Thèse n° 10 034

Geometric and Learning Methods to Navigate in Human Crowds with Application to Smart Mobility Devices

Présentée le 30 juin 2023

Faculté des sciences et techniques de l'ingénieur
Laboratoire d'algorithmes et systèmes d'apprentissage
Programme doctoral en robotique, contrôle et systèmes intelligents

pour l'obtention du grade de Docteur ès Sciences

par

David Julian GONON

Acceptée sur proposition du jury

Dr R. Boulic, président du jury
Prof. A. Billard, directrice de thèse
Dr J. Pettré, rapporteur
Prof. J. Alonso-Mora, rapporteur
Prof. A. Martinoli, rapporteur

École
polytechnique
fédérale
de Lausanne

2023

Abstract

The thesis at hand is concerned with robots' navigation in human crowds. Specifically, methods are developed for planning a mobile robot's local motion between pedestrians, and they are evaluated in experiments where a robot interacts with real pedestrians as well as in simulations of a crowd and a robot. The thesis is divided in three main contributions.

The first contribution describes a novel method for non-holonomic robots of convex shape to avoid imminent collisions with moving obstacles. The method assists navigation by correcting steering from the robot's path planner or driver. Its performance is evaluated using a custom simulator, which replicates real crowd movements from a campus dataset, and corresponding metrics which quantify agents' efficiency, the robot's impact on the crowd, and the number of collisions. Further, the method is implemented and evaluated on the standing wheelchair Qolo. In the experiments performed, it drives in autonomous mode using on-board sensing (LiDAR, RGB-D camera and a system to track pedestrians). It avoids collisions with up to five pedestrians and passes through a door.

The second contribution studies the Acceleration Obstacle (AO) for enabling a robot's navigation in human crowds. The AO's geometric properties are analyzed and a direct sampling-free algorithm is proposed to approximate its boundary by linear constraints. The resulting controller is formulated as a quadratic program and evaluated in interaction with simulated bi-directional crowd flow in a corridor. A comparison to alternative robotic controllers is carried out, considering the robot's and the crowd's performance and the robot's behavior with respect to emergent lanes. Results indicate that the robot can achieve higher efficiency outside lanes.

In the third contribution, the problem for a mobile robot to navigate seamlessly in a human crowd is treated by an inverse reinforcement learning (IRL) approach. A novel feature is proposed to model costs of anticipated collisions between agents. The feature approximates agents' pairwise interaction energy, a function which prior work has derived empirically from crowd data as an interaction potential driving pedestrians' mutual avoidance. Using a recent framework to perform IRL from locally optimal examples in continuous space, cost functions which incorporate the novel feature are learned efficiently from high-dimensional examples of real crowd motion. Examples are obtained from two public datasets containing pedestrians' and wheelchair users' trajectories. The

learned models are evaluated and compared in how accurately their local optima model the training examples and test examples. Furthermore, predictions based on test examples' initial states only are generated similarly by optimization, and their distance to recorded ground truth is measured. Both models' predictions compare favorably to a recent related approach from the literature. Finally, a control system which computes and executes in real-time an optimal trajectory according to the learned cost functions is implemented on a robotic wheelchair, to steer it between pedestrians perceived by an on-board tracking system. The robot is deployed on campus, where the controller's performance is evaluated qualitatively. Results show that the approach often generates apt motion plans, which complement pedestrians' motion in an efficient manner, albeit oscillations between locally optimal solutions may occur.

Finally, a comparison of the proposed methods is carried out in simulation, and their differences in terms of performance and underlying assumptions are discussed.

Résumé

La présente thèse porte sur la navigation des robots dans les foules humaines. Plus précisément, des méthodes sont développées pour planifier le mouvement local d'un robot mobile entre les piétons, et elles sont évaluées dans des expériences où un robot interagit avec des piétons réels ainsi que dans des simulations d'une foule et d'un robot. La thèse est divisée en trois contributions principales.

La première contribution décrit une nouvelle méthode pour les robots non holonomes de forme convexe afin d'éviter les collisions imminentes avec des obstacles en mouvement. La méthode aide à la navigation en corrigeant la direction du planificateur de trajectoire du robot ou du conducteur. Sa performance est évaluée à l'aide d'un simulateur sur mesure, qui reproduit les mouvements d'une foule réelle à partir d'un ensemble de données provenant d'un campus, et des mesures correspondantes qui quantifient l'efficacité des agents, l'impact du robot sur la foule et le nombre de collisions. En outre, la méthode est mise en œuvre et évaluée sur le fauteuil roulant Qolo. Dans les expériences réalisées, il se déplace en mode autonome en utilisant les capteurs embarqués (LiDAR, caméra RGB-D et système de suivi des piétons). Il évite les collisions avec un maximum de cinq piétons et franchit une porte.

La deuxième contribution étudie l'Obstacle d'Accélération (OA) pour permettre la navigation d'un robot dans les foules humaines. Les propriétés géométriques de l'OA sont analysées et un algorithme direct sans échantillonnage est proposé pour approximer sa frontière par des contraintes linéaires. Le système de commande qui en résulte est formulé sous la forme d'un programme quadratique et évalué en interaction avec un flux de foule bidirectionnel simulé dans un couloir. Une comparaison avec d'autres contrôleurs robotiques est effectuée, en tenant compte des performances du robot et de la foule, ainsi que du comportement du robot par rapport aux voies émergentes. Les résultats indiquent que le robot peut atteindre une plus grande efficacité en dehors des voies émergentes.

Dans la troisième contribution, le problème de la navigation d'un robot mobile dans une foule humaine est traité par une approche d'apprentissage par renforcement inverse (ARI). Une nouvelle caractéristique est proposée pour modéliser les coûts des collisions anticipées entre les agents. Cette caractéristique se rapproche de l'énergie d'interaction par paire des agents, une fonction que des travaux antérieurs ont dérivée empiriquement à partir de données de foule comme un potentiel d'interaction conduisant à l'évitement mutuel des piétons. En utilisant un cadre récent pour effectuer l'ARI à partir d'exemples locale-

ment optimaux dans des espacse continus, les fonctions de coût qui intègrent la nouvelle caractéristique sont apprises efficacement à partir d'exemples à haute dimension du mouvement réel de la foule. Les exemples proviennent de deux ensembles de données publiques contenant des trajectoires de piétons et d'utilisateurs de fauteuils roulants. Les modèles appris sont évalués et comparés en fonction de la précision avec laquelle leurs optima locaux modélisent les exemples d'apprentissage et les exemples de test. En outre, les prédictions basées sur les seuls états initiaux des exemples de test sont générées de manière similaire par optimisation, et leur distance par rapport à la vérité de terrain enregistrée est mesurée. Les prédictions des deux modèles se comparent favorablement à une approche connexe récente de la littérature. Enfin, un système de contrôle qui calcule et exécute en temps réel une trajectoire optimale en se basant sur les fonctions de coût apprises est implémenté sur un fauteuil roulant robotisé, afin de le diriger entre les piétons perçus par un système de suivi embarqué. Le robot est déployé sur le campus, où les performances du contrôleur sont évaluées qualitativement. Les résultats montrent que l'approche génère souvent des plans de mouvement appropriés, qui complètent le mouvement des piétons de manière efficace, bien que des oscillations entre les solutions localement optimales puissent se produire.

Enfin, une comparaison des méthodes proposées est effectuée par simulation, et leurs différences en termes de performance et d'hypothèses sous-jacentes sont discutées.

Zusammenfassung

Die vorliegende Arbeit befasst sich mit mobilen Robotern, welche zwischen Fussgängern navigieren. Es werden Methoden entwickelt zur Planung von lokalen Bewegungen eines mobilen Roboters zwischen Fussgängern. Diese Methoden werden evaluiert in physischen sowie simulierten Experimenten, worin ein Roboter mit Fussgängern interagiert. Die Arbeit ist in drei Hauptbeiträge gegliedert.

Der erste Beitrag beschreibt eine neuartige Methode für nicht-holonome Roboter mit konvexem Grundriss zur Vermeidung bevorstehender Kollisionen. Die Methode wird in einer Simulation evaluiert, welche auf aufgezeichneten Bewegungen echter Menschenmengen basiert. Zur Evaluation werden Kenngrössen berechnet, welche die Effizienz des Roboters und der Fussgänger, den Einfluss des Roboters auf die Geschwindigkeit der Fussgänger und die Anzahl der Kollisionen quantifizieren. Ausserdem wird die Methode auf dem Stehrollstuhl Qolo implementiert, um damit eine experimentelle Auswertung durchzuführen. Darin fährt der Stehrollstuhl Qolo autonom zwischen statischen Hindernissen und Fussgängern, welche er mittels Sensoren (LiDAR und RGB-D Kamera) wahrnimmt.

Der zweite Beitrag untersucht das Beschleunigungshindernis als ein Konzept zur Bewegungsplanung für mobile Roboter, welche sich in Menschenmengen bewegen. Dessen geometrische Eigenschaften werden untersucht, und ein direkter Algorithmus zur Konstruktion einer annähernden Halbebene wird vorgeschlagen. Davon ausgehend wird zur Steuerung eines Roboters ein quadratisches Optimierungsproblem formuliert. Die Steuerung wird in Simulationen evaluiert, worin ein Roboter mit Fussgängerströmen in einem Korridor interagiert. Ein Vergleich mit alternativen Steuerungsmethoden wird durchgeführt, wobei die Effizienz des Roboters und der Fussgänger sowie die Interaktion des Roboters mit dynamisch entstehenden Bahnen ausgewertet werden. Die Resultate zeigen, dass der Roboter ausserhalb solcher Bahnen eine höhere Effizienz erreicht.

Im dritten Beitrag wird das Problem eines Roboters, sich fliessend in einer Menschenmenge zu bewegen, durch einen Ansatz des inversen verstärkenden Lernens behandelt. Ein neues Merkmal zur Modellierung von Kosten antizipierter Kollisionen wird vorgeschlagen. Das Merkmal nähert die paarweise Interaktionsenergie an, eine Funktion welche in einer vorangehenden Arbeit empirisch aus aufgezeichneten Menschenmengenbewegungen hergeleitet wurde als ein Potenzial, welches Fussgänger bei der Kollisionsvermeidung antreibt. Mittels einer kürzlich entwickelten Methode für das inverse verstärkende Lernen

von lokal optimalen Beispielen in kontinuierlichen Räumen werden Kostenfunktionen, welche das neue Merkmal beinhalten, effizient von hochdimensionalen realen Beispielen von Fussgängerbewegungen gelernt. Solche Beispiele werden zwei öffentlichen Datensätzen entnommen, welche die Bewegungen von Fussgängern und Personen in Rollstühlen beinhalten. Die gelernten Modelle werden evaluiert und verglichen bezüglich der Genauigkeit, mit welcher ihre lokalen Optima die Trainings- und Testbeispiele modellieren. Ausserdem werden Vorhersagen für Bewegungen gemacht durch Kostenoptimierung bei gegebenem Anfangszustand, und deren Entfernung von den tatsächlichen Bewegungen werden gemessen. Die Vorhersagen unserer Modelle sind gesamthaft gesehen genauer als jene einer verwandten Methode aus der Literatur. Schliesslich wird ein Steuerungssystem, welches eine optimale Trajektorie gemäss der gelernten Kostenfunktionen in Echtzeit berechnet, auf dem Stehrollstuhl Qolo implementiert. Mit einem LiDAR-basierten Objektverfolgungssystem zur Wahrnehmung von Fussgängern ausgestattet, fährt Qolo für eine qualitative Evaluation der Steuerung autonom durch einen Korridor auf dem Campus. Die Resultate zeigen, dass die Steuerung oftmals passende Bewegungspläne generiert, welche die Bewegungen der Fussgänger effizient ergänzen, obgleich Schwankungen zwischen lokal optimalen Lösungen auftreten können.

Zuletzt werden die vorgeschlagenen Methoden miteinander verglichen in simulierten Experimenten, und ihre Unterschiede bezüglich ihrer Leistungsfähigkeit und der ihnen zugrunde liegenden Annahmen werden diskutiert.

Acknowledgement

I would like to thank Prof. Aude Billard for supervising my doctorate and critically reading and discussing my work. Furthermore, I am grateful to her for assigning me to exciting projects and connecting me with stimulating research communities. Also, I am thankful for being able to use her lab's equipment and infrastructure.

I also thank Dr. Diego Paez-Granados for advising me and working with me. Furthermore, I would like to thank him for building and developing the standing wheelchair Qolo. Finally, I am also grateful to him for his organizational efforts that enabled our experiments on campus and in the city of Lausanne.

I would like to thank the lab's secretary Joanna Erfani for her help with administrative tasks, such as booking rooms, trips, or expense reimbursements.

I would like to thank the jury of my private defence, namely Dr. Julien Pettré, Dr. Ronan Boulic, Prof. Alcherio Martinoli, and Dr. Javier Alonso-Mora for their constructive and critical comments and suggestions to improve my thesis.

I would like to thank my colleagues for interesting discussions, helpful comments, and advice in all kinds of practical matters.

I would like to thank my friends and my family for their support.

This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 779942 ("Crowdbot") and by the Hasler Foundation, Switzerland.

Contents

Chapter 1

Introduction

Public spaces typically serve a variety of populations and purposes in parallel, which necessitates coordination between different groups or individuals. Furthermore, regulations, societal norms, customs, and informal agreements shape the ways by which people pursue their objectives in such spaces. When making decisions there, people consider such external conditions as well as subjective goals, while reasoning not only about the present situation but also anticipating their actions' consequences in the near future. Encounters may lead to interactions of almost arbitrary complexity, involving implicit exchange of information, explicit communication, cooperation, and speculation about other people's intentions. The present thesis considers the problem for a mobile robot to navigate around pedestrians in a manner which is safe and allows all involved agents to move smoothly and efficiently.

Intelligent ground vehicles of similar diameter and speed as pedestrians have made their way in human-populated spaces. Smart wheelchairs and delivery robots are prime examples for such partially or fully autonomous systems. In contrast to autonomous cars, they operate in close proximity to pedestrians and use the same pathways, where traffic rules do not apply. Furthermore, in pedestrian zones, one encounters a much greater variety of floor layouts, human activity, and stationary objects than on designated pathways for cars. Navigation in such weakly structured environments involves subtle interactions with other agents and benefits strongly from situational awareness. Only by continuously monitoring the surrounding, by correctly interpreting its condition, and by leveraging specific prior knowledge about an environment and its population, an autonomous system can navigate successfully, safely, and efficiently there.

Since mobility is a fundamental need for humans and instrumental to participating in many econonomical and social activities, advances in such vehicles are of high interest to society. Particularly, persons with disabilities may gain further access and capabilities with such smart vehicles. While skilled users of manual wheelchairs can achieve high agility and adapt quickly to their environment, powered wheelchairs typically exhibit a higher weight and corresponding inertia, and their control interfaces are more limited in precision and bandwidth.

Therefore, users of powered wheelchairs may be unable to adapt to pedestrians in dense crowds, which may be too hectic or unpredictable to allow safe navigation. Thus, technical developments to extend safe and efficient mobility to all user groups are desirable.

Wheelchair users navigating in a crowded environment face a challenging task, as they must not only avoid erratic pedestrians who obstruct their way, but, at the same time, find a path that leads between immobile obstacles towards the goal. Smart wheelchairs feature automatic high-level control systems to augment their users' input. Traditionally, they are designed to assist navigation in environments which are static, for example domestic environments. In contrast, when navigating in human crowds, one must cope with highly dynamic, not fully predictable, and interacting agents. They are challenging to perceive, avoid and coordinate with for a smart wheelchair's on-board perception and control system.

A powered wheelchair that moves fast in close proximity to pedestrians constitutes a hazard, as it may collide with them. A collision may injur the wheelchair user as well as the involved pedestrians, due to the wheelchair's high weight, speed and rigidity. Thus, a smart wheelchair must reliably avoid any collision when driving in a dense crowd. This becomes even more important in the presence of vulnerable populations, such as children, pregnant women, and older adults.

The present thesis aims towards a better understanding of interactions which occur between pedestrians and which enable them to navigate smoothly and efficiently in close proximity to each other. Similarly, the thesis aims at techniques which enable mobile robots to navigate among pedestrians with a comparable level of skill. Thus, the state of the art in mobile robotics, focusing on crowded environments, is reviewed in the following Section 1.1. There, research gaps in joint crowd–robot navigation are identified. Finally, the Section 1.2 summarizes the present thesis' scientific contributions regarding these open issues. Also, the main assumptions underlying the thesis are stated. This is followed by a list of previous publications of the contents presented in the thesis' subsequent chapters, and by an explanation of the thesis' structure.

1.1 Related Work

Several research areas provide useful concepts, models, and techniques for enabling robotic systems to navigate in pedestrian zones. These include not only traditional areas of robotics, such as localization, mapping, object tracking, motion planning, and obstacle avoidance, but also the sciences of human walking behavior and crowd dynamics. Before reviewing these individual fields, a general overview on prior work tackling robotic navigation in crowds is given.

1.1.1 Robotic Navigation in Crowded Environments

Prior work focused on different challenges associated with robots' navigation in human crowds. According to a survey [1], a large body of work has focused on social aspects, aiming at robots that interact with humans in a comfortable, natural, and sociable manner while navigating through populated spaces. Another focus present in many prior works concerns the robot's navigation itself, i.e. motion planning between agents that interact with the robot, as documented by recent surveys [2, 3]. A third important aspect tackled by many works consists in perceiving and tracking pedestrians, as another recent survey [4] shows.

Different applications motivated researchers to work towards robots capable of autonomous navigation in crowded environments. On the one hand, applications as transportation systems have been an important motivation, such as given by smart wheelchairs [5], intelligent personal mobility devices [6], and delivery robots [3, 4]. On the other hand, an important application has been seen in robots that guide pedestrians, e.g. as museum guides [7], which operate under normal conditions, or as systems for desaster mitigation [8], which guide people to nearest exits or improve crowd flow dynamics during emergency situations.

1.1.2 Robotic Perception

As a basis for any intelligent behavior, mobile robots must be able to perceive their own motion, their static environment, and dynamic objects around them.

Ego-motion estimation

As a robot perceives its environment using on-board sensors, such as cameras and laserscanners, its perception is relative to itself. Therefore, it is vital to estimate the system's own motion independently, in order to retrieve any perceived object's actual motion by subtracting the robot's own motion from the perceived motion. Commonly, the robot's ego-motion is estimated by fusing and integrating acceleration and velocity estimates over time, obtained from inertial measurement units and odometers (see e.g. [9]). Integrating these estimates results in an estimate of the system's change of position over time. However, integration over longer periods of time accumulates errors present in the velocity and acceleration estimates, and therefore, the estimated trajectory deviates more and more from the robot's actual trajectory. To correct for this, it is necessary to localize the robot with respect to some fixed features of the environment, as will be discussed next.

Simultaneous Localization and Mapping

The robot can estimate its own location and, at the same time, build a map of its environment by using e.g. on-board vision [10], laserscans [11] or other measurements which contain information on the robot's location relative to some known references. This process is known as simultaneous localization and mapping (SLAM).

The constructed map can then be used to plan a path from the robot's current location to its goal. For example, its goal may be a particular landmark or a particular exit of a room, and the robot needs to continuously monitor its own position and orientation relative to this goal in order to direct itself towards the goal and notice when it has reached the goal. A frame of reference can be given a priori, for example as a map, or it can be constructed by the robot itself from its perception of the environment. Also, it is possible to build a local map and to register it to a known map by matching their geometric features [11]. Thus, by simultaneously localizing itself and mapping its environment, the robot acquires knowledge enabling it to plan an efficient path from its current location around static obstacles to its goal.

Detection and Tracking of Dynamic Agents

It is crucial for an autonomous system to distinguish between static parts of its environment, such as walls, pillars, or benches, and dynamic agents, such as humans, vehicles, or dogs. In doing so, the robot may ensure to include in its map only really static elements and thereby keep the option of planning a path through space which is currently occupied by dynamic agents. More importantly, by identifying dynamic agents and observing their individual motion, the robot may speculate about their behavior in the future and its own interaction with them. Finally, the robot may implement specific behaviors with respect to specific types of agents, for example, it may be additionally cautious around humans, especially if they are vulnerable populations.

Tracking-by-detection [12] denotes a common paradigm for designing a tracking system, where the task is divided in two steps at every update, namely detection and data association. Given a new frame (e.g. an image), in the first step, objects of interest are detected in the given frame by hand-crafted or data-driven techniques. In the second step, detections are associated with targets that presumably generated those detections, thereby linking detections in different frames that seem to belong to the same target.

1.1.3 Obstacle Avoidance, Path and Motion Planning

Avoiding obstacles is among the most basic capabilities which a mobile robot must possess in order to navigate successfully through an environment. Additionally, completing a navigation task requires to progress towards a given goal and to reach it eventually, after starting from a given initial location. For enabling a mobile robot to achieve both together, i.e. to reach its destination while avoiding obstacles, different paradigms have been developed in robotics. The list below defines three important paradigms, and gives examples and explanations, in order to guide readers through the remainder of this section.

- Under the paradigm of *reactive control* (or reactive navigation), the robot's behavior at any given instant is defined by a closed form rule or an optimization problem, which is formulated with reference to the robot's and

obstacles' states and actions at the current instant only. Typical approaches in closed form include artificial potential methods [13,14], methods based on dynamical systems [15–17]. Also vision-based navigation algorithms [18] can be treated as reactive control, since they respond directly to visual input. A classical type of optimization-based reactive control uses the Velocity Obstacles [19,20] to define constraints on admissible choices of the robot's velocity. We note that VO can be considered both among reactive techniques and motion planning (see below).

Despite using local information only, global properties of generated trajectories can be guaranteed sometimes, e.g. convergence to the goal [15–17].

We will briefly compare approaches in closed form against optimization-based approaches.

- **Computational complexity** for control laws in closed form tends to be lower than for those relying on optimization. Typically, closed form rules have a fixed runtime, which scales linearly with the number of obstacles, whose contributions can be accumulated when computing the control law [13–17]. Optimization-based approaches also require, in a first step, to compute a contribution by each obstacle, however in the form of a constraint. In a second step, an optimization problem is solved, taking into account each obstacle's constraint [19,20]. Thus, the first step may be of comparable complexity as computing obstacles' contributions for closed form rules, whereas the second step adds an additional burden, which is not negligible in contrast to algebraic operations by which approaches in closed form combine obstacles' contributions. Common solvers for such optimization problems may rely on randomization of constraints and incur an expected runtime that scales linearly with the number of constraints (see e.g. the algorithm of incremental linear programming [21]). Thus, at least in terms of expected runtime, both classes of controllers exhibit linear scaling with the number of obstacles, but the actual runtime in a particular execution of an optimization-based approach may be longer or shorter than expected, whereas closed form approaches yield a fixed runtime. Furthermore, it has to be noted that optimization-based approaches may require additional computations if the problem turns out to be infeasible due to conflicting constraints. To handle such a case, a slack variable may be introduced, which renders constraints soft and thereby guarantees that the corresponding optimization problem is feasible [20]. Accordingly, the new problem has to be solved after the original problem was found infeasible, which adds an additional burden, considering also that the new problem's dimensionality is increased from two to three dimensions due to the slack variable. The same solution algorithm can be used, and its expected runtime still scales linearly with the number of obstacles, although with a higher larger scaling factor due to the higher dimensionality [21].

- **Guarantees of collision avoidance** for approaches in closed form are often given at the level of positions [13–17], i.e. they guarantee that trajectories are collision-free. Optimiziation-based approaches [19, 20] may try to achieve stronger guarantees, e.g. ensuring that the robot attains only velocities which it could maintain for at least a fixed duration before a collision would occur. However, constraints which are derived from such a principle are not necessarily feasible in the presence of multiple obstacles. As discussed above, a remedy is found in softening the constraints, which invalidates the desired guarantees by consequence. In contrast, when constraints are feasible, the underlying anticipatory principle contributes to the robot's safety, as it ensures sufficient time for braking, assuming that obstacles do not change course towards a collision. While this assumption may be violated e.g. by a pedestrian who overlooks the robot, such an approach still forces the robot to adapt immediately and in an anticipatory fashion to unforeseen motions that would lead to a collision in the near future. For this reason, and also due to the increased perceived safety that comes with anticipatory behavior, the present thesis favors such approaches.
- **Guarantees of convergence to a goal** for closed form approaches are typically assuming static environments [13–17]. Even if dynamic obstacles are present and guarantees are only given for static environments, convergence can be achieved, assuming that the dynamic obstacles' are not actively preventing the robot from passing around them. Optimization-based approaches [19, 20] tend to neglect this aspect entierly, i.e. they do not even guarantee convergence in static environments. This thesis disregards convergence issues as well, since they are associated with static obstacles rather than dynamic and interacting obstacles, such as pedestrians, and could be tackled by integrating any prior approach (e.g. closed form or planning-based) with the methods for dynamic obstacles considered in this thesis.

To conclude the above comparison, it can be said that closed form and optimization-based approaches complement each other. Namely, when computational resources are scarce or it is unnecessary to plan according to a specific anticipatory principle, closed-form approaches are preferrable. In contrast, optimization-based approaches are suitable for cases where powerful computation is available and anticipatory planning is desirable.

- Under the paradigm of *path planning*, an optimization problem is formulated with reference to the robot's path (i.e. a sequence of positions without a specification of time), which is required to connect the given initial position and the goal while avoiding static obstacles with a sufficient margin. For fast path planning, sampling-based techniques can be used, such as rapidly exploring random trees [22], where trajectories are built incrementally by sampling random control inputs. A computation-

ally heavier approach is given by the A* algorithm [23], which discretizes space into grid cells and guarantees to find the optimal path.

Note that, since the environment may not be known completely or the robot's map may not be fully accurate, the robot may need to update its plan on the fly, as soon as new information is available. However, even if a planned path is feasible and optimal, the robot will need to deviate from it whenever a dynamic agent intercepts it.

- Under the paradigm of *motion planning*, an optimization problem is formulated with reference to the robot's and obstacles' motion over a time span. By solving this problem, the robot's trajectory over the time span is obtained, which defines the robot's behavior at particular instants. Typical examples are given by VO [19, 20], which assume linear motion, or alternative approaches assuming motion on circular arcs [24, 25]. Computationally heavier approaches optimize trajectories over several time steps, e.g. in the framework of model predictive control [26].

  Thus, key characteristics of such approaches consist in anticipation of current actions' effects in the future and the ability to incorporate kinematic and dynamic constraints. In particular, conflicts between satisfying acceleration limits and avoiding obstacles can be foreseen and avoided, and spaces of non-linear trajectories can be searched [24, 25]. Furthermore, the robot may account for obstacles' motion in the near future, if their motion is known [19, 20].

These paradigms can also be combined, e.g. by first planning a path or a long term motion for the robot, and then using a reactive control policy to avoid any obstacles which the robot encounters unexpectedly while executing the plan. It can be argued that reactive control is not clearly distinguishable from single-step motion planning, which merely extrapolates the current state in the future dependent on the action chosen, such as in methods based on the Velocity Obstacle [19, 20]. In any case, a reactive controller or a motion planner with a high update rate is required for a safe system, as it allows the robot to adapt its motion quickly in situations where a collision is immminent, in order to deviate to a non-colliding course.

Accounting for Uncertainty

If moving obstacles are present, their behavior in the future may be partially unknown. Furthermore, any obstacle's state may be uncertain due to the robot's imperfect perception. To achieve safety or performance guarantees in such cases, motion planning must be based on a model which describes any uncertainty concerning obstacles' states and behavior. For example, such a model may bound obstacles' acceleration [27] or describe their states and behavior in a probabilistic fashion [28, 29]. During motion planning, such uncertainty can then be resolved, e.g., by looking to ensure that the worst possible outcome does not violate safety constraints [27] or by optimizing a cost's expectation value [28, 29].

Accounting for Interactions

If obstacles are also intelligent agents, such as pedestrians, they may interact with the robot and with each other. There are different ways by which motion planning can account for such interactions. Naively, one could first predict obstacles' motion in the future, e.g. by assuming that they will simply maintain their current velocities, and then perform deterministic motion planning with respect to the predicted trajectories, e.g. using Velocity Obstacles [19]. A more elaborate approach would predict obstacles' behavior still independently from the robot's behavior but treat it as uncertain, thereby accounting for their interactions. Thus, the robot's motion could be planned with respect to the specific model of uncertainty, e.g. to ensure safety under worst case assumptions [27], as described in the previous section. However, the underlying idea still exhibits a fundamental flaw, as it first predicts pedestrians' trajectories, and then determines a complementary trajectory for the robot, without considering whether the robot's planned trajectory alters the chances of the pedestrians' predicted trajectories. Attempting to resolve this issue while performing prediction and planning in a sequential fashion would result in circular reasoning.

A self-consistent principle for motion planning in the presence of interacting agents can be found in reasoning jointly about all interacting agents' motions. Instead of introducing some hierarchy or sequence in determining individual motions, a principle is required which allows to find a plausible combination of individual actions. For instance, one could postulate that all agents will contribute equally to collision avoidance. This idea underlies the approach of Optimal Reciprocal Collision Avoidance (ORCA) [20], which determines individual actions such that they satisfy constraints that enforce balanced minimum contributions to collision avoidance. Thus, it can be seen that such approaches are equivalent to assuming the perspective of multi-agent motion planning, where the robot is considered as just one particular agent among multiple others.

Multi-agent navigation

The perspective of multi-agent navigation is the most adequate for a robot's decision making within a system of interacting agents, as argued in the previous section, as well as by other authors [2]. Then, a navigation problem for the multi-agent system can be formulated, whose solution determines all agents' nominal behavior, and the robot's behavior in particular. Such approaches can also incorporate uncertainty of individual intentions and actions, e.g. in the framework of a partially observable markov decision process [29]. The idea to jointly consider individual agents' motion and to look for reciprocal or mutually compatible maneuvers is a key concept for many recent approaches to robots' navigation in crowds, as will be discussed below.

To tackle multi-agent navigation tasks, centralized and de-centralized methods have been proposed. While centralized methods determine all agents' actions by solving a single, possibly large-scale problem, de-centralized methods exploit the structure of their underlying problem formulation to allow its de-

composition into individual sub-problems. Centralized methods include probabilistic models, which describe the system's behavior by a joint probability distribution over all agents' actions [30, 31]. Centralized methods based on reinforcement learning [32–34], inverse reinforcement learning [35], or game theory [36–38] and inverse game-theoretic optimal control [39] are also common. De-centralized methods include reciprocal velocity obstacles [20] and other reciprocal control obstacles [40–42], as well as de-centralized approaches based on reinforcement learning [43, 44].

The method of [30, 31], termed as Interacting Gaussian Processes (IGP), assumes a joint probability distribution over multiple interacting agents' trajectories, in which the highest probability is assigned to a combination of trajectories that are individually smooth and not colliding with each other. At runtime, the robot's and surrounding pedestrians' motion is predicted based on the conditional distribution over their trajectories during the near future, given their trajectory data in the past. The prediction for the robot is considered as its plan, which is executed subsequently. By planning under the assumption of mutual adaptation, solutions in which every agent progresses can be found even in dense conditions, where the robot could not determine a non-colliding way forward when assuming that other agents will maintain their current velocities. "Unfreezing" the robot in such situations has been stated as the motivation behind IGP [30].

Approaches based on ORCA [20] encode multiple agents' mutual adaptation by pairwise constraints on agents' velocities, such that admissible velocities can be maintained over a time horizon without leading to any collision. These constraints are constructed by approximating the set of non-colliding relative velocities by a halfplane and defining a pair of constraints on both agents' absolute velocities, splitting the responsibility for avoiding collisions between them.

Various formulations [33–35] treat multi-agent navigation as the problem to optimize a reward function in the space of joint multi-agent trajectories, where the objective function incorporates smoothness, social norms, collision avoidance, and agents' desired velocities. The objective may be hand-crafted when performing Reinforcement Learning (RL) [33, 34] or inferred from exemplary data via Inverse Reinforcement Learning (IRL) [35].

For systems consisting of a robot and multiple pedestrians, one may use existing models from the field of crowd dynamics to partially describe interactions in such systems (assuming that the robot's presence does not invalidate these descriptions), in combination with a separate and distinct specification of the robot's behavior. Alternatively, one may utilize a homogeneous description, which does not distinguish between pedestrians and the robot. It is noteworthy that such homogeneous approaches cannot be clearly assigned to one of the fields of crowd modeling/simulation or robotic motion planning, since both fields share a set of techniques and frameworks to describe navigation tasks for multi-agent systems.

1.1.4 Human Walking Behavior and Crowd Dynamics

The study of pedestrian behavior has been an active research area since several decades, due to its importance for urban planning and the design of efficient and safe transportation infrastructure and pedestrian facilities [45–47]. This section briefly reviews the field's developments, focusing on those considered the most relevant for the present thesis.

Empirical Studies

On the one hand, individual characteristics of pedestrians have been measured statistically. For example, an individual's energy consumption as a function of walking speed, and the distribution of individually preferred walking speeds have been quantified by empirical research [48]. Furthermore, individual acceleration behavior during transition from standstill to walking has been investigated in laboratory conditions [49].

On the other hand, studies focused on interactions which occur between multiple pedestrians who navigate simultaneously through the same area. An important source of interactions can be seen in pedestrians' need to avoid collisions and to maintain socially acceptable distances between each other. For such purpose, pedestrians continuously monitor an elliptic area extending in front of them to be able to react whenever another pedestrian is approaching them with a velocity that would lead to a collision in the near future. This process of continuous observation has been described qualitatively in [50], where it was termed as *scanning*. Once two pedestrians register in such way that a collision between them is imminent, they respond by quickly adapting their velocities by the smallest extent which will let them pass by each other at an acceptable distance, as experiments under laboratory conditions have shown [51]. There is a strong social norm demanding equal contributions in this mutual avoidance, as could be evidenced by experiments in pedestrian zones [52]. There, experimenters approached oncoming subjects without adapting to them, which often inhibited subjects to adapt to them in turn, until the very last moment, leading to slight physical contact. Accordingly, this phenomenon has been termed as *brushing* [52].

Also, interactions between pedestrians who move as a group have been investigated. It has been found that such groups of three or four pedestrians often assume a V- or U-like formation whose ends point in the direction of motion (i.e. contrary to e.g. birds' formations), since this facilitates group members to communicate and maintain eye contact with each other [53, 54].

Among the most important quantitative empirical results, fundamental diagrams of speed-density relationships are noteworthy. Such fundamental diagrams capture the extent to which pedestrians are slowed down by others as a function of their density, e.g. for pedestrians walking in the same direction on straight pathways [48] or through bottlenecks [55].

Crowd Motion Models

Based on such empirical findings as mentioned above, mathematical models have been developed to simulate and study collective behaviors of many pedestrians forming a crowd. A crowd can be defined as a large number of individuals who are within the same space at the same time, and whose movements last for a prolonged period of time and depend mostly on local interactions [56]. For sufficiently dense crowds, local interactions may result from physical contact between individuals [46]. In less dense crowds, pedestrians walking close to each other interact by coordinating their movements in order to avoid collisions, while aiming at smooth and efficient motions [47]. Thus, human crowds may be seen as self-organized systems, whose local interactions may give rise to complex behaviors at a global scale [46]. Such global behaviors, often referred to as emergent patterns, include, e.g., separation of streams heading in opposite directions in a corridor, a phenomenon which is termed as *dynamic lane formation* [46, 47, 56].

Crowd motion models allow to simulate pedestrian flows under general boundary conditions, as imposed, e.g., by geometric layouts of pathways. Thus, crowd motion models are particularly interesting from the perspectives of urban planning [46], but also for computer animation and computer games [56]. A first major category of crowd motion models is given by continuum models [56], which describe flow conditions by a velocity field defined in space and time (as in continuum mechanics). A second major category is given by agent-based models [20, 46, 47, 57], which explicitly describe individual pedestrians' trajectories and interactions between them and with static obstacles. The following discussion is limited to agent-based models, since continuum models can only describe macroscopic properties, whereas the thesis focuses on pedestrians' local interactions with a mobile robot.

Among agent-based models, a major category is given by velocity-based models [20, 56, 58, 59]. Typically, they determine agents' velocities such that anticipated collisions in the near future are avoided. Arguably, this principle is in agreement with empirical findings on pedestrians' pairwise avoidance behavior [51] mentioned in the foregoing section. Moreover, the approach of ORCA [20] constrains two given agents' velocities by the same amount with respect to their current velocities, thereby encapsulating the aforementioned social norm [52] that both involved agents should contribute equally to avoiding an anticipated collision.

Force-based models [46, 57, 60] constitute another major category of agent-based models. They are often inspired by statistical mechanics and model interactions by repulsive forces which are derived from a potential function. Social Force models [46, 53, 60] may describe both physical and non-physical forces, which include attractive potentials and anisotropic interactions (giving more weight to pedestrians ahead, to account for a pedestrian's scanning direction). Furthermore, anticipation of collisions has been included via velocity-dependent repulsive forces [60]. In contrast to such hand-crafted forces, the Universal Power Law [57] uses a potential whose functional shape has been inferred from real-

world crowd motion data representing various conditions, yielding a power law dependent on the time to collision. As a consequence, this model also encapsulates anticipation of and early adaptation to collisions in the future.

Outside these categories, a model based on optimal control and differential games [47] has been proposed. There, each agent is considered to maximize its subjective anticipated utility, which accounts for collisions between anticipated trajectories by a distance-dependent penalty. It is noteworthy that such a game-theoretic perspective can explicitly account for non-cooperative behavior, such as one pedestrian not being aware of or ignoring another pedestrian.

Crowd Prediction and Trajectory Forecasting

Recently, the task of predicting individual trajectories of pedestrians in crowded environments has gained attention of researchers. Both the task to predict people's motions in response to an environments layout [61], and the task to predict their interactions with others [62,63] have been tackled by deep learning approaches. Typically, the problem to predict interactions between persons is formulated as a regression problem, i.e. to find pedestrians' locations in the near future based on their locations in the near past.

1.1.5 Evaluations of Crowd–robot Interaction

Commonly, simulation-based and real-world experiments yield complementary insights about a robot's performance. For robots navigating in human crowds, simulations typically involve crowd motion models, such as those mentioned in the previous section.

Simulation-based Evaluations

The method of ORCA [20] is among the most popular crowd simulation models for evaluating methods of robotic navigation [2, 33, 64, 65]. Social Force models [46] are similarly popular for such evaluations [2,65,66]. Recent works [2,65] pointed out the need for a standardized benchmark to compare different approaches to robotic navigation in crowds. Thus, a simulation-based benchmark was proposed [65], featuring various crowd motion models in a unifying framework [58].

Most quantitative evaluations report metrics for the robot's efficiency, smoothness, and safety. Efficiency can be quantified by the time it takes to travel from an initial to a goal position [65], possibly normalized by the time a straight motion at the preferred speed would take [64]. Smoothness can be quantified by the trajectory's average jerk [65]. Safety can be quantified in terms of minimum distance to other agents and the number of collisions [64,65]. Sometimes, these metrics are evaluated not only for the robot, but also for pedestrians [65].

It has been remarked [2] that evaluations based on different crowd simulation methods may yield rather different results in terms of such metrics, e.g.,

for the robot's efficiency. A remedy can be seen in performing multiple evaluations of the same robotic navigation method under different crowd motion models [2, 65]. However, such an approach raises additional questions, e.g., how to interpret such varying performance across crowd models, or whether a particular selection of crowd models encompasses sufficient behavioral diversity to allow robust conclusions. Thus, the models which underly such simulation-based evaluations define the scope of the conclusions that can be drawn from their results. If it is not clear whether a model's assumptions can be extended to crowd–robot interaction, real-world experiments are certainly necessary.

Evaluations with Real Robots and Pedestrians

Real-world experiments can be performed in laboratory conditions with recruited participants or in open environments with random passersby, where the latter option typically offers greater variety and better representation of realistic human behavior. In some works [34], robots are tested for an extended period of time, i.e. they are operated in a pedestrian zone for hours or even days, and any incidents are noted and possibly analyzed, which ideally can demonstrate that the robot is capable of operating continuously and safely under various real-life conditions.

1.1.6 Challenges Addressed in the Thesis

As outlined in the aforegoing sections, enabling a mobile robot to navigate safely and efficiently in human crowds constitutes an inter-disciplinary problem. There, the perspectives of crowd modeling and robotic motion planning can be seen as equally important and even as coinciding to a certain extent. However, inevitable differences between pedestrians and mobile robots create new and challenging problems, which have not been addressed sufficiently by either field on its own.

Most aforementioned approaches assume that agents form a homogeneous crowd. Accordingly, they often assume symmetric relations between agents, e.g. when constructing the conditional probability density (for IGP), velocity constraints (for ORCA), or the reward function (for RL/IRL). However, the assumption of a homogeneous crowd constitutes a strong simplification whenever a robot is present. Often, a robot and pedestrians will differ regarding their shapes and dynamic and kinematic constraints.

Robots with Non-holonomic Kinematics and Non-circular Shape

Since many mobile robots are based on conventional wheels, which impose non-holonomic constraints on their kinematics, they do not possess the ability to move laterally. This fact limits their maneuverability and may hinder their efficient motion, particularly if they are surrounded by densely spaced obstacles. The issue is further aggravated if the robot's shape is not circular but elongated (from a top view), since rotations may cause collisions with obstacles around the

robot's rear or front parts. Many person carrier robots or small vehicles indeed exhibit an elongated shape in addition to non-holonomic constraints, and thus, their agility is limited when driving in a crowded environment. For such cases, suitable methods of reactive control or motion planning must not only take into account non-holonomic kinematics, but also work with a tight approximation of the robot's shape, in order to be able to exploit small amounts of free space.

Early works [24, 25] addressed non-holonomic kinematics but were limited to static environments. Several methods [67, 68] have extended ORCA [20] to account for non-holonomic kinematics. However, they have only considered circular shapes, and either rely on a computationally heavy search through a finite number of trajectories (obtained by integration of the robot's non-linear kinematic equations) or use an increased bounding circle to mask non-holonomic motions [68], which can be prohibitive for entering narrow passages. Furthermore, the aforementioned works have assumed circular shapes. Elongated shapes are addressed by similar works [69–71], but only for holonomic agents. Methods based on model predictive control [26] are general enough to address both non-holonomic kinematics and non-circular shapes, but they come with high computational costs. Thus, the aforementioned works do not fully address the need for a computationally lightweight method to plan short term motions in dense dynamic clutter for a non-holonomic robot whose shape is non-circular.

Control Obstacles' Geometry

There is a variety of methods [40, 42, 42, 69–71] extending the basic idea behind VO. They all define a set of inadmissible inputs or commands by assuming a particular motion model and looking for inputs to this model which result in a collision before a time horizon. This inadmissible set can be termed as Control Obstacle (CO) [42]. Notably, general linear dynamic motion models [72] and heterogeneous multi-agent systems [42] have been addressed in the framework of CO. These works have enlarged the class of behaviors and interactions that can be represented by reciprocal CO, which makes them interesting for modeling pedestrian–robot interactions.

However, prior work has viewed CO mostly from a performance-oriented perspective, without describing CO in detail from a theoretical viewpoint. Only little has been said about the geometric properties of the sets which are defined by CO. For example, important topological properties, such as simple connectivity, have not been established mathematically, but implicitly assumed, to the best of the author's knowledge. While such properties are trivial to deduce for VO, it is less obvious how to prove them for other CO.

While deriving linear constraints for an agent's velocity from VO is a matter of a few geometric constructions, the aforementioned extensions to more general types of CO rely on sampling the boundary of the respective CO to construct linear constraints. Thus, deeper knowledge of the geometric properties of CO could aid not only their theoretical understanding, but also their computational and algorithmic exploitation.

Motion Planning with Empirical Interaction Models

While IGP, RVO, and centralized reinforcement learning approaches perform multi-agent motion planning in a self-consistent and principled fashion, it remains largely unclear to which extent the different underlying models of cooperation may be considered as descriptions of actual pedestrians' cooperative behavior. If one assumes a scenario where a robot uses one objective function to navigate among pedestrians whose actions follow a different objective function, then the robot and pedestrians would predict each others' motions incorrectly and incur higher costs (measured under either of both alternative objectives) than predicted. Even if pedestrians' real behavior cannot be easily expressed by an objective function, it is clear that some objective functions will describe their behavior better than others, e.g. quantifying better their physical effort or their notion of risk. Accordingly, more representative objective functions will yield better predictions and more efficient trajectories by consequence. Thus, an important challenge can be seen in developing models that adequately describe mobile robots' and pedestrians' interactions which are driven by the objective to navigate simultaneously in the same space.

1.2 Contributions

The present thesis complements prior work on robots' navigation in crowds by three main contributions.

The first contribution (Chapter 3) addresses the need for a reactive control system suitable for real-world mobile robots operating in dense crowds. Such a control system is necessary for safety reasons, and therefore, it can be treated as separate from the topics of cooperation models and multi-agent motion planning, which were discussed in the Section 1.1.3. Such a reactive controller's role is to maintain safety even when pedestrians do not act as predicted by a cooperation model.

Thus, the Chapter 3 considers non-holonomic robots with non-circular shapes and proposes a novel reactive control scheme which extends the Velocity Obstacle [20] to such robots. It is implemented as a low-level safety system which corrects input commands whenever necessary to avoid collisions. Accordingly, an evaluation is carried out which quantifies the robot's performance at tracking its input trajectory while avoiding collisions. Further, the robot's impact on the crowd's speed is evaluated. These simulations are complemented by robotic experiments to demonstrate the method's feasibility.

The second contribution (Chapter 4) sheds light on control obstacles' geometric analysis by studying the Acceleration Obstacle (AO) as an exemplary control obstacle. The Chapter 4 develops an approach based on the AO to avoid collisions in a multi-agent environment. The AO's geometric properties are analyzed, and an algorithm is proposed to construct linear approximations thereof. A novel theoretical perspective is given to describe the shape of the AO and to reveal its dependence on the involved agents' initial conditions. In particu-

lar, its boundaries' local convexity or concavity are analyzed and their relation to the two considered agents' relative position and velocity is established and proven mathematically. Furthermore, we show that the AO can be obtained as a limit case of the Acceleration Velocity Obstacle [40]. An evaluation of simulated crowd–robot interactions in a corridor is presented, where the robot's interaction with emergent lanes is studied. Furthermore, simulated phenomena that are related to differences in the crowd's and the robot's controllers are discussed. The simulation-based evaluation of AO in Chapter 4 is complemented by an additional simulation study in Chapter 6, which also investigates other approximation schemes for AO.

The third contribution (Chapter 5) aims towards learning empirically based models of cooperative navigation. The Chapter 5 models cooperative navigation as an optimal control problem and applies inverse reinforcement learning (IRL) to recover a cost function from examples of jointly navigating pedestrians. This data-driven approach aims at obtaining a cost function which is representative of pedestrians' real behavior. In this spirit, a novel reward feature is proposed for IRL of multi-agent navigation, namely our approximation of the interaction energy defined in related work [57]. Results show that a meaningful reward can be learned with the proposed model, whose predictions avoid most collisions successfully and outperform a similar approach reported in the literature in terms of accuracy. Additionally, a qualitative evaluation of our IRL-based approach is performed, where a robot navigates in a real crowd.

The aforementioned three main contributions are complemented by a comparison of their methods (Chapter 6). In this comparison, the assumption of constant acceleration underlying AO is shown to be less suitable for longer time horizons, when navigating in a crowded environment, compared to the assumption of constant velocity, which underlies VO. Further, the main methods from the three main contributions are compared qualitatively, in order to highlight their differences and to compare their assumptions. There, it is shown that VO are more effective than AO at resolving conflicting motions of multiple agents whose desired ideal motions intersect in a point. In fact, it turns out that, in this case, the linearizing scheme proposed for AO in the Chapter 4 leads to stalling movement at the point where agents meet, which is not resolved timely. The cooperative optimal controller of the Chapter 5 outperforms VO and AO, since the underlying assumption of coordinated control strongly simplifies the task compared to the other methods.

1.2.1 Choices and Assumptions Made in the Thesis

The most important choices and assumptions that characterize the present thesis' overall approach are listed below.

- The control methods proposed in this thesis rely on the assumption that the robot is able to accurately perceive its own position, orientation, linear and angular velocity. Furthermore, it is assumed that the robot is capable of perceiving the same state variables for obstacles.

- In the present thesis, the focus is on reactive control and short term motion planning. This is due to the fact that human crowds constitute chaotic systems, such that planning a motion over a longer horizon is not expected to yield strong additional benefits.

- The present thesis assumes that a path around static obstacles is available, i.e. path planning is considered outside of the thesis' scope, since pedestrian zones do not pose any further complications for existing methods of path planning in static environments.

- The robot is modeled either as a non-holonomic vehicle with capsule shape (Chapter 3), which corresponds to the employed experimental platform Qolo, or as a holonomic agent with circle shape (Chapters 4, 5, 6). The latter, simpler model constitutes an abstraction, which allows later chapters to focus on other aspects of navigation in crowds.

- The present thesis focuses on agent-based crowd models, since it is straightforward to combine them with any model of a mobile robot which interacts with the crowd, whereas for continuum models, it is less clear how to incorporate such a local anomaly, as given by a mobile robot. Since agent-based models represent interactions between agents in an explicit fashion, they can be easily extended to include pedestrian–robot interactions.

- The present thesis focuses on interactions that serve only collision avoidance and maintaining acceptable distances.

- To include a robot in an existing crowd model, the present thesis assumes that pedestrians react to the robot in the same way as to other pedestrians. This assumption mainly concerns collision avoidance and acceptable distances, since the aforementioned crowd models represent only interactions for these purposes. Considering that they model interactions to depend only on relative positions and velocities, the assumption is justified if the following two conditions are met. Firstly, pedestrians need to be able to perceive the robot's position, velocity and physical dimensions similarly well as with regard to other pedestrians. Secondly, pedestrians need to accept similar distances to the robot as to other pedestrians. While the first condition seems plausible in general, the second condition may depend on the crowd's familiarity with the robot as well as the robot's perceived safety.

- The crowd models used in the present thesis are given by ORCA [20] (Chapter 3), the Social Force Model (SFM) [46] (Chapter 4), and the Universal Power Law (UPL) [57] (Chapter 6). For the Chapter 3, the macroscopic behavior is defined by a crowd's original reference trajectories. This choice is motivated by the focus on short term reactive control with given high-level input. Thus, the choice of local avoidance is considered less important, but RVO is chosen mainly because it does not interfere with the reference motion unless necessary for avoiding collisions. Furthermore,

it suits the velocity-based simulation framework. The SFM is chosen for its ability to generate dynamic lanes, which allows the Chapter 4 to study possible interactions between the robot and such lanes. The UPL is chosen in the Chapter 6 to complement the similar experiments of the Chapter 4 by a different crowd model, which also yields dynamic lanes.

1.2.2 Publications of Individual Thesis Chapters

The work reported in this thesis has been previously published or submitted for publication in the following individual articles.

- D. J. Gonon, D. Paez-Granados and A. Billard, "Reactive Navigation in Crowds for Non-Holonomic Robots With Convex Bounding Shape," in *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4728-4735, July 2021, doi: 10.1109/LRA.2021.3068660.

- D. J. Gonon, D. Paez-Granados and A. Billard, "Robots' Motion Planning in Human Crowds by Acceleration Obstacles," in *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11236-11243, Oct. 2022, doi: 10.1109/LRA.2022.3199818.

- D. J. Gonon and A. Billard, "Cooperative Navigation in Crowds by Inverse Reinforcement Learning," manuscript submitted to the *IEEE Robotics and Automation Letters*.

1.2.3 Thesis Structure

The next Chapter 2 introduces additional concepts used in later chapters. Then, the three main contributions are presented in the subsequent Chapters 3, 4, and 5. These are followed by a comparison in Chapter 6, where the methods from previous chapters are compared with each other. The final Chapter 7 gives the thesis' conclusions.

Chapter 2

Background

This chapter briefly introduces important modeling concepts and methods that are referred to in the subsequent chapters.

2.1 Kinematic and Geometric Modeling of Mobile Robots

In this thesis, we consider navigation on flat ground, which is very common in urban areas. Thus, space is modeled as two-dimensional. A mobile robot is described as a an area of fixed shape, which can translate and rotate in space. Accordingly, the robot's configuration is defined by its position $\mathbf{x} \in \mathbb{R}^2$ and its orientation $\varphi \in \mathbb{R}$. The position's and orientation's first derivatives with respect to time define the robot's linear velocity $\mathbf{v} := \dot{\mathbf{x}}$ and angular velocity $\omega := \dot{\varphi}$, respectively. The second derivatives define the linear acceleration $\mathbf{a} := \ddot{\mathbf{x}}$ and angular acceleration $\alpha := \ddot{\varphi}$, respectively.

If a robot's linear velocity's components and angular velocity are independent of each other, the robot is said to have holonomic kinematics. This is the case e.g. for robots with omni-wheels. In contrast, if the velocity variables are not independent, the robot's kinematics are said to be non-holonomic. Usually, their dependence is described by an equality involving the robot's orientation and velocity variables, which is termed as a non-holonomic constraint and which assumes that there is no slip occuring between the robot's wheels and the ground. Non-holonomic constraints typically apply to mobile robots which are based on conventional wheels. Many such robots' kinematics can be modeled as differential drive, unicycle, or bicycle models [40, 73]. As depicted in the Fig. 2.1, these models have in common that the robot's velocity state can be fully described by its forward velocity v, defined as one particular point's velocity's component in the direction aligned with its orientation, and by its angular velocity ω. The subsequent Chapter 3 will assume that this point is chosen on the wheel axle of the differential drive robot considered there.

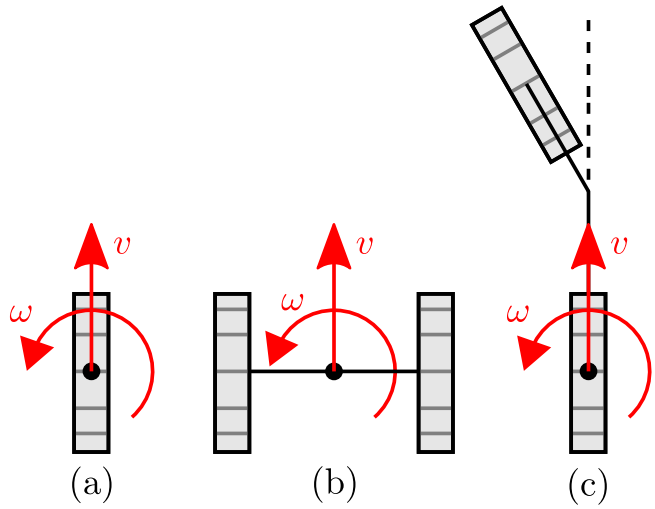

Figure 2.1: The unicycle (a), the differential drive (b), and the bicycle (c) are common non-holonomic kinematic models to describe mobile robots. For each model, the robot's velocity state can be characterized by the forward velocity v and the angular velocity ω, where v is defined with respect to one particular point (e.g. chosen as depicted), whose velocity's component in the direction of the robot's orientation defines v.

Any mobile robot's velocity and acceleration are subject to physical limitations, since actuators can neither move at infinite speeds nor exert infinite forces or torques. Furthermore, safety and stability considerations will often impose restrictions (e.g. to avoid tipping over). Thus, a model of a mobile robot needs to include such bounds in order to represent only realistic and safe motions. A suitable characterization of these bounds depends on the type of robot (particularly its kinematics). Generally speaking, a bound on one particular velocity or acceleration variable can depend on others' values at the given instant.

A mobile robot's shape is commonly modeled by an area which contains the projection of its three-dimensional body on the ground plane, sometimes termed as its footprint or baseprint, as illustrated by the Fig. 3.2. Circles, ellipses, and polygons constitute common models. Another common model for elongated bodies is given by a capsule, which is defined as the area swept by a circle as its center moves over a line segment (cf. Fig. 3.2). Generally, convex shapes are more common, since they are often sufficiently accurate and easier to handle from an algorithmic viewpoint. For example, there is a unique solution to the problem of choosing two points from two respective disjoint convex areas such that they have the smallest possible distance to each other.

2.2 Velocity Obstacle

The approach proposed in the Chapter 3 relies on a concept termed as Velocity Obstacle (VO) to avoid collisions between the robot and moving obstacles. The VO constitutes a geometric construction which allows to algorithmically find velocities that do not lead to collisions, assuming that agents maintain their velocities in the (near) future. The term VO has been introduced in [74], but earlier works [75, 76] similarly used equivalent concepts. For defining VO, we consider two agents A and B of circular shape with respective radii r_A and r_B and the relative position $\mathbf{p} := \mathbf{p}_A - \mathbf{p}_B$, as depicted in the Fig. 2.2-a. The agents' relative VO for a given time horizon τ is defined as [20]

$$VO^{\tau}_{A|B} := \{\mathbf{v} \mid \exists t \in (0, \tau] : |\mathbf{p} + t\mathbf{v}| \leq R\} \tag{2.1}$$

where $R := r_A + r_B$ denotes the combined radius. Thus, the set $VO^{\tau}_{A|B}$ contains all relative velocities $\mathbf{v} := \mathbf{v}_A - \mathbf{v}_B$ that would lead to a collision at some point in time no later than τ from now.

Any constant relative velocity from the complement of $VO^{\tau}_{A|B}$ guarantees collision avoidance in the near future (defined by τ). Accordingly, if the velocity $\mathbf{v}_B$ of B was known, then A would need to choose a velocity $\mathbf{v}_A$ from the complement of $VO^{\tau}_{A|B} \oplus \{\mathbf{v}_B\}$, where $\oplus$ denotes Minkowski addition. Since $VO^{\tau}_{A|B} \oplus \{\mathbf{v}_B\}$ is simply the relative VO shifted by B's velocity, we may term this shifted VO as the absolute velocity obstacle which B induces on A's velocity.

It may be practical to approximate a VO by a conservative halfplane, which contains the original VO. The complementary set of admissible velocities is then convex, as it is defined by a linear constraint. Furthermore, after we construct such a constraint for the absolute VO induced by B on A, we may repeat the same procedure for a different agent C's VO induced on A, and so on for more agents. This results in a convex set of admissible velocities for A that satisfy the constraints due to all surrounding agents.

The VO is constructed as shown in the Fig. 2.2-a [19,20,74,76], first assuming an infinite time horizon. In [20, 76], it has been proposed to truncate VO by assuming a finite time horizon (cf. Fig. 2.2-a). The cone's spherical truncation and linearization as a halfplane according to [20] is shown in the Fig. 2.2-b.

2.3 Acceleration Velocity Obstacle

The approach proposed in the Chapter 4 closely relates to an extension of VO to second order dynamics, termed as the Acceleration Velocity Obstacle (AVO). In prior work [40], AVO has been proposed as a concept similar to VO but capable of planning smooth motions, i.e. generating continuous velocity profiles while guaranteeing collision-free motion. Namely, for two agents A and B with the relative position $\mathbf{p}$ and the relative velocity $\mathbf{v}$, the AVO is defined as the set of all relative reference velocities $\bar{\mathbf{v}}$ that will lead to a collision before a given time horizon when being tracked by the proportional control rule $\dot{\mathbf{v}} = (\bar{\mathbf{v}} - \mathbf{v})/\delta$, where $1/\delta > 0$ defines a gain. Thus, the AVO defines a set of

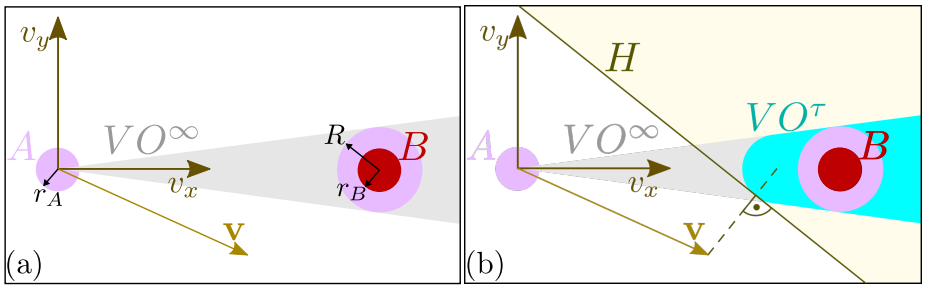

Figure 2.2: (a) The Velocity Obstacle with infinite time horizon is shown. (b) For a finite time horizon τ, the Velocity Obstacle is linearized by a halfplane H that maximizes the admissible space around the current velocity $\mathbf{v}$.

inadmissible reference velocities, which is derived based on the assumption of a continuous transition from the current velocity towards the reference velocity. In contrast to the VO, the AVO's boundary does not contain any straight parts, which makes it more difficult to handle from a computational viewpoint. The Chapter 4 investigates a concept which is closely related to AVO, termed as the Acceleration Obstacle (AO). It turns out that the AO can be obtained from AVO by taking the limit $\delta \to \infty$.

2.4 Inverse Reinforcement Learning

The Chapter 5 turns towards learning navigation from observations of pedestrian behavior. The field of Inverse Reinforcement Learning (IRL) is concerned with inferring an objective function from observed behavior, such that the behavior is close to optimal under the found objective function. These observations are also called demonstrations, and it is assumed that they have been performed by an expert. Thus, IRL addresses the inverse problem of Reinforcement Learning (RL), since RL looks to generate optimal behavior under a given objective function. However, IRL needs to impose some additional restrictions in order to have meaningful solutions only. Otherwise, a constant objective function (which is not informative at all) would be a valid solution to any IRL problem.

Different perspectives [77, 78] have been developed to define more strictly what constitutes a meaningful solution to the problem of IRL. Max-margin IRL [77] samples behavioral policies and chooses an objective function that maximizes the gap between the expert's and the sampled policies' performance. However, this approach does not yield a unique objective function. In contrast, Maximum Entropy (MaxEnt) IRL [78] finds the unique policy which has maximum entropy while achieving the same average performance as the expert under any hypothetical objective function. As shown in [78], this policy is found by maximum likelihood estimation of the objective function's parameters. Below, we discuss an extension of MaxEnt IRL to continuous state and action spaces.

We assume that observations are generated by a dynamic system's evolution

over K time steps. Let $\mathbf{x}^{(k)} \in \mathbb{R}^m$ denote the system's state at time step k. We assume that the action $\mathbf{u}^{(k+1)} \in \mathbb{R}^n$ taken there determines the state at the next time step according to

$$\mathbf{x}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)} + \mathbf{B}\mathbf{u}^{(k+1)}, \quad k = 0, \dots K-1 \tag{2.2}$$

where $\mathbf{A} \in \mathbb{R}^{m \times m}$ and $\mathbf{B} \in \mathbb{R}^{m \times n}$ are constant matrices. Then, each observation is given by a state sequence $\mathcal{X} := \left\{\mathbf{x}^{(k)}\right\}_{k=0}^{K}$ and an action sequence $\mathcal{U} := \left\{\mathbf{u}^{(k)}\right\}_{k=1}^{K}$, which together satisfy (2.2).

As objective functions, we consider cost functions of the form

$$J(\mathcal{X}, \mathcal{U}; \mathbf{w}) := \sum_{k=1}^{K} \mathbf{w}^{\mathrm{T}} \mathbf{f}\left(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}\right), \tag{2.3}$$

where the *feature vector* $\mathbf{f}(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^q$ consists of pre-defined functions that describe quantities contributing to costs, and where $\mathbf{w} \in \mathbb{R}^q$ is the vector of the features' unknown weights to be determined by IRL.

As discussed in [79], applying MaxEnt IRL [78] to such continuous state and action spaces would mean to learn the parameters $\mathbf{w}$ by maximizing the likelihood of observed trajectories under the exponential policy

$$p(\mathcal{U}|\mathbf{w}) = \frac{1}{\int_{\mathbb{R}^{Kn}} \exp\{-J(\mathcal{U}; \mathbf{w})\} \mathrm{d}\mathcal{U}} \exp\{-J(\mathcal{U}; \mathbf{w})\} \tag{2.4}$$

where the shorthand $J(\mathcal{U}; \mathbf{w})$ is based on the fact that $\mathcal{U}$ determines $\mathcal{X}$. However, computing the denominator is intractable due to the integral's high dimensionality.

As a remedy, it has been proposed to approximate J in the denominator by a second-order taylor expansion around $\mathcal{U}$, yielding the approximate likelihood [79]

$$p(\mathcal{U}|\mathbf{w}) \approx \exp\left\{\mathbf{g}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{g}/2\right\} (\det\{-\mathbf{H}\})^{1/2} (2\pi)^{-n/2}, \tag{2.5}$$

where $\mathbf{g} := -\partial J / \partial \mathcal{U}$ and $\mathbf{H} := -\partial^2 J / \partial \mathcal{U} \partial \mathcal{U}$ denote the negative cost's gradient and Hessian evaluated at an observed action sequence $\mathcal{U}$. Accordingly, the approximate log likelihood is given by

$$L(\mathcal{U}|\mathbf{w}) = \mathbf{g}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{g}/2 + \log(\det\{-\mathbf{H}\})/2 - \log(2\pi)n/2. \tag{2.6}$$

For maximizing the log likelihood, an efficient algorithm to evaluate its gradient $\partial L / \partial \mathbf{w}$ with respect to the parameters $\mathbf{w}$ was proposed in [79].

An interesting side-effect of this approximation is that it relaxes the assumption of globally optimal demonstrations to local optimality only. The Chapter 5 uses the approach [79] described above to recover objective functions of cooperatively navigating pedestrians.

Chapter 3

Reactive Navigation in Crowds for Non-holonomic Robots with Convex Bounding Shape

Note: this chapter's contents are based on the following publication.
D. J. Gonon, D. Paez-Granados and A. Billard, "Reactive Navigation in Crowds for Non-Holonomic Robots With Convex Bounding Shape," in *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4728-4735, July 2021, doi: 10.1109/LRA.2021.3068660.

For this chapter, a supplementary video is available online[1].

Abstract This chapter describes a novel method for non-holonomic robots of convex shape to avoid imminent collisions with moving obstacles. The method assists navigation by correcting steering from the robot's path planner or driver. Its performance is evaluated using a custom simulator, which replicates real crowd movements from a campus dataset, and corresponding metrics which quantify agents'efficiency, the robot's impact on the crowd, and the number of collisions. Further, the method is implemented and evaluated on the standing wheelchair Qolo. In the experiments performed, it drives in autonomous mode using on-board sensing (LiDAR, RGB-D camera and a system to track pedestrians). It avoids collisions with up to five pedestrians and passes through a door.

[1] `https://ieeexplore.ieee.org/ielx7/7083369/9399748/9385856/supp1-3068660.mp4?arnumber=9385856`

Figure 3.1: The robot Qolo is passing between pedestrians using the proposed reactive controller (from right to left).

3.1 Introduction

This work considers robots that need to navigate within crowds to reach their goal (as in Fig. 3.1), such as e.g. electric wheelchairs and delivery robots (see also Fig. 3.2). This is challenging because individual pedestrians' decisions are uncertain and also, pedestrians expect cooperative behaviour, as they anticipate and leave space for each other's future motion. Thus, robots need to coordinate with pedestrians but also to adapt quickly to surprising behaviour and avoid imminent collisions that can endanger humans. When evaluating such a robot's controller, one needs to take into account its impact on pedestrians to quantify the robot's social performance.

This chapter is about avoiding imminent collisions between pedestrians and a mobile robot that is non-holonomic (e.g. having wheels preventing sideways motion) and non-circular (e.g. being of elongated shape). We propose the method Reactive Driving Support (RDS) for such robots to correct nominal commands (from the driver or high-level planner) as far as necessary for avoiding previously unanticipated yet imminent collisions. RDS employs Velocity Obstacles (VO) [19, 80], whose basic concept we describe in detail in Sec. 3.2.1. RDS constructs VO between each obstacle and the robot's closest subpart and constrains its velocity accordingly to avoid collisions locally. This constitutes a novel way to extend VO to non-holonomic robots of non-circular shape. This chapter formulates RDS for a capsule, which is a generic shape that fits many delivery robots and robotic wheelchairs (see Fig. 3.2). RDS does not require pre-processing which merges overlapping obstacles (in contrast to e.g. [16, 81]) and is computationally lightweight itself even for very many obstacles. Its implementation is publicly available at `https://github.com/epfl-lasa/rds/`.

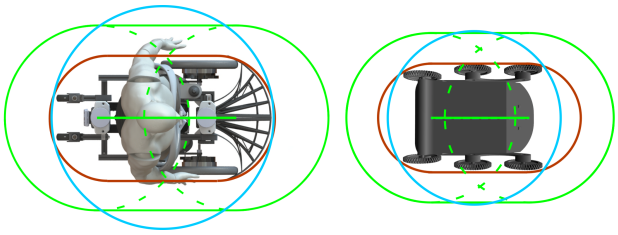

Figure 3.2: Qolo (left) and Starship's delivery robot (right) have footprints which capsules can bound well tightly (red) or more conservatively (green) and still yield a smaller width than the tightest bounding circle (blue).

3.1.1 Related Work

There have been diverse research efforts about robotic navigation in crowds recently. They have mostly modeled robots as circles that can move omni-directionally, thereby idealizing the shape and kinematics to ease investigating the specific properties and challenges which crowds create. Particularly, they have explored navigating cooperatively and according to social norms [33, 34, 66, 82], predicting the surrounding crowd's future motion [29], planning the robot's motion beyond the interactions with its immediate neighbours [83], and conservative collision avoidance under incomplete knowledge about obstacles' position and behaviour [27].

Several approaches exist for circular [24] or even non-convex [25] robots to avoid static obstacles by optimizing over discrete candidate trajectories under dynamic and kinematic constraints. Vector Field Histogram Plus [84] is another optimization-based method, which identifies narrow passages for the robot, but it also assumes static environments. The method in [27] guarantees that the robot is at rest when a collision happens for any behaviour of obstacles, which can be prohibitive in crowds. Another method [85] aims at increasing the feasible command space by conditioning on surrounding agents' most probable maneuvers, but it also assumes a circular robot.

Methods using VO are suitable for short-term planning in dynamic environments such as crowds. The classical VO [19] and its derivatives, e.g. Optimal Reciprocal Collision Avoidance (ORCA) [80], assume circular holonomic agents. Some frameworks enable their application to non-holonomic robots, including [68] and a simple approach that shifts the center (which we discuss in Sec. 3.2.2). However, they artificially increase the robot's radius and thus reduce the capability to plan through narrow passages. An extension of the VO concept to non-linear motion control models [67] can directly treat non-holonomic circular vehicles. However, this relies on forward-integration to yield trajectory candidates from a discretized solution space, which is computationally expensive. Other methods [70, 71] treat non-circular robots with holonomic kinematics, where they separate the step for computing rotations from the step applying VO to find translations. Such an approach is not applicable to robots

with non-holonomic constraints, where translation and rotation are coupled. [86] introduce a velocity-continuous formulation of VO which is applicable to non-holonomic but only circular robots.

Crowd simulations have been common as tools to evaluate and study methods for navigation in crowds [27, 33, 64, 66]. Many such evaluations assign a fixed goal to each agent and evaluate the ability to find an efficient trajectory to the goal by suitable metrics. Very common metrics include agents' path length and time for traveling from their starting to their goal positions for quantifying efficiency, the rate of success (reaching the goal without collision), and the number of collisions or minimum separation of agents for quantifying safety (e.g. in [31, 33, 34, 64, 66, 87]). The evaluation in [88] has the most similar perspective to ours, as it measures the robot's deviations from the high-level planner's (time-independent) reference path by the squared deviation integral.

3.1.2 Contributions

Our method RDS extends VO to non-holonomic non-circular robots for reactive control in crowds. This chapter presents RDS and its evaluation tailored to this context, where we employ a custom simulator and four novel metrics for quantitative evaluation. The simulator combines original time-dependent trajectories of video-tracked pedestrians (from [89]) with local collision avoidance such that agents incorporate high-level planning, social coordination, and local collision avoidance. By using empirical reference trajectories, we aim at making agents' arrangements and motion patterns representative of the original crowd. Corresponding to these time-varying goals, we introduce the robot's and the crowd's average deviation from their reference trajectories as metrics to quantify how efficient a reactive controller's corrections are at avoiding collisions while continuing tracking of the reference motions. Thus, our first two metrics reflect the task's focus on complementing high-level motion plans. Further, we introduce two metrics that directly measure how a robot's presence and its controller impact other agents' velocities. While it is common to consider velocities, measuring one specific agent's impact on others is far less common. Both of these metrics relate to the compatibility between a robot's and pedestrians' different ways of navigation.

Complementing this evaluation by experiments with the standing wheelchair Qolo [90] (in Fig. 3.1, 3.2, 3.3), on which we have implemented our method, we demonstrate the method to be effective at avoiding collisions and feasible in practice. The comparison with another method shows that our approach is more advantageous in crowds due to its ability to lead through narrow gaps between obstacles. The next section (Sec. 3.2) provides the concepts underlying this chapter's techniques. The method's description (Sec. 3.3), its evaluation in simulation (Sec. 3.4) and with the robot Qolo (Sec. 3.5) follow. Finally, Sec. 3.7 concludes this work.

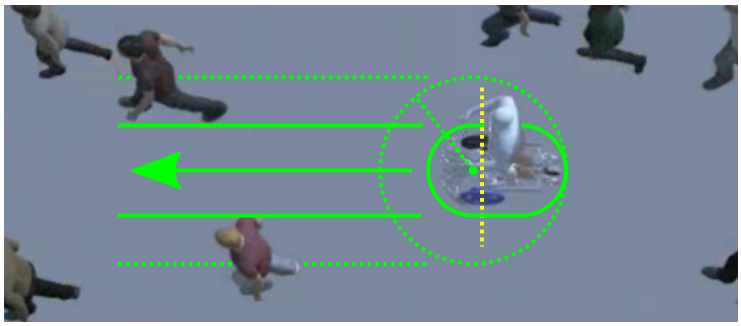

Figure 3.3: The capsule approximation (solid green) fits through gaps which are not accessible with an abstracting circle (dotted green) centered ahead of the wheel axle (yellow).

3.2 Background

This section reviews important technical prerequisites.

3.2.1 Velocity Obstacles for Circular Holonomic Robots

The original VO [19] describes for each circular obstacle the corresponding cone-shaped set of constant velocities for the circular robot that will eventually lead to a collision (as Fig. 3.4 shows). ORCA [80] uses VO for multi-agent navigation as follows (see also Fig. 3.4). It disregards collisions after the time horizon τ and thus spherically truncates the VO cone. Further, ORCA conservatively approximates the relative VO by the halfplane $\bar{H}_{rel}$ whose boundary touches the relative VO boundary at its point closest to the previous relative velocity $\mathbf{v}_{rel}^{-}$. Then, it obtains the reciprocal absolute VO for the first agent A by shifting the relative VO by the second agent B's previous velocity plus its reciprocal contribution $\mathbf{c}_B \in \mathbb{R}^2$, which accounts for the second agent's contribution to collision avoidance, and vice-versa for the second agent. ORCA thus generates a linear constraint for an agent's velocity due to each other agent, where each constraint is characterized by its normal $\mathbf{n}$ and offset b. Finally, it solves a quadratic program to find the velocity closest to the preferred one while adhering to the constraints.

3.2.2 Abstraction from the Robot's Shape and Kinematics

We define for comparison with our method later on the baseline method as using ORCA [80] (assuming no reciprocity, i.e. $\mathbf{c}_B = \mathbf{0}$) for the given non-holonomic non-circular robot by the following abstracting technique (from [91]). We consider a generic robot's kinematic model which is conceptually equivalent to an axle with two wheels that rotate independently around it and roll on a plane without slip. The model views the robot as a rigid body which moves in the plane and is subject to the non-holonomic constraint which requires its

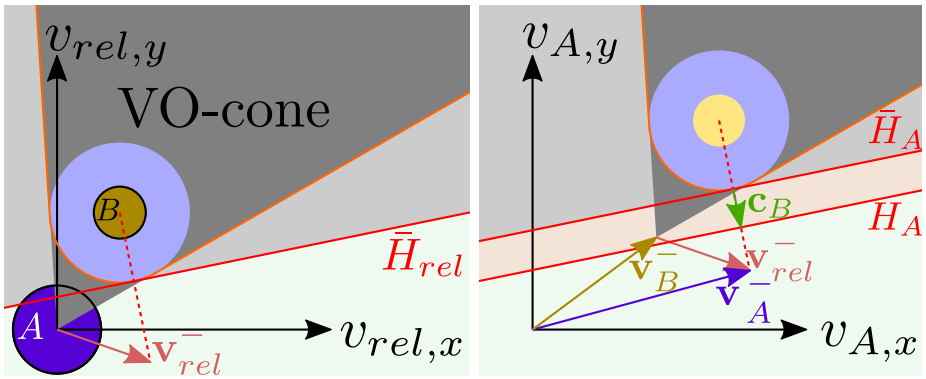

Figure 3.4: In relative velocity space (left), ORCA truncates and linearizes the relative VO (dark gray) between the robot A (blue) and the obstacle B (gold) around the previous relative velocity $\bar{\mathbf{v}}_{rel}$. Shifting the resulting halfplane $\bar{H}_{rel}$ by the obstacle's previous velocity $\bar{\mathbf{v}}_B$ yields the halfplane $\bar{H}_A$ of avoiding velocities in the robot's velocity space (right), if B keeps its velocity. Shifting by B's reciprocal contribution $\mathbf{c}_B$ yields the halfplane H_A of reciprocally avoiding velocities.

instantaneous center of rotation to be on the infinite line that contains the wheel axle.

For abstracting from such non-holonomic kinematics and the robot's possibly non-circular shape, [91] defines the control point as a point on the robot's body that does not lie on the wheel axle. The control point's cartesian velocity has degree of freedom two and thus it serves to receive velocity commands that address a holonomic robot. Further, a circle whose center is the control point and which contains the robot serves to mask its true shape such that one can apply conservatively to this virtual circle a method that avoids collisions for a circular robot. We note that the control point needs to be ahead of the wheel axle in the robot's preferred direction of travel (if it exists) to yield proper signs of angular velocities when avoiding obstacles, i.e. rotating clockwise/counter-clockwise when passing on the right/left, respectively. Consequentially, the virtual circle for this abstraction is particularly conservative when the robot is longer in the rearward than in the forward direction from the wheel axle (as Fig. 3.3 shows for Qolo).

3.3 Method

The method avoids collisions by constructing constraints for the velocity of a particular robot-fixed point, which we refer to as the reference point, according to Fig. 3.5. Each obstacle is taken into account by determining the robot's incircle which is closest to it and constructing the VO that the obstacle induces for the incircle. A linear constraint is derived for the incircle center's velocity

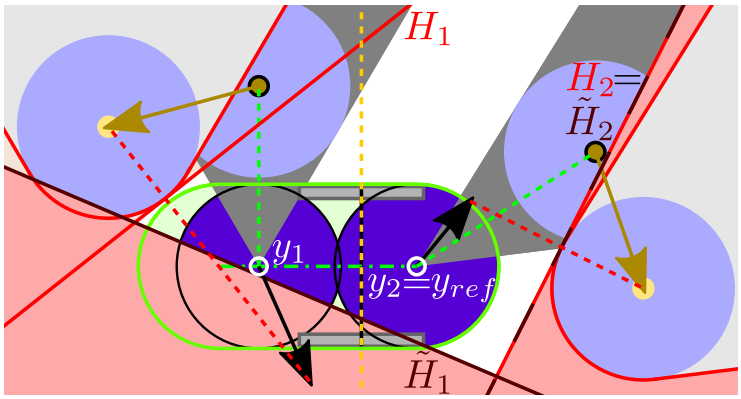

Figure 3.5: The method constructs the velocity obstacles (light grey) for obstacles (gold) and the robot's closest incircle (dark blue), linearizes them as H_i (red) and re-maps them as $\tilde{H}_i$ (dark red) to the reference point's velocity space.

and transformed into the equivalent constraint for the reference point's velocity via the robot's kinematic relation between different points' velocities.

3.3.1 Definitions

We assume the robot to have non-holonomic kinematics according to Sec. 3.2.2. The robot's fixed right-handed coordinate system is defined such that the y-axis separates the wheels symmetrically and points forward and the x-axis coincides with the wheel axle's line. Any cartesian vector components in the method's description refer to this coordinate system. The method's description here assumes the robot's shape as a capsule which is symmetric in the y-axis and corresponds to sweeping a circle of radius r with its center on $x = 0$ from the rear end $y_{rear} < 0$ to the front end $y_{front} > 0$. An incircle is any circle with its center on the line between the endpoints and radius r.

The command vector $\mathbf{u} = [v, \omega]^T$ defines the robot's linear and angular velocity command v and ω in such a way that positive values yield forward translations and counter-clockwise rotations around the origin, respectively. Given any point (x, y), its cartesian velocity $\mathbf{v}$ corresponding to $\mathbf{u}$ can be expressed via the Jacobian $\mathbf{J}(x, y)$ as $\mathbf{v} = \mathbf{J}(x, y)\mathbf{u}$. One can show that

$$\mathbf{J}(x, y) = \begin{bmatrix} 0 & -y \\ 1 & x \end{bmatrix}, \quad \mathbf{J}^{-1}(x, y) = \begin{bmatrix} x/y & 1 \\ -1/y & 0 \end{bmatrix}, \tag{3.1}$$

where $\mathbf{J}^{-1}$ exists for any point (x, y) with $y \neq 0$. Finally, (x_{ref}, y_{ref}) defines the reference point with $y_{ref} \neq 0$. Alternatively to $\mathbf{u}$, its velocity $\mathbf{v}_{ref}$ can describe via the inverse of its Jacobian $\mathbf{J}_{ref}$ the robot's motion, as $\mathbf{u} = \mathbf{J}_{ref}^{-1}\mathbf{v}_{ref}$.

3.3.2 Velocity Constraints for Local Incircles

Assuming circular obstacles (with known radii, positions and velocities), the method constructs for each obstacle O_i the VO which it induces for the respective closest incircle, located at $(0, y_i)$. The VO's construction, truncation by the time horizon τ, and linearization around the incircle center's and the obstacle's previous relative velocity are identical to the approach in [80] (as in Sec. 3.2.1) apart from reciprocity, i.e. the method assumes the obstacle to maintain its velocity (corresponding to $\mathbf{c}_B = \mathbf{0}$ in Fig. 3.4). Thus, each obstacle O_i creates for the velocity $\mathbf{v}_i$ of the point $(0, y_i)$ the linear constraint

$$\mathbf{n}_i^T \mathbf{v}_i \leq b_i, \tag{3.2}$$

which describes the feasible halfplane $H_i(\mathbf{n}_i, b_i)$ with the outwards unit normal $\mathbf{n}_i$ and the offset from the origin b_i.

3.3.3 Optimization Problem

The method aims at optimizing the command vector $\mathbf{u}$ for the robot to execute such that it minimally deviates from the nominal command $\bar{\mathbf{u}} = [\bar{v}, \bar{\omega}]^T$ e.g. by the driver. The optimization problem is formulated in the reference point's velocity space, wherein the nominal command is mapped as $\bar{\mathbf{v}}_{ref} = \mathbf{J}_{ref}\bar{\mathbf{u}}$. Further, the method maps in this space the constraints for the incircles' velocities due to N obstacles by expressing the local velocity in each constraint (3.2) as $\mathbf{v}_i = \mathbf{J}(0, y_i)\mathbf{J}_{ref}^{-1}\mathbf{v}_{ref}$ using (3.1). We incorporate these constraints and the objective in the quadratic program

$$\mathbf{v}_{ref}^* = \arg\min_{\mathbf{v}_{ref}} \ |\mathbf{v}_{ref} - \bar{\mathbf{v}}_{ref}|^2 \tag{3.3}$$

$$\text{s.t. } \mathbf{n}_i^T \mathbf{J}(0, y_i)\mathbf{J}_{ref}^{-1}\mathbf{v}_{ref} \leq b_i \quad \forall i \in 1, ..., N \tag{3.4}$$

$$\mathbf{n}_{v,j}^T \mathbf{v}_{ref} \leq b_{v,j} \quad \forall j \in 1, ..., 4 \tag{3.5}$$

$$\mathbf{n}_{a,k}^T \mathbf{v}_{ref} \leq b_{a,k} \quad \forall k \in 1, ..., 4 \tag{3.6}$$

where the additional constraints in (3.5), (3.6) represent the robot's velocity and acceleration limits, respectively. We assume that they result respectively from four fixed constraints for $\mathbf{u}$ and from the four box constraints around the previous command $[v^-, \omega^-]^T$ which encode $|v - v^-| \leq \hat{a}\Delta t$ and $|\omega - \omega^-| \leq \hat{\alpha}\Delta t$, with $\hat{a}$ and $\hat{\alpha}$ denoting the maximum absolute linear and angular acceleration, respectively, and the control cycle time Δt. These constraints' normals are multiplied by $\mathbf{J}_{ref}^{-T}$ and their offsets are adopted to yield $\mathbf{n}_{v,j}, b_{v,j}, \mathbf{n}_{a,k}, b_{a,k}$ in (3.5), (3.6). However, one can employ any given number and arrangement of constraints instead.

3.3.4 Solution and Command Computation

The method employs an incremental algorithm very similar to [92], [21] to solve the quadratic program (3.3)-(3.6) or determine that its constraints are infeasible.

Importantly, the maximum number of obstacles bounds a priori the number of iterations which the algorithm requires. If the constraints are feasible, the solution defines the velocity command according to $\mathbf{u}^* = \mathbf{J}_{ref}^{-1}\mathbf{v}_{ref}^*$. Otherwise, the pre-defined maximum linear and angular braking decelerations $\hat{a} > 0$ and $\hat{\alpha} > 0$ define the command according to $v^* = h(v^-, \hat{a})$ and $\omega^* = h(\omega^-, \hat{\alpha})$, where $h(u, m) \coloneqq u - \operatorname{sign}(u)\min(|u|, \Delta t\, m)$.

3.3.5 Discussion and Generalizations

Two simplifying approximations underlie the method. First, it reduces the robot's shape to the closest incircle for each obstacle. Second, it approximates the incircles' motions over the planning horizon as straight with individual constant velocities such that they only initially verify a rigid body's velocity distribution. Consequentially, the method requires a high control frequency to prevent collisions, i.e. it must update the robot's velocity command not just as the time horizon elapses but early enough such that both approximations remain valid. However, this is also necessary as obstacles may change their velocities faster than the horizon.

The choice (x_{ref}, y_{ref}) defines the relative costs for deviating from $\bar{\mathbf{u}}$ along different axes in the command space. When viewing $\mathbf{u}$ as the optimization variable in (3.3) by plugging in $\mathbf{v}_{ref} = \mathbf{J}_{ref}\mathbf{u}$, the objective becomes the quadratic form defined by $\mathbf{J}_{ref}^T\mathbf{J}_{ref}$, whose principal axes and eigenvalues can be tuned via (x_{ref}, y_{ref}).

The expression (3.4) maps each halfplane $H_i(\mathbf{n}_i, b_i)$ (constraining the local velocity $\mathbf{v}_i$) to the corresponding constraint for the reference point's velocity $\mathbf{v}_{ref}$. With $\mathbf{M}(x, y) \coloneqq \mathbf{J}_{ref}^{-T}\mathbf{J}(x, y)^T$, the latter constraint can be geometrically interpreted, if $\mathbf{M}(0, y_i)\mathbf{n}_i \neq \mathbf{0}$, as the transformed feasible halfplane $\tilde{H}_i(\tilde{\mathbf{n}}_i, \tilde{b}_i)$ with the unit normal $\tilde{\mathbf{n}}_i = \mathbf{M}(0, y_i)\mathbf{n}_i / ||\mathbf{M}(0, y_i)\mathbf{n}_i||$ and the offset $\tilde{b} = b / ||\mathbf{M}(0, y_i)\mathbf{n}_i||$ (see also Fig. 3.5). In the case $\mathbf{M}(0, y_i)\mathbf{n}_i = \mathbf{0}$, the constraint reads $0 \leq b_i$, which occurs if and only if $y_i = \mathbf{n}_{i,y} = 0$ (e.g. when an obstacle approaches the static robot along the line $y = 0$). There, the local VO constrains $\mathbf{v}_{i,x}$ independently (of $\mathbf{v}_{i,y}$) while kinematically any command must yield $\mathbf{v}_{i,x} = 0$. Thus, $\mathbf{v}_{ref}$ does not enter the constraint, which makes the optimization infeasible and triggers braking if $b_i < 0$. This effect limits the ability to escape close and fast obstacles approaching along $y \approx 0$.

The method's formulation here assumes a capsule-shaped robot. However, any convex shape which a convex polygon and a sweeping circle generate together can replace the capsule. This only requires a routine to compute for a given obstacle the robot's corresponding subpart as the closest instance of the sweeping circle.

3.4 Experiments in Simulation

The simulations in this section compare our method, RDS, to the baseline method (from Sec. 3.2.2). The comparison in Sec. 3.4.3 additionally includes

Algorithm 1 In each simulation update, pedestrians compute their velocities via ORCA, and the robot reacts via RDS.

for $i = 1, ..., N_p$ **do**
  $\mathbf{v}^p_{i,t} = ORCA\left(\bar{\mathbf{v}}^p_{i,t}, \mathbf{x}^p_{i,t}, \{\mathbf{x}_{j,t}, \mathbf{v}_{j,t-\Delta t}\}^{N_p+N_c}_{j=1\neq i}\right)$
end for
$\mathbf{u}^r_t = RDS\left(\bar{\mathbf{u}}^r_t, \mathbf{u}^r_{t-\Delta t}, \mathbf{x}^r_t, \varphi^r_t, \left\{\mathbf{x}^p_{j,t}, \mathbf{v}^p_{j,t}\right\}^{N_p}_{j=1}\right)$
$\mathbf{x}^p_{i,t+\Delta t} = \mathbf{x}^p_{i,t} + \Delta t\, \mathbf{v}^p_{i,t}$
$(\mathbf{x}^r_{t+\Delta t}, \varphi^r_{t+\Delta t}) = (\mathbf{x}^r_t + \Delta t\, \mathbf{v}^r(\mathbf{u}^r_t, \varphi^r_t), \varphi^r_t + \Delta t\, \omega^r(\mathbf{u}^r_t))$

the method "Blank", which is defined as outputting directly its input, such that the robot executes the nominal command, i.e. setting $\mathbf{u}^* = \bar{\mathbf{u}}$. With the method "Blank", only the nominal command and other agents contribute to cooperative navigation, allowing to estimate a compared reactive controller's additional contribution.

We define the control point for all methods and the reference point for RDS to coincide (for simplicity), where the control point (whose velocity the baseline method acts on) additionally represents the robot's position for tracking the reference trajectory given by the experiment. We choose the robot's parameters as x_{ref}=0, y_{rear}=-0.5 m, y_{front}=y_{ref}=0.18 m, r=0.45m (corresponding to Qolo's conservative capsule in Fig. 3.2), and τ=1.5 s, $\hat{a}$=2 m/s^2, $\hat{\alpha}$=3 rad/s^2.

The following Sec. 3.4.1 introduces the simulation framework and its specific performance metrics. A qualitative comparison follows in Sec. 3.4.2. A test series for quantitative comparison follows in Sec. 3.4.3, whose source code is available in the provided repository.

3.4.1 Simulation Framework and Performance Metrics

This section describes our framework for simulating how the robot navigates in environments with pedestrians. We represent them by circular agents (of radius 0.3m) which track individual reference trajectories while avoiding collisions with each other and the robot by applying ORCA (assuming reciprocity, and τ=1.5 s). The robot also tracks a reference trajectory and in turn uses the method RDS (or baseline) to avoid collisions with the pedestrians, reacting to their current position and updated velocities (or not, with the method "Blank"). The simulation's update scheme (for the case with RDS) is given in Algorithm 1, with Δt=0.05 s being its time step and also the cycle time for RDS and other agents' controllers. The following paragraphs introduce the notation and explain the simulation framework.

For each pedestrian i, let $\mathbf{x}^p_{i,t}$, $\mathbf{v}^p_{i,t} \in \mathbb{R}^2$ denote respectively the global position and velocity at time t. Let $\mathbf{x}^r_t \in \mathbb{R}^2$ and φ^r_t denote respectively the robot's position (i.e. where its control point is) and orientation at time t. Accordingly, let $\mathbf{v}^r_t \in \mathbb{R}^2$ and ω^r_t denote respectively the robot's global cartesian velocity (of its control point) and angular velocity at time t. Let $\mathbf{u}^r_t$ denote the robot's com-

mand vector $[v^*, \omega^*]^T$ at time t, whose components are respectively the forward and angular velocity which result from the method for collision avoidance. They prescribe the robot's global velocities, which are thus functions $\mathbf{v}_t^r = \mathbf{v}^r(\mathbf{u}_t^r, \varphi_t^r)$ and $\omega_t^r = \omega^r(\mathbf{u}_t^r)$.

Pedestrians perceive the robot as a collection of N_c virtual circular agents, which are attached to the robot, covering its actual capsule. When simulating with the baseline method for collision avoidance, the collection includes the enlarged bounding circle, otherwise it contains only several tightly fitting circles. The virtual agents adopt the position and velocity of their respective point of attachment on the robot. Let $\mathbf{x}_{j,t}, \mathbf{v}_{j,t} \in \mathbb{R}^2$ denote the position and velocity for a generic circular agent (i.e. a pedestrian or a virtual agent).

The robot's reference trajectory $\bar{\mathbf{x}}_t^r : \mathbb{R} \to \mathbb{R}^2$ defines the robot's reference position at each time t and prescribes the nominal velocity $\bar{\mathbf{v}}_t^r$ for the robot's control point according to

$$\bar{\mathbf{v}}_t^r = \frac{\mathrm{d}\bar{\mathbf{x}}_t^r}{\mathrm{d}t} + k\left(\bar{\mathbf{x}}_t^r - \mathbf{x}_t^r\right). \tag{3.7}$$

Therein, the reference trajectory's derivative forms a feedforward term and the tracking error is added as a feedback term (with the gain $k > 0$). For pedestrians, the reference trajectories $\bar{\mathbf{x}}_{i,t}^p : \mathbb{R} \to \mathbb{R}^2$ prescribe the corresponding nominal velocities $\bar{\mathbf{v}}_{i,t}^p$ in analogy to (3.7). Both terms in (3.7) together achieve a vanishing tracking error over time such that agents converge to their (moving) reference position even after perturbations. A set $\left\{\bar{\mathbf{x}}_t^r, \bar{\mathbf{x}}_{1,t}^p, ..., \bar{\mathbf{x}}_{N_p,t}^p\right\}$ which contains the robot's and N_p pedestrians' reference trajectories over a time window $[t_1, t_2]$ defines a particular *simulation configuration*.

Our metrics rely on the following definitions. The static area of evaluation A defines where relevant interactions are expected (Sec. 3.4.3). Let the indicator function $\mathbb{I}_{\{B\}}$ equal 1 if B is true or 0 otherwise. Further, let $\langle\cdot\rangle = \int_{t_1}^{t_2}(\cdot)\mathrm{d}t/(t_2 - t_1)$ and $\langle\langle\cdot\rangle\rangle = \sum_{i=1}^{N_p}\langle\cdot\rangle/N_p$ denote respectively averaging over the sample's time window and averaging over both the time window and the crowd. We use the following metrics.

- The robot's mean tracking error $E_r = \langle|\bar{\mathbf{x}}_t^r - \mathbf{x}_t^r|\rangle$.
- The pedestrians' mean tracking error $E_p = \langle\langle|\bar{\mathbf{x}}_{i,t}^p - \mathbf{x}_{i,t}^p|w_{i,t}\rangle\rangle$, with $w_{i,t} \propto \mathbb{I}_{\{\bar{\mathbf{x}}_{i,t}^p \in A\}}$ and $\langle\langle w_{i,t}\rangle\rangle = 1$.
- The crowd's velocity reduction due to the robot $V_c = V_{c,0}/V_{c,r}$, where $V_{c,r} = \langle\langle|\mathbf{v}_{i,t}^p|\hat{w}_i\rangle\rangle$ denotes the crowd's weighted average velocity for the case with the robot, and $V_{c,0}$ denotes the analogous quantity for the case without a robot. Herein, the pedestrians' weights $\hat{w}_i \propto \langle\mathbb{I}_{\{\mathbf{x}_{i,t}^p \in A\}}\rangle$ are proportional to their actual time in A and normalized as $\langle\langle\hat{w}_i\rangle\rangle = 1$.
- The neighbours-to-crowd velocity ratio $V_n = V_{n,r}/V_{c,r}$, where $V_{n,r} = \langle\langle|\mathbf{v}_{i,t}^p|\breve{w}_i\rangle\rangle$ and $\breve{w}_i \propto \langle\mathbb{I}_{\{|\mathbf{x}_{i,t}^p - \mathbf{x}_t^r| < D\}}\rangle$ are weights proportional to pedestrians' time D-close to the robot ($D = 3$m) and normalized as $\langle\langle\breve{w}_i\rangle\rangle = 1$.

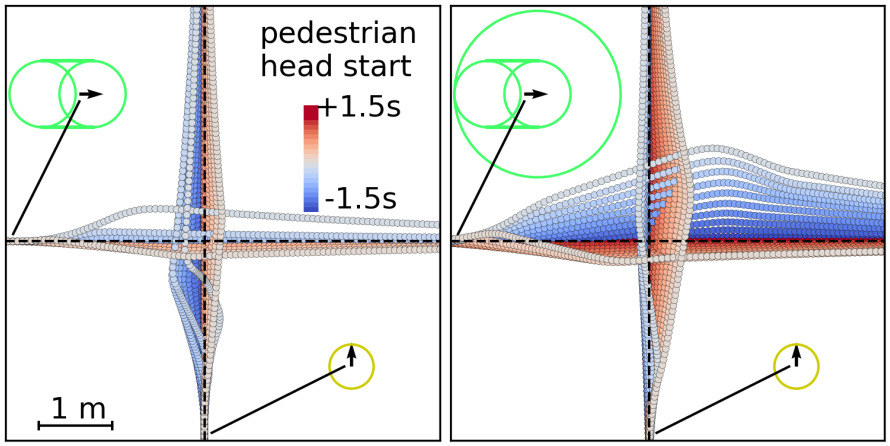

Figure 3.6: The robot (moving to the right) and a pedestrian (moving upwards) cross with variable relative head starts, resulting in the corresponding (color-coded) trajectories and the particular crossing order. The robot uses either RDS (left) or the baseline method (right) for collision avoidance.

- C_r counting collisions with the robot's capsule.

The metric V_c compares the crowd's speed when the robot is not present (leaving its place to a regular agent instead) to the crowd's speed when the robot is present. Thus, evaluating V_c for a given simulation configuration and method (e.g. RDS) requires to execute one simulation without and one with the robot. The metric V_n compares the speed of the robot's neighbours to the entire crowd's speed. If the robot tends to slow down pedestrians, we expect that $V_c > 1$ and $V_n < 1$.

3.4.2 Crossing with Variable Head Start

The following experiment lets the robot and a pedestrian cross while both contribute to collision avoidance according to our simulation framework (Sec. 3.4.1). Their reference trajectories move at the same speed 1.3 m/s and their paths cross orthogonally, however, the pedestrian starts from a variable distance to the crossing point. The pedestrian's head start t^p_{hs} denotes the time difference between the moment when the pedestrian's reference trajectory reaches the crossing point and the moment when the robot's reference trajectory reaches it. Fig. 3.6 shows both agent's trajectories that result respectively for different values of the pedestrian's head start in the range $t^p_{hs} \pm 1.5$ s. The result shows for both methods how the crossing order changes around $t^p_{hs} = 0$. Over the test series, the pedestrian's mean tracking error is similar with both methods to control the robot ($E^{rds}_p = 0.10 \pm 0.12$m, $E^{b.l.}_p = 0.09 \pm 0.06$m), whereas the robot's tracking error is clearly lower with RDS ($E^{rds}_r = 0.20 \pm 0.05$m,

Table 3.1: The metrics' mean and standard deviation (or for C_r, the sum) is shown over the sparse crowd simulations for the three methods. Between the baseline method and RDS, superior mean values are marked in bold and significant differences by asterisks (except for C_r).

Method	E_r [m]	E_p [m]	V_c [-]	V_n [-]	ΣC_r [-]
RDS	**0.8***±0.9	**0.20***±0.09	0.999*±0.007	1.07±0.24	0
Baseline	2.2*±2.0	0.21*±0.09	**0.994***±0.010	**1.08**±0.25	0
"Blank"	0.0±0.0	0.20±0.09	0.996±0.009	1.07±0.23	6

$E_r^{b.l.} = 0.35 \pm 0.14$m).

3.4.3 Navigating in a Sparse Crowd

For quantitative comparison, this experiment series evaluates RDS, the baseline method, and the trivial method "Blank" (that adopts nominal commands) in simulations that are driven by original crowd movements (from a pedestrian intersection on a campus) which are available in the "Crowds-by-Example" dataset [89] as timed waypoint sequences. We use these original trajectories to generate 430 different sample simulation configurations by replacing a different original pedestrian by the robot and simulating the remaining original pedestrians via regular agents. For each agent (including the robot), we define its reference trajectory as the two cubic splines fitting the respective original pedestrian's 2D-waypoints over time. The area of evaluation A is chosen as a tight bounding box of all the waypoints. The time window $[t_1, t_2]$ for a given sample configuration's simulations matches the time window of the waypoints for the robot's original pedestrian. Other agents' trajectories outside their original waypoints' time windows are linear extrapolations (which are mostly outside A). For each sample configuration, we evaluate the metrics from Sec. 3.4.1 for each method (simulating once without a robot). Fig. 3.7 shows for an exemplary sample configuration the robot's and the surrounding crowd's motion during sequential time windows for RDS and the baseline method, respectively. These motion snippets exemplify how the robot can often follow closely its nominal motion with RDS, whereas the baseline method leads it on detours around dense groups.

Table 3.1 reports the metrics' sample averages and standard deviations over the 430 sample configurations for the three methods (or for C_r, the sum over all configurations). For all the metrics except C_r we compare their distributions for RDS against the baseline using a two-sample t-test with a significance level $\alpha = 0.05$. We find $p < \alpha$, i.e. significant differences in the mean values, for all the metrics except V_n.

In comparison to the baseline method, we attribute RDS' significantly lower tracking error for the robot and the pedestrians (i.e. $E_r^{rds} < E_r^{b.l}$, $E_p^{rds} < E_p^{b.l}$) to the tighter shape representation. It allows the robot to maneuver through narrow gaps between pedestrians and requires less deviations from them. On

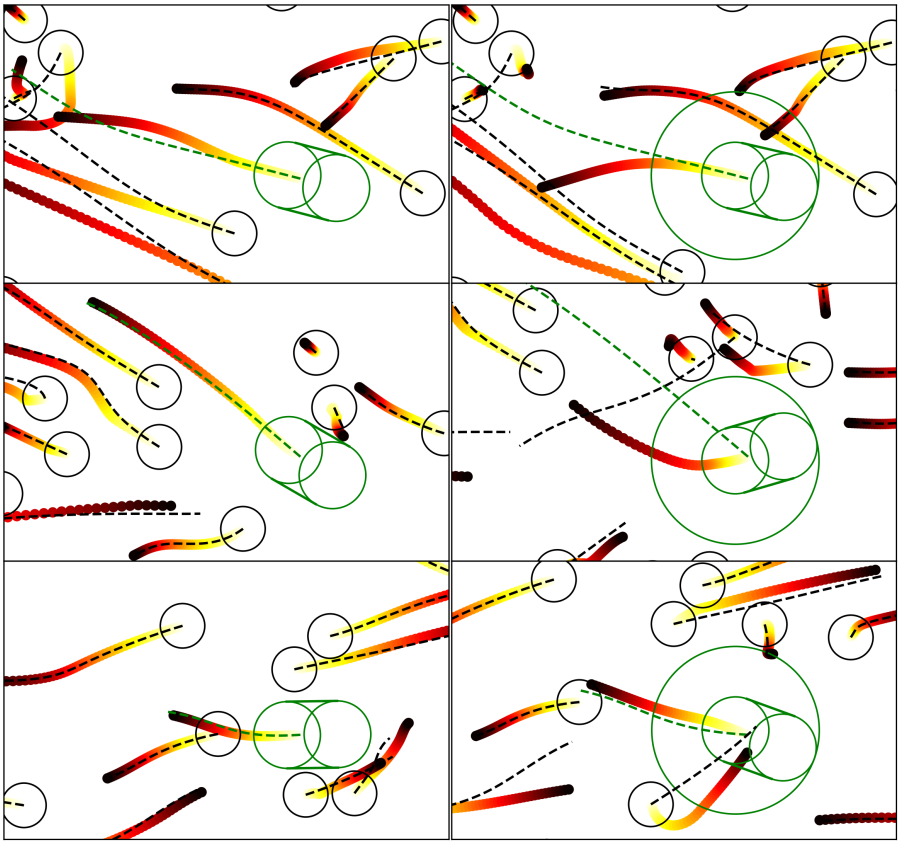

Figure 3.7: The robot uses RDS (left) or the baseline method (right) to traverse the dynamic crowd in this empirically based simulation example. For three sequential time windows, the initial state of the robot (green capsule) and crowd (black circles) and their future motion (yellow to red) and future nominal motion (dashed lines) are shown.

the other hand, the circular shape representation with the baseline method encourages agents to maneuver around the robot with increased velocity (as typical for ORCA), whereas the multi-circle shape representation they perceive for the robot with RDS often traps them between two such circles, thus $V_n^{b.l.} > V_n^{rds}$. While the assumption that other agents perceive the enlarged bounding circle for the robot with the baseline method is not realistic, we observe that it is actually favourable for the baseline method's performance, since otherwise the robot frequently experiences virtual collisions (with its bounding circle), and while trying to resolve them, it is prone to colliding truly, as it does not represent the robot's true capsule shape.

With RDS or the baseline method, collisions do not occur for the robot throughout the simulations. This is due to contributions from both the robot's reactive controller and the pedestrians' cooperative controller. The robot's contribution is still necessary sometimes to avoid collisions, as the method "Blank" (which does not contribute) leads to a few collisions ($\Sigma C_r > 0$).

Comparing E_p, V_c, and V_n between RDS and the method "Blank", we find that RDS does not facilitate other agents to follow their references and generally reduces the crowd's velocity in our simulation framework. However, also with the method "Blank" the robot already receives high-level guidance via the reference trajectory (which avoids other agents original positions) and therefore, this result mainly shows that the smaller holonomic pedestrians can resolve efficiently the slight collisions due to the robotic agent's larger shape even when the robot does not contribute.

In summary, these results show that RDS successfully corrects the robot's motion to account for its capsule shape (whose potential collisions the reference motion does not avoid) while at the same time achieving a low tracking error for the robot and the crowd.

3.5 Experiments with the robot Qolo

The robot Qolo [90] is an electrically powered standing wheelchair (in Fig. 3.1, 3.2, 3.3). In this section's experiments, Qolo is driven by RDS which receives a constant forward-pointing command (i.e. with vanishing nominal angular velocity) simulating a driver's primitive input. RDS uses the same parameters' values given already for the experiments in simulation (Sec. 3.4), except for $\hat{a}$=1.5 m/s^2, $\hat{\alpha}$=1.5 rad/s^2. The experiments' videos are available in the chapter's supplementary material.

3.5.1 Implementation

The robot's sensors include a front and a rear LiDAR and a front RGB-D camera. They inform the modules for SLAM, person detection tracking and collision avoidance.

SLAM

The robot estimates its own trajectory by matching scans from the rear LiDAR using the ROS package `hector_slam` [93], which allows the tracker to transform the sensors' spatial data into a static fixed reference frame and to estimate obstacles' absolute velocities.

Person detection tracking

The module tracks persons' positions from both LiDARs and the camera's RGB images (using the pipeline in [94]).

Collision avoidance

The module implements RDS or the baseline method (as in Sec. 3.4). It treats every scanpoint from both LiDARs as a separate circular obstacle with a small radius and zero velocity. Further, every person track is perceived as a circular obstacle with 0.3 m radius and with the track's estimated velocity.

3.5.2 Test in a Static Environment

The test (in Fig. 3.8, top) compares how RDS and the baseline method can assist passing through a door. As the nominal command would drive the robot forward into the door frame, correction is necessary to avoid the collision and ideally lead through the door. The trajectories for the baseline method and RDS in Fig. 3.8 right and left, respectively, show that among both methods, only RDS can lead the robot through the door due to the tighter capsule shape representation.

3.5.3 Tests with Pedestrians

The following two tests evaluate the robot's ability to overtake pedestrians that move in the same direction but are distributed ahead of and around the robot.

Row of pedestrians

The experiment (in Fig. 3.8, middle row) involves three pedestrians walking next to each other, forming a line with a larger gap between two of them such that the robot could pass in between while respecting a comfortable distance.

With the baseline method (Fig. 3.8, right), the robot approaches the moving pedestrians and then alternates between different angles while attempting to pass through the gap, which is due to the fact that the perceived orientation of the gap oscillates as the foot patterns and relative advancement of the pedestrians vary slightly over time. Using RDS (Fig. 3.8, left), the robot adjusts its orientation early towards the gap in order to avoid colliding with the middle pedestrian, and then it moves straight forward and passes through the gap.

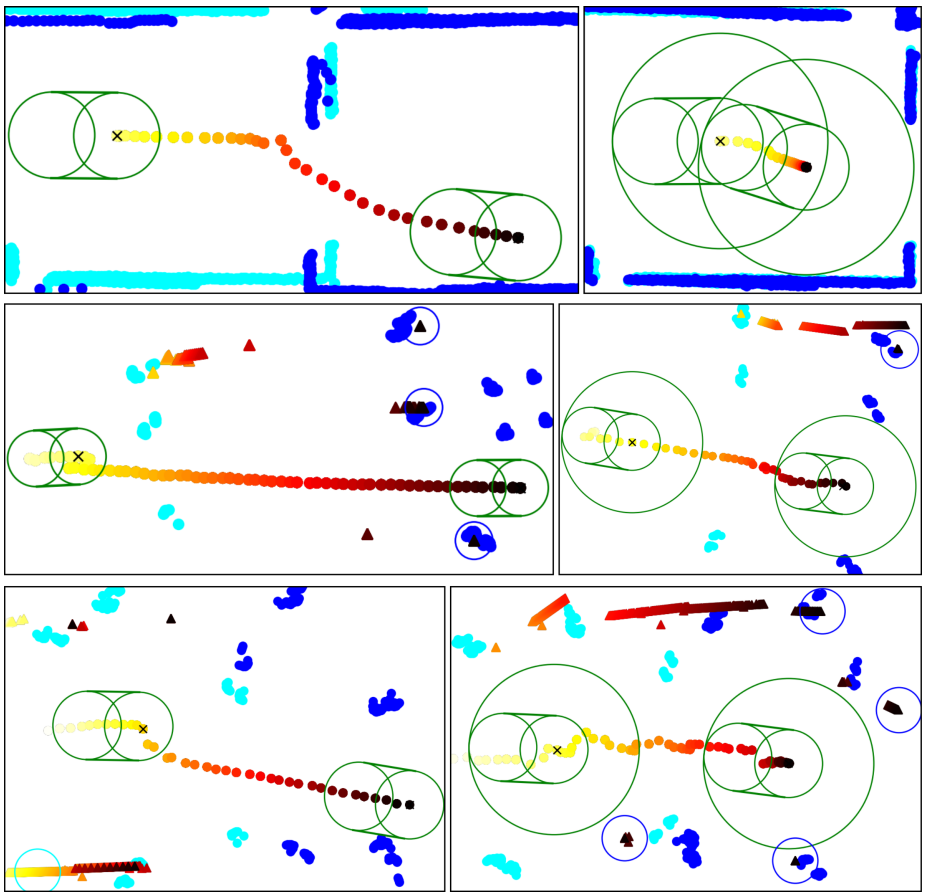

Figure 3.8: The robot Qolo uses RDS (left) or the baseline method (right) to pass through a door (top), overtake three pedestrians (middle) or a surrounding crowd (bottom). Its trajectory and the tracker's estimates of persons are shown (blobs and triangles, respectively, encoding time in yellow-red). Also, the robot's footprint (green capsule), tracked persons' footprints (circles) and LiDAR scanpoints (blobs) are shown at the beginning (in cyan) and at the end (in blue).

Unidirectional crowd

In the experiment (in Fig. 3.8, bottom), there are five pedestrians surrounding the robot. With the baseline method, the robot's motion is heavily constrained and it moves always towards small free areas created randomly by small irregularities in the crowd motion. With RDS, the robot adjusts its orientation early to avoid colliding with one pedestrian ahead, and subsequently it converges to a collision-free course and overtakes the surrounding crowd.

3.6 Discussion

The presented experimental results demonstrate the method's capability to maneuver through gaps and to avoid collisions with moving obstacles. As a baseline for comparison, we chose an ORCA-based method, which masks the robot's non-circular shape and non-holonomic kinematics by a circle. This choice is due to the focus on computationally lightweight methods that are applicable to such robots and address navigation in dynamic environments, which excludes most other methods based on velocity obstacles or more general control obstacles, as argued in the chapter's introduction. Methods which check candidate trajectories for collisions were not considered due to their increased computational burden or lack of continuity (if controls are sampled sparsely). Clearly, the chosen baseline method's performance is particularly negatively affected by the fact that the exemplary robot's bounding shape is rather elongated and to a large extent behind the wheel axle. However, many non-holonomic vehicles are indeed rather elongated, and in the case of Qolo, the robot's shape indeed extends more rearward than forward.

The simulations have been based on the "Crowds-by-Example" dataset [89] due to its high local densities of pedestrians, which allows to test the method's capability to maneuver through tightly spaced obstacles. Furthermore, the high-level reference trajectories given by the dataset have been used in order to focus solely on the task of local collision avoidance. If we compared our approach with the baseline without providing high-level reference trajectories, the baseline would possibly perform better than our approach, in sufficiently sparse crowds, since the baseline's enlarged bounding circle and abstraction from non-holonomic kinematics in combination with ORCA constitute an efficient planning method as long as space is not too constrained for such an approach.

To simulate the crowd's behavior for local collision avoidance, ORCA has been chosen. This choice is mainly due to the fact that ORCA, as a constraint-based method, does not interfere with the high-level reference trajectories unless a collision is imminent. Thus, the simulated crowd's motion remains close to the one given by the dataset, which is desirable for evaluating our approach under the most realistic conditions, as given by original arrangements of standing pedestrians and original patterns of crowd motion. Furthermore, this idea also underlies the metrics E_r and E_p, which measure agents' deviations from their reference trajectories. Assuming that agents would make the same choices

as recorded in the dataset unless there is an obstruction, these metrics serve to measure the robot's ability to blend in the crowd and find its way without obstructing pedestrians, despite its limited maneuverability and larger bounding shape (compared to the pedestrian whose place the robot is taking in the simulation).

Similarly, the metrics V_c and V_n, which measure the robot's impact on the crowd's speed serve to assess to which extent the robot obstructs pedestrians. As noted previously, the robot's enlarged bounding circle for the baseline method leads to unrealistic behavior of pedestrians, since they would not be aware of the enlarged bounding circle in reality. While this creates a bias towards larger values of the crowd's tracking error E_p, the metrics V_c and V_n are positively affected since the larger circle encourages efficient maneuvers around the robot's flat sides. Due to these effects, our simulations may not be conclusive regarding the obstruction which the modeled robot would constitute for pedestrians if deployed in a real crowd.

3.6.1 Limitations

The proposed method RDS has been presented in this chapter without giving guarantees for a minimum duration of collision-free motion. Its ability to guide the robot around obstacles has been demonstrated on a purely experimental basis, both in simulations and robot experiments. Thus, deriving such a guarantee for the proposed approach could be an interesting venue for future work. Also, the method relies on high-level guidance (e.g. to not get stuck in concave parts of obstacles' boundaries) and does not come with any guarantee of performing optimal maneuvers.

For its robotic implementation on the standing wheelchair Qolo, the method relies on the robot's on-board perception systems, which combine LiDAR and RGB-D cameras and include a pedestrian detection and tracking system. However, due to the tracking system's limited performance, obstacles' velocities could not be estimated reliably, and LiDAR scanpoints were used directly as obstacles with zero velocity to provide redundancy in case of tracking failure. While this introduces conservativeness in the robot's behavior with respect to moving obstacles, it is considered as an implementation-related issue that can be overcome in future work by using e.g. a faster but less selective object tracking method.

3.7 Conclusion

We have developed a method to apply the Velocity Obstacle (VO) to non-holonomic capsule-shaped robots and highlighted its effectiveness at avoiding collisions with static obstacles and interacting pedestrians, both in simulation and physical experiments with the robot Qolo. The comparison with another method using VO has demonstrated our method's advantage due to allowing maneuvers through narrow gaps. Our simulations of agents tracking real crowds'

motions show that the method avoids collisions efficiently such that the robot and pedestrians remain close to their references. We have described and evaluated four novel metrics to support this analysis.

Acknowledgement

We thank Dan Jia and Sabarinath Mahadevan for providing and supporting us with the person tracker. We thank Fabien Grzeskowiak and Julien Pettré for proposing the metrics that quantify the robot's impact on the crowd's speed.

Chapter 4

Robots' Motion Planning in Human Crowds by Acceleration Obstacles

Note: this chapter's contents are based on the following publication.
D. J. Gonon, D. Paez-Granados and A. Billard, "Robots' Motion Planning in Human Crowds by Acceleration Obstacles," in *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11236-11243, Oct. 2022, doi: 10.1109/LRA.2022.3199818.

For this chapter, a supplementary video is available online[1].

Abstract This chapter develops the theory of the Acceleration Obstacle (AO) as a mathematical construction for planning a robot's motion in dynamic environments (e.g. human crowds). The AO is analogous to the Velocity Obstacle, employed in the previous chapter, but assumes constant acceleration instead of velocity. The geometric properties of AO are analyzed and a direct sampling-free algorithm is proposed to approximate its boundary by linear constraints. The resulting controller is formulated as a quadratic program and evaluated in interaction with simulated bi-directional crowd flow in a corridor. A comparison to alternative robotic controllers is carried out, considering the robot's and the crowd's performance and the robot's behavior with respect to emergent lanes. Results indicate that the robot can achieve higher efficiency outside lanes.

4.1 Introduction

The aforegoing chapter proposed a method to correct a robot's input velocity commands such that it avoids collisions, based on the assumption that the

[1] `https://doi.org/10.1109/LRA.2022.3199818/mm1`

robot and obstacles move with constant velocities in the near future. This chapter assumes constant accelerations instead, in order to tackle more dynamic maneuvers and to account more naturally for agents' acceleration bounds.

Robots that navigate in human crowds as seamlessly as pedestrians are a persistent goal of recent work in robotics due to applications as delivery robots or smart wheelchairs. Some works [29, 95] focus on predicting pedestrians' motion in the near future, which allows to choose a complementary action for a robot amidst them. Other methods [27, 96] model the crowd's uncertain behavior to find motions for the robot that are safe under worst case assumptions. Geometric approaches, such as the Velocity Obstacle (VO) [19, 20], allow to command the robot such that it avoids collisions with obstacles whose motion is known in the near future. In this chapter, we investigate the Acceleration Obstacle (AO) in analogy to VO, assuming that the robot and obstacles maintain constant relative acceleration (instead of velocity as for VO) over a short time span. Our choice to assume constant acceleration is motivated by the mathematical simplicity of this motion model, which facilitates our analysis of AO.

The AO for a given obstacle defines a set of relative accelerations whose constant application will lead to a collision before reaching a time horizon. In comparison to the VO, whose shape can be described as a cone with a round truncated tip, the AO exhibits a more complex shape, whose entire boundary is curved in general. Similar shapes have been defined in prior work [40, 42], where typically, the boundary is approximated by a linear constraint, which is imposed on the robot's command. However, the boundary's geometric properties and dependence on initial conditions (i.e. the relative position and velocity for AO) has not been studied in depth in prior work. Our analysis reveals how the geometric properties of AO emerge (Sect. 4.2). Exploiting this analysis, a novel algorithm is proposed which derives a linear approximation of the AO via a few geometric and algebraic computations in closed form (Sect. 4.3).

Previous work [40] has conceived the AO without studying it further, arguing that constant accelerations are rarely maintained over extended time spans, and proposing instead the related concept of the Acceleration Velocity Obstacle (AVO), which assumes exponential adoption of a target velocity. However, our experiments of navigation in human crowds show that a small time horizon is sufficient to avoid collisions under realistic acceleration capabilities. As an advantage over AVO, the AO requires the smallest peak deceleration for braking in front of an obstacle, which we illustrate in simulation. We also show that the AO can be expressed via the AVO and a limit operation. Therefore, the theory and algorithm presented in this chapter appear to generalize to AVO.

Acceleration bounds are also considered by another geometric approach termed as the forbidden velocity map [97], which specifies for any given direction the largest speed which still allows the robot to brake before a collision occurs. The AO is a more general approach as its motion model includes curved trajectories and thus allows to plan maneuvers which apply maximum acceleration to avoid collisions by passing at the side of an obstacle rather than braking in front of it.

The experimental evaluation in this chapter focuses on navigation in bi-directional crowd flow through a corridor. On the one hand this scenario is a standard and well studied case in crowd simulation (see e.g. [56, 65]), for which standard metrics have been defined to quantify efficiency and interaction intensity with oncoming agents [98]. On the other hand, it is also a case of high practical relevance, considering that urban environments often exhibit a network of corridors, sidewalks, and promenades, on which motion is predominantly bi-directional. We simulate such bi-directional flows by the Social Force Model (SFM) [46]. The SFM is chosen due its ability to reproduce the phenomenon of dynamic lane formation and due to its simple formulation in terms of position-dependent repulsive forces.

We show that the AO with the proposed algorithm enables the robot to navigate efficiently and avoid almost all collisions, outperforming pedestrians, which are simulated by the SFM [46]. As a baseline, we use the SFM to control the robot like a pedestrian, in order to compare the case of a robot whose behavior is rather different from the crowd and the case of a robot with the same behavior as the crowd. To gain further insight in the two behaviors' specific characteristics and their possible interactions, our evaluation includes a third method, which combines the AO-based approach and the SFM to obtain a controller for the robot. Since both the AO and the SFM refer to accelerations, it is straightforward to combine them.

We also study the robot's behavior with respect to emergent lanes in the crowd's motion. Lanes lead to more efficient motions [98] and have been shown to emerge under a variety of interaction laws, including the SFM [46, 98] and the reciprocal VO [20, 99], which implements local pairwise collision avoidance. However, a robot's interaction with emergent patterns (such as lanes) arising in a crowd of agents whose characterstics differ from the robot, has rarely been evaluated, to the best of our knowledge. We systematically evaluate how the robot's method for navigation affects its own and the crowd's performance (Sect. 4.4). Specifically, we count collisions and near misses between the robot and the crowd and quantify the robot's and the crowd's efficiency of motion in terms of velocity and path length. Additionally, we measure the robot's integration in lanes and its interaction intensity with the counter flow.

Notation: Let $\mathbf{p} \times \mathbf{q} \coloneqq p_1 q_2 - p_2 q_1$ denote the cross product for vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}^2$. For a set $G \subset \mathbb{R}^2$, let $\partial G \subset \mathbb{R}^2$ denote its boundary and $|G|$ its cardinality. Let $D(\mathbf{p}, \varrho) \coloneqq \left\{\mathbf{z} \,|\, \varrho \geq |\mathbf{z} - \mathbf{p}|^2\right\} \subset \mathbb{R}^2$ denote the closed disk with radius ϱ centered on $\mathbf{p}$. We use $\dot{\mathbf{x}} \coloneqq \mathrm{d}\mathbf{x}/\mathrm{d}t$.

4.2 Acceleration Obstacles

For a mobile robot and an obstacle (cf. Fig. 4.1) with the respective positions $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^2$, we consider the task of collision-free navigation. Let $\mathbf{x} \coloneqq \mathbf{x}_1 - \mathbf{x}_2$ define the relative position. The two agents' bodies' shapes are bounded by two circles whose radii are given by r_1, r_2, and whose centers are at $\mathbf{x}_1$, $\mathbf{x}_2$, respectively. With the combined radius $R \coloneqq r_1 + r_2$, a collision is said to occur

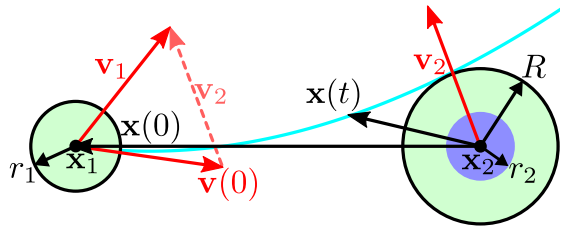

Figure 4.1: The state of the robot and the obstacle with the respective positions $\mathbf{x}_1$, $\mathbf{x}_2$ and velocities $\mathbf{v}_1$, $\mathbf{v}_2$ is represented for AO by the relative position $\mathbf{x}$ and velocity $\mathbf{v}$ and the collider with the combined radius $R = r_1 + r_2$. A motion (cyan) over time t with constant acceleration of $\mathbf{x}(t)$ is shown.

if $|\mathbf{x}| \leq R$.

For defining the AO, we consider the motion model of constant relative acceleration $\mathbf{a} = const$. The family of relative trajectories satisfying $\ddot{\mathbf{x}} = \mathbf{a}$ is given by

$$\mathbf{x}(t; \mathbf{a}) \coloneqq \mathbf{x}_o + \mathbf{v}_o t + \mathbf{a}t^2/2 \tag{4.1}$$

with the parameter $\mathbf{a}$, for given initial conditions $\mathbf{x}(0) = \mathbf{x}_o$ and $\dot{\mathbf{x}}(0) = \mathbf{v}_o$. The model $\ddot{\mathbf{x}} = const$ can represent two agents applying constant forces (to avoid each other) and thereby approximate pedestrians' behavior over a short time window. The AO is the set of accelerations $\mathbf{a}$ that lead to a collision before a time horizon τ. It is formalized as follows.

Definition 1 (Acceleration Obstacle) *The Acceleration Obstacle with the time horizon τ for the robot and a given obstacle with the relative position and velocity $\mathbf{x}_o$ and $\mathbf{v}_o$, respectively, and the combined radius R is the set*

$$AO_\tau(\mathbf{x}_o, \mathbf{v}_o, R) \coloneqq \{\mathbf{a} \,|\, \exists t \in [0, \tau] : |\mathbf{x}(t; \mathbf{a})| \leq R\}\,. \tag{4.2}$$

We note that $|\mathbf{x}(t; \mathbf{a})| \leq R \iff \mathbf{a} \in D(\mathbf{c}(t), r(t))$ with

$$\mathbf{c}(t) \coloneqq -2\left(\mathbf{x}_o + t\mathbf{v}_o\right)/t^2, \quad r(t) \coloneqq 2R/t^2, \tag{4.3}$$

due to (4.1). Thus, we express the AO_τ given by (4.2) as

$$\begin{aligned} AO_\tau &= \{\mathbf{a} \,|\, \exists t \in (0, \tau] : \mathbf{a} \in D(\mathbf{c}(t), r(t))\} \\ &= \bigcup_{t \in (0,\tau]} D(\mathbf{c}(t), r(t)) = \bigcup_{t \in (0,\tau]} D_t \end{aligned} \tag{4.4}$$

with $D_t \coloneqq D(\mathbf{c}(t), r(t))$. The final expression (4.4) describes the AO as a union of disks defined over the time span $(0, \tau]$ (cf. Fig. 4.2-a), i.e. a planar shape analogous to a *canal* (i.e. a union of spheres) in $\mathbb{R}^3$ [100]. The last disk's boundary $\partial D_\tau \cap \partial AO_\tau$ is associated with collisions at the time horizon, whereas the remaining boundary corresponds to grazing motions. It is generally curved, since the acceleration term in (4.1) is combined with the growing translation $t\mathbf{v}_o$, such that $\mathbf{a}$ must rotate to reach the collider at different times.

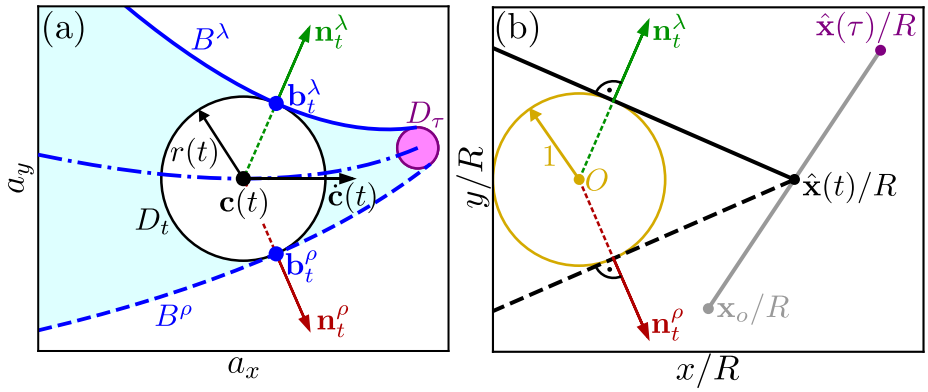

Figure 4.2: (a) Showing an AO_τ, its sweeping disk D_t with center $\mathbf{c}(t)$ and radius $r(t)$ for some time $t \in (0, \tau]$, the local boundary's elements $\mathbf{b}_t^\lambda$ and $\mathbf{b}_t^\rho$, which are left and right with respect to the center's derivative $\dot{\mathbf{c}}(t)$, and their unions B^λ and B^ρ, respectively, up to the time horizon's disk D_τ. (b) Showing the (normalized) extrapolation $\hat{\mathbf{x}}(t)/R$ from $\mathbf{x}_o/R$ over $[0, \tau]$ in the space of the normalized relative position coordinates x/R and y/R. The unit circle's tangents through $\hat{\mathbf{x}}(t)/R$ are orthogonal to the AO_τ's local normals $\mathbf{n}_t^\lambda$ and $\mathbf{n}_t^\rho$ at any given time $t \in (0, \tau]$ with $|\hat{\mathbf{x}}(t)| > R$.

4.2.1 Geometric Properties of Acceleration Obstacles

For discussing the shape of the AO, we first define precise notions of its boundary and normals (cf. also Fig. 4.2).

Definition 2 (Local Boundary/Normals) *For an AO_τ, the local boundary at $t \in (0, \tau]$ is the set*

$$B_t := \lim_{\Delta t \to 0} \partial D_t \cap \partial D_{t+\Delta t}. \tag{4.5}$$

The set of (outwards) normals at $t \in (0, \tau]$ is defined as

$$N_t := \{\mathbf{u} \,|\, \mathbf{u} = (\mathbf{b}_t - \mathbf{c}(t))/r(t), \mathbf{b}_t \in B_t\}. \tag{4.6}$$

Thus, we associate with any given element $\mathbf{b}_t \in B_t$ a normal $\mathbf{n}_t \in N_t$. If $|N_t| = 2$, the two normals are denoted by $\mathbf{n}_t^\lambda$ and $\mathbf{n}_t^\rho$, such that $\dot{\mathbf{c}} \times \mathbf{n}_t^\lambda > 0$ and $\dot{\mathbf{c}} \times \mathbf{n}_t^\rho < 0$, and termed as *left* and *right* (relative to $\dot{\mathbf{c}}$), respectively. Their corresponding boundary points are denoted by $\mathbf{b}_t^\lambda$, $\mathbf{b}_t^\rho$, accordingly.

The normals N_t can be constructed via the *extrapolation* $\hat{\mathbf{x}}(t) := \mathbf{x}_o + \mathbf{v}_o t/2$ according to the following proposition (cf. Fig. 4.2), which is proven in the Appendix 7.A.

Proposition 1 (AO-normals) *For the $AO_\tau(\mathbf{x}_o, \mathbf{v}_o, R)$, the set of normals at any $t \in (0, \tau]$ is given by*

$$N_t = \left\{\mathbf{u} \middle| \mathbf{u}^T \hat{\mathbf{x}}(t) - R = 0, |\mathbf{u}| = 1\right\}. \tag{4.7}$$

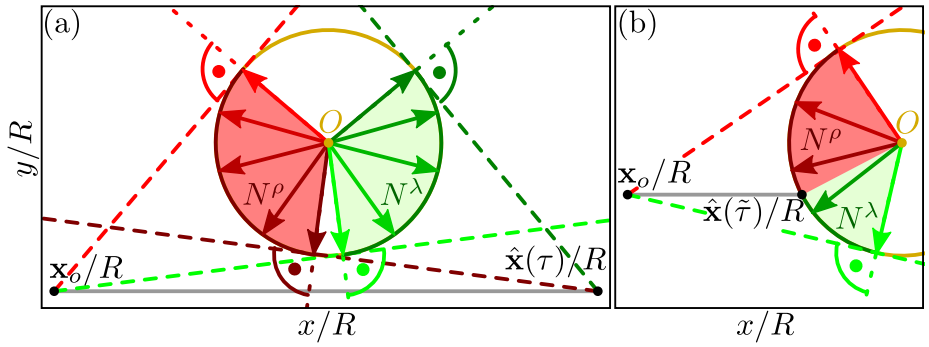

Figure 4.3: The tangents through the (normalized) extrapolation $\hat{\mathbf{x}}(t)/R$ touch the unit circle at the respective AO_τ's normals. Thereby, the left and right normals' unions N^λ and N^ρ are generated, respectively, as t varies over $(0, \tilde{\tau})$. The reduced time horizon $\tilde{\tau}$ can be (a) equal to the time horizon τ or (b) smaller than τ (if the line from $\mathbf{x}_o$ to $\hat{\mathbf{x}}(\tau)$ intersects the unit circle).

Thus, the normals are constructed from tangents to the unit circle (i.e. the normalized collider) through the normalized extrapolation $\hat{\mathbf{x}}(t)/R$ as the Fig. 4.2-b illustrates. One can also use $\hat{\mathbf{x}}$ instead of $\dot{\mathbf{c}}$ as reference to identify which is the left or right normal, since $\dot{\mathbf{c}}(t) = 4\hat{\mathbf{x}}(t)/t^3$ holds, as can be derived from (4.3), i.e. the two references differ only by scaling.

Since $|B_t| = |N_t|$, the local boundary exists at a given t if (4.7) admits any solutions, which is the case iff $|\hat{\mathbf{x}}(t)| \geq R$. Thus, assuming that the initial state is not in collision, the AO_τ's local boundary exists from time zero up to the *reduced time horizon* $\tilde{\tau}$ defined as (cf. Fig. 4.3-(b))

$$\tilde{\tau} := \max\left\{t | \forall t' < t : |\hat{\mathbf{x}}\left(t'\right)| > R, t \leq \tau\right\}. \tag{4.8}$$

Further, $\tilde{\tau}$ is sufficient as a horizon, in the sense of the following proposition, which is proven in the Appendix 7.B.

Proposition 2 (AO-reduction) *The interval* $(0, \tilde{\tau}]$ *already generates the complete* AO_τ*, i.e.* $AO_{\tilde{\tau}} = AO_\tau$*.*

Accordingly, we define the *left local boundary's union* $B^\lambda := \{\mathbf{z} | \mathbf{z} = \mathbf{b}_t^\lambda, t \in (0, \tilde{\tau})\}$ and the *left normals' union* $N^\lambda := \{\mathbf{u} | \mathbf{u} = \mathbf{n}_t^\lambda, t \in (0, \tilde{\tau})\}$, uniting elements up to $\tilde{\tau}$. The *right local boundary's union* B^ρ and the *right normals' union* N^ρ are defined analogously. The Fig. 4.2 and 4.3 illustrate respectively the local boundaries' and normals' unions.

We partition the AO_τ's boundary ∂AO_τ into the *left boundary* $\mathcal{L} \subseteq B^\lambda$, the *right boundary* $\mathcal{R} \subseteq B^\rho$, and the *cap* $\mathcal{C} \subset \partial D_{\tilde{\tau}}$, where $\mathcal{C}$ contains at least a single point and connects $\mathcal{L}$ and $\mathcal{R}$. As B^λ or B^ρ can form a self-intersecting curve, a point on B^λ or B^ρ does not necessarily belong to the AO_τ's actual boundary ∂AO_τ.

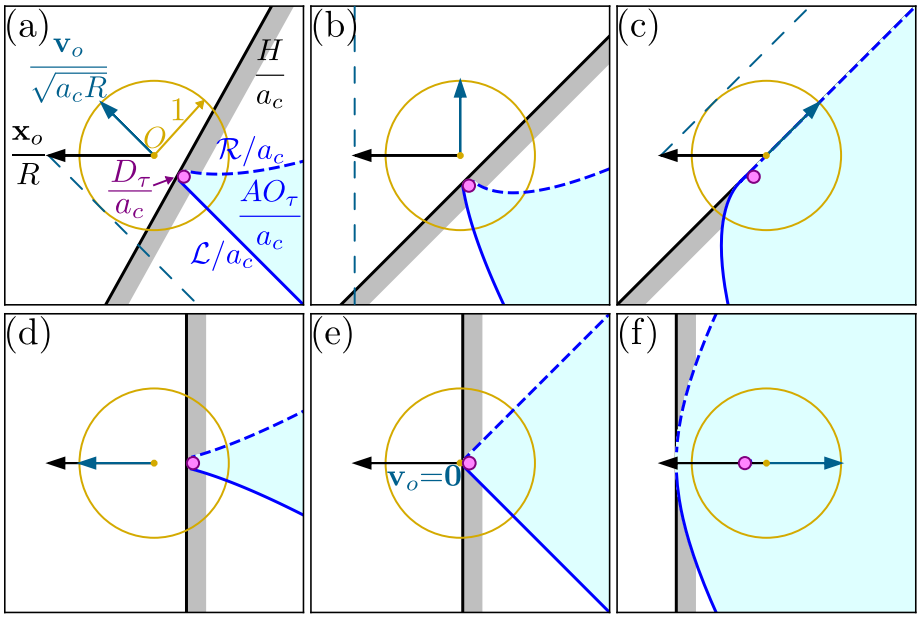

Figure 4.4: Each plot (a-f) shows an AO_τ, its left and right boundaries $\mathcal{L}$ and $\mathcal{R}$, respectively, the underlying relative initial position $\mathbf{x}_o$ (fixed) and velocity $\mathbf{v}_o$ (varying), the final disk D_τ, and the halfplane H by the Algorithm 2, normalized via the combined radius R and the unit acceleration a_c. The initial course is post-colliding (d), left-passing (b), and pre-colliding (f), or an edge case, if the dashed line through $\mathbf{x}_o/R$ and parallel to $\mathbf{v}_o$ touches the unit circle (a, c) or $\mathbf{v}_o = 0$ (e).

The Proposition 3 below characterizes the AO_τ's local convexity or concavity on its boundary's partitions $\mathcal{L}$ and $\mathcal{R}$ dependent on the following definition's terminology.

Definition 3 (Initial Course) *The initial course defined by the tuple* $(\mathbf{x}_o, \mathbf{v}_o, R)$ *is termed as*

- *pre-colliding if* $\exists t > 0 : |\hat{\mathbf{x}}(t)| < R$
- *post-colliding if* $\exists t < 0 : |\hat{\mathbf{x}}(t)| < R$
- *left-passing if* $\forall t : \mathbf{v}_o \times \hat{\mathbf{x}}(t) > 0 \wedge |\hat{\mathbf{x}}(t)| > R$
- *right-passing if* $\forall t : \mathbf{v}_o \times \hat{\mathbf{x}}(t) < 0 \wedge |\hat{\mathbf{x}}(t)| > R$.

The Fig. 4.4 illustrates the following proposition, which is proven in the Appendix 7.C.

Proposition 3 (AO-shape) *The* AO_τ *is locally convex on its left boundary, iff the initial course is left-passing or pre-colliding, or it is locally concave there iff*

it is right-passing or post-colliding. And the analogous statement interchanging the words "left" and "right" holds as well.

4.2.2 Connection to Acceleration Velocity Obstacles

The family of relative trajectories $\mathbf{x}(t;\mathbf{v})$ underlying AVO [40] results from proportional velocity control with some gain $1/\delta > 0$ and setpoint $\mathbf{v} = const$, i.e. the motion satisfies

$$\ddot{\mathbf{x}} = (\mathbf{v} - \dot{\mathbf{x}})/\delta = \mathbf{a} - \dot{\mathbf{x}}/\delta \tag{4.9}$$

with $\mathbf{a} := \mathbf{v}/\delta$. When $\delta \to \infty$ and $|\mathbf{v}| \to \infty$ while maintaining a given value of $\mathbf{a}$, (4.9) approaches the motion model underlying AO, namely $\ddot{\mathbf{x}} = \mathbf{a}$. Indeed, it holds

$$\lim_{\delta\to\infty} AVO_\tau(\delta)/\delta \to AO_\tau \tag{4.10}$$

in the following sense. Denoting the AVO's center and radius functions by $\tilde{\mathbf{c}}$ and $\tilde{r}$, respectively, it holds that $\tilde{\mathbf{c}}(t;\delta)/\delta \to \mathbf{c}(t)$ and $\tilde{r}(t;\delta)/\delta \to r(t)$ as $\delta \to \infty$, for any fixed $t > 0$. The proof is given in the Appendix 7.D.

4.3 Method

The technique we propose for a robot's navigation between multiple agents uses the AO and the theoretical results from the previous Sect. 4.2.

4.3.1 Construction Scheme for AO-constraints

From the AO_τ between the robot and a given obstacle, a linear constraint is constructed for the robot's acceleration. Let $H(\mathbf{n}, b) := \{\mathbf{z} \,|\, \mathbf{n}^T\mathbf{z} \le b\}$ denote the closed halfplane with the outwards normal $\mathbf{n}$ and its boundary's offset b from the origin (along $\mathbf{n}$). For shorter notation, let $\tilde{\mathbf{x}} := \hat{\mathbf{x}}(\tilde{\tau})$. Given the parameters defining an AO_τ, the Algorithm 2 constructs a halfplane H which covers the AO_τ and whose boundary touches its cap $\mathcal{C}$, as illustrated by the Fig. 4.4 for various AO_τ.

The Fig. 4.5 illustrates the geometric constructions performed by the Algorithm 2. To explain its steps and understand why the resulting halfplane is always conservative, we first note that for any $\mathbf{n}$, choosing b as in Line 4 ensures that ∂H touches $D_{\tilde{\tau}}$. As the Fig. 4.5 indicates, the algorithm constructs $\mathbf{n}$ such that the left and right boundary's normals are to its left and right, respectively, i.e. $\mathbf{n} \times \mathbf{n}' > 0, \forall \mathbf{n}' \in N^\lambda$ and $\mathbf{n} \times \mathbf{n}' < 0, \forall \mathbf{n}' \in N^\rho$. Thus, ∂H can neither intersect B^λ nor B^ρ, leading to the following proposition.

Proposition 4 (Conservative Halfplane) *The halfplane H constructed by the Algorithm 2 is a conservative approximation of the AO_τ, i.e. $H \supseteq AO_\tau$.*

Algorithm 2 Conservative Approximation of AO_τ

Input: $\mathbf{x}_o$, $\mathbf{v}_o$, R
Output: $H(\mathbf{n}, b) \supset AO_\tau(\mathbf{x}_o, \mathbf{v}_o, R)$
1: $OC \leftarrow$ occlusion cone $OC(\mathbf{x}_o, R)$
2: $\bar{\mathbf{x}} \leftarrow \tilde{\mathbf{x}}$'s orthogonal projection on ∂OC
3: $\mathbf{n} \leftarrow OC$'s outwards normal at $\bar{\mathbf{x}}$
4: $b \leftarrow \mathbf{n}^T \mathbf{c}(\tilde{\tau}) + r(\tilde{\tau})$
5: **return** $H(\mathbf{n}, b)$

A more formal proof is given in the Appendix 7.E.

As AO may cover an infinitely large radius around the origin, we impose an upper bound $b_{max} >> a_{max}$ on b, i.e. we actually compute the limited offset $\hat{b} := \min(b, b_{max})$, where b is defined by the Algorithm 2. This step ensures that the resulting halfplane $\hat{H} := H(\mathbf{n}, \hat{b})$ can serve as a (soft) constraint in a numerical optimization and also, from a theoretical point of view, does not dominate entirely the optimization's solution (Sect. 4.3.3). In this work, we assume the obstacle's acceleration to be zero, such that $\hat{H}$ can be treated as in the space of the robot's absolute acceleration.

4.3.2 Limits of Acceleration and Velocity

Assuming that the robot's acceleration and velocity are restricted to given feasible convex sets A and V, respectively, our approach represents them as well by linear constraints on the robot's acceleration. This work sets $A = D(\mathbf{0}, a_{max})$ and $V = D(\mathbf{0}, v_{max})$, for simplicity, with a_{max}=2 ms^{-2} and v_{max}=1.7 m/s. Since V is in the space of velocity, it is transformed into the set of accelerations that keep the velocity within V over some time horizon $T_v > 0$, chosen as T_v=0.25 s. Thus, let $\tilde{V} := (-\mathbf{v}_1(0) \oplus V)/T_v$, where $\mathbf{v}_1(0)$ is the robot's current velocity and $\oplus$ denotes Minkowski addition. Then, A and $\tilde{V}$ are under-approximated by the convex polygons $\mathcal{A} \subseteq A$ and $\mathcal{V} \subseteq \tilde{V}$, respectively. In this work, we define both approximations as the intersection of 16 respective halfplanes.

4.3.3 Command Optimization

Denoting the halfplane's parameters computed for the i-th obstacle by $\mathbf{n}_i$, $\hat{b}_i$, we combine these constraints from M obstacles with the objective of applying a given nominal acceleration $\bar{\mathbf{a}}_1$ in the quadratic program

$$\begin{aligned} \mathbf{a}_1^* := \arg\min_{\mathbf{a}} & \left|\mathbf{a} - \bar{\mathbf{a}}_1\right|^2 \\ \text{s.t. } & \mathbf{n}_i^T \mathbf{a} \geq \hat{b}_i, \ i = 1, ..., M \\ & \mathbf{a} \in \mathcal{A} \cap \mathcal{V} \end{aligned}$$

whose solution $\mathbf{a}_1^*$ defines the robot's command. For the case that the above problem turns out to be infeasible, the control law is defined by the alternative

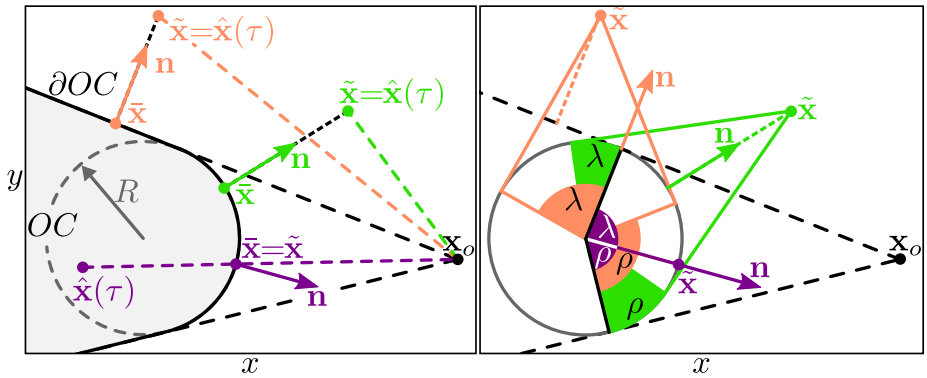

Figure 4.5: Left: The Algorithm 2 projects the point $\tilde{\mathbf{x}}$ on the boundary ∂OC of the occlusion cone OC for the collider of radius R and the relative position $\mathbf{x}_o$. There at $\bar{\mathbf{x}}$, the surface normal $\mathbf{n}$ is extracted (shown for three $\tilde{\mathbf{x}}$ in purple, green, orange). The point $\tilde{\mathbf{x}}$ is the halftime prediction at the time horizon $\hat{\mathbf{x}}(\tau)$ if the latter is outside OC (green, orange cases), else $\tilde{\mathbf{x}}$ is at the intersection $\overline{\mathbf{x}_o\hat{\mathbf{x}}(\tau)} \cap \partial OC$ (purple case). Right: the $AO_{\tilde{\tau}}$'s left and right local boundaries' normals are constructed by drawing tangents to the circle and through $\hat{\mathbf{x}}(t)$ as t varies over $(0, \tilde{\tau}]$, and their resulting orientations' respective intervals are shown and labeled by λ and ρ, respectively.

problem to minimize the largest violation of an obstacle's constraint, as in prior work using VO [20]. This corresponds to the linear program

$$\begin{aligned} \mathbf{a}_1^*, \xi^* := \arg\min_{\mathbf{a},\xi} \; & \xi \\ \text{s.t. } & \mathbf{n}_i^T \mathbf{a} \geq \hat{b}_i - \xi, \; i = 1, ..., M \\ & \mathbf{a} \in \mathcal{A} \cap \mathcal{V} \end{aligned}$$

whose feasibility is that of $\mathcal{A} \cap \mathcal{V}$. If it is infeasible, the robot must be moving with a velocity outside $\mathcal{V}$, which may happen in practice. In the infeasible case, the command is chosen as $\mathbf{a}_1^* = -a_{max}\mathbf{v}_1/|\mathbf{v}_1|$ (opposed to the current velocity).

4.4 Experiments

The next Sect. 4.4.1 compares AO with AVO. Robots' navigation using AO and the SFM in bi-directional crowd flow simulations is evaluated in the Sect. 4.4.2. We provide videos of the crowd simulations in the supplementary material.

4.4.1 Comparison with AVO

In this experiment, the robot approaches a flat wall and brakes to avoid a collision. Since the obstacle is conceptually equal to a circle with infinite radius, the corresponding AVO and AO are halfplanes and one can directly interpret

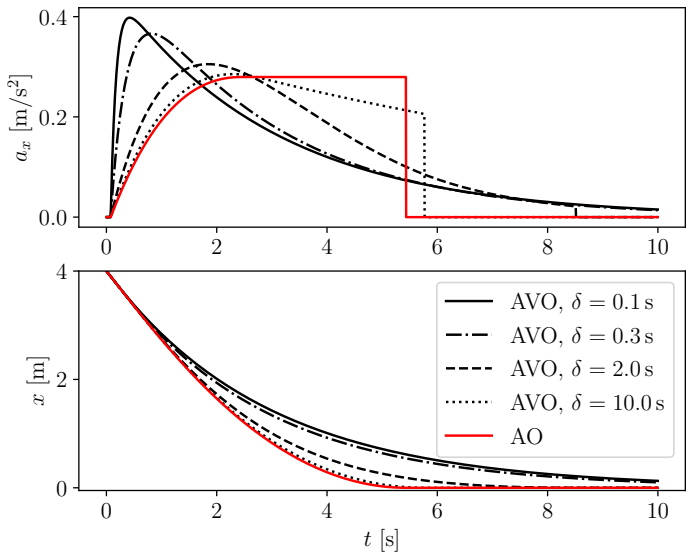

Figure 4.6: A robot's braking acceleration a_x (top) and position x (bottom) over time are shown as it perpendicularly approaches a flat wall and avoids a collision (at $x = 0$) using AVO (with different relaxation times δ) or AO.

them as a linear constraint, which we then impose on the robot's motion. The largest absolute acceleration which the robot applies when braking in front of an obstacle is smaller for AO than for AVO with any finite relaxation time δ, as the Fig. 4.6 illustrates.

4.4.2 Crowd–robot interaction in a Corridor

In the simulations, the robot travels in a corridor amidst two streams of pedestrians which move in mutually opposite directions (cf. Fig. 4.7) and are governed by the Social Force model (SFM) for pedestrian dynamics [46].

The Social Force Model (SFM)

The SFM describes a pedestrian as a body with position $\mathbf{x}_i$ and velocity $\mathbf{v}_i$. Its acceleration is described as the sum of a driving force $\mathbf{f}_i$ towards the preferred velocity and repelling forces $\mathbf{f}_{i,j}, \tilde{\mathbf{f}}_{i,k}$ due to other pedestrians and walls, respectively. The driving force is defined as

$$\mathbf{f}_i \coloneqq (\mathbf{v}_{des,i} - \mathbf{v}_i) / \gamma, \tag{4.11}$$

where $\mathbf{v}_{des,i}$ denotes the desired velocity and γ is the relaxation time. We denote by $\mathbf{e}_{i,j}$ the unit vector pointing from the j-th to the i-th pedestrian and by $d_{i,j}$ the distance between them (minus their radii). The force due to other pedestrians comprises long-range interactions with strength A_1 and range B_1

and short-range (physical) interactions with strength A_2 and range B_2 according to

$$\mathbf{f}_{i,j} = \mathbf{e}_{i,j} \left(A_1 e^{-d_{i,j}/B_1} w(\varphi_{i,j}) + A_2 e^{-d_{i,j}/B_2} \right), \tag{4.12}$$

with $w(\varphi_{i,j}) := \lambda + (1-\lambda)(1+\cos\varphi_{i,j})/2$ and $\cos\varphi_{i,j} = -\mathbf{e}_{i,j} \cdot \mathbf{v}_i$, where $\lambda \in [0,1]$ can be chosen below 1 to reduce interactions with pedestrians not in the viewing direction.

Similarly for walls, we denote by $\tilde{\mathbf{e}}_{i,k}$ the unit vector pointing from the k-th wall's nearest point to the i-th pedestrian and by $\tilde{d}_{i,k}$ the distance between them (minus the pedestrian's radius). The interaction force due to walls is defined as

$$\tilde{\mathbf{f}}_{i,k} = \tilde{\mathbf{e}}_{i,k} A e^{-\tilde{d}_{i,k}/B}. \tag{4.13}$$

with A and B specifying the repulsion's strength and range, respectively. Thus the i-th pedestrian's acceleration is given by

$$\ddot{\mathbf{x}}_i = \mathbf{f}_i + \sum_{j \neq i} \mathbf{f}_{i,j} + \sum_k \tilde{\mathbf{f}}_{i,k}. \tag{4.14}$$

In our simulations, we set the SFM parameters as [46,101] to $\gamma = 1\,\mathrm{m/s}$, $A_1 = 0.42\,\mathrm{m/s^2}$, $B_1 = 1.65\,\mathrm{m}$, $A_2 = 3\,\mathrm{m/s^2}$, $B_2 = 0.2\,\mathrm{m}$, $\lambda = 0.75$, $A = 5\,\mathrm{m/s^2}$, $B = 0.2\,\mathrm{m}$.

Control of the robot

For controlling the robot, we compare the method using the Acceleration Obstacle (AO) as defined in the previous Sect. 4.3 to two alternative methods. The first alternative applies the SFM to determine the robot's acceleration in the same way as for pedestrians. The second alternative combines the SFM and AO by computing the robot's nominal acceleration $\bar{\mathbf{a}}_1$ from the SFM and executing the method of AO to determine the actual acceleration $\mathbf{a}_1^*$ taking into account constraints due to pedestrians. We abbreviate the three methods for controlling the robot as AO, SF, and SF-AO, respectively. For AO, the nominal acceleration $\bar{\mathbf{a}}_1$ is computed according to (4.11).

Experimental Protocol

To obtain diverse states of the crowd in the corridor for initializing the experiments, we run a preliminary simulation of a system comprising 200 pedestrians (100 per direction) for a long duration. Starting from two separated streams (cf. Fig. 4.7), the system's state becomes less ordered after two minutes (in simulation), when we start to periodically take a snapshot of the system's state, with period Δt=20 s, until a set of K states is obtained (K=48). Due to chaotic fluctuations and spontaneous congestion, the crowd's density in the corridor varies across snapshots (from 0.28 to 0.5 persons per square meter).

In each snapshot, one particular pedestrian is chosen to be replaced by the robot, whereas the others serve to initialize the crowd in the actual experiments.

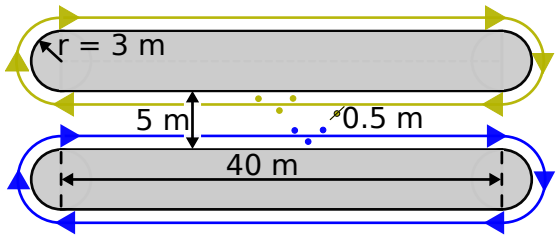

Figure 4.7: The crowd forms two streams (blue and yellow) through a corridor.

For each snapshot, we carry out one simulation of duration Δt for each method to control the robot and for three different preferred speeds for the robot, namely 1.1, 1.3, and 1.5 m/s. A radius of 0.25 m is assumed for both the robot and pedestrians. Each pedestrian's preferred speed is constant throughout all experiments and sampled from the normal distribution with mean 1.3 and standard deviation 0.3 m/s. For AO and SF-AO, we set the time horizon to τ=1 s.

Metrics

We compute the following metrics by considering only the time span when the robot is inside the corridor.

The robot's efficiency is quantified in two ways. Firstly, the metric v_r is defined as the ratio of its average velocity in its preferred direction over its preferred speed [98]. Secondly, the metric p_r is defined as the ratio of its change in position in its preferred direction over its path length. Thus, p_r measures the path's efficiency independent from the velocity. The metrics v_c and p_c measure the analogous quantities' averages over pedestrians' trajectories in the corridor.

The robot's integration in lanes is measured by the metric L, which denotes the temporal average of the number of pedestrians around the robot (i.e. 5 m close) that have the same preferred direction of motion and differ in the transversal coordinate (in the direction orthogonal to the walls) by at most 0.5 m from the robot. The metric I measures for pedestrians heading the other way than the robot the average acceleration induced by the robot via (4.12) in the component opposed to their preferred direction.

When two agents' distance falls below the sum of their radii, a collision is counted. A near miss is defined similarly as the event that the distance between two agents becomes lower than the sum of their radii plus 0.1 m but no collision occurs while this condition is present. The metrics C and NM count pedestrian-robot collisions and near misses, respectively, only considering events in the corridor.

Results

Comparing the robot's control methods SF, AO, and SF-AO, the Table 4.1 gives separate results for the robot's three different preferred velocities. It gives the

mean and standard deviation over the K executions for the velocity and path efficiency metrics, and the sum over the K executions for the collision and near miss metrics. Asterisks (*) indicate that a t-test rejects the null-hypothesis that the expected value of the difference between the respective metric of AO and SF-AO is equal to zero (p-value < 0.05). In this case, the difference is considered significant. Values in bold indicate the best performance.

When the robot uses AO, both the robot and the crowd move more efficiently in their preferred direction (and also faster), as the values of v_r, v_c, p_r, and p_c show. On the other hand, the method SF-AO leads to the smallest number of collisions and near misses between the robot and the crowd. The robot's tendency to integrate itself in lanes as measured by L is higher with SF-AO than with AO. However, its interaction intensity with the opposite stream as measured by I is lower with AO than with SF-AO. With SF, the robot's efficiency's mean value and the number of near misses are between SF-AO and AO. The largest number of collisions is reported for SF.

Table 4.1: Results of simulated bi-directional crowd flow in a corridor including a robot that uses SF, AO, or SF-AO.

	v_r [-]	v_c [-]	p_r [-]	p_c [-]	L [-]	I [m/s^2]	C [-]	NM [-]
robot's preferred speed = 1.1 m/s								
SF	0.80 ±0.24	0.73 ±0.14	0.92 ±0.13	0.919 ±0.059	4.1 ±1.8	0.008 ±0.011	30	64
AO	**0.87***±0.16	**0.74***±0.13	**0.98***±0.03	**0.921***±0.056	3.7*±1.9	**0.007** ±0.009	2	89*
SF-AO	0.76*±0.28	0.73*±0.14	0.88*±0.31	0.918*±0.060	**4.1***±1.8	0.008 ±0.011	**1**	**36***
robot's preferred speed = 1.3 m/s								
SF	0.72 ±0.19	0.73 ±0.14	0.93 ±0.10	0.918 ±0.059	4.1 ±1.7	0.008 ±0.009	32	55
AO	**0.85***±0.18	**0.74***±0.13	**0.98***±0.03	**0.920** ±0.057	3.6*±1.8	**0.007***±0.008	3	98*
SF-AO	0.67*±0.25	0.73*±0.14	0.87*±0.30	0.919 ±0.060	**4.0***±1.7	0.009*±0.012	**2**	**50***
robot's preferred speed = 1.5 m/s								
SF	0.65 ±0.19	0.73 ±0.14	0.92 ±0.13	0.918 ±0.060	3.9 ±1.9	0.009 ±0.011	36	67
AO	**0.83***±0.19	**0.74** ±0.13	**0.99***±0.03	0.919 ±0.057	3.4*±1.8	**0.006***±0.009	4	99*
SF-AO	0.62*±0.23	0.73 ±0.14	0.91*±0.17	**0.920** ±0.057	**3.9***±1.9	0.009*±0.014	**1**	**54***

Discussion

The robot progresses faster with AO because it can use any free space to navigate, in particular the corridor's middle, which is often free because pedestrians repell each other on a long range and therefore tend to stay close to the walls (cf. Fig. 4.8). In contrast to both other methods, AO lets the robot come arbitrarily close to pedestrians, which is reflected by the large number of near misses. Pedestrians tend to restore the distance since the robot repells them like a pedestrian. Therefore, the robot also makes pedestrians progress faster by pushing them forward when approaching from behind. This happens frequently because the crowd tends to form lanes and the robot is initialized in place of a pedestrian and thus often in a lane.

With SF-AO, the robot maintains a similar distance to pedestrians as they do among each other, as indicated by the smaller number of near misses than with AO. The distance margin leaves the robot more time to react to pedestrians' changing velocities, which explains why fewer collisions occur than with AO. Since a higher number of near misses implies a higher (actual and perceived) risk of collisions, navigation with AO is riskier but more efficient than with SF-AO, while only very few actual collisions occur, comparing to SF. Thus, the two methods AO and SF-AO represent different priorites on the safety–efficiency spectrum, which has been described in prior work [65] as trade-off inherent to navigation in a crowd.

While related work often uses efficiency to quantify lane formation [98, 102], the robot in our simulations can achieve higher efficiency outside lanes. The fact that AO achieves higher values of v_r, v_c, p_r, and p_c while exhibiting lower values of L and I than SF-AO can be explained as follows. The robot with AO may leave lanes earlier than with SF-AO and avoid the opposite stream successfully by exploiting narrow gaps without decreasing the opposite stream's efficiency. Thus, the robot's superior performance at avoiding oppositely headed pedestrians by AO (also reflected by low interaction intensity I) seems to overcompensate being less often in lanes. Considering that the crowd's efficiency is positively affected by AO (considering v_c, p_c), the robot does not seem to obstruct lane formation, even when not participating in it. The efficiency metrics exhibit larger standard deviations with SF-AO and SF, since the robot is repelled by the crowd and thus more sensitive to its fluctuating density.

The simplicity of the SFM, particularly the fact that it does not avoid collisions in a reliable manner, arguably render the robot's navigation task more difficult than under a more sophisticated crowd model. On the other hand, simulations are started from steady state conditions in the sense that lanes have been established already, which compensates for this lack of reliable collision avoidance in pedestrians' behavior, and which facilitates navigation for the robot.

An important qualitative difference between the SFM and AO can be seen in the fact that the SFM makes agents spread out over the entire space available to them, whereas AO in itself only generates motions to avoid collisions. Thus, navigating in a crowd governed by the SFM is particularly easy for an agent us-

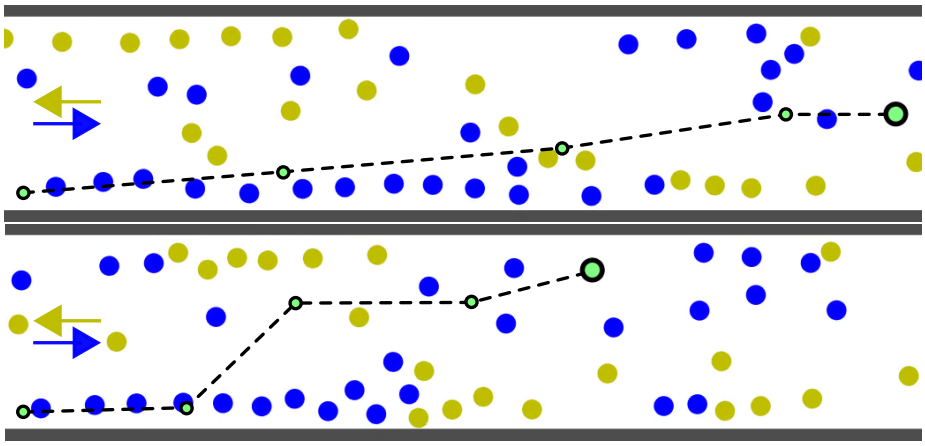

Figure 4.8: Both snapshots show a simulation from the same initial condition once with AO (top) and SF-AO (bottom). Pedestrians (blue/yellow) tend to form lanes close to the walls due to mutual repulsion. The robot (large black-green circle) has the same tendency with SF-AO, whereas it can navigate in the free middle between lanes when using AO. Its path is shown by connecting four past waypoints (dashed line/small black-green circles).

ing AO, since the crowd will tend to be distributed evenly in space and leave free gaps, which the robot can pass through. In contrast, if the crowd was governed by a velocity constraint-based approach, such as ORCA [20], agents would not repell each other, and thus, higher local crowd density could occur and hinder the robot's motion. The combination SF-AO is particularly conservative, as it combines the territorial effect of repulsion with anticipatory collision avoidance.

Limitations

The presented results obtained in simulation for AO and SF-AO are based on a model of a heterogeneous system, consisting of a homogeneous crowd and a robot. The crowd's behavior is described by the Social Force Model (SFM), whereas the robot's behavior is described by controllers based on AO. While this combination leads to observing counter-intuitive effects, such as the fact that the robot can be more efficient outside of dynamic lanes, the heterogeneous model is not based on empirical observations. Thus, it does not readily describe a real robot's interactions with a crowd, whereas it can point out the theoretical possibility for the simulated effects. For example, a robot which is more agile than pedestrians and accepted in close proximity could indeed exploit the space between lanes to progress faster.

Furthermore, it has to be noted that the SFM does not represent realistic behavior of individual pedestrians, but rather allows to reproduce realistic crowd dynamics on a more macroscopic level. As a consequence, evaluating the robot's performance in a simulated crowd governed by the SFM corresponds to

evaluating its ability to cope with obstacles whose macroscopic distribution in terms of positions, densities, and velocities can be representative of real crowds, whereas individual interactions cannot be expected to be realistic. Particularly, the presented simulations are not informative of systematic effects arising from the robot's interaction with pedestrians. For example, the robot's navigation task could be easier in real crowds than in simulation if, hypothetically, all pedestrians were to avoid it proactively upon noticing its apparent lack of agility.

The presented simulations focus specifically on the practically important case of bi-directional flow, and accordingly, the environment has been chosen as a corridor, and the metrics have been chosen to characterize agents' motion and interaction under these conditions, as proposed in prior work [46, 98]. As pointed out in the previous section, this type of flow and environment simplify collision avoidance for the robot due to emerged lanes. Thus, the presented results do not allow to assess the proposed AO-based method's performance in more general conditions.

Since the simulated conditions render the robot's task of navigation rather easy, the small time horizon of τ=1 s for the AO-based method has been sufficient for efficient navigation and for avoiding most collisions. However, the simulations' results presented in this chapter do not allow any conclusions about the AO-based method's performance with larger time horizons. Furthermore, the simulations have not investigated how the utilized scheme for determining an approximate halfplane for a given AO affects the AO-based method's performance. The Chapter 6 investigates navigation based on AO with a larger time horizon and with alternative schemes to choose approximating halfplanes.

4.5 Conclusion

In this chapter, we have studied Acceleration Obstacles (AO) in the context of robotic navigation in human crowds, focusing particularly on bi-directional flow in corridors. We have described the AO geometrically and analyzed how its shape depends on parameters. We have related the Acceleration Velocity Obstacle (AVO) to the AO via a limit case, which requires the least braking deceleration. A method has been proposed for a robot's navigation using AO via a novel algorithm which exploits our geometric analysis of AO to compute conservative linear approximations of AO in closed form. For the experimental evaluation, we have used the Social Force Model (SFM) to simulate crowds which interact with a robot. Applying our method, the robot progresses faster and undergoes very few collisions in comparison to pedestrians. Since many near misses still indicate a risk of collision, future work could address this issue e.g. by adaptive distance margins or objective functions that include social norms. The combination of the SFM and the AO has been evaluated as an alternative robotic controller, which was found to be significantly less efficient. Furthermore, we have shown that the robot could achieve higher performance when being less often in lanes.

Chapter 5

Cooperative Navigation in Crowds by Inverse Reinforcement Learning

Abstract In this chapter, the problem for a mobile robot to navigate seamlessly in a human crowd is treated by an inverse reinforcement learning (IRL) approach. A novel feature is proposed to model costs of anticipated collisions between agents. The feature approximates agents' pairwise interaction energy, a function which prior work has derived empirically from crowd data as an interaction potential driving pedestrians' mutual avoidance. Using a recent framework to perform IRL from locally optimal examples in continuous space, cost functions which incorporate the novel feature are learned efficiently from high-dimensional examples of real crowd motion. Examples are obtained from two public datasets, which have been recorded on a university campus and in a shopping mall, respectively, and which contain pedestrians' and wheelchair users' trajectories.

The learned models are evaluated and compared in how accurately their local optima model the training examples and test examples. Furthermore, predictions based on test examples' initial states only are generated similarly by optimization, and their distance to recorded ground truth is measured. Both models' predictions compare favorably to a recent related approach from the literature.

Finally, a control system which computes and executes in real-time an optimal trajectory according to the learned cost functions is implemented on a robotic wheelchair, to steer it between pedestrians perceived by an on-board tracking system. The robot is deployed on campus, where the controller's performance is evaluated qualitatively. Results show that the approach often generates apt motion plans, which complement pedestrians' motion in an efficient manner, albeit oscillations between locally optimal solutions may occur.

5.1 Introduction

The geometric approaches presented in the two previous chapters allow a robot to avoid collisions in an anticipatory fashion by considering agents' motion in the near future, as defined by a time horizon τ. In this chapter, the same idea is employed to model and execute cooperative collision avoidance with anticipation. However, instead of using constraints to define collision-prone velocities, the approach in this chapter associates a measure of risk with relative velocities, which becomes higher, the sooner a collision would happen. This risk or interaction energy [57] is incorporated in a cost function. While the time horizon τ needs to be adapted to the density of a crowd for best performance of the previous approaches, the cost function in the approach here can naturally account for different densities.

For navigating smoothly and safely in human crowds, robots need to take into account how their own actions affect surrounding humans. Furthermore, cooperative navigation requires to reason about each involved agent's individual costs and benefits. We propose to consider such factors in a collective cost function, which describes how desirable any particular combination of individual trajectories is, from the collective point of view of all involved agents. However, such a cost function can provide useful guidance to a robot navigating in a populated environment only if it captures actual characteristics of human behavior. Thus, a principled approach to design a suitable cost function and to calibrate or learn its parameters from empirical data is instrumental.

The problem to recover a cost function underlying observed behavior is at the heart of inverse optimal control (IOC) and inverse reinforcement learning (IRL). Traditional methods in this field assume that observations are globally optimal, which makes their computational cost scale exponentially with the number of state dimensions. Thus, they are not well suited to a system with many agents and respective state dimensions. Furthermore, human crowds may evolve in multiple qualitatively different ways with similar probabilities. For example, a pedestrian may pass another pedestrian on the left or on the right side, whereas trajectories in between the two alternatives are of high costs, as they lead to a collision. Thus, the two alternatives can be described as local optima, since they are separated by trajectories with higher costs. Consequentially, it is more reasonable to consider navigation in multi-agent systems as a local optimization problem and to assume that observed behavior is only locally optimal.

Fortunately, more recent approaches to IOC/IRL have been developed which drop the assumption of global optimality and, in doing so, free themselves from the curse of dimensionality by restricting their attention to parts of the state space in the neighborhood of observed examples. We employ the framework [79] to perform IOC from examples of up to 13 simultaneously navigating agents, which corresponds to 52 state dimensions in our formulation. To the best of our knowledge, approaches to IOC assuming local optimality have not been applied so far to navigation problems involving more than three agents.

Despite promising algorithms being available, designing appropriate features for the problem of multi-agent navigation IOC/IRL remains a challenging and

critical step for achieving a cost function that generalizes in a meaningful way and proves useful in real-world situations. To extend the set of available features for IOC/IRL of cooperative navigation and collision avoidance, we propose a novel feature which approximates an interaction energy [57] between agents. This interaction energy has been derived empirically from crowd data as an interaction potential driving pedestrians' mutual avoidance [57]. As it constitutes an important development for the state of the art in modeling and simulating human crowds, it seems worthwhile to incorporate a corresponding feature in an IRL approach to learning navigation in crowds.

We obtain the training samples for our approach from the dataset DIAMOR [54], containing bi-directional motions in a corridor. We focus again on this scenario, as in the previous chapter, as it constitutes an important motion base case [56] for urban environments, and as it allows to simplify modeling of agents' preferred velocities, which can be assumed to be aligned with the corridor's axis. We evaluate our approach on the dataset ETH [103], which is a common benchmark for data-driven crowd modeling and prediction methods (e.g. [62, 63, 104]).

The chapter's main contributions are:

- The use of a novel feature quantifying agents' interaction energy [57] to formulate cost functions for IRL of navigation in pedestrian crowds.
- An approximation of the original discontinuous interaction energy [57] by a smooth formula, which provides Hessian matrices and gradients for local IRL [79] and local optimization.
- A quantitative evaluation of the approach [79] for IRL of navigation in crowds, demonstrating more accurate and smoother predictions of pedestrians' trajectories in comparison to a state-of-the-art alternative [35].
- A qualitative evaluation of our IRL-based approach's performance at controlling a mobile robot navigating in a real crowd.

5.2 Related Work

5.2.1 Models of Optimal Crowd Behavior

Many works [33–38,43,47,57,95,105,106] in the context of crowd modeling/simulation and robotic motion planning describe interacting agents' navigation behavior as a result of inter-dependent optimization processes.

Approaches using (inverse) reinforcement learning often consider a crowd of agents as a single system aiming at maximizing a scalar reward function [33–35, 43, 106]. Similarly, optimization-based techniques for crowd simulation [57, 105] consider a single cost function which is jointly minimized over all agents' actions to determine their behavior. Formulating a joint optimization problem in such a way is appealing since it allows to model agents' reciprocal actions, i.e. actions that complement each other.

To model interactions that reflect competing interests and imperfect information, some works [36–38, 47] adopt a game theoretic perspective instead. However, as has been argued in [47], for the important case of two agents approaching each other, both their interests are aligned, such that a game theoretic perspective coincides with single-objective models. Bearing this in mind, and since IOC/IRL techniques are less developed for game theoretic models than for single objectives, this chapter assumes a single objective.

5.2.2 Frameworks for IRL of Navigation

For learning navigation in crowds, several previous works have employed IRL. Some works [66,107–110] use discrete state and action spaces. For learning a reward, these approaches operate on grid representations of the entire state space, and thus, they cannot easily incorporate multiple agents due to exponentially increasing computational cost. Instead, they typically consider pedestrians as exogeneous inputs which affect the cost/reward features only. While such an approach may yield sensible behavior, it cannot produce a model which reasons about interactions between agents, since other agents' behavior is treated as a given input to the model.

In contrast, the works [35, 79, 95, 106] have adapted the popular method of Maximum Entropy IRL [78] to continuous state and action spaces. In the works on continuous spaces, trajectories are either parametrized by the control actions at each discrete point in time [111] or as splines [35, 95, 106]. For dealing with the high dimensionality of the space of possible multi-agent trajectories, [111] resort to a local approximation [79] of the exponential policy for Maximum Entropy IRL, whereas [35, 95, 106] simplify the policy by discretizing it into toplogical variants (who is passing on which side of who) and/or use Monte Carlo techniques for performing IRL.

5.2.3 Cost/Reward Features for IRL of Navigation

Different cost/reward structures have been employed in prior work. In [66, 110, 112], a robot-centric approach is adopted, where the space around the robot is partitioned in areas, with corresponding features being activated e.g. by the presence of pedestrians in these areas. This structure is tailored to the single-agent representation treating pedestrians as exogeneous inputs.

For multi-agent state representations, other works [35, 95, 106, 111] define features that depend on each agent or on each pair of agents, similarly as in our approach. Common features to account for collisions include $f = d^{-2}$ [35] and $f = \exp(-d^2/\sigma^2/2)$ [111], where d denotes the distance between agents' centers. Both are only position-dependent, and thus, they do not encapsulate any anticipatory behavior for avoiding collisions. They can still generate such behavior, but only due to optimizing a trajectory's cost over an extended time horizon. In summary, we believe that these and similar position-based features do not reflect empirically known characteristics of human walkers, such as early adaptation and regulation of the minimum predicted distance.

Finally, some works [110,112] use deep neural networks as cost/reward models which are more flexible than traditional linear combinations. However, such an approach yields a cost/reward function which is more difficult to interpret, since the learned weights are not directly linked to single features anymore.

5.3 Method

5.3.1 System Model

For a system comprising n agents, let $\mathbf{p}_i, \mathbf{v}_i, \mathbf{a}_i \in \mathbb{R}^2$ denote the i-th agent's position, velocity, and acceleration, respectively, where $i \in \{1, 2, \dots n\}$. Let the system's *state*

$$\mathbf{x} \coloneqq \begin{bmatrix} \mathbf{p}_1^{\mathrm{T}} & \mathbf{p}_2^{\mathrm{T}} & \dots & \mathbf{p}_n^{\mathrm{T}} & \mathbf{v}_1^{\mathrm{T}} & \mathbf{v}_2^{\mathrm{T}} & \dots & \mathbf{v}_n^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{5.1}$$

contain all positions and velocities. The system's *action*

$$\mathbf{u} \coloneqq \begin{bmatrix} \mathbf{a}_1^{\mathrm{T}} & \mathbf{a}_2^{\mathrm{T}} & \dots & \mathbf{a}_n^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{5.2}$$

is defined by all accelerations, on the other hand.

The system's transition from one state $\mathbf{x}^{(k)}$ to another state $\mathbf{x}^{(k+1)}$, under an action $\mathbf{u}^{(k+1)}$ over a time step of duration h, is described by the linear dynamic system

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{A}\mathbf{x}^{(k)} + \mathbf{B}\mathbf{u}^{(k+1)}, \\ \mathbf{A} &\coloneqq \begin{bmatrix} \mathbf{I} & h\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{B} \coloneqq \begin{bmatrix} (h^2/2)\mathbf{I} \\ h\mathbf{I} \end{bmatrix}, \end{aligned} \tag{5.3}$$

where $\mathbf{I} \in \mathbb{R}^{2n \times 2n}$ denotes the identity matrix. The above dynamic model (5.3) corresponds to applying constant accelerations over a duration h.

A *state–action trajectory* $\mathcal{S} \coloneqq (\mathcal{X}, \mathcal{U}, \mathcal{T})$ is defined by a state sequence $\mathcal{X} \coloneqq \left\{\mathbf{x}^{(k)}\right\}_{k=0}^{K}$ and an action sequence $\mathcal{U} \coloneqq \left\{\mathbf{u}^{(k)}\right\}_{k=1}^{K}$ which satisfy the dynamic model (5.3) for $0 \leq k < K$ with a uniform time step h, and by a time sequence $\mathcal{T} \coloneqq \left\{t^{(k)}\right\}_{k=0}^{K}$, where each $t^{(k)}$ denotes the time at which the state $\mathbf{x}^{(k)}$ is attained, and $t^{(k+1)} - t^{(k)} = h, \forall k$.

5.3.2 Forward Optimal Control Problem

We assume that there is a scalar cost function $J(\mathcal{X}, \mathcal{U})$ which reflects the considered multi-agent system's collective objectives, e.g. navigating efficiently and without any collision. Given such a cost function J, a sequence $\mathcal{T}$ of $K+1$ time instants with time step h, and an initial state $\mathbf{x}_o$, we define a finite horizon optimal control problem

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{U}} \ & J(\mathcal{X}, \mathcal{U}) \\ \text{s.t. } & \mathbf{x}^{(k)} = \mathbf{A}\mathbf{x}^{(k-1)} + \mathbf{B}\mathbf{u}^{(k)}, \ k = 1, \dots K \\ & \mathbf{x}^{(0)} = \mathbf{x}_o. \end{aligned} \tag{5.4}$$

Then, any locally optimal solution $\mathcal{X}^*$, $\mathcal{U}^*$ of (5.4) is considered as a suitable choice of behavior for the system, and the corresponding locally optimal state–action trajectory is defined as $\mathcal{S}^* := (\mathcal{X}^*, \mathcal{U}^*, \mathcal{T})$.

5.3.3 Framework for Inverse Reinforcement Learning

Let $\mathcal{E} := \{\mathcal{S}_l\}_{l=1}^{L}$ denote a given set of L state–action trajectories recorded from a crowd of agents. We assume that the behavior recorded in the examples $\mathcal{E}$ is locally optimal with respect to some unknown cost function J. Inverse reinforcement learning (IRL) aims at identifying a meaningful cost function such that observed behavior can be understood as attempts to minimize the corresponding cost. We consider cost functions of the form

$$J(\mathcal{X}, \mathcal{U}; \mathbf{w}) := \sum_{k=1}^{K} \mathbf{w}^{\mathrm{T}} \mathbf{f}\left(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}\right), \tag{5.5}$$

where the *features* $\mathbf{f}(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^q$ are specified a priori to describe relevant properties for the system dependent on its current state and action, and where $\mathbf{w} \in \mathbb{R}^q$ is the vector of the features' unknown weights to be determined by IRL. We use a linear technique as we hypothesize that our features describe non-interacting cost components.

In the original maximum entropy (MaxEnt) framework [78], the weights are learned by maximizing the examples' likelihood under the policy

$$p\left(\mathcal{U} | \mathbf{x}_o, \mathbf{w}\right) = \frac{\exp\left(-J\left(\mathcal{X}\left(\mathcal{U}, \mathbf{x}_o\right), \mathcal{U}; \mathbf{w}\right)\right)}{Z\left(\mathbf{x}_0, \mathbf{w}\right)}, \tag{5.6}$$

which assigns a probability density to an action sequence in proportion to the exponential of its negative cost (i.e. reward), where Z is the partition function, which normalizes the distribution. Note that $\mathcal{X}\left(\mathcal{U}, \mathbf{x}_o\right)$ is the state sequence that results when starting from $\mathbf{x}_o$ and applying the action sequence $\mathcal{U}$. Computing Z requires integration over the space of $\mathcal{U}$, which is intractable for more than a few action dimensions and time steps per example. Therefore, we adopt the framework [79], where the policy (5.6) is approximated as a Gaussian in $\mathcal{U}$ around the examples, which relaxes the original MaxEnt framework's assumption of their global optimality to local optimality only.

5.3.4 Cost Features for Cooperative Navigation

This section defines the features, i.e. functions that we use as components of $\mathbf{f}$. Since the present work models multiple agents as a single system optimizing a collective reward, its features are defined to receive contributions from all agents or all pairs of agents. Thus, individual and pairwise feature contributions are defined in the following two respective sections. For simplicity, we employ only four features, which we consider as the most essential ones for multi-agent navigation, describing respectively individual effort and goal-directed motion, and pairwise distance and anticipatory collision avoidance.

Individual feature contributions

An agent's effort is described by the feature contribution

$$f_i^{(\mathrm{a2})} = |\mathbf{a}_i|^2/2 \tag{5.7}$$

where $\mathbf{a}_i$ denotes the agent's acceleration. As an alternative to measuring effort by squared acceleration, we define

$$f_i^{(\mathrm{a1})} = |\mathbf{a}_i| + \left(\log(1+\exp(-2\lambda|\mathbf{a}_i|)) - \log(2)\right)/\lambda \tag{5.8}$$

as a feature contribution smoothly approximating $|\mathbf{a}_i|$, where $\lambda > 0$ controls the function's sharpness at the origin.

An agent's deviation from its desired motion is described by the feature contribution

$$f_i^{(\mathrm{v})} = |\mathbf{v}_i - \check{\mathbf{v}}_i|^2/2 \tag{5.9}$$

where $\mathbf{v}_i$ and $\check{\mathbf{v}}_i$ denote the agent's actual and desired velocity, respectively.

Pairwise feature contributions

We employ a simple pairwise feature contribution commonly found in related work, namely a gaussian function of both agents' distance [111]

$$f_{ij}^{(\mathrm{c})} = \exp\left(-|\mathbf{p}_{ij}|^2/(2\sigma^2)\right), \tag{5.10}$$

where $\sigma > 0$ controls the activation distance, and $\mathbf{p}_{ij} \coloneqq \mathbf{p}_i - \mathbf{p}_j$ denotes the pair's relative position.

In [57], two agents' *interaction energy* has been defined as $E \coloneqq \eta\tau^{-2}\exp(\tau/\tau_o)$, where τ denotes the time to collision, τ_o denotes a time horizon, and η is an arbitraty constant setting the units of energy. The Fig. 5.1 plots E as a function of the relative velocity $\mathbf{v}$ for a pair of agents with a given relative position $\mathbf{p}$, where we set their radius' sum R=0.4 m, τ_o=3 s, and η=1 s^2 to obtain a dimensionless energy. E has been derived empirically from crowd data as an interaction potential which drives pedestrians' mutual avoidance [57]. The technique for crowd simulation proposed in [105] updates agents' velocities at each time step by minimizing a continuous approximation of E. Since τ is defined to take a finite value when agents are on a colliding course, and to be $+\infty$ otherwise, E rises discontinuously from zero to a finite positive value as the two agents enter a colliding course (cf. Fig. 5.1).

In order to use E as a feature for IRL in the continuous framework [79] adopted in the present work, an approximation of E is required, which is not only continuous but also twice differentiable. Thus, we approximate E as

$$\tilde{E} \coloneqq \eta\tilde{\tau}^{-2}g \tag{5.11}$$

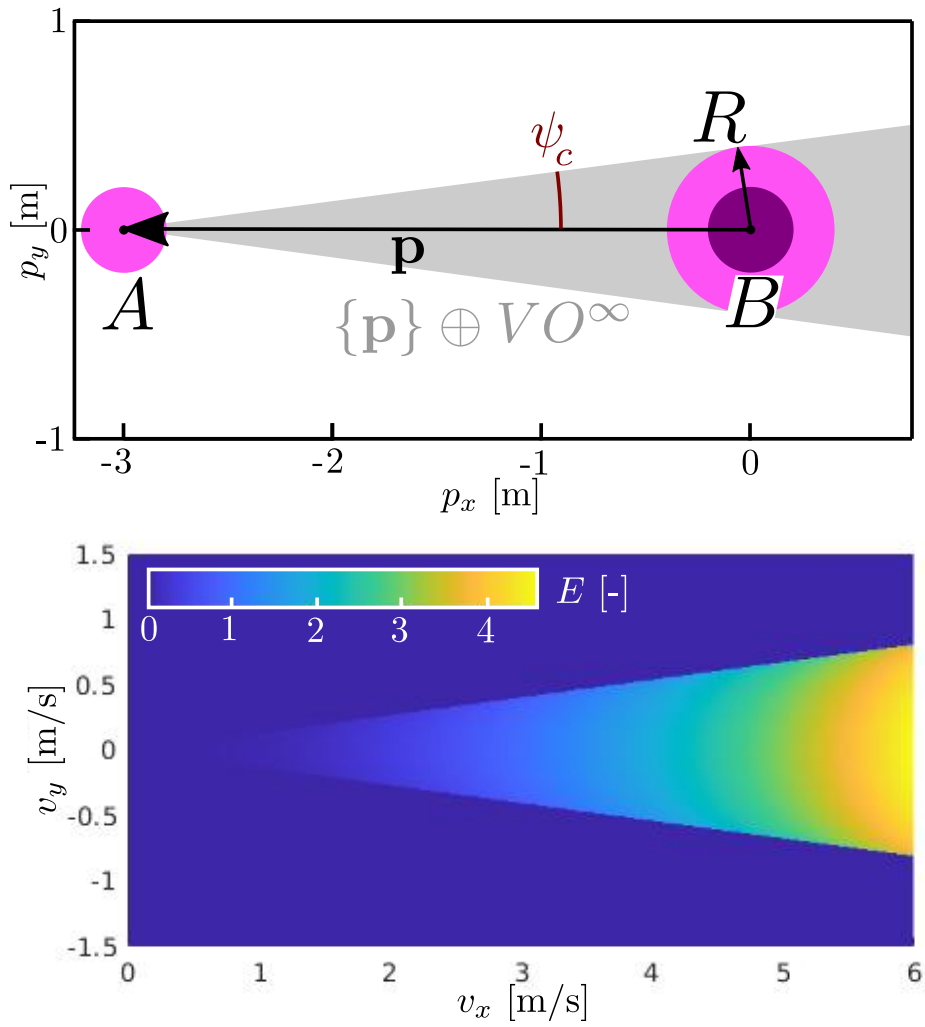

Figure 5.1: Top: two agents A and B are shown at exemplary positions. The cone of colliding relative velocities VO^{∞} is constructed dependent on their relative position and their combined radius R. Bottom: for the two agents' given relative position, their interaction energy E is shown as a function of their relative velocity $\mathbf{v}$. E is uniformly zero at velocities outside VO^{∞} and rises discontinously at its edge.

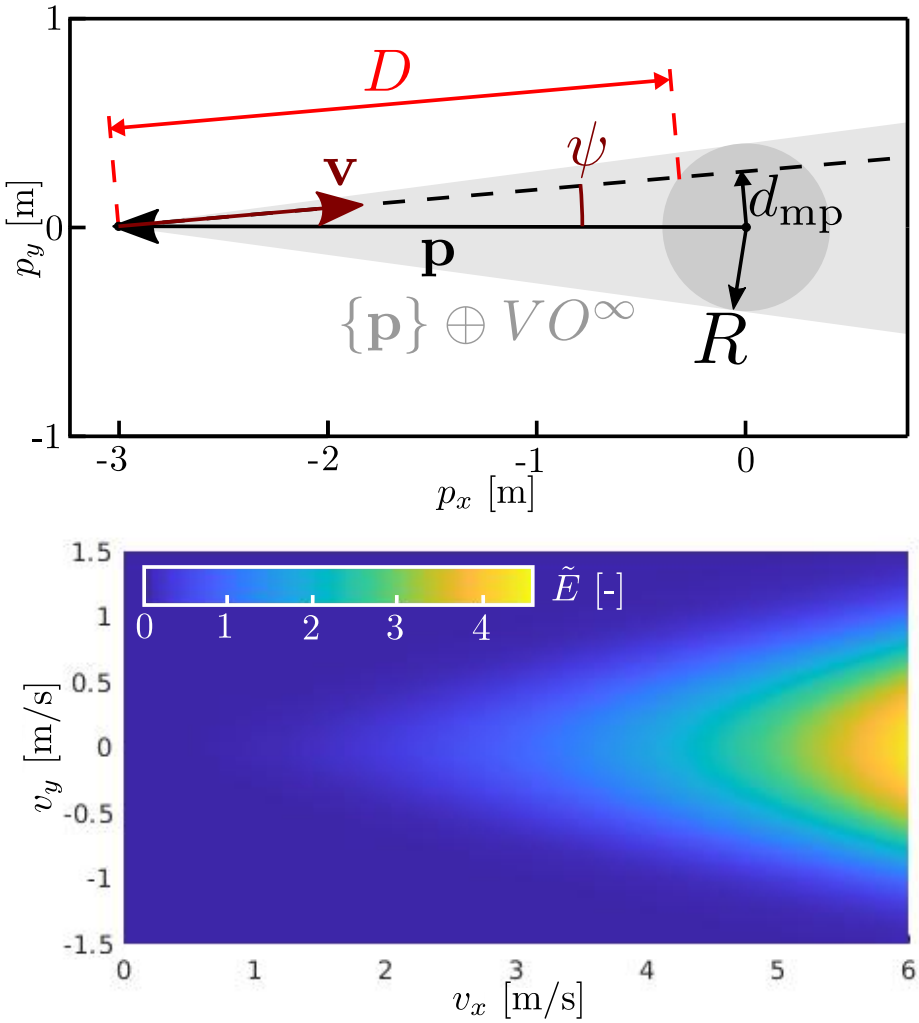

Figure 5.2: Top: for the two agents from the Fig. 5.1, the minimum predicted distance d_{mp} and the free relative path length D for an exemplary relative velocity $\mathbf{v}$ are shown. Bottom: the agents' approximate interaction energy $\tilde{E}$ is shown as a function of their relative velocity. At the edge of the cone of colliding relative velocities VO^{∞}, $\tilde{E}$ is smooth, unlike E (cf. Fig. 5.1).

where g is an activation function which rises from almost zero to almost one as the relative velocity enters the cone of colliding velocities VO^{∞}. We define g as

$$g(z) := (1 + \exp(-sz))^{-1}$$
$$z := -\mathbf{p}^{\mathrm{T}}\mathbf{v} - \frac{|\mathbf{p}|^2|\mathbf{v}|}{\sqrt{|\mathbf{p}|^2 + R^2}}$$

where $s > 0$ controls the sigmoid activation's steepness. Note that the activation function's argument z can be written as $z = |\mathbf{p}|\,|\mathbf{v}|\,(\cos(\psi) - \cos(\psi_c))$, with $\psi := \angle(-\mathbf{p}, \mathbf{v})$ and ψ_c denoting half the opening angle of the cone VO^{∞}, respectively. Thus, it holds $z = 0$ if the relative velocity is on the edge of VO^{∞}, which becomes the activation threshold.

For relative velocities in VO^{∞}, $\tilde{\tau} := \tilde{D}/|\mathbf{v}|$ approximates the time to collision, where $\tilde{D}$ approximates the distance D which the relative position $\mathbf{p}$ can travel along the relative velocity $\mathbf{v}$ before entering the disk of radius R centered at the origin (cf. Fig. 5.2). We define $\tilde{D}$ as

$$\tilde{D}^2(\mathbf{p}, \mathbf{v}) := \varepsilon_1 + (|\mathbf{p}| - R)^2 + 2\,(|\mathbf{p}|/R - 1)\, d^2_{\mathrm{mp}}(\mathbf{p}, \mathbf{v})$$
$$d^2_{\mathrm{mp}}(\mathbf{p}, \mathbf{v}) := |\mathbf{p}|^2 - \frac{\left(\mathbf{p}^{\mathrm{T}}\mathbf{v}\right)^2}{|\mathbf{v}|^2 + \varepsilon_2}$$
$$\varepsilon_1 := 0.22R^2$$

where $\varepsilon_2 > 0$ prevents division by zero. The function d^2_{mp} denotes the squared minimum predicted distance (cf. Fig. 5.2), and the constant ε_1 ensures that $\tilde{D}^2$ remains positive even when $|\mathbf{p}| < R$, i.e. during a collision. The Fig. 5.2 shows the resulting approximation for the same configuration as for the Fig. 5.1, where we set s=10 and ε_2=0.01 $\mathrm{m}^2/\mathrm{s}^2$.

Accordingly, we define the pairwise feature contribution

$$f^{(\mathrm{e})}_{ij} := \tilde{E}\,(\mathbf{p}_{ij}, \mathbf{v}_{ij}) \tag{5.12}$$

where $\mathbf{v}_{ij} := \mathbf{v}_i - \mathbf{v}_j$ denotes the pair's relative velocity.

Feature vectors

We consider two alternative vectors

$$\mathbf{f}_{(L^2)} := \begin{bmatrix} f^{(\mathrm{a2})} & f^{(\mathrm{v})} & f^{(\mathrm{c})} & f^{(\mathrm{e})} \end{bmatrix}^{\mathrm{T}}, \tag{5.13}$$
$$\mathbf{f}_{(L^1)} := \begin{bmatrix} f^{(\mathrm{a1})} & f^{(\mathrm{v})} & f^{(\mathrm{c})} & f^{(\mathrm{e})} \end{bmatrix}^{\mathrm{T}}, \tag{5.14}$$

which differ only in their first component, which is proportional to either the L^2- or the L^1-norm of the system's accelerations, respectively. Any feature f collecting individual contributions (with superscripts a2, a1, v) is defined as $f := (1/n)\sum_{i=1}^{n} f_i$. And any feature f collecting pairwise contributions (with superscripts c, e) is defined as $f := (1/n)\sum_{i=1}^{n}\sum_{j=i+1}^{n} f_{ij}$.

5.3.5 Dimensionless Features and Weights

In order to obtain dimensionless features and weights, and to facilitate assessing their relative importance, we define their normalized counter parts as follows. Let $\tilde{f}(\mathcal{E}) := P_{80}^{\mathcal{E}}\{f\}$ define the *normalizer* for any feature f as its 80-th percentile over a sample set $\mathcal{E}$. Then, for any feature f and respective weight w, we define the corresponding *normalized feature* as $\phi := f/\tilde{f}$ and the corresponding *normalized weight* as $\theta := w\tilde{f}$, such that the corresponding term of the cost function can be written equally as $wf = \theta\phi$.

5.4 Learning experiments

Using the framework [79] for IRL in conjunction with the multi-agent model and the features described in the previous section, a corresponding cost function is learned from examples of real multi-agent trajectories from the public dataset DIAMOR [54]. This dataset has been recorded in a shopping mall and contains mostly pedestrians' but also wheelchair users' data. The resulting model is evaluated on a separate test set of samples from DIAMOR and, additionally, on the public dataset ETH [103], which has been recorded on a university campus and contains pedestrians' data only. In the remainder of this section, our approach is termed NavIOC. Our code is available at `https://github.com/epfl-lasa/navioc`.

5.4.1 Data Pre-processing

Fitting individual trajectories

An agent's trajectory data as given originally in a dataset is denoted as $\left\{\bar{t}^{(i)}, \bar{\mathbf{p}}^{(i)}\right\}_{i=1}^{M}$, consisting of M sequential time values $\bar{t}^{(i)}$ and corresponding positions $\bar{\mathbf{p}}^{(i)}$. A state–action trajectory $\mathcal{S}$ describing the single agent at hand (n=1) according to Sect. 5.3.1 is fit to the data by solving the following quadratic program. We jointly optimize $\mathcal{X}$, $\mathcal{U}$ and a sequence of interpolated positions $\hat{\mathcal{P}} := \left\{\hat{\mathbf{p}}^{(i)}\right\}_{i=1}^{M}$ in the problem

$$\begin{aligned} \min_{\mathcal{X},\mathcal{U},\hat{\mathcal{P}}} \quad & \sum_{i=1}^{M} \left|\hat{\mathbf{p}}^{(i)} - \bar{\mathbf{p}}^{(i)}\right|^2 + \alpha \sum_{k=1}^{K} \left|\mathbf{u}^{(k)}\right|^2 \\ \text{s.t. } & \mathbf{x}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)} + \mathbf{B}\mathbf{u}^{(k+1)},\ k = 0, \dots K-1 \\ & \hat{\mathbf{p}}^{(i)} = \begin{bmatrix}\mathbf{I} & h_i\mathbf{I}\end{bmatrix} \mathbf{x}^{(\kappa(i))} + \frac{h_i^2}{2}\mathbf{u}^{(\kappa(i)+1)},\ i = 1, \dots M, \end{aligned} \tag{5.15}$$

where $\alpha > 0$ is a regularizer controlling smoothness, $\kappa(i) := \text{floor}((\bar{t}^{(i)} - t^{(0)})/h)$ indicates for each $\bar{t}^{(i)}$ the temporally closest foregoing state/action extrapolating $\hat{\mathbf{p}}^{(i)}$, and $h_i := \bar{t}^{(i)} - t^{(0)} - h\kappa(i)$ denotes each extrapolation's time step.

To keep the dimensionality of the optimization problem (5.15) at an efficiently tractable level, for any trajectory longer than 100 h, multiple temporally

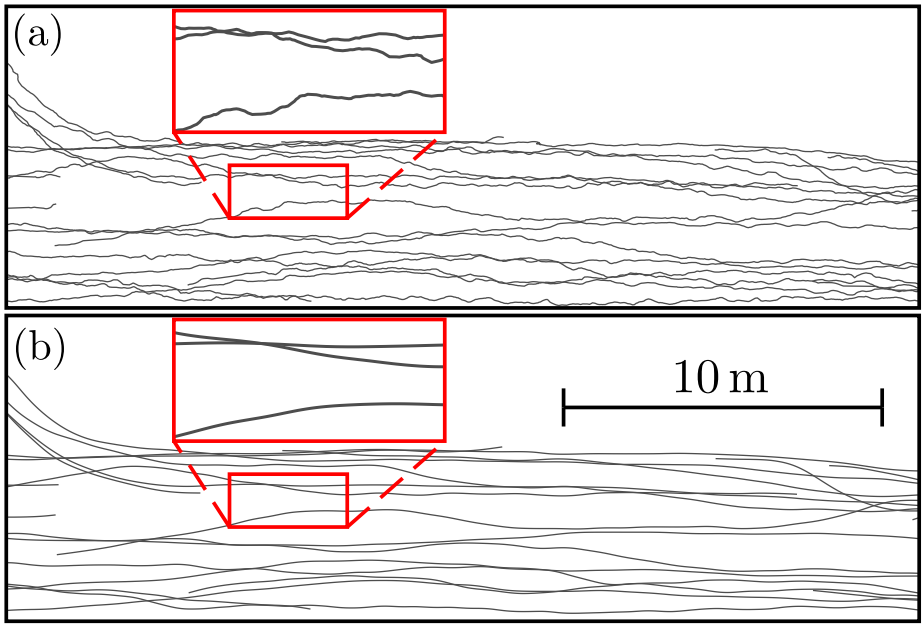

Figure 5.3: Showing (a) original tracks from the dataset DIAMOR and (b) the corresponding fit trajectories. The exemplary area delineated in red is magnified for better visibility only.

overlapping state–action trajectories of 100 time steps each are fit sequentially to sub-sequences of the original trajectory's data, with additional constraints enforcing continuity at stitching points. The Fig. 5.3 shows original tracks from DIAMOR, and the corresponding fit trajectories for $h = 0.05\,\mathrm{s}$ and $\alpha = 0.01\mathrm{s}^4$.

Estimating desired velocities

To estimate time-varying desired orientations $\breve{\varphi}^{(k)}$, for any agent from DIAMOR, $\breve{\varphi}^{(k)}$ at any instant k is estimated to be left/right for a negative/positive actual velocity $v_x^{(k)}$, where we restrict our attention to agents in the corridor aligned with the x-axis (cf. Fig 5.3). For any agent from ETH, firstly, the agent's goal is identified among the dataset's four designated goals by choosing the one which the agent is facing mostly during the later half of its trajectory. Secondly, $\breve{\varphi}^{(k)}$ is set in the direction from the agent to the goal at each instant k.

Assuming that any particular agent's desired speed $\breve{v}$ is constant over time, its value is estimated as the mode of the agent's speed histogram, disregarding speeds below a threshold $v_{\min}$=0.3 m/s. Finally, for each agent, the desired velocity is defined as $\breve{\mathbf{v}}^{(k)} := \breve{v}[\cos\breve{\varphi}^{(k)} \sin\breve{\varphi}^{(k)}]^{\mathrm{T}}$, if $|\mathbf{v}^{(k)}| > v_{\min}$, or as zero, else. Thus, for each agent, a sequence $\breve{\mathcal{V}} := \{\breve{\mathbf{v}}^{(k)}\}_{k=0}^{K}$ of desired velocities is obtained.

Sampling multi-agent trajectories

For each dataset, the union of individually fit trajectories' time domains is divided into adjacent intervals of uniform duration T=4.8 s, by convention [35]. For each such interval, all individual trajectories' sub-sequences which are defined on the entire interval are aggregated into a multi-agent state-action trajectory. For DIAMOR, sub-sequences that do not overlap with the area of interest in the main corridor (shown partially in the Fig. 5.3) are disregarded, whereas for ETH, all agents are considered.

Defining training and test sets

We inspect each multi-agent sample obtained from DIAMOR and discard those containing agents whose desired motion does not seem aligned with the x-direction, e.g. agents who traverse the corridor along the y-direction or diagonally due to intrinsic motives and not to perform avoidance maneuvers. To group the remaining samples from DIAMOR in a training set $\mathcal{E}_1$ and a test set $\mathcal{E}_2$, they are sorted in their temporal order and then assigned in an alternating fashion to $\mathcal{E}_1$ or $\mathcal{E}_2$, to ensure that the two sets reflect similar crowd conditions. In contrast, all multi-agent samples obtained from ETH are grouped in the set $\mathcal{E}_3$, without performing any manual selection. The Fig. 5.4 shows histograms of the number of agents n in a given sample over the three datasets. Their respective sizes are $|\mathcal{E}_1|$=22, $|\mathcal{E}_2|$=21, and $|\mathcal{E}_3|$=99.

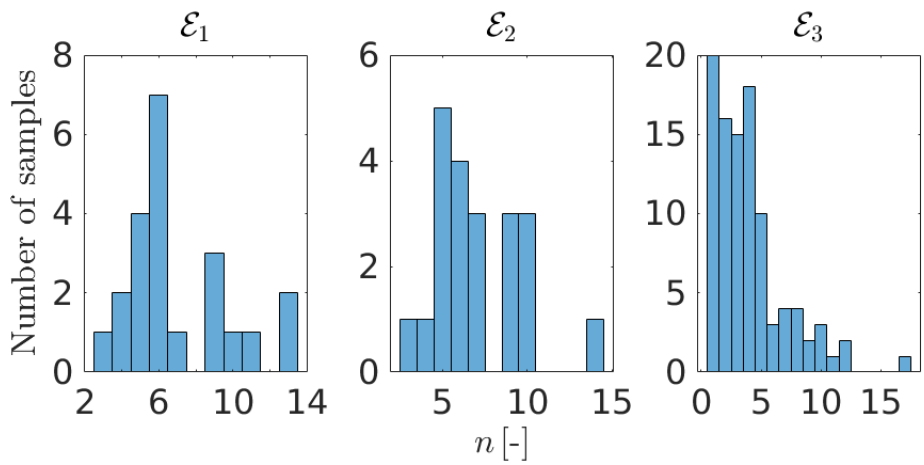

Figure 5.4: For the three datasets $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$, histograms of the number of agents n in a given sample are shown.

Table 5.1: Features' normalizers $\tilde{f}$ and learned normalized weights θ

f	$10\tilde{f}$	$\theta_{(L^2)}$	$\theta_{(L^1)}$	Parameters
$f^{(a2)}$	$0.185\,\mathrm{m}^2/\mathrm{s}^4$	1.0000	n.a.	–
$f^{(a1)}$	$1.039\,\mathrm{m}/\mathrm{s}^2$	n.a.	1.0000	λ=10 s^2/m
$f^{(v)}$	$0.131\,\mathrm{m}^2/\mathrm{s}^2$	0.0385	7.5471	–
$f^{(c)}$	0.635	0.0007	0.0976	σ=0.5 m
$f^{(e)}$	0.021	0.0137	0.6593	$\begin{cases}\eta=1\,\mathrm{s}^2,\ s=25\mathrm{s}^2/\mathrm{m}^2\\ R=0.4\,\mathrm{m},\ \varepsilon_2=0.01\,\mathrm{m}^2/\mathrm{s}^2\end{cases}$

5.4.2 Training

We have adapted the publicly available software package[1] by [79] to learn from examples of varying dimensionality (due to varying numbers of agents) and complemented the package with our system and feature definitions. Training on the dataset $\mathcal{E}_1$ takes around 4 and 7 minutes for the feature vectors $\mathbf{f}_{(L^2)}$ and $\mathbf{f}_{(L^1)}$, respectively. The resulting models are termed as NavIOC-L^2 and NavIOC-L^1, respectively.

The Table 5.1 reports, for any feature f, the normalizer $\tilde{f}(\mathcal{E}_1 \cup \mathcal{E}_2)$, which has been computed with respect to all the selected samples from DIAMOR, and the learned normalized weights $\theta_{(L^2)}$ and $\theta_{(L^1)}$, which have been obtained for the feature vectors $\mathbf{f}_{(L^2)}$ and $\mathbf{f}_{(L^1)}$, respectively. Note that each learned weight vector has been scaled such that the acceleration feature's normalized weight equals one.

[1] `https://graphics.stanford.edu/projects/cioc/`

Choice of hyper parameters

The weights θ are initialized as random noise with zero mean and standard deviation 0.01, except for the acceleration weight, which needs to be negative for convergence and thus is set to -1. We observe that for too small α, only a negligible weight is learned for the interaction feature $f^{(\mathrm{e})}$. The sharpness s for the feature $f_{(a^1)}$ strongly affects the learned behavior, as for both very small or large s, acceleration is penalized more strongly relative to the velocity error, such that agents slowly adopt their target velocity.

5.4.3 Evaluation

We evaluate the learned models' accuracy on the dataset DIAMOR, and their predictive capability on the dataset ETH, both in terms of the euclidean distance between actual trajectories from a dataset and the corresponding trajectories generated by optimizing the learned cost functions. Examples of actual and generated trajectories are shown in the Fig. 5.5.

To evaluate accuracy on a given example, local optimization is performed, initialized at the example's actual trajectory (ground truth), to compute a locally optimal solution to the problem (5.4). Then, the *modeling error* is computed as the euclidean distance between the obtained locally optimal trajectory's positions and the actual trajectory's corresponding positions (ground truth). The Fig. 5.6 plots the modeling error as a function of the time relative to the considered trajectory's initial time, both for the training and test sets $\mathcal{E}_1$ and $\mathcal{E}_2$ from DIAMOR. As a baseline, a model assuming constant velocity (termed as CV) is evaluated as well.

In contrast, for evaluating the models' predictive capability, we do not provide any informative initialization, and start optimizing from $\mathcal{U} \equiv \mathbf{0}$. Furthermore, predictions are not informed by the non-causal state fits but instead by a causal state estimate of each trajectory's state at its initial time, which is provided in the ETH dataset. The prediction error is calculated similarly as the modeling error by computing the euclidean distance between agents' actual and predicted positions. The Fig. 5.7 reports our models' prediction error for the ETH dataset [103] in comparison to the baseline model, which assumes constant velocity over the prediction horizon, and the results reported by [35] for their IRL approach.

Furthermore, the Table 5.2 compares our results on the dataset ETH to those reported in [63] for their deep learning approach. The reported metrics are the Average Displacement Error (ADE) and the Final Displacement Error (FDE). The ADE at a given time is defined as the prediction error's average over all values up to this time of prediction, whereas the FDE at a given time is defined simply as the prediction error at this time. Thus, the ADE can be computed by averaging a given curve in the Fig. 5.7 up to a specific time, whereas the FDE can be read directly as the curves' values at specific times.

However, it has to be noted that our models and their models [63] rely on different information, since ours rely on non-causal knowledge of pedestrian's

preferred velocities as well as a causal initial state estimate, whereas theirs rely on (causal) knowledge of positions measured during the last 4.8 s.

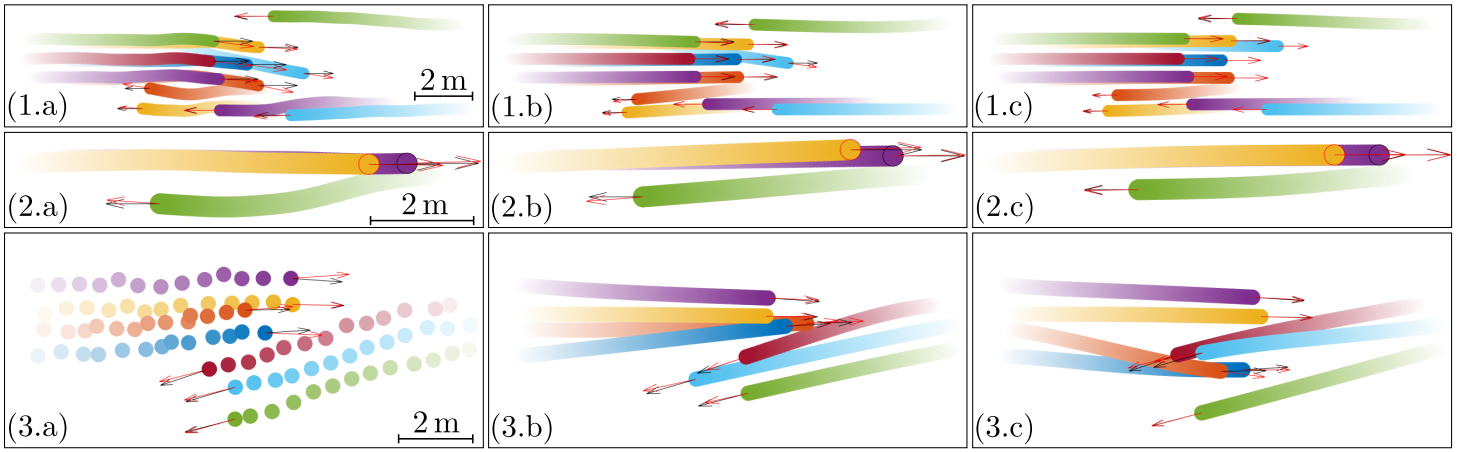

Figure 5.5: The plots show multi-agent trajectories from the example sets (left column) and their optimized counter parts generated from our two models NavIOC-L^2 (middle column) and NavIOC-L^1 (right column). Agents are depicted by their bounding circles of diameter R centered at their positions over time, such that circles for later instants are drawn on top of earlier circles and in more saturated colors. Red arrows indicate the most recent velocity, whereas black arrows indicate the assumed desired velocity for each agent. The examples in (1.a, 2.a, 3.a) belong to the sets $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$, respectively.

Table 5.2: Comparison with the deep learning approach SGAN [63] at prediction on the dataset ETH. The column SGAN collects the best values reported for different models by [63]. The best value per row is set in bold.

Metric	SGAN	NavIOC-L^2	NavIOC-L^1	CV
ADE at 3.2 s	0.60	**0.30**	0.35	0.34
ADE at 4.8 s	0.81	**0.47**	0.49	0.55
FDE at 3.2 s	1.19	**0.58**	0.60	0.68
FDE at 4.8 s	1.52	0.97	**0.87**	1.16

Table 5.3: Number of collisions

Dataset	Ground truth	NavIOC-L^2	NavIOC-L^1	CV
$\mathcal{E}_1$	0	0	0	5
$\mathcal{E}_2$	0	0	0	5
$\mathcal{E}_3$	2	4	0	38

For making a prediction, each agent's desired velocity at the instant when the prediction is issued is provided to our model. We provide this information to our model for comparison with related work [35], which similarly leverages knowledge of goals, even though, in practice, one can only estimate such information based on past observations.

As an additional metric, we count the number of collisions in the learned model's generated multi-agent trajectories, as reported in the Table 5.3 in comparison with the baseline assuming constant velocity and ground truth trajectories. Here, any isolated period in which the distance between two agents is below their combined radius R=0.4 m counts as a single collision.

5.4.4 Discussion

It can be seen from the Fig. 5.6 that both learned models provide a more accurate description of pedestrians' trajectories than the baseline model CV assuming constant velocity, and that good generalization from the training to the test set is achieved for both models NavIOC-L^2 and NavIOC-L^1. In terms of mean modeling error, NavIOC-L^2 outperforms NavIOC-L^1 especially at low Δt.

Similarly, the Fig. 5.7 shows that our models generate more accurate predictions than the baseline. For Δt <4 s, our models' predictions are also more accurate than those by [35]. Comparing NavIOC-L^2 and NavIOC-L^1, it is again visible that for low Δt, predictions by NavIOC-L^2 are more accurate on average than those by NavIOC-L^1, whereas for high Δt, the opposite can be said. For Δt >4 s, it seems that the approach by [35] would yield the most accurate predictions, considering the trend in the error curve (which remains speculation, however, due to missing data in this regime).

Regarding the Fig. 5.7, it is noteworthy that the error curves' sense of curva-

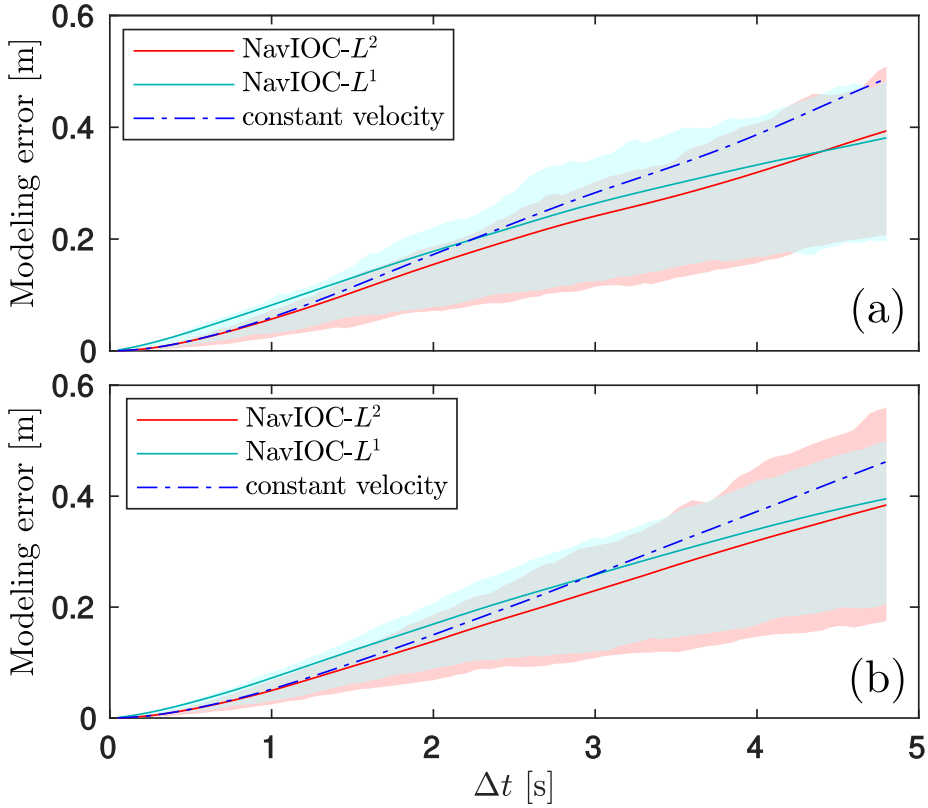

Figure 5.6: The modeling error is calculated as the distance between agents' observed and re-optimized positions as a function of time Δt from the trajectory's initial time. The modeling error's sample mean and the interval between its lower and upper quartiles (shaded area) are shown for our two models, as well as the sample mean for a baseline model assuming constant velocity. They are evaluated on (a) the training set $\mathcal{E}_1$ and (b) on the test set $\mathcal{E}_2$, which are both sampled from the dataset DIAMOR [54].

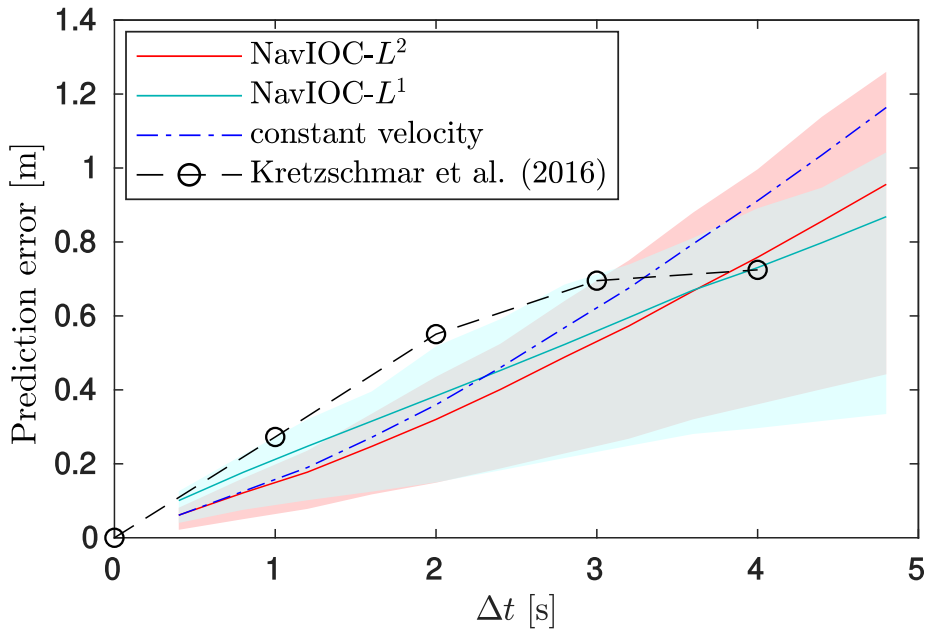

Figure 5.7: The prediction error is calculated as the distance between agents' observed and predicted positions as a function of time Δt from the instant at which the prediction is issued. The plot shows the prediction error's sample mean and the interval between its lower and upper quartiles (shaded area) for our two models, as well as the sample mean reported in [35] for their IRL approach and the sample mean for a baseline assuming constant velocity, all for the test set $\mathcal{E}_3$ sampled from the dataset ETH [103].

ture differs across models, namely, the curves for NavIOC-L^2 and CV bend upwards, whereas the curve by [35] bends downwards, and the curve for NavIOC-L^1 is rather straight. This observation is linked to how the different approaches take into account, on the one hand, agents' desired velocities or goals, and on the other hand, their initial velocities. In the framework of [35], trajectories' endpoints are specified a priori. Thus, when accurate estimates of agents goals are available, predicted and actual trajectories will tend to converge again for later points in time with their method. On the other hand, our models consider desired velocities and trade off following them versus keeping accelerations small. Since NavIOC-L^2 gives higher weight to accelerations than NavIOC-L^1 (cf. Table 5.1), it leads to slower adoption of desired velocities in favor of maintaining initial velocities (similarly as CV). This difference can also be observed in the generated trajectories, comparing e.g. the ones shown in the Fig. 5.5-(2.b, 2.c).

However, in contrast to the baseline model CV, which generates a large number of collisions, ours generate very few (NavIOC-L^2) or zero (NavIOC-L^1) collisions, as indicated in the Table 5.3. Our models can therefore be used to generate avoiding trajectories in collision-prone situations, which can guide an autonomous mobile robot, as shown in the next section.

Based on the Table 5.2, our approach appears to generate better predictions than the deep learning approach [63]. This comparison does not allow to identify one method as superior to the other, however, due to the different information given to the respective models. First of all, a key information which we provide to ours and which is not leveraged by theirs consists in knowing the direction in which a pedestrian's goal lies relative to the pedestrian's initial position. This clearly helps for long term predictions. Secondly, a good estimate of initial velocities helps short term prediction, which we exploit as provided in the form of causally filtered estimates given in the dataset, but which they do not exploit, since their approach is only based on positions during the past 8 time steps, i.e. 4.8 s. It is not clear how well one could estimate an agent's velocity based on these few steps, but since the estmitate provided by the dataset is likely obtained via Kalman filtering, it can be expected to be optimal and containing the trajectory's entire history.

On the other hand, our choice of comparison with the related IRL approach [35] is motivated by its similarity with our framework, as both are based on Maximum Entropy IRL in continuous spaces. Thus, the comparison presented here mainly serves to highlight differences within the class of such IRL approaches, and to show that an IRL approach based on the assumption of local optimality is suitable for modeling multi-agent navigation problems.

5.4.5 Limitations

According to the above discussion, our approach can only be used as a means to predict or interact with pedestrians if provided with an estimate of their desired velocity. For practical purposes, a pedestrian's observed velocity may be used as an estimate of the desired velocity, but this will likely diminish long

term prediction accuracy. On the other hand, the task to estimate pedestrians' desired velocities or goals on-line based on past observations has been tackled in prior work [104], which could be combined with our approach.

As another practical limitation, the current models do not include any information about the environment. Such information could be added in our framework by creating additional features describing the environment. This would be necessary for avoiding collisions with walls or other static obstructions.

Further, The chosen training and test data contains mostly bi-directional crowd flow. It is therefore not clear how well the learned models would generalize to more complex flow conditions. The focus on bi-directional flow is in agreement with the aforegoing chapter, but constitutes a limitation of the presented work. We note that pedestrians' desired direction of motion can be estimated more easily if they are moving in a corridor. Accordingly, our choice here to focus on such data is mostly motivated by the need for consistent training data.

Again, the learned models' capability to describe human–robot interactions rather than human–human interactions is not validated, given that they were trained and evaluated on the latter kind of data. Thus, the proposed approach only provides a technique that could also be applied to modeling human–robot interactions and possibly extended with additional features that capture their specific relations. The implicit assumption that position and velocity are sufficiently descriptive of such interactions also represents a limitation of the current approach. By leveraging additional sensing capabilities that a robot could be endowed with, a richer set of interactions could be modeled and learned from empirical data.

Finally, even if a representative pedestrian–robot interaction model could be learned by our approach, its scope would be limited to typical and cooperative behavior. Pedestrians who are not behaving according to the model will always remain a non-negligible challenge for a robot relying on such a model that is based on a single collective cost function. Therefore, such models can only reflect cooperative aspects of interactions, whereas behavior due to conflicting or unrelated individual objectives would need to be considered as well to obtain a wholistic approach, for example via a game theoretic model.

5.5 Robot Experiments

We implement a controller for the automated wheelchair Qolo [90], which we refer to as the robot (cf. Fig. 5.8), based on solving the forward optimal control problem (5.4) on-line, where the initial state $\mathbf{x}_o$ is specified in each control cycle by the robot's and surrounding pedestrians' estimated states of motion. The cost function is defined by the model NavIOC-L^1, i.e. by the feature vector $\mathbf{f}_{(L^1)}$ and the corresponding learned weights. We evaluate the implemented method's performance in a series of tests where the robot drives autonomously through a busy corridor of EPFL campus (cf. Fig. 5.8).

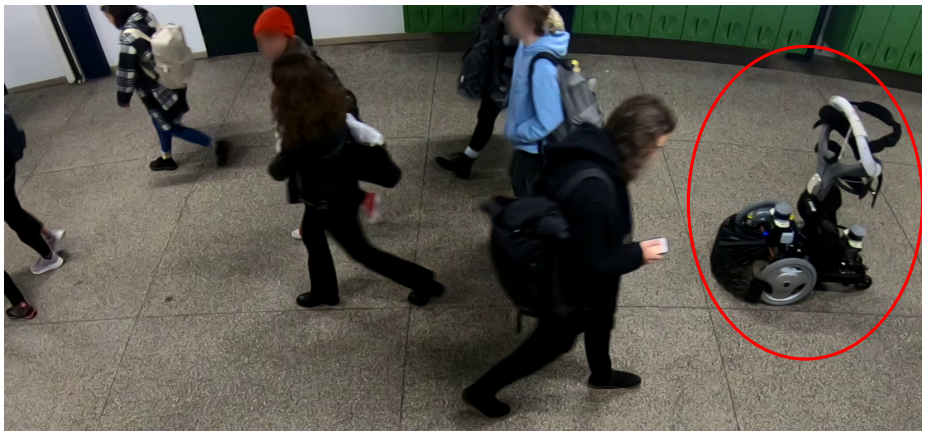

Figure 5.8: The robot Qolo is driving autonomously on EPFL campus.

5.5.1 Implementation

For our approach, three specific domains of functionality are essential, namely, 1) estimating the robot's state of motion, 2) detecting and tracking pedestrians and estimating their state of motion, and 3) computing and executing the optimal trajectory dependent on these estimates.

Odometry

The robot's linear and angular position and velocity are estimated by a commercially available on-board camera with integrated inertial sensors and a corresponding software which fuses inertial and optical flow measurements.

Pedestrian tracking system

We implement a tracking system on the robot to track pedestrians who are perceived by a LiDAR sensor (mounted on the robot's front), which delivers planar range measurements (in total 900 points on 360°) at a frame rate of 20 Hz. We make the tracking system's source code in `C++` publicly available[2].

Following the paradigm of tracking-by-detection, we implement a fast algorithm which detects legs in range measurements. It first segments a frame's scanpoints by passing over them in angular order and setting breakpoints wherever the line between two consecutive scanpoints defines a too shallow angle with the rays from the sensor to the respective scanpoints [113]. Then, circles are fit to segments with endpoints at most 0.3 m apart, using the approach in [114]. Fit circles with a radius below 7 cm are retained as detections, and the circles' centers define the detections' positions.

[2] `https://github.com/epfl-lasa/fast-pedestrian-tracker`

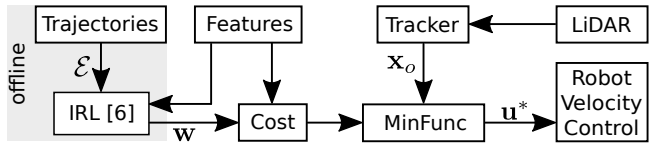

Figure 5.9: A cost function, which is parametrized by weights $\mathbf{w}$, is learned offline via IRL from a set $\mathcal{E}$ of multi-pedestrian trajectories. The robot navigates on-line by seeking collective actions $\mathbf{u}^*$ which optimize the learned cost, given the robots' and tracked pedestrians' estimated state $\mathbf{x}_o$.

This detector is operated at 20 Hz and robustly identifies peoples' legs, but it also detects various other objects having a slightly curved outline. However, falsely identifying objects as persons turns out to be beneficial for navigation around obstacles. Using the robot's estimated ego-motion, detections' positions are transformed from the moving LiDAR sensor's frame of reference to a static world frame, for further processing by the tracking system.

Our greedy approach to data association links detections over time by assigning each new detection to the track with the closest predicted position within a search radius of 0.75 m, or by initiating a new track if there is no existing sufficiently close track. Our track management removes tracks whose most recent associated detection is older than 1s. For each track, we update its estimate of position and velocity by fusing their predictions (assuming constant velocity) with the average position of the track's new associated detections using an α-β-filter (α=0.15, β=0.02).

Control system

The model NavIOC-L^1 is chosen for on-line prediction and motion planning, since it gives sufficient weight to desired velocities, such that the robot can be commanded via its desired velocity. Comparing to the learning phase, we increase the time step to h=0.4 s and use only K=12 steps to obtain the same planning time horizon of 4.8 s at lower computational cost. To account for the robot's larger diameter, we set R=0.6 m for the interaction energy feature.

Two asynchronous control loops are implemented, where the first loop computes a reference trajectory by solving the forward optimal control problem (5.4), operating at around 8–10 Hz, and the second loop computes velocity commands at 20 Hz for the robot's low-level velocity control system by evaluating the aforementioned reference trajectory's velocity at the current time and at the agent's index representing the robot. We use the package MinFunc[3] to optimize a cost over the robot's and tracked pedestrians' actions on-line (cf. Fig. 5.9).

The position and velocity variables in the initial state $\mathbf{x}_o$ are defined by the robot's and pedestrians' position and velocity estimates. We set pedestrians' desired velocities equal to their estimated velocities. To estimate the robot's velocity, we assume that it perfectly adopts the given velocity commands as

[3]https://www.cs.ubc.ca/~schmidtm/Software/minFunc.html

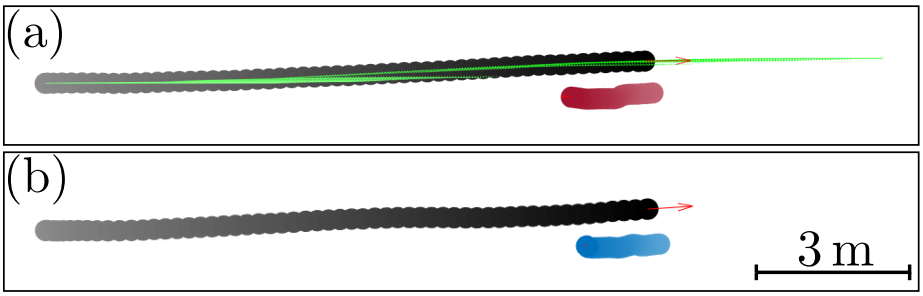

Figure 5.10: The robot's estimated trajectory is shown (black circles) as it avoids a static obstacle (red/blue), which it detects and tracks using its on-board tracking system. Darker colors indicate later points in time. Note that the obstacle is not moving and that its apparent motion is due to errors in the robot's self-localization. (a) The robot follows a reference trajectory (green) planned on-line by our approach NavIOC-L^1. (b) The robot uses a Velocity Obstacle approach based on [20] to determine its velocity command, where the time horizon is set to τ_{vo}=7 s.

long as they are within its acceleration bounds, since the velocity estimated by our odometry module is too noisy to yield smooth performance.

The number of pedestrians included in $\mathbf{x}_o$ is limited to 3 in order to keep the time to compute the robot's reference trajectory, i.e. to solve (5.4), lower than around 0.15 s, which is still practically useful for responding to pedestrians, considering that the solution incorporates a non-linear prediction of their motion. For each tracked pedestrian, a score is computed as

$$\text{score} = -|y| - 0.25\max(x, 0) + 3\min(x, 0) \tag{5.16}$$

where x and y denote the pedestrian's position relative to the robot in its forward and lateral direction, respectively. The 3 highest scoring pedestrians, typically those directly in front of the robot, are selected.

Due to the robot's non-holonomic differential drive kinematics, its velocity command space is only two-dimensional and represented here by the linear forward velocity v (of its wheel axle's midpoint) and its angular velocity ω. The linear reference velocity is interpreted as the desired velocity for a reference point located 0.25 m ahead of the robot's wheel axle, which is mapped via the non-holonomic constraints on ω and v.

5.5.2 Experimental Protocol

Static obstacle

First, a preliminary experiment was performed involving only the robot and a static obstacle, chosen as a cylindrical shape such that it would trigger the robot's leg detector and could be tracked as if it was a pedestrian. In each trial,

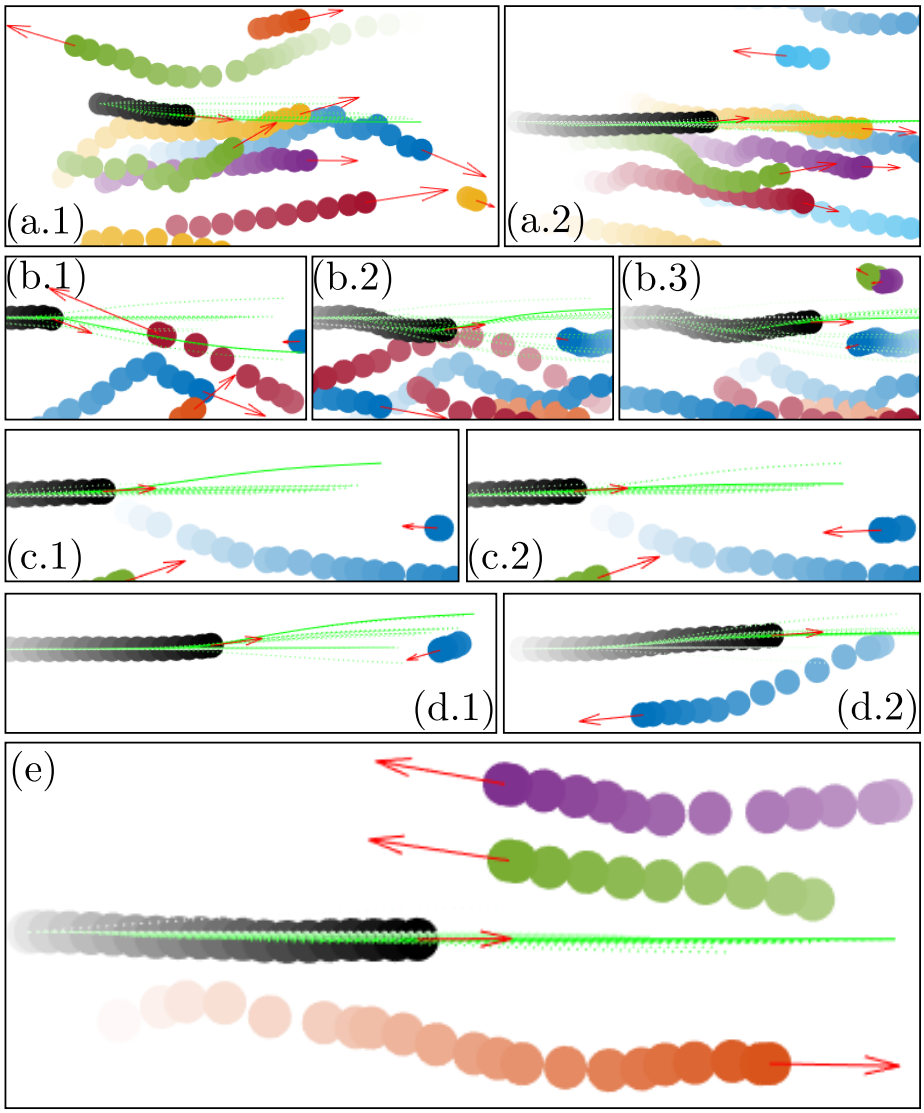

Figure 5.11: Exemplary motions recorded during our experiments with the robot Qolo on EPFL Campus are visualized. The robot (black) and pedestrians (various colors) are depicted by circles of diameter 0.4 m, such that circles for later instants are drawn on top of earlier circles and in more saturated colors. Red arrows indicate the most recent velocity. Similarly, the robot's recently planned paths are drawn (green, dotted), where more saturated colors indicate later plans, and its current plan is drawn as well (green, solid). Each row of subfigures depicts a different situation, where time increases from left to right.

the robot was positioned around 15 meters away from the obstacle and oriented such that it was facing the obstacle as precisely as possible. Then a constant desired velocity of 1 m/s aligned with its initial orientation was issued to the robot, and the robot used either our method NavIOC-L^1 or a Velocity Obstacle approach based on [20] to determine its velocity command. Motions were also recorded by an external camera. One trial per method was performed.

Human crowds

Experiments with pedestrians were performed in a corridor on EPFL campus. For each trial, the robot was positioned at the same starting point and received a constant desired velocity aligned with the corridor and of magnitude 1 m/s. It then drove autonomously over a distance of around 20 m and was then stopped by a supervisor, or earlier if necessary for pedestrians' safety. In total, 6 trials were performed, where one trial involved interaction with the supervisor, which is shown in the Fig. 5.11-(d.1-2), and other trials only involved interactions with uninformed pedestrians. Interactions were recorded by an external camera and the robot's on-board systems for sensing and tracking pedestrians.

5.5.3 Results

The Fig. 5.10 shows the robot's motion in the preliminary experiments with the static obstacle. It successfully avoids a collision and passes the obstacle on the left side in both trials.

For the experiments involving pedestrians, we first report qualitative observations of the robot's behavior and its interactions with pedestrians, which are illustrated by the Fig. 5.11.

Qualitative description of interactions

The Fig. 5.11 shows selected interactions between the robot and pedestrians, which are briefly described here. **(a.1)** The robot starts to move and immediately avoids a pedestrian (green) heading the other way. **(a.2)** The robot has joined a lane of pedestrians, and one of them (green) steps away from the robot. **(b.1)** The robot attempts to avoid a pedestrian (red) on the less efficient side. **(b.2)** The pedestrian (red) has avoided the robot on the more efficient side, and the robot attempts to avoid a standing pedestrian (blue). **(b.3)** The robot attempts to pass in between the two standing pedestrians (blue and green/purple) and is about to be stopped by the supervisor. **(c.1-2)** At first, the robot plans to avoid a pedestrian (green), but the pedestrian changes course faster, such that the robot's plan changes back to a straight path. **(d.1-2)** The robot and a pedestrian (the supervisor) reciprocally avoid each other. **(e)** A pedestrian (red) overtakes the standing robot and then changes direction away from the robot as it starts moving; subsequently, the robot shows slight adaptation to approaching pedestrians (green, purple).

Reciprocal avoidance

We generally observe that, in order to prevent imminent collisions, pedestrians adapt more to the robot than vice-versa. Nonetheless, the robot's contribution to collision avoidance is mostly constructive, i.e. it changes direction in such a way that the minimum predicted distance increases. This is exemplified by the situations depicted in the Fig. 5.11-(a.1, b.3, c.2, d.2, e). We also observe one case where the robot's response to an imminent collision does not match that of the involved pedestrian, as the robot plans to avoid the pedestrian on the right, whereas the pedestrian plans to avoid the robot on the left, which is depicted in the Fig. 5.11-(b.1, b.2). In this particular case, at a time when the pedestrian has already changed course such that their paths would not cross, the robot rotates, attempting to cross the pedestrian's future path.

Unilateral avoidance

We observe one case where a distracted pedestrian stands in the robot's way, such that the robot needs to avoid a collision by itself, which is depicted in the Fig. 5.11-(b.3). Here, the robot exhibits the correct tendency but does not achieve sufficient clearance, and thus, the supervisor needs to stop it before the robot's right-hand wheel runs over the pedestrian's foot.

Backwards interactions

We observe two cases where a pedestrian actively increases clearance to the robot which is driving behind the pedestrian, which are depicted in the Fig. 5.11-(a.2, e).

Quantitative results

In 5 out of 6 trials, the robot arrived at the end, whereas in one trial, the supervisor had to stop it to guarantee a pedestrian's safety. The robot's speed's average and standard deviation over all trials are $0.81 \pm 0.16\,\mathrm{m/s}$. For the distance traveled along the corridor's axis and the duration of a trial, the mean and standard deviation are given by $14.68 \pm 4.74\,\mathrm{m}$ and $18.00 \pm 5.02\,\mathrm{s}$, respectively.

5.5.4 Discussion

The robot's contribution to collision avoidance is often constructive, as shown by the examples in the Fig. 5.11, but relatively small, in comparison to pedestrians' contributions. On the one hand, the robot's ability to execute planned trajectories is limited by the robot's acceleration bounds and the absence of position-feedback for tracking the trajectory (since our system's position estimates were found to be too unreliable for feedback control). On the other hand, in the example depicted in the Fig. 5.11-(b.1), it is visible that planned trajectories may lack consistency over time, i.e. they may fluctuate between alternative plans, such as avoiding someone on the left or on the right. It is clear that the

robot will not execute either of the alternative plans properly as long as it keeps switching between them. A related issue becomes apparent when considering the situation depicted in the Fig. 5.11-(b.3). There, the robot plans its path between two pedestrians, but they do not stand far enough apart and one of them does not see the robot. However, since the robot performs a local optimization, it sticks to the plan to pass between them, which has a lower cost than driving head on into either of them. A remedy for both aforementioned issues could be found in restricting solutions to some admissible set, or in performing some global analysis to guarantee that a suitable local minimum is chosen in a unique fashion.

The case where the robot had to be stopped by the supervisor not only highlights the issue of local minima discussed above. It also shows that our current approach is insufficient for modeling non-cooperative behavior as in the present case, where a pedestrian is talking to another pedestrian and not giving way to the robot. This framework generates motion plans in which all agents contribute to minimizing a collective cost by adapting to some extent to each other. Thus, such plans entail incorrect predictions of non-interacting agents' motion, and only by updating its plan at a high frequency, the robot can successfully avoid completely passive agents or objects (as in the preliminary experiment, cf. Fig. 5.10).

5.5.5 Limitations

A technical limitation stems from assuming agents' desired velocities to equal their current velocities, which could be overcome by a more elaborate probabilistic estimation technique. Arguably, this would help to predict crossing order in situations as in Fig. 5.11-(b.1, b.2).

The presented experimental results are by no means systematic or representative of interactions that are likely to occur between mobile robots and pedestrians. On the one hand, a corridor in a campus environment has been chosen for practical reasons, which introduces a bias towards over-representing younger adult populations, and which does not contain particularly challenging or dangerous elements that mobile robots need to cope with in open environments, e.g. staircases going down, roads next to sidewalks, open train tracks, traffic lights, etc. On the other hand, the amount of collected results is too small to allow to infer any statistically significant tendencies or characteristics of such human–robot interactions.

The results can, at best, point out pertinent issues and directions for further research. In this respect, they show that autonomous mobile robots driving seemingly unsupervised alert pedestrians to a certain degree, as evidenced by the case where a pedestrian actively increased the distance to the robot, without a collision being imminent. Further, it becomes evident that a robot needs to correctly assess whether a person is attentive to it or not. Lastly, considering that pedestrians who noticed early that the robot was in their way adapted to it much more than vice versa, a clear challenge can be seen in working towards a robot's capability to be proactive in such cases, which is not only an algorithmic

challenge, but also a challenge for robotic hardware and system engineers.

5.6 Conclusion

In this chapter, we have applied a framework [79] for inverse reinforcement learning (IRL) to recover a cost function from examples of jointly navigating pedestrians. This cost function includes a novel feature for IRL of multi-agent navigation, namely our approximation of an interaction energy defined in related work [57]. Our results show that a meaningful cost function can be learned with the proposed model, which yields collision avoidance in combination with smooth transitioning from the initial to the desired velocity. Our models' predictions outperform the most closely related approach's predictions [35] on test samples from the dataset ETH [103], and we attribute this fact partially to slower adoption of the desired velocity. Interestingly, the results of learning strongly depend on the feature employed to penalize for acceleration, where penalty by the L^2 norm yields a cost function which gives much lower weight to tracking the desired velocity, in comparison to the L^1 norm.

Experiments with the robot Qolo have demonstrated that our approach can plan reasonable trajectories on-line in real-world interactions with pedestrians. However, the robot's hardware limits the actual performance at executing those plans. Also, issues associated with local minima require further developments to ensure that the robot safely avoids contact. Thus, we have validated that the approach holds the potential to smoothly guide a robot in human crowds, once local minima are handled appropriately and given that the robot is equiped with sufficiently accurate state estimation and trajectory tracking control.

Chapter 6

Comparisons of Methods

This chapter presents a comparison in simulation of methods proposed or used in previous chapters, in order to facilitate readers to view their individual pros and cons. For this purpose, a system description is established in the following section, which allows to compare the different methods within a common framework. Then, we briefly describe the methods' implementation details for the comparisons in this chapter, before presenting the experiments that were performed.

6.1 System Definition

We consider a crowd of n agents in a given environment. We characterize each agent's motion as the motion of a point with holonomic kinematics. Let $\mathbf{p}_i \coloneqq \begin{bmatrix} p_{i,x} & p_{i,y} \end{bmatrix}^{\mathrm{T}}$ denote the i-th agent's position. We assume disk-shaped agents, i.e. that the space occupied by the i-th agent is a disk of radius r_i, centered at $\mathbf{p}_i$. We assume that agents attempt to avoid collisions of their bounding shapes. In other words, for any two agents with indices i and j, we assume that they aim to ensure that $|\mathbf{p}_i - \mathbf{p}_j| > r_i + r_j$ always holds.

Let $\mathbf{v}_i \coloneqq \begin{bmatrix} v_{i,x} & v_{i,y} \end{bmatrix}^{\mathrm{T}}$ denote the i-th agent's velocity. We define the system's state as

$$\mathbf{x} \coloneqq \begin{bmatrix} p_{1,x} & p_{1,y} & \dots & p_{n,y} & v_{1,x} & v_{1,y} & \dots & v_{n,y} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{4n}, \tag{6.1}$$

consisting of all individual positions and velocities. Furthermore, let $\mathbf{a}_i \coloneqq \begin{bmatrix} a_{i,x} & a_{i,y} \end{bmatrix}^{\mathrm{T}}$ denote the i-th agent's acceleration. We define the system's action (or control input) as

$$\mathbf{u} \coloneqq \begin{bmatrix} a_{1,x} & a_{1,y} & \dots & a_{n,y} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{2n}, \tag{6.2}$$

assembling individual accelerations. We model the system's dynamics as

$$\mathbf{x}^{(k+1)} = \begin{bmatrix} \mathbf{I} & h\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{x}^{(k)} + \begin{bmatrix} h^2\mathbf{I}/2 \\ h\mathbf{I} \end{bmatrix} \mathbf{u}^{(k+1)}, \tag{6.3}$$

denoting by $\mathbf{0} \in \mathbb{R}^{2n \times 2n}$ the matrix with all elements equal to zero, by $\mathbf{I} \in \mathbb{R}^{2n \times 2n}$ the identity matrix, and by $h > 0$ the discrete time step. Accordingly, we assume that each agent updates its acceleration at the beginning of each time step and maintains a constant acceleration until the next time step. This assumption of a constant control input over each time step is common in the context of digital control systems (where it is termed as "zero-order hold"), as it approximates how such systems update their commands at every control cycle. We therefore consider it a reasonable assumption for modeling robots, whereas for modeling pedestrians, we can only see it as a practical assumption without a particular connection to actual human motor control, whose complexity is beyond the scope of this work.

6.2 Methods

We consider several alternative strategies for avoiding collisions, corresponding to the methods presented in aforegoing chapters. To simplify a comparison, we assume a holonomic robot with circular bounding shape. Thus, the method Reactive Driving Support (RDS) from Chapter 3 will be used here in a form equivalent to the method of Optimal Reciprocal Collision Avoidance [20], which will be termed generically as VO in this chapter. Further, the Acceleration Obstacle (AO) from Chapter 4 will be used in three variants, which employ different techniques to approximate the curved shape of acceleration obstacles. The first variant, termed as AO-0, is identical to the method presented in the Chapter 4, Section 4.3.1. The other two variants, termed as AO-1 and AO-2, will be described briefly in the following sections. Furthermore, the method proposed in Chapter 5 will be included in the qualitative comparison in Section 6.3.2.

6.2.1 VO (RDS)

Considering that VO are formulated in velocity space, it is necessary to map the command computed via VO on accelerations, as we use a second order simulation framework for comparing different control approaches. Let $\hat{\mathbf{v}}_i^{(k)}$ denote the optimal new velocity computed via VO for the i-th agent. We assume that it is mapped on a corresponding acceleration according to $\mathbf{a}_i^{(k)} = \left(\hat{\mathbf{v}}_i^{(k)} - \mathbf{v}_i^{(k-1)}\right)/\gamma$, where $\gamma > 0$ defines an inverse gain. We use γ=0.5 s for all experiments in this chapter.

6.2.2 AO-1

For obtaining a linear constraint approximating a given AO, we consider an alternative approach which is more similar to the technique for obtaining a normal vector when approximating VO by linear constraints. Namely, the constraint's normal vector $\mathbf{n}$ is computed in exactly the same way as for VO by projecting the previous relative velocity on the VO's boundary. However, the offset b is computed such that the constraint's linear boundary touches the AO (in the

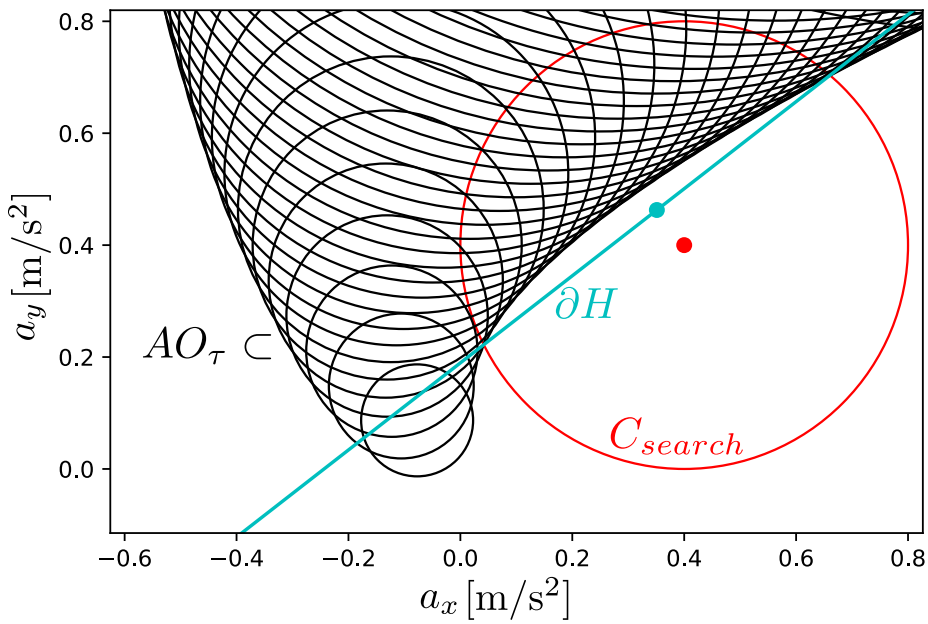

Figure 6.1: The figure illustrates a technique [40,42,72] employed by the method AO-2 to derive a convex constraint which tightly approximates the acceleration obstacle's locally concave boundary. Instead of the entire free acceleration space, a local search region C_{search} around the previous acceleration command is considered. Then, a linear constraint is computed whose boundary ∂H touches the convex hull of the intersection of C_{search} with a circles-based over-approximation of the acceleration obstacle AO_τ.

space of relative acceleration). Thus, the alternative scheme to obtain the normal vector differs from the one in the Section 4.3.1, firstly, by computing the extrapolation to the time horizon (rather than only half of it), and secondly, by projecting the computed extrapolation on the closest point of the occlusion cone's boundary even if it is inside the occlusion cone (rather than computing the point at which its ray enters the occlusion cone).

Since with this approach, the point at which the linear constraint's boundary touches the AO_τ can tend towards infinity, it is necessary to impose an upper bound on the constraint's boundary's offset from the origin, as already discussed in the Chapter 4 for AO-0. For AO-1, this happens when two agents' anticipated trajectories graze each other. Thus, AO-1 generates the maximum acceleration around the transition from colliding to non-colliding anticipated motions.

6.2.3 AO-2

As a third way to obtain a linear approximation, we implement an approach similar to the one proposed in prior work on Acceleration Velocity Obstacles [40],

which is illustrated in the Fig. 6.1. It limits the search for relative accelerations to a circle around the relative acceleration from the previous time step, and computes the convex hull of this circle's intersection with the acceleration obstacle. This convex hull is then replaced by a linear constraint which touches the hull and has maximum distance from the circle's center. Note that the same principle has also been used by other works on more general control obstacles [42, 72]. Our approach differs from these works by first generating a circles-based over-approximation which is used in place of the AO, whereas we do not sample the AO's boundary. However, it is considered beyond the scope of this investigation how these implementation details affect performance.

Among the three variants of AO considered here, AO-2 is the only one which allows to plan maneuvers into visually occluded regions behind obstacles (cf. Fig. 6.3). Such maneuvers correspond to accelerations that are inside the acceleration obstacle's convex hull but outside of the acceleration obstacle. The variant AO-2 can access admissible accelerations inside the acceleration obstacle's convex hull, because it only computes the local convex hull around the intersection of its circular search region with the acceleration obstacle. The ability of AO-2 to operate close to the acceleration obstacle's locally non-convex boundary motivates its inclusion in the comparison here.

6.2.4 NavIOC-L^1

For the qualitative experiments in this chapter, the optimal control method proposed in Chapter 5 with the learned objective NavIOC-L^1 is used. At each simulation step, an optimization problem over the horizon is formulated and solved, which yields trajectories for all agents, whose first actions are applied at the current time step. This is repeated at the next time step, such that a form of model predictive control is implemented.

6.3 Experiments

The Sections 6.3.1 and 6.3.2 compare the aforementioned methods quantitatively and qualitatively, respectively.

6.3.1 Quantitative Study

To compare RDS/VO and AO regarding their suitability for navigation in crowded environments, a quantitative simulation study is performed. To facilitate comparisons with previous chapters, particularly with the Chapter 4, a corridor environment is chosen again for the simulations in this section. The corridor environment is defined on a rectangle of width and height equal to 25 m and 8 m, respectively (cf. Fig. 6.2), whose vertical limits define walls that agents may not surpass.

To simulate pedestrians moving in both directions of the corridor, we use the Universal Power Law (UPL) [57] as crowd motion model. On the one hand,

Table 6.1: Quantitative results for the corridor flow experiments are shown, with best values set in bold. Significant differences ($p < 0.05$) from the best mean value(s) per sub-table and per column are marked by asterisks (*).

Method	Efficiency [-]	Acceleration [m/s^2]	Collisions [-]
Robot with time horizon τ=1 s			
VO	**0.98***±0.04	**0.03***±0.03	0.43*±0.84
AO-0	0.94*±0.05	0.17*±0.08	**0.12***±0.37
AO-1	0.97*±0.04	0.11*±0.09	**0.12***±0.37
AO-2	0.88*±0.24	0.20*±0.13	0.17±0.45
Robot with time horizon τ=6 s			
VO	**0.91***±0.21	**0.08***±0.06	0.35±0.65
AO-0	0.52*±0.22	0.12*±0.05	**0.22***±0.52
AO-1	0.53*±0.46	0.41*±0.19	0.23±0.59
AO-2	0.20*±0.37	0.49*±0.14	0.67*±1.06
Pedestrian			
UPL	0.95±0.05	0.20±0.12	0.18±0.39

this choice allows to gain insight in how the crowd's motion model influences results by comparing with the simulations from the Chapter 4. On the other hand, the UPL defines a highly efficient behavior, which can be considered as a baseline to compare the robot's performance with.

The UPL generates repulsive forces due to anticipated collisions with other agents and the corridor's walls. Recall that the potential whose spatial gradient defines the repulsive forces for UPL is given by $E := \eta\tau^{-2}\exp(\tau/\tau_o)$. Here, the parameters chosen for UPL are η=1 J and τ_o=3 s.

Each trial involves 50 agents. The crowd is initialized at random positions in the corridor environment, with a random choice of preferred direction (left or right) and individual preferred speeds sampled uniformly from the interval 1.33±0.4 m/s. Upon leaving the corridor on one side, agents are re-inserted at the other side. Thus, the crowd's spatially averaged density in the corridor is constant and equals 0.25 P/m^2. Each agent's radius equals 0.25 m. Agents are subject to radial bounds on velocity and acceleration, which are set to 1.5 m/s and 2 m/s^2, respectively.

Four methods to control the robot were tested, namely VO (RDS), AO-0, AO-1, and AO-2. For each method, two time horizons were considered, namely a very short one given by τ=1 s and a medium one equal to τ=6 s. Apart from pedestrians, the walls are also encoded as obstacles for all methods controlling the robot. The diameter of C_{search} for AO-2 is set to 1 m/s^2. Per method and time horizon, 60 trials were performed, where each trial's duration is 20 s.

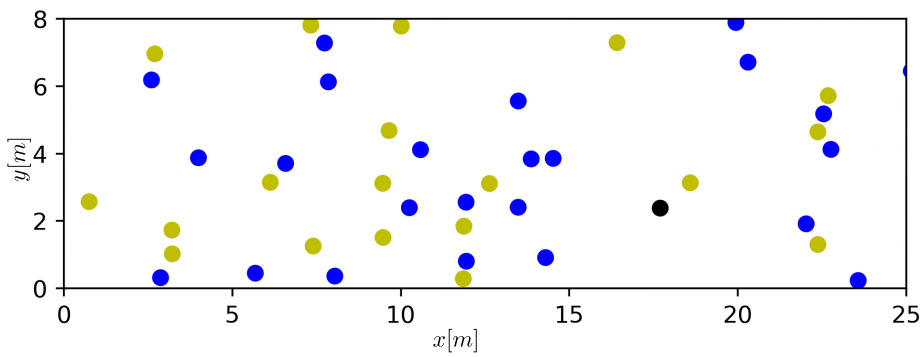

Figure 6.2: Simulated agents are shown in the corridor environment, where walls confine agents in the area between $y = 0$ and $y = 8\,\mathrm{m}$. They are either heading to the right (blue) or left (yellow), and the robot (black) is also heading to the right in this example. In the depicted situation, horizontal lanes have emerged.

Results

To quantify performance, three metrics are evaluated for the robot, namely average efficiency, average acceleration effort, and the number of collisions in a given trial. The Table 6.1 reports each metric's average and standard deviation over all trials, for each method and for both time horizons. It also reports the metrics' values measured for an arbitrary pedestrian, as a reference to compare with the robot.

The robot's efficiency is measured as its average velocity in the preferred direction divided by its preferred speed. Acceleration effort is measured simply by the robot's acceleration's average magnitude. Collisions are counted as isolated periods of inter-penetration between the robot and any pedestrian, where only the time after the first two seconds have passed is considered, in order to disregard collisions at initialization.

The results in the Table 6.1 can be summarized as follows. In terms of efficiency and acceleration effort, VO outperforms all other methods controlling the robot, for both time horizons. In contrast, regarding the expected number of collisions per trial, AO-0 and AO-1 are superior to VO and AO-2, also for both time horizons. Among the three variants of AO, the variant AO-2 consistently achieves the lowest performance with respect to all metrics. It is also important to notice that the reference pedestrian undergoes fewer collisions than the robot with any controller with a time horizon of τ=6 s. Furthermore, it can be seen that increasing the time horizon leads to fewer collision for VO, but to more collisions for AO.

Discussion

The results presented above and in the Table 6.1 contain expected but also possibly surprising observations. For VO, as the time horizon is increased,

the robot undergoes fewer collisions but also becomes slightly less efficient and spends more effort. This is not surprising, as the robot will contribute more to avoiding collisions and behave more conservatively with a larger time horizon, whereas pedestrians will need to make major contributions if the robot's time horizon is small. On the other hand, for AO, the robot undergoes more collisions with a larger time horizon. Therefore, it has to be assumed that the larger time horizon often leads to infeasibility of the set of constraints obtained from all surrounding agents, which is handled by introducing a slack variable as described in the Chapter 4, and which reduces the robot's safety. In AO-2, this issue is aggravated by restricting the search to C_{search}, which limits the robot's maximum jerk. The fact that all three variants of AO are drastically less efficient than VO for the larger time horizon can be explained by considering that each obstacle gives rise to a halfplane which the robot must avoid throughout the time horizon. If the horizon is large, the robot's accelerations are limited to very small values since the robot must be able to maintain them for a long period without colliding. This has been noted in prior work on the Acceleration Velocity Obstacle (AVO) [40], where large inverse gains δ where associated with slow response by the robot, which applies to AO, since AO are equivalent to AVO with $\delta \to \infty$ (cf. this thesis, Sec. 4.2.2).

Comparing with the results reported for AO-0 in Chapter 4, which were also obtained with $\tau = 1$s, we obtain similar results here in the sense that the robot is relatively efficient and capable of avoiding collisions. This can be explained to some extent by the phenomenon of lane formation and pedestrians' efficient way of avoiding collisions in the simulations here. In the simulations in Chapter 4, pedestrians are controlled by the Social Force Model [46], which is inferior to UPL at avoiding collisions, but due to the phenomenon of lane formation and the fact that these simulations were started from a steady state (in contrast to those in this chapter), individual collisions are less likely there.

6.3.2 Qualitative Comparisons

For the experiments in this section, the time horizon for VO and all AO variants is set to τ=8 s, whereas all other parameters are chosen as in the quantitative evaluation given in the previous section.

First, the Fig. 6.3 illustrates the different planning assumptions of constant velocity and constant acceleration for VO and AO, respectively. In the shown example, the robot needs to avoid a large circular obstacle to reach its goal on the right-hand side.

To gain further insights in qualitative differences between VO/RDS and AO, but also between the different variants AO-0, AO-1, and AO-2, a standard experiment [20,40] for reciprocal collision avoidance is performed in this section. Here, six agents are positioned on a circle (cf. Fig. 6.4), and we assign them goals on the circle opposite to their respective initial positions. Thus, they will meet in the middle and need to resolve the conflicts in their desired motions. In addition to VO/RDS and the three variants of AO, we include the crowd model UPL used in the previous section, and the cooperative optimal control method

which was proposed in the Chapter 5 with the learned cost function termed as NavIOC-L^1.

Results

The coordination experiments' results are documented in the Fig. 6.4. It can be seen that for all methods except AO-0, agents succeed at resolving their conflicting desired motions and reaching their goals. While the motions for VO are characterized by early adaptations and mostly straight segments, those for UPL adapt later and in a more successive fashion. In contrast, the motions for NavIOC-L^1 appear as the smoothest and as the most symmetric ones. The trajectories for AO-1 are similar to the ones as for VO, but use more space for avoiding each other. Finally, with AO-2, some trajectories exhibit oscillations, and their conflicts are resolved relatively late but in a more dynamic fashion.

Discussion

In the experiment depicted in the Fig. 6.3, the difference between planning with constant velocity (VO) or acceleration (AO) can be seen by comparing the respective paths' shapes. For VO, the robot moves almost straight to the the obstacle's furthest visible point, and then traces its boundary until the goal is in sight. Thus, the anticipated point of contact remains the same until the robot touches the obstacle at that point. Accordingly, the distance decreases to (almost) zero and remains constant while the robot traces the boundary. For AO, in contrast, the robot anticipates contact at points which are not visible from its current position, and thus, the robot never touches the obstacle's boundary but keeps getting closer to it for a longer time.

In the second experiment, depicted in the Fig. 6.4, it is not surprising that the smoothest and most symmetric motions result with NavIOC-L^1, considering that it finds the commands for all agents by solving a single joint optimization problem, that the cost favors smoothness, and that the problem is formulated over a time horizon. Thus, agents are allowed to resolve conflicts and to coordinate in a single time step, whereas for other methods, agents need to coordinate based on observed individual motions, such that their motions always remain sub-optimal, unless some form of steady state could be found, which is not the case in the given example until each agent has converged to its goal.

Comparing the different variants of AO in the Fig. 6.4, it becomes evident that the technique to determine an approximate linear constraint strongly influences agents' behavior. First of all, the stalled motion for AO-0 is a direct consequence of the scheme presented in Chapter 4 to determine the normal vector, which only reluctantly responds to an established sense of passing by an obstacle (i.e. left or right). In contrast, the scheme for AO-1 adopted from VO is effective for AO as well. For AO-2, an issue can be seen in the visible oscillations, which can be explained as follows. Once an agent's velocity exhibits a tendency to pass by another agent on one side (e.g. to the right), the nominal acceleration will point in the other direction (i.e. to the left), since the desired

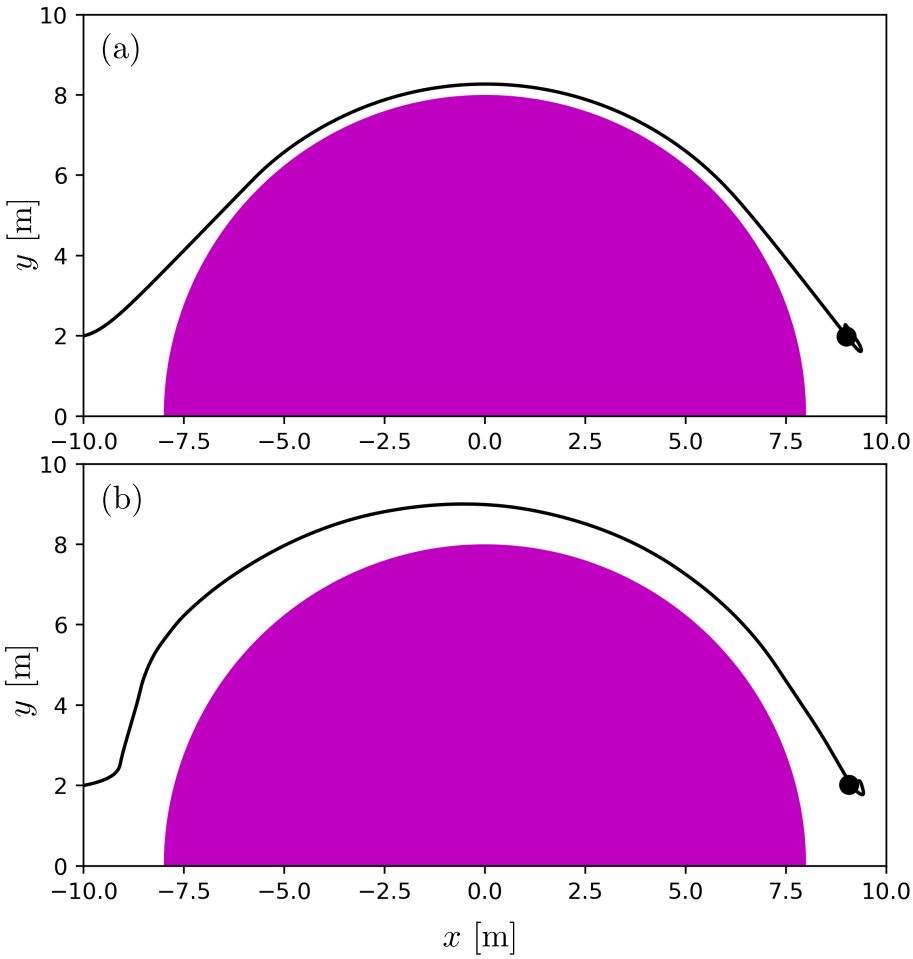

Figure 6.3: The plots (a,b) show a robot's simulated motion around an obstacle (magenta), where the robot uses either VO (a) or AO-2 (b). The robot's path (black curve) and its final position (black circle), which roughly coincides with its goal, are shown. Using VO, the robot moves tangentially with respect to the obstacle until the goal is in sight. Using AO-2, the robot plans parabolic trajectories that may enter regions which are visually occluded by the obstacle.

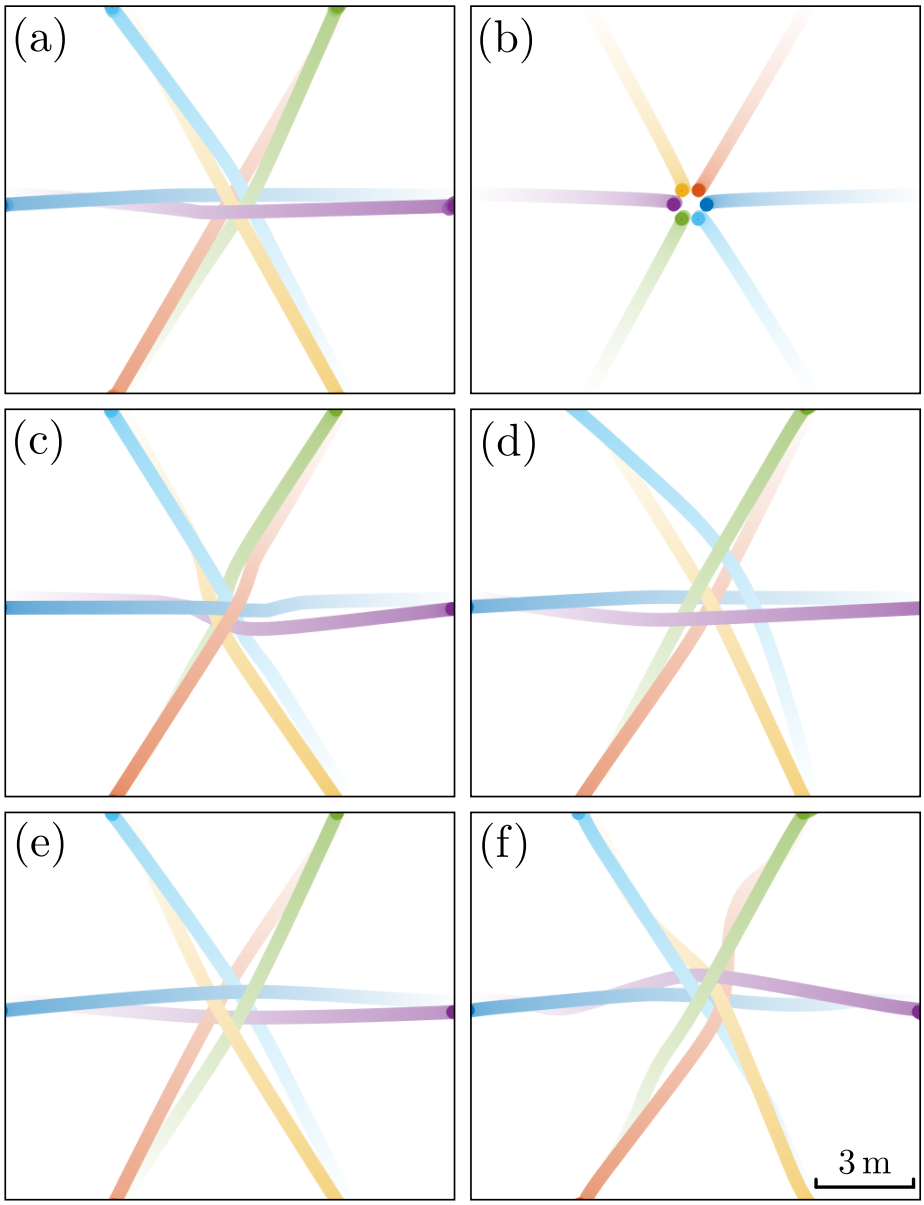

Figure 6.4: Each plot(a-f) shows six agents' motions, aiming to reach the point opposed to their initial position on a circle of radius 7 m. Later positions are drawn on top of earlier positions and with higher saturation. The initial condition is the same for each plot, whereas the control method, which all agents apply, is varied. The methods VO (a), UPL (c), and NavIOC-L^1 (e), which are illustrated in the left column, embed anticipation of collisions based on the assumption of constant velocity. The methods AO-0 (b), AO-1 (d), AO-2 (f), which are shown in the right column, anticipate collisions by assuming constant accelerations.

velocity (which points to the goal) is on that side of the current velocity. Then, the agent's acceleration will tend in this direction and the acceleration obstacle's boundary will be locally approximated around a point corresponding to passing on the other side. This process repeats until the moment when agents actually pass by each other at whichever side is currently favored. Thus, the scheme adopted from AVO [40] turns out to be ineffective for AO here. However, the choice of nominal acceleration law strongly affects the results, due to interactions with the approximation scheme of AO-2 as described above.

6.4 Conclusion

Based on the quantitative results and their discussion in this chapter, we can conclude that in crowded environments, AO is only suitable for very short term motion planning, but ineffective at avoiding collisions with longer anticipation, as shown for all three variants investigated here. While to some extent, the underlying schemes to determine linear approximations of AO could be improved, a basic issue remains due to the assumption of constant acceleration, which renders even small accelerations inadmissible in the presence of many obstacles.

The qualitative comparison has shown that the optimal control approach from Chapter 5 performs equally well as or even better than VO or UPL at coordinating agents, but the underlying assumption of centralized control needs to be viewed critically. In practice, some degree of understanding among pedestrians beyond pure observation of each others' motions can be expected. However, it remains an open research question how to model, possibly encourage, and exploit such forms of understanding between humans and robots, for the purpose of smoother and more efficient navigation.

Chapter 7

Conclusion

The thesis at hand has studied different problems related to robots navigating in crowded environments. In the introduction Chapter 1, recent approaches to planning such robots' motions have been reviewed and related to insights from crowd motion science. The Chapter 2 introduced some prerequisites for later chapters, including Acceleration Velocity Obstacles (AVO) and Inverse Reinforcement Learning (IRL).

The Chapter 3 has proposed a reactive control scheme developed for non-holonomic robots of non-circular shape to respond quickly to any imminent collisions with obstacles nearby. The quantitative evaluation in simulation has shown that the scheme is effective at avoiding collisions and suitable to complement a motion planner. Experiments on the standing wheelchair Qolo have confirmed the method's feasibility and shown that it allows to avoid collisions based on raw sensor data while following the given commands otherwise.

The Chapter 4 has investigated a technique for motion planning based on the Acceleration Obstacle (AO). It has shown how the AO's qualitative shape, particularly its boundary's sense of curvature, can be characterized dependent on the two involved agents' relative motion. Furthermore, the AVO has been shown to become equivalent to the AO under a limit operation. A method has been proposed for a robot's navigation using AO via a novel algorithm which exploits a geometric analysis of AO to compute conservative linear approximations of AO in closed form. The Social Force Model (SFM) has been employed to simulate crowds which interact with a robot using AO. The robot with AO progresses faster and undergoes very few collisions in comparison to the simulated pedestrians. Furthermore, it has been shown that the robot does not strongly interact with lane formation in the simulations and tends to move in between lanes rather than as part of them.

In the Chapter 5, models based on IRL have been developed to describe jointly navigating pedestrians. An approximation of the interaction energy defined in related work has been developed as part of the models. Predictions generated by the models are mostly collision-free and transition smoothly from the initial to the desired velocity. The predictions outperform the most closely

related approach's predictions on test samples from the dataset ETH. Furthermore, our models' predictions are more accurate than those by a recent deep learning approach to trajectory forecasting, but our approach relies on additional prior information in the form of non-causal estimates of desired velocities. Finally, the method has been implemented on the standing wheelchair Qolo, and a qualitative evaluation in real crowds has been performed. Its results show that the approach is capable of generating reference trajectories for the robot that exhibit complementary contributions to collision avoidance with respect to pedestrians' maneuvers around the robot.

The Chapter 6 presented a comparison across methods from previous chapters. The chapter's quantitative study has shown that AO are only suitable for motion planning with very short time horizons, whereas for larger time horizons, AO-based methods yield poor performance while Velocity Obstacles (VO) perform well. The chapter's qualitative comparison illustrated that the proposed centralized IRL method can coordinate multiple agents more efficiently than de-centralized approaches, such as VO or AO, which is expected, considering that it solves a joint optimization problem for all agents.

7.1 Limitations

The work presented in this thesis has not fully overcome the following issues and the proposed methods are subject to the following limitations.

Robot–pedestrian interaction The techniques have been developed under consideration of how humans interact with each other in crowds. Specific knowledge of how humans interact with robots has not been taken into account. Similarly, the evaluations performed in simulation are based on models of interactions between humans only. On the other hand, experiments with the robot Qolo in real interactions with pedestrians have provided some evidence that pedestrians tend to avoid robots similarly as other pedestrians obstructing their preferred motion. Accordingly, a limitation of this work consists in the assumption that pedestrians will not distinguish strongly between the type of agent they encounter. The methods do not address e.g. how to interact with pedestrians who are curious and actively pursue the robot.

Non-cooperative navigation For the IRL approach presented in the last chapter, it is assumed that pedestrians cooperate with the robot. This assumption excludes interactions with distracted persons or persons who are reluctant to adapt to the robot for some other reasons. However, the techniques presented in the Chapters 3 and 4 can address such cases by assuming that other agents' velocities remain constant over the planning horizon. Their limitation consists in the inability to reason about non-cooperative behaviors beyond short time horizons, which would likely violate the assumption of constant velocity.

Complex environments The focus in this work has been on simple environments, e.g. corridors or open squares, that do not contain specific hazards or features that strongly affect pedestrians' behavior, as opposed to e.g. a platform in a train station or a road crossing with traffic lights. Thus, the presented methods would not be sufficient to yield reasonable and safe behavior in such complex environments.

7.2 Future work

The approach presented in the Chapter 4 to analyze the geometric properties of AO could likely be extended to other types of control obstacles. More sophisticated iteration-free algorithms to construct constraints from control obstacles could be devised. Particularly, constructing convex admissible sets that contain some points of the curved boundary (apart from the circular cap) would be interesting for achieving higher efficiency of the robot's maneuvers.

To fully exploit the potential of IRL for modeling navigation in crowds according to our approach in Chapter 5, it appears worthwhile to develop and investigate other features, in addition to the ones in our work. Features that model pairwise, triplet-wise or even n-wise relations are straightforward to incorporate in our approach, and sufficient amounts of data can be found in the orignial dataset we used for training. Also, simple features to account for static obstacles and more sophisticated, even time-dependent features to account for other environmental factors can be considered.

As a general promising direction to advance the field of robotic navigation in crowds, the author considers the establishment of a common taxonomy of capabilities which contribute to successful navigation in crowded environments. Accordingly, a series of benchmarks to evaluate these skills one by one as well as in combination could be developed. It is also conceivable to adopt a certification-oriented perspective, and to evaluate robotic systems as a whole, rather than individual components or algorithms. In either way, the most crucial question is probably how to define a suitable set of skills or respective test criteria, such that they represent a robot's actual performance in a real crowded environment of interest. They could be divided e.g. in individual pedestrians' state estimation (moving/standing, robot-aware/distracted, etc.), group state estimation, interaction-aware motion planning, environment recognition, and task- and environment-specific metrics (e.g. point-to-point motion efficiency). Considering that the difficulty and algorithmic solution of the associated problems depends strongly on the entire robotic system's hardware equipment and on environmental conditions, it appears more sensible to adopt a wholistic system perspective for evaluating and benchmarking the developments to come.

Appendices

7.A Proof of Proposition 1 (AO-normals)

With $f(\mathbf{z},t) \coloneqq |\mathbf{z}-\mathbf{c}(t)|^2 - r^2(t)$, the intersection of infinitesimally close circles (4.5) is expressed by the two equations $f(\mathbf{z},t)=0$ and $f(\mathbf{z},t+\mathrm{d}t)=0$ in the unknown $\mathbf{z}\in\mathbb{R}^2$. Subtracting the first from the second equation and dividing by $\mathrm{d}t$ yields $\partial f(\mathbf{z},t)/\partial t = 0$ as $\mathrm{d}t \to 0$, or

$$0 = -\frac{1}{2r(t)}\frac{\partial f(\mathbf{z},t)}{\partial t} = \frac{(\mathbf{z}-\mathbf{c}(t))^T\,\dot{\mathbf{c}}}{r(t)} + \dot{r}. \tag{7.1}$$

With $\mathbf{n}_t = (\mathbf{z}-\mathbf{c})/r$ according to (4.6), one has equivalently

$$\mathbf{n}_t^T\dot{\mathbf{c}} + \dot{r} = 0, \quad |\mathbf{n}_t| = 1. \tag{7.2}$$

Computing derivatives from (4.3), one obtains

$$\begin{aligned}\mathbf{n}_t^T\frac{4}{t^3}(\mathbf{x}_o+\mathbf{v}_o t/2) - \frac{4R}{t^3} &= 0\\ \mathbf{n}_t^T(\mathbf{x}_o+\mathbf{v}_o t/2) - R &= 0, \quad |\mathbf{n}_t| = 1.\end{aligned}$$

Thus, the proof is complete. In [100], the condition given for a canal's boundary to exist is $|\dot{\mathbf{c}}| \geq |\dot{r}|$, which corresponds to (7.2) admitting any solution.

7.B Proof of Proposition 2 (AO-reduction)

In the non-trivial case that $\tilde{\tau} \neq \tau$, the line segment from $\mathbf{x}_o$ to $\hat{\mathbf{x}}(\tau)$ intersects the disk $D(\mathbf{0},R)$ at the point $\hat{\mathbf{x}}(\tilde{\tau})$. According to the Proposition 1, at every $t<\tilde{\tau}$, there are two distinct normals and boundary points, whereas at $t=\tilde{\tau}$, there is only one normal and boundary point. Since (4.7) is continuous in t, it follows that the local boundary forms two distinct branches for $t<\tilde{\tau}$ which meet when $t=\tilde{\tau}$. Hence, it remains to show that the local boundary over $(0,\tilde{\tau}]$ delimits a shape which contains the canal for any $t'>\tilde{\tau}$. By the Definition 2, the local boundary point for any time t with a given normal $\mathbf{n}_t$ is

$$\mathbf{b}_t = \mathbf{c}(t) + \mathbf{n}_t r(t) = 2\left(R\,\mathbf{n}_t - \mathbf{x}_o - \mathbf{v}_o t\right)/t^2.$$

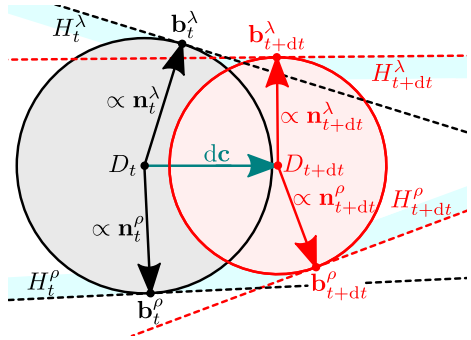

Figure 7.C.1: For a given t and small increment dt, an AO_τ's disks D_t and $D_{t+\mathrm{d}t}$ are shown with their left and right local boundaries $\mathbf{b}^\lambda$, $\mathbf{b}^\rho$, the associated normals $\mathbf{n}^\lambda$, $\mathbf{n}^\rho$, and the tangent halfplanes H^λ, H^ρ, respectively. One side's boundary is locally convex (concave) when each disk's tangent halfplane does (not) contain the other disk's tangent point as $\mathrm{dt} \to 0$.

Let H_t denote the halfplane with normal $\mathbf{n}_t$ whose boundary ∂H_t touches the local boundary at $\mathbf{b}_t$, calculated as

$$\begin{aligned} H_t &= \left\{\mathbf{a} | \mathbf{a}^T \mathbf{n}_t \leq \mathbf{n}_t^T \mathbf{b}_t\right\} \\ &= \left\{\mathbf{a} | \mathbf{a}^T \mathbf{n}_t \leq -\mathbf{n}_t^T \mathbf{v}_o / t\right\} \end{aligned}$$

Thus, for some (different) time t', it holds $D_{t'} \subset H_t$ iff

$$\begin{aligned} \mathbf{c}(t')^T \mathbf{n}_t + r(t') \leq -\mathbf{n}_t^T \mathbf{v}_o / t \iff \\ R - \hat{\mathbf{x}}(t)^T \mathbf{n}_t + \mathbf{n}_t^T \mathbf{v}_o \Delta t^2 / (2t) \leq 0 \iff \\ \mathbf{n}_t^T \mathbf{v}_o \Delta t^2 / (2t) \leq 0 \end{aligned}$$

with $\Delta t := t' - t$, where the last step uses (4.7). It is easy to see that for any $t \in (0, \tilde{\tau}]$, it holds $\mathbf{n}_t^T \mathbf{v}_o \leq 0$ (cf. Fig. 4.3-b). Hence, the condition is true for all t', which means that H_t contains the entire AO_τ. As this holds for every point on the local boundary with $t < \tilde{\tau}$, the AO_τ cannot intersect this boundary, which proves that the $AO_{\tilde{\tau}}$ contains subsequent disks, i.e. $AO_\tau = AO_{\tilde{\tau}}$.

7.C Proof of Proposition 3 (AO-shape)

For any time t and a small increment dt, we consider the two disks D_t and $D_{t+\mathrm{d}t}$ (cf. Fig. 7.C.1). As $\mathrm{d}t \to 0$, we consider for one local boundary (left or right) how $\mathbf{b}_t$ and $\mathbf{b}_{t+\mathrm{d}t}$ converge, i.e. their constellation with respect to each other's tangent halfplane H_t and $H_{t+\mathrm{d}t}$. If $\mathbf{b}_t \notin H_{t+\mathrm{d}t}$ and $\mathbf{b}_{t+\mathrm{d}t} \notin H_t$ as for the left boundary in the Fig. 7.C.1, they describe a locally concave boundary. Else if $\mathbf{b}_t \in H_{t+\mathrm{d}t}$ and $\mathbf{b}_{t+\mathrm{d}t} \in H_t$, a locally convex boundary is described, as for the right boundary in the Fig. 7.C.1. The other two possible constellations

(mixing $\in$ and $\notin$) are equivalent to the point $\mathbf{b}_t$ being contained within nearby disks' interiors and thus not contributing to the AO's boundary.

One finds (cf. Fig. 4.3) that both the left and right normal rotates monotonically with t on $(0, \tilde{\tau}]$. From the aforementioned constellations, the one describing a concave boundary is compatible with the normal rotating counter-clockwise, if it is the left boundary, or clockwise, if it is the right boundary. And the constellation describing a convex boundary is compatible with the normal rotating clockwise, if it is the left boundary, or counter-clockwise, if it is the right boundary. Both constellations are incompatible with the other sense of monotonic rotation in each case. Thus, the left boundary is convex iff its normals rotate clockwise, and it is concave iff its normals rotate counter-clockwise, whereas the right boundary is convex iff its normals rotate counter-clockwise, and it is concave iff its normals rotate clockwise.

Looking at the Fig. 4.3, one can verify quickly that the direction of rotation for both the left and the right normals is counter-clockwise if $\hat{\mathbf{x}}$ passes on the right of the disk $D(\mathbf{0}, R)$, whereas it is clockwise if passing on the other side. On the other hand, when on a pre-colliding course, the left and right normals rotate clockwise and counter-clockwise, respectively. When on a post-colliding course, the opposite holds. Combining these observations with the above considerations, the Proposition 3 follows.

7.D Proof of AVO converging to AO

AVO's center and radius functions [40] are respectively

$$\tilde{\mathbf{c}}(t) = \frac{\delta\left(e^{-t/\delta}\mathbf{v}_o - \mathbf{x}_o\right)}{t + \delta\left(e^{-t/\delta} - 1\right)}, \ \tilde{r}(t) = \frac{R}{t + \delta\left(e^{-t/\delta} - 1\right)}.$$

One can then obtain the expressions

$$\frac{\tilde{\mathbf{c}}(t)}{\delta} = -\frac{\mathbf{x}_o + \mathbf{v}_o t}{g(\delta)} + \frac{\mathbf{v}_o}{\delta}, \ \frac{\tilde{r}(t)}{\delta} = \frac{R}{g(\delta)},$$

with $g(\delta) \coloneqq \delta\,(t + \delta(\,e^{-t/\delta} - 1\,))$. Their limits for $\delta \to \infty$ are determined by g's limit, which can be calculated as $\lim_{\delta\to\infty} g(\delta) = t^2/2$, e.g. via the exponential's power series expansion. Comparing with (4.3), it follows that

$$\lim_{\delta\to\infty} \frac{\tilde{\mathbf{c}}(t)}{\delta} = -\frac{\mathbf{x}_o + \mathbf{v}_o t}{t^2/2} = \mathbf{c}(t), \ \lim_{\delta\to\infty} \frac{\tilde{r}(t)}{\delta} = \frac{2R}{t^2} = r(t).$$

7.E Proof of the Proposition 4 (Conservative Halfplane)

To prove the proposition, we show that the halfplane H also admits the interpretation as the set of accelerations that would cause within the time horizon a

collision not with the actual combined collider $D(\mathbf{0}, R)$ (shown in Fig. 4.5) but with the halfplane $\bar{H} \supset D(\mathbf{0}, R)$ whose boundary touches $D(\mathbf{0}, R)$ at $R\mathbf{n}$. To show this, let $\overline{AO}_\tau(\mathbf{n}) \coloneqq \{\mathbf{a} | \exists t \in [0, \tau] : \mathbf{x}(t; \mathbf{a}) \in H(\mathbf{n}, R)\}$ define the acceleration obstacle for halfplanes. It contains all accelerations that would make the two bodies collide within the time horizon τ under the motion model (4.1) if their shapes were halfplanes whose boundaries touch the original bounding circles and are orthogonal to $\mathbf{n}$. One can show that $\overline{AO}_\tau(\mathbf{n}) = H(\mathbf{n}, a)$, where $a \coloneqq \max_{t \in [0,\tau]} f(t)$ with $f(t) \coloneqq 2\left(R - \mathbf{n}^T\left(\mathbf{x}_o + \mathbf{v}_o t\right)\right)/t^2$. If there is $t^* > 0$ solving $f'(t) = 0 \iff R = \mathbf{n}^T \hat{\mathbf{x}}(t)$, then $f(t^*)$ is a global maximum on $(0, \infty)$, because $f'(t) > 0 \, \forall t \in (0, t^*)$ and $f'(t) < 0 \, \forall t \in (t^*, \infty)$. Else, $f'(t) > 0 \, \forall t \in (0, \infty)$. Therefore, if there is $t^* > 0$, $a = f(\min\{t^*, \tau\})$, else $a = f(\tau)$. It can be seen now that $a = f(\tilde{\tau})$ when the normal $\mathbf{n}$ is chosen according to the Algorithm 2. There, the Line 4 can also be expressed as $b = f(\tilde{\tau})$. Thus, it holds $a = b$, which proves that aforementioned interpretation is valid, i.e. both approaches lead to the same halfplane in acceleration space, which is conservative as $D(\mathbf{0}, R) \subset \bar{H}$.

Bibliography

[1] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[2] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *arXiv preprint arXiv:2103.05668*, 2021.

[3] F. Camara, N. Bellotto, S. Cosar, F. Weber, D. Nathanael, M. Althoff, J. Wu, J. Ruenz, A. Dietrich, G. Markkula, A. Schieben, F. Tango, N. Merat, and C. Fox, "Pedestrian models for autonomous driving part ii: High-level models of human behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5453–5472, 2021.

[4] F. Camara, N. Bellotto, S. Cosar, D. Nathanael, M. Althoff, J. Wu, J. Ruenz, A. Dietrich, and C. W. Fox, "Pedestrian models for autonomous driving part i: Low-level models, from sensing to tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6131–6151, 2021.

[5] B. Zhang, G. Barbareschi, R. Ramirez Herrera, T. Carlson, and C. Holloway, "Understanding interactions for smart wheelchair navigation in crowds," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, (New York, NY, USA), Association for Computing Machinery, 2022.

[6] H. Andersen, Y. H. Eng, W. K. Leong, C. Zhang, H. X. Kong, S. Pendleton, M. H. Ang, and D. Rus, "Autonomous personal mobility scooter for multi-class mobility-on-demand service," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1753–1760, 2016.

[7] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, *et al.*, "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *The International Journal of Robotics Research*, vol. 19, no. 11, pp. 972–999, 2000.

[8] M. Zhou, H. Dong, P. A. Ioannou, Y. Zhao, and F.-Y. Wang, "Guided crowd evacuation: approaches and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 5, pp. 1081–1094, 2019.

[9] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Intelligent autonomous systems 13*, pp. 335–348, Springer, 2016.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[11] M. Bosse and R. Zlot, "Map matching and data association for large-scale two-dimensional laser scan-based slam," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 667–691, 2008.

[12] D. Stadler and J. Beyerer, "Improving multiple pedestrian tracking by track management and occlusion handling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10958–10967, June 2021.

[13] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*, pp. 396–404, Springer, 1986.

[14] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[15] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, pp. 433–454, 2012.

[16] L. Huber, A. Billard, and J.-J. Slotine, "Avoidance of convex and concave obstacles with convergence ensured through contraction," *IEEE Robotics and Automation Letters*, vol. 4, pp. 1462–1469, Jan. 2019.

[17] L. Huber, J.-J. Slotine, and A. Billard, "Fast obstacle avoidance based on real-time sensing," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1375–1382, 2023.

[18] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of intelligent and robotic systems*, vol. 53, pp. 263–296, 2008.

[19] 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.

[20] 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.

[21] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational geometry: algorithms and applications*, ch. 4, pp. 71–82. Berlin, Heidelberg: Springer, 2008.

[22] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[23] 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.

[24] 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.

[25] C. Schlegel, "Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, (Victoria, BC, Canada), pp. 594–599, Oct. 1998.

[26] T. Howard, M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Model-predictive motion planning: Several key developments for autonomous mobile robots," *IEEE Robotics & Automation Magazine*, vol. 21, no. 1, pp. 64–73, 2014.

[27] S. Bouraine, T. Fraichard, and H. Salhi, "Provably safe navigation for mobile robots with limited field-of-views in dynamic environments," *Autonomous Robots*, vol. 32, pp. 267–283, Nov. 2012.

[28] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2012.

[29] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, (Brisbane, QLD, Australia), pp. 4601–4607, May 2018.

[30] 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.

[31] 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, pp. 335–356, Feb. 2015.

[32] 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.

[33] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6015–6022, 2019.

[34] 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, 2017.

[35] 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.

[36] A. Turnwald, W. Olszowy, D. Wollherr, and M. Buss, "Interactive navigation of humans from a game theoretic perspective," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 703–708, 2014.

[37] A. Turnwald, D. Althoff, D. Wollherr, and M. Buss, "Understanding human avoidance behavior: interaction-aware decision making based on game theory," *International Journal of Social Robotics*, vol. 8, no. 2, pp. 331–351, 2016.

[38] A. Turnwald and D. Wollherr, "Human-like motion planning based on game theoretic decision making," *International Journal of Social Robotics*, vol. 11, no. 1, pp. 151–170, 2019.

[39] C. Neumeyer, F. A. Oliehoek, and D. M. Gavrila, "General-sum multi-agent continuous inverse optimal control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3429–3436, 2021.

[40] J. van den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3475–3482, 2011.

[41] M. Rufli, J. Alonso-Mora, and R. Siegwart, "Reciprocal collision avoidance with motion continuity constraints," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 899–912, 2013.

[42] D. Bareiss and J. van den Berg, "Generalized reciprocal collision avoidance," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1501–1514, 2015.

[43] 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.

[44] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, 2020.

[45] B. D. Hankin and R. A. Wright, "Passenger flow in subways," *Journal of the Operational Research Society*, vol. 9, no. 2, pp. 81–88, 1958.

[46] D. Helbing, L. Buzna, A. Johansson, and T. Werner, "Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions," *Transportation science*, vol. 39, no. 1, pp. 1–24, 2005.

[47] S. Hoogendoorn and P. H.L. Bovy, "Simulation of pedestrian flows by optimal control and differential games," *Optimal Control Applications and Methods*, vol. 24, no. 3, pp. 153–172, 2003.

[48] U. Weidmann, "Transporttechnik der fußgänger: transporttechnische eigenschaften des fußgängerverkehrs, literaturauswertung," *IVT Schriftenreihe*, vol. 90, 1993.

[49] 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: Biological Sciences*, vol. 276, no. 1668, pp. 2755–2762, 2009.

[50] E. Goffman, *Relations in public.* Transaction Publishers, 2009.

[51] A.-H. Olivier, A. Marin, A. Crétual, and J. Pettré, "Minimal predicted distance: A common metric for collision avoidance during pairwise interactions between walkers," *Gait & Posture*, vol. 36, no. 3, pp. 399–404, 2012.

[52] R. S. Sobel and N. Lillith, "Determinants of nonstationary personal space invasion," *The Journal of Social Psychology*, vol. 97, no. 1, pp. 39–45, 1975.

[53] 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.

[54] F. Zanlungo, T. Ikeda, and T. Kanda, "Potential for the dynamics of pedestrians in a socially interacting group," *Phys. Rev. E*, vol. 89, p. 012811, Jan 2014.

[55] W. Daamen and S. P. Hoogendoorn, "Flow-density relations for pedestrian traffic," in *Traffic and granular flow'05*, pp. 315–322, Springer, 2007.

[56] D. C. Duives, W. Daamen, and S. P. Hoogendoorn, "State-of-the-art crowd motion simulation models," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 193–209, 2013.

[57] I. Karamouzas, B. Skinner, and S. J. Guy, "Universal power law governing pedestrian interactions," *Phys. Rev. Lett.*, vol. 113, p. 238701, Dec 2014.

[58] W. van Toll, F. Grzeskowiak, A. L. Gandía, J. Amirian, F. Berton, J. Bruneau, B. C. Daniel, A. Jovane, and J. Pettré, "Generalized microscropic crowd simulation using costs in velocity space," in *Symposium on Interactive 3D Graphics and Games*, I3D '20, (New York, NY, USA), Association for Computing Machinery, 2020.

[59] S. Paris, J. Pettré, and S. Donikian, "Pedestrian reactive navigation for crowd simulation: a predictive approach," *Computer Graphics Forum*, vol. 26, no. 3, pp. 665–674, 2007.

[60] F. Zanlungo, T. Ikeda, and T. Kanda, "Social force model with explicit collision prediction," *Europhysics Letters*, vol. 93, p. 68005, mar 2011.

[61] J. Amirian, J.-B. Hayet, and J. Pettré, "Social ways: Learning multi-modal distributions of pedestrian trajectories with gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[62] 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.

[63] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[64] T. Fraichard and V. Levesy, "From crowd simulation to robot navigation in crowds," *IEEE Robotics and Automation Letters*, vol. 5, pp. 729–735, Jan. 2020.

[65] 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, 2021.

[66] D. Vasquez, B. Okal, and K. O. Arras, "Inverse reinforcement learning algorithms and features for robot navigation in crowds: An experimental comparison," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Chicago, IL, USA), pp. 1341–1346, Sep. 2014.

[67] D. Wilkie, J. van den Berg, and D. Manocha, "Generalized velocity obstacles," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5573–5578, 2009.

[68] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed autonomous robotic systems*, pp. 203–216, Springer, 2013.

[69] A. Giese, D. Latypov, and N. M. Amato, "Reciprocally-rotating velocity obstacles," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3234–3241, 2014.

[70] A. Best, S. Narang, and D. Manocha, "Real-time reciprocal collision avoidance with elliptical agents," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, (Stockholm, Sweden), pp. 298–305, May 2016.

[71] Y. Ma, D. Manocha, and W. Wang, "Efficient reciprocal collision avoidance between heterogeneous agents using CTMAT," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, (Stockholm, Sweden), p. 1044–1052, International Foundation for Autonomous Agents and Multiagent Systems, July 2018.

[72] D. Bareiss and J. van den Berg, "Reciprocal collision avoidance for robots with linear dynamics using lqr-obstacles," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3847–3853, 2013.

[73] B. A. Francis and M. Maggiore, *Models of Mobile Robots in the Plane*, pp. 7–23. Cham: Springer International Publishing, 2016.

[74] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using the relative velocity paradigm," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pp. 560–565 vol.1, 1993.

[75] F. S. Miller and A. F. Everett, *Instructions for the Use of Martin's Mooring Board and Battenberg's Course Indicator*. 1903. (Published by Authority of the Lords Commissioners of the Admiralty, 1903).

[76] L. Tychonievich, D. Zaret, J. Mantegna, R. Evans, E. Muehle, and S. Martin, "A maneuvering-board approach to path planning with moving obstacles," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'89, (San Francisco, CA, USA), p. 1017–1021, Morgan Kaufmann Publishers Inc., 1989.

[77] 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.

[78] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning.," in *Aaai*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.

[79] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 475–482, 2012.

[80] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research, (Springer Tracts in Advanced Robotics (STAR))* (C. Pradalier, R. Siegwart, and G. Hirzinger, eds.), vol. 70, (Berlin, Heidelberg, Germany), pp. 3–19, Springer, 2011.

[81] A. V. Savkin and C. Wang, "A simple biologically inspired algorithm for collision-free navigation of a unicycle-like robot in dynamic environments with moving obstacles," *Robotica*, vol. 31, no. 6, pp. 993–1001, 2013.

[82] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation," in *Proceedings of Robotics: Science and Systems*, (Sydney, Australia), July 2012.

[83] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "Porca: Modeling and planning for autonomous driving among many pedestrians," *IEEE Robotics and Automation Letters*, vol. 3, pp. 3418–3425, July 2018.

[84] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, (Leuven, Belgium), pp. 1572–1577, May 1998.

[85] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Taipei, Taiwan), pp. 797–803, Oct. 2010.

[86] J. Van Den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3475–3482, IEEE, 2011.

[87] H.-T. (Lewis) Chiang, B. HomChaudhuri, L. Smith, and L. Tapia, "Safety, challenges, and performance of motion planners in dynamic environments," in *Robotics Research, (Springer Proceedings in Advanced Robotics (SPAR))* (N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, eds.), vol. 10, (Cham), pp. 793–808, Springer, 2020.

[88] P. Bevilacqua, M. Frego, D. Fontanelli, and L. Palopoli, "Reactive planning for assistive robots," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1276–1283, Jan. 2018.

[89] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Computer Graphics Forum*, vol. 26, no. 3, pp. 655–664, 2007.

[90] D. Paez Granados, H. Kadone, and K. Suzuki, "Unpowered Lower-Body Exoskeleton with Torso Lifting Mechanism for Supporting Sit-to-Stand Transitions," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, (Madrid, Spain), pp. 2755–2761, Oct. 2018.

[91] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "Smooth and collision-free navigation for multiple robots under differential-drive constraints," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Taipei, Taiwan), pp. 4584–4589, Oct. 2010.

[92] R. Seidel, "Small-dimensional linear programming and convex hulls made easy," *Discrete & Computational Geometry*, vol. 6, no. 3, pp. 423–434, 1991.

[93] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, (Kyoto, Japan), pp. 155–160, Nov. 2011.

[94] S. Breuers, L. Beyer, U. Rafi, and B. Leibe, "Detection- Tracking for Efficient Person Analysis: The DetTA Pipeline," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Madrid, Spain), pp. 48–53, Oct. 2018.

[95] M. Pfeiffer, U. Schwesinger, H. Sommer, E. Galceran, and R. Siegwart, "Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2096–2101, 2016.

[96] S. B. Liu, H. Roehm, C. Heinzemann, I. Lütkebohle, J. Oehlerking, and M. Althoff, "Provably safe motion of mobile robots in human environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1351–1357, 2017.

[97] B. Damas and J. Santos-Victor, "Avoiding moving obstacles: the forbidden velocity map," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4393–4398, 2009.

[98] D. Helbing and T. Vicsek, "Optimal self-organization," *New Journal of Physics*, vol. 1, pp. 13–13, aug 1999.

[99] J. van den Berg, S. Patil, J. Sewall, D. Manocha, and M. Lin, "Interactive navigation of multiple agents in crowded environments," in *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games*, I3D '08, (New York, NY, USA), p. 139–147, Association for Computing Machinery, 2008.

[100] M. Peternell and H. Pottmann, "Computing rational parametrizations of canal surfaces," *Journal of Symbolic Computation*, vol. 23, no. 2, pp. 255–266, 1997.

[101] 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, no. supp02, pp. 271–288, 2007.

[102] J. Kirkland and A. Maciejewski, "A simulation of attempts to influence crowd dynamics," in *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, vol. 5, pp. 4328–4333 vol.5, 2003.

[103] 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.

[104] 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.

[105] 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.

[106] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation," in *Proceedings of Robotics: Science and Systems*, (Sydney, Australia), July 2012.

[107] B. Kim and J. Pineau, "Socially adaptive path planning in human environments using inverse reinforcement learning," *International Journal of Social Robotics*, vol. 8, no. 1, pp. 51–66, 2016.

[108] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3931–3936, 2009.

[109] R. Alsaleh and T. Sayed, "Modeling pedestrian-cyclist interactions in shared space using inverse reinforcement learning," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 70, pp. 37–57, 2020.

[110] M. Fahad, Z. Chen, and Y. Guo, "Learning how pedestrians navigate: A deep inverse reinforcement learning approach," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 819–826, 2018.

[111] Y. Che, A. M. Okamura, and D. Sadigh, "Efficient and trustworthy social navigation via explicit and implicit robot–human communication," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 692–707, 2020.

[112] A. Konar, B. H. Baghi, and G. Dudek, "Learning goal conditioned socially compliant navigation from demonstration using risk-based features," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 651–658, 2021.

[113] G. A. Borges and M.-J. Aldon, "Line extraction in 2d range images for mobile robotics," *Journal of intelligent and Robotic Systems*, vol. 40, no. 3, pp. 267–297, 2004.

[114] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3402–3407, 2007.

Curriculum Vitae

David J. Gonon

Sep. 2018 – Dec. 2022	**Doctoral Assistant**, Learning Algorithms and Systems Laboratory (LASA), EPFL, Switzerland
Aug. 2017 – Aug. 2018	**Research assistant**, Institute of Structural Engineering (IBK), ETHZ, Switzerland
Sep. 2014 – Jun. 2017	**Student in MSc. Robotics, Systems and Control**, ETHZ, Switzerland
Mar. 2016 – Jun. 2017	**Student assistant** in research software development, Institute of Structural Engineering (IBK), ETHZ, Switzerland
Sep. – Dec. 2015	**Intern** in computer vision research, Sony Corporation, Japan
Sep. 2014 – June 2015	**Student assistant** in research software development, Institute of Machine Tools and Manufacturing (IWF), ETHZ, Switzerland
Sep. 2014 – Feb. 2015	**Teaching assistant** in fluid dynamics practicals, Institute of Fluid Dynamics (IFD), ETHZ, Switzerland
Sep. 2011 – Aug. 2014	**Student in BSc. Mechanical Engineering**, ETHZ, Switzerland
Jan. – Feb. 2012	**Intern** in workshop practice, Login AG, Switzerland