TUDelft

Delft University of Technology

Safe and Interaction-Aware Local Motion Planning in Human-Centered Environments

Safe and Interaction-Aware Local Motion Planning in Human-Centered Environments

Luzia Knödler

Safe and Interaction-Aware Local Motion Planning in Human-Centered Environments

Safe and Interaction-Aware Local Motion Planning in Human-Centered Environments

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended in public on
Tuesday, September 16, 2025, at 15:00

by

Luzia KNÖDLER

Master of Science in Engineering Cybernetics, University of Stuttgart, Germany,
born in Stuttgart, Germany.

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	Chairperson
Dr. J. Alonso-Mora,	Delft University of Technology, *promotor*
Prof. dr. R. Babuška,	Delft University of Technology, *promotor*

Independent members:

Prof. dr. ir. M. Wisse,	Delft University of Technology
Prof. Dr. V. Evers,	University of Twente
Prof. Dr. C. Fan,	Massachusetts Institute of Technology
Dr. Ö. Arslan,	Eindhoven University of Technology
Prof. dr. ir. J. Hellendoorn,	Delft University of Technology, *reserve member*

An electronic version of this dissertation is available at `http://repository.tudelft.nl/`.

Every real story is a never ending story.

Michael Ende, The Neverending Story

Contents

Summary

Autonomous mobile robots, once limited to structured settings like warehouses, are now operating in dynamic, human-centered environments such as hospitals, streets, and homes, where interaction with humans is unavoidable. This shift introduces significant challenges for the local motion planning of the robot. The robot must navigate complex environments where human behavior is difficult to predict and influenced by the robot's actions. Additionally, the robot's movements must prioritize safety, adapt to changing conditions, and adhere to social norms. Consequently, local motion planners must be interaction-aware, safe, socially compliant, and adaptive. Motion planning in human-centered environments, particularly navigation, is typically addressed from two complementary perspectives: collision-free motion planning and socially compliant navigation. Collision-free motion planning aims to avoid collisions with static and dynamic obstacles while handling uncertainty in both the environment and robot dynamics. Methods like Model Predictive Control (MPC) provide constraint guarantees, while Reinforcement Learning (RL)-based approaches focus on interaction rather than strict constraint satisfaction. However, these methods typically rely on a cost or reward function that prioritizes task completion, often neglecting how the robot's behavior is perceived by the surrounding humans. Socially compliant navigation emphasizes human-robot interaction, aiming for socially acceptable movements and interaction awareness, which can be achieved by learning the cost/reward function or directly the policy using Imitation Learning (IL). However, such approaches may lack safety guarantees. In addition to safety, interaction awareness, and social compliance, the ability to adapt to long-term environmental changes, such as structural modifications or evolving navigation patterns, is often overlooked, as addressing these other challenges is already highly complex.

This thesis thus explores the key limitations in existing local motion planning approaches and presents solutions to address these challenges. It develops methods that enable safe, interaction-aware, socially compliant, and adaptive local motion planning for robots operating in human-centered environments. To this end, this thesis presents an approach to represent human-robot interaction in an MPC framework, leveraging the Social Force Model (SFM) to model pedestrian responses (Chapter 2), develops safety filters for learning-based policies, such as those learned through IL (Chapter 3), and introduces a self-supervised continual learning framework for the online adaptation of policies learned from observations (Chapter 4).

We first examine the standard approach of splitting local motion planning into two subproblems: predicting human trajectories and solving constrained trajectory optimization to avoid collisions. While this approach provides safety guarantees and

can handle uncertainty through robust and stochastic formulations, it neglects the interactions between the robot and the humans. In Chapter 2, we address this by formulating an MPC problem in which the robot's action influences both its own state and the human states. Focusing on navigation among pedestrians, we leverage the interpretable and established SFM to model the human response dynamics to robot actions. By accounting for the robot's influence on pedestrian behavior during planning, we demonstrate that the robot can guide pedestrian behavior through a well-crafted cost function. However, this approach relies on expert knowledge to design a cost function that not only encourages desired behaviors but also prevents the exploitation of pedestrian cooperation.

Next, we focus on the limitations of learning-based approaches that can address social compliance, specifically IL. While IL enables the learning of socially compliant behaviors from demonstrations or observations, the resulting policy lacks formal safety guarantees. Safety filters based on, e.g., Control Barrier Functions (CBFs), adapt control inputs to ensure safety and can be combined with IL policies. While CBFs are an effective tool to certify safety, two challenges remain: constructing them for complex systems with input constraints and accounting for model uncertainties. In Chapter 3, we propose Robust Policy Control Barrier Functions, a method for constructing robust CBFs that guarantees safety under worst-case bounded disturbances through policy evaluation of any policy. Furthermore, we present a practical approximation and demonstrate its effectiveness in simulation and on a hardware quadcopter platform, treating model errors as disturbances.

So far, we have discussed interaction awareness and have presented methods that allow the use of potentially unsafe, socially compliant policies. Chapter 4, instead, focuses on adaptability, specifically in data-driven pedestrian prediction models, which are crucial for the decoupled prediction and planning approach. While existing models are trained offline on general datasets, they may not reflect the behavior of pedestrians in the robot's environment. To address this, we propose a self-supervised continual learning framework that refines models during deployment using online data from the robot's perception pipeline, preserving prior knowledge through regularization and selective retraining. Experiments show improved performance compared to naive online training. Although this chapter focuses on pedestrian prediction, the approach could extend to other methods applying learning-from-observations.

In conclusion, in this thesis, we first explore how interaction awareness can be integrated into the standard decoupled motion planning approach by leveraging the established SFM. We then develop components for a comprehensive framework to address the challenges in local motion planning for human-centered environments. This includes a practical approach for defining safety filters for learning-based policies, addressing the aspect of safety. Additionally, we introduce a self-supervised continual learning framework that addresses the aspect of adaptability, enabling robots to learn and adjust to new observations online. To integrate all four aspects, the developed safety filter could be combined with a socially compliant, interaction-

aware policy learned from observations, and adapted online using the self-supervised continual learning framework. This combination could pave the way for local motion planning that is interaction-aware, safe, socially compliant, and adaptive.

Samenvatting

Autonome mobiele robots, die in het verleden beperkt waren tot gestructureerde omgevingen zoals magazijnen, opereren nu in dynamische, mensgerichte omgevingen zoals ziekenhuizen, straten en huizen, waar interactie met mensen onvermijdelijk is. Deze verschuiving brengt aanzienlijke uitdagingen met zich mee voor de lokale bewegingsplanning van de robot. De robot moet navigeren door complexe omgevingen waarin menselijk gedrag moeilijk te voorspellen is en wordt beïnvloed door de acties van de robot. Daarnaast moeten zijn bewegingen de prioriteit geven aan veiligheid, zich aanpassen aan veranderende omgevingen en voldoen aan sociale normen. Het is daarom van essentieel belang dat lokale bewegingsplanners rekening houden met interactie, veiligheid, sociale acceptatie en aanpasbaarheid. Bewegingsplanning in mensgerichte omgevingen, met name navigatie, wordt doorgaans benaderd vanuit twee complementaire perspectieven: obstakel-ontwijkende bewegingsplanning en sociaal acceptabele navigatie. Obstakel-ontwijkende bewegingsplanning houdt rekening met onzekerheid in zowel de omgeving als de dynamiek van de robot om statische en dynamische obstakels te vermijden. Methoden zoals Model Predictive Control (MPC) bieden garanties met betrekking tot de opgelegde restricties, terwijl op Reinforcement Learning (RL) gebaseerde benaderingen zich richten op interactie in plaats van strikte naleving van restricties. Deze methoden vertrouwen echter meestal op een kost- of beloningsfunctie die gericht is op taakvoltooiing, waarbij vaak wordt verwaarloosd hoe het gedrag van de robot wordt waargenomen door de omringende mensen. Sociaal acceptabele navigatie legt de nadruk op mens-robot-interactie en streeft naar sociaal acceptabele bewegingen en interactiebewustzijn, wat kan worden bereikt door het leren van de kost-/beloningsfunctie of direct vanuit demonstraties door middel van Imitation Learning (IL). Dergelijke benaderingen kunnen echter een gebrek aan veiligheidsgaranties hebben. Naast veiligheid, interactiebewustzijn en sociale acceptatie wordt de aanpassing aan langdurige veranderingen in de omgeving, zoals structurele wijzigingen of veranderende navigatiepatronen, vaak over het hoofd gezien vanwege de complexiteit van het aanpakken van de andere aspecten.

Deze thesis onderzoekt daarom de belangrijkste limitaties in bestaande lokale bewegingsplanningsmethoden en presenteert oplossingen om deze uitdagingen aan te pakken. In deze thesis, zijn medthoden ontwikkeld die interactiebewuste, veilige, sociaal acceptabele en aanpasbare lokale bewegingsplanning mogelijk maken voor robots in mensgerichte omgevingen. Daartoe presenteert deze thesis een aanpak om mens-robot-interactie te vertegenwoordigen in een MPC-kader door gebruik te maken van het Social Force Model (SFM) om de reacties van voetgangers te modelleren (Hoofdstuk 2), ontwikkelt het veiligheidsfilters voor op leren gebaseerde strategieën, zoals die geleerd via IL (Hoofdstuk 3), en introduceert het een self-

supervised continual learning framework voor de online aanpassing van strategieën die zijn geleerd uit observaties (Hoofdstuk 4).

We onderzoeken eerst de standaardbenadering van het opsplitsen van lokale bewegingsplanning in twee subproblemen: het voorspellen van menselijke trajecten en het oplossen van geoptimaliseerde trajectbeperkingen om botsingen te vermijden. Hoewel deze aanpak veiligheidsgaranties biedt en onzekerheid kan verwerken via robuuste en stochastische formuleringen, negeert het de interacties tussen de robot en de mensen. In Hoofdstuk 2 pakken we dit aan door een MPC-probleem te formuleren waarin de acties van de robot zowel de eigen toestand als die van de mens beïnvloeden. Door te focussen op navigatie tussen voetgangers, maken we gebruik van het interpreteerbare en gevestigde SFM om de reacties van mensen op robot-acties te modelleren. Door tijdens de planning rekening te houden met de invloed van de robot op het gedrag van voetgangers, laten we zien dat de robot het gedrag van voetgangers kan sturen via een goed geformuleerde kostfunctie. Deze aanpak vertrouwt echter op expertkennis om een kostfunctie te ontwerpen die niet alleen gewenst gedrag stimuleert, maar ook voorkomt dat de wegwillendheid van voetgangers om samen te werken wordt misbruikt.

Vervolgens richten we ons op de beperkingen van geleerde benaderingen die sociale acceptatie kunnen aanpakken, met name IL. Hoewel IL het mogelijk maakt om sociaal acceptabel gedrag te leren uit tele-operaties of observaties, mist het resulterende veiligheidsgaranties. Veiligheidsfilters, geconstrueerd met behulp van Control Barrier Functions (CBFs), passen de besturingsinvoer aan om veiligheid te garanderen en kunnen worden gecombineerd met IL-strategieën. Hoewel CBFs effectief zijn voor het garanderen van veiligheid, blijven er twee uitdagingen bestaan: de constructie ervan voor complexe systemen met limieten op de invoer en het rekening houden met modelonzekerheden. In Hoofdstuk 3 introduceren we Robust Policy Control Barrier Functions, een methode om robuuste CBFs te construeren die veiligheid garanderen onder worst-case verstoringen, door de uitkomst van een willekeurig strategie te evalueren. Bovendien presenteren we een praktische benadering en demonstreren we de effectiviteit ervan in simulaties en op een quadcopter-platform, waarbij modelfouten als verstoringen worden behandeld.

In Hoofdstuk 2 and Hoofdstuk 3, hebben we interactiebewustzijn besproken en methoden gepresenteerd voor het waarborgen van veiligheid bij mogelijk onveilige, sociaal acceptabele strategieën. Hoofdstuk 4 richt zich in plaats daarvan op aanpasbaarheid, met name in data-gedreven voetgangersvoorspellingsmodellen, die cruciaal zijn voor de gescheiden voorspelling- en planningsbenadering. Terwijl bestaande modellen offline worden getraind op algemene datasets, kunnen ze mogelijk niet het gedrag van voetgangers in de specifieke omgeving van de robot weerspiegelen. Om dit aan te pakken, stellen we een *self-supervised continual learning framework* voor dat modellen verfijnt tijdens implementatie met online gegevens uit de perceptie-module van de robot, waarbij eerdere kennis behouden blijft via regularisatie en selectieve hertraining. Experimenten tonen verbeterde prestaties in vergelijking met naïeve online training. Hoewel dit hoofdstuk zich richt op voetgangersvoorspelling,

kan de aanpak worden uitgebreid naar andere methoden die die leren met behulp van observaties.

Tot slot hebben we in deze thesis eerst onderzocht hoe interactiebewustzijn kan worden geïntegreerd in de standaard gescheiden bewegingsplanningsbenadering door gebruik te maken van het gevestigde SFM. Vervolgens hebben we componenten ontwikkeld voor een kader om de uitdagingen in lokale bewegingsplanning in mensgerichte omgevingen aan te pakken. Dit omvat een praktische aanpak voor het definiëren van veiligheidsfilters voor op leren gebaseerde strategieën, waarmee het aspect van veiligheid wordt aangepakt. Daarnaast introduceert deze thesis een self-supervised continual learning framework dat het aspect van aanpasbaarheid aanpakt, waardoor robots online kunnen leren zich aan te passen aan nieuwe observaties.

Om alle vier de aspecten te integreren, an het ontwikkelde veiligheidsfilter gecombineerd worden met een sociaal acceptabel, interactiebewust strategie geleerd vanuit observaties via IL en online aangepast door een self-supervised continual learning framework. Deze combinatie maakt het mogelijk voor lokale bewegingsplanning om zowel interactiebewust, veilig sociaal acceptabel en aaanpasbaar te zijn.

Zusammenfassung

Mobile autonome Roboter, die zunächst vorwiegend in strukturierten Umgebungen wie Lagerhäusern eingesetzt wurden, operieren inzwischen vermehrt in dynamischen, menschenzentrierten Umgebungen wie Krankenhäusern, Straßen und Wohnungen. Hier sind Interaktionen mit Menschen unvermeidlich. Dieser Wandel bringt erhebliche Herausforderungen für die lokale Bewegungsplanung des Roboters mit sich. Der Roboter muss komplexe Umgebungen durchqueren, in denen das Verhalten der Menschen schwer vorhersehbar ist und durch das Verhalten des Roboters beeinflusst wird. Seine Bewegungen müssen sicher sein, sich an wechselnde Bedingungen anpassen und soziale Normen einhalten. Folglich müssen lokale Bewegungsplaner interaktionsbewusst, sicher, sozial angemessen und anpassungsfähig sein.

Die Bewegungsplanung, insbesondere die Navigation, in menschenzentrierten Umgebungen wird typischerweise aus zwei sich ergänzenden Perspektiven betrachtet: kollisionsfreie Bewegungsplanung und sozial angemessen Navigation. Die kollisionsfreie Bewegungsplanung hat das Ziel, statischen und dynamischen Hindernissen auszuweichen. Dabei werden Unsicherheiten in der Umgebung sowie in der Dynamik des Roboters berücksichtigt. Methoden wie Model Predictive Control (MPC) bieten Garantien für die Einhaltung von Zwangsbedingungen, e.g. keine Kollisionen, während Reinforcement Learning (RL) sich eher auf Interaktion als auf die strikte Einhaltung der Zwangsbedingungen fokussiert. Diese Methoden basieren in der Regel auf einer Kostenfunktion, die vorrangig darauf ausgelegt ist, die Aufgabe erfolgreich zu erfüllen, jedoch oft außer Acht lässt, wie das Verhalten des Roboters von Menschen wahrgenommen wird. Die sozial angemessene Navigation legt den Fokus auf die Mensch-Roboter-Interaktion mit dem Ziel, Bewegungen zu generieren, die von den Menschen akzeptiert werden und die Interaktionen zwischen Mensch und Roboter berücksichtigen. Dies kann erreicht werden, indem entweder die Kostenfunktion oder direkt die Steuerstrategie mittels Imitation Learning (IL) erlernt wird. Allerdings fehlen solchen Ansätzen oft formale Sicherheitsgarantien. Neben Sicherheit, Interaktionsbewusstsein und sozialer Verträglichkeit wird die Anpassung an Veränderungen in der Umgebung, wie zum Beispiel sich wandelndes menschliches Verhalten, aufgrund der Komplexität der anderen Aspekte häufig vernachlässigt.

Diese Dissertation untersucht daher die zentralen Schwächen bestehender lokaler Bewegungsplanungsansätze und stellt Lösungen zur Bewältigung dieser vor. Es werden Methoden entwickelt, die eine sichere, interaktionsbewusste, sozial angemessene und anpassungsfähige lokale Bewegungsplanung für Roboter in menschenzentrierten Umgebungen ermöglichen. Dazu präsentiert diese Arbeit einen MPC-basierten Ansatz, der mithilfe des Social Force Model (SFM) berücksichtigt, wie die Ak-

tionen des Roboters das Bewegungsverhalten der Fußgänger beeinflussen. (Kapitel 2), entwickelt einen Sicherheitsfilter für gelernte Steuerstrategien, (Kapitel 3), und präsentiert ein *self-supervised continual learning framework* das die fortlaufende Verbesserung von Strategien ermöglicht, die aus Beobachtungen gelernt wurden (Kapitel 4).

Zunächst betrachten wir den gängigen Ansatz, die lokale Bewegungsplanung in zwei Teilprobleme zu unterteilen: die Vorhersage des Bewegungsverhaltens der Menschen in der Umgebung und die Optimierung der Robotertrajektorie mit dem Ziel, Kollisionen zu vermeiden und weitere Zwangsbedingungen einzuhalten. Während dieser Ansatz Sicherheitsgarantien bietet und Unsicherheiten durch robuste und stochastische Formulierungen bewältigen kann, vernachlässigt er die Interaktion zwischen Roboter und Menschen. In Kapitel 2 gehen wir dieses Problem an, indem wir ein MPC-Problem formulieren, bei dem die Aktionen des Roboters sowohl den Roboterzustand als auch den Zustand der Menschen beeinflussen. Im Kontext der Navigation unter Fußgängern nutzen wir das etablierte und interpretierbare SFM, um die Reaktionen von Fußgängern auf Roboteraktionen zu modellieren. Indem wir den Einfluss des Roboters auf das Verhalten der Fußgänger in die Planung einbeziehen, zeigen wir, dass der Roboter das Fußgängerverhalten durch eine gut gestaltete Kostenfunktion steuern kann. Dieser Ansatz setzt jedoch Expertenwissen voraus, um eine Kostenfunktion zu entwerfen, die nicht nur gewünschte Verhaltensweisen fördert, sondern auch verhindert, dass die Kooperation der Fußgänger ausgenutzt wird.

Als Nächstes betrachten wir Einschränkungen daten-getriebener Ansätze wie IL. Zwar ermöglicht IL das Erlernen sozial verträglicher Verhaltensweisen aus Demonstrationen, doch fehlen den resultierenden Steuerstrategien formale Sicherheitsgarantien. Sicherheitsfilter, die beispielsweise mithilfe von Control Barrier Functions (CBFs) konstruiert werden, passen Steuerbefehle an, um Sicherheit zu gewährleisten, und können mit IL-basierten Steuerstrategien kombiniert werden. Während CBFs ein bewährtes Werkzeug sind, um Sicherheitsgarantien effektiv zu gewährleisten, gibt es zwei wesentliche Schwierigkeiten: die Konstruktion von CBFs für komplexe Systeme mit Eingangsbegrenzungen ist nicht trivial und die zuverlässige Behandlung von Modellunsicherheiten bleibt eine bedeutende Herausforderung. In Kapitel 3 schlagen wir *Robust Policy Control Barrier Functions* (RPCBs) vor, eine Methode zur Konstruktion robuster CBFs, die für begrenzte Störungen im schlimmsten Fall Sicherheit garantiert. Die RPCB wird durch die Bestimmung der schlimmsten Verletzung der Zwangsbedingungen unter einer beliebigen Steuerstrategie konstruiert. Darüber hinaus präsentieren wir eine RPCB Approximationen und demonstrieren deren Wirksamkeit in Simulationen und auf einer realen Quadcopter-Plattform, wobei Fehler im Dynamikmodell als Störungen behandelt werden.

Bis jetzt haben wir die Roboter-Mensch Interaktionen berücksichtigt und Methoden erarbeitet, die den Einsatz von Steuerstrategien ermöglichen, die zwar potenziell unsicher, aber sozial verträglich sind. Kapitel 4 konzentriert sich hingegen auf die fortlaufende Verbesserung, insbesondere von datenbasierten Vorhersagemodellen

des Fußgängerverhaltens. Diese Vorhersagemodelle sind für den entkoppelten Vorhersage- und Planungsansatz entscheidend. Während bestehende Modelle offline auf öffentlich zugänglichen Datensätzen trainiert werden, spiegeln diese möglicherweise nicht das Verhalten der Fußgänger in der Umgebung des Roboters wider. Um dieses Problem zu lösen, schlagen wir ein self-supervised continual learning framework vor, das Vorgersagemodelle während des Einsatzes des Roboters mithilfe von Sensordaten des Roboters verbessert. Dabei wird vorheriges Wissen durch Regularisierung und selektives Retraining bewahrt. Experimente zeigen eine verbesserte Performance im Vergleich zum naiven Online-Training. Obwohl sich dieses Kapitel auf die Vorhersage von Fußgängerverhalten konzentriert, könnte der Ansatz auch auf andere Methoden erweitert werden, die von Beobachtungen lernen.

In dieser Dissertation haben wir untersucht, wie Interaktionsbewusstsein in den standardmäßigen entkoppelten Bewegungsplanungsansatz integriert werden kann, indem das etablierte SFM genutzt wird. Anschließend entwickelten wir Komponenten für ein Framework, das die Herausforderungen der lokalen Bewegungsplanung in menschenzentrierten Umgebungen angeht. Dazu gehört ein praktischer Ansatz zur Definition von Sicherheitsfiltern für Steuerstrategien. Darüber hinaus wurde ein self-supervised continual learning framework eingeführt, das es Robotern erlaubt, ihr Verhalten online zu verbessern. Um alle vier Aspekte zu integrieren, könnte der entwickelte Sicherheitsfilter mit einer sozial verträglichen, interaktionsbewussten Steuerstrategie kombiniert werden, die mithilfe von IL aus Beobachtungen gelernt wurde und mithilfe des self-supervised continual learning frameworks online verbessert wird. Diese Kombination schafft die Grundlage für die Generierung lokaler Bewegungsstrategien, die interaktionsbewusst, sicher, sozial verträglich und anpassungsfähig sind.

1
Introduction

1

Autonomous robots are gradually evolving from operating in structured, controlled, and predictable environments like warehouses and factories to navigating unstructured, dynamic, and uncertain *human-centered* spaces like hospitals, restaurants, urban streets, and homes. Here, human-centered refers to environments specifically designed for humans. This shift is driven by the increasing demand for automation due to population growth, aging societies, labor shortages, and the need for greater efficiency and productivity [1]. For example, in hospitals, robots can transport medication, samples, and supplies, perform cleaning tasks, and handle repetitive laboratory duties, see Fig. 1.1a. Operating around the clock, they reduce the workload of healthcare professionals, thus improving the quality of care [2]. In urban environments, mobile robots, including autonomous vehicles, can offer on-demand services like transportation and delivery, enhancing urban mobility and accessibility. These advancements offer new opportunities but also bring significant technical challenges for robots, particularly in perception, localization, motion planning, and control. While all these aspects, accurate perception, localization, motion planning, and control, are essential, this thesis focuses specifically on motion planning and the unique challenges it entails.

1.1 Challenges in Human-Centered Environments

Motion planning involves determining a sequence of actions that allows a robot to move from its current pose to a desired goal pose while avoiding obstacles and meeting specific constraints. The motion planning problem can be broadly divided into two main components: *global motion planning*, which generates a rough path from the start to the goal considering static obstacles, and *local motion planning*, which focuses on following the path while dynamically avoiding obstacles. We will focus on the local motion planning problem, as it directly addresses how robots move in dynamic environments and respond to human movements. In the remainder of this thesis, we will refer to the local motion planning problem simply as motion planning and concentrate on the challenges it poses in human-centered environments

Unlike industrial settings, where robots operate in isolation under controlled conditions and can follow pre-programmed paths, human-centered environments demand more sophisticated motion planning capabilities. Adapting to these environments is particularly challenging due to the inherent uncertainty of human behavior, the often highly dynamic setting, and the requirement to provide some form of safety guarantees. Motion planning in human-centered environments finds itself at the intersection between research on robot motion planning and (human-centered) Human-Robot Interaction (HRI) [3]. While the former typically focuses on planning safe trajectories for robots to efficiently operate in the environment, neglecting how humans perceive their behavior, the latter focuses on operating in a for humans *acceptable and comfortable* manner [4].

Even without considering how robot behavior is perceived, collision avoidance remains a significant challenge. Robots must navigate complex, dynamic environments and generalize to unpredictable situations. To ensure **safe** and effective operation, they must reason quickly and adapt in real-time to changes in their surroundings [7]. In this context, the robot must account for the **interactions** with humans, recog-

(a) Mobile manipulator operating in a human-centered hospital laboratory [5].

(b) Pedestrians crossing the Shibuya Scramble Crossing in Japan [6], frequently referred to as the busiest pedestrian crossing in the world.

Figure 1.1: Examples for human-centered environments.

nizing that they are also decision-making entities with their own goals. Here, interactions refer to any change in the behavior of a human or a robot resulting from the presence of the other [3]. While similar challenges arise in scenarios where multiple robots operate or where humans enter environments designed for robotic tasks (*robot-centered*), the focus shifts significantly in environments designed for humans such as hospitals and urban streets, see Fig. 1.1b.

In human-centered environments, the interaction with humans becomes inevitable. Therefore, not only are guarantees on the robot's performance required, but the robot's acceptance by humans also becomes a key requirement. As discussed above, achieving pure functionality is already a complex challenge. The same holds for achieving acceptance, as it is challenging to specify quantitatively. For instance, in addition to formal safety guarantees, the robot's behavior must be perceived as safe, which is difficult to define precisely. Furthermore, the robot should adhere to explicit high-level cultural conventions, e.g., walking on the right side of a corridor, and move *naturally*, with low-level behavior patterns similar to human behavior [3]. For example, a mobile robot navigating through a crowded hospital entrance hall should maintain appropriate interpersonal distances, give the right of way, and avoid situations like moving between people engaged in a conversation. These social norms are not formally defined in some rulebook but are implicitly understood by humans. In the following, we will refer to behaviors that adhere to these goals as *socially compliant behaviors*. These **socially compliant** behaviors make HRI more "convenient" for humans [4].

While the robot's behavior is crucial, other factors like its appearance, voice, and other non-verbal cues also significantly impact HRI. However, since this thesis focuses on motion planning, we narrow our scope to the robot's motion as a key representation of desired behavior, precisely the motion required to complete the task, such as navigating from one location to another. This approach contrasts the broader focus typically found in the field of HRI, which also considers factors like head movements and other social cues.

Another challenge lies in the dynamic nature of human-centered environments.

1

As humans become more familiar with the robot, their behavior may evolve or vary depending on the environment or cultural context. This evolving behavior requires the robot to continuously **adapt** to changing conditions.

Despite the inherent challenges of motion planning in human-centered environments, robots are increasingly being deployed in roles such as delivering dishes in restaurants, guiding visitors in museums, cleaning the floors in supermarkets, or assisting in hospitals. However, their seamless integration into these settings remains in its early stages. Currently, most robots operate in predefined, controlled settings in human-centered environments, where they can follow pre-programmed paths and stop/ or slowly move out of the way when encountering humans. However, as robots move into more dynamic, unstructured environments, they must account for **interactions** with humans, **move safely** among humans in a **socially compliant** way, and must be able to **adapt** to changing conditions. Next, we describe methods that address aspects of motion planning in human-centered environments.

1.2 Local Motion Planning in Human-Centered Environments

Depending on the specific task, navigation, manipulation, or mobile manipulation, the requirements for motion planning vary based on the complexity of the associated configuration spaces. While some methods are versatile enough to apply across tasks, others are tailored primarily to a single type of task. The remainder of this thesis will focus on navigation tasks, in which a mobile robot needs to move efficiently between locations while avoiding collisions with obstacles and demonstrating socially compliant behaviors. As discussed earlier, motion planning in human-centered environments is typically addressed from two complementary perspectives: planning safe trajectories which focuses on collision avoidance and socially compliant motion planning which focuses on operating in a for humans acceptable manner. The following sections will review the related work in these areas and examine their respective limitations.

1.2.1 Collision-Free Motion Planning

Various methods have been proposed to tackle the challenge of collision avoidance in human-centered environments. These approaches can be broadly classified into two main categories: *reactive* methods and *predictive* methods.

Reactive methods focus on real-time decision making based on the robot's current perception of the environment, typically with a short planning horizon limited to the immediate next step. Examples include the Dynamic Window Approach [8], Velocity Obstacles [9, 10], Elastic Bands [11], Potential Fields [12], and Social Forces [13]. These methods are computationally efficient and can handle dynamic environments. However, they lack long-term planning and may lead to suboptimal trajectories, especially in complex scenarios with multiple agents, i.e., robots or humans.

To address these limitations, predictive methods incorporate a longer planning horizon which enables more informed decision-making, improving trajectory smoothness and safety. A common strategy in such environments is to decompose the

motion planning problem into two disconnected steps: a prediction step and a planning step. The prediction step relies on models to anticipate the future behavior of the surrounding humans over the planning horizon. In contrast, the planning step generates a trajectory that avoids the predicted human trajectories. The primary challenge here stems from the inherent uncertainty in predicting human behavior, both in forecasting their future trajectories and in planning robust trajectories that account for this uncertainty. This motion planning approach paves the way for research directions focused on enhancing prediction models and planning trajectories that account for the uncertainty in these predictions [14]. Prediction models are typically data-driven, designed to learn patterns and accurately predict the future trajectories of humans, e.g., pedestrians or human-driven vehicles, often represented as a distribution [15]. The research focuses on improving the accuracy of these models while accounting for the multimodal nature of human behavior. However, one key limitation is their lack of **adaptability** over time: these models are typically trained offline on standard datasets and struggle to adapt to new or unseen behaviors and environments. For instance, environments such as hospitals, rarely included in training datasets, present a challenge, as the models have difficulty generalizing effectively in such scenarios. This thesis addresses adaptability by continuously updating pedestrian prediction models based on observations of the surrounding pedestrians' behavior. The adaptation is achieved using methods from Continual Learning (CL), which are designed to handle learning in scenarios where the data distribution evolves over time [16]. Based on the predictions, a trajectory is planned.

The planning step is typically formulated as an optimal control problem over a finite horizon, where a cost function is minimized subject to constraints like collision avoidance and system dynamics. The cost function encodes the desired behavior. We refer to this in the following as decoupled trajectory optimization. Model Predictive Control (MPC), also known as receding horizon control, is a widely used framework for solving this problem by repeatedly optimizing a control sequence over the planning horizon and executing only the first control input at each step. The MPC problem can be solved using gradient-based optimization [17–19] or sampling-based methods, such as Model Predictive Path Integral (MPPI) control [20–22]. However, splitting motion planning into a prediction and a planning step does not account for dynamic interactions between the robot and the humans. This can result in overly defensive and opaque behaviors [23]. These behaviors arise because the actual decision-making processes of humans and robots are inherently coupled. For instance, if a robot yields the right of way to a human, the human may adjust their behavior accordingly, leading to a different outcome than if the robot had not yielded. Therefore, it is crucial to account for these interactions to ensure more effective and predictable behaviors.

To better model the **interactions** between robots and humans, an alternative approach assumes that humans act rationally to minimize a defined cost function rather than relying on open-loop or feedback behavior models. Dynamic noncooperative game theory [24] presents a mathematical formulation for modeling interactions between rational, self-interested agents, but it is computationally expensive, especially in real-time applications. A practical approach that incorporates game-

1

theoretic principles is model-predictive game-play, where the robot plans its future trajectory in equilibrium with the anticipated trajectories of other agents in a receding horizon fashion. However, despite the receding horizon, it remains computationally demanding. Similarly, multi-agent optimization [25] can model interactions between the robot and humans by optimizing a shared objective function. However, these methods remain computationally intensive. A limitation of both approaches is the assumption that the cost functions of other agents are known.

Other approaches implicitly account for interactions between agents and the future effect of their actions on the environment. For instance, in a Reinforcement Learning (RL) setting, the robot learns a *policy*—a decision-making strategy, through interacting with the environment. By taking actions and receiving rewards based on the outcomes, the robot adapts its behavior. In contrast, Imitation Learning (IL) focuses on imitating expert, i.e., human, behavior, inherently capturing the interactions between decision-making agents. RL has shown impressive results in various domains such as playing go, chess, or video games [26–28]. However, these environments do not have an embodied agent, i.e., a robot, interacting with the environment. In these settings, making a wrong move might only result in losing the game. In robotics applications, it can have far more serious consequences. While RL has been applied to motion planning in robotics applications [29–31], several challenges remain. RL, though computationally efficient during runtime, typically requires extensive interactions with the environment during training. On one side, training in simulation requires highly accurate representations of the real-world environment, including human behavior. Providing such a highly accurate simulation environment is particularly challenging, as modeling human behavior, as described before, is a significant area of research itself. Additionally, the ability to transfer the learned policy to the real world is largely determined by the accuracy of the simulation. Conversely, training in the real world is constrained by the large number of required interactions with the environment that can strain hardware and the absence of effective ways to integrate safety constraints during training. These issues contribute to the difficulty in ensuring robust and reliable performance in real-world applications. Furthermore, the resulting policy lacks formal safety guarantees, whether training is done in simulation or the real world. IL, which is described in more detail in the next section, also suffers from a lack of safety guarantees. This is a main limitation of learning-based methods, as they often cannot provide formal safety guarantees, limiting their applicability in safety-critical applications.

In human-centered environments, providing **safety** guarantees is crucial. However, in real-world applications, robots often encounter the challenge of making decisions with incomplete knowledge of the environment or under conditions of uncertainty. As a result, ensuring formal safety guarantees becomes challenging, as the robot may not have access to all necessary information to accurately predict and avoid potential risks. Thus, it has to be noted that guarantees can only be established based on the assumptions and structure outlined in the problem formulation. While learning-based methods can address the aspects of adaptability and interaction awareness, they lack safety guarantees. To address this challenge, Safety Filters (SFs) [32] and shielding [33] have been developed to make minimal adjustments to

the learned control input if future constraint satisfaction cannot be guaranteed. SFs can be defined either by leveraging MPC techniques [34] or by utilizing the concept of *control invariant* sets. Control invariant sets refer to regions of the state space where, if the system starts within the set, a feasible control input will always exist to keep the system within the set. While MPC-based SFs have the advantage of not requiring any additional formulation, as the constraints are already embedded in the problem setup, they are limited by considering only a finite horizon or requiring a control invariant terminal set. Additionally, they result in a significant computational increase, as the MPC problem must be solved at each timestep. Furthermore, the problem may become infeasible if the initial state violates the constraints or if the assumptions underlying the MPC formulation are incorrect. Control invariant sets, on the other side, can be defined using either Control Barrier Functions (CBFs) or Hamilton-Jacobi Reachability Analysis (HJRA), each with its own strengths and limitations. The SF based on CBFs nicely trades off between safety and performance, but there is no systematic method to construct CBFs, especially for systems with input constraints. In contrast, HJRA provides a general approach to constructing safe sets and can recover the maximal safe set. However, it suffers from the *curse of dimensionality*, becoming computationally infeasible as the state space dimension increases. Overall, the state-of-the-art methods for safety assurance have significant limitations, particularly when applied to complex, high-dimensional systems with input constraints. This thesis will introduce a practical approach to constructing CBFs that addresses one of their key limitations.

1.2.2 Socially-Compliant Motion Planning

Most of the above-mentioned methods require defining the desired behavior through a cost or reward function. In industrial settings, defining the cost/reward function based on metrics such as time, energy consumption, and distance traveled is often sufficient. However, in human-centered environments, the desired **socially compliant** behavior is more challenging to define due to the complexity and variability of social norms. Assuming that humans can demonstrate socially compliant behaviors or that such behaviors can be observed, various methods have been developed to either infer the underlying cost or reward function that captures human preferences or directly learn a policy that replicates the observed behavior.

IL [35, 36] offers an intuitive way to define and transfer behaviors, making it accessible even to non-experts in robotics. Similar to how humans learn through demonstrations, corrections, or feedback, robots imitate demonstrated actions, bypassing the challenge of designing an appropriate cost/reward function. Demonstrations can come from a human operator via teleoperation, physical guidance, or by observing human behavior. One approach involves learning the cost/reward function that best explains the observed behavior using Inverse Reinforcement Learning (IRL) [37]. The reward function is commonly represented as either a linear combination of reward features or through neural network architectures. While the learned reward function provides a valuable basis even when the agent's specifications change slightly, many expert demonstrations are required to learn the reward function accurately. Furthermore, the reward function is often not unique. Addi-

1

tionally, if the reward function is represented by a linear combination of features, the features must be carefully selected to ensure the reward function is expressive enough to capture the desired behavior.

An alternative approach is thus to directly learn a policy that replicates the desired behavior, achieved through e.g. Behavior Cloning (BC) [38]. BC is a straightforward approach where the robot learns a mapping from states to actions based on expert demonstrations. However, it suffers from the problem of covariate shift, where the training and test distributions differ, leading to poor generalization. Furthermore, BC shares limitations with RL, such as the lack of safety guarantees and reliance on the quality and quantity of demonstrations. Safety concerns can be addressed using methods discussed earlier, and recent research focuses on making BC methods more data-efficient and robust [39, 40].

Learning-based methods, such as those described above, play a crucial role in achieving desired behaviors, but they also raise important concerns about generalization and safety. A fundamental challenge, therefore, is determining how to apply such methods while ensuring formal safety guarantees in complex, dynamic settings.

1.3 This Thesis: Contributions and Outline

The previous sections highlighted the challenges in motion planning for autonomous robots in human-centered environments and the limitations of existing approaches. The challenges mainly arise from the complexity and uncertainty of human behavior and the highly dynamic nature of the environment. The limitations of existing methods stem from the difficulty in accounting for **interactions** to enable more efficient and predictable solutions, in ensuring **safety**, in providing **socially compliant** behaviors to enhance the robot's acceptance, and in demonstrating **adaptability** to changes in the environment. Notably, some aspects, such as safety, are more critical for the actual deployment of robots in human-centered environments. In contrast, others, such as social compliance, are key to the robot's acceptance and integration into these settings. However, they are not mutually exclusive as accounting for interactions can improve safety and social compliance, and adaptability can enhance the robot's ability to handle uncertainty and dynamic changes. While various methods have been proposed to address interaction-awareness, safety, social compliance, and adaptability, none fully resolve them. Furthermore, many struggle to be applied to more complex or uncertain systems or require domain-specific knowledge. This thesis aims to address the key challenges of motion planning in human-centered environments by enhancing existing methods and presenting approaches that simultaneously tackle multiple aspects. To this end, the goal of this thesis is:

> To develop methods that enable safe, interaction-aware, socially compliant, and adaptive local motion planning for robots operating in human-centered environments.

Addressing this goal resulted in several scientific contributions, which will be discussed next.

1.3.1 Thesis Contributions

We first consider the widely used approach of dividing motion planning into a separate prediction step and planning step, resulting in a decoupled optimization problem, and focus on tackling the challenge of generating interaction-aware motion plans. This leads to the following contribution:

> **Social-Forces-Informed Interaction-Aware Model Predictive Control.** In Chapter 2, we formulate the interactions as an underactuated system and propose to leverage the Social Force Model (SFM) as the pedestrians' response dynamics. The SFM is a well-established model widely used for describing pedestrian behaviors. We thus preserve the simplicity of the approach and do not require a learning-based model.

While formulating interactions as an underactuated system enables the robot to influence other agents in a targeted manner and facilitates active information gathering, it still requires specifying the desired behavior through a well-designed cost function. As mentioned earlier, learning-based methods can be used to develop policies that produce desired behaviors, but they lack formal safety guarantees. To address this challenge, we propose a method for constructing SFs that address the aspect of safety while allowing the use of learning-based approaches or other potentially unsafe policies.

> **Robust Policy Control Barrier Functions.** In Chapter 3, we propose the concept of Robust Policy CBFs (RPCBFs), a practical method for constructing CBF approximations that is easy to implement and yields to improved safety compared to existing methods.

Next, we move towards the challenge of adaptability. As the environment the robot is in might change over time, the robot must be able to adapt to these changes. We specifically focus on the adaptability of pedestrian prediction models. However, the framework could also be applied in a learning-from-observations setting. The contribution of Chapter 4 is as follows:

> **A Self-Supervised Continual Learning framework.** We introduce a self-supervised continual learning framework to improve data-driven pedestrian prediction models online utilizing the continuous stream of pedestrian observations from the robot's perception pipeline. This framework allows the prediction model to adapt to behaviors and environments not seen in the initial training set.

With these contributions in mind, the outline below details the structure of this thesis.

1.3.2 Thesis Outline

An overview of the thesis structure, highlighting the challenges addressed in each chapter, is provided in Fig. 1.2. We begin by introducing the interaction-aware motion planner in the context of decoupled planning methods in Chapter 2. Next, in

1

Chapter 3, we address the critical aspect of safety. Specifically, we focus on leveraging CBFs to formulate safety through set invariance. We propose a method for constructing SFs that are robust to model errors and disturbances, utilizing RPCBFs. In Chapter 4, we address the aspect of adaptability in the setting of pedestrian prediction models by developing a self-supervised continual learning framework. Finally, Chapter 5 summarizes and discusses the thesis contributions and outlines potential directions for future work.

In Appendix A, we present additional works, including papers published based on master's theses of supervised students, as well as a paper that, while related to this thesis, does not fully align with its main narrative. Appendix A.1 presents a framework for simultaneously learning and verifying neural CBFs to address safety. Appendix A.2 explores an alternative approach to safety through compliant control, which enables the robot to adapt to interactions and mitigate collision impacts in dynamic environments. Finally, Appendix A.3 addresses the challenge of motion planning for multi-robot systems in close proximity. While this work focuses on multi-robot systems rather than HRI and does not fully align with the thesis narrative, it still provides valuable insights on motion planning among decision-making agents.
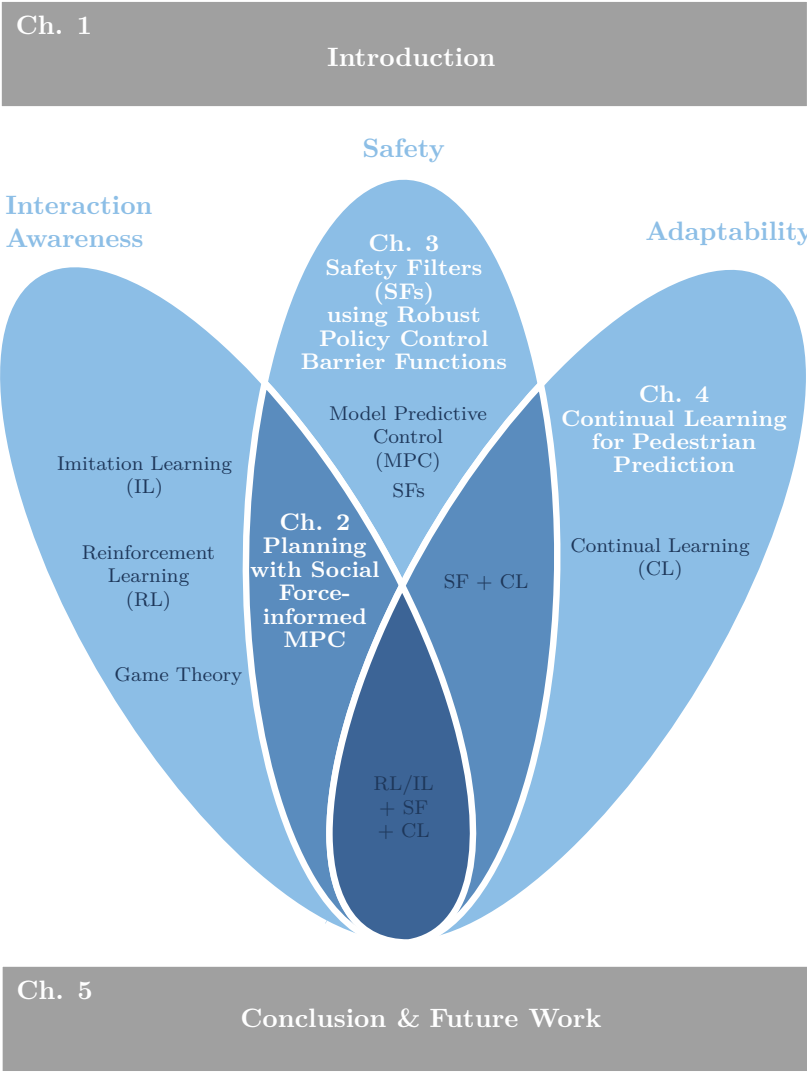
Figure 1.2: This thesis focuses on three key aspects of local motion planning in human-centered environments: interaction awareness, safety, and adaptability. Each chapter addresses one or a combination of these aspects.

2

Interaction-Aware Autonomous Navigation Among Pedestrians Using Social Forces

In the Introduction, we discussed the challenges of local motion planning in human-centered environments. Traditional decoupled trajectory optimization approaches to navigation assume that the pedestrian behavior is independent of the robot's actions. However, this assumption neglects the interactions between the robot and pedestrians, often resulting in overly conservative or inefficient navigation, particularly in crowded settings.

To address this, we model interactions as an underactuated dynamical system and use the interpretable Social Force Model to capture the pedestrian response dynamics. By using a Model Predictive Control framework, formal safety guarantees can be provided. This formulation allows the robot to effectively leverage agent behaviors and, with a suitable cost function, plan, e.g., motions that minimize its impact on surrounding pedestrians.

This chapter is a copy of the workshop paper:
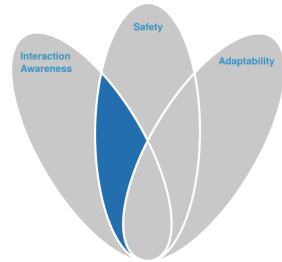
2.1 Motivation and Related Work

The rise of autonomous mobile robots in our daily environments makes it essential to ensure their safe and efficient interaction with pedestrians. This necessitates that robots can accurately predict the behavior of the surrounding pedestrians. In recent years, prediction models based on various deep neural network architectures have demonstrated notable progress in both prediction accuracy and scalability [41–43]. Traditionally, these prediction models are used to forecast the future trajectories of surrounding pedestrians. Subsequently, a planning step follows, during which the robot plans its trajectory in response to these predictions. This invariably results in the pedestrians being solely treated as moving obstacles, fostering a one-way interaction where only the robot adjusts its behavior. This can give rise to overly conservative or opaque navigation behaviors, see the left column in Fig. 2.1, and in dense crowds, it may lead to the so-called Freezing Robot Problem (FRP) [44], even when perfect predictions are considered. Hence, performing coupled prediction and planning and, thus, joint collision avoidance is crucial for realizing decision-making that is more interactive and akin to human behavior.
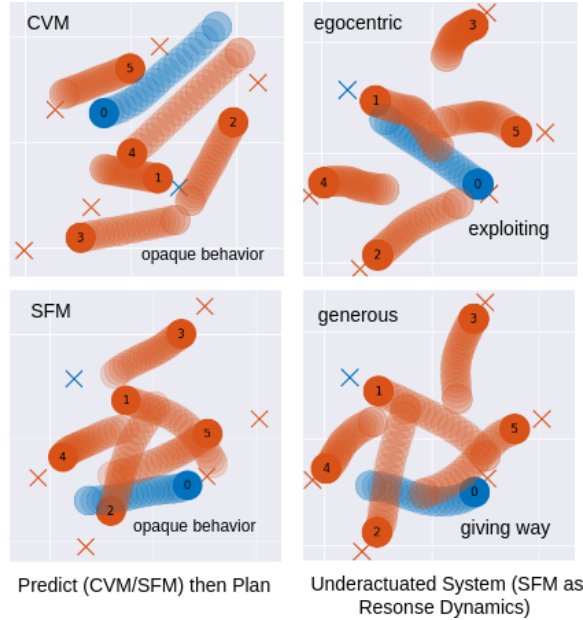
Figure 2.1: Robot (blue) navigating among pedestrians (orange). Crosses represent the goals, respectively. Transparent circles indicate the future plan for the robot and the predicted behavior of the pedestrians.

An important body of work addresses multi-agent interactions from a game-theoretic perspective, specifically using general sum dynamic games [45, 46]. Since each agent's action depends on the decisions of the others, solving games presents a considerable computational challenge, particularly with an increasing number of

agents. The computational complexity of these models imposes constraints on their applicability. The authors in [47] apply Model Predictive Path Integral (MPPI), a parallelizable sampling-based Model Predictive Control (MPC) algorithm, assuming knowledge of the other agents' objective functions and predicting their goals using the Constant Velocity Model (CVM). Nevertheless, the computational complexity scales linearly with the number of agents if a constant number of samples is assumed while the sample efficiency decreases.

A different approach is presented by [48], which bridges the gap between the use of prediction models discussed previously and coupled planning. They utilize predictions as an initial guess and incorporate an objective function to encourage proximity to these predictions. To reduce the computational complexity, [49] formulate the interaction between human-driven vehicles and autonomous vehicles as an underactuated dynamical system, meaning that the robot directly influences its own state and indirectly the state of the humans. The dynamics model thus includes the interaction dynamics between the agents. However, evaluating the dynamics model still requires solving for the optimal human response. Additionally, it requires the identification of the human objective function.

To address these issues, rather than estimating objectives explicitly and online, [50] build on top of the pedestrian prediction literature and learn an interactive multi-agent prediction policy. Using the policy they formulate the multi-agent motion planning problem as an optimization problem over only the ego agent's action sequence.

Instead of learning a multi-agent motion policy, we propose to use the Social Force Model (SFM) [51]. The SFM is a widely used and well-established model for describing the motion of pedestrians, e.g. in simulations for benchmarking [52]. It offers several advantages: it provides a well-established and interpretable framework for modeling pedestrian interactions, it does not require training and therefore does not rely on available data of the considered context, and it can be easily adapted to different environments. We solve the multi-agent underactuated motion planning problem using MPPI.

2.2 Problem Formulation

2.2.1 Preliminaries

Social Force Model (SFM)

The SFM [51] is a widely used and well-established model for describing the motion of pedestrians in 2D. It considers three main effects that determine the motion of a pedestrian i: the attraction towards their destination $\mathbf{f}_{\text{dest}}^i$, the repulsive effects of static obstacles $\mathbf{f}_{\text{static}}^i$, and the repulsive effects of other agents $\mathbf{f}_{\text{dyn}}^i$. At time t, the SFM describes the change in velocity of pedestrian i, that is, $\frac{d\mathbf{v}^i}{dt}$, as the result of a composition of social forces:

$$\frac{d\mathbf{v}^i}{dt} = \mathbf{a}^i = \begin{bmatrix} a_x^i \\ a_y^i \end{bmatrix} = p_{\text{dest}}\mathbf{f}_{\text{dest}}^i + p_{\text{static}}\mathbf{f}_{\text{static}}^i + p_{\text{dyn}}\mathbf{f}_{\text{dyn}}^i,$$

where p_{dest}, p_{static}, and p_{dynamic} are weighting parameters implemented according to PedSim[1]. We make use of the SFM formulation presented in [53]. When a pedestrian encounters no disruptions, they move from their current position $\mathbf{p}$ towards their goal position $\mathbf{p}_g$ at a specific desired speed v^{des}. The destination force $\mathbf{f}_{\text{dest}}^i$ is determined according to

$$\mathbf{f}_{\text{dest}}^i = \frac{1}{\tau}(v^{\text{des}}\mathbf{e}^0 - \mathbf{v}),$$

where $\mathbf{e}^0 = (\mathbf{p}_g^i - \mathbf{p}^i)/||\mathbf{p}_g^i - \mathbf{p}^i||$ is the desired direction of motion, $\mathbf{v}$ is the current velocity, and τ is a relaxation time. The static obstacle repulsive force $\mathbf{f}_{\text{static}}^i$ is defined by:

$$\mathbf{f}_{\text{static}}^i = ae^{-\text{d}^i/\text{b}}\mathbf{e}^i,$$

where d^i is the orthogonal distance of the i-th pedestrian to the obstacle, $\mathbf{e}^i$ is the unit vector pointing from the obstacle to the pedestrian, and a, b are scalar parameters. The pedestrian interaction force $\mathbf{f}_{\text{dyn}}^i$ is given by

$$\mathbf{f}_{\text{dyn}}^i = \sum_{j=0, j\neq i}^{N} \mathbf{f}_{\text{dyn,j}}^i, \tag{2.1}$$

$$\mathbf{f}_{\text{dyn,j}}^i = -Ae^{-d^{ij}/B}\left[e^{-(n'B\theta^{ij})^2}\mathbf{t}^{ij} + e^{-(nB\theta^{ij})^2}\mathbf{n}\right], \tag{2.2}$$

where d^{ij} denotes the distance between two pedestrians i and j, and θ^{ij} denotes the angle between the interaction direction $\mathbf{t}^{ij}$ and the vector pointing from pedestrian i to pedestrian j. They are defined as follows:

$$d^{ij} = ||\mathbf{d}^{ij}|| = ||\mathbf{p}^j - \mathbf{p}^i||, \tag{2.3}$$

$$\mathbf{D}^{ij} = \lambda(\mathbf{v}^i - \mathbf{v}^j) + \mathbf{d}^{ij}/d^{ij}, \tag{2.4}$$

$$\mathbf{t}^{ij} = \mathbf{D}^{ij}/||\mathbf{D}^{ij}||, \tag{2.5}$$

$$B = \gamma||\mathbf{D}^{ij}||, \tag{2.6}$$

where A, γ, n, n', and λ are model scalar parameters.

In this work, we make use of the SFM to represent the response dynamics of pedestrians. We choose the SFM parameters according to [53] which are summarized in Tab. 2.1. We do not consider static obstacles; however, they can be easily incorporated if needed.

Model Predictive Path Integral Control Algorithm

MPPI can solve optimal control problems for discrete-time dynamical systems $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \tilde{\mathbf{u}}_k)$ with state $\mathbf{x}$, time step k, and noisy input $\tilde{\mathbf{u}}$ with variance $\mathbf{\Sigma}$ and mean $\mathbf{u}$. The mean input will be provided to the system. The algorithm generates M input sequence samples $\tilde{U}_m$ with $m \in [1, M]$ from a distribution $\mathcal{N}(\mathbf{u}_k, \nu\mathbf{\Sigma})$ over a horizon K with ν being a scaling factor. Using $\tilde{U}_m$ and the dynamics model, the state sequence is generated over a horizon K. For each sample, the cost consisting

---

1   https://github.com/srl-freiburg/pedsim_ros/tree/master

of a stage and a terminal cost is computed. The input sequence U^*, which approximates the optimal control input sequence, is computed using importance sampling. For more information, we refer to [47, 54].

2.2.2 Problem Statement

We consider a scenario where a mobile robot, must navigate from an initial position $\mathbf{p}_0$ to a goal position $\mathbf{p}_g$ in the $\mathbb{R}^2$ plane populated by N pedestrians, each also navigating towards its respective goal position. The robot's physical state at time step k is denoted as $\mathbf{x}_k^0 \in \mathbb{X}^R$, while each pedestrian's respective physical state is denoted by $\mathbf{x}_k^i \in \mathbb{X}^H$ for $i \in \{1, \ldots, N\}$.

Notation: We omit the indice when referring to the collection of variables over the indice. For example, $\mathbf{x}_k = (\mathbf{x}_k^0, \mathbf{x}_k^1, \ldots, \mathbf{x}_k^N)$ denotes the joint state of all agents at time step k and $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K)$ the sequence of joint states over a time horizon which spans K discrete steps. We use the superscript $\neg i$ to denote a collection of variables over all agents *except* agent i.

At every time step k, each agent influences the next joint state by applying the control input $\mathbf{u}_k^i$, respectively. We refer to the joint control input at time step k as $\mathbf{u}_k$. We assume that the joint state $\mathbf{x}_k$ evolves dynamically according to the discrete-time dynamics

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k).$$

We seek to solve a motion planning problem for an underactuated system:

$$\hat{\mathbf{u}}^0 = \arg\min_{\mathbf{u}^0} \sum_{k=0}^{K-1} c_k^0(\mathbf{x}_k, \mathbf{u}_k^0) + c_K^0(\mathbf{x}_K) \tag{2.8}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \tag{2.7a}$$

$$\mathbf{x}_0 = x_{\text{init}}, \tag{2.7b}$$

$$\mathbf{u}_k = [\mathbf{u}_k^0, \hat{\mathbf{u}}_k^1, \cdots, \hat{\mathbf{u}}_k^N]^T, \tag{2.7c}$$

$$\hat{\mathbf{u}}_k^i = \arg\min_{\mathbf{u}_k^i} c_k^i(\mathbf{x}_k, \mathbf{u}_k^i), \tag{2.7d}$$

$$g^i(\mathbf{x}_k) \leq 0, \tag{2.7e}$$

$$\forall i \in \{1, \ldots, N\}, \forall k \in \{0, \ldots, K-1\}, \tag{2.7f}$$

where $\mathbf{x}_0$ denotes the joint state at the current time $k = 0$ and $\mathbf{u}^0 = (\mathbf{u}_0^0, \ldots, \mathbf{u}_{K-1}^0)$ is the sequence of robot control inputs. Collision avoidance constraints are imposed by (2.7f). With $\mathbf{u}^0$ the robot directly controls its state and indirectly influences $\mathbf{x}^{\neg 0}$ through (2.7d). The pedestrians' plans become a function of the robot's input. In contrast to formulating the interactions as a joint optimization, each pedestrian computes their best response to the other agents instead of trying to influence them. Note, that this formulation assumes that the pedestrians can estimate the robot's future states.

While the cost c_k^0 is a design parameter, the cost functions c_k^i of the pedestrians are typically unknown. In this work, we propose to use the well-known SFM to provide analytical response dynamics instead of solving for the optimal pedestrian

response $\tilde{\mathbf{u}}_k^i$. We consider a second-order point mass model for the dynamics of both the robot and the pedestrians:

$$\begin{bmatrix} \dot{\mathbf{p}}^i \\ \dot{\mathbf{v}}^i \end{bmatrix} = \begin{bmatrix} \mathbf{v}^i \\ \mathbf{a}^i \end{bmatrix}.$$

2.3 Social-Forces-Informed Interaction-Aware Model Predictive Control

In this section, we introduce the **So**cial-**F**orces-**I**nformed **I**nteraction-**A**ware **M**odel **P**redictive **C**ontrol (SoFIIA-MPC) framework. Instead of solving for the optimal pedestrian response at each iteration like [49], we make use of a response policy that implicitly encodes the pedestrians' cost function. Contrary to [50], we do not learn a policy of the other agents but make use of the well-established SFM.

2.3.1 Overview

We assume that a parameterized approximation $\hat{\pi}_\theta^i(\mathbf{x}_k) = \arg\min_{\mathbf{u}_k^i} c_k^i(\mathbf{x}_k, \mathbf{u}_k^i)$ of the pedestrians' response dynamics exists. This reduces the general multi-agent interaction problem to a single-agent optimization problem:

$$\hat{\mathbf{u}}^0 = \arg\min_{\mathbf{u}^0} \sum_{k=0}^{K-1} c_k^0(\mathbf{x}_k, \mathbf{u}_k^0) + c_K^0(\mathbf{x}_K) \tag{2.8}$$

$$\text{s.t. } \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \tag{2.8a}$$

$$\mathbf{x}_0 = x_{\text{init}}, \tag{2.8b}$$

$$\mathbf{u}_k = [\mathbf{u}_k^0, \tilde{\mathbf{u}}_k^1, \cdots, \tilde{\mathbf{u}}_k^N]^T, \tag{2.8c}$$

$$\tilde{\mathbf{u}}_k^i = \hat{\pi}_{\theta^i}(\mathbf{x}_k), \tag{2.8d}$$

$$g_k^i(\mathbf{x}_k) \leq 0, \tag{2.8e}$$

$$\forall i \in \{1, \dots, N\}, \forall k \in \{0, \dots, K-1\}. \tag{2.8f}$$

We approximate $\pi_\theta^i(\mathbf{x}_k)$ using the SFM resulting in

$$\hat{\pi}_\theta^i(\mathbf{x}_k) = p_{\text{dest}} \mathbf{f}_{\text{dest}}^i(\mathbf{x}_k) + p_{\text{static}} \mathbf{f}_{\text{static}}^i(\mathbf{x}_k) + p_{\text{dyn}} \mathbf{f}_{\text{dyn}}^i(\mathbf{x}_k).$$

2.3.2 Cost Function

We consider two cost function designs: one part addresses the costs related to the ego agent c_{ego}, and the other part aims to influence the behavior of other agents c_{affect}. We design c_{ego} to encourage the robot to reach its goal, to avoid collisions, and to maintain a velocity limit. The cost terms are defined as follows:

$$c_{\text{ego}} = c_{\text{goal},0} + c_{\text{vel-limit}} + c_{\text{collision}}, \tag{2.9}$$

$$c_{\text{affect}} = (w_{ego} c_{\text{goal},0} + w_{others} c_{\text{goal}\neg\text{i}})/N \tag{2.10}$$

$$+ c_{\text{vel-limit}} + c_{\text{collision}}.$$

Collisions between the robot and a pedestrian and velocities higher than a maximum velocity $v_{\max}$ are penalized with a constant cost, i.e. $C_{\text{collision}}$ and $C_{\text{vel}-\text{limit}}$:

$$c_{\text{collision}} = C_{\text{collision}} \sum_{i=0}^{N} \mathbf{1}(d^{0i} \leq r^0 + r^i), \tag{2.11}$$

$$c_{\text{vel}-\text{limit}} = C_{\text{vel}-\text{limit}} \mathbf{1}(||\mathbf{v}|| > v_{\max}). \tag{2.12}$$

The indicator function $\mathbf{1}(\cdot)$ returns 1 if the condition inside the parentheses is true, and 0 otherwise. The radii of the robot and the pedestrian i are given by r^0 and r^i, respectively. The goal cost of agent i is defined as

$$c_{\text{goal,i}} = ||\mathbf{p}_k^i - \mathbf{p}_g^i|| / ||\mathbf{p}_0^i - \mathbf{p}_g^i||, \tag{2.13}$$

and the goal cost of the other agents $\neg i$ is defined as

$$c_{\text{goal}\neg\text{i}} = \sum_{j \neq i}^{N} c_{\text{goal,j}}. \tag{2.14}$$

2.4 Results

We consider two versions of our planner: SoFIIA-MPC using c_{ego} and SoFIIA-MPC-affect using c_{affect}. Specifically, we consider a case with $w_{ego} = 0.8$ and $w_{others} = 1$. In this section, we compare our planners with the following baselines:

1. **MPC-CVM**: Predict-then-Plan approach. Assumes that the pedestrians continue moving with their current velocities and uses c_{ego},

2. **MPC-SFM**: Predict-then-Plan approach. The behavior of the pedestrians is predicted using the SFM assuming that the robot also follows the SFM, uses c_{ego}.

We evaluate the different planners in simulation, with the pedestrian behavior modeled using the SFM. The SFM parameters are chosen according to [53] and are depicted in Tab. 2.1. While the current evaluations do not provide insights into how well SoFIIA performs in environments with real pedestrians, these experiments offer valuable insights into the benefits of accounting for interactions in planning. MPC-SFM incorporates the true model for the simulated pedestrians. However, an assumption about the robot's future behavior has to be made to predict the pedestrians' future behaviors. To generate the SFM predictions, we assume that the robot

Table 2.1: Parameters

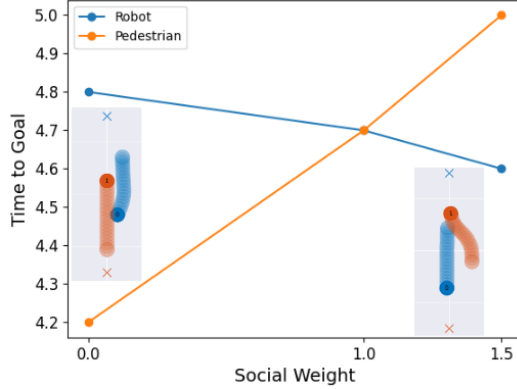| SFM A	4.5	SFM τ	0.54
SFM γ	0.35	Horizon K	20
SFM n	2.0	Radius agents	0.3 m
SFM n'	3.0	Time step Δt	0.1 s
SFM λ	2		

2

Figure 2.2: SoFIIA-MPC: Time to goal over different social weights p_{dyn} for head-on scenario with two agents.

behaves as an SFM agent. Since we consider the robot as an SFM agent, the SFM predictions inherently account for interactions between the robot and the pedestrians. The CVM was shown to outperform even state-of-the-art learning-based prediction models [55] and was applied in many state-of-the-art motion planners [56]. We solve the MPC problem using the MPPI-torch implementation[2] [57] with a horizon K of 20 time-steps, and a step-size $\Delta t = 0.1s$. Furthermore, we evaluate how the cost function can be used to adapt the behavior.

2.4.1 Comparative Analysis of Motion Planning: Non-Reactive vs. Reactive Agent Models

We first show that by considering the interactions the robot applying SoFIIA-MPC can exploit the other agents, see Fig. 2.2. The achieved time to goal for the robot decreases with increasing socialness of the other agent. Furthermore, we compare the navigation metrics over 10 random scenarios. We consider the following metrics:

- Traveled Distance Ratio: Distance to the goal divided by the straight line distance to the goal,

- Time to goal ratio: Time to goal divided by the time required to reach the goal in a straight line with maximum speed,

- Minimum distance: The minimum distance between agents.

The results, presented in Fig. 2.3, show that the Traveled Distance Ratio of the robot decreases as the planner accounts more for interactions. Furthermore, the variance in goal-reaching times decreased, which may be attributed to a reduction in erratic behaviors, e.g., as seen in the top left corner in Fig. 2.1. The distance to the pedestrians did not change significantly. This is expected, as the pedestrians

---

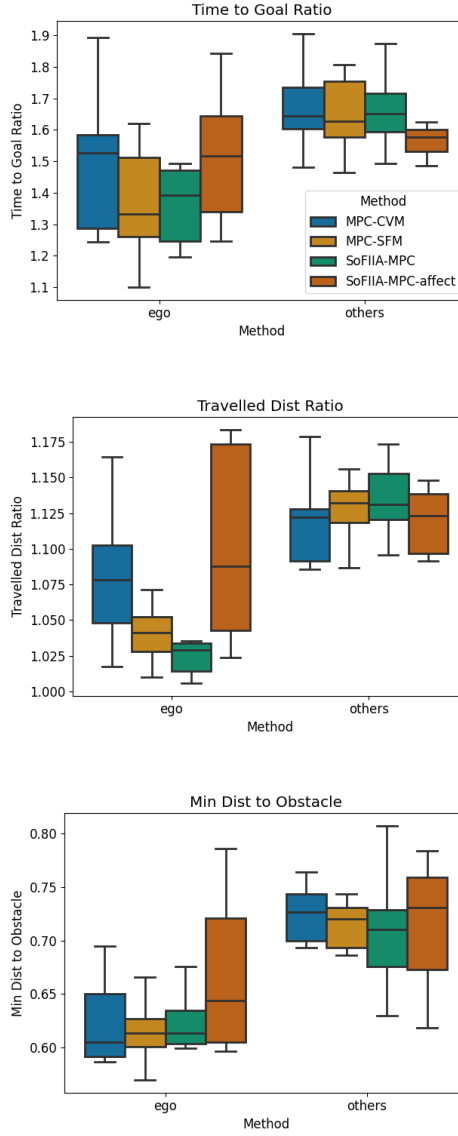2 `https://github.com/tud-airlab/mppi_torch/tree/main`

Figure 2.3: Metrics over 10 random scenarios. We show each metric for the different planners separated by agent type, i.e., robot/ego-agent and other agents. For the other agents we consider the mean value.

are also goal-directed and tend to maintain a certain separation from others due to the repulsive interaction forces in the model.

2.4.2 Exploiting Interactions for Desired Agent Behaviors

By explicitly considering the interactions, it is possible to influence the other agents to certain behaviors. While previous works in the autonomous driving field [49] demonstrate that cars can be slowed down or influenced to merge into another lane, these strategies are not directly applicable in the social navigation context. This is because the considered driving scenarios are more structured, e.g., by considering lanes. Thus, we set the robot's objective to navigate while trying to reduce its influence on the other agents. In Fig. 2.3 it can be seen that we were able to decrease the Time to Goal ratio for the other agents. However, these are preliminary results, which have to be further evaluated for a higher number of scenarios.

2.5 Conclusion & Future Work

In this work, we addressed the challenge of enhancing interaction in multi-agent motion planning while maintaining computational efficiency. Specifically, we formulate the interactions as an underactuated system and leverage the Social Forces Model (SFM) to represent pedestrians' response dynamics. Since our aim was to evaluate the effect of accounting for interactions, we assumed the parameters of the SFM as well as the pedestrians' goals to be known. How the parameters can be estimated remains to be explored. Future work will focus on further validating our approach in more diverse scenarios including static obstacles and in real-world scenarios.

3

Constructing Safety Filters Robust to Model Error and Disturbances via Robust Policy Control Barrier Functions

In Chapter 2, we discussed an approach that considers the robot's impact on pedestrians but not vice versa and requires a carefully designed cost function to avoid exploiting pedestrians. Imitation learning bypasses the need for a cost function, implicitly accounting for interactions but lacking safety guarantees. Safety guarantees for potentially unsafe controllers can be provided by safety filters, such as those based on Control Barrier Functions (CBFs), but constructing CBFs for high relative degree systems with input constraints is challenging, and their effectiveness depends on an accurate model.

This chapter introduces Robust Policy CBF (RPCBF), a practical method for constructing CBF approximations that are easy to implement and robust to disturbances. We demonstrate its effectiveness in simulation and show how RPCBFs compensate for model errors on a hardware quadcopter platform by treating these errors as disturbances.
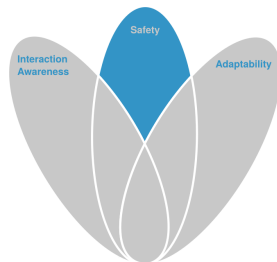
3.1 Motivation and Related Work

In the realm of autonomous systems, providing safety guarantees is crucial, especially in critical applications such as autonomous driving [58] and healthcare robotics [59]. Control Barrier Functions (CBFs) [60, 61] have proven to be an effective tool to maintain and certify the safety of dynamical systems. In particular, they can be applied as a Safety Filter (SF) that minimally modifies arbitrary control inputs to ensure safety, making them especially valuable when integrated with learning-based controllers, see Fig. 3.1.

Despite their theoretical advantages, significant challenges remain in the practical application and construction of Control Barrier Functions (CBFs). First, constructing CBFs is non-trivial, specifically for high relative degree systems with input constraints. Second, the safety guarantees of CBF-based controllers depend on having an accurate system model, which is rarely the case for systems in real life. This can result in the safety of these controllers being sensitive to model uncertainties.
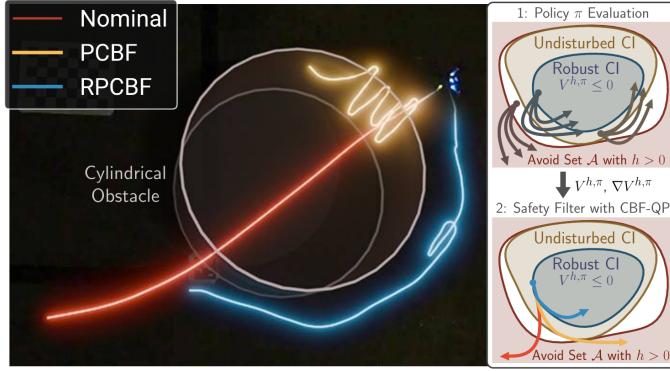
Figure 3.1: We propose the Robust Policy Control Barrier Function (RPCBF), which approximates the value function $V^{h,\pi}$ of a system under bounded disturbances for policy π online. The zero sublevel set of the RPCBF is a robust controlled-invariant set (CI) and can be used online as a RPCBF safety filter to ensure safety for any unsafe nominal policy. We demonstrate its superior performance compared to a safety filter using a PCBF safety filter on a quadcopter platform with model errors treated as disturbances.

3.1.1 Learning Control Barrier Functions

To minimize reliance on extensive domain knowledge, a recent trend is to learn neural CBFs that approximate CBFs using Neural Networks (NNs) [62–70]. Leveraging the flexibility of Neural Networks (NNs), neural CBFs have been successfully applied to high-dimensional systems, including multi-agent control scenarios [68, 71]. Furthermore, they have been extended to handle parametric uncertainties [72] and obstacles with unknown dynamics [73]. Although utilizing NNs as CBFs offers universal approximation capabilities, it requires their certification as CBF to provide safety guarantees and limits their interpretability. Furthermore, using a naive approach of learning neural CBFs by minimizing a loss that encourages the CBF

conditions can lead to a small or even empty forward-invariant set. Thus, [65] presents a method to construct CBFs using policy evaluation of *any* policy. They show that the policy value function is a CBF and learn an NN approximation of it. In this setting, the policy value function represents the maximum-over-time constraint violation and is generally a measure of how good a particular state is for a system following a specific policy over infinite time. However, their approach does not consider uncertainties in the system dynamics.

3.1.2 Robust Safety

Controllers robust to disturbances in the real world are important for safe autonomous systems. This has been studied before in *robust* CBFs [72, 74, 75] which guarantee safety under bounded disturbances. However, constructing robust CBFs is inherently more difficult than standard CBFs, especially under input constraints. Hamilton-Jacobi reachability analysis [76] can be used to compute robust control-invariant sets, which can then be subsequently used for constructing robust CBFs [77–80]. However, reachability analysis in itself is challenging, with grid-based Partial Differential Equation solvers being limited to state dimensions less than 5 [81], while deep learning-based solvers [82–85] require subsequent neural-network verification to check for solution accuracy. Moreover, both learning-based CBF approaches and deep learning-based reachability solvers rely on predefined disturbance assumptions that cannot be easily adapted as new information emerges, limiting their flexibility once deployed. While [86] train a value network with the avoidance set and disturbance bounds as inputs, this increases training data requirements and complicates evaluating how well the learned network represents the true value function.

As an alternative to robust safety, other works focus on risk-aware safety, which aims to ensure safety with high probability by modeling disturbances probabilistically and incorporating risk measures [87, 88]. Unlike robust methods, which ensure constraint satisfaction for all disturbances within a bounded set but may be overly conservative, risk-aware approaches typically rely on knowledge of the disturbance's probability distribution. In this work, we focus on robust safety.

3.1.3 Contributions

We propose a practical approach for constructing a CBF approximation at runtime, which can be derived for any system dynamics and disturbance bounds without requiring (re)training. We establish conditions under which the resulting CBF approximation qualifies as a valid CBF. Our method constructs CBFs by evaluating the value function of *any* policy, which has been shown to be a valid CBF in [65]. By leveraging finite-horizon policy rollouts, we enable a more detailed analysis of safety guarantees than NN approximations. We apply this approach to construct approximations of robust CBFs. We summarize our contributions as follows.

1. We propose a method of constructing (robust) CBFs using the (robust) policy value function and a real-time approximation that can be used at runtime.

2. We demonstrate real-time performance and the benefits of our robust CBFs on

a hardware quadcopter, where robustness to model errors is key for collision prevention.

3.2 Preliminaries

3.2.1 Problem Statement

We consider a *disturbed* continuous-time, control-affine dynamical system of the form

$$\dot{\mathbf{x}}_t = f(\mathbf{x}_t, \mathbf{d}_t) + g(\mathbf{x}_t, \mathbf{d}_t)\mathbf{u}_t, \tag{3.1}$$

with state $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^n$, control input $\mathbf{u}_t \in \mathcal{U} \subseteq \mathbb{R}^m$ and unknown, bounded, smooth disturbance $\mathbf{d}_{\min} \leq \mathbf{d}_t \leq \mathbf{d}_{\max}$ with $\mathbf{d}_{\min}, \mathbf{d}_{\max} \in \mathbb{R}^d$ (e.g., estimated from empirical data). Note that $\mathbf{d}_t$ can be time-varying. The functions f and g are assumed to be locally Lipschitz continuous. Let $\mathcal{A} \subset \mathcal{X}$ denote the set of states to be avoided. In this chapter, we address the following Safety Filter (SF) synthesis problem:

Problem 1 (Safety Filter Synthesis)**.** *Given a dynamical system* (3.1) *and an avoid set* $\mathcal{A} \subset \mathcal{X}$*, find a control policy* $\pi_{\mathrm{filt}} : \mathcal{X} \to \mathcal{U}$ *that ensures the system state remains outside the avoid set* $\mathcal{A}$ *while staying close to a performant but possibly unsafe nominal policy* $\pi_{\mathrm{nom}} : \mathcal{X} \to \mathcal{U}$*:*

$$\begin{aligned}
\min_{\pi} \ & \|\pi_{\mathrm{filt}} - \pi_{\mathrm{nom}}\| \\
\text{s.t. } & \dot{\mathbf{x}}_t = f(\mathbf{x}_t, \mathbf{d}_t) + g(\mathbf{x}_t, \mathbf{d}_t)\pi_{\mathrm{filt}}(\mathbf{x}_t), \\
& \mathbf{x}_t \notin \mathcal{A}, \ \forall t \geq 0,
\end{aligned} \tag{3.2}$$

where $\|\cdot\|$ *is some distance metric.*

We focus on solving Prob. 1 using (zeroing) CBFs [89].

3.2.2 Safety Filters using Control Barrier Functions

We begin by providing a standard definition of a CBF in the non-robust case, which we extend to the robust case for Policy Control Barrier Functions (PCBFs) in the next section. Define the *undisturbed* system to be a particular case of the disturbed system (3.1) without disturbances ($\mathbf{d} = 0$), by

$$\dot{\mathbf{x}}_t = f(\mathbf{x}_t, 0) + g(\mathbf{x}_t, 0)\mathbf{u}. \tag{3.3}$$

Let $B : \mathcal{X} \to \mathbb{R}$ be a continuously differentiable function, with $\mathcal{C} = \{\mathbf{x} \in \mathcal{X} \mid B(\mathbf{x}) \leq 0\}$ as its 0-sublevel set. Let $\alpha : \mathbb{R} \to \mathbb{R}$ be an extended class-κ_∞ function[1]. Then, B is a CBF for the undisturbed system (3.3) on $\mathcal{X}$ [60] if

$$B(\mathbf{x}) > 0, \ \forall \mathbf{x} \in \mathcal{A}, \tag{3.4a}$$

$$B(\mathbf{x}) \leq 0 \Rightarrow \inf_{\mathbf{u} \in \mathcal{U}} L_f B(\mathbf{x}) + L_g B(\mathbf{x})\mathbf{u} \leq -\alpha(B(\mathbf{x})), \tag{3.4b}$$

---

1   Extended class-κ_∞ is the set of continuous, strictly increasing functions $\alpha : (-\infty, \infty) \to (-\infty, \infty)$ with $\alpha(0) = 0$.

with $L_f B := \nabla B^\mathsf{T} f$ and $L_g B := \nabla B^\mathsf{T} g$. It then follows that any control input $\mathbf{u} \in K_{\mathrm{cbf}}$ with

$$K_{\mathrm{cbf}}(\mathbf{x}) = \{\mathbf{u} \in \mathcal{U} \,|\, L_f B(\mathbf{x}) + L_g B(\mathbf{x})\mathbf{u} + \alpha(B(\mathbf{x})) \le 0\}$$

renders $\mathcal{C}$ forward-invariant [60]. In other words, there exists an $\mathbf{u} \in \mathcal{U}$ such that any trajectory starting within $\mathcal{C}$ remains in $\mathcal{C}$. Asymptotic stability of $\mathcal{C}$ can be achieved by extending (3.4b) to hold for all $\mathbf{x} \in \mathcal{X}$ [89]. Since the right-hand side of condition (3.4b) is linear in $\mathbf{u}$, given a CBF B where (3.4b) is satisfied, we can solve Prob. 1 for (3.3) using the following Quadratic Program (QP)-based controller:

$$\mathbf{u}_{\mathrm{CBF-QP}} = \arg\min_{\mathbf{u} \in \mathcal{U}} \|\mathbf{u} - \pi_{\mathrm{nom}}(\mathbf{x})\|^2 \qquad \text{(CBF-QP)}$$
$$\text{s.t. } L_f B(\mathbf{x}) + L_g B(\mathbf{x})\mathbf{u} \le -\alpha(B(\mathbf{x})).$$

While CBFs can be applied to guarantee safety for a known undisturbed system, two major challenges remain:

1. How do we synthesize a valid CBF that satisfies (3.4b) for high relative degree systems with input constraints?

2. How do we synthesize a robust CBF that ensures safe control for the disturbed system?

3. How can we efficiently derive a CBF at runtime for different system dynamics and disturbance assumptions?

3.3 Robust Policy Control Barrier Functions

To address the above challenges, we leverage the insight from [65] that CBFs can be constructed by deriving the policy value function through the evaluation of *any* policy. Rather than approximating the policy value function with an NN as in [65], we propose an alternative that avoids NNs by instead performing a finite-horizon numerical approximation. We further extend this approach to the robust case and introduce Robust Policy CBFs (RPCBFs). Robust Policy Control Barrier Functions (RPCBFs) enable the reactive adjustment of assumptions made on the disturbances as new information emerges. In contrast, robust extensions of neural CBFs [72] require disturbance assumptions to be predefined before training, preventing any adaptation once deployed. Next, we revisit the formulation of PCBFs and describe our extensions and approximations.

3.3.1 Constructing CBFs via Policy Evaluation

Based on [65], we first derive the PCBF formulation for the undisturbed system in (3.3). Assume that the avoid set $\mathcal{A}$ can be described as the super-level set of a function $h : \mathcal{X} \to \mathbb{R}$ (e.g., the negative distance to the constraint):

$$\mathcal{A} = \{\mathbf{x} \in \mathcal{X} \,|\, h(\mathbf{x}) > 0\}. \qquad (3.5)$$

We denote by $\mathbf{x}_t^\pi$ the resulting state at time t when starting from the initial state $\mathbf{x}_0$ and following policy $\pi : \mathcal{X} \to \mathcal{U}$. Furthermore, we define the *maximum-over-time* value function for the undisturbed system in (3.3) as

$$V_\infty^{h,\pi}(\mathbf{x}_0) := \sup_{t \geq 0} h(\mathbf{x}_t^\pi). \tag{3.6}$$

As stated in [65, Theorem 1], the *policy value function* $V_\infty^{h,\pi}$ is a CBF for the undisturbed system in (3.3) for any π, since $V_\infty^{h,\pi}$ satisfies the following two inequalities $\forall \mathbf{x} \in \mathcal{X}$

$$V_\infty^{h,\pi}(\mathbf{x}) \geq h(\mathbf{x}), \tag{3.7}$$

$$\nabla V_\infty^{h,\pi}(\mathbf{x})^T (f(\mathbf{x}) + g(\mathbf{x})\pi(\mathbf{x})) \leq 0, \tag{3.8}$$

which imply (3.4a) and (3.4b). For details, we refer to [65]. The key intuition here is that $V_\infty^{h,\pi}$ provides an upper bound on the worst future constraint violation h under the optimal policy since the optimal policy will do no worse than π. Thus, CBFs can be constructed via policy evaluation of any policy. We refer to π as the design policy, noting that the nominal policy π_{nom} differs from the design policy.

3.3.2 Finite Horizon Approximation of PCBFs

A key challenge with the value function $V_\infty^{h,\pi}$ is that its definition requires an *infinite-horizon*. While [65] tackles this problem by using an NN to learn $V_\infty^{h,\pi}$ with a loss derived using dynamic programming, we take a different approach and perform a *finite-horizon* approximation that can be computed *without* the use of NN, enabling a more in-depth analysis of the resulting safety guarantees. Expanding $V_\infty^{h,\pi}$:

$$V_\infty^{h,\pi}(\mathbf{x}_0) = \max \left\{ \sup_{0 \leq t < T} h(\mathbf{x}_t^\pi), V_\infty^{h,\pi}(\mathbf{x}_T^\pi) \right\} \tag{3.9}$$

$$\approx \sup_{0 \leq t < T} h(\mathbf{x}_t^\pi) := V_T^{h,\pi}(\mathbf{x}_0), \tag{3.10}$$

where the approximation is made by dropping the $V_\infty^{h,\pi}(\mathbf{x}_T)$ "tail". The question is then whether the finite-horizon approximation $V_T^{h,\pi}$ is a CBF and has safety guarantees.

We can at least answer this in the affirmative when the approximation in (3.10) is an equality, i.e., the maximum occurs in $[0, T]$. We state this in the following theorem.

Theorem 1. *Suppose that for all* $\mathbf{x}_0 \in \mathcal{X}$,

$$V_T^{h,\pi}(\mathbf{x}_0) \leq 0 \implies \sup_{0 \leq t < T} h(\mathbf{x}_t^\pi) > V_\infty^{h,\pi}(\mathbf{x}_T^\pi). \tag{3.11}$$

Then, $V_T^{h,\pi}$ is a CBF.

Proof. Since $V_T^{h,\pi}(\mathbf{x}) \geq h(\mathbf{x})$ by definition, (3.4a) is satisfied by $V_T^{h,\pi}$. Moreover, by (3.11), $V_T^{h,\pi}(\mathbf{x}_0) = V_\infty^{h,\pi}(\mathbf{x}_0)$ when $V_T^{h,\pi}(\mathbf{x}_0) \leq 0$. Hence, since $V_\infty^{h,\pi}$ is a CBF, (3.4b) holds for $V_\infty^{h,\pi}$, and thus also holds for $V_T^{h,\pi}$. Thus, $V_T^{h,\pi}$ is a CBF. $\qquad\square$

This enables us to prove the following corollary.

Corollary 1. Suppose there exists a $\bar{T} < \inf$ such that

$$\underset{t \geq 0}{\arg\max}\, h(\mathbf{x}_t^\pi) < \tilde{T}, \quad \forall \mathbf{x}_0 \text{ where } V_T^{h,\pi}(\mathbf{x}_0) \leq 0 \qquad (3.12)$$

Then, $V_T^{h,\pi}$ is a CBF for $T \geq \tilde{T}$.

Proof. (3.12) implies (3.11) for $T \geq \tilde{T}$. The proof then follows from Theorem 1. $\quad\square$

The value of $\tilde{T}$ depends on the chosen policy π. In principle, any policy π can be selected, which may result in a conservative Control Invariant (CI) set. If π is chosen as a controller that steers the system towards a safe, steady state within a finite horizon $\tilde{T}$ Corollary 1 holds, thus $V_T^{h,\pi}$ is a CBF.

If $V_T^{h,\pi}(\mathbf{x}_0) = 0$, then applying the design policy π exactly guarantees that the system will remain safe for at least the time horizon $0 \leq t < T$, even if $V_T^{h,\pi}(\mathbf{x}_0) < V_\infty^{h,\pi}(\mathbf{x}_0)$.

Remark 1 (Connections to Backup Controller / CBFs)**.** Since the zero sublevel set of $V_\infty^{h,\pi}$ is a controlled-invariant set under π, (3.9) can also be seen as Backup CBF [90, 91] with backup controller π. Unlike this (and other similar approaches [92]), our approach replaces the need for a known forward-invariant set with the requirement of a sufficiently long horizon T. Thus, the design policy π can be chosen arbitrarily and is not required to steer the system into a CI set. Furthermore, we demonstrate that the naive approximation of h over a time-discretized state trajectory introduces gradient errors. To address this, we present an improved time-discretization using cubic splines in Sec. 3.3.3.

Remark 2 (Connections to Model Predictive Control (MPC))**.** The finite-horizon approximation here is closely related to the use of MPC by practitioners. More precisely, although a terminal constraint set is often required to theoretically guarantee recursive feasibility of finite-horizon MPC [18, 93], practitioners often apply MPC without the use of such a terminal constraint set to wide success [94–97]. Our decision to drop the $V_\infty^{h,\pi}(\mathbf{x}_T^\pi)$ term can be viewed as being similar to dropping the terminal constraint set. Another similarity is the choice of horizon T. Namely, recursive feasibility holds in MPC given a sufficiently large horizon [98], similar to Corollary 1. However, the MPC horizon length is limited, as it requires solving a potentially nonlinear and non-convex optimization problem online, with computational complexity typically scaling cubically with T [99]. A key advantage of PCBF-SFs is that they only solve the simpler (CBF-QP), whose computation time is unaffected by T, see Section 3.4.4.

Remark 3 (Connections to Predictive Safety Filter (PSF) [100])**.** The finite-horizon approximation is closely related to the PSF, which implicitly represents the safe set via a finite-horizon MPC problem with terminal constraints or long horizons for recursive feasibility. Unlike PCBF-SFs, the PSF requires solving a potentially nonlinear and nonconvex optimization problem online with complexity scaling cubically

in T [99]. PCBF-SFs only require solving the simpler (CBF-QP). However, while the PSF may find a locally optimal solution, the conservativeness of the PCBF-SF depends on π.

3.3.3 Time Discretization of Policy Control Barrier Functions

Another challenge lies in how to compute the maximum in (3.10). The states $\mathbf{x}_t^\pi$ can be solved numerically using an Ordinary Differential Equation solver, resulting in a time-discretized state trajectory. It is tempting to then consider taking the maximum h over this trajectory, i.e., for time discretization Δt,

$$V_T^{h,\pi}(\mathbf{x}_0) \approx \max_{0 \le k < H} h(\mathbf{x}_{k\Delta t}^\pi). \tag{3.13}$$

However, the gap between (3.10) and (3.13) is particularly disastrous when computing the gradient. We illustrate this in the following example for the Double Integrator (DI).

Example: Gradient Error on the Double Integrator. Consider a DI with positive velocity $v_0 > 0$ decelerating with $\pi(\mathbf{x}) = a = -1$. The dynamics are defined by $\dot{p} = v$, $\dot{v} = a$ with initial state $\mathbf{x}_0 = [p_0, v_0]$ and constraints $h(\mathbf{x}) = p \le 0$. For the continuous-time case, the gradient can be derived as

$$\nabla V_\infty^{h,\pi}(\mathbf{x}_0) = \frac{\partial p_{\max}}{\partial \mathbf{x}_0} = [1, v_0]. \tag{3.14}$$

After (exact) time discretization with timestep Δt, the time-discretized states can be computed as

$$p_k = p_0 + v_0 k \Delta t + 0.5 a (k \Delta t)^2, \tag{3.15a}$$

$$v_k = v_0 + a k \Delta t. \tag{3.15b}$$

We now show that the gradient of $V_\infty^{h,\pi}$ depends on Δt and denote by $\nabla V_{\infty,\Delta t}^{h,\pi}$ the resulting gradient. Let τ be the *integer* time step k at which the maximum position is reached. The maximum position is then given by

$$V_{\infty,\Delta t}^{h,\pi} = p_{\max} = p_0 + v_0 \tau \Delta t + 0.5 a (\tau \Delta t)^2, \tag{3.16}$$

with $\frac{\partial p_{\max}}{\partial v_0} = \tau \Delta t$, which is a function of τ. Although τ also depends on v_0, it is piecewise constant and has zero derivative since it only takes integer values. Comparing the gradients of $V_\infty^{h,\pi}$ with $V_{\infty,\Delta t}^{h,\pi}$ in Fig. 3.2, we see a large error between the two with discontinuities in the discrete-time gradient in Δt. This is particularly problematic when the gradient is used in a gradient-based optimization algorithm such as (CBF-QP).

Improved Time-Discretization using Cubic Splines. To reduce the error in the time-discretized value function approximation (3.13), we propose to approximate $h(\mathbf{x}_t^\pi)$ by fitting a cubic spline to the points $\{h(\mathbf{x}_{k\Delta t}^\pi)\}_{k=0}^{H-1}$. The max over the cubic spline can then be computed in closed-form by solving the roots of a quadratic to yield a better approximation of $\sup_{0 \le t < T} h(\mathbf{x}_t^\pi)$ than the naive maximization
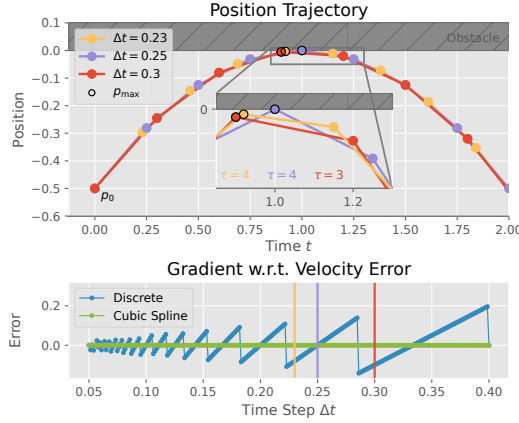
Figure 3.2: Value Function Gradient Error for Discrete-Time Double Integrator. We highlight the discretized trajectory (top) and corresponding gradient error (bottom) for three different choices of Δt (yellow, purple, red). The gradient of the naive discrete-time approximation has large errors and varies with the choice of Δt. Taking the maximum of the cubic spline leads to much smaller errors.

(3.13). Intuitively, this resolves the gradient error due to integer-valued τ from the previous example because the maximum of the cubic spline can now happen *between* timesteps. We can also formally quantify the error in both the cubic spline value and its gradient. Let $\tilde{h} : [0, \infty) \to \mathbb{R}$ denote the cubic spline approximation of h as a function of time, and let $\tilde{V}_{\infty,\Delta t}^{h,\pi}(\mathbf{x}_0) \coloneqq \sup_{t \geq 0} \tilde{h}(t)$. Using [101, Chapter 5], we obtain the error bounds

$$\left\| V_\infty^{h,\pi} - \tilde{V}_{\infty,\Delta t}^{h,\pi} \right\| \leq \frac{1}{16} \Delta t^4 \max_{t \geq 0} \left\| \frac{\mathrm{d}^4}{\mathrm{d}t} h(\mathbf{x}_t) \right\|, \tag{3.17a}$$

$$\left\| \nabla V_\infty^{h,\pi} - \nabla \tilde{V}_{\infty,\Delta t}^{h,\pi} \right\| \leq \frac{1}{24} \Delta t^3 \max_{t \geq 0} \left\| \frac{\mathrm{d}^4}{\mathrm{d}t} h(\mathbf{x}_t) \right\|. \tag{3.17b}$$

In the previous example of the DI, since h is exactly quadratic, applying cubic splines results in zero gradient error (Fig. 3.2). If $\mathrm{d}^4/\mathrm{d}t\, h(\mathbf{x}_t)$ can be bounded, the bounds (3.17) can then be used to suitably modify the QP (CBF-QP) to guarantee safety. A larger Δt will therefore result in a more conservative SF

3.3.4 Robust Extension of PCBFs

In this subsection, we now consider a *robust* extension of PCBF to handle disturbances. Defining the *robust* value function equivalent of (3.6) as

$$V_\infty^{h,\pi}(\mathbf{x}_0) \coloneqq \sup_{t \geq 0} \sup_{\mathbf{d}(\cdot)} h(\mathbf{x}_t^\pi), \tag{3.18}$$

it can be shown with similar proof [65, 102] that $V_\infty^{h,\pi}$ is a *robust* CBF [103], i.e., it satisfies (3.4a) and, for $B(\mathbf{x}) \leq 0$,

$$\sup_{\mathbf{d} \in \mathcal{D}} \inf_{\mathbf{u} \in \mathcal{U}} \nabla B^\mathsf{T} \big(f(\mathbf{x}, \mathbf{d}) + g(\mathbf{x}, \mathbf{d})\mathbf{u} \big) \leq -\alpha\big(B(\mathbf{x})\big). \tag{3.19}$$
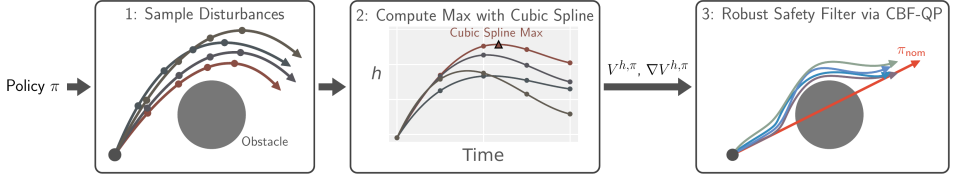
Figure 3.3: Summary of RPCBF Algorithm. Given a policy π, we sample disturbance trajectories, then compute the maximum h with cubic splines to obtain $V^{h,\pi}$ and $\nabla V^{h,\pi}$ (using autodiff). This is used in a CBF-QP to obtain a robust safety filter.

Algorithm 1 Robust Policy CBF (RPCBF)

1: **Input:** Initial State $\mathbf{x}_0$, Policy π, Constraint function h, Horizon $T = H\Delta t$, Number of disturbance samples N
2: **for** $i = 1 : N$ **do**
3:      Sample disturbance trajectory $\{\mathbf{d}_k^i\}_{k=1}^{H-1}$
4:      Rollout the policy π on disturbed system (3.1)
5:      Compute $\sup_{0 \leq t < T} h(\mathbf{x}_t^i)$ using cubic splines
6: **end for**
7: Compute $V_{T,N}^{h,\pi}(\mathbf{x}_0)$ according to (3.20)
8: Compute the gradient $\nabla V_{T,N}^{h,\pi}(\mathbf{x}_0)$ using automatic differentiation

---

Solving for robust controls that satisfy (3.19) renders the zero sublevel set *robust* forward-invariant [74]. However, deriving the worst-case disturbance is generally intractable because it requires evaluating all possible disturbance trajectories. Instead, we propose to only consider N disturbance trajectories and take the worst-case out of the N samples, resulting in the following practical RPCBF approximation.

$$V_T^{h,\pi}(\mathbf{x}_0) \approx V_{T,N}^{h,\pi}(\mathbf{x}_0) := \max_{i=1,\cdots,N} \sup_{0 \leq t < T} h(\mathbf{x}_t^i), \tag{3.20}$$

$$\dot{\mathbf{x}}^i = f(\mathbf{x}^i, \mathbf{d}^i) + g(\mathbf{x}^i, \mathbf{d}^i)\mathbf{u}^i. \tag{3.21}$$

We summarize our approach in Alg. 1 and Fig. 3.3. Note that Alg. 1 must be executed once per control loop to obtain the value $V_{T,N}^{h,\pi}$ and gradient $\nabla V_{T,N}^{h,\pi}$ for the current state for use in the CBF-QP (CBF-QP). Different approaches can be implemented to perform informed sampling of disturbances. For bounded disturbances, the worst-case scenario often occurs at the vertices of the disturbance set (e.g., for disturbance-affine dynamics). Consequently, we choose to sample from a mixture of the uniform distribution $\mathcal{U}(\mathbf{d}_{\min}, \mathbf{d}_{\max})$ and the uniform distribution over the vertices. This can be extended to better optimizers to approximate the worst-case samples which we leave as future work.

While the finite-sample approximation does not guarantee robustness to any disturbance, it does ensure robustness to the specific sampled disturbances within the finite horizon. As the number of informed samples approaches infinity, the approximation increasingly captures the true worst-case scenarios. For the simulation and hardware experiments, we use PCBF and RPCBF to refer to their time-discretized finite-horizon and finite-sample approximations as described in this section.

3.4 Simulation Experiments

To study the performance of PCBF and RPCBF, we perform a series of simulation experiments on high relative degree systems under box control constraints.

Baselines

We compare against the following SFs that also do not use NNs in their approach.

- **Handcrafted Candidate CBF (HOCBF) [104, 105]:** We construct a *candidate* CBF via a Higher-Order CBF on h without considering input constraints.

- **Approximate Nominal MPC-based Predictive SF (MPC) [100]:** A trajectory optimization problem is solved, imposing the safety constraints while penalizing deviations from the nominal policy. We consider the undisturbed system and do not assume access to a known robust forward-invariant set and hence do not impose this terminal constraint.

Systems

We consider four systems: a DI, a Segway, an F-16 fighter jet (ground collision avoidance problem) [106, 107], and AutoRally [108], a 1/5 autonomous vehicle. On the DI, we consider position bounds ($|p| \leq 1$), while the Segway asks for an upright handlebar and considers position bounds ($|\theta| \leq 0.3\pi$, $|p| \leq 2$), $\Delta t = 0.1$. For the F-16, safety is defined as box constraints on states like altitude. Since this system is not control-affine in the throttle, we leave the throttle as the output of a P controller, resulting in a 16-dimensional state space and a 3-dimensional control space. In AutoRally, a crash occurs when the car stops after hitting the track boundary, while a collision involves contact without stopping. For each system, we define J continuously differentiable constraint functions h_j tailored to the problem at hand. For example, for the DI system we set $h_0 = p - 1$ and $h_1 = -(p + 1)$. From these we derive J corresponding CBFs, which yield to J constraints in (CBF-QP). For the DI and the Segway, we assume unknown but bounded time-varying disturbances on the mass, for the F-16, unknown but bounded matched disturbances ($d = 1$), and for the AutoRally, additive truncated Gaussian noise. Keep in mind that our approach does not require designing/learning a new CBF for different systems, disturbance assumptions, or input constraints, but simply requires swapping the dynamics, specifying the disturbance, and constraints. During testing, we consider a constant zero-control nominal policy for the DI, maximum acceleration for the Segway, a PID controller for the F-16, and Model Predictive Path Integral (MPPI) control [109] for AutoRally.

3.4.1 Influence of Horizon Length on Segway

While the infinite-horizon policy value function is a CBF, we use a finite-horizon approximation, making the SF performance horizon-dependent. To illustrate this, we assess the impact of different horizon lengths on the PCBF-SF, see Fig. 3.4. We plot the state space from where π_{nom} can influence the output of the SF (*Filter*
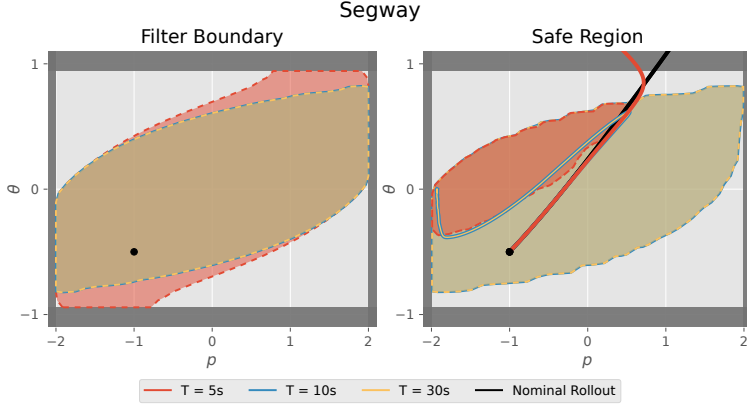
Figure 3.4: Filter Boundary and Safe Region for PCBFs with varying Horizon T on the Segway. We plot the states where the nominal policy can influence the output of the safety filter (Filter Boundary) and the initial states from which the safety filter can preserve safety over a horizon $\bar{T} = 30s$ (Safe Region). A trajectory from an initial state inside the filter boundary marked by a black dot (●) is displayed. If the horizon is too short, the filter boundary is overapproximated, resulting in an unsafe trajectory.

Boundary) and from where the SF preserves safety (*Safe Region*). For CBF-based filters, the filter boundary is defined by the CBF's zero level set. The safe region is determined for a π_{nom} by solving (CBF-QP) and rolling out the system over a horizon $\bar{T} = 30s$. For a short PCBF horizon, i.e., $T = 5s$, the true CI set is overapproximated. Consequently, the SF fails to preserve safety, as illustrated by the resulting unsafe example trajectory. In contrast, a longer horizon of $T = 10s$ provides a much closer approximation of the true CI set. This is evident when comparing it to an even longer horizon, such as $T = 30s$, which does not result in a visibly smaller safe region, indicating that $T = 10s$ is already sufficient.

3.4.2 Behavior on Disturbed Double Integrator and Segway

We explore the robustness of different SFs, examining the filter boundary and safe region, derived for one sampled disturbance trajectory per state, as shown in Fig. 3.5. We visualize rolled-out trajectories for $\bar{N}$ sampled disturbance trajectories (uniformly sampled and on the vertices) from selected initial states within the filter boundary. On the DI, only the RPCBF-SF achieves safety for all $\bar{N}$ sampled disturbance trajectories. Since the RPCBF accounts for the worst-case among the N sampled disturbances, the filter boundary is more conservative. On the Segway, MPC violates the safety constraints in all cases and hence has an empty filter boundary and safe region. Only the RPCBF-SF achieves safe trajectories for all considered samples. Next, we evaluate the (R)PCBF-SFs at uniformly distributed initial states, see Fig. 3.6. The RPCBF-SF achieves safety for all evaluated states within its zero-level set and the sampled disturbances, while the PCBF overapproximates the safe set.
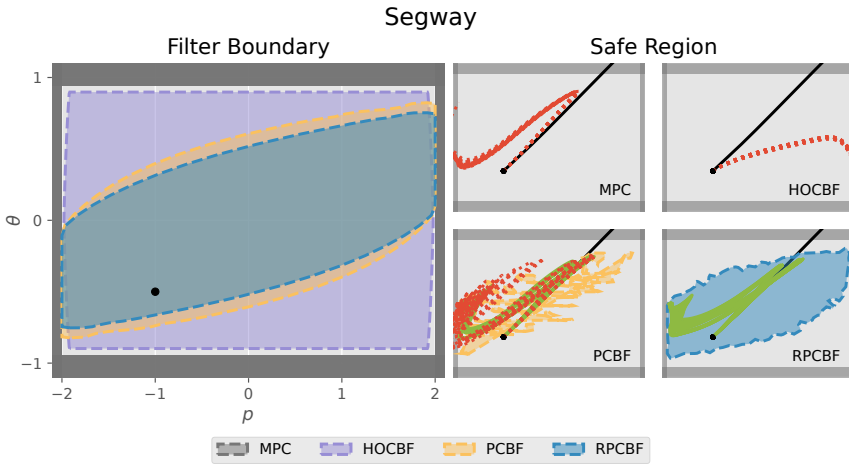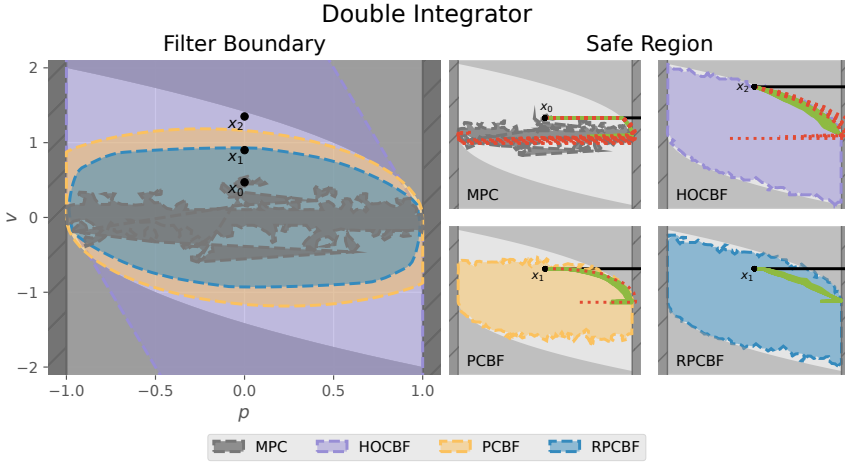
(a) $T = 6.4s$, $N = 100$ and $\bar{T} = 15s$.

(b) $T = 20s$, $N = 100$ and $\bar{T} = 30s$

Figure 3.5: Comparison of Filter Boundary and Safe Region on the Double Integrator (a) and Segway (b). The true unsafe region for the undisturbed system is shaded in gray for the double integrator. PCBF-based methods use a horizon T and N samples to derive the policy value function. The Safe Region is determined for a horizon $\bar{T}$. Trajectories from selected initial states for $\bar{N} = 25$ sampled disturbance trajectories are displayed. Red dotted lines and green solid lines indicate unsafe and safe trajectories, respectively. The nominal trajectory is shown in black.
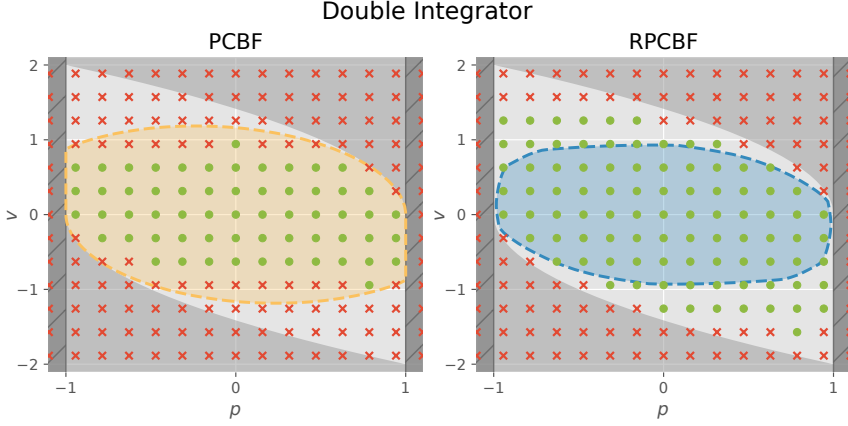
Figure 3.6: Robust Safety Evaluation. We plot the zero level set of the CBFs. Green dots indicate safe rollouts for all $\bar{N} = 25$ sampled disturbance trajectories, while red crosses indicate a failure in at least one trajectory. RPCBF achieved safety for all states within its zero level set for the sampled disturbances.

3.4.3 Simulations on AutoRally

Finally, to assess the safety improvements brought about by the proposed PCBF and RPCBF, we integrate the HOCBF and the proposed methods with Shield-MPPI [108, 110], and evaluate their performance on AutoRally. The Fig. 3.7 shows that Shield-MPPI using RPCBF generates the safest trajectories, while other controllers generate trajectories that collide and crash more often. The statistics of the safety performance of the controllers are shown in Tab. 3.1.
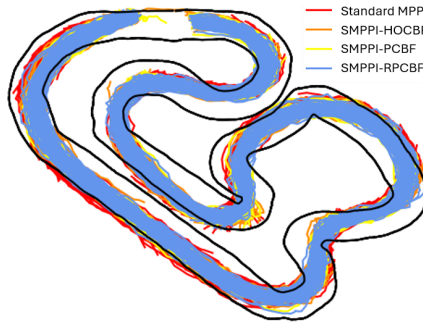

Figure 3.7: Trajectory Comparisons on AutoRally. We visualize the trajectories for each controller, where SMPPI-RPCBF (in blue) leads to the tightest spread of states inside the track.

Table 3.1: Collision and Crash Rate on AutoRally. The standard MPPI leads to most collisions and crashes. While Shield-MPPI with HOCBF and PCBF significantly improves safety, Shield-MPPI using RPCBF achieves the lowest rate of collision and crashes.

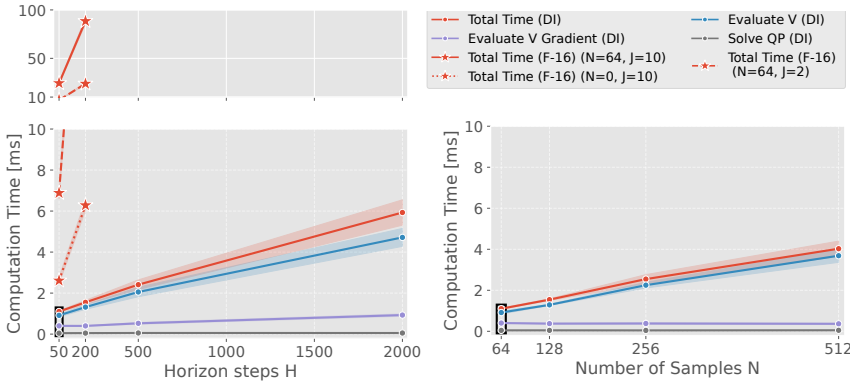| Controller	Mean Collisions per Lap	Crash Rate
MPPI	4.53	0.80
SMPPI-HOCBF	1.38	0.15
SMPPI-PCBF	1.23	0.12
SMPPI-RPCBF	**1.13**	**0.09**

Figure 3.8: Per-Timestep Computation Times for Double Integrator and F-16. Computation times for the total and individual components for varying H ($N = 64$) and N ($H = 50$), including mean and standard deviation across initial conditions and timesteps. The black box highlights the settings considered in the hardware experiments.

3.4.4 Comparison of Computation Times

Figure 3.8 shows the real-time feasibility of RPCBF-SF on the DI and F-16, highlighting how (component) computation times scale with increasing T and N, evaluated on a laptop CPU.

3.5 Hardware Experiments

We conduct hardware experiments on the Crazyflie platform to determine whether the proposed RPCBF can be robust to disturbances encountered in the real world (see Fig. 3.1). We use the onboard position PID controller on the Crazyflie, treating the system as a DI, and assume that the position setpoints are converted into accelerations onboard. The error between this simple model and the true dynamics is treated as an acceleration disturbance. We randomly generate a nominal trajectory and treat the corresponding positions as the nominal control. A circular obstacle is placed at the densest part of the nominal trajectory to encourage collisions. We use a $T = 5s$ (50 steps at $\Delta t = 0.1\,\text{s}$) and sample 64 disturbance trajectories. The

RPCBF-SF controller runs at a frequency of $100\,\mathrm{Hz}$ on a laptop.

We first run the PCBF and RPCBF controllers with $\alpha = 5$ on 6 different nominal trajectories and plot the results from 3 of the random trajectories in Fig. 3.9. The RPCBF maintains safety in all cases, while the PCBF collides in all cases. We next
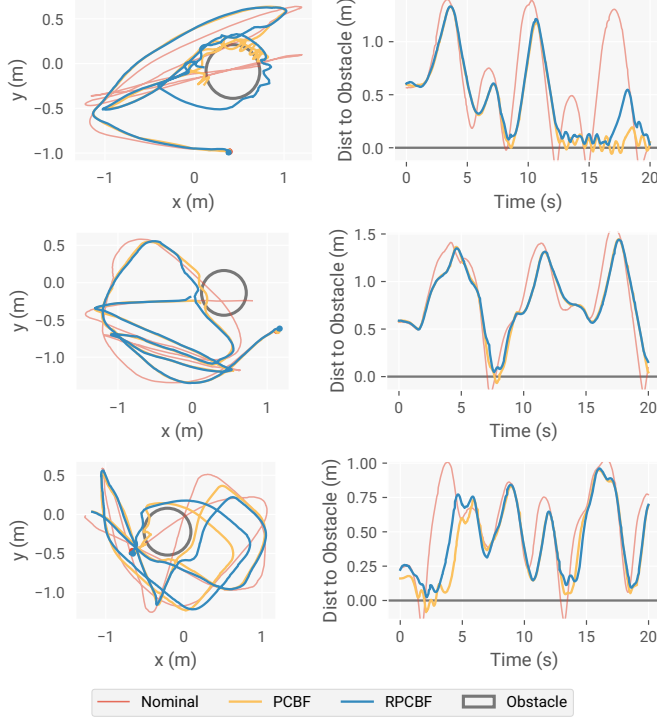

Figure 3.9: Hardware Traces. RPCBF maintains safety while following the unsafe nominal trajectory despite error between the modeled and true Crazyflie dynamics. On the other hand, PCBF assumes the modeled dynamics are perfect and collides with the obstacle.

vary the choice of the class-κ function α and plot the results in Fig. 3.10. While the non-robust PCBF does not collide with the obstacle when α is sufficiently small, this requires fine-tuning and is difficult to know beforehand. On the other hand, RPCBF is safe for all values of α we tested, allowing α to be used as a parameter that controls the behavior without also simultaneously affecting safety.

3.6 Conclusion & Future Work

In this work, we proposed the Robust Policy Control Barrier Function (RPCBF), a method for constructing robust CBFs using the robust policy value function derived from rolling out a design policy. Subsequently, we introduced a real-time approximation that can be derived online, with conditions for its validity as a CBF. Simulation
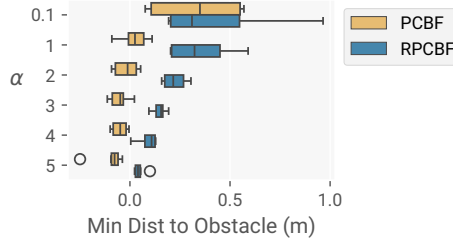
Figure 3.10: Safety with different values of α. We plot the minimum distance to the obstacle, over 6 random nominal trajectories. Although PCBF is safe when α is sufficiently small, this requires fine-tuning and is difficult to know beforehand. RPCBF is safe for all values of α we tested.

experiments demonstrate that a safety filter constructed using the RPCBF yields improved safety and more accurate estimation of the robust control-invariant set compared to existing methods. Hardware experiments on a quadcopter highlight the importance of accounting for model errors to ensure safety.

Future work will focus on analyzing the safety guarantees of the RPCBFs approximation, considering the finite-horizon, time-discretization, and sampling-based approach. Additionally, while the RPCBF acts as a CBF for any design policy if a long enough horizon is considered, conservativeness depends on the design policy. Deriving a policy to reduce conservativeness while maintaining infinite horizon guarantees is an important research direction. Moreover, extending our method to time-varying constraints and integrating real-time onboard perception are important directions for future work.

4

Improving Pedestrian Prediction Models with Self-Supervised Continual Learning

The previous chapters discussed various approaches for ensuring interactive and safe local motion planning. However, these methods are generally designed offline and may struggle to adapt to changing environmental conditions and the resulting shifts in expected behaviors.

This chapter focuses on the adaptability aspect in the context of pedestrian prediction models. It proposes a self-supervised continual learning framework to refine pedestrian prediction models during deployment using online pedestrian data from the robot's perception pipeline. The framework preserves prior knowledge through regularization and selective retraining. Experiments on real and simulated data show enhanced performance compared to naive online training. While this chapter considers pedestrian prediction models, the approach extends to other methods applying learning from observations. It can then be used in conjunction with the safety filter presented in Chapter 3.
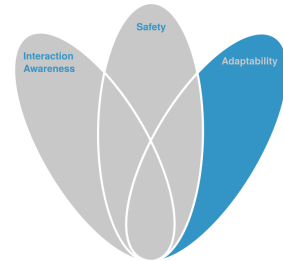
4.1 Motivation and Contribution

Autonomous mobile robots increasingly populate human environments, such as hospitals, airports and restaurants, to perform transportation, assistance and surveillance tasks [111]. In these continuously changing environments robots have to navigate in close proximity with pedestrians. To efficiently and safely navigate around them, robots must be able to reason about human behavior [112]. Predicting pedestrian trajectories is challenging, especially in crowded spaces where humans closely interact with their neighbors. This is the case, since the occurring interactions are complex, often subtle, and follow social conventions [113]. Furthermore, humans are influenced by the robot's presence [114], features of the static environment, such as its geometry or obstacle affordance, and various internal stimuli, such as urgency, which are difficult to measure [115, 116].

A large amount of research has been done on pedestrian prediction models [115]. Recently, the focus has mainly been on data-driven models which do not rely on hand-crafted functions and thus allow to capture more complex features and leverage large amounts of data. They address various aspects of pedestrian behavior such as stochasticity [117] and multi-modality [18, 118]. Moreover, they consider the influence of static obstacles [112], interactions among pedestrians [113] and the robot's presence [119]. However, these models are trained offline using supervised learning and thus do not adapt to unseen behaviors or environments and may fail if the testing data distribution differs from the training data distribution.

These limitations can be overcome by continuously training pedestrian prediction models on new streams of data. Hurdles in applying supervised continuous learning to existing prediction models are the slow and expensive creation of labeled data sets or the lack of supervision [16]. Robots operating in the same environment as pedestrians can autonomously collect training examples based on the robot's never-ending stream of observations. If a robot can efficiently and autonomously collect examples, its internal prediction models can be updated on the fly and the robot can effectively adapt its behavior. However, neural networks are prone to forget previously learned concepts while sequentially learning new concepts [16]. This phenomenon is referred to as *catastrophic forgetting*. To overcome catastrophic forgetting, we use a *regularization* strategy, namely Elastic Weight Consolidation (EWC) [120], to selectively slow down learning for important model parameters, in combination with *rehearsing* a small set of examples from previous tasks.

The main contribution of this chapter is therefore the introduction of a self-supervised continual learning framework that uses online streams of data of pedestrian trajectories to continuously refine data-driven pedestrian prediction models, see Fig. 4.1. Our approach overcomes catastrophic forgetting by combining a regularization loss and a data rehearsal strategy. We evaluate the proposed method in simulation, showing that our framework can improve prediction performance over baseline methods and avoid catastrophic forgetting, and in experiments with a mobile robot, showing that our framework can continuously improve a prediction model without the need for external supervision.
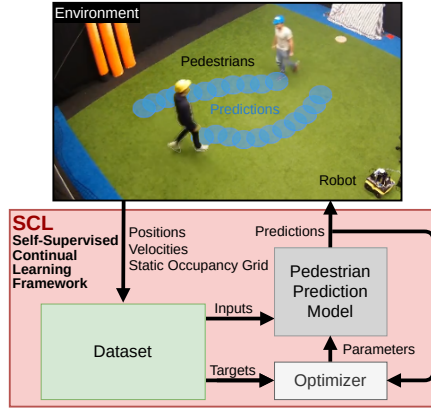
Figure 4.1: Self-supervised Continual Learning (SCL) framework used to continuously improve data-driven pedestrian prediction models online across various scenarios.

4.2 Related Work

In this section, we describe relevant approaches for pedestrian motion prediction and continual learning.

4.2.1 Pedestrian Motion Prediction

There has been a vast amount of work devoted to pedestrian trajectory prediction [115]. Early works are mainly model-based, such as the well-known Social Force Model (SFM) which uses attracting and repulsive potentials to model the social behaviors of pedestrians [121], and the velocity-based models which compute collision-free velocities for trajectory prediction [122, 123]. A limitation of these model-based approaches is that they only utilize handcrafted features, thus not being able to capture complex interactions in crowded scenarios. To overcome the limitation, Recurrent Neural Networks (RNNs) have been used for human trajectory prediction, which allows representing complex features and leverage large amounts of data [124]. Building on RNNs, [113] utilized Long Short-Term Memory (LSTM) networks to model time dependencies and employed a pooling layer to model interactions. [119] proposed a network model that is aware of the environment constraints. In addition, other network models have been developed to predict pedestrian trajectories, including Generative Adversarial Networks (GANs) [118, 125] and Conditional Variational Autoencoders (CVAEs) [126, 127]. Albeit being efficient, these models are usually trained and evaluated using (offline) bench-marking datasets [128–131], which limits their online adaption to unseen scenarios. In this chapter, we propose an approach to improve these models online by introducing a self-supervised continual learning framework.

4.2.2 Continual Learning

Continual Learning (CL) addresses the training of a model from a continuous stream of data containing changing input domains or multiple tasks [132]. The goal of CL is to adapt the model continually over time while preventing new data from overwriting previously learned knowledge. Existing CL approaches that mitigate catastrophic forgetting for neural network-based models can be divided into three categories: architecture-, memory- and regularization-based [16, 133].

Architecture-based approaches change the architecture of the neural network by introducing new neurons or layers [134–136]. Intuitively, these approaches prevent forgetting by populating new untouched weights instead of overwriting existing ones. However, the model complexity grows with the number of tasks.

Memory-based approaches save samples of past tasks to rehearse previous concepts periodically [16]. There are two types of memory-based methods that differ in the way they memorize past experiences: *rehearsal* methods explicitly saving examples [137] and *pseudo-rehearsal* methods saving a generative model from which samples can be drawn [138]. The data stored in the memory of rehearsal methods can be randomly chosen or carefully selected [137, 139]. Some methods require task boundaries [137] while other methods can be applied to the task free setting [139]. Since memory-based approaches require a separate memory, they can become unsustainable with an increasing number of tasks.

Regularization-based approaches add a regularization term to the loss to prevent modification of model parameters. This can be done using basic regularization techniques, such as weight sparsification, early stopping, and dropout, or with more complex methods which selectively prevent changes in parameters that are important to previous tasks [16]. [120] introduced *Elastic Weight Consolidation (EWC)*, a regularization approach limiting the plasticity of specific neurons based on their importance determined from the diagonal of the Fisher Information Matrix (FIM). To compute the FIM, clear task boundaries are required. Other regularization approaches focus on relaxing this assumption by automatically inferring task-boundaries [140], or by calculating the importance in an online fashion over the entire learning trajectory [141]. In contrast to other categories of approaches, these regularization-based methods do not require much computational and memory resources. However, one downside of regularization-based approaches is that an additional loss term is added, which can lead to a trade-off between knowledge consolidation and performance on novel tasks.

Most of the time, combining different continual learning strategies results in better performance [16]. Hence, in this chapter, we employ the EWC regularization technique combined with a data rehearsal strategy to achieve continual learning to improve pedestrian prediction models.

4.3 Problem Formulation

Throughout this chapter, we denote vectors, $\mathbf{x}$, in bold lowercase letters, matrices, M, in uppercase letters, and sets, $\mathcal{X}$, in calligraphic uppercase letters.

We address the problem of continuously improving a trajectory prediction model online using streams of pedestrian data. This data includes the position and ve-

locity of all n tracked pedestrians over time, and an occupancy map of the static environment $\mathcal{S}$. The position, velocity, and the surrounding static environment of the i-th pedestrian at time t are denoted by $\mathbf{p}_t^i = [p_{\mathrm{x},t}^i, p_{\mathrm{y},t}^i]$, $\mathbf{v}_t^i = [v_{\mathrm{x},t}^i, v_{\mathrm{y},t}^i]$, and $\mathcal{O}_t^{\mathrm{env},i} \subset \mathcal{S}$, respectively. The sub-scripts x and y indicate the x and y direction in the world frame. The super-script i denotes the *query-agent*, i.e., the pedestrian whose trajectory we want to predict.

Denote by $\mathcal{X}_t^i$ the observations acquired within a past time horizon t_{obs} for predicting pedestrian i's future trajectory, which typically includes its own states, the states of the other pedestrians and environment information. Further, denote by $\hat{\mathcal{Y}}_t^i$ the predicted trajectory of pedestrian i over the future prediction horizon t_{pred}.

We seek a data-driven prediction model $\hat{\mathcal{Y}}_t^i = f_{\boldsymbol{\theta}}(\mathcal{X}_t^i)$, with parameters $\boldsymbol{\theta}$, that best approximates the true trajectory $\mathcal{Y}_t^i$ across the entire previous stream of states for every tracked pedestrian $i \in \{1, \ldots, n\}$. The true trajectory $\mathcal{Y}_t^i$ will only become available in hindsight after observing the trajectory taken by pedestrian i during t_{pred}. Thus, we formulate the problem of continually learning a data-driven prediction model from past observations at time t as a regret minimization problem:

$$\min_{\theta} \sum_{i=1}^{n} \sum_{\tau=t-t_{\mathrm{his}}}^{t} \mathscr{L}_{\mathrm{pred}}(\hat{\mathcal{Y}}_\tau^i, \mathcal{Y}_\tau^i), \qquad (4.1)$$

where t_{his} is the entire elapsed time until t and $\mathscr{L}_{\mathrm{pred}}(\hat{\mathcal{Y}}_\tau^i, \mathcal{Y}_\tau^i)$ is the regret at one past time step τ for pedestrian i, which will be described in later sections.

4.4 Self-Supervised Continual Learning Framework

In this section, we introduce the Self-Supervised Continual Learning (SCL) framework, an online learning framework to continually improve pedestrian prediction models. Sec. 4.4.1 presents the overall structure of SCL, Sec. 4.4.2 the prediction network architecture, Sec. 4.4.3 the data aggregation and Sec. 4.4.4 the model adaption.

4.4.1 Framework Overview

The SCL architecture, consisting of two phases: a task aggregation and a model adaptation phase is presented in Fig. 4.2. Firstly, we use a prediction model which was pre-trained on publicly available datasets [128, 129] and aggregate new training examples using the surrounding pedestrians as experts (task aggregation) for a period of time T. Then, we update the prediction model using the aggregated data of the current task and a small constant sized coreset, which contains examples from previous tasks (model adaptation). During the model adaption phase, we apply an EWC loss to preserve the prediction performance on previous tasks. The two phases run alternately over time to create a continuous learning autonomous robot. During the task aggregation phase, we associate a new task to a new environment on which the model was previously not trained on. To distinguish between tasks, we will refer to the currently considered task as task_k. The previous tasks are referred to as $\mathrm{task}_{0:k-1}$ where the subscript 0 refers to the initial task.
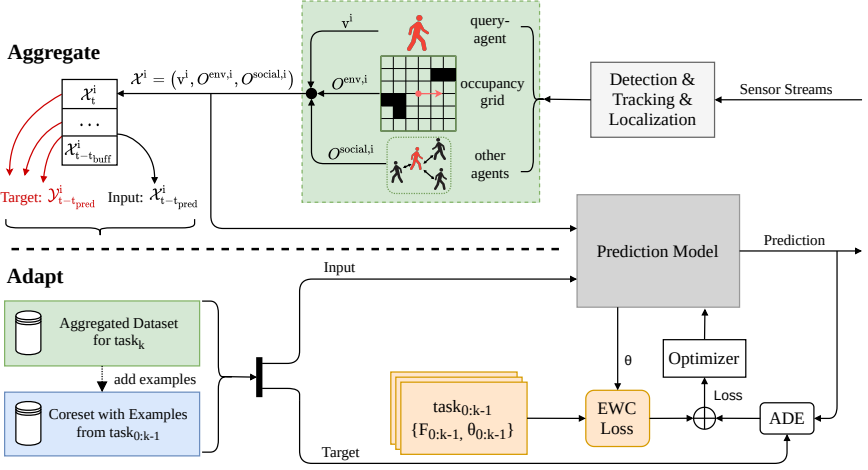
Figure 4.2: Schematics of the SCL framework. The aggregation dataset is collected by extracting examples from the stream of tracked surrounding pedestrians (task aggregation). The prediction model is trained using the aggregated dataset and a separately saved coreset applying an EWC regularization to prevent catastrophic forgetting (model adaption).

4.4.2 Prediction Network Architecture

To evaluate our online learning framework we use a data-driven pedestrian prediction model building on [112]. Please note that our approach does not depend on which network model we use. However, the memory requirements scale linearly with the number of tasks and model parameters. Figure 4.3 shows the network model which uses three streams of information. The first input is the query-agent's velocity over an observation time window t_{obs}, $\mathbf{v}^i_{t-t_{\text{obs}}:t}$, which enables the model to capture the pedestrian's dynamics. The second input is the occupancy grid information $O^{\text{env,i}}_{t-t_{\text{obs}}:t}$ that contains information about the static obstacles centered on the query-agent. In contrast to [112], the third input is a vector containing information about the relative position and velocity of surrounding pedestrians $O^{\text{social,i}}_{t-t_{\text{obs}}:t}$. This
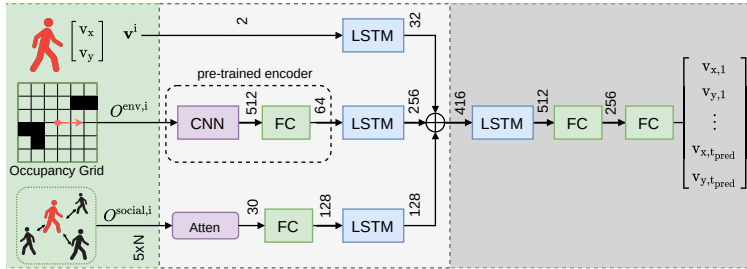

Figure 4.3: Pedestrian motion prediction model architecture.

adaption was done because the model using an angular pedestrian grid, presented in [112], has shown difficulties learning social interactions [127]. For one neighbor pedestrian j the vector including the relative measurements to the query-agent i at time t is

$$\mathbf{e}_t^{i,j} = [\mathbf{p}_t^j - \mathbf{p}_t^i, \mathbf{v}_t^j - \mathbf{v}_t^i].$$

Thus, the information vector at time t is defined as

$$O_t^{\text{social,i}} = [\mathbf{e}_t^{i,1}, \ldots, \mathbf{e}_t^{i,i-1}, \mathbf{e}_t^{i,i+1}, \ldots, \mathbf{e}_t^{i,n}].$$

Hence, the information used for trajectory prediction of pedestrian i is

$$\mathcal{X}_t^i = (\mathbf{v}_{t-t_{\text{obs}}:t}^i, O_{t-t_{\text{obs}}:t}^{\text{env,i}}, O_{t-t_{\text{obs}}:t}^{\text{social,i}}),$$

and the prediction model is given by

$$\hat{\mathbf{v}}_{t+1:t+t_{\text{pred}}}^i = f_{\boldsymbol{\theta}}(\mathbf{v}_{t-t_{\text{obs}}:t}^i, O_{t-t_{\text{obs}}:t}^{\text{env,i}}, O_{t-t_{\text{obs}}:t}^{\text{social,i}}),$$

where the trajectory predictions are represented by a sequence of velocities, i.e. $\hat{\mathcal{Y}}_t^i = \hat{\mathbf{v}}_{t+1:t+t_{\text{pred}}}^i$. We use the permutation invariant *sort* function as an attention mechanism by sorting the relative vectors of surrounding agents by Euclidean distance [142]. To handle a variable number of pedestrians, only the closest n pedestrians are considered. For situations with fewer than n surrounding pedestrians, the relative vector of the closest pedestrian is repeated.

4.4.3 Task Aggregation

For each task k, SCL saves the inputs of the prediction model,

$$\mathcal{X}_t^i = (\mathbf{v}_{t-t_{\text{obs}}:t}^i, O_{t-t_{\text{obs}}:t}^{\text{env,i}}, O_{t-t_{\text{obs}}:t}^{\text{social,i}}),$$

in a buffer for each time step $t = \{-t_{\text{buff}}, \ldots, 0\}$, see Fig. 4.2. Then, for each time step t, the ground truth velocity sequence $\mathbf{v}_{t+1:t+t_{\text{pred}}}^i$ is extracted in hindsight from the buffer and the corresponding input to the prediction model $\mathcal{X}_t^i$ (red arrows in Fig. 4.2). We aggregate the velocity vectors (Target) together with the corresponding model inputs (Input) and store them in the aggregated task dataset $\mathcal{D}_k$ is an example. The examples are aggregated as a sequence. As we use a recurrent prediction model and train the model with truncated back-propagation through time t_{tbptt}, we only aggregate sequences of examples with a length of $t_{\text{buff}} = t_{\text{pred}} + t_{\text{tbptt}}$. We collect training examples for T seconds.

4.4.4 Model Adaption

We present the overall SCL procedure in Algorithm 2. For each task, we aggregate a dataset $\mathcal{D}_k$ over T seconds. Then, the model is adapted using $\mathcal{D}_k$ and a set containing examples of previous tasks referred to as coreset $\mathcal{D}_{\text{coreset}}$. The *Coreset Rehearsal* strategy is applied to mitigate forgetting. Thus, the training dataset is defined as follows:

$$\hat{\mathcal{D}} = \mathcal{D}_k \bigcup \mathcal{D}_{\text{coreset}}.$$

Algorithm 2 The Self-Supervised Continual Learning (SCL) Framework

---

1: Load pretrained model: $f_{\boldsymbol{\theta}}$
2: Load map: $\mathcal{S}$
3: Initialize coreset: $\mathcal{D}_{\text{coreset}} \leftarrow \emptyset$
4: **for** $k = 0$ to ∞ **do**
5:     Initialize the empty task dataset: $\mathcal{D}_k \leftarrow \emptyset$
6:     Aggregate examples for T seconds as follows:
7:     **for** $t = 0$ to T **do**
8:         Process pedestrian positions $\mathbf{p}_t^i$, velocities $\mathbf{v}_t^i$, and the occupancy grid $O_t^{\text{env},i}$ to model inputs $\mathcal{X}_t^i$ and save them to a buffer for $i \in \{1, \dots, n\}$
9:         Get examples from buffer: $\mathcal{E}_t = \{(\mathcal{X}_t^1, \mathcal{Y}_t^1), \dots, (\mathcal{X}_t^n, \mathcal{Y}_t^n)\}$
10:        Update task dataset: $\mathcal{D}_k \leftarrow \mathcal{D}_k \bigcup \mathcal{E}_t$
11:    **end for**
12:    Combine coreset and task: $\hat{\mathcal{D}} \leftarrow \mathcal{D}_k \bigcup \mathcal{D}_{\text{coreset}}$
13:    Train prediction model $f_{\boldsymbol{\theta}}$ on $\hat{\mathcal{D}}$ using EWC
14:    Save EWC importances F_k and the updated parameters $\boldsymbol{\theta}_k$ of task_k
15:    Update coreset $\mathcal{D}_{\text{coreset}}$ with M random examples from $\mathcal{D}_k$
16:    Clear $\mathcal{D}_k$, $\hat{\mathcal{D}}$ from memory
17: **end for**

In the model adaptation phase, SCL uses the training dataset $\hat{\mathcal{D}}$ to train the network for Q epochs. The training loss is composed by a prediction loss and a regularization loss to avoid catastrophic forgetting: $\mathscr{L}_{\text{train}} = \mathscr{L}_{\text{pred}} + \mathscr{L}_{\text{reg}}$. We define the prediction loss as the average norm between the predicted velocity sequence and the ground truth:

$$\mathscr{L}_{\text{pred}}(\hat{\mathcal{Y}}_t^i, \mathcal{Y}_t^i) = \frac{1}{t_{\text{pred}}} \sum_{\tau=t+1}^{t+t_{\text{pred}}} |\hat{\mathbf{v}}_\tau^i - \mathbf{v}_\tau^i|^2. \tag{4.2}$$

We employ EWC [120] as regularization loss method to preserve prediction performance on the previous tasks ($\text{task}_{0:k-1}$) and overcome catastrophic forgetting. EWC penalizes the distance between the new model parameters, $\boldsymbol{\theta}$, and the previous task parameters, $\boldsymbol{\theta}_{0:k-1}$, depending on their importance to keep the knowledge of previous tasks. After learning each task, EWC computes the corresponding importance parameter by using the diagonal elements of the FIM F, which are defined as:

$$F_{k,jj} = \frac{1}{|\mathcal{D}_k|} \sum_{\mathcal{X} \in \mathcal{D}_k} \left(\frac{\delta \log f_{\boldsymbol{\theta}}(\mathcal{X})}{\delta \theta^j} \bigg|_{\boldsymbol{\theta} = \boldsymbol{\theta}_k^*} \right)^2, \tag{4.3}$$

where k and j represent the task and parameter number, respectively, $\mathcal{D}_k$ is the training data containing trajectories from task k, $f_{\boldsymbol{\theta}}(\mathcal{X})$ is the predicted output of the network with parameters $\boldsymbol{\theta}$ given data $\mathcal{X} \in \mathcal{D}_k$. The importance measure F_k is saved together with the network weights $\boldsymbol{\theta}_k$. Based on $F_{0:k-1}$ and $\boldsymbol{\theta}_{0:k-1}$ the

following regularization term is added to the loss function:

$$\mathscr{L}_{\text{reg}}(\theta) = \frac{\lambda}{2} \sum_{l=0}^{k-1} \sum_{j} F_{l,jj}(\theta^j - \theta_l^j)^2, \tag{4.4}$$

where $\boldsymbol{\theta}$ is the current set of weights for the current task k and λ is the hyperparameter that dictates how important not forgetting the old task is compared to learning the new one.

After the model adaptation phase is completed, we update the coreset with M examples of the latest task (task_k). Importantly the new examples replace existing ones to ensure the coreset remains of constant length N. We randomly select which examples to drop to update the coreset. After training, the data sets $\mathcal{D}_k$ and $\hat{\mathcal{D}}$ are cleared.

4.5 Results

In this section, we present quantitative and qualitative results in both simulation and real-world experiments.

4.5.1 Experimental Setup

The prediction model parameters are displayed in Fig. 4.3. We pre-train the prediction model on the ETH and UCY pedestrian datasets [128, 129] for 60 epochs. Our online learning framework will improve this pre-trained model based on the behavior of surrounding pedestrians. The applied hyperparameters are summarized in Tab. 4.1. Note that although $t_{\text{obs}} = 0$, the past states are implicitly taken into account through the internal memory of the LSTMs. First, we evaluate our framework in simulation assuming full knowledge of the map and current states of all pedestrians. The pedestrian behavior is simulated using the SFM [121] and Reciprocal Velocity Obstacle (RVO) model [9]. We train the prediction model incrementally on arbitrary orders of these environments. To evaluate how well our framework scales to complex scenarios with more pedestrians we rerun the above experiments in simulation with an increased number of pedestrians.

Then, we apply SCL in real-world experiments. Here, the true pedestrian behavior differs from the models assumed during simulation. To eliminate the perception-related errors as much as possible, we first test our framework with an optical

Table 4.1: Hyperparameters.

timestep	0.2 s	# training epochs Q	250
task length T	200 s	learning rate	2×10^{-3}
buffer size t_{buff}	6 s	L2 regularization	5×10^{-4}
predict. time t_{pred}	3 s	EWC parameter λ	1×10^6
tbptt time t_{tbptt}	3 s	coreset size N/update size M	100/ 20
observ. time t_{obs}	0 s	validation set size L_v	100

tracking system (Optitrack) that provides pose information of all tracked pedestrians. We set up three scenarios to replicate the simulation environments. Finally, we evaluate our framework in an uncontrolled hall using only the on-board sensing and, a detection and tracking pipeline.

Baseline Methods

We evaluate our method against three baseline approaches in both simulation and real-world experiments:

1. **Offline:** The prediction model is trained offline on all tasks. This baseline represents a performance upper-bound assuming that all data is available.

2. **Vanilla:** The prediction model is trained using only the aggregated data and standard gradient descent without any regularization loss.

3. **EWC:** The prediction model is trained using only the aggregated data with EWC regularization, but without coreset rehearsal.

Tasks

In simulation, we additionally consider the following baselines:

- **Coreset:** The prediction model is trained using the aggregated data and coreset data.

- **CV:** The human behavior is predicted using the Constant Velocity (CV) model, no learning is applied.

The CV model was added since it was shown to outperform state-of-the-art data-based prediction models [143] and to enable robust navigation around humans [144]. A limitation of the CV model is that it does not consider obstacles.

Since our focus is on applying continual learning strategies to improve pedestrian prediction models on the fly without forgetting, we only change the learning strategy across baselines and keep the prediction network architecture fixed. Similar to other works on pedestrian prediction models, we use the Average Displacement Error (ADE) and Final Displacement Error (FDE) as performance metrics [127, 142].

We consider three distinct environments, i.e., tasks, displayed in Fig. 4.4:

1. **Square:** An infinite corridor setting with three pedestrians walking clockwise and three anticlockwise.

2. **Obstacles:** Pedestrians walking towards each other in an obstacle filled space.

3. **Hall:** Pedestrians walking towards each other in a hall while behaving cooperatively.

The scenarios were selected since they include encounters typically experienced in everyday situations. The specific environments were chosen to investigate social interactions (Hall), obstacle interactions (Obstacle) and semantic knowledge of the
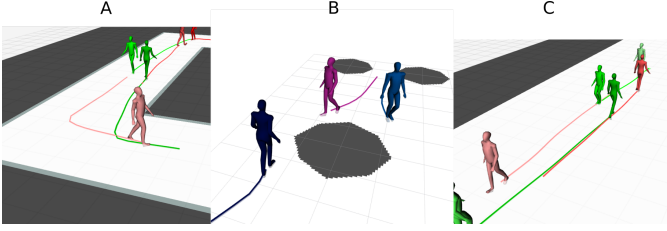
Figure 4.4: The considered simulation environments consist of (A) Square, (B) Obstacle, and (C) Hall environments.

map (Square). To additionally evaluate our framework in scenarios with more interacting agents we consider the above environments with 10 and 20 pedestrians. For the obstacle-free environments, we use the open-source pedsim simulation framework[1] employing the Social Force Model (SFM) [121] to simulate the pedestrian behavior. For environments with static obstacles, we employ the RVO method [9][2] as pedestrians following the SFM may still collide with obstacles.

4.5.2 Simulation Results

We evaluate the prediction performance of the network model trained with our method (SCL) versus the baselines on different sequences of environments (square, obstacle, hall) starting from a pre-trained model. Each environment is observed for T seconds to create the aggregated dataset on which the model is trained. Thus, each environment corresponds to a new task. To compute the ADE/FDE performance metrics, we collect a validation set for each environment including L_v examples not used during training.

Table 4.2 reports the mean and standard deviation (std) of ADE/FDE evaluated at the end of the sequence on all three environments, under the columns denoted by *seq. end*. The columns denoted by *forgotten* report the mean and std of ADE/FDE increase of the prediction model on previous environments after training on new environments. It can be seen that SCL outperforms Vanilla. The significant increase in mean forgotten ADE/FDE for Vanilla indicates that naive online training over changing environments using standard gradient descent results in catastrophic forgetting. Our method is independent of sequence order, arriving at within ±0.02 of the same mean ADE/FDE for all orders. To gain insight into where catastrophic forgetting occurs, we save the models trained for the sequence (square → obstacle → hall) after each training step and apply them to the validation set of the square scenario only. Figure 4.5 compares the performance of the different training methods on the square scenario validation set at each training step. By evaluating a single environment over time, we can clearly visualize when and how much the models degraded in prediction performance in the respective environment. For ease

---

1 `https://github.com/srl-freiburg/pedsim_ros`

2 `https://github.com/sybrenstuvel/Python-RVO2`

Table 4.2: Quantitative results of the CV prediction and Vanilla, EWC, Coreset and SCL training approaches for four environment sequences. The results for the dense scenarios are included as SCL-10 and SCL-20. The table lists the mean±standard deviation (std) of ADE/FDE, for all environments at the sequence end under *seq. end* and the mean±std of the *forgotten* ADE/FDE, which refers to the average increase in ADE/FDE on previous environments across the learning sequence. All error measures are presented in meters.

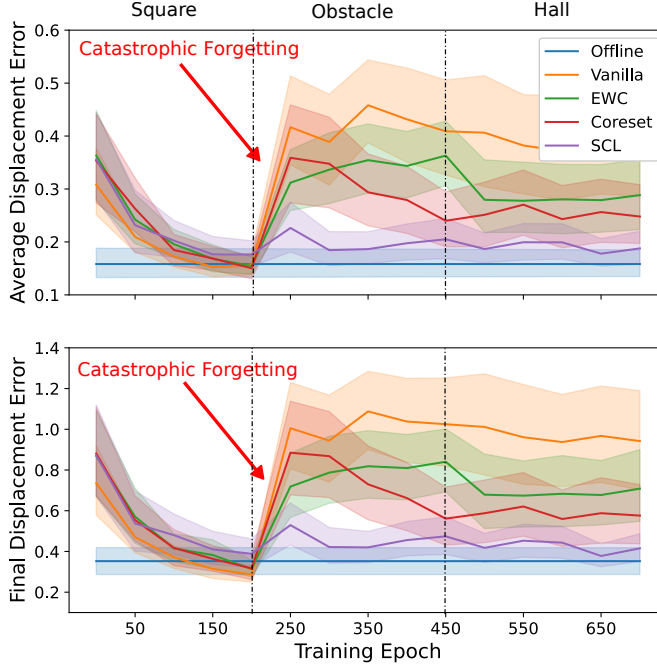| Method	square → obstacle → hall		obstacle → square → hall		hall → obstacle → square		obstacle → hall → square	
	forgotten (mean±std)	seq. end (mean±std)	forgotten (mean±std)	seq. end (mean±std)	forgotten (mean±std)	seq. end (mean±std)	forgotten (mean±std)	seq. end (mean±std)
CV	+0.00 ± 0.00/ +0.00 ± 0.00	1.12 ± 1.18/ 1.09 ± 1.69	+0.00 ± 0.00/ +0.00 ± 0.00	1.25 ± 1.29/ 1.10 ± 1.65	+0.00 ± 0.00/ +0.00 ± 0.00	1.12 ± 1.17/ 1.16 ± 1.91	+0.00 ± 0.00/ +0.00 ± 0.00	1.26 ± 1.31/ 1.24 ± 1.92
Vanilla	+0.12 ± 0.29/ +0.31 ± 0.74	0.21 ± 0.31/ 0.49 ± 0.79	+0.10 ± 0.23/ +0.27 ± 0.62	0.21 ± 0.27/ 0.47 ± 0.63	+0.20 ± 0.27/ +0.51 ± 0.62	0.27 ± 0.26/ 0.62 ± 0.58	+0.18 ± 0.20/ +0.52 ± 0.51	0.26 ± 0.21/ 0.63 ± 0.51
EWC	+0.10 ± 0.25/ +0.28 ± 0.67	0.19 ± 0.25/ 0.46 ± 0.66	+0.05 ± 0.13/ +0.12 ± 0.37	0.17 ± 0.13/ 0.37 ± 0.35	+0.12 ± 0.17/ +0.33 ± 0.44	0.22 ± 0.16/ 0.51 ± 0.41	+0.10 ± 0.18/ +0.27 ± 0.47	0.21 ± 0.19/ 0.48 ± 0.45
Coreset	+0.03 ± 0.09/ +0.09 ± 0.27	**0.16 ± 0.12**/ **0.36 ± 0.34**	+0.05 ± 0.11/ +0.12 ± 0.29	0.17 ± 0.14/ 0.38 ± 0.34	+0.03 ± 0.10/ +0.08 ± 0.25	**0.17 ± 0.13**/ **0.36 ± 0.28**	**+0.04 ± 0.09**/ +0.12 ± 0.24	0.19 ± 0.15/ 0.40 ± 0.31
SCL	**+0.02 ± 0.10**/ **+0.07 ± 0.29**	0.16 ± 0.14/ 0.36 ± 0.40	**+0.01 ± 0.08**/ **+0.04 ± 0.20**	**0.15 ± 0.12**/ **0.34 ± 0.27**	**+0.03 ± 0.08**/ **+0.07 ± 0.20**	**0.17 ± 0.12**/ 0.37 ± 0.30	**+0.04 ± 0.09**/ **+0.08 ± 0.21**	**0.17 ± 0.13**/ **0.36 ± 0.28**
SCL-10	+0.00 ± 0.10/ +0.00 ± 0.23	0.20 ± 0.17/ 0.45 ± 0.40	+0.01 ± 0.12/ +0.03 ± 0.25	0.20 ± 0.16/ 0.44 ± 0.37	+0.00 ± 0.08/ +0.01 ± 0.19	0.20 ± 0.15/ 0.45 ± 0.33	+0.01 ± 0.10/ +0.01 ± 0.26	0.20 ± 0.14/ 0.44 ± 0.33
SCL-20	+0.04 ± 0.12/ +0.10 ± 0.31	0.22 ± 0.19/ 0.49 ± 0.42	+0.02 ± 0.10/ +0.05 ± 0.25	0.20 ± 0.16/ 0.46 ± 0.37	+0.04 ± 0.11/ +0.08 ± 0.26	0.21 ± 0.18/ 0.47 ± 0.40	+0.03 ± 0.12/ +0.06 ± 0.28	0.22 ± 0.18/ 0.50 ± 0.41

Figure 4.5: Prediction performance of models trained on square → obstacle → hall sequence evaluated on **only** the square scenario. Shows ADE (top) and FDE (bottom) of all training methods on the validation set of the square task, while learning new tasks. Note that the offline model is added for comparison purposes only.

Table 4.3: Statistical Significance Analysis using the Mann-Whitney U test. Comparison of SCL's performance (i.e., ADE and FDE) against all baselines on each environment for the obstacle hall square sequence. Significant results are displayed in bold considering a 5% confidence-level.

Method	obstacle		hall		square	
	ADE	FDE	ADE	FDE	ADE	FDE
CV	**p = 0.00**	**p = 0.00**	**p = 0.00**	**p = 0.00**	**p = 0.00**	**p = 0.00**
Vanilla	**p = 0.00**	**p = 0.00**	**p = 0.00**	**p = 0.00**	p = 0.97	p = 0.53
EWC	p = 0.06	**p = 0.02**	**p = 0.04**	**p = 0.01**	p = 0.65	p = 0.71
Coreset	p = 0.29	p = 0.09	p = 0.63	p = 0.68	p = 0.22	p = 0.33

of comparison, the offline trained model is also plotted as a constant line. In the first section, all models are trained on the aggregated dataset of the square scenario and, as expected, the error measures decrease for all online learning methods reaching better performance than offline trained prediction model due to overfitting. However, when changing from the square environment to the obstacle environment, the ADE/FDE performance quickly and drastically degrades for the Vanilla baseline (red arrow). It can be seen that using EWC to selectively slow down learning on important parameters helps to significantly mitigate the magnitude of the loss in ADE/FDE. Nevertheless, after two subsequent tasks, the EWC baseline performed $\sim 30\%$ worse on FDE and $\sim 20\%$ worse on ADE. Rehearsing a set of past examples enables to retain more knowledge after two subsequent tasks than applying EWC. Combining EWC and the coreset rehearsal as done in SCL helps to further mitigate forgetting. SCL was able to train in two subsequent scenarios while retaining knowledge of the initially experienced scenario.

We performed pair-wise Mann-Whitney U tests between our proposed method and each baseline to evaluate the statistical significance of the presented results. Tab. 4.3 shows the *p*-values comparing the performance results (i.e., ADE and FDE) on each scenario for the obstacle → hall → square sequence. SCL significantly outperforms CV on all environments, the Vanilla baseline on all past environments, and EWC on one environment. Rehearsing alone achieves marginally worse results than SCL. Please note that the presented results consider a limited set of environments with limited complexity. We expect that as the number of scenarios and complexity increase, differences in performance between the baselines become significant.

4.5.3 Dense Scenarios

To evaluate how well our framework scales to complex scenarios with more pedestrians we employ the above simulation environments with increased numbers of pedestrians ($n = \{10, 20\}$). The results are presented in Tab. 4.2. It can be seen that SCL scales well to dense scenarios with more agents achieving similar performance for 10 and 20 pedestrians. The forgotten ADE/FDE even decreases for some sequences, indicating that observing more pedestrians can improve the preservation of past experiences.

Table 4.4: Quantitative results of Vanilla, EWC and SCL on real-world data collected using an optical tracking system. The table lists the mean±standard deviation (std) of ADE/FDE, on all environments at the sequence end under *seq. end* and the mean±std of the *forgotten* ADE/FDE, which refers to the average increase in ADE/FDE, on previous environments across the learning sequence. All error measures are presented in meters.

Method	square → obstacle → coop.		obstacle → square → coop.	
	forgotten (mean±std)	seq. end (mean±std)	forgotten (mean±std)	seq. end (mean±std)
Vanilla	+0.24±0.28/ +0.58±0.67	0.46±0.29/ 0.97±0.66	+0.21±0.26/ +0.50±0.64	0.45±0.29/ 0.94±0.63
EWC	+0.19±0.29/ +0.42±0.67	0.43±0.27/ 0.86±0.61	+0.12±0.23/ +0.31±0.58	0.41±0.25/ 0.87±0.56
SCL	**+0.04±0.21/ +0.13±0.50**	**0.36±0.23/ 0.73±0.56**	**+0.05±0.22/ +0.11±0.50**	**0.40±0.28/ 0.80±0.60**

4.5.4 Real-world Results

We first evaluate our method in real-world experiments assuming perfect perception capabilities by using an external high-precision Optitrack tracking system. Secondly, we use the robot's on-board sensing capabilities combined with a detection and tracking pipeline.

Perfect Perception

To evaluate our framework using the Optitrack system, we set up three environments to replicate the ones considered in simulation (i.e., square, obstacle, cooperative). Each environment is observed for T seconds. Table 4.4 reports quantitative results on two different sequence orders similar to Tab. 4.2. Our framework significantly outperformed the Vanilla baseline on both metrics indicating that we can not only learn a prediction model from real human motion but also that we need to consolidate the learned knowledge. SCL was able to improve prediction performance and learn certain concepts, such as avoiding crashing into walls, pillars, or fences. Figure 4.6 shows a qualitative example of the experiment, where our framework learns to avoid both static obstacles and pedestrians.

On-board Perception

We now evaluate our framework in an uncontrolled hall environment using the robot's detection and tracking pipeline (i.e., LiDAR and cameras). In Fig. 4.7 we show qualitative results of the experiments with a moving robot. The fact that the robot is constantly moving reduced the average collected trajectory length of the interacting pedestrians making the prediction problem harder. Thus, employing SCL in more dense environments is expected to further improve the resulting prediction performance. Nevertheless, the prediction model learned online when pedestrians are likely to take corners, by observing how real pedestrians walk in the environment. Note that the ETH and UCY datasets, on which our model was pre-trained, contain almost no interactions with static obstacles, yet our framework
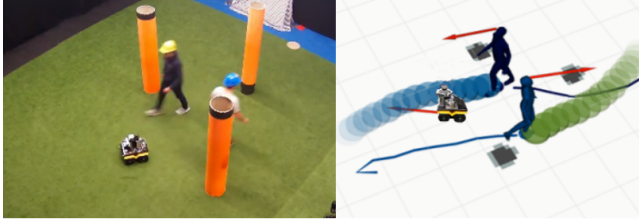
Figure 4.6: Real-world validation using an Optitrack system that streams the pedestrian states. The predicted pedestrian trajectories are depicted as green and blue disks.

Figure 4.7: Map view of the real-world application of SCL on moving robot using on-board perception. The green and blue disks depict the predicted trajectories employing the pre-trained model and the SCL-trained model, respectively. The red dotted lines depict the pedestrians' past trajectories. The pedestrians' and robot's future trajectories are shown as solid red lines.

autonomously learns obstacle interactions. Furthermore, the occupancy map shown in Fig. 4.7 is generated by the robot itself using the depth information from its Li-DAR. Thus, our framework can continuously learn in new and unseen environments autonomously.

4.6 Conclusion & Future Work

This chapter introduces a Self-Supervised Continual Learning (SCL) framework to improve pedestrian prediction models using online streams of data. We combined Elastic Weight Consolidation (EWC) and the rehearsal of a small constant sized set of examples to overcome catastrophic forgetting. We showed through experiments that SCL significantly outperforms vanilla gradient descent and performs similarly to offline trained models with full access to pedestrian data in all considered environments. Additionally, we showed in real-world experiments that our pedestrian prediction model can learn to generalize to new and unseen environments over time. Future work can investigate different methods to determine when the model should be updated, how different pedestrian behavior types could be integrated into our framework and the integration of our approach with a motion planner to improve the interaction-awareness between pedestrians and the robot.

5

Conclusion and Future Work

Z ooming out, this final chapter reflects on the broader picture by summarizing the key contributions of the thesis, discussing their significance, and outlining potential directions for future research.

5.1 Conclusion

5.1.1 Summary

This thesis was motivated by the challenges faced in local motion planning for autonomous robots in human-centered environments, i.e., environments specifically designed for humans. These challenges arise from the complexity and unpredictability of human behavior, stringent safety requirements, the desired adherence to social norms, and the ability to adapt to ever-changing environments. While safety is generally a fundamental aspect of local motion planning and interaction awareness is important when navigating among decision-making agents such as robots or humans, human-centered environments demand an additional focus on socially compliant behavior that is comfortable and acceptable to humans. Additionally, adaptability is key, as humans often adjust their behavior over time in response to the robot. Altogether, the key elements of local motion planning in these environments can be summarized as **interaction awareness**, **safety**, **social compliance**, and **adaptability**.

Existing approaches to motion planning lack several key features. On the one hand, there are optimization-based approaches such as Model Predictive Control (MPC), which generate control inputs by solving an optimization problem subject to explicit constraints. While optimization-based methods can offer safety guarantees, the common approach of splitting motion planning into trajectory prediction and trajectory planning often results in behaviors that are opaque and hard to predict. Additionally, game-theoretic approaches, while modeling interactions, do not scale well to scenarios involving many agents. On the other hand, learning-based methods implicitly capture interactions. Reinforcement Learning (RL) methods, although effective during runtime, lack safety guarantees and, like optimization-based methods, require the desired behavior to be defined explicitly through a cost/reward function. Imitation Learning (IL) methods eliminate the need to explicitly define a cost function, but they typically lack formal safety guarantees.

Thus, we raised the question: How can we achieve interaction-aware, safe, socially compliant, and adaptive local motion planning for robots in human-centered environments? We can now provide answers based on the findings of this thesis:

Interaction Awareness. Initially, we considered the decoupled motion planning approach, which separates the tasks of trajectory prediction and trajectory planning. Focusing on navigation among pedestrians, we leverage the interpretable and established Social Force Model (SFM) to model the human response dynamics to robot actions. While this approach allows the robot to be aware of its influence on the other agents in the environment, it still requires a formulation of the desired behavior. If the cost is formulated to reach the goal as fast as possible, the robot will exhibit behaviors that exploit the cooperation of the pedestrians, which can be perceived as unnatural or even dangerous by humans.

Social Compliance and Safety. This led us to ask: How can we design motion planning systems that integrate socially compliant and interaction-aware behaviors while maintaining safety guarantees? IL offers a way to achieve socially compliant and interaction-aware behaviors by through learning from demonstrated, interaction-aware behaviors. However, IL, specifically approaches that directly learn a policy, lack formal safety guarantees, making them unsuitable for safety-critical environments. To address this, we introduced Robust Policy Control Barrier Functions (RPCBFs), a practical framework for constructing Control Barrier Function (CBF) approximations used to define a Safety Filter (SF). We demonstrated through simulations and real-world experiments that our approach enhances safety and provides a more accurate estimation of the robust control-invariant set.

Adaptability. Lastly, we tackled the problem of adaptability in changing environments. For instance, when a robot operates in a new environment or lacks prior data on the present pedestrian behavior, it must be able to adapt to the new setting. We specifically considered the scenario of pedestrian prediction models, which are typically trained offline on commonly available datasets and do thus not adapt to unseen behaviors. To overcome this limitation, we proposed a self-supervised continual learning framework that leverages observed pedestrian behavior from the perception pipeline to refine and improve the prediction models iteratively. Our framework combines regularization techniques with a rehearsal strategy to mitigate catastrophic forgetting, a phenomenon which refers to Neural Networks forgetting previously acquired knowledge while learning new concepts. Our experiments showed that using our framework, the pedestrian prediction model can progressively generalize to new and unseen environments, showing its capability to adapt. While we looked at the problem from a perspective focusing on pedestrian prediction, the proposed framework is general and can be applied to other tasks that require continual learning. For instance, the framework can be used to improve the performance of IL policies by continually learning from new observations.

5.1.2 Discussion

Reflecting on Fig. 1.2 and Chapter 1, this thesis has addressed various challenges in motion planning within human-centered environments, tackling key aspects both individually and in combination. However, the explicit consideration of social compliance has not been directly addressed. While IL is a promising approach to learning socially compliant behavior, this thesis instead focused on ensuring that policies learned from demonstrations can be applied safely. Additionally, this thesis did not propose a comprehensive framework that seamlessly integrates interaction awareness, safety, social compliance and adaptability. Nonetheless, the methods developed in this work lay the groundwork for such a holistic approach. The SF offers a practical and interpretable approach to address safety and can be used to modify potentially unsafe policies learned from demonstrations. The continual learning framework enables the system to dynamically adapt to new environments and tasks. Together, these components form a solid foundation for the future development of an integrated framework capable of addressing the multifaceted demands of human-centered motion planning. While this work focused on the problem from

a navigation perspective, the proposed approaches to SFs and continual learning are general and can be applied to other tasks. This opens the door for future research to explore the application of these methods in other domains, such as manipulation or mobile manipulation tasks. The following section discusses this and other directions for future work.

5.2 Future Work

The goal of this thesis was to advance the application of autonomous robots in human-centered environments, transitioning the field from its infancy to a more mature and practical stage. However, motion planning in human-centered environments remains an ongoing challenge. Below, we outline some of the key challenges that remain and suggest potential directions for future research:

A holistic framework: As discussed before, this thesis provides components for a holistic framework that integrates interaction awareness, safety, social compliance, and adaptability. Future work should focus on developing a comprehensive framework that seamlessly integrates these components to address the multifaceted demands of human-centered motion planning.

Safety: As mentioned, providing safety guarantees is always contingent on the assumptions made in the problem formulation. Future work should focus on developing methods that provide safety guarantees under more general assumptions. Additionally, the introduced Robust Policy Control Barrier Function (RPCBF) is only an approximation of the true Control Barrier Function (CBF), as are neural network-based CBFs. While these approximations provide a practical approach to addressing safety in dynamic environments, a more rigorous analysis of the provided safety guarantees is needed. Future work could explore a deeper evaluation of the safety assurances provided by RPCBFs and the development of more informed strategies for sampling disturbance trajectories. For instance, by updating the assumptions on the disturbances based on the observed behavior. Furthermore, it should be investigated how the policy used for evaluation should be selected to ensure that the resulting SF is a valid CBF while reducing conservativeness. Other future work could focus on addressing safety beyond collision avoidance, improving the efficiency of verification tools for neural network-based CBFs and developing methods that integrate learning-based and model-based safety approaches.

Other Applications: This thesis addressed the problem from a navigation perspective, as does much of the existing work. Future work should explore applying the proposed methods to other domains, such as mobile manipulation tasks.

Evaluation of Socially-Compliant Behaviors in Real-World Scenarios: The lack of comprehensive evaluation frameworks hinders research on socially compliant behaviors. While benchmark datasets are available in fields like computer vision, evaluating autonomous robots in real-world environments presents challenges, including the need for ethics approval, reliable localization, and perception systems. As a result, many approaches are typically evaluated in controlled lab environments,

limiting the scope and applicability of the findings. Future work should focus on developing a comprehensive evaluation framework that enables easier testing of autonomous robots in human-centered environments. Furthermore, closer collaboration between researchers from robotics and human-robot interaction fields is needed to develop standardized evaluation metrics that can be used across different research groups.

5

But that is another story and shall be told another time.

Michael Ende, The Neverending Story

Appendices

A

Complementary Projects

A.1 Simultaneous Synthesis and Verification of Neural Control Barrier Functions

Chapter 3 introduced a practical, interpretable method for constructing approximations of robust CBFs. In contrast, neural CBFs leverage the universal approximation capabilities of neural networks to learn CBFs candidates. However, this requires their certification as valid CBFs, which is the focus of this work. Unlike Chapter 3, this paper does not account for disturbances.

We leverage bound propagation techniques and the Branch-and-Bound scheme to efficiently verify that a neural network satisfies the conditions to be a CBF over the continuous state space. To accelerate training, we present a framework, see Figure A.1, that embeds the verification scheme into the training loop to synthesize and verify a neural CBF simultaneously. In particular, we employ the verification scheme to identify partitions of the state space that are not guaranteed to satisfy the CBF conditions. We then expand the training dataset by incorporating additional data from these partitions, referred to as counter-examples (CE). The neural network is then optimized using the augmented dataset to meet the CBF conditions. We show that our framework can efficiently certify a neural network as a CBF for a non-linear control-affine system and render a larger safe set than state-of-the-art neural CBF works. We further employ our learned neural CBF to derive a safe controller to illustrate the practical use of our framework.
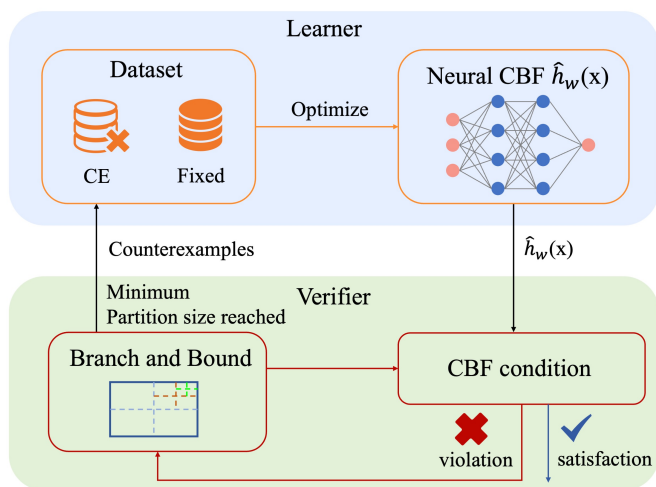
A

Figure A.1: A schematic overview of the presented Branch-and-Bound Verification-in-the-Loop Training.

A.2   Current-Based Impedance Control for Interacting with Mobile Manipulators

While Chapter 3 and Appendix A.1 address safety through set invariance using SFs based on CBFs, an alternative approach is compliant control. Unlike motion control, which regulates a robot's movement, compliant control focuses on controlling the applied forces to achieve safer interactions with the environment. The work presented here adapts impedance control for mobile manipulators consisting of off-the-shelf components without the use of force or torque sensors.

A calibration method is designed that enables estimation of the actuators' current/torque ratios and frictions, used by the adapted impedance controller, and that can handle model errors. The calibration method and the performance of the designed controller are experimentally validated using the Kinova GEN3 Lite arm. Results show that the calibration method is consistent and that the designed controller for the arm is compliant while also being able to track targets with five-millimeter precision when no interaction is present. Additionally, this paper presents two operational modes for interacting with the mobile manipulator: one for guiding the robot around the workspace through interacting with the arm and another for executing a tracking task, both maintaining compliance to external forces, see Figure A.2. These operational modes were tested in real-world experiments, affirming their practical applicability and effectiveness
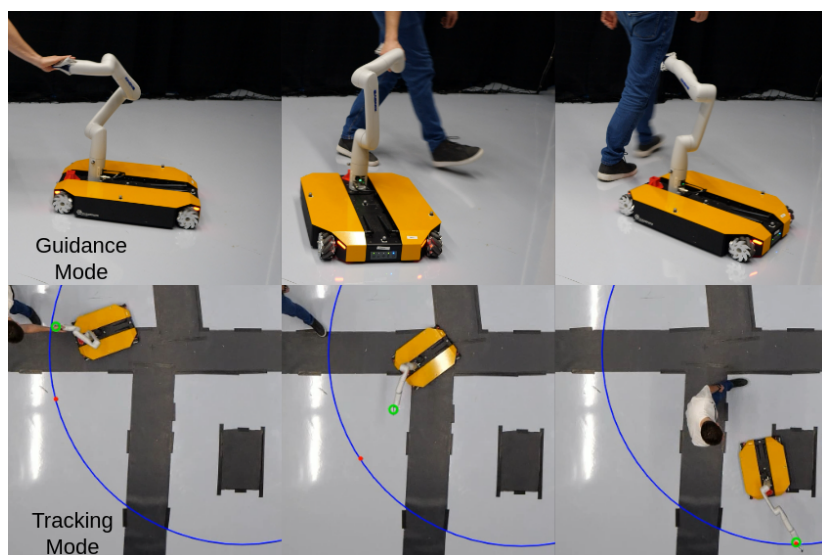
A

Figure A.2: Compliance-enabled operational modes implemented on mobile manipulator without force/torque sensors. Guidance mode: mobile manipulator is led through inter-action with arm. Tracking mode: end-effector (green circle) tracks target (red dot) while being compliant with user interactions

A.3 Multi-Robot Local Motion Planning Using Dynamic Optimization Fabrics

In contrast to the previous works, this study addresses multi-robot local motion planning in close proximity. While there are parallels to motion planning in human-centered environments, assuming the motion policies of the other robots are (partly) known is a valid and simplifying assumption. This allows us to focus on the complexities of systems with many degrees of freedom (DOF). We approach this using geometric fabrics, extended to multi-robot systems.

Geometric approaches, like Riemannian Motion Policies and geometric fabrics, offer better scalability than optimization-based methods like Model Predictive Control. By integrating components like collision avoidance with Riemannian metrics, fabrics enable symbolic pre-computation, reducing computational costs. They also promote asymptotic stability, making them ideal for reactive motion planning. While successful in single-robot applications, fabrics are prone to local minima, which can lead to deadlocks in multi-robot scenarios.

This work develops an online local motion planning algorithm for multiple high-DOF manipulators in shared workspaces, focusing on geometric fabrics for close-proximity pick-and-place tasks (see Figure A.3). We introduce Rollout Fabrics (RF), which simulates multi-robot fabric motion over a prediction horizon to detect and resolve deadlocks by adapting goal-reaching parameters.
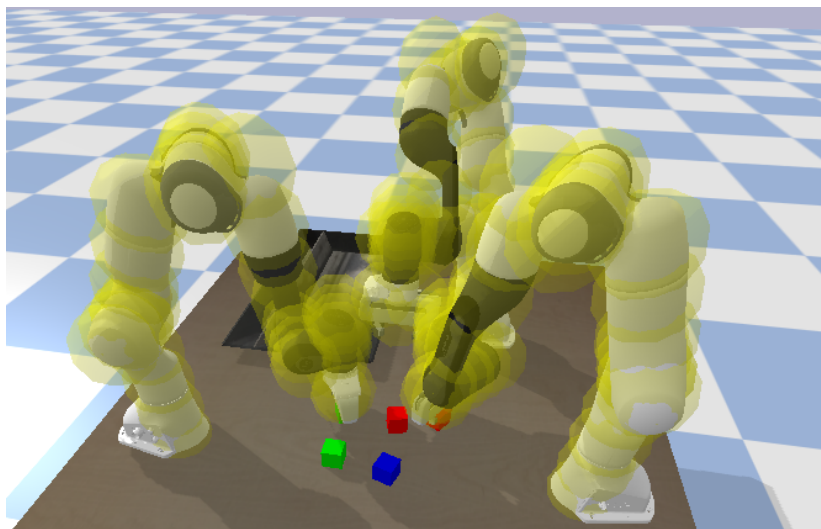
A

Figure A.3: Multi-robot pick-and-place scenario in close proximity. Franka Emika Pandas pick cubes avoiding collisions.

Bibliography

[1] S. Schaal. "The new robotics—towards human-centered machines". In: *HFSP journal* 1.2 (2007), pp. 115–126.

[2] G. Fragapane, H.-H. Hvolby, F. Sgarbossa, and J. O. Strandhagen. "Autonomous mobile robots in hospital logistics". In: *IFIP International Conference on Advances in Production Management Systems*. Springer. 2020, pp. 672–679.

[3] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch. "Human-aware robot navigation: A survey". In: *Robotics and Autonomous Systems* 61.12 (2013), pp. 1726–1743.

[4] K. Dautenhahn. "Socially intelligent robots: dimensions of human–robot interaction". In: *Philosophical transactions of the royal society B: Biological sciences* 362.1480 (2007), pp. 679–704.

[5] ABB. *ABB demonstrates concept of mobile laboratory robot for Hospital of the Future*. 2019. URL: `https://new.abb.com/news/detail/37301/abb-demonstrates-concept-of-mobile-laboratory-robot-for-hospital-of-the-future` (visited on 05/29/2025).

[6] D. Grozdev. *Tokyo Shibuya crossing during rush hour*. 2018. URL: `https://www.youtube.com/watch?v=Bi61ue_cpMY` (visited on 05/29/2025).

[7] W. Schwarting, J. Alonso-Mora, and D. Rus. "Planning and decision-making for autonomous vehicles". In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.1 (2018), pp. 187–210.

[8] D. Fox, W. Burgard, and S. Thrun. "The dynamic window approach to collision avoidance". In: *IEEE Robotics & Automation Magazine* 4.1 (1997), pp. 23–33.

[9] J. Van den Berg, M. Lin, and D. Manocha. "Reciprocal velocity obstacles for real-time multi-agent navigation". In: *2008 IEEE international conference on robotics and automation*. Ieee. 2008, pp. 1928–1935.

[10] 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: The 10th international symposium*. Springer. 2013, pp. 203–216.

[11] S. Quinlan and O. Khatib. "Elastic bands: Connecting path planning and control". In: *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE. 1993, pp. 802–807.

[12] O. Khatib. "Real-time obstacle avoidance for manipulators and mobile robots". In: *The international journal of robotics research* 5.1 (1986), pp. 90–98.

[13] G. Ferrer, A. Garrell, and A. Sanfeliu. "Robot companion: A social-force based approach with human awareness-navigation in crowded environments". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 1688–1694.

[14] F. Leon and M. Gavrilescu. "A review of tracking and trajectory prediction methods for autonomous driving". In: *Mathematics* 9.6 (2021), p. 660.

[15] C. Zhang and C. Berger. "Pedestrian behavior prediction using deep learning methods for urban scenarios: A review". In: *IEEE Transactions on Intelligent Transportation Systems* 24.10 (2023), pp. 10279–10301.

[16] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez. "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges". In: *Information Fusion* 58 (2020), pp. 52–68.

[17] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus. "Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model". In: *IEEE Transactions on Intelligent Transportation Systems* 19.9 (2017), pp. 2994–3008.

[18] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora. "Model predictive contouring control for collision avoidance in unstructured dynamic environments". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4459–4466.

[19] O. De Groot, L. Ferranti, D. M. Gavrila, and J. Alonso-Mora. "Topology-driven parallel trajectory optimization in dynamic environments". In: *IEEE Transactions on Robotics* 41 (2024), pp. 110–126.

[20] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou. "Aggressive driving with model predictive path integral control". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 1433–1440.

[21] G. Williams, A. Aldrich, and E. A. Theodorou. "Model predictive path integral control: From theory to parallel computation". In: *Journal of Guidance, Control, and Dynamics* 40.2 (2017), pp. 344–357.

[22] J. Yin, Z. Zhang, E. Theodorou, and P. Tsiotras. "Trajectory distribution control for model predictive path integral control using covariance steering". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 1478–1484.

[23] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan. "Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state". In: *Autonomous Robots* 42 (2018), pp. 1405–1426.

[24] T. Başar and G. J. Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

[25] L. Streichenberg, E. Trevisan, J. J. Chung, R. Siegwart, and J. Alonso-Mora. "Multi-agent path integral control for interaction-aware motion planning in urban canals". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 1379–1385.

[26] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.* "Mastering the Game of Go without Human Knowledge". In: ().

[27] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.* "Mastering chess and shogi by self-play with a general reinforcement learning algorithm". In: *arXiv preprint arXiv:1712.01815* (2017).

[28] G. Lample and D. S. Chaplot. "Playing FPS games with deep reinforcement learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1. 2017.

[29] K. Zhu and T. Zhang. "Deep reinforcement learning based mobile robot navigation: A review". In: *Tsinghua Science and Technology* 26.5 (2021), pp. 674–691.

[30] 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)*. IEEE. 2017, pp. 285–292.

[31] D. Han, B. Mulyana, V. Stankovic, and S. Cheng. "A survey on deep reinforcement learning algorithms for robotic manipulation". In: *Sensors* 23.7 (2023), p. 3762.

[32] K.-C. Hsu, H. Hu, and J. F. Fisac. "The safety filter: A unified view of safety-critical control in autonomous systems". In: *Annual Review of Control, Robotics, and Autonomous Systems* 7 (2023), pp. 47–72.

[33] S. Li and O. Bastani. "Robust model predictive shielding for safe reinforcement learning with stochastic dynamics". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 7166–7172.

[34] K. P. Wabersich, L. Hewing, A. Carron, and M. N. Zeilinger. "Probabilistic model predictive safety certification for learning-based control". In: *IEEE Transactions on Automatic Control* 67.1 (2021), pp. 176–188.

[35] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters, *et al.* "An algorithmic perspective on imitation learning". In: *Foundations and Trends® in Robotics* 7.1-2 (2018), pp. 1–179.

[36] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. "A survey of robot learning from demonstration". In: *Robotics and autonomous systems* 57.5 (2009), pp. 469–483.

[37] A. Y. Ng, S. Russell, *et al.* "Algorithms for inverse reinforcement learning." In: *Icml*. Vol. 1. 2. 2000, p. 2.

[38] M. Bain and C. Sammut. "A Framework for Behavioural Cloning." In: *Machine Intelligence 15*. 1995, pp. 103–129.

[39] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.* "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018).

[40] R. Pérez-Dattari, C. Della Santina, and J. Kober. "PUMA: Deep Metric Imitation Learning for Stable Motion Primitives". In: *Advanced Intelligent Systems* 6.11 (2024), p. 2400144.

[41] J. Gu, C. Sun, and H. Zhao. "Densetnt: End-to-end trajectory prediction from dense goal sets". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15303–15312.

[42] K. Mangalam, Y. An, H. Girase, and J. Malik. "From goals, waypoints & paths to long term human trajectory forecasting". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15233–15242.

[43] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani. "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9813–9823.

[44] P. Trautman and A. Krause. "Unfreezing the robot: Navigation in dense, interacting crowds". In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2010, pp. 797–803.

[45] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin. "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games". In: *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2020, pp. 1475–1481.

[46] S. Le Cleac'h, M. Schwager, and Z. Manchester. "Lucidgames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5485–5492.

[47] L. Streichenberg, E. Trevisan, J. J. Chung, R. Siegwart, and J. Alonso-Mora. "Multi-agent path integral control for interaction-aware motion planning in urban canals". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 1379–1385.

[48] Y. Chen, S. Veer, P. Karkus, and M. Pavone. "Interactive joint planning for autonomous vehicles". In: *IEEE Robotics and Automation Letters* (2023).

[49] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan. "Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state". In: *Autonomous Robots* 42 (2018), pp. 1405–1426.

[50] J. L. V. Espinoza, A. Liniger, W. Schwarting, D. Rus, and L. Van Gool. "Deep interactive motion prediction and planning: Playing games with motion prediction models". In: *Learning for Dynamics and Control Conference.* PMLR. 2022, pp. 1006–1019.

[51] D. Helbing and P. Molnar. "Social force model for pedestrian dynamics". In: *Physical review E* 51.5 (1995), p. 4282.

[52] N. Tsoi, A. Xiang, P. Yu, S. S. Sohn, G. Schwartz, S. Ramesh, M. Hussein, A. W. Gupta, M. Kapadia, and M. Vázquez. "Sean 2.0: Formalizing and generating social situations for robot navigation". In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 11047–11054.

[53] M. Moussaıd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz. "Experimental study of the behavioural mechanisms underlying self-organization in human crowds". In: *Proceedings of the Royal Society B: Biological Sciences* 276.1668 (2009), pp. 2755–2762.

[54] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou. "Information-theoretic model predictive control: Theory and applications to autonomous driving". In: *IEEE Transactions on Robotics* 34.6 (2018), pp. 1603–1622.

[55] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll. "What the constant velocity model can teach us about pedestrian motion prediction". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1696–1703.

[56] B. Brito, M. Everett, J. P. How, and J. Alonso-Mora. "Where to go next: Learning a subgoal recommendation policy for navigation in dynamic environments". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 4616–4623.

[57] C. Pezzato, C. Salmi, M. Spahn, E. Trevisan, J. Alonso-Mora, and C. H. Corbato. "Sampling-based model predictive control leveraging parallelizable physics simulations". In: *arXiv preprint arXiv:2307.09105* (2023).

[58] J. Betz, A. Heilmeier, A. Wischnewski, T. Stahl, and M. Lienkamp. "Autonomous driving—a crash explained in detail". In: *Applied Sciences* 9.23 (2019), p. 5126.

[59] T. Haidegger. "Autonomy for surgical robots: Concepts and paradigms". In: *IEEE Transactions on Medical Robotics and Bionics* 1.2 (2019), pp. 65–76.

[60] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. "Control barrier function based quadratic programs for safety critical systems". In: *IEEE Transactions on Automatic Control* 62.8 (2016), pp. 3861–3876.

[61] P. Wieland and F. Allgöwer. "Constructive safety using control barrier functions". In: *IFAC Proceedings Volumes* 40.12 (2007), pp. 462–467.

[62] C. Dawson, S. Gao, and C. Fan. "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control". In: *IEEE Transactions on Robotics* 39.3 (2023), pp. 1749–1767.

[63] A. Peruffo, D. Ahmed, and A. Abate. "Automated and formal synthesis of neural barrier certificates for dynamical models". In: *International conference on tools and algorithms for the construction and analysis of systems*. Springer. 2021, pp. 370–388.

[64] L. Lindemann, H. Hu, A. Robey, H. Zhang, D. Dimarogonas, S. Tu, and N. Matni. "Learning hybrid control barrier functions from data". In: *Conference on robot learning*. PMLR. 2021, pp. 1351–1370.

[65] O. So, Z. Serlin, M. Mann, J. Gonzales, K. Rutledge, N. Roy, and C. Fan. "How to train your neural control barrier function: Learning safety filters for complex input-constrained systems". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 11532–11539.

[66] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni. "Learning control barrier functions from expert demonstrations". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 3717–3724.

[67] M. Saveriano and D. Lee. "Learning barrier functions for constrained motion planning with dynamical systems". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 112–119.

[68] S. Zhang, O. So, K. Garg, and C. Fan. "Gcbf+: A neural graph control barrier function framework for distributed safe multi-agent control". In: *arXiv preprint arXiv:2401.14554* (2024).

[69] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela. "Synthesis of control barrier functions using a supervised machine learning approach". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 7139–7145.

[70] X. Wang, L. Knoedler, F. B. Mathiesen, and J. Alonso-Mora. "Simultaneous synthesis and verification of neural control barrier functions through branch-and-bound verification-in-the-loop training". In: *2024 European Control Conference (ECC)*. IEEE. 2024, pp. 571–578.

[71] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan. "Learning safe multi-agent control with decentralized neural barrier certificates". In: *arXiv preprint arXiv:2101.05436* (2021).

[72] C. Dawson, Z. Qin, S. Gao, and C. Fan. "Safe nonlinear control using robust neural lyapunov-barrier functions". In: *Conference on Robot Learning*. PMLR. 2022, pp. 1724–1735.

[73] H. Yu, C. Hirayama, C. Yu, S. Herbert, and S. Gao. "Sequential neural barriers for scalable dynamic obstacle avoidance". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 11241–11248.

[74] M. Jankovic. "Robust control barrier functions for constrained stabilization of nonlinear systems". In: *Automatica* 96 (2018), pp. 359–367.

[75]   M. H. Cohen, C. Belta, and R. Tron. "Robust control barrier functions for nonlinear control systems with uncertainty: A duality-based approach". In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE. 2022, pp. 174–179.

[76]   I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games". In: *IEEE Transactions on automatic control* 50.7 (2005), pp. 947–957.

[77]   J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert. "Robust control barrier–value functions for safety-critical control". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 6814–6821.

[78]   K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger. "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems". In: *IEEE Control Systems Magazine* 43.5 (2023), pp. 137–177.

[79]   S. Tonkens and S. Herbert. "Refining control barrier functions through Hamilton–Jacobi reachability". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 13355–13362.

[80]   S. Tonkens, A. Toofanian, Z. Qin, S. Gao, and S. Herbert. "Patching Neural Barrier Functions Using Hamilton-Jacobi Reachability". In: *arXiv preprint arXiv:2304.09850* (2023).

[81]   i. m. mitchell ian m. "the flexible, extensible and efficient toolbox of level set methods". In: *journal of scientific computing* 35 (2008), pp. 300–329.

[82]   S. Bansal and C. J. Tomlin. "Deepreach: A deep learning approach to high-dimensional reachability". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 1817–1824.

[83]   K.-C. Hsu, V. Rubies-Royo, C. J. Tomlin, and J. F. Fisac. "Safety and liveness guarantees through reach-avoid reinforcement learning". In: *arXiv preprint arXiv:2112.12288* (2021).

[84]   M. Ganai, S. Gao, and S. Herbert. "Hamilton-jacobi reachability in reinforcement learning: A survey". In: *IEEE Open Journal of Control Systems* (2024).

[85]   O. So and C. Fan. "Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning". In: *arXiv preprint arXiv:2305.14154* (2023).

[86]   A. Lin, S. Peng, and S. Bansal. "One Filter to Deploy Them All: Robust Safety for Quadrupedal Navigation in Unknown Environments". In: *arXiv preprint arXiv:2412.09989* (2024).

[87]   M. Ahmadi, X. Xiong, and A. D. Ames. "Risk-averse control via CVaR barrier functions: Application to bipedal robot locomotion". In: *IEEE Control Systems Letters* 6 (2021), pp. 878–883.

[88]   S. Liu and C. A. Belta. "Risk-Aware Adaptive Control Barrier Functions for Safe Control of Nonlinear Systems under Stochastic Uncertainty". In: *arXiv preprint arXiv:2503.19205* (2025).

[89]   X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames. "Robustness of control barrier functions for safety critical control". In: *IFAC-PapersOnLine* 48.27 (2015), pp. 54–61.

[90]   A. Singletary, A. Swann, Y. Chen, and A. D. Ames. "Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 2897–2904.

[91]   Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames. "Backup control barrier functions: Formulation and comparative study". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 6835–6841.

[92]   D. R. Agrawal, R. Chen, and D. Panagou. "gatekeeper: Online safety verification and control for nonlinear systems in dynamic environments". In: *IEEE Transactions on Robotics* (2024).

[93]   D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814.

[94]   D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim. "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control". In: *arXiv preprint arXiv:1909.06586* (2019).

[95]   E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia. "Autonomous racing using linear parameter varying-model predictive control (LPV-MPC)". In: *Control Engineering Practice* 95 (2020), p. 104270.

[96]   Z. Wang, O. So, J. Gibson, B. Vlahov, M. S. Gandhi, G.-H. Liu, and E. A. Theodorou. "Variational inference MPC using Tsallis divergence". In: *arXiv preprint arXiv:2104.00241* (2021).

[97]   O. So, Z. Wang, and E. A. Theodorou. "Maximum entropy differential dynamic programming". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 3422–3428.

[98]   A. Boccia, L. Grüne, and K. Worthmann. "Stability and feasibility of state constrained MPC without stabilizing terminal constraints". In: *Systems & control letters* 72 (2014), pp. 14–21.

[99]   C. Kirches, L. Wirsching, H. G. Bock, and J. P. Schlöder. "Efficient direct multiple shooting for nonlinear model predictive control on long horizons". In: *Journal of Process Control* 22.3 (2012), pp. 540–550.

[100]   K. P. Wabersich and M. N. Zeilinger. "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems". In: *Automatica* 129 (2021), p. 109597.

[101]   C. De Boor. "A practical guide to splines". In: *Springer-Verlag google schola* 2 (1978), pp. 4135–4195.

[102] A. Altarovici, O. Bokanowski, and H. Zidani. "A general Hamilton-Jacobi framework for non-linear state-constrained control problems". In: *ESAIM: Control, Optimisation and Calculus of Variations* 19.2 (2013), pp. 337–357.

[103] Q. Nguyen and K. Sreenath. "Robust safety-critical control for dynamic robotics". In: *IEEE Transactions on Automatic Control* 67.3 (2021), pp. 1073–1088.

[104] Q. Nguyen and K. Sreenath. "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints". In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 322–328.

[105] W. Xiao and C. Belta. "Control barrier functions for systems with high relative degree". In: *2019 IEEE 58th conference on decision and control (CDC)*. IEEE. 2019, pp. 474–479.

[106] P. Heidlauf, A. Collins, M. Bolender, and S. Bak. "Verification Challenges in F-16 Ground Collision Avoidance and Other Automated Maneuvers." In: *ARCH@ ADHS* 2018 (2018).

[107] B. L. Stevens, F. L. Lewis, and E. N. Johnson. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

[108] J. Yin, C. Dawson, C. Fan, and P. Tsiotras. "Shield Model Predictive Path Integral: A Computationally Efficient Robust MPC Method Using Control Barrier Functions". In: *IEEE Robotics and Automation Letters* 8.11 (2023), pp. 7106–7113. DOI: 10.1109/LRA.2023.3315211.

[109] J. Yin, Z. Zhang, E. Theodorou, and P. Tsiotras. "Trajectory Distribution Control for Model Predictive Path Integral Control using Covariance Steering". In: *2022 International Conference on Robotics and Automation (ICRA)*. 2022, pp. 1478–1484. DOI: 10.1109/ICRA46639.2022.9811615.

[110] J. Yin, P. Tsiotras, and K. Berntorp. *Chance-Constrained Information-Theoretic Stochastic Model Predictive Control with Safety Shielding*. 2024. arXiv: 2408.00494 [cs.RO]. URL: https://arxiv.org/abs/2408.00494.

[111] W. He, Z. Li, and C. P. Chen. "A survey of human-centered intelligent robots: issues and challenges". In: *IEEE/CAA Journal of Automatica Sinica* 4.4 (2017), pp. 602–609.

[112] M. Pfeiffer, G. Paolo, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena. "A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 5921–5928.

[113] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. "Social lstm: Human trajectory prediction in crowded spaces". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 961–971.

[114]   P. Trautman and A. Krause. "Unfreezing the robot: Navigation in dense, interacting crowds". In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE. 2010, pp. 797–803.

[115]   A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras. "Human motion trajectory prediction: A survey". In: *The International Journal of Robotics Research* 39.8 (2020), pp. 895–935.

[116]   R. Löhner. "On the modeling of pedestrian motion". In: *Applied Mathematical Modelling* 34.2 (2010), pp. 366–382.

[117]   B. Ivanovic and M. Pavone. "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2019, pp. 2375–2384.

[118]   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.* 2019, pp. 0–0.

[119]   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).* IEEE. 2016, pp. 2096–2101.

[120]   J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.* "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.

[121]   D. Helbing and P. Molnar. "Social force model for pedestrian dynamics". In: *Physical review E* 51.5 (1995), p. 4282.

[122]   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". In: *The International Journal of Robotics Research* 34.2 (2015), pp. 201–217.

[123]   A. Bera, S. Kim, T. Randhavane, S. Pratapa, and D. Manocha. "GLMP-realtime pedestrian path prediction using global and local movement patterns". In: *2016 IEEE International Conference on Robotics and Automation (ICRA).* IEEE. 2016.

[124]   S. Becker, R. Hug, W. Hubner, and M. Arens. "Red: A simple but effective baseline predictor for the trajnet benchmark". In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops.* 2018, pp. 0–0.

[125]   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.* 2018, pp. 2255–2264.

[126] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker. "Desire: Distant future prediction in dynamic scenes with interacting agents". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 336–345.

[127] B. Brito, H. Zhu, W. Pan, and J. Alonso-mora. "Social-vrnn: One-shot multi-modal trajectory prediction for interacting pedestrians". In: *2020 Conference on Robot Learning (CoRL)*. 2020.

[128] 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*. IEEE. 2009, pp. 261–268.

[129] A. Lerner, Y. Chrysanthou, and D. Lischinski. "Crowds by example". In: *Computer graphics forum*. Vol. 26. Wiley Online Library. 2007, pp. 655–664.

[130] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. "nuscenes: A multimodal dataset for autonomous driving". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11621–11631.

[131] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.* "Argoverse: 3d tracking and forecasting with rich maps". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8748–8757.

[132] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. "A continual learning survey: Defying forgetting in classification tasks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[133] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. "Continual lifelong learning with neural networks: A review". In: *Neural Networks* 113 (2019), pp. 54–71.

[134] J. Yoon, E. Yang, J. Lee, and S. J. Hwang. "Lifelong Learning with Dynamically Expandable Networks". In: *International Conference on Learning Representations*. 2018.

[135] C.-Y. Hung, C.-H. Tu, C.-E. Wu, C.-H. Chen, Y.-M. Chan, and C.-S. Chen. "Compacting, picking and growing for unforgetting continual learning". In: *Advances in Neural Information Processing Systems* 32 (2019).

[136] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong. "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3925–3934.

[137] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. "icarl: Incremental classifier and representation learning". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 2001–2010.

[138]   H. Shin, J. K. Lee, J. Kim, and J. Kim. "Continual learning with deep generative replay". In: *Advances in neural information processing systems*. 2017, pp. 2990–2999.

[139]   R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. "Gradient based sample selection for online continual learning". In: *Advances in neural information processing systems* 32 (2019).

[140]   R. Aljundi, K. Kelchtermans, and T. Tuytelaars. "Task-free continual learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11254–11263.

[141]   F. Zenke, B. Poole, and S. Ganguli. "Continual learning through synaptic intelligence". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3987–3995.

[142]   A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese. "Sophie: An attentive gan for predicting paths compliant to social and physical constraints". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1349–1358.

[143]   C. Schöller, V. Aravantinos, F. Lay, and A. Knoll. "What the constant velocity model can teach us about pedestrian motion prediction". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1696–1703.

[144]   C. I. Mavrogiannis, W. B. Thomason, and R. A. Knepper. "Social momentum: A framework for legible navigation in dynamic multi-agent environments". In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 2018, pp. 361–369.

Acknowledgments

I approached these acknowledgments by staring at a blank page for a very long time, reminiscing about the past four years filled with challenges, growth, incredible people, countless tea breaks, and memories I will keep forever. Having tackled the hardest part, the first sentence, I am now ready to thank everyone. After all, acknowledgments are about recognizing that this journey was never truly mine alone. I will also take this opportunity to capture my favorite memories to make sure I don't forget them. Please bear with me if these acknowledgments run a little long!

First and foremost, I would like to thank my supervisor and promotor, Javier, for giving me the chance to do a PhD surrounded by such kind and supportive people. I have felt truly lucky to be part of the Autonomous Multi-Robots (AMR) lab. Thanks for your support and the many opportunities you have provided me with. I am also sincerely grateful to my (co-)promotor, Robert, for his guidance throughout this journey. I truly appreciated your thoughtful advice and the insightful, in-depth discussions on my propositions. My sincere thanks also go to my PhD committee, Martijn Wisse, Vanessa Evers, Chuchu Fan, Ömür Arslan, and Hans Hellendoorn, for their time, thoughtful feedback, and (hopefully) engaging discussion.

I am also incredibly grateful to the H2020 Harmony project, which funded my PhD and allowed me to be part of something bigger. Through it, I had the chance to visit Zürich, Twente, Västerås, Lisbon, and Edinburgh, each trip brought new collaborations, perspectives, and memories. Thank you to everyone involved for the shared work and effort, especially in making the final demo a success. Special thanks to Stefano, Aftab, and Fabio for bridging the gap between academic research and the ROS2 demo. Thanks Gianluca, for your input on impedance control. Thanks also to Nicky, Matteo, Haofei, Mohammad, Paula, Francesco, Giulio, Jen Jen, and Lionel for making the integration weeks in Västerås such a great experience. We may forever wonder when the pizzaiolo will actually show up. I also appreciated the insights from Hideki, Jan, and Bob on human-robot interaction. Some personal highlights: hiking the Highlands with Paula and Francesco, and listening to fado in Lisbon.

Next, I would like to thank Chuchu for warmly welcoming me into the Reliable Autonomous Systems Lab at MIT (REALM) and the entire REALM group for making my three-month stay so enjoyable. Special thanks to Oswin for being a fantastic coauthor. I learned so much from you. Thank you to Weichao for our many shared lunches, and to Kunal for taking me to the free pizza Thursdays. It was a pleasure getting to know Songyuan, Eric, Jake, Yue, Ruixiao, Yongchao, and

Ben. I especially enjoyed the bubble tea outing with the REALM girls, Anjali, Yilun, Mingxin, and Yingke, as well as the Blue Hills hike and post-hike dinner with the group. A big thank you to Laurence for making my start in Boston so easy and for filling my stay with endless activities: swing dancing, karaoke, museum trips, raclette, a 100 km Cape Cod bike ride, visiting Providence and more. You made my time there truly unforgettable. I'm also grateful to Maxi for bringing a bit of TU Delft to Boston and for sharing the experience of watching the Boston Red Sox game in the rain. Thanks also to David and Marco for being great lunch buddies, to Adrien for taking Magda and me sailing, and to my roommates Molly and Smita. Finally, I thank Het Cultuurfonds Wetenschapsbeurzen for supporting my stay financially.

Starting my PhD in a new country during a global pandemic was challenging, which made the friendships I found at Leeghwaterstraat all the more valuable. I cherished the evening walks, Chinese meals, and weekend bike rides with the "bike trips" group, thank you Xiaohuan, Jing, Desong, Yifei, Tianlong, Sahar, Fatemeh, and Laura. Special thanks to the "bike trips survivors", Til, Leon, Yujie, Ensieh, and Saeed, for the great adventures, especially the memorable Texel trip with Til, Leon, and Yujie. Although Yujie will be mentioned multiple times throughout these acknowledgments, I want to take a moment to express how much sharing this PhD journey with her has meant to me. Yujie, you were the first person I met in Delft, and I am deeply grateful for your friendship and support. I'll always cherish our time together, from long hours in the office and a rainy bike ride with Fatemeh, to mildly spicy hotpot dinners and your joy over pork skewers in Athens. I'm so happy we finally made our trip to Greece together, and I look forward to visiting you and Hai in China and to many more years of friendship ahead.

When we were finally allowed to work from the office, I felt very lucky to have such wonderful office mates. Thank you, Elia, for inviting us over for homemade pizza; Yujie, for bringing interesting Chinese snacks; and Maxi for introducing me to bubble tea (I'm trying to limit the number of bubble tea mentions in these acknowledgments, but let's just say we've shared quite a few in New York, Boston, and Delft). Anna, thanks for the great times collecting Sagar's puns, cooking meatballs, baking Bärentatzen, and more. Khaled, thank you for taking me to the gym and patiently explaining where I should feel the exercises, though somehow, I still felt everything in my biceps. And Nils, thanks for joining our office outings, even when the only vegetarian options were edamame and fries. Later on, Saray (Replacement Nils) joined our office, and I was glad she became part of roommates in F-2-140. We shared an amazing trip to MRS in Boston, as well as adventures in New York and Toronto. Thank you for always inviting me over, for being my "only Dutch friend", and for taking Yujie and me to visit your family in Limburg. I'm already looking forward to our trip to China, and maybe a visit to Stockholm next? I also want to thank our honorary office mate, Julian, for the amazing cakes and motivating me to go bouldering every week. Beyond office life, I really appreciated all F-2-140 office events we shared: go-karting, countless dinners, movies, skiing, and I want to thank everyone who joined. Thanks as well to everyone who spent time with us in the

office, even briefly, Domenico, Riccardo, Shaohang, Tomáš, Dennis, and others.

Next, I want to thank all former and current members of the AMR group. Bruno, thank you for guiding me through my first year and first paper deadline, and for the dinners and board game nights during those early weeks, they meant a lot. It was great to reconnect in Boston, twice!, share tacos, and celebrate your wedding in Porto. Thanks, Hai, for being the notation expert, and Xiaoshan, even though we never met, for collaborating on the task assignment paper. I also want to thank the postdocs Andrés, Xinwei, and Daniel. Maximilian K, thank you for reminiscing about the Länd; Alvaro, for the long discussions on bike rides home after work; Max S, for valuing reproducible code and being yourself; Oscar, for helping me with the Jackal demo, and Max L, for restructuring gym environments together. Dennis, I'll never forget our peanut butter sandwiches on top of the Empire State Building during those four hours we sat there to avoid paying the sunset fee! It was great having you join the office at the end, too. Thanks to Lasse, for bringing structure to the group, sharing your research advice, leading by example, and always being available to help, and to Clarence for being a great new office mate. Andreu, thank you for joining our office and for sharing your wide-ranging knowledge from Japanese knives and finance to AirPods and art. Sihao, thank you for always being there for a chat, and Ahmad, thank you for your interest in Control Barrier Functions. Thank you to Diego, I'm grateful to be your coauthor and glad our paths crossed again in Stockholm. I also want to thank my master students Denesh, Sunny (Xinyu), and Jelmer for the enthusiasm and effort they put into their theses. I really enjoyed working with you. And thanks to Chadi for collaborating with me to turn your thesis into a paper.

Outside F-2-140 and the AMR group, I'd like to thank everyone at CoR for the shared lunches, tea and cake breaks, and faculty drinks that made everyday life more enjoyable. Thanks to Tasos for letting Anna and me use your oven; Italo and Lorenzo for being great roommates when visiting Melanie (read in Lorenzo's voice); Giovanni for always sharing his opinions; and Rodrigo for continuing the dinner and board game tradition, and for teaching me about imitation learning. Thank you, Linda, for being my roommate in Philadelphia and now again in Stockholm. Thanks Chris, I really appreciated your (career) advice. Thanks also to Corrado, Chadi, and Cong for the fun RoboHouse Fridays, and to Carlos, Jihong, Sagar, Ashwin, Jelle, Alex, Tomás, Mariano, Forough, Fiorella, Gustavo, Manuel, Ebrahim, and everyone else who made CoR such a friendly and supportive place. Finally, thanks to André, Maurits, Thomas, and Kseniia for your help, whether building the Jackal sensor setup or working on the Dingo.

Outside of work, I enjoyed being part of the Blue Falcons Floorball Association. Playing for the Ladies team, The Falconites, Los Falconites, and playing in Arosa. Thanks to everyone from the Blue Falcons for the great time, and especially Ana for our many trips to Utrecht. I also want to thank my friends from home and everyone who visited me in Delft, it meant a lot to be able to share my beautiful new home with you. Thanks Lisa, Sarah, Andrea, Anne, Kathrin, Kerstin, Salehah, Theresa,

Vera, Annette, Marion and Andy.

Finally, I want to thank my family for their unwavering support throughout my life. Thanks to my extended family for showing interest in what I am doing, although it is difficult to grasp. Thanks to my sister Frida for exploring Delft and the Netherlands together. I'm really proud of you and always enjoy our time together. Thank you to my parents for always being there for me, especially for your patience and support, even in those moments during childhood (and still now?) when I got frustrated because things didn't work out immediately. Danke, Mama und Papa!

Luzia
Stockholm, July 2025

List of Publications

Journal Articles

🏆 X. Bai, A. Fielbaum, M. Kronmüller, **L. Knoedler**, and J. Alonso-Mora. "Group-based distributed auction algorithms for multi-robot task assignment." IEEE Transactions on Automation Science and Engineering (T-ASE) 20.2 (2022): 1292-1303. T-ASE Best Paper Award 2024.

📄 **L. Knoedler***, C. Salmi*, H. Zhu, B. Brito, and J. Alonso-Mora. "Improving Pedestrian Prediction Models with Self-Supervised Continual Learning." IEEE Robotics and Automation Letters 7.2 (2022): 4781-4788.

D. Martinez-Baselga, O. de Groot, **L. Knoedler**, L. Riazuelo, J. Alonso-Mora, and L. Montano. "SHINE: Social homology identification for navigation in crowded environments." The International Journal of Robotics Research (IJRR) (2025).

📄 **L. Knoedler***, O. So*, J. Yin, M. Black, Z. Serlin, P. Tsiotras, J. Alonso-Mora, and C. Fan. "Safety on the Fly: Constructing Robust Safety Filters via Policy Control Barrier Functions at Runtime." IEEE Robotics and Automation Letters (2025).

Conference Papers

D. Martinez-Baselga, O. de Groot, **L. Knoedler**, J. Alonso-Mora, L. Riazuelo, and L. Montano. "Hey Robot! Personalizing Robot Navigation through Model Predictive Control with a Large Language Model." 2025 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2025.

J. de Wolde, **L. Knoedler**, G. Garofalo, and J. Alonso-Mora. "Current-Based Impedance Control for Interacting with Mobile Manipulators." 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2024.

X. Wang, **L. Knoedler**, F.B. Mathiesen, and J. Alonso-Mora. "Simultaneous Synthesis and Verification of Neural Control Barrier Functions Through Branch-and-Bound Verification-in-the-Loop Training." 2024 European Control Conference (ECC). IEEE, 2024.

S. Bakker*, **L. Knoedler***, M. Spahn, W. Böhmer, and J. Alonso-Mora. "Multi-Robot Local Motion Planning Using Dynamic Optimization Fabrics." 2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). IEEE, 2023.

Workshop Papers

📄 **L. Knoedler**, N. Wilde, and J. Alonso-Mora. "Interaction-Aware Autonomous Navigation Among Pedestrians Using Social Forces Response Dynamics." In Workshop on Unsolved Problems in Social Robot Navigation at Robotics: Science and Systems (RSS), 2024.

D. Martinez-Baselga, O. de Groot, **L. Knoedler**, L. Riazuelo, J. Alonso-Mora, and L. Montano. "Learning Social Homologies for Navigation." In Workshop on Unsolved Problems in Social Robot Navigation at Robotics: Science and Systems (RSS), 2024.

L. Knoedler, B. Brito, M. Everett, J.P. How, and J. Alonso-Mora. "Learning a Guidance Policy from Humans for Social Navigation." In Social Robot Navigation: Advances and Evaluation Workshop at IEEE International Conference on Robotics and Automation (ICRA), 2022.

◼ Included in this thesis.
🏆 Won a best paper award.
* Authors contributed equally.

Curriculum Vitæ

Luzia Knoedler

Personal Data

Place and Date of Birth:	Stuttgart, Germany	10 June 1996
Email:	luzia.knoedler@gmail.com

Education

2021 - 2025	**Ph.D Robotics**, Delft University of Technology, The Netherlands. Topic: Motion planning for mobile robots in human-centered environments. Supervisor: Javier Alonso-Mora.
Aug. - Oct. 2024	**Visiting PhD**, Massachusetts Institute of Technology, USA. Topic: Robust Policy Control Barrier Functions. Supervisor: Chuchu Fan.
2017 - 2020	**M.Sc. Engineering Cybernetics**, University of Stuttgart, Germany. Specialization: Control Engineering & Autonomous Systems and Control. Thesis: Human Driver Modeling during Acceleration Processes.
2018 - 2019	**Erasmus Semester**, Chalmers University of Technology, Sweden. Topics: Discrete Event Systems, Advanced Topics in Control, Automotive Systems.
2014 - 2017	**B.Sc. Medical Engineering**, Univ. of Stuttgart & Tübingen, Germany. Specialization: Nanotechnology & Control Engineering. Thesis: Finite Element Simulation of Breast Tissue.