, F. Zanlungo, T. Kanda, A. Garrell, and A. Sanfeliu, "People's v-formation and side-by-side model adapted to accompany groups of people by social robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2082–2088, 2019.

[255] D. C. Moore, A. S. Huang, M. Walter, E. Olson, L. Fletcher, J. Leonard, and S. Teller, "Simultaneous local and global state estimation for robotic navigation," in *2009 IEEE International Conference on Robotics and Automation*, pp. 3794–3799, 2009.

[256] S. Macenski, M. Booker, and J. Wallace, "Open-source, cost-aware kinematically feasible planning for mobile and surface robotics," 2024.

[257] J. R. Sánchez-Ibáñez, C. J. Pérez-del Pulgar, and A. García-Cerezo, "Path planning for autonomous mobile robots: A review," *Sensors*, vol. 21, no. 23, 2021.

[258] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Systems with Applications*, vol. 227, p. 120254, 2023.

[259] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao, "Review of autonomous path planning algorithms for mobile robots," *Drones*, vol. 7, no. 3, 2023.

[260] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, "A survey of path planning algorithms for mobile robots," *Vehicles*, vol. 3, no. 3, pp. 448–468, 2021.

[261] L. Yang, P. Li, S. Qian, H. Quan, J. Miao, M. Liu, Y. Hu, and E. Memetimin, "Path planning technique for mobile robots: A review," *Machines*, vol. 11, no. 10, 2023.

[262] R. Smierzchalski and Z. Michalewicz, *Path Planning in Dynamic Environments*, pp. 135–153. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[263] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Computing*, vol. 8, pp. 239–287, Jun 2009.

[264] J.-C. Latombe, *Robot Motion Planning.* Springer-Verlag New York, Inc., 1991.

[265] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[266] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[267] A. Stentz, "Optimal and efficient path planning for unknown and dynamic environments," Tech. Rep. CMU-RI-TR-93-20, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, Aug 1993.

[268] A. Stentz, "The focussed d* algorithm for real-time replanning," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, (San Francisco, CA, USA), pp. 1652–1659, Morgan Kaufmann Publishers Inc., 1995.

[269] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning a*," *Artificial Intelligence*, vol. 155, no. 1, pp. 93–146, 2004.

[270] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354–363, 2005.

[271] R. Philippsen and R. Siegwart, "An interpolated dynamic navigation function," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 3782–3789, 2005.

[272] K. Daniel, A. Nash, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," *J. Artif. Intell. Res. (JAIR)*, vol. 39, pp. 533–579, 01 2014.

[273] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

[274] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, "A human aware mobile robot motion planner," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[275] X.-T. Truong and T.-D. Ngo, "„To Approach Humans?": A unified framework for approaching pose prediction and socially aware robot navigation," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 557–572, 2018.

[276] A. Vega-Magro, L. V. Calderita, P. Bustos, and P. Núñez, "Human-aware robot navigation based on time-dependent social interaction spaces: a use case for assistive robotics," in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 140–145, 2020.

[277] F. Melo and P. Moreno, "Socially reactive navigation models for mobile robots," in *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 91–97, 2022.

[278] J. Borenstein and Y. Koren, "High-speed obstacle avoidance for mobile robots," in *Proceedings IEEE International Symposium on Intelligent Control 1988*, pp. 382–384, 1988.

[279] M. Khatib and R. Chatila, "An extended potential field approach for mobile robot sensor-based motions," in *Proceedings of the Intelligent Autonomous Systems IAS-4*, pp. 490–496, IOS Press, Karlsruhe, Germany, Mar 1995.

[280] S. Iizuka, T. Nakamura, and S. Suzuki, "Robot navigation in dynamic environment for an indoor human monitoring," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 698–703, 2014.

[281] T. Weerakoon, K. Ishii, and A. A. F. Nassiraei, "An artificial potential field based mobile robot navigation method to prevent from deadlock," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 5, no. 3, pp. 189–203, 2015.

[282] A. Azzabi and K. Nouri, "An advanced potential field method proposed for mobile robot path planning," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 11, pp. 3132–3144, 2019.

[283] R. Szczepanski, "Safe artificial potential field - novel local path planning algorithm maintaining safe distance from obstacles," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4823–4830, 2023.

[284] S. Garrido, L. Moreno, M. Abderrahim, and F. Martin, "Path planning for mobile robot navigation using voronoi diagram and fast marching," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2376–2381, 2006.

[285] S. Friedman, H. Pasula, and D. Fox, "Voronoi random fields: extracting the topological structure of indoor environments via place labeling," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, (San Francisco, CA, USA), pp. 2109–2114, Morgan Kaufmann Publishers Inc., 2007.

[286] M.-C. Lu, C.-C. Hsu, Y.-J. Chen, and S.-A. Li, "Hybrid path planning incorporating global and local search for mobile robot," in *Advances in Autonomous Robotics* (G. Herrmann, M. Studley, M. Pearson, A. Conn, C. Melhuish, M. Witkowski, J.-H. Kim, and P. Vadakkepat, eds.), (Berlin, Heidelberg), pp. 441–443, Springer Berlin Heidelberg, 2012.

[287] S. M. LaValle, *Planning Algorithms*. USA: Cambridge University Press, 2006.

[288] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[289] S. M. LaValle and J. James J. Kuffner, *Rapidly-Exploring Random Trees: Progress and Prospects*, pp. 293–308. AK Peters/CRC Press, 2001.

[290] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, pp. 995–1001 vol.2, 2000.

[291] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[292] C.-b. Moon and W. Chung, "Kinodynamic planner dual-tree rrt (dt-rrt) for two-wheeled mobile robots using the rapidly exploring random tree," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 2, pp. 1080–1090, 2015.

[293] M. Svenstrup, T. Bak, and H. J. Andersen, "Trajectory planning for robots in dynamic human environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4293–4298, 2010.

[294] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "Understanding human interaction for probabilistic autonomous navigation using Risk-RRT approach," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2014–2019, 2011.

[295] M. C. Shrestha, Y. Nohisa, A. Schmitz, S. Hayakawa, E. Uno, Y. Yokoyama, H. Yanagawa, K. Or, and S. Sugano, "Using contact-based inducement for efficient navigation in a congested environment," in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 456–461, 2015.

[296] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 2262–2269, 2006.

[297] N. Pérez-Higueras, F. Caballero, and L. Merino, "Teaching robot navigation behaviors to optimal rrt planners," *International Journal of Social Robotics*, vol. 10, pp. 235–249, Apr 2018.

[298] N. Pérez-Higueras, R. Ramón-Vigo, F. Caballero, and L. Merino, "Robot local navigation with learned social cost functions," in *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, vol. 02, pp. 618–625, 2014.

[299] C. Lakhmissi and M. Boumehraz, "Fuzzy logic and reinforcement learning based approaches for mobile robot navigation in unknown environment," *Mediterranean Journal of Measurement and Control*, vol. 9, pp. 109–117, 01 2013.

[300] A. Pandey, R. K. Sonkar, K. K. Pandey, and D. R. Parhi, "Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller," in *2014 IEEE 8th International Conference on Intelligent Systems and Control (ISCO)*, pp. 39–41, 2014.

[301] H. Omrane, M. S. Masmoudi, and M. Masmoudi, "Fuzzy logic based control for autonomous mobile robot navigation," *Computational Intelligence and Neuroscience*, vol. 2016, p. 9548482, Sep 2016.

[302] L. M. Zeinalova and B. O. Jafarov, "Mobile robot navigation with preference-based fuzzy behaviors," in *11th International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions and Artificial Intelligence - ICSCCW-2021* (R. A. Aliev, J. Kacprzyk, W. Pedrycz, M. Jamshidi, M. Babanli, and F. M. Sadikoglu, eds.), (Cham), pp. 774–782, Springer International Publishing, 2022.

[303] J. P. Vásconez, M. Calderón-Díaz, I. C. Briceño, J. M. Pantoja, and P. J. Cruz, "A behavior-based fuzzy control system for mobile robot navigation: Design and assessment," in *Advanced Research in Technologies, Information, Innovation and Sustainability* (T. Guarda, F. Portela, and J. M. Diaz-Nafria, eds.), (Cham), pp. 412–426, Springer Nature Switzerland, 2024.

[304] R. Palm., R. Chadalavada., and A. J. Lilienthal., "Fuzzy modeling and control for intention recognition in human-robot systems," in *Proceedings of the 8th International Joint Conference on Computational Intelligence (IJCCI 2016) - FCTA*, pp. 67–74, INSTICC, SciTePress, 2016.

[305] T. Obo and E. Yasuda, "Intelligent fuzzy controller for human-aware robot navigation," in *2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics*, pp. 392–397, 2018.

[306] A. T. Rifqi, B. S. B. Dewantara, D. Pramadihanto, and B. S. Marta, "Fuzzy social force model for healthcare robot navigation and obstacle avoidance," in *2021 International Electronics Symposium (IES)*, pp. 445–450, 2021.

[307] S. K. Sampathkumar, D. Choi, and D. Kim, "Fuzzy inference system-assisted human-aware navigation framework based on enhanced potential field," *Complex Engineering Systems*, vol. 4, no. 1, p. 3, 2024.

[308] P. Glorennec and L. Jouffe, "Fuzzy q-learning," in *Proceedings of 6th International Fuzzy Systems Conference*, vol. 2, pp. 659–662 vol.2, 1997.

[309] Y. Duan and Xin-Hexu, "Fuzzy reinforcement learning and its application in robot navigation," in *2005 International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 899–904 Vol. 2, 2005.

[310] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pp. 802–807 vol.2, 1993.

[311] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.

[312] S. Hoogendoorn, F. Kessels, W. Daamen, and D. Duives, "Continuum modelling of pedestrian flows: From microscopic principles to self-organised macroscopic phenomena," *Physica A: Statistical Mechanics and its Applications*, 08 2014.

[313] B. Liu, H. Liu, H. Zhang, and X. Qin, "A social force evacuation model driven by video data," *Simulation Modelling Practice and Theory*, vol. 84, pp. 190–203, 2018.

[314] X.-T. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," *IEEE Transactions on Automation Science and Engineering*, vol. 14, pp. 1–18, 08 2017.

[315] G. Ferrer, A. Zulueta, F. Cotarelo, and A. Sanfeliu, "Robot social-aware navigation framework to accompany people walking side-by-side," *Autonomous Robots*, vol. 41, pp. 775–793, 04 2017.

[316] I. Karamouzas, P. Heil, P. van Beek, and M. H. Overmars, "A predictive collision avoidance model for pedestrian simulation," in *Motion in Games* (A. Egges, R. Geraerts, and M. Overmars, eds.), (Berlin, Heidelberg), pp. 41–52, Springer Berlin Heidelberg, 2009.

[317] Y.-Q. Jiang, B.-K. Chen, B.-H. Wang, W.-F. Wong, and B.-Y. Cao, "Extended social force model with a dynamic navigation field for bidirectional pedestrian flow," *Frontiers of Physics*, vol. 12, no. 5, p. 124502, 2017.

[318] L. Huang, J. Gong, W. Li, T. Xu, S. Shen, J. Liang, Q. Feng, D. Zhang, and J. Sun, "Social force model-based group behavior simulation in virtual geographic environments," *ISPRS International Journal of Geo-Information*, vol. 7, no. 2, 2018.

[319] J. Sochman and D. C. Hogg, "Who knows who - inverting the social force model for finding groups," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, (Los Alamitos, CA, USA), pp. 830–837, IEEE Computer Society, nov 2011.

[320] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo, "Walking ahead: The headed social force model," *PLOS ONE*, vol. 12, pp. 1–23, 01 2017.

[321] W. Wu, M. Chen, J. Li, B. Liu, and X. Zheng, "An extended social force model via pedestrian heterogeneity affecting the self-driven force," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7974–7986, 2022.

[322] O. Gil, A. Garrell, and A. Sanfeliu, "Social robot navigation tasks: Combining machine learning techniques and social force model," *Sensors*, vol. 21, no. 21, 2021.

[323] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[324] M. Daza, D. Barrios-Aranibar, J. Diaz-Amado, Y. Cardinale, and J. Vilasboas, "An approach of social navigation based on proxemics for crowded environments of humans and robots," *Micromachines*, vol. 12, no. 2, 2021.

[325] M. C. Lin, A. Sud, J. Van den Berg, R. Gayle, S. Curtis, H. Yeh, S. Guy, E. Andersen, S. Patil, J. Sewall, and D. Manocha, "Real-time path planning and navigation for multi-agent and crowd simulations," in *Motion in Games* (A. Egges, A. Kamphuis, and M. Overmars, eds.), (Berlin, Heidelberg), pp. 23–32, Springer Berlin Heidelberg, 2008.

[326] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, pp. 1928–1935, 05 2008.

[327] A.-H. Olivier, A. Marin, A. Crétual, A. Berthoz, and J. Pettré, "Collision avoidance between two walkers: Role-dependent strategies," *Gait & Posture*, vol. 38, no. 4, pp. 751–756, 2013.

[328] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research* (C. Pradalier, R. Siegwart, and G. Hirzinger, eds.), (Berlin, Heidelberg), pp. 3–19, Springer Berlin Heidelberg, 2011.

[329] S. Matsuzaki, S. Aonuma, and Y. Hasegawa, "Dynamic window approach with human imitating collision avoidance," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8180–8186, 2021.

[330] M. Kobayashi, H. Zushi, T. Nakamura, and N. Motoi, "Local path planning: Dynamic window approach with q-learning considering congestion environments for mobile robot," *IEEE Access*, vol. 11, pp. 96733–96742, 2023.

[331] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 1986–1991, 2007.

[332] M. Sebastian, S. B. Banisetty, and D. Feil-Seifer, "Socially-aware navigation planner using models of human-human interaction," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 405–410, 2017.

[333] V. B. Hoang, V. H. Nguyen, T. D. Ngo, and X.-T. Truong, "Socially aware robot navigation framework: Where and how to approach people in dynamic social environments," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 1322–1336, 2023.

[334] S. Forer, S. B. Banisetty, L. Yliniemi, M. Nicolescu, and D. Feil-Seifer, "Socially-aware navigation using non-linear multi-objective optimization," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, 2018.

[335] C. Mavrogiannis, P. Alves-Oliveira, W. Thomason, and R. A. Knepper, "Social momentum: Design and evaluation of a framework for socially competent robot navigation," *J. Hum.-Robot Interact.*, vol. 11, feb 2022.

[336] D. Mehta, G. Ferrer, and E. Olson, "Autonomous navigation in dynamic social environments using multi-policy decision making," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1190–1197, 2016.

[337] Z. Tang, R. Cunha, T. Hamel, and C. Silvestre, "Formation control of a leader-follower structure in three dimensional space using bearing measurements," *Automatica*, vol. 128, p. 109567, 2021.

[338] K. Nguyen, V. T. Dang, D. D. Pham, and P. N. Dao, "Formation control scheme with reinforcement learning strategy for a group of multiple surface vehicles," *International Journal of Robust and Nonlinear Control*, vol. 34, no. 3, pp. 2252–2279, 2024.

[339] J. Truc, P.-T. Singamaneni, D. Sidobre, S. Ivaldi, and R. Alami, "KHAOS: a kinematic human aware optimization-based system for reactive planning of flying-coworker," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 4764–4770, 2022.

[340] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[341] N. Akalin and A. Loutfi, "Reinforcement learning approaches in social robotics," *Sensors*, vol. 21, no. 4, 2021.

[342] B. Kim and J. Pineau, "Socially adaptive path planning in human environments using inverse reinforcement learning," *International Journal of Social Robotics*, vol. 8, pp. 51–66, Jan 2016.

[343] M. Kuderer, H. Kretzschmar, and W. Burgard, "Teaching mobile robots to cooperatively navigate in populated environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3138–3143, 2013.

[344] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, "Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation," *IEEE Robotics and Automation Letters*, vol. 7, pp. 1–8, 10 2022.

[345] M. Bain and C. Sammut, "A framework for behavioural cloning," in *Machine Intelligence 15, Intelligent Agents [St. Catherine's College, Oxford, July 1995]*, (GBR), pp. 103–129, Oxford University, 1999.

[346] G. Silva and T. Fraichard, "Human robot motion: A shared effort approach," in *2017 European Conference on Mobile Robots (ECMR)*, pp. 1–6, 2017.

[347] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 285–292, 2017.

[348] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, IEEE Press, 2017.

[349] J. Jin, N. M. Nguyen, N. Sakib, D. Graves, H. Yao, and M. Jagersand, "Mapless navigation among dynamics with social-safety-awareness: a reinforcement learning approach from 2d laser scans," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6979–6985, 2020.

[350] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6015–6022, 2019.

[351] K. Li, Y. Xu, J. Wang, and M. Meng, "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, 12 2019.

[352] R. Guldenring, M. Görner, N. Hendrich, N. J. Jacobsen, and J. Zhang, "Learning local planners for human-aware navigation in indoor environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6053–6060, 2020.

[353] J. Qin, J. Qin, J. Qiu, Q. Liu, M. Li, and Q. Ma, "SRL-ORCA: A socially aware multi-agent mapless navigation algorithm in complex dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 143–150, 2024.

[354] W. Ding, S. Li, H. Qian, and Y. Chen, "Hierarchical reinforcement learning framework towards multi-agent navigation," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, p. 237–242, IEEE Press, 2018.

[355] X. Lu, H. Woo, A. Faragasso, A. Yamashita, and H. Asama, "Socially aware robot navigation in crowds via deep reinforcement learning with resilient reward functions," *Advanced Robotics*, vol. 36, no. 8, pp. 388–403, 2022.

[356] P. Bachiller, D. Rodriguez-Criado, R. R. Jorvekar, P. Bustos, D. R. Faria, and L. J. Manso, "A graph neural network to model disruption in human-aware robot navigation," *Multimedia Tools and Applications*, vol. 81, pp. 3277–3295, Jan 2022.

[357] C. I. Mavrogiannis, W. B. Thomason, and R. A. Knepper, "Social Momentum: A framework for legible navigation in dynamic multi-agent environments," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, (New York, NY, USA), pp. 361–369, Association for Computing Machinery, 2018.

[358] C. Pérez-D'Arpino, C. Liu, P. Goebel, R. Martín-Martín, and S. Savarese, "Robot navigation in constrained pedestrian environments using reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1140–1146, 2021.

[359] X.-T. Truong and T. D. Ngo, "Dynamic social zone based mobile robot navigation for human comfortable safety in social environments," *International Journal of Social Robotics*, vol. 8, 11 2016.

[360] R. M. d. Sousa, D. Barrios-Aranibar, J. Diaz-Amado, R. E. Patiño-Escarcina, and R. M. P. Trindade, "A new approach for including social conventions into social robots navigation by using polygonal triangulation and group asymmetric gaussian functions," *Sensors*, vol. 22, no. 12, 2022.

[361] A. Corrales-Paredes, D. O. Sanz, M.-J. Terrón-López, and V. Egido-García, "User experience design for social robots: A case study in integrating embodiment," *Sensors*, vol. 23, no. 11, 2023.

[362] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "The interactive museum tour-guide robot," in *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, (USA), pp. 11–18, American Association for Artificial Intelligence, 1998.

[363] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Minerva: a second-generation museum tour-guide robot," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 3, pp. 1999–2005, 1999.

[364] C. Bartneck, T. Belpaeme, F. Eyssel, T. Kanda, M. Keijsers, and S. Šabanović, *Human-Robot Interaction: An Introduction*. Cambridge University Press, 2020.

[365] E. Senft, S. Satake, and T. Kanda, "Would you mind me if i pass by you? socially-appropriate behaviour for an omni-based social robot in narrow environment," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '20, (New York, NY, USA), p. 539–547, Association for Computing Machinery, 2020.

[366] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 261–268, 2009.

[367] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Computer Graphics Forum*, vol. 26, no. 3, pp. 655–664, 2007.

[368] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal, "ThÖr: Human-robot navigation data collection and accurate motion trajectories dataset," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 676–682, 2020.

[369] L. J. Manso, P. Nuñez, L. V. Calderita, D. R. Faria, and P. Bachiller, "Socnav1: A dataset to benchmark and learn social navigation conventions," *Data*, vol. 5, no. 1, 2020.

[370] A. Wang, A. Biswas, H. Admoni, and A. Steinfeld, "Towards rich, portable, and large-scale pedestrian data collection," 2023.

[371] D. Paez-Granados, Y. He, D. Gonon, L. Huber, and A. Billard, “3d point cloud and rgbd of pedestrians in robot crowd navigation: detection and tracking,” 2021.

[372] J. Bae, J. Kim, J. Yun, C. Kang, J. Choi, C. Kim, J. Lee, J. Choi, and J. W. Choi, “Sit dataset: Socially interactive pedestrian trajectory dataset for social navigation robots,” in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[373] D. M. Nguyen, M. Nazeri, A. Payandeh, A. Datar, and X. Xiao, “Toward human-like social robot navigation: A large-scale, multi-modal, social human navigation dataset,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7442–7447, 2023.

[374] C. Camargo, J. Gonçalves, M. Á. Conde, F. J. Rodríguez-Sedano, P. Costa, and F. J. García-Peñalvo, “Systematic literature review of realistic simulators applied in educational robotics context,” *Sensors*, vol. 21, no. 12, 2021.

[375] O. Michel, “Webotstm: Professional mobile robot simulation,” *International Journal of Advanced Robotic Systems*, vol. 1, 03 2004.

[376] N. P. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2149–2154, 2004.

[377] J. Karwowski, W. Dudek, M. Węgierek, and T. Winiarski, “HuBeRo — a framework to simulate human behaviour in robot research,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 15, pp. 31–38, Jul. 2021.

[378] N. Tsoi, A. Xiang, P. Yu, S. S. Sohn, G. Schwartz, S. Ramesh, M. Hussein, A. W. Gupta, M. Kapadia, and M. Vázquez, “Sean 2.0: Formalizing and generating social situations for robot navigation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11047–11054, 2022.

[379] F. Grzeskowiak, D. Gonon, D. Dugas, D. Paez-Granados, J. J. Chung, J. Nieto, R. Siegwart, A. Billard, M. Babel, and J. Pettré, “Crowd against the machine: A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3879–3885, IEEE Press, 2021.

[380] C. Li, F. Xia, R. Martín-Martín, M. Lingelbach, S. Srivastava, B. Shen, K. E. Vainio, C. Gokmen, G. Dharan, T. Jain, A. Kurenkov, K. Liu, H. Gweon, J. Wu, L. Fei-Fei, and S. Savarese, "iGibson 2.0: Object-centric simulation for robot learning of everyday household tasks," in *Proceedings of the 5th Conference on Robot Learning* (A. Faust, D. Hsu, and G. Neumann, eds.), vol. 164 of *Proceedings of Machine Learning Research*, pp. 455–465, PMLR, 08–11 Nov 2022.

[381] A. Favier, P. T. Singamaneni, and R. Alami, "An intelligent human avatar to debug and challenge human-aware robot navigation systems," in *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '22, pp. 760–764, IEEE Press, 2022.

[382] O. Hauterville, C. Fernández, P.-T. Singamaneni, A. Favier, V. Matellán, and R. Alami, "IMHuS: Intelligent Multi-Human Simulator," in *IROS2022 Workshop: Artificial Intelligence for Social Robots Interacting with Humans in the Real World*, (Kyoto, Japan), Oct 2022.

[383] Z. Sprague, R. Chandra, J. Holtz, and J. Biswas, "SOCIALGYM 2.0: Simulator for multi-agent social robot navigation in shared human spaces," 2023.

[384] N. Pérez-Higueras, R. Otero, F. Caballero, and L. Merino, "HuNavSim: A ros 2 human navigation simulator for benchmarking human-aware robot navigation," *IEEE Robotics and Automation Letters*, vol. 8, pp. 7130–7137, September 2023.

[385] A. Toma, H. Hsueh, H. Jaafar, R. Murai, P. J. Kelly, and S. Saeedi, "Pathbench: A benchmarking platform for classical and learned path planning algorithms," in *2021 18th Conference on Robots and Vision (CRV)*, (Los Alamitos, CA, USA), pp. 79–86, IEEE Computer Society, may 2021.

[386] L. Rocha and K. Vivaldini, "Plannie: A benchmark framework for autonomous robots path planning algorithms integrated to simulated and real environments," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 402–411, 2022.

[387] J. Tani, A. F. Daniele, G. Bernasconi, A. Camus, A. Petrov, A. Courchesne, B. Mehta, R. Suri, T. Zaluska, M. R. Walter, E. Frazzoli, L. Paull, and A. Censi, "Integrated benchmarking and design for reproducible and accessible evaluation of robotic agents," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6229–6236, 2020.

[388] D. Mishkin, A. Dosovitskiy, and V. Koltun, "Benchmarking classic and learned navigation in complex 3d environments," 2019.

[389] D. Perille, A. Truong, X. Xiao, and P. Stone, "Benchmarking metric ground navigation," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 116–121, 2020.

[390] L. Kästner, T. Bhuiyan, T. A. Le, E. Treis, J. Cox, B. Meinardus, J. Kmiecik, R. Carstens, D. Pichel, B. Fatloun, N. Khorsandi, and J. Lambrecht, "Arena-bench: A benchmarking suite for obstacle avoidance approaches in highly dynamic environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9477–9484, 2022.

[391] C. Chamzas, C. Quintero-Peña, Z. Kingston, A. Orthey, D. Rakita, M. Gleicher, M. Toussaint, and L. E. Kavraki, "Motionbenchmaker: A tool to generate and benchmark motion planning datasets," *IEEE Robotics and Automation Letters*, vol. 7, pp. 882–889, apr 2022.

[392] S. Tafnakaji, H. Hajieghrary, Q. Teixeira, and Y. Bekiroglu, "Benchmarking local motion planners for navigation of mobile manipulators," in *2023 IEEE/SICE International Symposium on System Integration (SII)*, pp. 1–6, 2023.

[393] I. Dynnikov and B. Wiest, "On the complexity of braids," *Journal of the European Mathematical Society*, vol. 9, no. 4, pp. 801–840, 2007. Version 2: added section on Teichmueller geometry, removed section on train tracks.

[394] F. Xia, W. B. Shen, C. Li, P. Kasimbeg, M. E. Tchapmi, A. Toshev, R. Martín-Martín, and S. Savarese, "Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 713–720, 2020.

[395] P.-T. Singamaneni, A. Favier, and R. Alami, "Towards Benchmarking Human-Aware Social Robot Navigation: A New Perspective and Metrics," in *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 2023*, (Busan, South Korea), Aug 2023.

[396] R. Raj and A. Kos, "A comprehensive study of mobile robot: History, developments, applications, and future research perspectives," *Applied Sciences*, vol. 12, no. 14, 2022.

[397] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*. Reading: Addison-Wesley, 1989.

[398] A. Vega Magro, R. Cintas, L. Manso, P. Bustos, and P. Núñez, *Socially-Accepted Path Planning for Robot Navigation Based on Social Interaction Spaces*, pp. 644–655. Springer International Publishing, 01 2020.

[399] R. Kirby, *Social Robot Navigation.* PhD dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, PA, USA, May 2010.

[400] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, 1991.

[401] S. Macenski, F. Martín, R. White, and J. G. Clavero, "The Marathon 2: A Navigation System," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2718–2725, IEEE Press, 2020.

[402] O. Hosseini Jafari, D. Mitzel, and B. Leibe, "Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 06 2014.

[403] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "A floor and obstacle height map for 3d navigation of a humanoid robot," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1066–1071, 2005.

[404] F. Zanlungo, Z. Yücel, F. Ferreri, J. Even, L. Y. Morales Saiki, and T. Kanda, "Social group motion in robots," in *Social Robotics - 9th International Conference, ICSR 2017, Tsukuba, Japan, November 22-24, 2017, Proceedings* (A. Kheddar, E. Yoshida, S. S. Ge, K. Suzuki, J.-J. Cabibihan, F. Eyssel, and H. He, eds.), vol. 10652 of *Lecture Notes in Computer Science*, (Cham), pp. 474–484, Springer International Publishing, 2017.

[405] A. Schadschneider, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried, *Fundamentals of Pedestrian and Evacuation Dynamics*, pp. 124–154. IGI Global, 06 2009.

[406] V. J. Blue and J. L. Adler, "Emergent fundamental pedestrian flows from cellular automata microsimulation," *Transportation Research Record*, vol. 1644, no. 1, pp. 29–36, 1998.

[407] A. Schadschneider, A. Kirchner, and K. Nishinari, "Ca approach to collective phenomena in pedestrian dynamics," in *Cellular Automata* (S. Bandini, B. Chopard, and M. Tomassini, eds.), (Berlin, Heidelberg), pp. 239–248, Springer Berlin Heidelberg, 2002.

[408] C. Rudloff, T. Matyus, S. Seer, and D. Bauer, "Can walking behavior be predicted? analysis of calibration and fit of pedestrian models," *Transportation Research Record Journal of the Transportation Research Board*, vol. 2264, pp. 101–109, 12 2011.

[409] D. Helbing and A. Johansson, *Pedestrian, Crowd and Evacuation Dynamics*, pp. 6476–6495. New York, NY: Springer New York, 2009.

[410] N. Taherifar, H. Hamedmoghadam, S. Sree, and M. Saberi, "A macroscopic approach for calibration and validation of a modified social force model for bidirectional pedestrian streams," *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1637–1661, 2019.

[411] N. Geroliminis and C. F. Daganzo, "Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings," *Transportation Research Part B: Methodological*, vol. 42, no. 9, pp. 759–770, 2008.

[412] S. Xu and H. Duh, "A simulation of bonding effects and their impacts on pedestrian dynamics," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 153 – 161, 04 2010.

[413] W. Wu, M. Chen, J. Li, B. Liu, and X. Zheng, "An extended social force model via pedestrian heterogeneity affecting the self-driven force," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021.

[414] H. Strasburger, "Seven myths on crowding and peripheral vision," *i-Perception*, vol. 11, no. 3, p. 2041669520913052, 2020. PMID: 32489576.

[415] E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[416] R. Knepper and D. Rus, "Pedestrian-inspired sampling-based multi-robot collision avoidance," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 94–100, 09 2012.

[417] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proceedings 1999 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1322–1328 vol.2, 1999.

[418] L. Kästner, T. Buiyan, L. Jiao, T. A. Le, X. Zhao, Z. Shen, and J. Lambrecht, "Arena-rosnav: Towards deployment of deep-reinforcement-learning-based obstacle avoidance into conventional autonomous navigation systems," in *2021 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), pp. 6456–6463, 2021.

[419] J. Rada-Vilela, "The fuzzylite libraries for fuzzy logic control," 2018.

[420] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On evaluation of embodied navigation agents," 2018.

[421] S. Seer, N. Brändle, and C. Ratti, "Kinects and human kinetics: A new approach for studying pedestrian behavior," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 212–228, 2014.

List of Tables

List of Figures

Appendix A

Asymmetric Gaussian function

The method of calculating the value of the multivariate asymmetric Gaussian function [399], commonly referred to as f_{mag} in Chapter 4 is presented in Alg. 12.

Algorithm 12 Algorithm to compute the value at $(x,\ y)$ of a bivariate Asymmetric Gaussian function with a mean of $(x_c,\ y_c)$, an orientation of θ and variances of var_h, var_s, and var_r

---

$\alpha \leftarrow \arctan 2(y - y_c, x - x_c) - \theta + \frac{\pi}{2}$

Normalize α

if $\alpha \leqslant 0$ **then**

  $\text{var} \leftarrow \text{var}_r$

else

  $\text{var} \leftarrow \text{var}_h$

end if

$a \leftarrow \dfrac{\cos^2\theta}{2 \cdot \text{var}^2} + \dfrac{\sin^2\theta}{2 \cdot \text{var}_s^2}$

$b \leftarrow \dfrac{\sin(2\theta)}{4 \cdot \text{var}^2} - \dfrac{\sin(2\theta)}{4 \cdot \text{var}_s^2}$

$c \leftarrow \dfrac{\sin^2\theta}{2 \cdot \text{var}^2} + \dfrac{\cos^2\theta}{2 \cdot \text{var}_s^2}$

return $e^{-(a(x-x_c)^2 + 2b(x-x_c)(y-y_c) + c(y-y_c)^2)}$

Appendix B

Gaussian function in the angular domain

The algorithm of calculating the value of the univariate Gaussian function appointed in the normalised angle domain, f_{ang} is presented in Alg. 13.

Algorithm 13 Algorithm to compute the value at x of the angle-based univariate Gaussian function with a mean of μ and a standard deviation of σ

$g_1 \leftarrow \textsc{calculateGaussian}(x, \mu, \sigma)$
$g_2 \leftarrow \textsc{calculateGaussian}(x, \mu - 2\pi, \sigma)$
$g_3 \leftarrow \textsc{calculateGaussian}(x, \mu + 2\pi, \sigma)$
return $\max(g_1, g_2, g_3)$
function $\textsc{calculateGaussian}(x,\ \mu,\ \sigma)$
    return $\dfrac{1}{\sigma\sqrt{2\pi}} e^{-\dfrac{(x-\mu)^2}{2\sigma^2}}$
end function