Michael Hartmann, Master of Science (M. Sc.)

Stochastic pedestrian models for autonomous vehicles

Dissertation

to achieve the university degree of

Doktor der technischen Wissenschaften (Ph.D.)

Doktoratsstudium der technischen Wissenschaften: Elektrotechnik

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Daniel Watzenig

Institute of Automation and Control
Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Martin Horn

Graz, June 2022

This document is set in Palatino, compiled with pdfLaTeX2e and Biber.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material that has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present Ph.D. Thesis.

Date

Signature

Abstract

This dissertation addresses the modeling of pedestrians in dynamic and urban environments interacting with autonomous vehicles. The collision avoidance system of an autonomous vehicle has contrary safety and efficiency requirements. On the one hand, there might be collisions in following a risky driving policy. On the other hand, a safe driving policy might bring the passenger very slow to the target to prevent all kinds of risks. The autonomous vehicle does not know or perceive all relevant information, such as the unknown intention and the environmental and situational factors influencing the pedestrian's behavior. There is a resulting decision dilemma for autonomous vehicles between road safety for all road users and efficient motion planning in environments with vulnerable road users. There also exists a lack of knowledge by predicting the future movements of pedestrians, where one could compute worst-case reachable state-sets. The areas of possible reach sets could get very large. An autonomous vehicle is not allowed to drive into these areas, making motion planning inefficient. The adaption to real-world scenarios is not trivial. The decision-making process in motion planning is challenging due to the enormous variety of situations and the uncertainty of predicting future human movements with absolute certainty. There is a potential risk of accidents in adapting and predicting human locomotion. These problems influence the trust and acceptance of autonomous vehicles with additional technological and legal challenges. This dissertation aims not to ensure total safety because of pedestrians' technical and diverse physical, cognitive, situational, and environmental complexity. This work uses a new method that combines machine learning with reachability analysis (resulting in an adaptive funnel, hull, or belief set computation). Machine learning adapts the reachability analysis to current situations. Therefore adaptive reachability analysis and corresponding motion planning are presented and evaluated in vehicle simulations. Adaptive hull computational methods for adaptive implementation of reachability analysis lead to risky pedestrian bypassing. These computational methods provide a trade-off between safety and efficiency in motion planning. However, the proposed approach cannot guarantee an exact threshold for overall safety due to the complexity of the problem (unknown intent and environmental factors). A very intuitive approach predicts the future situation's maximal velocity, acceleration, and jerk of a pedestrian. Afterward, it computes the adaptive reach sets with conventional methods. Adapting classical worst-case reachability analysis could drastically reduce the cumulative volume of adaptive reachable sets compared to classical reachable sets (less than 70 percent of the cumulative area compared to classical computation in a presented use case). The results show a massive potential to reduce the areas from reach sets to belief sets. The causal inference could model the intention change. Intent changes are modeled with causal inference without considering structural learning and tested with model predictive control and adaptive hull computation in a simulation environment. This thesis also presents a similarity of human locomotion with (Partially Observable) Markov Decision Processes (MDPs and POMDPs). Nevertheless, many causal relationships describing pedestrian behavior are still unexplored. Concepts for building novel test environments provide an outlook on fundamental research. New test environments might bring new insights into the causal information structure between cognition and human locomotion in different settings to improve accident statistics further. These new test environments could help make the adaptive reachable sets more robust and provide a quantitive guarantee for trusting novel hull computations.

Kurzfassung

Diese Dissertation befasst sich mit der Modellierung von Fußgängern in dynamischen und urbanen Umgebungen, die mit autonomen Fahrzeugen interagieren. Das Kollisionsvermeidungssystem eines autonomen Fahrzeugs hat konträre Anforderungen an Sicherheit und Effizienz. Einerseits kann es zu Kollisionen kommen, wenn eine riskante Fahrweise verfolgt wird. Andererseits könnte eine sichere Fahrweise den Mitfahrer sehr langsam zum Ziel bringen, um alle Arten von Risiken zu vermeiden. Das autonome Fahrzeug (er-)kennt nicht alle relevanten Informationen, wie die unbekannte Absicht des Fußgängers und die Umwelt- und Situationsfaktoren, die sein Verhalten beeinflussen. Daraus ergibt sich für autonome Fahrzeuge ein Entscheidungsdilemma zwischen Verkehrssicherheit für alle Verkehrsteilnehmer und effizienter Bewegungsplanung in Umgebungen mit gefährdeten Verkehrsteilnehmern. Es besteht auch ein Wissensdefizit bei der Vorhersage der zukünftigen Bewegungen von Fußgängern. Man könnte hierbei worst-case Zustandsmengen berechnen. Die Flächen der möglichen Erreichbarkeitsmengen könnten sehr groß werden. Ein autonomes Fahrzeug dürfte nicht in diese Bereiche hineinfahren, was die Bewegungsplanung ineffizient macht. Die Anpassung an reale Szenarien ist daher nicht trivial. Der Entscheidungsprozess bei der Bewegungsplanung ist aufgrund der enormen Vielfalt an Situationen und der Ungewissheit, zukünftige menschliche Bewegungen mit absoluter Sicherheit vorherzusagen, eine Herausforderung. Es besteht ein potenzielles Unfallrisiko bei der Anpassung und Vorhersage menschlicher Fortbewegung. Diese Probleme beeinflussen das Vertrauen und die Akzeptanz von autonomen Fahrzeugen und stellen zusätzliche technische und rechtliche Herausforderungen dar. Aufgrund der technischen und vielfältigen physischen, kognitiven, situativen und umweltbedingten Komplexität von Fußgängern kann diese Dissertation keine absolute Sicherheit gewährleisten. In dieser Arbeit wird ein neues Verfahren eingesetzt, bei dem maschinelles Lernen mit der Erreichbarkeitsanalyse verknüpft wird (was zu einer adaptiven Trichter-, Hüllen- oder Belief-Set-Berechnung führt). Maschinelles Lernen passt die Erreichbarkeitsanalyse an aktuelle Situationen an. Daher werden die adaptive Erreichbarkeitsanalyse und die entsprechende Bewegungsplanung vorgestellt und in Fahrzeugsimulationen evaluiert. Adaptive Hull-Berechnungsmethoden zur adaptiven Umsetzung der Erreichbarkeitsanalyse führen zu einer riskanten Umgehung von Fußgängern. Diese Berechnungsmethoden bieten einen Kompromiss zwischen Sicherheit und Effizienz bei der Bewegungsplanung. Allerdings kann der vorgeschlagene Ansatz aufgrund der Komplexität des Problems (unbekannte Absicht und Umweltfaktoren) keinen exakten Schwellenwert für die Gesamtsicherheit garantieren. Ein sehr intuitiver Ansatz sagt die maximale Geschwindigkeit, Beschleunigung und den Ruck eines Fußgängers in einer zukünftigen Situation voraus. Anschließend werden die adaptiven Reichweitensätze mit herkömmlichen Methoden berechnet. Durch die Anpassung der klassischen Worst-Case-Erreichbarkeitsanalyse konnte das kumulative Volumen der adaptiven Erreichbarkeitsmengen im Vergleich zu klassischen Erreichbarkeitsmengen drastisch reduziert werden (weniger als 70 Prozent der kumulativen Fläche im Vergleich zur klassischen Berechnung in einem vorgestellten Anwendungsfall). Die Ergebnisse zeigen ein massives Potential zur Reduzierung der Flächen von reach sets auf belief sets. Die kausale Inferenz könnte die Absichtsänderung modellieren. Absichtsänderungen werden mit kausaler Inferenz ohne Berücksichtigung von strukturellem Lernen modelliert und mit Model Predictive Control und adaptiver Rumpfberechnung in einer Simulationsumgebung getestet. In dieser Arbeit wird auch eine Ähnlichkeit der menschlichen Fortbewegung mit (teilweise beobachtbaren) Markov-Entscheidungsprozessen (MDPs und POMDPs) vorgestellt. Dennoch sind viele kausale Zusammenhänge, die das Verhalten von Fußgängern beschreiben, noch unerforscht. Konzepte zum Aufbau neuartiger Testumgebungen geben einen Ausblick auf die Grundlagenforschung. Neue Testumgebungen könnten neue Einblicke in die kausale Informationsstruktur zwischen Kognition und menschlicher Fortbewegung in verschiedenen Umgebungen bringen, um die Unfallstatistik weiter zu verbessern. Diese neuen Testumgebungen könnten dazu beitragen, die adaptiven erreichbaren Mengen robuster zu machen und eine quantitative Garantie für das Vertrauen in neuartige Rumpfberechnungen zu bieten.

Acknowledgement

The Ph.D. thesis was primarily written in Virtual Vehicle Research GmbH. Due to research stays (non-) European countries in different have influenced the content to a certain extent.

Virtual Vehicle Research GmbH has received funding within COMET Competence Centers for Excellent Technologies from the Austrian Federal Ministry for Climate Action, the Austrian Federal Ministry for Digital and Economic Affairs, the Province of Styria (Dept. 12), and the Styrian Business Promotion Agency (SFG). The Austrian Research Promotion Agency (FFG) has been authorized for the program management.

The European Commission is gratefully acknowledged for its support of the Marie Skłodowska-Curie research and innovation program ITEAM (Interdisciplinary Training Network in Multi-Actuated Ground Vehicles - grant agreement No. 675999) and SHOW-program (grant agreement No 875530).

Thank my professor Daniel Watzenig for his encouragement, technical feedback, and recommendation for a scholarship. I want to thank him, the Virtual Vehicle Research GmbH, on behalf of CEO Dr. Jost Bernasch, the Technical University Graz, and the European Commission for their excellent programs and many opportunities. For example, during my research stay at the University of Berkeley, California, other research projects in European universities. Thank my colleague Johannes Rumetshofer for his technical input and thesis review. All my colleagues and Joachim and Markus for the cooperation in the SHOW project. Thanks to Prof. Antonella Ferrara for her service as a second examiner and for her thesis reviews. I say thanks to Prof. Ivanov for the inspiration during the ITEAM project. Thanks to my current and former colleagues for good collaboration in the company and

the ITEAM research network. I want to thank the German Scholar Organization for the scholarship in the Leadership Academy and the professional development opportunities. I want to thank my fiancée Katarina for her help and friendship. I am looking forward to our future together. Thanks to my parents for their encouragement and the possibilities for my education. My fiancée and parents have always supported, motivated, taught, and stood by my side. I want to thank you for that. Thanks to my siblings, who inspired me, and the great time together. Thanks to many friends, mentors, and companions who inspired me during my personal growth. Especially my friend Ermanno, who helped me in some challenging situations during my Ph.D. thesis. Many more persons helped, inspired, and supported me during my education. I want to thank you!

Danksagung

Die Doktorarbeit wurde hauptsächlich in der Virtual Vehicle Research GmbH geschrieben. Aufgrund von Forschungsaufenthalten in verschiedenen (nicht-)europäischen Ländern wurde der Inhalt in gewissem Umfang beeinflusst.

Die Virtual Vehicle Research GmbH wurde im Rahmen von COMET Competence Centers for Excellent Technologies vom Bundesministerium für Klimapolitik, dem Bundesministerium für Digitales und Wirtschaft, dem Land Steiermark (Abt. 12) und der Steirischen Wirtschaftsförderungsgesellschaft (SFG) gefördert. Die Österreichische Forschungsförderungsgesellschaft (FFG) wurde mit dem Programmmanagement beauftragt.

Der Europäischen Kommission wird für die Unterstützung des Marie Skłodowska-Curie Forschungs- und Innovationsprogramms ITEAM (Interdisciplinary Training Network in Multi-Actuated Ground Vehicles - Grant Agreement No. 675999) und des SHOW-Programms (Grant Agreement No 875530) gedankt.

Ich danke meinem Professor Daniel Watzenig für seine Ermutigung, sein technisches Feedback und seine Empfehlung für ein Stipendium. Ich möchte ihm, der Virtual Vehicle Research GmbH, stellvertretend für den Geschäftsführer Dr. Jost Bernasch, der Technischen Universität Graz und der Europäischen Kommission für ihre hervorragenden Programme und die vielen Möglichkeiten danken. Zum Beispiel während meines Forschungsaufenthalts an der University of Berkeley, Kalifornien, andere Forschungsprojekte an europäischen Universitäten. Meinem Kollegen Johannes Rumetshofer danke ich für seinen technischen Input und die Überprüfung der Arbeit. Allen meinen Kollegen und Joachim und Markus für die Zusammenarbeit im SHOW-Projekt. Danke an Prof. Antonella Ferrara für ihren Dienst als Zweitprüferin und für die Begutachtung der Arbeit. Ich bedanke mich bei

Prof. Ivanov für die Inspiration während des ITEAM-Projekts. Danke an meine jetzigen und ehemaligen Kollegen für die gute Zusammenarbeit im Unternehmen und im ITEAM-Forschungsnetzwerk. Ich möchte mich bei der German Scholar Organization für das Stipendium in der Leadership Academy und die beruflichen Entwicklungsmöglichkeiten bedanken. Ich möchte meiner Verlobten Katarina für ihre Hilfe und Freundschaft danken. Ich freue mich auf unsere gemeinsame Zukunft. Ich danke meinen Eltern für ihre Ermutigung und die Möglichkeiten, die sie mir für meine Ausbildung bieten. Meine Verlobte und meine Eltern haben mich immer unterstützt, motiviert, unterrichtet und mir zur Seite gestanden. Dafür möchte ich mich bei ihnen bedanken. Danke an meine Geschwister, die mich inspiriert haben, und die tolle gemeinsame Zeit. Danke an die vielen Freunde, Mentoren und Wegbegleiter, die mich in meiner persönlichen Entwicklung inspiriert haben. Insbesondere meinem Freund Ermanno, der mir in einigen schwierigen Situationen während meiner Doktorarbeit geholfen hat. Viele weitere Personen haben mir während meiner Ausbildung geholfen, mich inspiriert und unterstützt. Ich möchte mich bei allen herzlich bedanken!

Contents

Abstract v

Zusammenfassung vii

Acknowledgement ix

Danksagung xi

1. Introduction 1
   1.1. Motivation 1
   1.2. Focused problem 4
      1.2.1. Challenges 4
      1.2.2. Gap of existing methods 5
   1.3. Contribution 6
      1.3.1. Topic of the thesis 6
      1.3.2. Achievements 6
      1.3.3. Scientific publications 7
      1.3.4. Structure of the thesis 8
   1.4. Non-Goals 14

2. State of the art 17
   2.1. Reachability Analysis 17
   2.2. Machine Learning 26
      2.2.1. Movement prediction 26
      2.2.2. Reinforcement Learning 37
      2.2.3. Causal Inference 41
   2.3. Cooperative interacting automobiles 45
      2.3.1. Motion Planning 47
      2.3.2. Consequences 55

3. Developed concepts **59**
3.1. Overview of chapter 3 59
3.2. Motion planning 64
3.3. Adaptive set deformation 69
3.3.1. Motivation for adaptive set deformation 69
3.3.2. Group actions and set operations 70
3.3.3. Special case 73
3.3.4. Examples for set-based movement prediction 74
3.4. Funnel prediction 75
3.4.1. Motivation for intelligent funnel prediction 76
3.4.2. Exemplary view on deforming sets and incorporation of prediction models 79
3.4.3. Data-based approaches 84
3.4.4. Adaptive reachability analysis 86
3.4.5. Example with a data-based approach 90
3.4.6. Example of a funnel prediction with real pedestrian movements 90
3.5. Pedestrians walking on Manifolds 92
3.5.1. Motivation for topological spaces 92
3.5.2. Mathematical concepts for human locomotion 96
3.5.3. Manifolds for human locomotion 101
3.5.4. Optimal control on manifolds 101
3.5.5. Use case for the University Campus 103
3.5.6. Use-case: Stanford dataset 104
3.5.7. Motivation for synthetic data 110
3.5.8. Quantification of computational complexity 114
3.5.9. Decision making of pedestrians 116
3.5.10. Computation of set-based movement prediction 120
3.5.11. Algorithms for set-based movement prediction 126
3.6. Causal inference 127
3.6.1. Motivation for Causal Inference 127
3.6.2. Example for non-cyclic interventions 128
3.6.3. Interventions on cyclic models 132
3.6.4. Pedestrian movements with causal do-operator 133
3.7. Cognitive Decision Models 134

4. Evaluation **145**
4.1. Mathematical details 145
4.2. Simulations 153
4.3. Causal inference and goals for the agents 154
4.4. Connected reachable sets topological spaces 160
4.5. Evaluation over different pedestrian models 163

5. Conclusion **169**
5.1. Discussion and contribution 169
5.2. Limitations 170
5.3. Outlook 171

A. Mathematical Background **183**
A.1. Differential Geometry 183
A.2. Computational geometry 187

B. Cognitive systems **193**
B.1. Human behavior 193
B.2. Human locomotion 195
B.2.1. Human perception 196
B.2.2. Reasoning 197
B.2.3. Human movements 198
B.2.4. Consequences 198
B.3. Technology 200
B.4. Urban environment 202

C. Additional information **205**
C.1. Simulator 205
C.2. Algorithms 212

Bibliography **221**

List of Figures

1.1. Introduction example 3
1.2. Scenario of pedestrian movement prediction 4
1.3. Overview of the Ph.D. thesis 9
1.4. Structure of chapter 1 - Introduction 9
1.5. Structure of chapter 2 - State of the art 10
1.6. Structure of chapter 3 - Developed concepts 11
1.7. Structure of chapter 4 - Evaluation 12
1.8. Structure of chapter 5 13
1.9. Structure of the appendix 14

2.1. Deterministic system and machine learning model 17
2.2. Funnel and Reachability Analysis 25
2.3. Grid Reachability Analysis 25
2.4. Human skeleton as a topological space 30
2.5. Human as a dynamic system 30
2.6. Inference mapping 31
2.7. Motion planning with Reachability Analysis 36
2.8. Motion planning with Belief Sets 37
2.9. Collision avoidance with Belief Sets 38
2.10. Different Markov models 42
2.11. Value iteration in the regular grid 43
2.12. Causal Inference 44
2.13. Conditioning and intervening 45
2.14. Interventions in cyclic experiments 45
2.15. Flow chart of cooperative interacting vehicles 46
2.16. Vehicle set computation with nonlinear dynamics 54
2.17. Pedestrian moving on a vectorfield 56
2.18. Vehicle simulation with kinematic single-track model 56

3.1. Movement prediction with pedestrian 61
3.2. Timeline 61
3.3. Multimodal movement prediction and reachability analysis . 62
3.4. Adaptive reachability analysis and intelligent funnel prediction 63
3.5. Urban environment as topological space 63
3.6. Interaction between vehicle and pedestrian 64
3.7. Outcome of a driving situation 64
3.8. Description of future positions, velocities and control inputs . 66
3.9. Prediction of all vertices 67
3.10. Optimization with and without static obstacle 68
3.11. Set prediction and reachability analysis with zonotopes . . . 71
3.12. Stochastic process for maximal parameters 74
3.13. Example of set-based movement prediction 76
3.14. An example for urban network 77
3.15. Multi-vector-field and deforming homogeneous sets 80
3.16. Deforming sets with (group) actions 81
3.17. Observable Markov-Chain 82
3.18. Inference with sequential Markov-Chains. 83
3.19. Observable state sets 83
3.20. Reachability analysis and machine learning 85
3.21. Concept for funnel prediction 88
3.22. Set based movement prediction 89
3.23. Concept for adaptive reachability analysis 89
3.24. Reachability analysis and adaption of maximal parameters . . 89
3.25. Stochastic process and data-based belief sets 91
3.26. Movement prediction 93
3.27. Transformed path 96
3.28. Curved path 97
3.29. Circle Bridge 97
3.30. Human locomotion 98
3.31. Classical representation 99
3.32. Transformed grid 102
3.33. Optimal control on a manifold 103
3.34. Intersection in Graz 105
3.35. Intersection in Graz 105
3.36. Campus in Graz 106
3.37. Starting- and end-positions 108

3.38. Tangency space with voronoi diagram 109
3.39. Causal models and reasoning 113
3.40. Spiral reach set 116
3.41. Reachability analysis 117
3.42. Modeling of pedestrian movements 119
3.43. Epistemic and aleatory uncertainty 120
3.44. Reachable sets instead of Multivectorsystem 121
3.45. Support function and over-approximation 122
3.46. Three dimensional funnel 123
3.47. Extrapolation of the support function 124
3.48. Results of the hypothetical example (observational setting) . . 129
3.49. Results of the hypothetical example (interventional setting) . 129
3.50. Results of the hypothetical example (counterfactual setting) . 130
3.51. Causal relationship 135
3.52. Causal relationship: cognition, actuation, and interior sensing 135
3.53. Cognitive MDP and cognitive POMDP 136
3.54. Synthetic movement with causal intervention 137
3.55. Comparison between technical and human-centered systems 139
3.56. The interaction between the vehicle and pedestrian 140
3.57. Cognitive MDP 141
3.58. Cognitive POMDP 143
3.59. Legend and testing platform 143

4.1. Trajectory representation 151
4.2. Simulation program in Julia programming 151
4.3. Obstacle avoidance constraint for a current situation 152
4.4. Simulation of the vehicle with pedestrian 155
4.5. Simulation results 156
4.6. Different target distributions 159
4.7. Cumulative area of reachable sets 161
4.8. Connected reachable sets 162
4.9. Connected reachable sets 162
4.10. Maximal velocity and acceleration 165
4.11. Simulation results for different pedestrian models 166
4.12. Simulation time for different pedestrian models 167

A.1. Topological space 183

A.2. Manifold 184
A.3. Earth as a manifold 187
A.4. Set representation 189
A.5. Minkowski-Sum 190
A.6. Table for different (convex-)set representations 191

B.1. Taxonomy 195
B.2. Brodman atlas and selection of different activation regions . . 202
B.3. Brain and Body simulation technologies 203

C.1. Simulation of a vehicle with pedestrian 208
C.2. Simulation of a vehicle with pedestrian 209
C.3. Simulation of a vehicle with pedestrian 210
C.4. Simulation results 211
C.5. Urban environment as a manifold 214
C.6. Pedestrian- and vehicle sets 220

List of Algorithms

1. Algorithm for Reachability Analysis 22

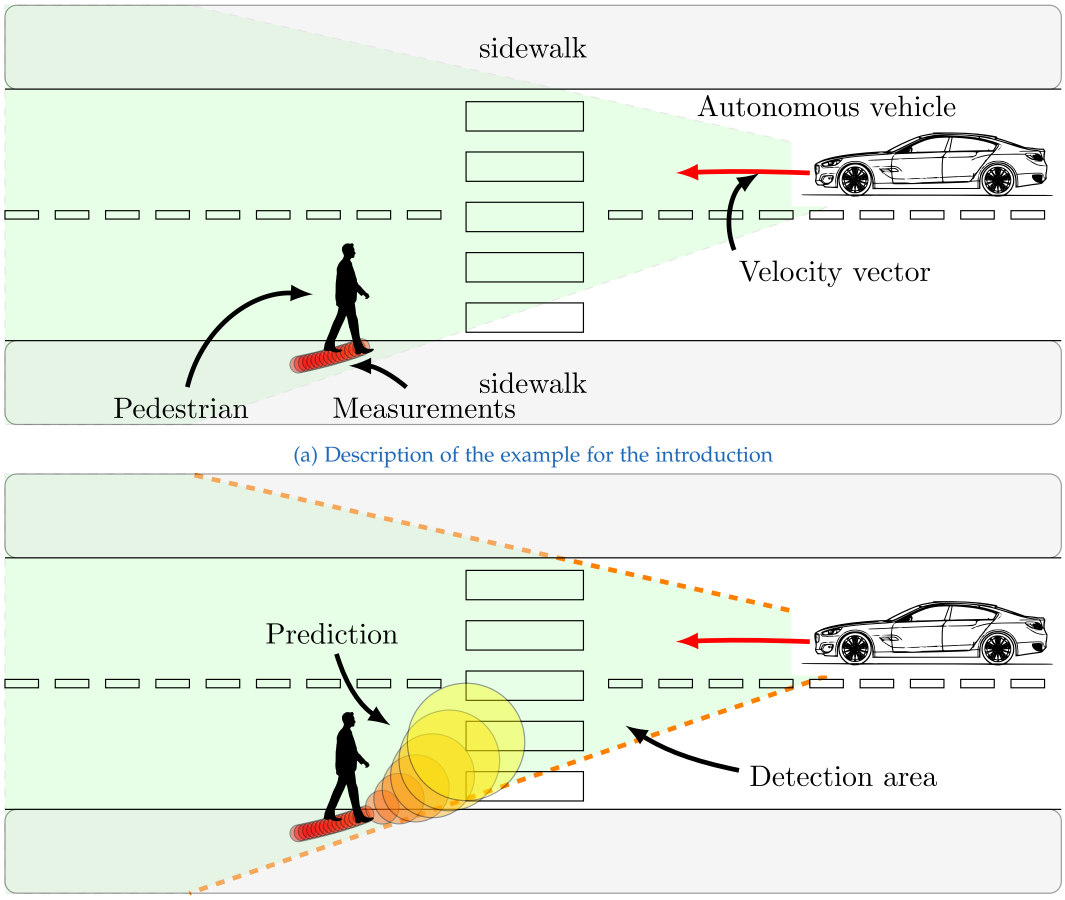
2. Monte-carlo sampling in a manifold cell 215
3. Motion planning and -prediction 216
4. Algorithm for reachability analysis in a manifold cell 216
5. Algorithm for set-deformation of a zonotope in a manifold cell 217
6. Local set deformation in a manifold cell 218
7. General set-based-movement-prediction 218
8. Set-based-movement-prediction with static support function . 219

1. Introduction

Section 1.1 gives the motivation of the thesis, section 1.2 specifies the current state of the art, and section 1.3 specifies the thesis contributions and the publications published during the Ph.D. thesis. The non-goals are summarized in section 1.4.

1.1. Motivation

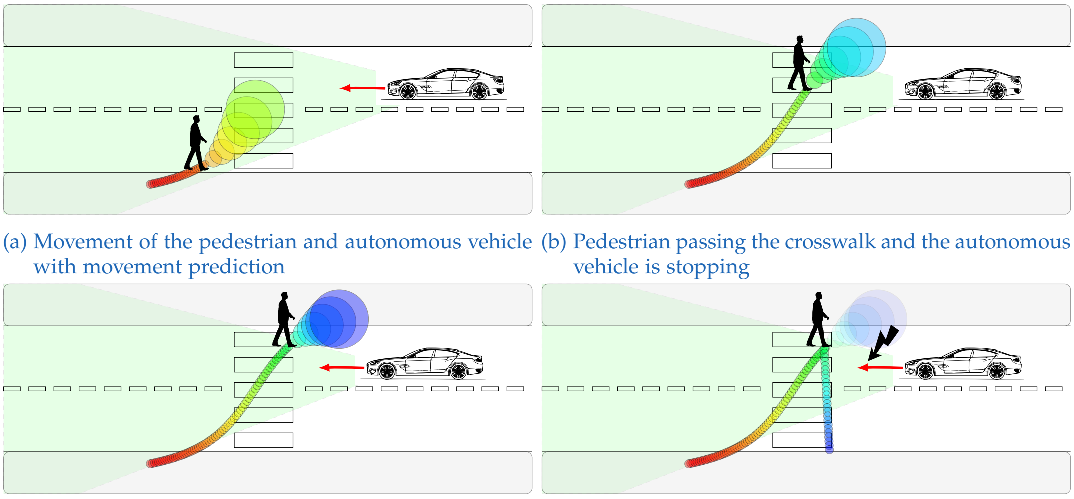
During the Covid 19 pandemic, the number of road accidents reduced drastically from 2.6 to 2.2 million accidents (statistics from 2021 from [Kords, 2021]). The reduced mobility during lockdowns causally influenced this number of accidents. Nevertheless, traffic accidents are too serious. This thesis contributes to reducing accidents. Pedestrians are the most influential group admitted to hospital after a non-fatal road accident ([Observatory, 2018], [Safetynet, 2009]). The fatality of a collision depends on multimodal factors (age of the person, speed of the vehicle, and others) [Observatory, 2018], [Safetynet, 2009]. Safety and collision avoidance play a crucial role in developing autonomous vehicles in accepting this new technology. However, efficiency (traffic speed in urban environments) is also essential for accepting autonomous vehicles. This thesis focuses on autonomous driving for restricted areas(with speed limitations), where direct interaction between autonomous vehicles and pedestrians is plausible. This topic has scientific, philosophical, and social value. The scientific value is how we develop autonomous vehicles to understand human behavior. Human behavior depends on several factors (intention change, body movements, and others). The philosophical component relates to the freedom of vulnerable road users and whether the actions are predictable. The development of safe and intelligent transport has an enormous social impact. The driver must make intuitive decisions and compromise on contrary targets (speed vs. safety). In some situations, a driver has to take some risks into account. This risk-taking might lead to some decision conflicts. The cognitive ability of the driver's brain can predict a pedestrian's future movements and decide which future trajectory to take. This introduction should give a simple real-world example in an urban environment for an intuitive introduction and an application for an automotive use-case, and figure 1.1 shows an example for addressing the topics discussed in the Ph.D. thesis. It illustrates an urban environment with a street, two sidewalks on each side, and a crosswalk. A vehicle is coming in the right direction. The autonomous vehicle drives and gets information on the historical positions of the pedestrian with the use of sensors. It might be possible to predict the person's movements to avoid collisions. However, the degree of certainty to trust these predictions cannot be well-defined (unknown intention of the pedestrian and situation factors). In figure 1.2a the vehicle is driving, but due to a movement prediction and the recognition of the crosswalk the vehicle stops in figure 1.2b and figure 1.2c. The autonomous vehicle predicts that the pedestrian will reach the sidewalk, and the vehicle will drive in the figure. If the movement prediction is false, it might lead to a collision between the vehicle and the pedestrian (compare figure 1.2d). This example is further used in chapter 3 to highlight developed concepts.



(a) Description of the example for the introduction

(b) Perception of the pedestrian with sensors

Figure 1.1.: Introduction example



(a) Movement of the pedestrian and autonomous vehicle with movement prediction

(b) Pedestrian passing the crosswalk and the autonomous vehicle is stopping

(c) Autonomous vehicle predicts movement of the pedestrian to the sidewalk

(d) False prediction of the movements due to an intention change of the pedestrian and collision risk

Figure 1.2.: Scenario of pedestrian movement prediction

1.2. Focused problem

This section discusses current challenges of existing state of the art (section 1.2.1) and the gap of existing methods (section 1.2.2).

1.2.1. Challenges

The current challenge in autonomous vehicles is to program computational intelligence to perform adequate and safe driving policies. An autonomous vehicle cannot observe all relevant factors influencing the pedestrian, so there exists a non-reducible uncertainty in each movement prediction (Aleatoric uncertainty). A movement prediction with absolute certainty is impossible, and some risky driving policies could lead to collisions. How can an autonomous vehicle handle this decision-risk dilemma technically? Reachable sets addressing pedestrians' situations compute worst-case scenarios, which might lead to conservative motion planning. This thesis can not solve the question of which situation the reachable sets (worst-case) or the unreliable movement prediction (risky bypassing). On the one hand, a pedestrian's sure and confident movement prediction is impossible. On the other hand, the autonomous vehicle must admit risk-taking to not stop in each situation. These contradicting targets lead to the challenge of developing autonomous vehicles.

1.2.2. Gap of existing methods

Reachable sets use simple formulations for dynamic systems to compute worst-case scenarios. On the other side, machine learning models, such as movement prediction, try to predict future positions. However, this thesis assumes that there is no absolute certainty of the movement prediction for pedestrians (because the person's intention is not measurable). Therefore this thesis introduces belief sets. We use machine learning models to predict the most probable areas in future movements. Reachability analysis might reference the belief set, which computes the plausible areas (what areas are physically possible under physical laws). How reachable sets could be adapted to situations by machine learning is a new field for intelligent computing funnels capturing the pedestrian trajectory. Causal inference forms another aspect of the interaction between the vehicle and pedestrian. There are methods for causal inference but no direct application to the interaction process between vehicle and pedestrian.

1.3. Contribution

Subsection 1.3.1 discusses the topics of the thesis and summarize the achievements (section 1.3.2) and publications (section 1.3.3).

1.3.1. Topic of the thesis

The thesis explores set-based movement prediction algorithms. It focuses on complex dynamic systems with elements of aleatoric- and epistemic uncertainty. Aleatoric uncertainty is non-reducible uncertainty. Epistemic uncertainty is reducible uncertainty due to measurements. An example is the cognitive process of human decisions and the resulting actions and movements (compare section B.2). It should highlight some challenges in understanding the whole complexity of the interaction between autonomous vehicles and humans. The thesis discusses set-based prediction models (adaptive reachability analysis and funnel prediction) and new environment descriptions using differential geometry and causal inference. The causal inference is for the interaction process in chapter 3 and gives a proposal for a pedestrian in the loop environments for the basic research on the biological cybernetic cycle (compare section B.2). This thesis gives an overview of the complexity of the interaction with vehicles and pedestrians and considers aspects from the environment, biomechanics, and neuroscience (compare section B.2). This thesis uses the concept of epistemic and aleatoric uncertainty. This distinction helps to see the limitations of movement prediction.

1.3.2. Achievements

The main contributions of this thesis are the development of set-based movement prediction algorithms for uncertain dynamic systems (e.g., pedestrian movements). The urban environment might influence a road user in several ways. It is not certain that a movement prediction algorithm leads to a guaranteed good performance for a pedestrian, and the intention is unknown. A common approach uses multi-modal movement predictions (several predictions by different parameters). We present approaches to using adaptive set-based methods. In contrast to reachability analysis, do not lead to worst-case scenarios but also offers a risk for the motion planning. The idea is to use data-based approaches to adapt the set-based methods for each situation. This thesis also highlights topological spaces, where funnels connect spatial areas. These approaches lead to new mathematical descriptions for pedestrian movements.

1.3.3. Scientific publications

The thesis is a summary of some previous scientific publications. The thesis focus on applying the prediction of aleatoric dynamic systems with a use case to human locomotion and the interaction process to an autonomous vehicle ([Hartmann et al., 2017a], [Hartmann and Watzenig, 2019a], [Hartmann et al., 2018c], [Hartmann, 049A] [Hartmann, 897A] [Hartmann et al., 2018b] [Schratter et al., 2019], [Hartmann and Watzenig, 2019b], [Hartmann and Watzenig, 2019b]). The appendix shows the complexity of human locomotion, a developed theoretical model, and the need for a test environment for the basic research on human locomotion and the application to "Pedestrian in the Loop" environment for the automotive industry [Hartmann, 774A,Hartmann, 400A,Hartmann, 049A], [Hartmann et al., 2017b,Hartmann et al., 2018d], [Hartmann et al., 2018a] [Aksjonov et al., 2019]. The first publications originated during the European Marie Curie research fellowship ITEAM. [Hartmann et al., 2017a] originated at the Institute for Risk and Uncertainty at the University of Liverpool. [Hartmann et al., 2017a] proposes a motion planning algorithm under uncertain environment conditions modeled via Monte-Carlo simulations and models for uncertainty quantification. The work [Hartmann et al., 2017b,Hartmann et al., 2018d] was a collaboration project with scientists from the KU Leuven university and proposes a Pedestrian in the Loop test environment under virtual, augmented reality. The paper [Hartmann et al., 2018a] was the result of the research stay at the University of Compiègne in France. This work proposes a similar approach of a test environment, where autonomous vehicles are tested with drones with real measurements. [Hartmann and Watzenig, 2019b] proposes modeling pedestrian behavior in topological spaces and motion planning of autonomous vehicles with adaptive reachable sets. [Hartmann et al., 2018b] emerged at the University of Pavia in Italy during a research stay and proposes adaptive motion planning under uncertain environments with a data-based reachability approach. [Hartmann et al., 2018c] provides extensive parameter variation of the motion planning under uncertainty. [Hartmann and Watzenig, 2019a] proposes a Mixed Integer Learning approach for motion planning with a new jerk-constrained model for adaptive reachability analysis. In [Schratter et al., 2019] we tested the adaptive reachability analysis approach with the combination of a real autonomous vehicle collision-avoidance system. In [Aksjonov et al., 2019] a collaboration project of the European ITEAM project proposed a Driver–Vehicle–Environment. The research stay at the University of Berkeley has beneficially influenced this thesis. [Hartmann, 774A, Hartmann, 897A, Hartmann, 400A, Hartmann, 049A] are developed patent applications for the intellectual protection of some innovations.

1.3.4. Structure of the thesis

This section should guide the reader through the document. Contributions are marked with green color

Overview

Figure 1.3 gives an overview over the main-chapters of the Ph.D. thesis. It starts with an introduction, gives an overview of the state of the art and continuous with the developed concepts. Evaluation is done with simulations and finishes with the conclusion.

Chapter 1 - Introduction

Figure 1.4 shows the sections of the first chapter. The thesis starts with the motivation for the topic 1.1 of this work. We continue with the problem of movement prediction of pedestrians and the existing gap of existing methods in section 1.2. Section 1.3 summarizes the achievements and the

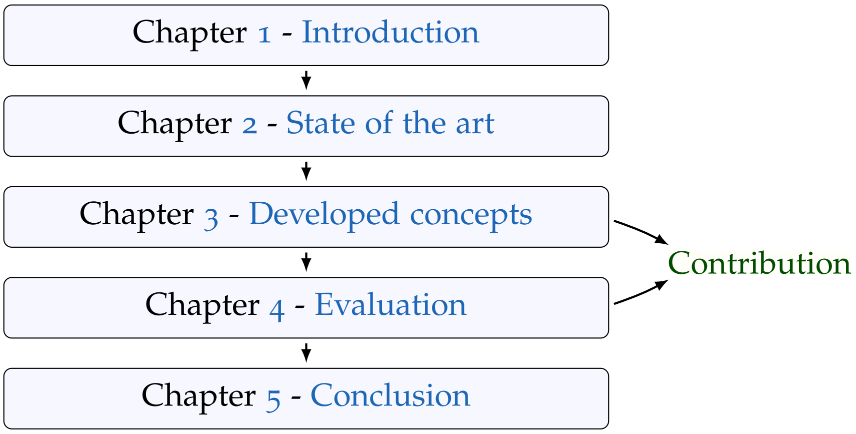


Figure 1.3.: Overview of the Ph.D. thesis

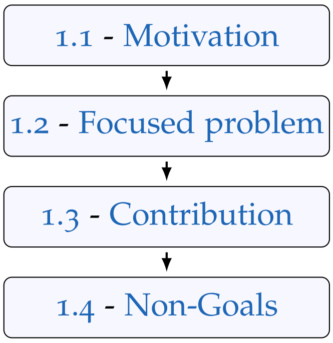


Figure 1.4.: Structure of chapter 1 - Introduction

contribution. The section 1.4 comment non-goals to show boundaries of the thesis.

Chapter 2 - State of the art

Chapter 2 gives an overview of the state-of-the-art and the structure is visualised in figure 1.5. It discusses methods like reachability analysis in section 2.1 movement prediction in section 2.2.1, reinforcement learning 2.2.2 or causal inference in section 2.2.3. In approaches from cooperative interacting vehicles in section 2.3 it focuses only on topics like motion

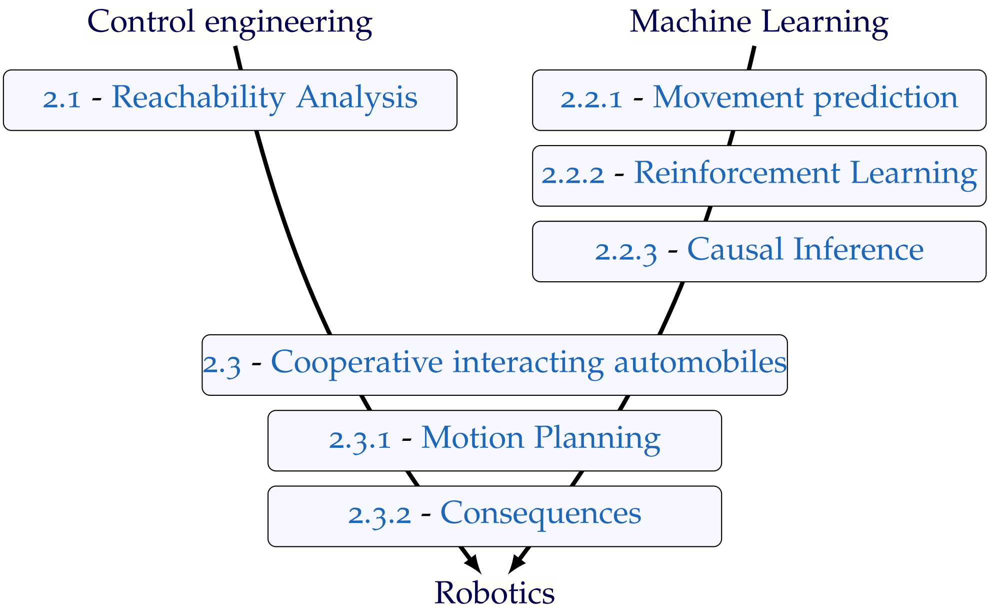

Control engineering
Machine Learning
2.1 - Reachability Analysis
2.2.1 - Movement prediction
2.2.2 - Reinforcement Learning
2.2.3 - Causal Inference
2.3 - Cooperative interacting automobiles
2.3.1 - Motion Planning
2.3.2 - Consequences
Robotics

Figure 1.5.: Structure of chapter 2 - State of the art

planning in section 2.3.1 and discusses limitations of technical approaches in section 2.3.2.

Chapter 3 - Developed concepts

Chapter 3 focuses on the set-based movement prediction algorithms, with causal inference, new human locomotion models and topological spaces. The structure is visualized in figure 1.6. The chapter starts with section 3.1 to give an overview of the developed concepts. Section 3.2 discusses a standard motion planning algorithm to see how the pedestrian models could be tested. The idea of this chapter is to introduce new models for adaptive set deformations in section 3.3 and intelligent funnel prediction 3.4. Two theoretical section discusses the mathematical approaches for human locomotion to represent the urban environment of pedestrians with manifolds in section 3.5 and the interaction between vehicle and pedestrian with causal inference in section 3.6. Section 3.7 proposes a theoretical model of human movements and discusses the need for a new Pedestrian-in-the

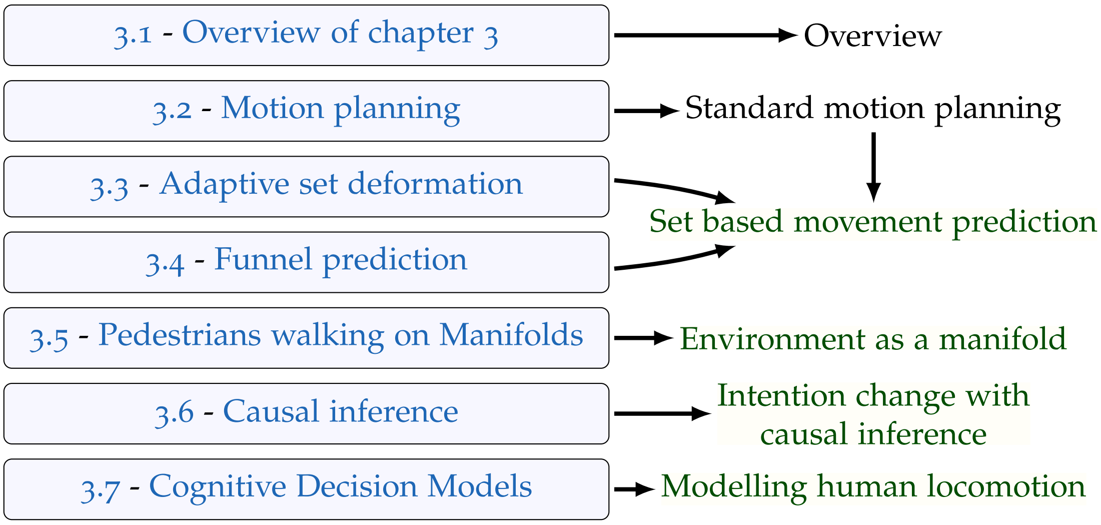

3.1 - Overview of chapter 3
3.2 - Motion planning
3.3 - Adaptive set deformation
3.4 - Funnel prediction
3.5 - Pedestrians walking on Manifolds
3.6 - Causal inference
3.7 - Cognitive Decision Models
Overview
Standard motion planning
Set based movement prediction
Environment as a manifold
Intention change with causal inference
Modelling human locomotion

Figure 1.6.: Structure of chapter 3 - Developed concepts

Loop environment for basic research on the biological cybernetic cycle of human locomotion and to ensure pedestrians' safety.

Chapter 4 - Evaluation

Chapter 4 present some a selection of further simulations for the evaluation and figure 1.7 shows the visual structure. Section 4.1 gives mathematical details of the simulator with some simulation runs presented in section 4.2 and different pedestrian models. Section 4.3 presents models with causal inference to formulate the spatial target for the intention of a virtual agent. Section 4.4 shows connected reachable sets in topological spaces. To see the performance over different pedestrian models one can read section 4.5.

Chapter 5 - Conclusion

The thesis finishes with a conclusion and an outlook for future work in chapter 5 (compare figure 1.8). Section 5.1 discusses the thesis and the

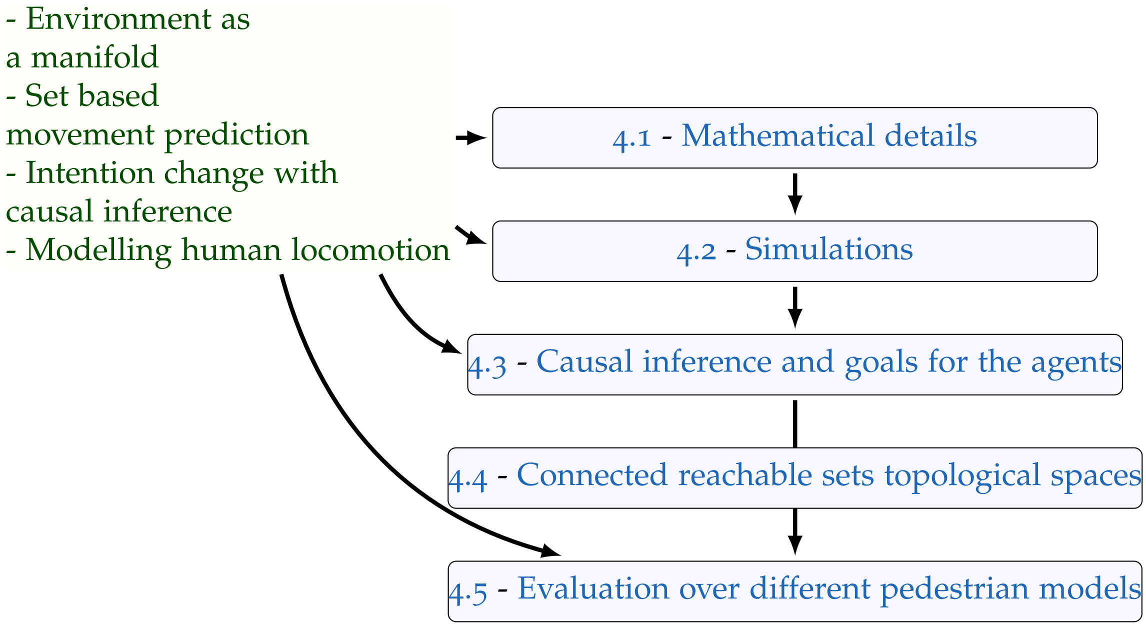

- Environment as a manifold
- Set based movement prediction
- Intention change with causal inference
- Modelling human locomotion
4.1 - Mathematical details
4.2 - Simulations
4.3 - Causal inference and goals for the agents
4.4 - Connected reachable sets topological spaces
4.5 - Evaluation over different pedestrian models

Figure 1.7.: Structure of chapter 4 - Evaluation

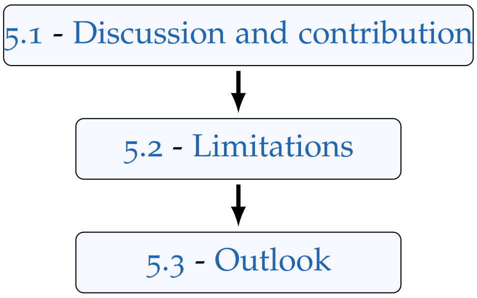
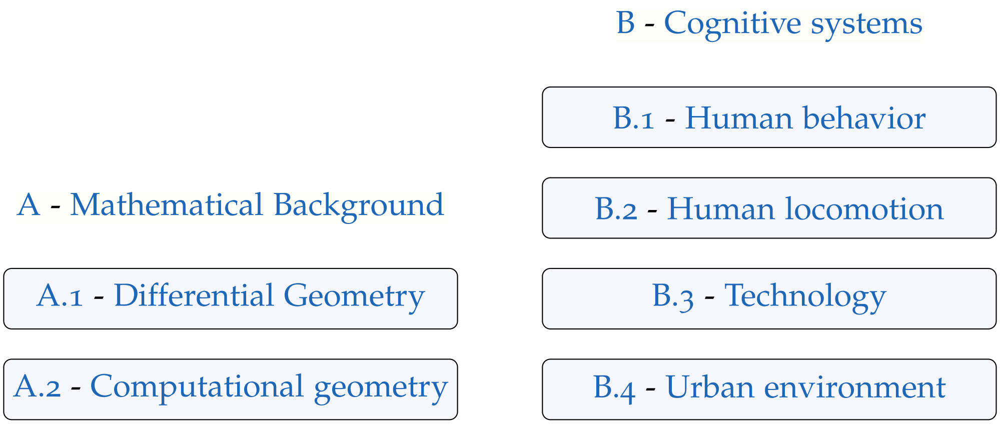


Figure 1.8.: Structure of chapter 5 - Conclusion

contribution. Section 5.2 analyses the limitations of the proposed approaches and finishes the thesis with an outlook in section 5.3.

Appendix A

For better understanding, the appendix compare figure 1.9 gives an introduction to some mathematical concepts in chapter A (differential geometry A.1 and computational geometry A.2). This section might be helpful for the interested reader to understand the use of topological spaces and the concept of hull computation and adaptive reachability analysis. Chapter B discusses the cognitive systems and some basic statements to see the complexity of the cybernetic cycle of human locomotion (human intelligence, body, and environment). Section B.1 discusses basic information about the cognition of human decision making and the resulting movements in section B.2. Section B.3 discusses current technology of measurements and section B.4 more on the environmental measurement technologies.



(a) Structure of chapter A - Mathematical Background

(b) Structure of the chapter B - Cognitive systems

Figure 1.9.: Structure of the appendix

1.4. Non-Goals

The complexity of human locomotion and the technical technology of autonomous vehicles is vast. Some non-goals should also guide the reader through the thesis. These non-goals should show what is not expected from the content of the thesis. It is not the intention of the thesis to follow the following goals:

- The development of a perfect movement prediction of pedestrians, which predicts each position of a random person and an unexpected situation with absolute certainty and precision, is not the intention of the thesis. There are some reasons why this is not realistic and discussed during the thesis.
- There will be no development of perception algorithms and optimization of motion planning algorithms. There will be motion planning algorithms (MPC approach) presented. However, the focus is more on developing pedestrian models with Reachability Analysis, Markov Decision Processes, and Causal Inference on manifolds.
- Causal inference is a relatively new field in science. There exist not many structure learning algorithms for the practice. That is why only some simple applications are proposed for modeling the interaction. The counterfactual analysis is also only described in some contexts.

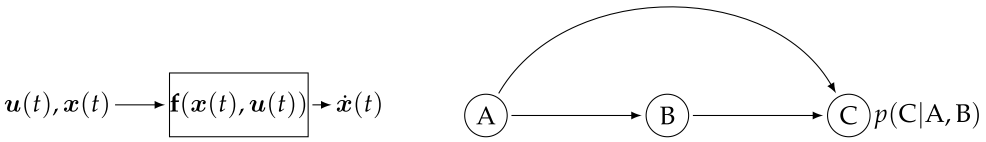
- The state of the art in this thesis is not complete. There exist different methods and applications of vehicle control and perception technologies. Computer vision, sensor- and measurement technologies are not in focus and some literature is summarized in B.3 with a focus on pedestrian perception technologies. Systems of the vehicle drivetrain, the stabilization of the vehicle, and other technologies to increase the comfort and safety like in advanced driver assistance systems can be found in [Bengler et al., 2014].
- It was not the goal of measuring and perceiving information from the environment and pre-processing (e.g., computer vision, pedestrian tracking, automatic semantic segmentation).

2. State of the art

This chapter starts with reachability analysis in section 2.1 and with deterministic system models (compare figure 2.1a). Topics from machine learning are described in section 2.2 (with probabilistic descriptions, compare figure 2.1b) are used. Statistical movement prediction in section 2.2.1, reinforcement learning in section 2.2.2 and causal inference 2.2.3 are also used in the thesis. Section 2.3 uses these methods for cooperative interacting automobiles, where motion planning algorithms in section 2.3.1 and some technical limitations in section 2.3.2 are discussed with some non-technical aspects of human locomotion.

2.1. Reachability Analysis for worst-case scenarios

Reachability analysis was originally used in testing large software projects, where system errors might lead to fatal consequences. It is nowadays also used in dynamic systems, like vehicles, where safety plays a key-role (e.g. for collision-avoidance). Normally the control input $\boldsymbol{u}(t)$ or the whole constant control-input set $\mathcal{U}$ are assumed to be known. We start with some basics from


$\boldsymbol{u}(t), \boldsymbol{x}(t)$
$\mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$
$\dot{\boldsymbol{x}}(t)$
A
B
C
$p(\mathrm{C}|\mathrm{A}, \mathrm{B})$

(a) Deterministic system models

(b) Probabilistic perspective

Figure 2.1.: Deterministic system and machine learning model

control-engineering and definitions from [Lutz and Wendt, 2007, Föllinger, 1985]. Following system is defined by an ordinary differential equation

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \\ t, t_0 &\in \mathbb{R}, \\ \boldsymbol{x}(t_0) &= \boldsymbol{x}_0 \in \mathbb{R}^n, \\ \boldsymbol{x}(t) \in \mathbb{R}^n, \boldsymbol{u}(t) &\in \mathbb{R}^m \; \forall \; t \in [t_0, t] \end{aligned} \tag{2.1}$$

In the autonomous case the control input $\boldsymbol{u}(t) = \mathbf{0}$ disappears. The nonlinear function $\mathbf{f}$ is a mapping from the domain-space $\mathcal{X} \times \mathcal{U}$ to the image space with $\mathbf{f} : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$ with the mapping $(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mapsto \dot{\boldsymbol{x}}(t)$. We might define the whole domain-space as an entire set $\mathcal{X} \subseteq \mathbb{R}^n$ or sets $\{\mathcal{X}_i\}_{i \in \mathcal{I}}$ connected via a topology $\mathcal{T}$ (differential geometry and topological spaces/manifolds compare section A.1) and a label set $\mathcal{I}$. If the actual function of $\mathbf{f}(\cdot)$ is unknown or too complex, it might be necessary to formulate an approximation by $\hat{\mathbf{f}}(\cdot)$in the whole domain-space or a sub-domain. The future positions of the pedestrian are uncertain. The intention of the pedestrian is normally unknown. Therefore, reachability analysis is used to find all future positions the pedestrian could reach. So the idea of this section is not to predict the exact position of the pedestrian but rather the worst-case possibilities, where a person could be hypothetical. The function $\mathcal{R}(t_r, t_0, \mathcal{X}_0, \mathcal{U}(t))$ describes the reachable tube of a dynamic system with initial state set $\mathcal{X}_0$, the time-varying control input $\mathcal{U}(t)$, from initial continuous timestep t_0 and to step t_r. A shortcut is $\mathcal{R}[t_r, \mathcal{U}(t)]$ is used if it is clear from the context which initial time and initial state space were mentioned. $\mathcal{R}[t_r]$ is used, when the control input $\mathcal{U}$ is constant over time. The definition of reachability analysis [Althoff, 2010] is:

Definition 2.1.1. Reachable Set at a point in time (adapted from [Althoff, 2010]): Given is a dynamical system $\dot{\boldsymbol{x}} = f(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}(t))$, where t is the time, u is the control input and $\boldsymbol{\theta}$ are the parameters. The set of possible initial states, the input, and the parameters are bounded by sets: $\boldsymbol{x}(t_0) \in \mathcal{X}_0 \subset \mathbb{R}^n, \boldsymbol{u} \in \mathcal{U}(t) \subset \mathbb{R}^m$ and $\boldsymbol{\theta} \in \mathcal{P} \subset \mathbb{R}^p$. The reachable set at a certain point in time t_r is defined as the union of possible system states at $t = t_r$:

$$\mathcal{R}[t_r,\mathcal{U}(t)] = \mathcal{R}(t_r, t_0, \mathcal{X}_0,\mathcal{U}(t)) = \{\boldsymbol{x}(t_r) = \int_{t_0}^{t_r} \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}(t)dt| \\ \boldsymbol{x}(0) \in \mathcal{X}_0, \boldsymbol{u}([t_0, t_r]) \in \mathcal{U}(t), \boldsymbol{\theta}([t_0, t_r]) \in \mathcal{P}\} \tag{2.2}$$

$\boldsymbol{u}([t_0, t_r])$ is a short form of $\cup_{t\in[t_0,t_r]}\boldsymbol{u}(t)$.

In this thesis we will use machine learning to predict some parts of the definition for the parameter set $\hat{\mathcal{P}} \simeq \mathcal{P}$, $\hat{\mathcal{U}}(t) \simeq \mathcal{U}(t)$ or $\hat{\mathcal{X}}_0 \simeq \mathcal{X}_0$ (what we understand here as adaptive reachability analysis). The definition for a longer time interval:

Definition 2.1.2. Reachable Set of a Time Interval (RSTI) [Althoff, 2010]: The reachable set of a time interval is the union of reachable sets at points in time within the interval $t \in [0, r]$

$$\mathcal{R}([t_0, t_r]) = \bigcup_{t\in[t_0,t_r]} \mathcal{R}[t] \tag{2.3}$$

For simplicity we could linearize the nonlinear system with a Taylor series with a linearization working point $\boldsymbol{z}^* = \begin{bmatrix} \boldsymbol{x}^* \\ \boldsymbol{u}^* \end{bmatrix}$ [Alanwar et al., 2021]. We also assume a Lipschitz nonlinear twice differentiable function:

$$\mathbf{f}(\boldsymbol{z}) = \mathbf{f}(\boldsymbol{z}^*) + \frac{\partial \mathbf{f}(\boldsymbol{z})}{\partial \boldsymbol{z}}\bigg|_{\boldsymbol{z}=\boldsymbol{z}^*} (\boldsymbol{z} - \boldsymbol{z}^*) + \underbrace{\frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}^*)^T \frac{\partial^2 \mathbf{f}(\boldsymbol{z})}{\partial^2 \boldsymbol{z}}\bigg|_{\boldsymbol{z}=\boldsymbol{z}^*} (\boldsymbol{z} - \boldsymbol{z}^*) + \dots}_{\mathrm{L}(\boldsymbol{z})} \tag{2.4}$$

We formulate the function with an error-term $\mathrm{L}(\boldsymbol{x}, \boldsymbol{u})$:

$$\mathbf{f}(\boldsymbol{x}, \boldsymbol{u}) = \mathbf{f}(\boldsymbol{x}^*, \boldsymbol{u}^*) + \underbrace{\frac{\partial \mathbf{f}(\boldsymbol{z})}{\partial \boldsymbol{z}}|_{\boldsymbol{x}=\boldsymbol{x}^*,\boldsymbol{u}=\boldsymbol{u}^*}}_{\tilde{\boldsymbol{A}}}(\boldsymbol{x} - \boldsymbol{x}^*) + \\ \underbrace{\frac{\partial \mathbf{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}|_{\boldsymbol{x}=\boldsymbol{x}^*,\boldsymbol{u}=\boldsymbol{u}^*}}_{\tilde{\boldsymbol{B}}}(\boldsymbol{u} - \boldsymbol{u}^*) + \mathrm{L}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.5}$$

For better understanding we could find a linear subspace in the manifold of the function domain with the Taylor approximation. For more details read section A.1:

$$\mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \begin{bmatrix}\mathbf{f}(\boldsymbol{x}^*, \boldsymbol{u}^*) & \tilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}}\end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{x} - \boldsymbol{x}^* \\ \boldsymbol{u} - \boldsymbol{u}^* \end{bmatrix} + \mathrm{L}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.6}$$

For simplicity we assume that we have a total linear system for the whole function domain and for the whole time t:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A} \cdot \boldsymbol{x}(t) + \boldsymbol{B} \cdot \boldsymbol{u}(t),\ \boldsymbol{A} \in \mathbb{R}^{n \times n},\ \boldsymbol{B} \in \mathbb{R}^{n \times m} \tag{2.7}$$

The solution of the linear system is

$$\boldsymbol{x}(t) = \boldsymbol{x}_h(t) + \boldsymbol{x}_p(t) = e^{\boldsymbol{A}(t-t_0)} \cdot \boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{\boldsymbol{A}(t-\tau)} \cdot \boldsymbol{B} \cdot \boldsymbol{u}(\tau) d\tau \tag{2.8}$$

It is consisting of the homogenous $\boldsymbol{x}_h(t)$ and particular solution $\boldsymbol{x}_p(t)$. The transition-matrix is $\phi(t) = e^{\boldsymbol{A}t}$. Then we get:

$$\boldsymbol{x}(t) = \phi(t - t_0) \cdot \boldsymbol{x}(t_0) + \int_{t_0}^{t} \phi(t - \tau) \cdot \boldsymbol{B} \cdot \boldsymbol{u}(\tau) d\tau \tag{2.9}$$

If we consider the initial state set $\mathcal{X}(t_0) \subset \mathbb{R}^n$ instead of a single initial state vector $\boldsymbol{x}(t_0)$ and a control input set $\mathcal{U}(t) \subset \mathbb{R}^m$, we could write the system as:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t),\ \boldsymbol{x}(0) \in \mathcal{X}(t_0) \subset \mathbb{R}^n,\ \boldsymbol{u}(t) \in \mathcal{U}(t) \subset \mathbb{R}^m \tag{2.10}$$

The reachable sets are then a set of all sums from the homogeneous and particular solutions (adapted from [Althoff, 2010]:

$$\mathcal{R}(t, \mathcal{U}(t)) = \{\boldsymbol{x}_h(t) + \boldsymbol{x}_p(t, \mathcal{U}(t)) | \boldsymbol{x}_h(t) \in \mathcal{H}(t), \boldsymbol{x}_p(t) \in \mathcal{P}(t, \mathcal{U}(t))\} \tag{2.11}$$

With the formulation of the Minkowski-sum:

$$\mathcal{R}(t, \mathcal{U}(t)) = \mathcal{H}(t) \oplus \mathcal{P}(t, \mathcal{U}(t)) \tag{2.12}$$

The reachable sets have the semigroup property (adapted from [Kurzhanski and Varaiya, 2002]) for constant $\mathcal{U}$:

$$\mathcal{R}(t, t_0, \mathcal{X}_0) = \mathcal{R}(t, r, \mathcal{R}(r, t_0, \mathcal{X}_0)) \tag{2.13}$$

This has the consequence:

$$\mathcal{R}(t, t_0, \mathcal{X}_0) = \bigcup_{s \in [t_0, t]} \mathcal{R}(s, t_0, \mathcal{X}_0) \tag{2.14}$$

The set of all homogenous solutions $\mathcal{H}(t)$:

$$\mathcal{H}(t, t_0, \mathcal{X}_0) = \phi(t - t_0)\mathcal{X}(t_0), \ \mathcal{X}(t_0) \subset \mathbb{R}^n \tag{2.15}$$

The set of all particular solutions:

$$\mathcal{P}(t, t_0, \mathcal{X}_0, \mathcal{U}(t)) = \int_{t_0}^{t} \phi(t - \tau) \cdot \boldsymbol{B} \cdot \mathcal{U}(\tau) d\tau \tag{2.16}$$

If we could find an constant upper bound $\tilde{\mathcal{U}}$ with $\boldsymbol{B} \cdot \boldsymbol{u}(\tau) \in \tilde{\mathcal{U}} \ \forall \ \boldsymbol{u} \in \mathcal{U}(\tau), \tau \in [t_0, t]$, we could have:

$$\tilde{\mathcal{P}}(t) = \int_{t_0}^{t} \phi(t - \tau) \cdot \tilde{\mathcal{U}} d\tau = \int_{t_0}^{t} \phi(t - \tau) \cdot d\tau \cdot \tilde{\mathcal{U}} \tag{2.17}$$

The same $\tilde{\mathcal{P}}$ for $t \in [0, r]$ (from [Althoff, 2010]):

$$\tilde{\mathcal{P}} = \boldsymbol{A}^{-1}(\phi(r) - \boldsymbol{I})\tilde{\mathcal{U}} \tag{2.18}$$

We switch now from continuous-time $t \in \mathbb{R}$ to a discretization with sample time T_S. We get the discrete time-set:

$$\mathbb{T} := \{k_i := i \cdot T_S | i \in \mathbb{N}_0^+, T_S > 0, T_S \in \mathbb{R}\} \tag{2.19}$$

We are looking for discrete reachable sets for worst-case analysis $\{\mathcal{R}_{k_i}\}_{i=1}^{N}$[1]

$$\mathcal{R}_{k_N} = \mathcal{R}(k_N, k_0, \mathcal{X}_0, \mathcal{U}) = \bigcup_{s \in \{k_0, k_1, \ldots, k_N\}} \mathcal{R}(s, k_0, \mathcal{X}_0, \mathcal{U}) \tag{2.20}$$

[1] The intersection of two sets $\mathcal{V}, \mathcal{U}$ is: $\mathcal{V} \cap \mathcal{U} = \{\boldsymbol{s} | \boldsymbol{s} \in \mathcal{U}, \boldsymbol{s} \in \mathcal{V}\}$. The union of two sets $\mathcal{V}, \mathcal{U}$ is: $\mathcal{V} \cup \mathcal{U} = \{\boldsymbol{a} \ \& \ \boldsymbol{b} | \boldsymbol{a} \in \mathcal{U}, \boldsymbol{b} \in \mathcal{V}\}$. Two sets $\mathcal{V}$, $\mathcal{U}$ are disjoint, if the intersection of both sets is empty $\mathcal{V} \cap \mathcal{U} = \emptyset$

The algorithm 1 describes the algorithm for reachability analysis [Girard, 2005].

Algorithm 1: Algorithm for Reachability Analysis

```
Output: $\mathcal{R}(k_N, k_0, \mathcal{X}_0)$
$\mathcal{P} \leftarrow \mathcal{X}_0$
$\mathcal{R} \leftarrow \mathcal{P}$
for $i = 0 \rightarrow N - 1$:
    $\mathcal{P} \leftarrow \mathcal{R}(k_{i+1}, k_i, \mathcal{P})$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{P}$
```

The abbreviation for the control input between the time range $[k_0, k_N]$:

$$\boldsymbol{u}[k_0, k_N] := \{\boldsymbol{u}_i | \boldsymbol{u}_i \in \mathcal{U}, i \in \{k_0, k_1, \dots, k_N\}\} \tag{2.21}$$

We define a shorthand for the autonomous reach set $\mathcal{R}_k^{\neg\mathcal{U}}$, which is necessary for set-based algorithms:

$$\mathcal{R}_{k_i}^{\neg\mathcal{U}} := \mathcal{R}_{k_i} \setminus \mathcal{U} \; \forall \; k_i \in \mathbb{T} \tag{2.22}$$

and $\mathcal{R}(k_N, k_0, \mathcal{X}_0) = \mathcal{H}(k_N, k_0, \mathcal{X}_0)$. For each state over time should always $\boldsymbol{x}_{k_i} \in \mathcal{R}_{k_i}$.

State set evolution

We have a dynamic system representing the vehicle the dynamic system

$$\dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{2.23}$$

and discrete system

$$\frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{T_S} = \mathbf{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{2.24}$$

The elements of all

$$\dot{\mathcal{X}}(t) = \bigcup_{\boldsymbol{x}(t) \in \mathcal{X}(t), \boldsymbol{u}(t) \in \mathcal{U}} \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{2.25}$$

The discrete case:

$$\mathcal{X}_{k+1} = \bigcup_{\boldsymbol{x}_k \in \mathcal{X}_k, \boldsymbol{u}_k \in \mathcal{U}} \boldsymbol{x}_k + \underbrace{T_S \cdot \mathbf{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)}_{\Delta \mathcal{X}_{k,k+1}} \tag{2.26}$$

The discrete case:

$$\Delta \mathcal{X}_{k,k+1} = \bigcup_{\boldsymbol{x}_k \in \mathcal{X}_k, \boldsymbol{u}_k \in \mathcal{U}} T_S \cdot \mathbf{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{2.27}$$

set based discrete evolution of the system:

$$\begin{aligned} \mathcal{X}_{k+1} &= \mathcal{X}_k \oplus \Delta \mathcal{X}_{k,k+1} \\ = \underbrace{\bigcup_{\boldsymbol{x}_k \in \mathcal{X}_k} \boldsymbol{x}_k}_{\text{state-dependent}} &\oplus \underbrace{\bigcup_{\boldsymbol{x}_k \in \mathcal{X}_k, \boldsymbol{u}_k \in \mathcal{U}} T_S \cdot \mathbf{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)}_{\text{state-action-dependent}} \end{aligned} \tag{2.28}$$

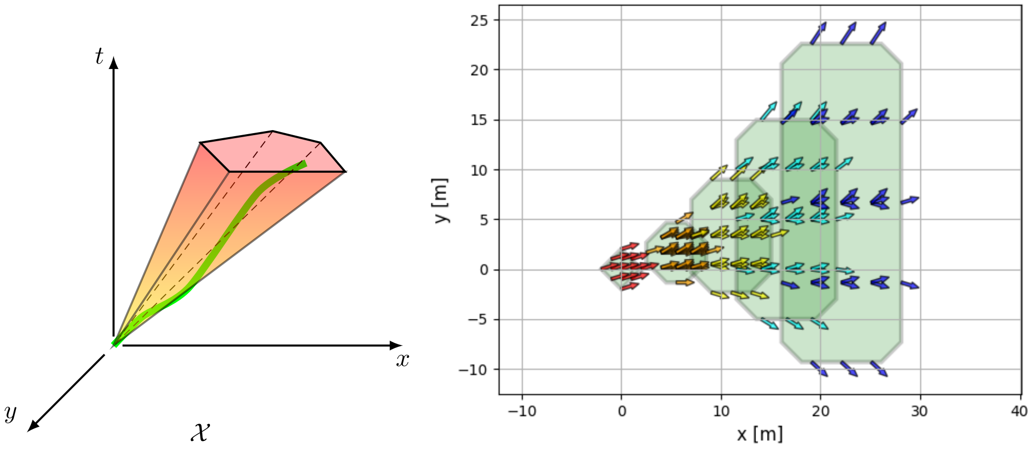
A formulation in recursive manner:

$$\begin{aligned} &\mathcal{X}_0 \\ \mathcal{X}_1 &= \mathcal{X}_0 \oplus \Delta \mathcal{X}_{0,1} \\ \mathcal{X}_2 = \mathcal{X}_1 \oplus \Delta \mathcal{X}_{1,2} &= \mathcal{X}_0 \oplus \Delta \mathcal{X}_{0,1} \oplus \Delta \mathcal{X}_{1,2} \end{aligned} \tag{2.29}$$

As described in [Girard et al., 2006] there exist also approaches with optimal control the boundaries in the most of outward manner. It is also showing a formulation

$$\mathcal{R}_{k_{i+1}} = \phi \mathcal{R}_{k_i} \oplus \mathcal{U} \tag{2.30}$$

with a bounded convex set $\mathcal{U}$. The derivation of that formula can be found in [Asarin et al., 2003]. There exist different approaches for ellipsoids, parallelograms, zonotopes, interval-functions, and support-functions for reachability analysis [Girard, 2005,Girard et al., 2006,Kurzhanski and Varaiya, 2002,Asarin et al., 2000,Althoff, 2010,Althoff and Frehse, 2016,Liu et al., 2017,Schilling, 2018,Bogomolov et al., 2019]. These different sets have different properties. Some sets are closed under some mathematical operation (e.g. multiplication), which means that the type of the set in the domain is the same type in the image space (also in the amount of sets). The algorithms also differ in the computational complexity. Origin of these approaches was software development applications to check models and algorithms to detect failures in software programs. [Koschi and Althoff, 2017a] presents an open-source tool for a set-based reachability analysis. The theory behind reachability analysis and the usage for autonomous vehicles are presented in [Althoff, 2010,Pek, 2020]. A described in the previous section Figure 2.7, 2.8 and 2.9 show the results with cooperative motion planning from [Hartmann and Watzenig, 2019a,Schratter et al., 2019,Hartmann et al., 2018b,Hartmann and Watzenig, 2019b]. Figure 2.2a shows the funnel capturing the green trajectory. Funnel set prediction is in this thesis a consecutive state set prediction $\mathcal{X}_{k_{ri}}^{j}$. Figure 2.2b shows the result of reachability Analysis with zonotypes programmed with the algorithm from [Girard et al., 2006]. The belief sets in [Hartmann and Watzenig, 2019a,Schratter et al., 2019,Hartmann et al., 2018b,Hartmann and Watzenig, 2019b] are developed with point-wise approaches or parallelograms. In [Hartmann and Watzenig, 2019a] the $v_{\max}, a_{\max}, e_{\max}$ (maximal velocity, acceleration and jerk) was predicted by Gaussian Processes for a specific future time horizon. The advantage of this approach is that the maximal parameters' dynamics are less dynamic than predicting the real movement. The publication used this approach for developing collision avoidance models [Schratter et al., 2019] for autonomous vehicles. There exist other libraries for reachability analysis [Bogomolov et al., 2019] and the application in automotive scenarios [Althoff et al., 2017]. In Figure 2.3 we show a grid-based approach for reachability analysis. If we have a discrete grid and an agent can walk in $\mathcal{U} = \{"N","W","S","U"\}$ direction, we can compute the reachable sets, for example, for MDPs and grid-based agents.



(a) Funnel in xy-space over time t.

(b) Reachability Analysis with zonotypes

Figure 2.2.: Funnel and Reachability Analysis [Hartmann, 2021]

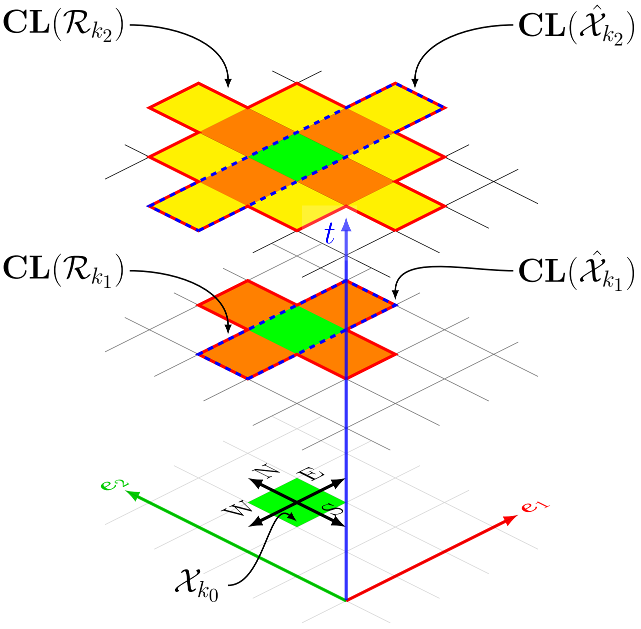

$\mathbf{CL}(\mathcal{R}_{k_2})$
$\mathbf{CL}(\hat{\mathcal{X}}_{k_2})$
t
$\mathbf{CL}(\mathcal{R}_{k_1})$
$\mathbf{CL}(\hat{\mathcal{X}}_{k_1})$
e_2
N
E
W
S
e_1
$\mathcal{X}_{k_0}$

Figure 2.3.: Grid based reachability analysis: Pedestrian starting on $\mathcal{X}_{k_0}$ and having options to walk in $\{N, E, S, W\}$

2.2. Machine Learning

Subsection 2.2.1 discusses current approaches for statistical movement prediction of road users (also with some approaches using reachability analysis), reinforcement learning (section 2.2.2) and causal inference (section 2.2.3).

2.2.1. Movement prediction of road users

This section has a focus on statistical movement prediction algorithms. However, also reachability analysis (section 2.1) from the previous section is also applied for movement prediction of pedestrians. Many good textbooks introduce the basics of machine learning [Friedman et al., 2001, Bishop, 2006, Murphy, 2012, Kochenderfer, 2015, Goodfellow et al., 2016, Peters et al., 2017]. Machine Learning deals with inference techniques for data. Typically, a dataset $\boldsymbol{X}$ is prior knowledge of a problem obtained from an experiment. In Supervised Learning, a labeling vector $\boldsymbol{y}$, where the elements of $\boldsymbol{X}$ are labeled. The task is to find a function $\hat{\mathbf{f}}$ which should be similar to the objective function $\mathbf{f} : \boldsymbol{X} \rightarrow \boldsymbol{y}$. With the introduction of a loss function, the problem can be cast to a maximation problem [Friedman et al., 2001]. In Unsupervised Learning, there is no labeling vector available, and only the dataset is available. The data inference technique learns the data structure and tries to recognize patterns in the data. There is a common assumption that the data lies in a manifold [Dollár et al., 2007, Van Der Maaten et al., 2009]. An example of unsupervised learning could be clustering techniques and manifold learning techniques. In Semisupervised Learning, only a subset of data $\boldsymbol{X}$ has labels. Reinforcement Learning is a particular case where an agent interacts with the environment compare section 2.2.2. In Deep Learning models [Goodfellow et al., 2016], neurons are interconnected in graphs to perform complex computations. There currently exist many different libraries and frameworks like TensorFlow, PyTorch, Caffe, Keras, and others in Python, C++/C, Julia [Pedregosa et al., 2011, Müller and Guido, 2016, Goodfellow et al., 2016]. Inference in Machine Learning gets more difficult in higher than lower dimensions (curse of dimensionality).

Comparison of the vehicle and the pedestrian as dynamic systems.

In this thesis, we focus on two dynamic systems, a pedestrian (with state $\boldsymbol{x}^h(t)$ and label h for human) and a vehicle (state $\boldsymbol{x}^v(t)$ and with superscript v for autonomous vehicle). There are some differences between vehicles and pedestrians:

- Physical mass: A single physical mass can represent a vehicle and pedestrian. It might be possible to represent a single particle in a vector space. The modeling might lead to substantial simplifications. Stiffness at some parts of the vehicle might not be modeled when the vehicle is modeled by particle. The pedestrian has multiple interconnected masses in the form of limbs connected with a topological structure (compare figure 2.4). The modeling might lead to loss of information (what action does the pedestrian do, jumping, clapping with the hands).
- System-dynamics: Several deterministic models exist for the vehicle as a dynamic system. [Koschi and Althoff, 2017b]. A car-like vehicle is normally a nonholonomic system (there exist holonomic vehicles). For pedestrians exists some models, but there are are often very strong simplifications (stochastic system [De Nicolao et al., 2007a], linear system [Liu et al., 2017]).
- Measurements/Sensing and Perception: Measurements by sensors for the vehicle and sense organs for the pedestrian (compare figure 2.5)
- Intelligence: Computational intelligence in the vehicle and intellectual and emotional intelligence by the pedestrian
- Actuation: Actuators on in the drivetrain and steering for the vehicle and muscles for the pedestrian (compare figure 2.5).
- Environment: The roads in urban environments usually have a topological structure where cars can drive. They often have a unique structure and in each country exist special transportation rules. Physical obstacles might influence the behavior of vehicles and pedestrians, which might lead to collisions in fatal situations, (opposing) forces, or cognitive influences on the behavior from the perception of these objects.

Systematical classification of statistical movement prediction

This section focuses on pedestrians (some methods are also adaptable to technical systems). In technical systems (like vehicles), the state over time is often constrained by physical laws and energy. Humans also have cognitive constraints, which are affecting consciousness and free will (compare section B.1). The first step of movement prediction is often taking measurements of a pedestrian. In a movement prediction problem as a machine learning problem, the following information might be relevant:

- Discrete time-space $\mathbb{T}$: Historical timesteps $\{k_r, \ldots, k_i\}$ and future timesteps $\{k_{i+1}, \ldots, k_j\}$. For the timesteps hold $k_r < k_i < k_j$, $k_r, k_i, k_j \in \mathbb{T}$ with the time label set $\mathcal{I} = \{r, \ldots, i, \ldots, j\}$
- Measurements $\mathcal{Y}_{r \to j} = \{\mathbf{y}_{k_q}\}_{q=r}^{j}$: Information from the human body and environment in form of measurements. The whole body might be represented as a particle or as a topological system (figure 2.4).
- Additional label sets: There exist different possibilities to use additional label sets for machine learning. It might be necessary to label the environment of the pedestrian (semantic labeling) or use variants of (semi-)supervised learning techniques.
- States $\mathcal{K}_{r \to j} = \{\mathbf{x}_{k_q}\}_{q=r}^{j}$: Normally the states $\mathbf{x}_{k_i}$ are not directly known and must estimated by measurements
- Principles of the system: A human has physical components constrained by the energy-flow (body movements) or information-theoretical components as the cognitive system (brain, nervous system).
- Propagation of states (Prediction or Guessing): Historical states are mathematically mapped by an operator (matrix, tensor, or something else) to the future. For optimal prediction, it is advantageous to have a valid system model. All physical, information-theoretical mechanisms and causal structures are modeled under certain quality criteria, which might ensure the quality of the prediction. A problem is a complexity of modeling the human and the (urban) environment as a mathematical system.
- Type of prediction algorithm: Is the machine learning problem described as a supervised, semi-supervised, unsupervised learning, or reinforcement learning problem. The supervised, semi-supervised and

unsupervised learning tasks are different approaches depending on the availability of input data and labels for a machine learning task. Another question is whether the prediction is made in a classical machine learning or Deep Learning framework. Probabilistic formulations and graphic descriptions in classical machine learning frameworks are very common. Deep Learning is a method inspired by biology as interconnected neurons perform learning tasks.

- Causality: Another problem is the underlying structure of physical and cognitive mechanisms interconnected with each other. It might be possible that a person changes his behavior/policy by interacting with other persons or changing his intention.
- Modelling of the human as a system: When the human is represented as a particle or a topological system, relevant information might not be available for the machine learning algorithm. This simplification might cause a lack of knowledge, which might cause an increase in the aleatoric (non-reducible) uncertainty.
- State-space description: The mathematical description of the human as a system and the environment might change prediction results. Is the environment represented as a graph, manifold, or simply as a vector space.
- Kind of analysis: It is also a difference if a single particle or a set is propagated over time. In reachability analysis, the system is analyzed under a worst-case scenario instead of predicting the future movements.

A mathematical model for information propagation of spatial information will predict the future trajectory, set of particles, or funnel. The authors [Lefèvre et al., 2014,Hirakawa et al., 2018,Rudenko et al., 2020,Brunetti et al., 2018] give a broad overview of movement prediction of vehicles or pedestrians. The paper in [Lefèvre et al., 2014] classifies three different classes: Physically inspired like the Bayesian filters, Maneuver-based, and interaction-based approaches. There might be a knowledge database for measurements $\mathcal{Y}_{r\rightarrow j}$ or states $\mathcal{K}_{r\rightarrow j}$ to compute future belief states $\mathbf{b}_{k_j}$. The Bayesian filters use only the current measurement in a prediction and innovation cycle to predict under certain conditions on the system model and noise (compare figure 2.6a). Maneuver-based approaches would not take only one measurement, but rather a set of measurements $\mathcal{Y}^{h_t}_{r\rightarrow j} := \{\boldsymbol{y}_{k_q}\}^{i}_{q=r}\ k_r < k_i$

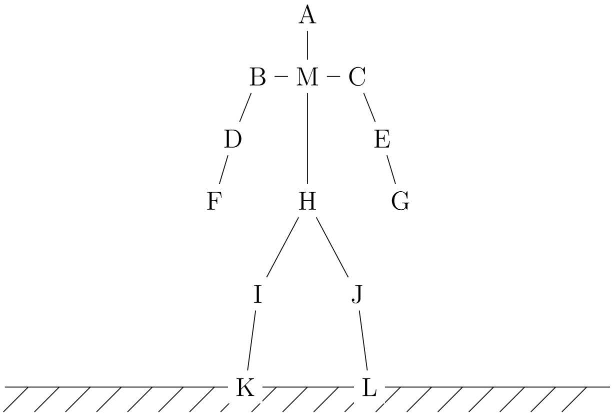


Figure 2.4.: A human skeleton can be described as a topological space. The labeling of the limb positions is chosen arbitrarily and can be extended. A: head, B: left shoulder, C: right shoulder, D: left elbow, E: right elbow, F: left hand, G: right hand, H: Hip, I: left knee, J: right knee, K: left foot, L: right foot, M: neck

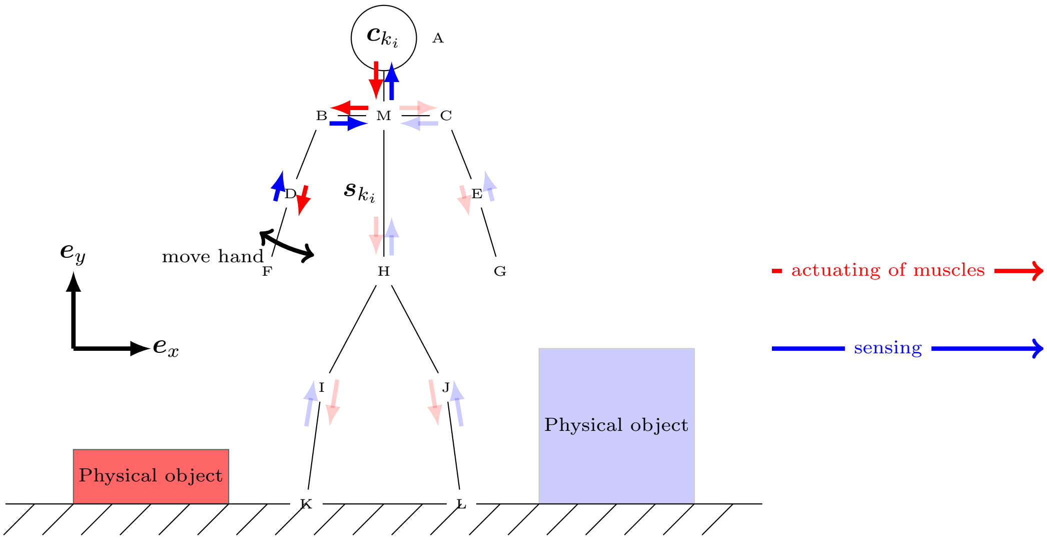
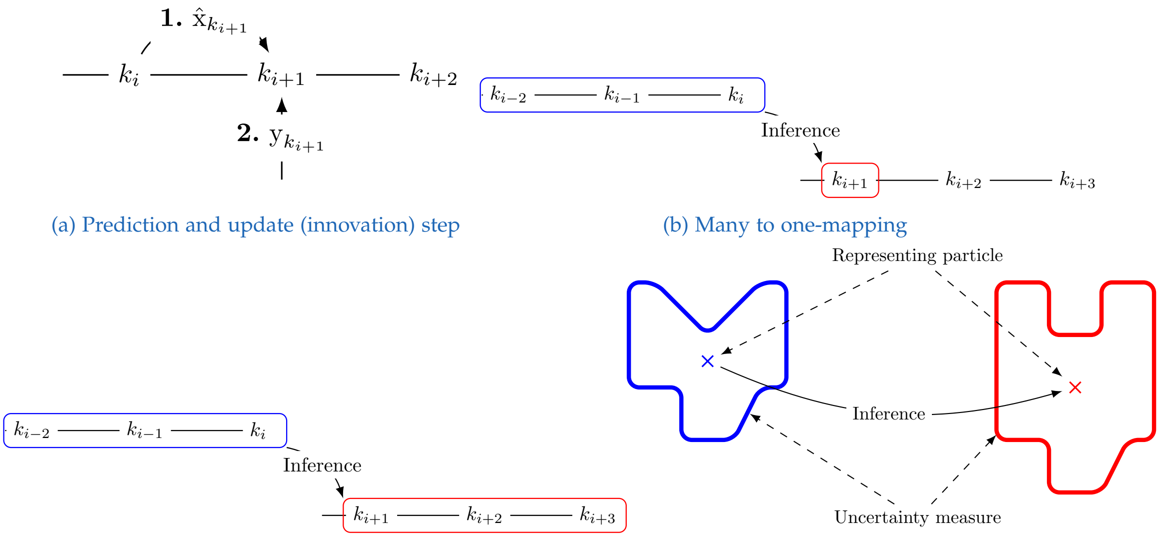


Figure 2.5.: Human with the cognitive state c_{k_i} and body state s_{k_i}. The body state s_{k_i} consists of multiple positions in a topological structure and is defined in a Euclidean vector space. A complex communication system exists between the brain and the limbs in the human body, with control elements (actuating the muscles) and measurements (sensing). Physical laws and objects constrain the human body's movements in the environment. However, cognitive and medical conditions constrain the dynamics of decision-making.



(a) Prediction and update (innovation) step

(b) Many to one-mapping

(c) Many to many mapping

(d) Past blue set to the future red set

Figure 2.6.: Types to predict the future entities

of a human (label h) with label h_t and extrapolate the trajectory (compare figure 2.6b and figure 2.6c). In the many to one mapping from figure 2.6b, we consider information (states or measurements) from a collection of historical time steps to predict one single state in the future. In the many-to-many mapping, we predict multiple future states. In interaction-based approaches, it would consider measurements for m different pedestrians $\mathcal{Y}_{r\to i}^{[1,m]} := \{\mathcal{Y}_{r\to i}^{h_t}\}_{t=1}^{m}$ and predict their movements. There are also differences in the type of inference (classical machine learning, deep learning, causal inference). The set-based prediction in figure 2.6d stands for not predicting a single particle but rather a whole set (reachability analyses). The set might be convex- or non-convex.

List of selected relevant work

[Ziebart, 2010] incorporates causal inference techniques in movement prediction. Instead of describing the measurements in an euclidean space, the approach in [Govea, 2010] uses adaptive or static [Ikeda et al., 2013, Mohamed et al., 2020] graphs. [Govea, 2010] uses incremental learning to predict the movements with Hidden Markov Models. From the driver's perspective, images from onboard cameras on vehicles are processed in [Keller and Gavrila, 2013] to predict pedestrians' future movements. [Ziebart, 2010, Kitani et al., 2012, Fahad et al., 2018, Zhang et al., 2018] use an inverse reinforcement learning approach by learning from observations and estimating a reward function. [Karasev et al., 2016] models a pedestrian with rational behavior and a Markov process with a stochastic policy to reach a hidden target. The authors in [De Nicolao et al., 2007b] describe how to generate pedestrians' synthetic trajectories with a stochastic process. The authors in [Ellis et al., 2009] use Gaussian processes to compute vector fields and Monte-Carlo simulation to predict the particle movements. [Mohamed et al., 2020] uses convolutional neural networks (CNNs). [Alahi et al., 2016, Crivellari and Beinat, 2020, Manh and Alaghband, 2018, Xue et al., 2018, Hug et al., 2018] use Long short-term memory neural networks (LSTMs) to predict movements from datasets. The methods in [Xue et al., 2018, Manh and Alaghband, 2018, Xue et al., 2018] offer a complex hierarchical structure (person, social, scene) of several LSTMs to predict the movements. [Alahi et al., 2016] uses a set of interconnected LSTMs for different pedestrians, whereas [Xue et al., 2018] uses another convolutional neural network (CNN) for feature detection on the scene scale. In [Hug et al., 2018] uses a particle-based approach. [Hug et al., 2018, Ziebart, 2010, Kitani et al., 2012] are using beneficial multi-modal movement predictions (several predictions for one subject). The approach [Crivellari and Beinat, 2020] uses the movement in a large-scale environment (kilometers). Figure 2.1 shows a comparison of different movement predictions. A common approach is to use multimodal movement prediction algorithms because the movement prediction of the pedestrian is not certain in advance. This motivates us to generalize multimodal approaches to set-based approaches. It compares the movement prediction algorithms in different categories. Is a static topological space or euclidean space used instead? Does it use a deep learning approach? Causal inference is trying to decode the causal structure in data. Another example to represent movements is by a single euclidean system and a stochastic process, compare figure 3.25a, which shows trajectories from a stochastic process [De Nicolao et al., 2007b]:

$$x(t+T_S) = x(t) + T_S \cdot v_x(t) \tag{2.31}$$

$$v_x(t+T_S) = v_x(t) + T_S \cdot \omega_x(t) \tag{2.32}$$

$$\omega_x(\cdot) \sim \text{WGN}(0, \sigma_x^2) \tag{2.33}$$

$$y(t+T_S) = y(t) + T_S \cdot v_y(t) \tag{2.34}$$

$$v_y(t+T_S) = v_y(t) + T_S \cdot \omega_y(t) \tag{2.35}$$

$$\omega_y(\cdot) \sim \text{WGN}(0, \sigma_y^2(t)) \tag{2.36}$$

with WGN for White Gaussian noise and initial state:

$$\begin{bmatrix} x(0) \\ v_x(0) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_x \\ m_{v_x} \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & \sigma_{v_x}^2 \end{bmatrix} \right) \tag{2.37}$$

$$\begin{bmatrix} y(0) \\ v_y(0) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_y \\ 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \right) \tag{2.38}$$

Table 2.1.: Comparison of movement prediction

Paper	Deep Learning	Causal Inference	(Monte-Carlo) Sampling	Multimodal state prediction	Propagation of sets	Method
Keller. [Keller and Gavrila, 2013]	✗	✗	✗	✗	✗	Gaussian process/Kalman filter
Karasev. [Karasev et al., 2016]	✗	✗	✗	✓	✗	Markov decision process/Rao-Blackwellized filter
Nicolao. [De Nicolao et al., 2007b]	✗	✗	✓	✗	✗	Stochastic dynamic model
Moh. [Mohamed et al., 2020]	✓	✗	✗	✗	✗	Convolutional neural network
Alahi. [Alahi et al., 2016]	✓	✗	✗	✗	✗	LSTM
Huynh. [Manh and Alaghband, 2018]	✓	✗	✗	✗	✗	LSTM
Xue. [Xue et al., 2018]	✓	✗	✗	✗	✗	LSTM
Hug. [Hug et al., 2018]	✓	✗	✗	✗	✗	LSTM
Flohr. [Flohr et al., 2015]	✗	✗	✓	✗	✗	Probabilistic framework head localization and orientation
Bonnin. [Bonnin et al., 2014]	✗	✗	✗	✗	✗	Heuristics
Schulz. [Schulz and Stiefelhagen, 2015]	✗	✗	✗	✗	✗	Conditional random fields
Neogi. [Neogi et al., 2017]	✗	✗	✗	✗	✗	Conditional random fields
Kooij. [Kooij et al., 2014]	✗	✗	✗	✓	✗	Switching dynamic systems

Gold. [Goldhammer et al., 2013]	✗	✗	✗	✗	✗	Piecewise linear model and a sigmoid model
Voelz. [Völz et al., 2016]	✗	✗	✗	✗	✗	Linear Quantile Regression (LQR) and the Quantile Regression Forests
Rehder. [Rehder and Kloeden, 2015]	✓	✗	✗	✗	✗	RNN
Rehder. [Rehder et al., 2018]	✓	✗	✗	✗	✗	RNN
Vasquez. [Vasquez, 2016]	✗	✗	✓	✓	✓	Fast Marching Method
Vasishta. [Vasishta et al., 2017]	✗	✗	✗	✗	✗	Multiple potential fields
Wu. [Wu et al., 2018]	✗	✗	✗	✓	✗	Markov-Chain on Grids
Liu. [Liu et al., 2017]	✗	✗	✓	✓	✓	Reachability Analysis
Hartmann. [Hartmann et al., 2018b]	✗	✗	✓	✓	✓	Data-based-reachability analysis
Ellis. [Ellis et al., 2009]	✗	✗	✓	✓	✗	Gaussian Process/Monte Carlo S.
Vasquez. [Vasquez, 2010a]	✗	✗	✗	✗	✗	Incremental Topological Hidden Markov Models
Ziebarth. [Ziebart, 2010]	✗	✓	✗	✓	✗	Incremental Topological Hidden Markov Models
Kitani. [Kitani et al., 2012]	✗	✓	✗	✓	✗	Incremental Topological Hidden Markov Models

The statistical movement prediction assumes that there exist statistical patterns in the data. The prediction model has to learn it. An adequate dataset is necessary to perform predictions based on historical data. There exist different datasets for vehicles [Zhan et al., 2019a, Krajewski et al., 2018, Huang et al., 2018] and for pedestrians [Robicquet et al., 2016b]. [Bock et al., 2019, Zhan et al., 2019b] used drones to capture the traffic flow from the bird-view perspective. The papers in [Rasouli et al., 2017,Geiger et al., 2013,Huang et al., 2019] present traffic datasets from the vehicle perspective. This thesis uses the aerial drone dataset [Robicquet et al., 2016a] to observe pedestrians. The authors in [Pellegrini et al., 2009] published another dataset of pedestrian movements. Figure 2.7, 2.8 and 2.9 shows own contributions for cooperative motion planning and adaptive reachability analysis from [Hartmann and Watzenig, 2019a,Schratter et al., 2019]. Further contributions in publications and visualizations are in [Hartmann et al., 2018b,Hartmann and Watzenig, 2019b]. Past approaches used reachability analysis to only make worst-case scenarios, predictions only on vehicles (which are easier to predict than pedestrians), or based on traffic rules [Althoff, 2010, Liu et al., 2017]. We proposed machine learning techniques to make adaptive reachable sets based on the observations of a current situation. The deficit of the approaches is that there is a risk dilemma that false predictions could lead to fatal consequences. On the other side, very conservative movement predictions would lead to inefficient motion planning because the autonomous would have to halt every time. Figure 2.7 shows optimal motion planning with mixed-integer linear programming for the vehicle driving in the positive x-direction. The colorful rectangles (blue) represent different timesteps. Model A is used to find reachable sets for the pedestrian (colorful polygons), with acceleration- and velocity-constrained models for the pedestrian's reachable sets. These assumptions lead to reachable sets, which intersect with the street (compare red-filled polygon on the street). Figure 2.8 shows optimal motion planning with mixed-integer linear

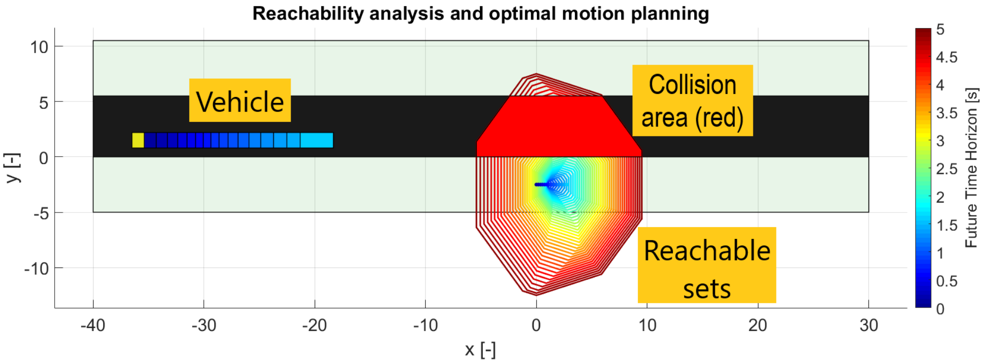


Figure 2.7.: Motion planning with Reachability Analysis [Hartmann and Watzenig, 2019a]

programming [Hartmann and Watzenig, 2019a] and adaptive reachable sets for the pedestrian. Figure 2.9 shows other collision avoidance models also with adaptive reachability analysis.

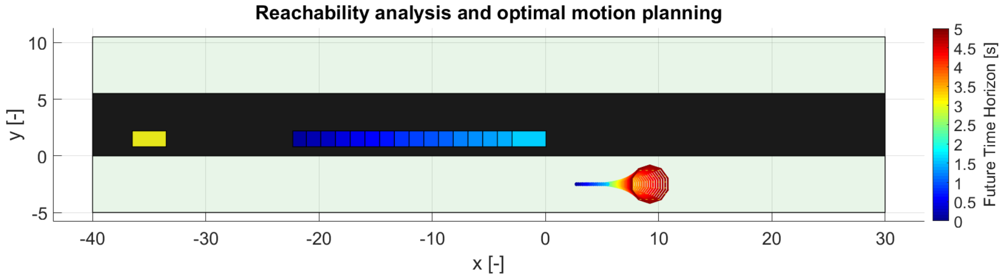


Figure 2.8.: Motion planning with Belief Sets [Hartmann and Watzenig, 2019a]

2.2.2. Reinforcement Learning

Reinforcement Learning is an approach where an agent interacts with the environment and gets feedback by measurements. The agent learns to adapt to experience and learn to make (near-) optimal or adequate decisions (optimizing a cumulative reward). An agent is also making decisions in an uncertain environment. [Kochenderfer, 2015,Jurafsky and Martin, 2020] gives a broad introduction to the concepts of decision making under uncertainty. Figure 2.10 shows the composition of Partially Observable Markov Decision Process (POMDP) from a first-order Markov-chain, Hidden Markov Model (HMM), and a Markov Decision Process (MDP). Figure 2.10a shows a Markov-chain with hidden states. For the first order Markov-chain (figure 2.10a) the Markov property is fullfilled for all $i \in \mathbb{N}, k_r < k_i,\ k_r, k_i \in \mathbb{T}$:

$$P(\mathbf{x}_{k_i} | \underbrace{\mathbf{x}_{k_{i-1}}, \mathbf{x}_{k_{i-2}}, \ldots, \mathbf{x}_{k_r}}_{\mathcal{K}_{r \to i-1}}) = P(\mathbf{x}_{k_i} | \mathbf{x}_{k_{i-1}}) \tag{2.39}$$

With Markov-chains, it is possible to model cyclic processes dependent on time k_i. The First-order Markov-chain means that the Markov property holds for all states. $\mathbf{x}_{k_i}$ is causally affecting only $\mathbf{x}_{k_{i+1}}$ for all $k_i \in \mathbb{T}$. HMMs (figure 2.10b) are models where the states $\mathbf{x}_{k_i}$ are hidden (grey node) and

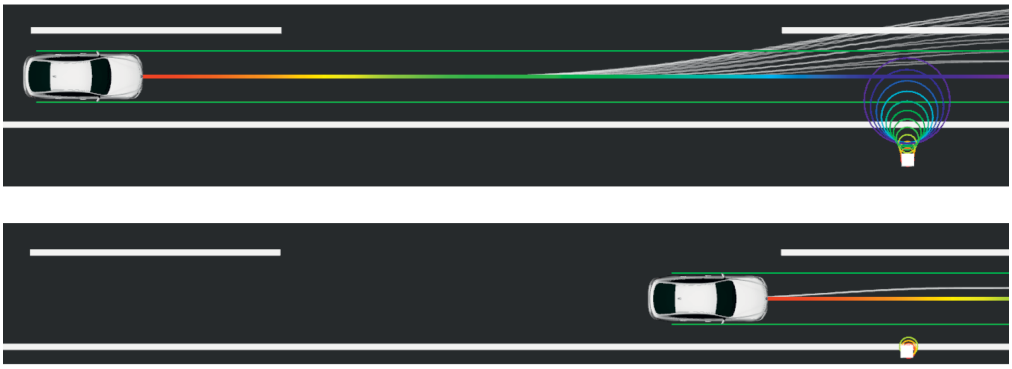


Figure 2.9.: Collision avoidance with belief sets [Schratter et al., 2019]

only observations $\mathbf{y}_{k_i}$ and belief states $\mathbf{b}_{k_i}$ (which should be the estimated state $\hat{\mathbf{x}}_{k_i}$) are available. The output independence lead to fact that only the current state is affecting the current observation for $k_r < k_i < k_j$:

$$\mathrm{P}(\mathbf{y}_{k_i}|\underbrace{\mathbf{x}_{k_r},\dots,\mathbf{x}_{k_i},\dots,\mathbf{x}_{k_j}}_{\mathcal{K}_{r\to j}},\underbrace{\mathbf{y}_{k_r},\dots,\mathbf{y}_{k_i},\dots,\mathbf{y}_{k_j}}_{\mathcal{Y}_{r\to j}}) = \mathrm{P}(\mathbf{y}_{k_i}|\mathbf{x}_{k_i}) \tag{2.40}$$

The same for the belief states $\boldsymbol{b}_{k_i}$ and $k_r < k_i < k_j$:

$$\mathrm{P}(\mathbf{b}_{k_i}|\mathcal{K}_{r\to j},\mathcal{Y}_{r\to j},\underbrace{\mathbf{b}_{k_r},\dots,\mathbf{b}_{k_i},\dots,\mathbf{b}_{k_j}}_{\mathcal{B}_{r\to j}}) = \mathrm{P}(\mathbf{b}_{k_i}|\mathbf{y}_{k_i}) \tag{2.41}$$

HMMs are good to capture and predict temporal sequences. We could compute the measurement sequence:

$$\mathrm{P}(\mathcal{Y}_{r\to j}|\mathcal{K}_{r\to j}) = \prod_{q=r}^{j} \mathrm{P}(\mathbf{y}_{k_q}|\mathbf{x}_{k_q}) \tag{2.42}$$

MDPs are models where the agent is trying to maximize the expected reward over time $\boldsymbol{r}_{k_i}$ by choosing an action at each timestep $\boldsymbol{a}_{k_i}$ where all the states are known (figure 2.10c). POMDP (figure 2.10d)is an extension to MDPs for computing optimal actions in presence of hidden and unobserved states. There are numerous variants of these models and combinations with

Deep Learning Models. Deep Learning could be used in Reinforcement Learning for the learning part, for example, in pattern recognition, e.g., recognizing objects on cameras or approximating some functions. In Inverse Reinforcement Learning, an algorithm learns rewards by examples. These variants are used for movement prediction in [Kitani et al., 2012, Ziebart et al., 2009, Ziebart, 2010]. There exist different variants of reinforcement models, but very common is that the expected reward function is maximized by an agent in a sequential problem solved by dynamic programming. For a finite horizon problem the expected rewards (or return) is:

$$\sum_{k=0}^{n-1} \gamma^k r_k, \quad 0 \leq \gamma \leq 1 \tag{2.43}$$

There exist offline and online methods depending on when the optimization problem is solved. The transition function $\mathrm{T}(\boldsymbol{s}_{k+1}|\boldsymbol{s}_k, \boldsymbol{a}_k)$ is describing the state dynamics based on the actions. Figure 2.11a shows the initial reward and the other pictures of figure 2.11 show a implementation of value iteration. During the execution of the value iteration, the algorithm is learning the value function $\mathrm{V}(s)$, which describes the expected utility for being in a state. An optimal policy π^* is a policy that maximizes the expected utility:

$$\pi^*(\boldsymbol{s}) = \arg\max_{\pi} \mathrm{V}^{\pi}(\boldsymbol{s}) \tag{2.44}$$

One could solve the problem with classical dynamic programming. In reality, there exists often uncertainty to the dynamics and rewards. The agent learns the unknown transition- and reward function with experience (reinforcement learning).

$$\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}_k \in \mathcal{A}} \mathbb{E} \sum_{k=1}^{N_T} P(\boldsymbol{r}_{k+1}|\boldsymbol{s}_k, \boldsymbol{a}_k) \tag{2.45}$$

Therefore the Q-function $\mathrm{Q}(\boldsymbol{s}, \boldsymbol{a})$ is describing the expected value for the agent with executing an action $\boldsymbol{a}$. The formal definition of a Markov Decision Process (MDP) is:

Definition 2.2.1. A MDP is a tuple of
$\mathfrak{M} \equiv (\mathcal{S}, \mathcal{A}, \mathrm{P}(\boldsymbol{s}_{k+1}|\boldsymbol{s}_k, \boldsymbol{a}_k), \mathrm{P}(\boldsymbol{r}_{k+1}|\boldsymbol{s}_k, \boldsymbol{a}_k), \gamma, \pi)$

- $\mathcal{S}$ is a set of possible states
- $\mathcal{A}$ is set of possible actions
- $P(s_{k+1}|s_k, a_k) : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ probabilistic transition function
- $P(r_{k+1}|s_k, a_k) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ probabilistic reward function, shorthand for $R(s_k, a_k)$
- $\gamma \in [0, 1]$: discount factor
- π: policy shorthand for $P(a_k|s_k)$

The utility function $U_k^{\pi}(s_k) : \mathcal{S} \rightarrow \mathbb{R}$ can be computed for the policy evaluation [Braziunas, 2003]:

$$U_k^{\pi}(s_k) = R(s_k, a_k) + \gamma \sum_{s_k, s_{k+1} \in \mathcal{S}} P(s_{k+1}|s_k, \pi) U^{\pi}(s_{k+1}) \tag{2.46}$$

With the new utility function the policy can be improved [Braziunas, 2003], [Kochenderfer, 2015]:

$$Q_{k+1} = R(s_k, a_k) + \gamma \sum_{s_{k+1} \in \mathcal{S}} P(s_{k+1}|s_k, a_k) U^{\pi_k}(s_{k+1}) \tag{2.47}$$

$$\pi_{k+1}(s_k) = \arg\max_{a_k \in \mathcal{A}} Q_{k+1}(s_k, a_k) \ \forall \ s_k \in \mathcal{S} \tag{2.48}$$

The policy improvement and utility function computation can be computed sequentially. Another approach is to compute the next action with the maximum expected utility principle ([Kochenderfer, 2015]):

$$a_k^* = \arg\max_{a_k \in \mathcal{A}} \mathbb{E} \sum_{s_{k+1} \in \mathcal{S}} P(s_{k+1}|a_k, o_k) U(s_{k+1}) \tag{2.49}$$

One can also compute the most probable policy on state s_k with:

$$\pi^*(s_k) = \arg\max_{\pi \in \Pi} (R(s_k, \pi(s_k)) + \gamma \cdot \sum_{s' \in \mathcal{S}} P(s_{k+1}|s_k, \pi(s_k)) \cdot U_{k-1}^{\pi}(s')) \tag{2.50}$$

A classical POMDP with Markov-property is defined as:

Definition 2.2.2. A POMDP is a tuple of
$\mathfrak{P} \equiv (\mathcal{S}, \mathcal{A}, \mathcal{B}, \Omega, P(s_{k+1}|a_k, b_k), P(r_{k+1}|s_k, a_k), P(b_k|o_k), \gamma)$

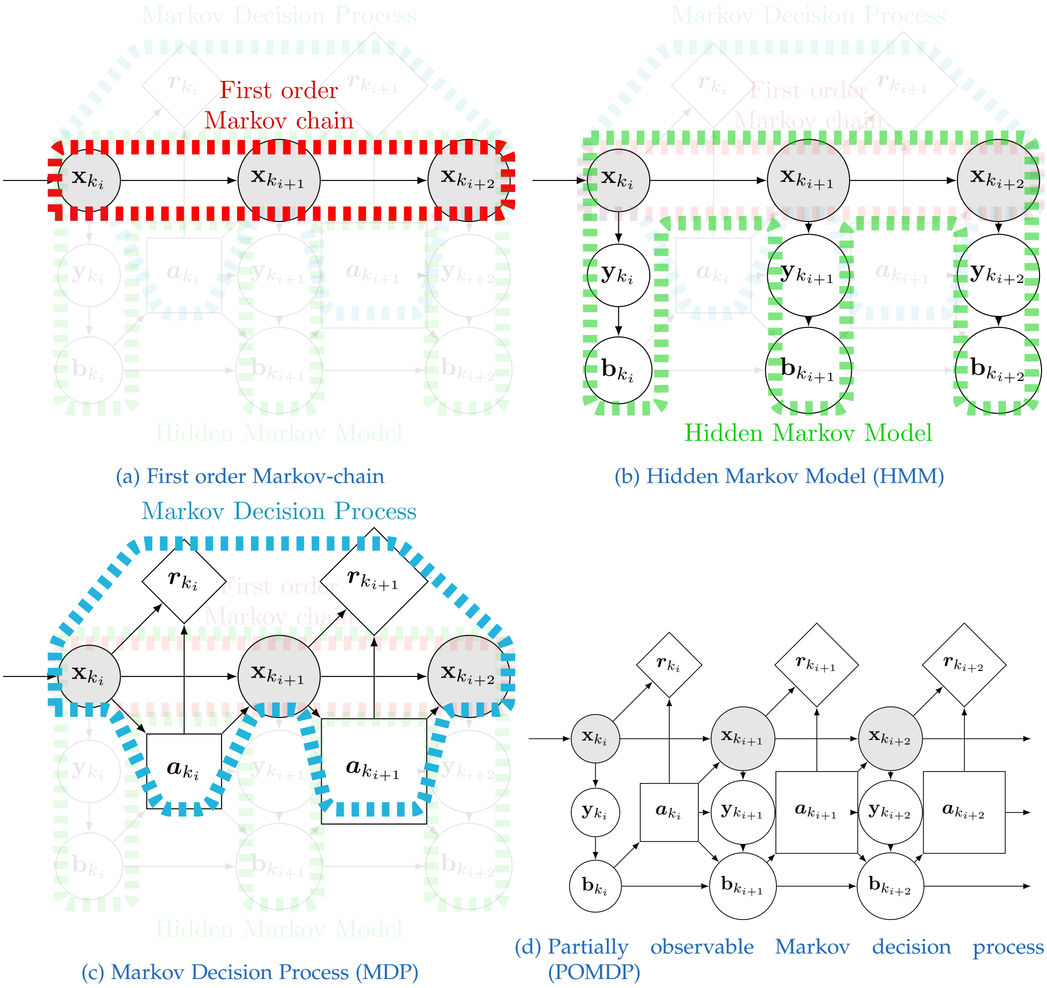
- $\mathcal{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{N_\mathcal{S}}\}$ is set of states
- $\mathcal{A} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{N_\mathcal{A}}\}$ is set of actions
- $\mathcal{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{N_\mathcal{B}}\}$ is set of belief-states
- $\Omega = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{N_\Omega}\}$ is a set of observations
- $P(\boldsymbol{s}_{k+1}|\boldsymbol{a}_k, \boldsymbol{b}_k) : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ transition function
- $P(\boldsymbol{r}_{k+1}|\boldsymbol{s}_k, \boldsymbol{a}_k) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ reward function
- $P(\boldsymbol{b}_k|\boldsymbol{o}_k) : \mathcal{S} \times \mathcal{O} \rightarrow \mathbb{R}$ conditional probabilities from observation to belief state
- $\gamma \in [0, 1]$ discount factor

2.2.3. Causal Inference

In experiments, humans can learn from data to model the experiment and predict the outcome of future events. We could model the experiment with concepts from probability theory. An experiment is formed by a tuple $(\Omega, \mathcal{F}, P)$ probability space could model the sample space Ω, the event space $\mathcal{F}$ and the probability function P. In an experiment with a fair dice and six numbers, someone could ask the probability of getting even numbers ($|2, 4, 6| = 3$). A frequentist counts the total amount of events in comparison to the total amount of samples ($P(\text{"even number"}) = 3/6$). There exist experiments where it is not always possible to get more data from an experiment in practice. It is impossible to generate more data, and one could formulate a hypothesis of the experiment with a Bayesian viewpoint. From the Bayesian viewpoint, we would formulate an initial/ prior belief of the outcome of an experiment. With the Bayes rule, one could compute the posterior belief:

$$P(B|A) = \frac{P(A|B)P(A)}{P(A,B)} \tag{2.51}$$

We can visualize the relationship between A and B by a simple graph. The Bayes rule could be used in many applications. In sequential experiments with Markov property, a Bayesian filter (Kalman- and Particle filter) uses new evidence in measurements/observations, leading to new state predictions. In causal inference settings, the probabilities might be changed by external interventions. This section gives an overview of some causal-inference



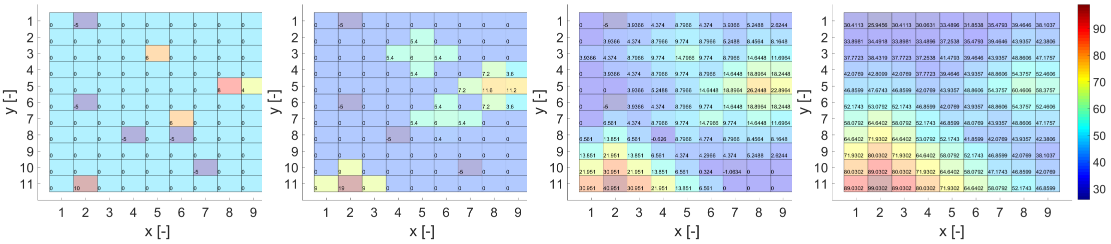
(a) First order Markov-chain

(b) Hidden Markov Model (HMM)

(c) Markov Decision Process (MDP)

(d) Partially observable Markov decision process (POMDP)

Figure 2.10.: First order Markov-chain, Hidden Markov Model (HMM), Markov Decision Process (MDP), Partially observable Markov decision process (POMDP)



(a) Initial reward map (b) Selected iteration (c) Selected iteration (d) Result

Figure 2.11.: The pictures show the initial reward map and value iteration on a classical regular grid in a simple MATLAB® simulation.

strategies to model the interaction between the vehicle and pedestrian. The literature in [Pearl, 2003, Pearl et al., 2009, Pearl, 2009, Imbens and Rubin, 2015, Pearl et al., 2016, Peters et al., 2017, Pearl and Mackenzie, 2018] gives an excellent introduction to the topic of causal inference. In causal inference, scientists develop models which could find causal relationships between some physical mechanisms (structural learning). The idea is to get the causal relationship between data sources. The relationship between the cause and an effect between some data entities offers more insight into a problem than correlation. It could also help to answer why an event is happening. Causal inference is handling settings with interventions and counterfactuals and extending machine learning with classical probability theory. Prerequisites to understanding causal inference models are Bayesian networks and probability theory. Structural causal models are simple models for describing causal relationships [Peters et al., 2017]:

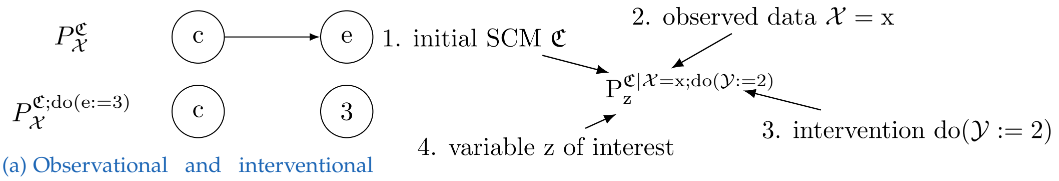
Definition 2.2.3. A structural causal model $\mathfrak{C}$ with the variables: $X = \{C, E\}$ with observational graph $\mathcal{C} \rightarrow \mathcal{E}$ consists of two assignments,

$$C := N_c \tag{2.52}$$

$$E := f(C, N_e) \tag{2.53}$$

where N_e is independent from N_c with the shorthand $N_e \perp\!\!\!\perp N_c$.

Structural causal models can change their structure due to an intervention and the resulting do-Operator. It is also possible to compute counterfactuals


$P_{\mathcal{X}}^{\mathfrak{C}}$
c
e
$P_{\mathcal{X}}^{\mathfrak{C};do(e:=3)}$
c
3
1. initial SCM $\mathfrak{C}$
2. observed data $\mathcal{X} = x$
$P_z^{\mathfrak{C}|\mathcal{X}=x;do(\mathcal{Y}:=2)}$
3. intervention do($\mathcal{Y} := 2$)
4. variable z of interest

(a) Observational and interventional graph

(b) Causal description

Figure 2.12.: Causal graphs and description of structural causal models [Peters et al., 2017]

after an intervention. We distinguish between observational, conditional, interventional, and counterfactual distributions. The causal graph and counterfactuals are essential features describing the causal model. Figure 2.12a visualizes the causal relationship between a cause c and the effect e. The edge disappears when an intervention influences the effect e with the do-Operator. When the do-operator affects the cause c, the edge between c and e is further existent, and the causal relationship between c and e. The following example is adapted from [Peters et al., 2017] Figure 2.12b shows the declaration of the probability from a causal model $\mathfrak{C}$ conditioned on the observed data x for $\mathcal{X}$ with an intervention on $\mathcal{Y}$ set to 2. Variable of interest is z. Figure 2.13 shows an example from [Neal, 2020] from a population described as a circle and mathematical as a set, where there exist two subpopulation. The set is describing an experiment with treatment $T = a$ and $T = b$ (e.g. taking the medicine $T = a$ or not taking it $T = b$). It also shows how the conditioning and the intervention differ from each other. In conditioning, one looks at the dataset from the experiment. It depends on how much of each population exists. In intervention, the whole population is changed by external interventions. The dependence on time might also differ from cyclic and non-cyclic models. In cyclic models, time dependence is important. Figure 2.14 shows an example where two interventions before k_{i+2} and k_{j-1} happen and changing the Bayesian network representing the causal relationships.

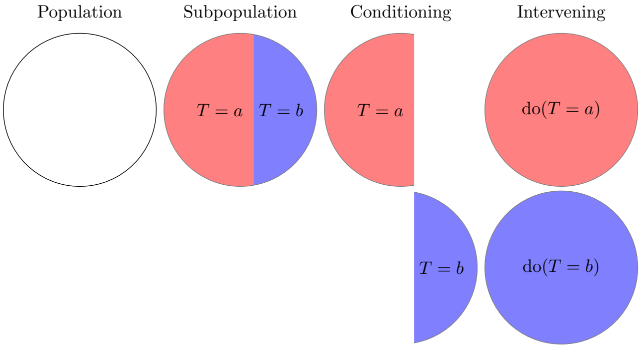


Figure 2.13.: The difference between conditioning and intervening (Source: [Neal, 2020])

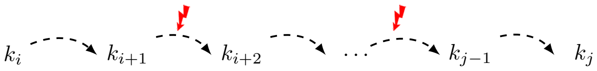


Figure 2.14.: Intervention (red lightning) on cyclic physical mechanisms

2.3. Cooperative interacting automobiles in urban environments in the presence of pedestrians

The work of [Bengler et al., 2014] gives a broad overview of the development of vehicles. The vehicle development began by trying to stabilize the vehicle for exemplary performance in driving performance. Using external perception sensors (Radar, Lidar, Cameras) made it possible to increase safety, comfort, and efficiency. This section gives an overview of the interaction and cooperation between an autonomous vehicle (AV) and a pedestrian. This interaction's physical- and cognitive nature is of interest in building safe AVs. Current approaches often use open source platforms like the robot operating system (ROS), and Autoware [Kato et al., 2018, Koubâa et al., 2017]. Figure 2.15 shows the processing steps for cooperative interacting vehicles. With prior knowledge provided by a cloud service or a database, historical movement data and indirect environmental influences can result in the human prediction algorithm. The blue steps in figure 2.15 are information processing tasks to get information about the pedestrian and the environment (exterior-perspective). The green steps represent information processing tasks incorporating the vehicle's dynamic system (interior per-

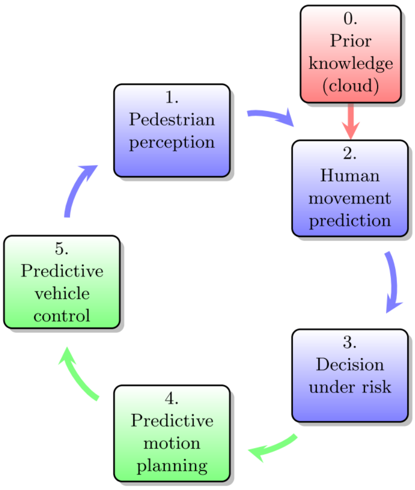


Figure 2.15.: Flow chart of the processing steps of a cooperative interacting vehicle. Red: Cloud services. Blue: Information processing. Green: Control & motion planning [Hartmann et al., 2018b]

spective). [Eilbrecht et al., 2017] presents an example of a cooperative control approach with pedestrian information and movement prediction from a research intersection. An accurate pedestrian perception significantly impacts movement prediction, risk-taking, motion planning, and vehicle control. Pedestrian recognition is an essential prerequisite for safe path planning in urban environments. Also, the availability of data is of interest. Is data only available from the onboard vehicle sensors (dynamic), or is (historical-) data available from the infrastructure and transmitted via internet communication (e.g., LTE, 5G). Another safety aspect is that autonomous vehicles can drive safely and independently without any environmental sensors from the infrastructure. However, on the other hand, additional information from environmental sensors might affect the risk policy. [Geronimo et al., 2009] is a survey of pedestrian detection for advanced driver assistance systems. [Janai et al., 2017] gives a general survey paper about computer vision for autonomous vehicles. A state-of-the-art calibration procedure can be found in [Zhang, 2000]. [Redmon and Farhadi, 2017] is a good reference for object classification.

2.3.1. Motion Planning

The books of [Papageorgiou et al., 2015,Boyd et al., 2004] give a good introduction to optimization. We start with a simple definition of a mathematical optimization:

Definition 2.3.1. Definition adapted from [Boyd et al., 2004]: A mathematical optimization has the form:

$$\min \mathbf{f}_0(\boldsymbol{x}) \tag{2.54}$$

$$\text{subject to}$$

$$\mathbf{f}_i(\boldsymbol{x}) \leq b_i,\ i = 1, \ldots, m \tag{2.55}$$

The vector $\boldsymbol{x} = [x_1, \ldots, x_n]$ is the optimization variable of the problem. $\mathbf{f}_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function. $\mathbf{f}_i : \mathbb{R}^n \rightarrow \mathbb{R},\ i = 1, \ldots, m$ are inequality functions representing the constraints. The constants $b_1, \ldots, b_m$ representing the limits or bounds. A special class for optimization problems linear programs:

$$\min \boldsymbol{c}^T \boldsymbol{x} \tag{2.56}$$

$$\text{subject to}$$

$$\boldsymbol{a}_i^T \boldsymbol{x} \leq b_i,\ i = 1, \ldots, m \tag{2.57}$$

The vectors $\boldsymbol{c}, \boldsymbol{a}_1 \ldots, \boldsymbol{a}_m \in \mathbb{R}^n$ and scalars $b_1, \ldots, b_m \in \mathbb{R}$ are the parameters of the linear program. The problem with robots or autonomous vehicles is that they are receiving information from outside (perception), and they have to plan future movements (motion planning) before acting (control). Suppose the perception unit has an object list of obstacles. In that case, the vehicle can try to plan a trajectory to avoid collisions. The definition for motion planning:

Definition 2.3.2. Motion planning is the approach to finding suitable future reference trajectories for a dynamic system under different constraints depending on the environment perception and internal processes and constraints. The motion planning algorithm must fulfill some requirements, e.g., following the trajectory optimally. A typical technical realization is to optimize a cost function with constraints on the system dynamics, the control input, and other internal (initial state) and environmental constraints (collision avoidance):

$$J_{\min} = \min_{\boldsymbol{u}(t) \in \mathcal{U}} \int_{t=0}^{T} \mathrm{J}(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta) \, dt, \tag{2.58}$$

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta), \tag{2.59}$$

$$\boldsymbol{x}(t=0) \in \mathcal{X}(t=0), \tag{2.60}$$

$$\boldsymbol{x}(t) \in \mathcal{X}^{s}(t) \subset \mathcal{X} \;\forall\; t \in [0, T], \tag{2.61}$$

$$\boldsymbol{x}(t) \notin \mathcal{X}^{c}(t) \subset \mathcal{X} \;\forall\; t \in [0, T], \tag{2.62}$$

$$\boldsymbol{u}(t) \in \mathcal{U} \;\forall\; t \in]0, T] \tag{2.63}$$

For the discrete case:

$$J_{\min} = \min_{\boldsymbol{u}_{k_{i+q}} \in \mathcal{U}} \sum_{q=0}^{n_T} \mathrm{J}(\boldsymbol{x}_{k_{i+q}}, \boldsymbol{u}_{k_{i+q}} \theta) \tag{2.64}$$

$$\boldsymbol{x}_{k_{i+1}} = f(\boldsymbol{x}_{k_i} \boldsymbol{u}_{k_i}, \theta) \tag{2.65}$$

$$\boldsymbol{x}_{k_i} \in \mathcal{X}_{k_i} \tag{2.66}$$

$$\boldsymbol{x}_{k_q} \in \mathcal{X}^{s}_{k_q} \tag{2.67}$$

$$\boldsymbol{x}_{k_q} \notin \mathcal{X}^{c}_{k_q} \tag{2.68}$$

$$\boldsymbol{u}_{k_q} \in \mathcal{U} \tag{2.69}$$

$$\forall\; k \in \mathbb{T} \tag{2.70}$$

Formula 2.58 shows an optimization problem with cost-function J for the vehicle or robot for the time-horizon $[0, T]$ with the domain dependent from state $\boldsymbol{x}(t)$, control-input $\boldsymbol{u}(t)$ and parameter vector θ. The other formulas Formula 2.59-2.63 are constraints for the optimization problem. Formula 2.59 is describing that the vehicle constrained under nonlinear dynamics (valid also for linear dynamics $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$). Formula 2.60 constraints the initial state. Formula 2.61 describes that the state is only allowed in $\mathcal{X}^{s}$ for the whole time horizon and not in the unsafe time-dependent collision states Formula 2.62. Both time dependent sets $\mathcal{X}^{s}(t), \mathcal{X}^{c}(t) \subset \mathcal{X}$

are subset of the fixed state space (only for the time horizon). Formula 2.63 is only allowing a fixed control input set. [LaValle, 2006, Bertsekas et al., 2000, Thrun, 2002, Kirk, 2004, Athans and Falb, 2013, Kouvaritakis and Cannon, 2016] gives an overview of topics about motion planning techniques, optimal control, and model predictive control (optimal control with prediction models). The robotics community introduced many algorithms for motion planning. These algorithms are not always adequate for vehicles because of the nonlinear dynamics of the vehicle [Rajamani, 2012] (e.g., holonomic vs. non-holonomic systems). Sampling-based approaches are well known for motion planning [Lavalle, 1998, Karaman and Frazzoli, 2011]. Rapidly-exploring random trees are representatives ((kinodynamic) RRT, RRT*) for sampling-based motion planning and graph searching algorithms [Starek et al., 2014, Dolgov et al., 2008] Optimal control [Bertsekas et al., 1995, Lewis et al., 2012, Kirk, 2004] is a systematic approach to find future reference trajectories (e.g. for quadrocopters [Allen and Pavone, 2016]). Mixed Integer Linear Programming (MILP) [Schouwenaars, 2006] is well known to formulate logical expressions, e.g., for collision avoidance and as constraints for optimization. Motion planning is not the primer focus for further development but rather the use of adaptive reachable sets mentioned in the following sections. Finally, in [De Nicolao et al., 2007b], an approach to assess the risk of collision with vehicles and pedestrians, based on the scenarios and pedestrian behavior, is discussed. Approaches for motion planning like Mixed Integer Linear Programming (MILP) approaches from [Schouwenaars, 2006] have been used in [Hartmann et al., 2018b, Hartmann et al., 2018c, Hartmann and Watzenig, 2019b, Hartmann and Watzenig, 2019a]. Dynamic obstacles were described as reachable sets to compute adaptive driving maneuvers. [Hartmann et al., 2018c] provides extensively parameter variation to get optimal parameters. In [Hartmann et al., 2017a] optimal control methods were tested under parameter variation of pedestrian models. The model-based formulation of state-prediction could defined as following: We have an known dynamic system $\dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$. We can observe realizations of the dynamic system. We get measurements from the dynamic system. We have an unknown dynamic system where we only see measurement data. We could have some assumptions to solve the state-prediction problem if there is macroscopic energy causing the state change. [2] We have a

[2] Described in Newton-mechanics, without any particular kind of dynamics (relativistic

pedestrian with unknown dynamic system $\dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ (body). The control signals are coming from the human brain with an unknown control policy $\pi(\boldsymbol{u}(t), \boldsymbol{c}(t))$, where $\boldsymbol{c}(t)$ is the cognitive state. This control policy is influenced by different factors (compare section B.1 and section B.2) and time-dependent. The complexity of control mechanisms of the human body is enormous and the control input set $\mathcal{U}(t)$. Also the reachable sets of the dynamic system are changing over time $\mathcal{R}(t)$ and are not known. We can observe realizations of the dynamic system. We get could measure the body movements. It might be easier to model the physical dynamics like in figure 2.17 than modeling the cognitive dynamics. We could represent an external force on a topological space in form of a kinematic or dynamic formulation represented in figure 2.17. This thesis proposes to do basic research on B and a test environment for measurements analysis is proposed in section 3.7. Also, the questions how the cognitive dynamics are influenced by external influences (interaction, causal-chain between environment, brain, and body). It might be that we could have enough trajectories to estimate movements for a dataset with good performance. However, there might be the problem of only considering correlations rather than causality. If the situational aspects might change also, the behavior could cause different behavior[3]. We assume some simplifications (modeling pedestrian behavior by a constant control input set). If we have a dynamic system $\dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ with constant control-input set $\mathcal{U}$. The reachability analysis $\mathcal{R}$. We observe trajectories from this system. Can we infer or over-approximate it by $\{\mathcal{Y}_{k_q}\}$ by only considering the observed trajectories? If we have a pedestrian walking with control input set $\mathcal{U}(t)$ changing over time t. The idea is to over-approximate $\hat{\mathcal{U}}(t) \subset \mathcal{U}(t) \; \forall \; t \in [t_i, t_j]$ if we would know the initial state set $\mathcal{X}_{t_i}$ and the function of the system $\mathbf{f}$, which is not the case. Therefore we could also use a data-based approach. We use the idea of figure 3.44a in the form of a kinematic approach to computing the closure of the velocity set for the translation of the particle. The multi-vector field computes the closure of the velocity set for pedestrian movement. In each manifold cell $\mathcal{X}_q$, we define a random velocity set in the form of a random set. Instead of defining a velocity vector-like in formula 3.53, we rather formulate a velocity vector

effects).

[3] Unusual events like the Covid19 pandemic caused lockdowns with effect on the behavior

set. We compute the estimation funnel $\{\hat{\mathcal{Y}}(t)\}_{t\in[t_i,t_j]}$ to capture all movements. We define a random topological space on a clothoid compare figure 3.44b. Each cell $\mathcal{X}_q$ has its own maximal velocity $\boldsymbol{v}_{\text{max}}^{\mathcal{X}_q}$ and acceleration $\boldsymbol{a}_{\text{max}}^{\mathcal{X}_q}$ (represented as blue and red bars).

Motion planning with non-linear vehicle dynamics

We have a vehicle with nonlinear system dynamics $\dot{\boldsymbol{x}}^v(t) = \mathbf{f}(\boldsymbol{x}^v(t), \boldsymbol{u}^v(t))$ with vehicle state $\boldsymbol{x}^v(t)$ and control input $\boldsymbol{u}^v(t)$. The superscript v stands for the vehicle. We could define a cost function $\mathrm{J}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ and a control policy π which tries to minimize the cost function. A control policy π_r is better than another control policy π_o if the cost $J_r > J_o$. As a constraint we might consider a safety distance to any obstacle, forbidden areas (sidewalks) or dynamic obstacles like pedestrians with state $\boldsymbol{x}^p(t)$ and set $\mathcal{X}^p(t)$. We might have different strategies to incorporate the constraints (multi-cost optimization and Pareto optimal strategies, hard, soft-constraints). We could assume the vehicle with state $\boldsymbol{x}^v(t) = [s_x^v(t), s_y^v(t), v_x^v(t), v_y^v(t)]^T$ and position $\boldsymbol{p}^v(t) = [s_x^v(t), s_y^v(t)]^T$ and velocity $\boldsymbol{v}^v(t) = [v_x^v(t), v_y^v(t)]^T$ with a linear dynamic system (like the point mass model in [Althoff and Wuersching, 2020]):

$$\dot{\boldsymbol{x}}^v(t) = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{x}^v(t) + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{B}} \boldsymbol{u}^v(t) \tag{2.71}$$

Instead of using a linear model, we can also use a kinematic single-track model [Althoff and Wuersching, 2020, Althoff et al., 2017] with $\dot{\boldsymbol{x}}^v(t) = \mathbf{f}(\boldsymbol{x}^v(t), \boldsymbol{u}^v(t))$. The following formulas defining the vehicle dynamics are copied from [Althoff and Wuersching, 2020] and programmed with Python in PyVista©. Figure 2.16 shows the state of the vehicle:

$$\boldsymbol{x}^v(t) = [s_x(t), s_y(t), \delta(t), v(t), \Psi(t)] \tag{2.72}$$

The vehicle state (blue variables) with the position of the vehicle $\boldsymbol{p}^v(t) = [s_x(t), s_y(t)]^T$, the steering angle $\delta(t)$, the vehicle velocity $v(t)$ and the vehicle heading angle $\Psi(t)$. The control input $\boldsymbol{u} = [v_\delta(t), a_{\text{long}}(t)]^T$ in this model consists of steering velocity $v_\delta(t)$ and vehicle acceleration $a_{\text{long}}(t)]^T$. The parameters of the vehicle $\theta = [l_{wb}, l, w]$, with the wheelbase l_{wb}, the vehicle width w and length l. In figure 2.16 we use a slightly different approach with a rectangle $\mathcal{R}^v$ (yellow on the right bottom picture in figure 2.16), defined with l_b and l_{wb}. Afterwards with the Minkowski sum we could define the light yellow area with a circle $\boxed{\mathcal{C}}$ with radius r_m $\boxed{\mathcal{R}}^v \oplus \boxed{\mathcal{C}}$ which could be used later. The height of the vehicle is l_h.

$$
\begin{align}
\dot{\delta}(t) &= v_\delta(t) \tag{2.73}\\
\dot{\Psi}(t) &= \frac{v(t)}{\mathrm{l_{wb}}} \tan\left(\delta(t)\right) \tag{2.74}\\
\dot{\boldsymbol{v}} &= a_{\text{long}}(t) \tag{2.75}\\
\dot{s}_x(t) &= -v\cos(\Psi(t)) \tag{2.76}\\
\dot{s}_y(t) &= v\sin(\Psi(t)) \tag{2.77}
\end{align}
$$

The states

$$x_1(t) = s_x(t),\ x_2(t) = s_y(t),\ x_3(t) = \delta(t),\ x_4(t) = v(t),\ x_5(t) = \Psi(t) \tag{2.78}$$

$$u_1(t) = v_\delta(t),\ u_2(t) = a_{\text{long}}(t) \tag{2.79}$$

and the state space model

$$
\begin{align}
\dot{x}_1(t) &= x_4(t)\cos(x_5(t)) \tag{2.80}\\
\dot{x}_2(t) &= -x_4(t)\sin(x_5(t)) \tag{2.81}\\
\dot{x}_3(t) &= \mathbf{f}_{\text{steer}}(x_3(t), u_1(t)) \tag{2.82}\\
\dot{x}_4(t) &= \mathbf{f}_{\text{acc}}(x_4(t), u_2(t)) \tag{2.83}\\
\dot{x}_5(t) &= \frac{x_4(t)}{\mathrm{l_{wb}}}\tan(x_3(t)) \tag{2.84}\\
& \tag{2.85}
\end{align}
$$

We incorporate the friction circle (Kamm's circle) limiting absolute acceleration:

$$\sqrt{a_{\text{long}}^2 + (v\dot{\Psi})^2} \leq a_{\max} \; (a_{\text{lat}} = v\dot{\Psi}) \tag{2.86}$$

We consider the friction circle by limiting the control input variable u_2:

$$u_2 \leq \sqrt{a_{\max}^2 + (x_4 \cdot \dot{x}_5)^2} \tag{2.87}$$

The parameters for the vehicle are chose for the vehicle length $l = 4.298\ [m]$, vehicle width $w = 1.674\ [m]$ and wheelbase $\mathrm{l}_{\text{wb}} = 2.391\ [m]$. We have steering and acceleration constraints:

$$v_\delta \in [\underline{v}_\delta, \overline{v}_\delta] \tag{2.88}$$

$$\delta \in [\underline{\delta}, \overline{\delta}] \tag{2.89}$$

$$v \in [\underline{v}, \overline{v}] \tag{2.90}$$

With the limited engine power and braking power:

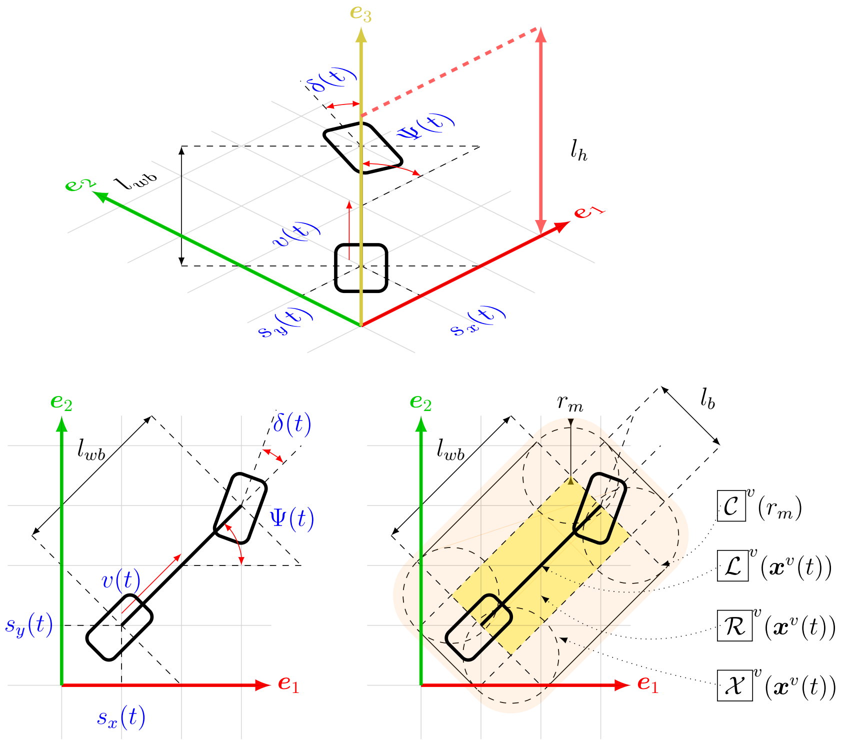
$$a_{\text{long}} \in [\underline{a}, \overline{a}(v)],\ \overline{a}(v) = \begin{cases} a_{\max} \frac{v_S}{v} \text{ for } v > v_S \\ a_{\max} \text{ otherwise} \end{cases} \tag{2.91}$$

For the steering, velocity and acceleration constraints:

$$v_\delta = \mathbf{f}_{\text{steer}}(\delta, v_{\delta,d}) = \begin{cases} 0 \text{ for } (\delta \leq \underline{\delta} \wedge v_{\delta,d} \leq 0) \vee (\delta \geq \overline{\delta} \wedge v_{\delta,d} \geq 0) \text{ (C1)} \\ \underline{v}_\delta \text{ for } \neg\text{C1} \wedge v_{\delta,d} \leq \underline{v}_\delta, \\ \overline{v}_\delta \text{ for } \neg\text{C1} \wedge v_{\delta,d} \geq \overline{v}_\delta, \\ v_{\delta,d} \text{ otherwise} \end{cases} \tag{2.92}$$

For the longitudinal acceleration a_{long} we get:

$$a_{\text{long}} = \mathbf{f}_{\text{acc}}(v, a_{\text{long,d}}) = \begin{cases} 0 \text{ for } (v \leq \underline{v} \wedge a_{\text{long},d} \leq 0) \vee (v \geq \overline{v} \wedge a_{\text{long},d} \geq 0) \text{ (C2)} \\ \underline{a} \text{ for } \neg\text{C2} \wedge a_{\text{long},d} \leq \underline{a}, \\ \overline{a}(v) \text{ for } \neg\text{C2} \wedge a_{\text{long},d} \geq \overline{a}(v), \\ a_{\text{long,d}} \text{ otherwise} \end{cases} \tag{2.93}$$


e_3
$\delta(t)$
$\Psi(t)$
l_h
e_2
l_{wb}
e_1
$v(t)$
$s_y(t)$
$s_x(t)$
e_2
l_{wb}
$\delta(t)$
$\Psi(t)$
$v(t)$
$s_y(t)$
e_1
$s_x(t)$
e_2
r_m
l_b
l_{wb}
$\boxed{\mathcal{C}}^v(r_m)$
$\boxed{\mathcal{L}}^v(\boldsymbol{x}^v(t))$
$\boxed{\mathcal{R}}^v(\boldsymbol{x}^v(t))$
e_1
$\boxed{\mathcal{X}}^v(\boldsymbol{x}^v(t))$

$$\boldsymbol{x}^v(t) = [s_x(t), s_y(t), \delta(t), v(t), \Psi(t)]$$

Figure 2.16.: Vehicle set computation with nonlinear dynamics

For the motion planning we could use the classical formulation with optimization of a cost function

$$J^* = \min_{\boldsymbol{u}(t)\in\mathcal{U}} \int_{t\in[t_i,t_j],\boldsymbol{x}(t)\in\mathcal{X}} \mathrm{J}(\boldsymbol{x}(t),\boldsymbol{u}(t)) \tag{2.94}$$

There could also be some constraints considering collision avoidance. We would like to avoid the prediction sets $\hat{\mathcal{Y}}(t)$ or over-approximations $\lceil\mathcal{Y}\rceil$. For the constraints, we could get the set for the vehicle $\mathcal{X}^v(t)$ of the vehicle. We define the yellow rectangle in left bottom picture of figure 2.16. For the black line between the tires $\mathcal{L}^v(t)$, we can define it like this:

$$\boxed{\mathcal{L}}^v(t) = \bigcup_{l_i\in[0,l_{wb}]} (s_x(t) + \cos(\Psi)\cdot l_i, s_y(t) + \sin(\Psi)\cdot l_i) \tag{2.95}$$

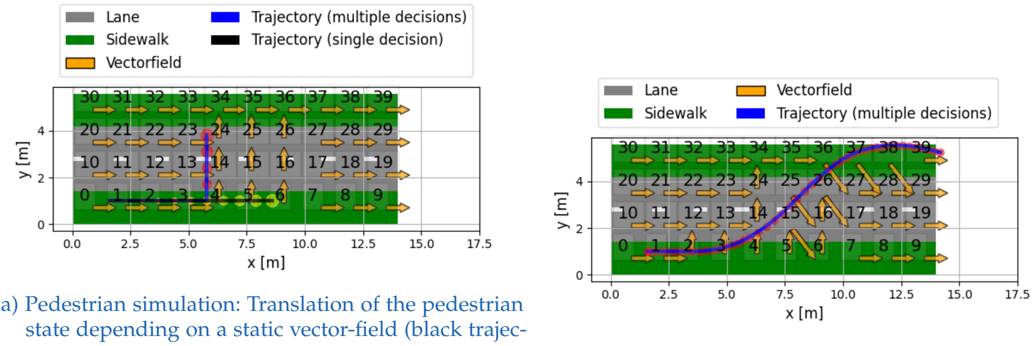
The visualization of figure 2.18 is implemented in Python© with the vehicle dynamics and parameters from [Althoff and Wuersching, 2020]. Only the vehicle parameters and some parts from the formulation of system dynamics are taken from table C.2 (source: [Althoff and Wuersching, 2020,Rajamani, 2012]). For the inner yellow area of the vehicle we have $\boxed{\mathcal{R}}^v(t)$ (compare figure 2.16) we can simply rotate Ψ by $90°$ degrees and compute the corner points (compare figure 2.18). The area of the circle is $\boxed{\mathcal{C}} = \{(p_x, p_y)|p_x^2 + p_y^2 \leq r_m\}$. We can compute the area of the vehicle $\mathcal{X}^v(t)$:

$$\boxed{\mathcal{X}}^v(t) = \boxed{\mathcal{R}}^v(t) \oplus \boxed{\mathcal{C}} \tag{2.96}$$

For the case that we choose r_m as time-dependent, we could $\boxed{\mathcal{X}}^v(t) = \boxed{\mathcal{R}}^v(t) \oplus \boxed{\mathcal{C}}(t)$. In figure 2.18 we compute the convex hull of to compute $\boxed{\mathcal{X}}^v(t) = \mathbf{CH}(\boxed{\mathcal{C}}_1(t) \ldots \boxed{\mathcal{C}}_4(t))$.

2.3.2. Consequences and technical limitations in the presence to pedestrians

Guaranteed safety for all road users is one of the biggest challenges for autonomous vehicles. The consequences of errors could be fatal. For example, human drivers take a risk and make assumptions about the behavior



(a) Pedestrian simulation: Translation of the pedestrian state depending on a static vector-field (black trajectory with one single decision and blue trajectory with multiple decisions depending on the manifold cell)

(b) Pedestrian simulation: Acceleration in topological spaces

Figure 2.17.: Pedestrian moving on a vectorfield

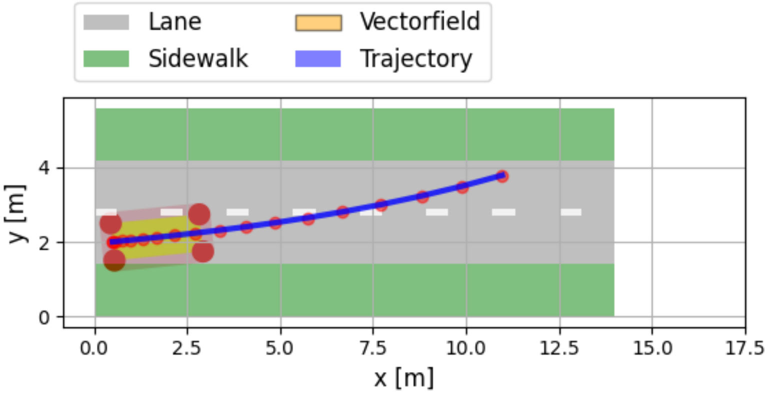


Figure 2.18.: Vehicle simulation with kinematic single-track model

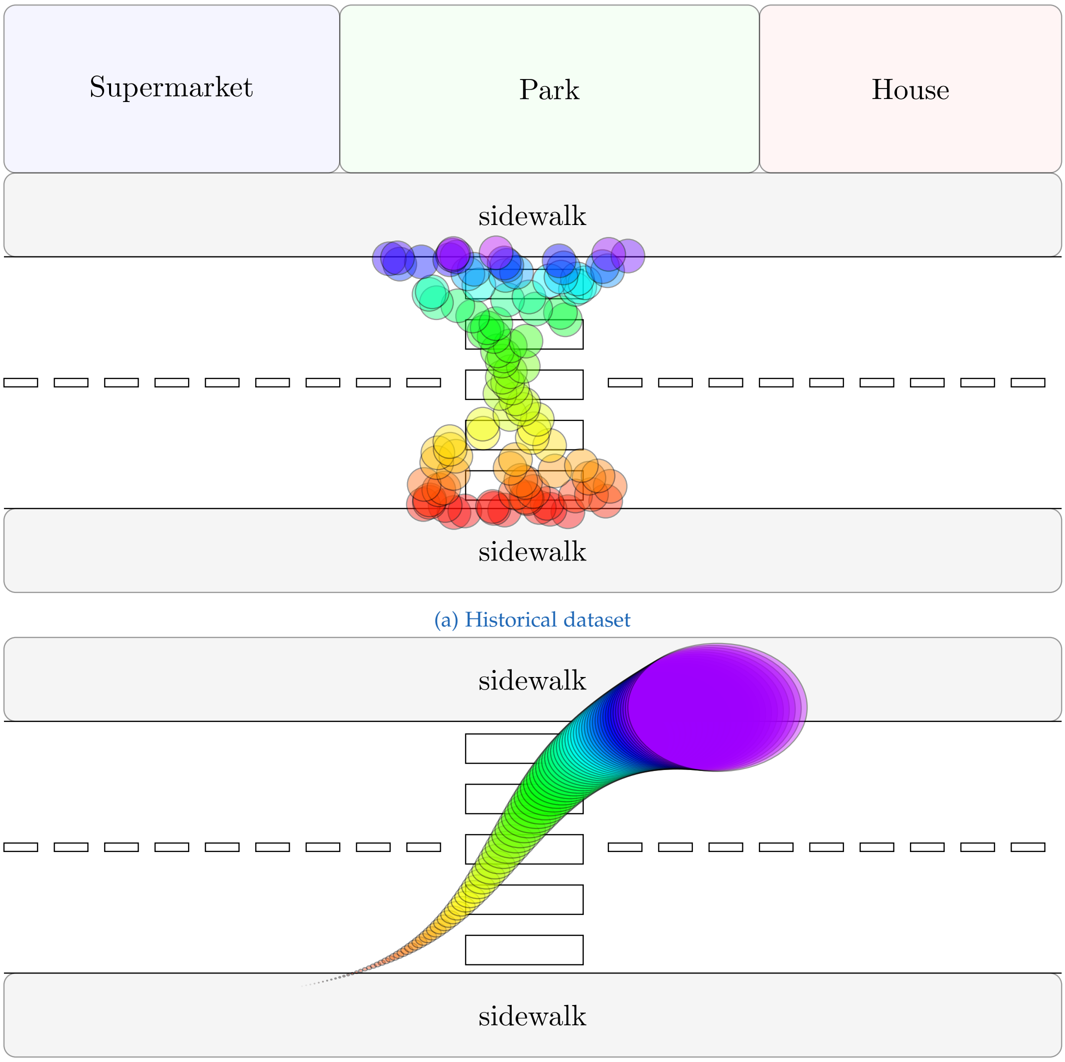
of other road users instead of considering worst-case scenarios in each situation. If an autonomous vehicle predicts a pedestrian's movement, there is tremendous uncertainty in predicting the exact future position. A vehicle cannot measure and detect all relevant information. Predicting future movements with absolute certainty is problematic because the cognitive dynamics are often not observable (intention, emotion, and other dependencies). However, it is unknown and most likely impossible how the prediction model can be generalized to allow certainty in every movement prediction. The complexity also comes from different situations and humans (from all cultural and emotional backgrounds). Current risk indicators (threshold based on probabilistic models) are not enough. This thesis discusses ideas to use concepts from reachability analysis to combine them with machine learning. The aim is to adapt set-based methods with current information from each situation and compute adaptive funnels for motion planning. The interaction between the vehicle and pedestrian might influence the pedestrian's current policy. Therefore the use of causal inference is further discussed.

3. Problem formulation and developed solutions

Section 3.1 gives an overview of the whole chapter.

3.1. Overview of chapter 3

Section 3.2 and 3.3 presents some parts from the approach from [Hartmann and Watzenig, 2019a,Hartmann and Watzenig, 2019b,Hartmann et al., 2018c,Schratter et al., 2019] with a standard motion planning algorithm (section 3.2) adaptive set deformation (section 3.3). Section 3.4 presents the computation of intelligent funnel prediction. Section 3.3 and section 3.4 discusses how to bring the concepts from multimodal movement prediction and reachability analysis together to get adaptive set deformations like in figure 3.4. Section 3.5 and 3.6 focus on theoretical aspects of human locomotion. Section 3.5 is presenting mathematical concepts from differential geometry to model the urban environment as a manifold (compare figure 3.5). Section 3.6 presents methods from causal inference for cooperative interacting automobiles. Causal inference might help to model the interaction between the vehicle and the pedestrian (compare figure 3.6). Section 3.7 gives a theoretical description for this approach. Following comes a visual overview of chapter 3 to explain the context of each section intuitively. Figure 3.1a shows the urban environment from the introduction example. The colored dots represent a fictive historical dataset, which shows the positions of pedestrians walking from the bottom sidewalk to the upper sidewalk. The colors of the dots or areas in the following figures represent timestamps; compare figure 3.2. The measured positions could not tell us the intention of the pedestrian. Only under some circumstances might this dataset lead to good predictions. The pedestrian's intention is not observable (e.g., if he or she wants to go to the supermarket, park, or house). It might also be the case that the pedestrian has an intention to change the direction at the crosswalk. Many movement prediction algorithms (e.g. statistical movement prediction or reachability analysis) represent a funnel-like object (set of sets like in figure 3.1b). Again the timeline of figure 3.2 is used. Figure 3.3a shows an example of a multimodal movement prediction with three scenarios (A, B, C). Suppose the autonomous vehicle has to choose one of the predictions. In that case, it might be interesting for safety how trustable is one of these three options. Worst-case reachability analysis considers all possible movements under consideration of physical constraints (compare figure 3.3b. The problem with this approach is that the vehicle cannot enter the reachable sets (black lightning). This restriction might be a problem when the vehicle drives on the street and the pedestrian-only walks on the sidewalk. The autonomous vehicle might not pass the region of the pedestrian. The approach in the next sections is to consider machine learning with reachability analysis to compute adaptive sets. It would be ideal to have something like in figure 3.4 where a trustable prediction might help to find probable positions with very high certainty. Figure 3.5 shows another urban environment which is structured as a topological space. The semantic meaning for a pedestrian of each building or street might influence the decisions of a person and the resulting behavior (compare also section B.1 and B.2 for further details). Another problem is that the perception of the autonomous vehicle might influence the movements of the pedestrian (compare figure 3.6). The perception of an oncoming vehicle might trigger an intention change so that he or she goes back to the sidewalk, especially in situations without crosswalks. The prediction does not consider the interaction. The funnel of the prediction does not capture the real movement. Figure 3.7 illustrates an exemplary outcome of the situation, where the trajectories have an intersection. However, there is no collision because the intersection of the trajectories is on different timestamps.



(a) Historical dataset

(b) Prediction of the future movement (funnel)

Figure 3.1.: Movement prediction with pedestrian

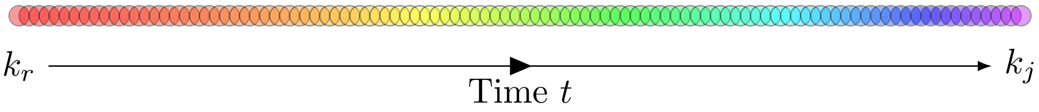
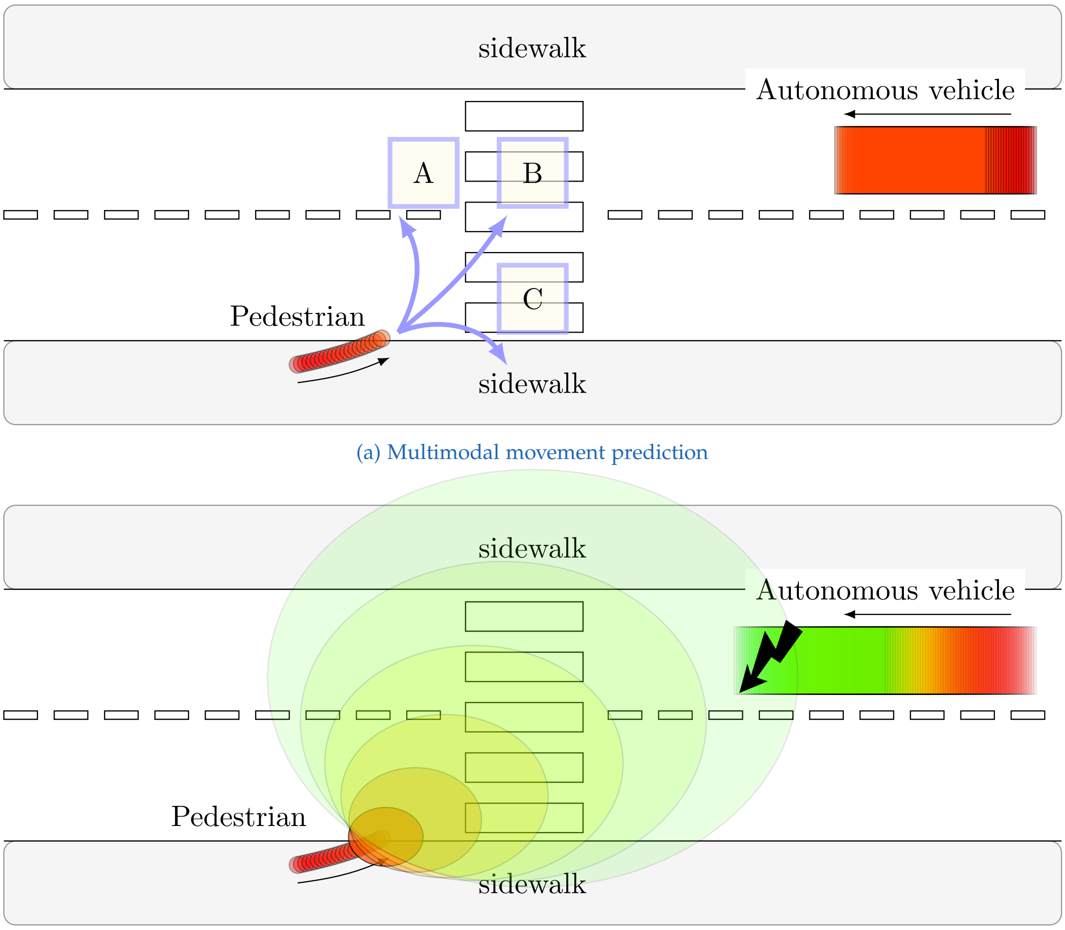


Figure 3.2.: Colorized timeline for trajectories and reachable sets



(a) Multimodal movement prediction

(b) Worst-case reachability analysis

Figure 3.3.: Multimodal movement prediction and reachability analysis

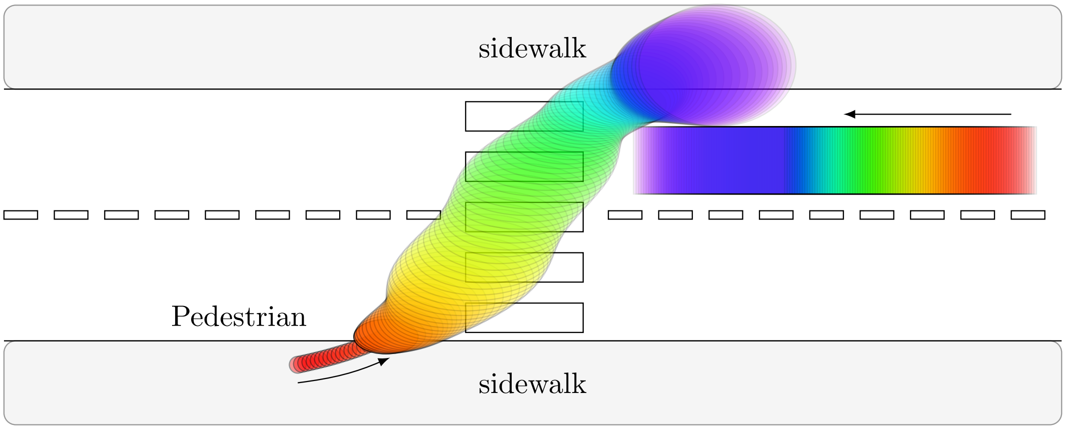


Figure 3.4.: Adaptive reachability analysis and intelligent funnel prediction

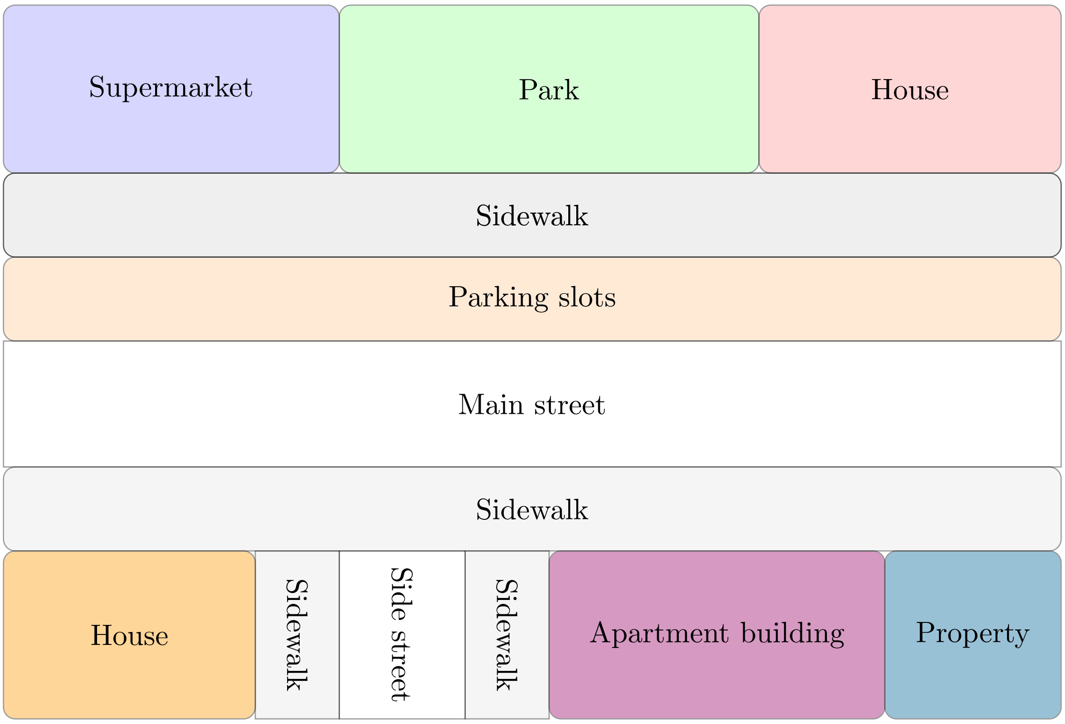


Figure 3.5.: Urban environment as topological space

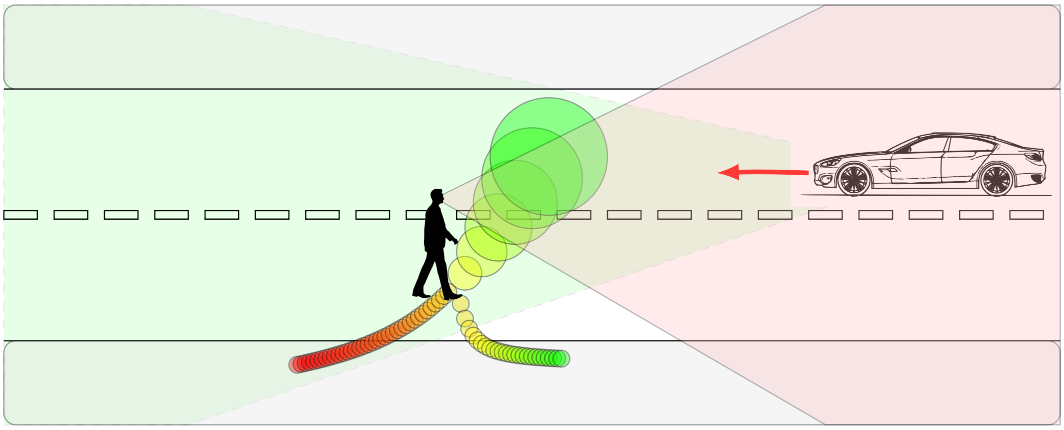


Figure 3.6.: Interaction between vehicle and pedestrian

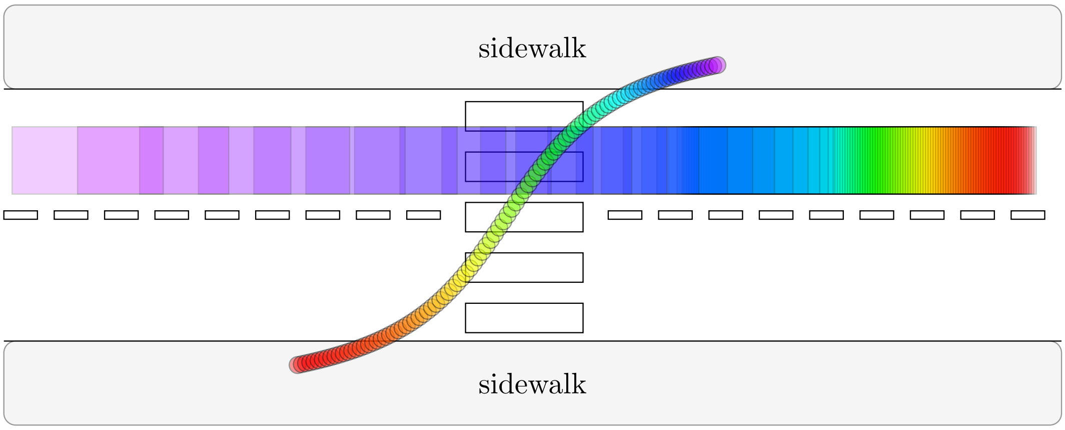


Figure 3.7.: Exemplary outcome of the situation without a collision

3.2. Motion planning for the autonomous vehicle

A particular emphasis plays vulnerable road users, but this short section should dig into the autonomous vehicle and how the vehicle incorporates the movement predictions. The idea of motion planning is to find reference trajectories. A significant requirement is that the reference trajectories do not lead to collisions and are valid for the vehicle dynamics, and are optimal in a certain sense. This section presents ideas from [Hartmann and Watzenig, 2019a] with Mixed Integer Linear Programming (MILP) for computing optimal reference trajectories. The represented approach is relatively simple compared to modern model predictive approaches, incorporating non-linearities. New is that it was combined with movement prediction algorithms for pedestrians and in interconnected coordinate systems (e.g., manifold), the content of later sections. The origin of the approach is [Schouwenaars, 2006,Hartmann and Watzenig, 2019a,Hartmann and Watzenig, 2019b,Hartmann et al., 2018c]. A linear continuous system approximates the vehicle dynamics. Nonlinear vehicle models can be found in [Althoff et al., 2017,Rajamani, 2012]. We start with a continuous system:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A} \cdot \boldsymbol{x}(t) + \boldsymbol{B} \cdot \boldsymbol{u}(t),\ \boldsymbol{A} \in \mathbb{R}^{n \times n},\ \boldsymbol{B} \in \mathbb{R}^{n \times m} \tag{3.1}$$

We choose a simple integrator model with the state vector $\boldsymbol{x}(t)$ and the positions of a particle $\boldsymbol{p}(t) = [p_x(t), p_y(t)]^T$ and the velocities $\boldsymbol{v}(t) = [v_x(t), v_y(t)]^T$. The control input is simply a force changing the velocity $\boldsymbol{u}(t) = [u_x(t), u_y(t)]^T$:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.2}$$

$$\boldsymbol{x}(t) = \begin{bmatrix} p_x(t) \\ p_y(t) \\ v_x(t) \\ v_y(t) \end{bmatrix} \boldsymbol{u}(t) = \begin{bmatrix} u_x(t) \\ u_y(t) \end{bmatrix} \tag{3.3}$$

For simplicity we assume that the state is directly measurable $\boldsymbol{y}(t) = \boldsymbol{x}(t)$ and without considering discretization problems we get the following discrete system:

$$\boldsymbol{x}_{k_{i+1}} = \Phi \boldsymbol{x}_{k_i} + \boldsymbol{H} \boldsymbol{u}_{k_i} \tag{3.4}$$

Again we use the timespace $\mathbb{T}$ from formula 2.19 and the current time-stamp k_i to the future time-stamp k_j with $k_i, k_j \in \mathbb{T}$. We misuse the arrow-symbol $\rightarrow$ in the subscript of the following variables to represent the time-horizon like $k_i \rightarrow k_j$. Figure 3.8 shows the description for the positions, velocities

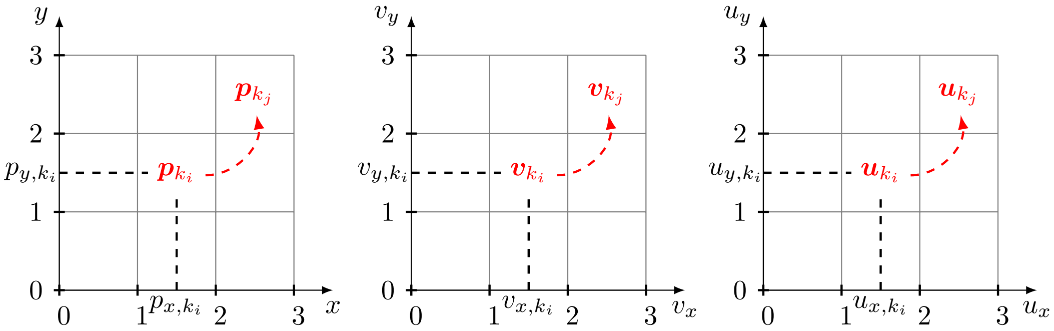


Figure 3.8.: Description of future positions, velocities and control inputs

and control inputs. An optimization problem is formulated to get future positions in x-direction $\boldsymbol{p}_{x,k_i \to k_j}$ and y-direction $\boldsymbol{p}_{y,k_i \to k_j}$:

$$\boldsymbol{p}_{x,k_i \to k_j} = \begin{bmatrix} p_{x,k_j} & \cdots & p_{x,k_i} \end{bmatrix}^T \tag{3.5}$$

$$\boldsymbol{p}_{y,k_i \to k_j} = \begin{bmatrix} p_{y,k_j} & \cdots & p_{y,k_i} \end{bmatrix}^T \tag{3.6}$$

For the velocities in x-direction $\boldsymbol{v}_{x,k_i \to k_j}$ and y-direction $\boldsymbol{v}_{y,k_i \to k_j}$:

$$\boldsymbol{v}_{x,k_i \to k_j} = \begin{bmatrix} v_{x,k_j} & \cdots & v_{x,k_i} \end{bmatrix}^T \tag{3.7}$$

$$\boldsymbol{v}_{y,k_i \to k_j} = \begin{bmatrix} v_{y,k_j} & \cdots & v_{y,k_i} \end{bmatrix}^T \tag{3.8}$$

For the control inputs we get:

$$\boldsymbol{u}_{x,k_i \to k_j} = \begin{bmatrix} u_{x,k_j} & \cdots & u_{x,k_i} \end{bmatrix}^T \tag{3.9}$$

$$\boldsymbol{u}_{y,k_i \to k_j} = \begin{bmatrix} u_{y,k_j} & \cdots & u_{y,k_i} \end{bmatrix}^T \tag{3.10}$$

We stack the variables together to represent the optimization variable for the motion planning:

$$\boldsymbol{v}_{k_i \to k_j} := \begin{bmatrix} \boldsymbol{p}_{x,k_i \to k_j}^T & \boldsymbol{p}_{y,k_i \to k_j}^T & \boldsymbol{v}_{x,k_i \to k_j}^T & \boldsymbol{v}_{y,k_i \to k_j}^T & \boldsymbol{u}_{x,k_i \to k_j}^T & \boldsymbol{u}_{y,k_i \to k_j}^T \end{bmatrix}^T \tag{3.11}$$

We could also find other arrangements for the optimization variable $\boldsymbol{v}_{k_i \to k_j}$. Important is that the optimization variable has the information from all

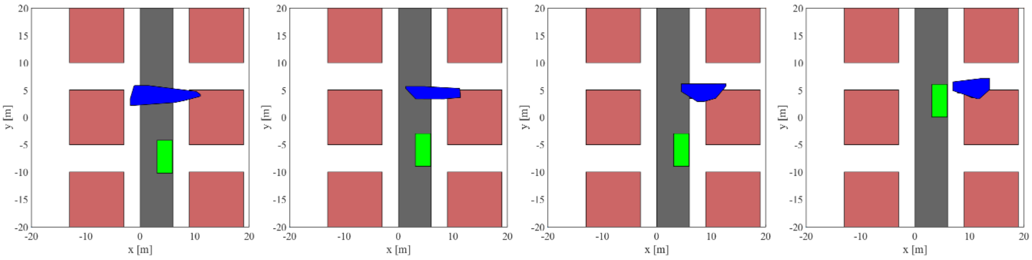


Figure 3.9.: Prediction of all vertices [Hartmann et al., 2018b]

state vectors in the future time horizon $\boldsymbol{x}_{k_i}, \boldsymbol{x}_{k_{i+1}} \dots, \boldsymbol{x}_{k_j}$. Instead of using the arrow-symbol to represent one single timestamp $\boldsymbol{v}_{k_q \to k_q}$, we use $\boldsymbol{v}_{k_q}\ \forall\ q \in \mathbb{T}$. The same holds for other variables with the arrow symbol. It is not possible to stack them together in advanced motion planning algorithms. However, in later sections, we will show how to use multiple connected coordinate systems instead of one single euclidean system. Target is to minimize the cost function $\mathrm{J}(\overline{\boldsymbol{v}}_{k_i \to k_j})$:

$$\boldsymbol{v}^*_{k_i \to k_j} = \arg \min_{\boldsymbol{v}_{k_i \to k_j}} \mathrm{J}(\boldsymbol{v}_{k_i \to k_j}) \tag{3.12}$$

subject to

$$\boldsymbol{x}_{k_i} \in \mathcal{X}_{k_i} \tag{3.13}$$

$$\boldsymbol{x}_{k_j} \in \mathcal{X}^{\text{target}}_{k_j} \tag{3.14}$$

$$(p_{x,k_q}, p_{y,k_q}) \notin \mathcal{O},\ \forall\ q \in \{i, i+1, \dots, j\} \tag{3.15}$$

$$(u_{x,k_q}, u_{y,k_q}) \in \mathcal{U},\ \forall\ q \in \{i+1, i+2, \dots, j\} \tag{3.16}$$

$$\boldsymbol{x}_{k_{i+1}} = \Phi \boldsymbol{x}_{k_i} + \boldsymbol{H} \boldsymbol{u}_{k_i} \tag{3.17}$$

Figure 3.9 shows an example from [Hartmann et al., 2018b]. The green rectangle represents the vehicle position. Four illustrations show four different timesteps, where the vehicle is moving with optimal control and not entering the blue set (representing possible positions of the pedestrian). The grey area representing the street and the red rectangles representing houses without any further importance for the mathematical approach. [Hartmann et al., 2018b, Hartmann et al., 2018c, Hartmann and Watzenig, 2019a, Hartmann and Watzenig, 2019b] provide further details on optimization strategies and

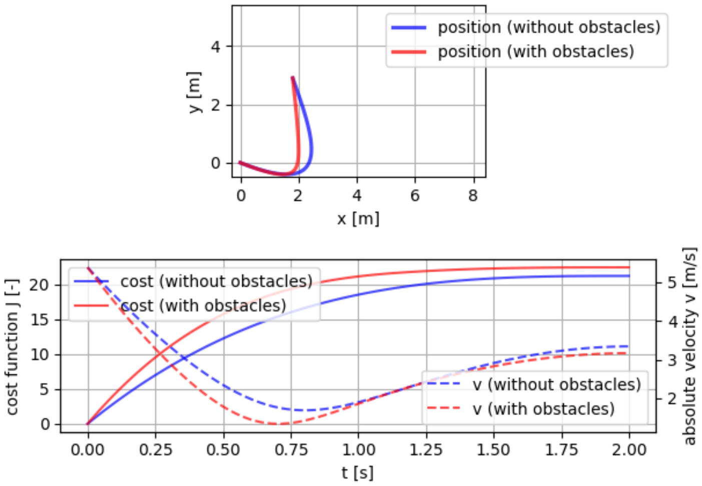


Figure 3.10.: Optimization with and without static obstacle

results. Also how the constraints have been adapted for each scenario is documented in the documents. Figure 3.10 shows an example with two cases, with and without formula 3.15. In the case with constant obstacle $\mathcal{O} = \{x | x >= 2\}$ the x-coordinate at 2 is border for red trajectory in figure 3.10. In both situations all parameters and target variables are the same but only the obstacle formula 3.15. The optimization procedure is done with GEKKO [Beal et al., 2018]. The time horizon is $[0, 2s]$ for 101 discretization points. The target area is

$$\mathcal{X}_{k_j}^{\text{target}} = \{(x, y) | 1.6 < x < 1.8 \ \ \& \ \ 2.9 < y < 3.1\} \tag{3.18}$$

and the control input set:

$$\mathcal{U} = \{(u_x, u_y) | -10 < u_x < 10 \ \ \& \ \ -10 < u_y < 10\} \tag{3.19}$$

and with initial state $\boldsymbol{x}_{k_0} = \begin{bmatrix} 0 & 0 & 5 & -2 \end{bmatrix}^T (= \boldsymbol{x}_{k_i})$. The cost function is defined as $\mathrm{J} = \sum_{q=i}^{j} \frac{1}{2} u_{x,k_q}^2 + \frac{1}{2} u_{y,k_q}^2$. A drawback is that the optimization procedure does not find a solution, if the selection of the constraints is not well balanced.

Model predictive control takes predicted variables into account. In nonlinear vehicles, we use the knot points for linearization with Jacobi-Approximation. So we only consider a linear subspace of the nonlinear system. This method is the approach to performing simulations in the next chapter.

3.3. Adaptive set deformation

In section 3.2 an optimization example was presented for static obstacles. It is necessary to formulate a collision-avoidance approach for the autonomous vehicle with state $\boldsymbol{x}^v(t)$ to avoid also dynamic obstacle sets $\boldsymbol{x}^v(t) \notin \mathcal{O}(t) \; \forall \; t \in \mathbb{R}$. It is not trivial to compute adequate obstacle sets. The obstacle sets of a pedestrian with state $\boldsymbol{x}^p(t)$ (discrete case $\boldsymbol{x}^p_{k_i}$ for time-stamp k_i) depends on the decisions and the future and behavior of the pedestrian. In this section we discusses approaches to deform sets and in section 3.4 how to use this deformations of sets for movement prediction. After computing the predictions it these funnels can be used for motion planning $\mathcal{O}^p(t) \subset \mathcal{O}(t)$. Section 3.3.1 gives an intuitive motivation for adaptive set deformations, section 3.3.2 introduces group actions acting on sets and section 3.3.3 highlights reachability analysis as a special case for adaptive set deformations.

3.3.1. Motivation for adaptive set deformation

Human body movements could be approximated by a nonlinear dynamic system $\boldsymbol{x}_{k_{i+1}} = \mathbf{f}(\boldsymbol{x}_{k_i}, \boldsymbol{u}_{k_i})$. The problem is that much information is unknown (the intention in form of a cognitive state and the resulting control inputs are often hidden, compare section 3.4) so that the common approach is to predict the movements in a probabilistic manner. If we have $p(\mathrm{x}_{k_j}|\mathrm{x}_{k_i})$ a fictive model[1] between the current random variable x_{k_i} (e.g. current position

[1] The model $p(\mathrm{x}_{k_j}|\mathrm{x}_{k_i})$ is only for explanation. Section 2.2.1 discusses movement prediction models. Many models use the historical positions of the person. It can be assumed that the Markov property does not hold for human locomotion.

of the pedestrian) and the future variable x_{k_j} (e.g. future position of the pedestrian) we could compute the most probable value $\hat{\mathrm{x}}_{k_j}$:

$$\hat{\mathrm{x}}_{k_j} = \mathbb{E}_{\mathrm{x}_{k_j} \sim p(\mathrm{x}_{k_j} | \mathrm{x}_{k_i})}[\mathrm{x}_{k_j}] \tag{3.20}$$

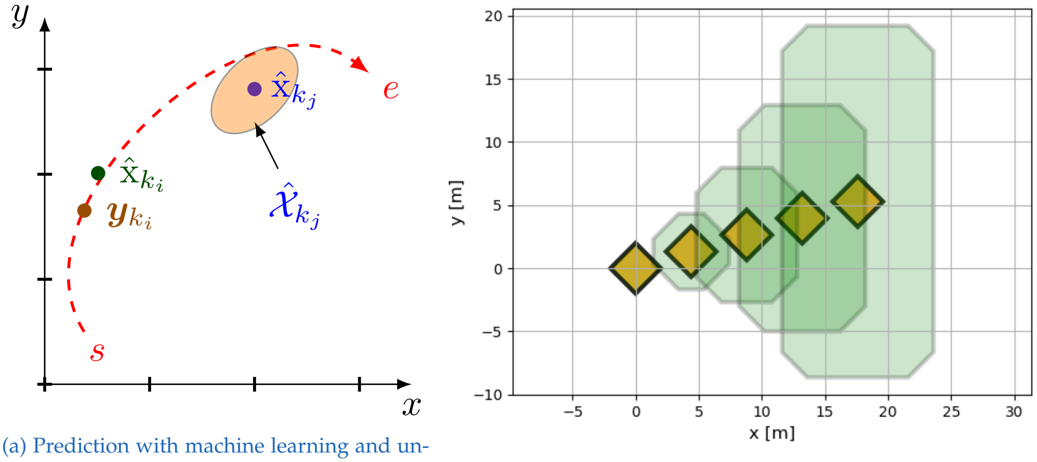
With the variance $\mathrm{Var}(\mathrm{x}_{k_j})$ we could compute an area around the future random variable with approaches from uncertainty quantification [Zio and Pedroni, 2013] or simply by introducing a threshold θ. An example is the Kalman-Filter, where the mean value and variance of the state are predicted and updated iteratively: First, the state is predicted, and after getting a measurement $\boldsymbol{y}_{k_i}$, the prediction is updated. The contours of the Gaussian probability form an ellipse. Figure 3.11a shows an example with a red trajectory from start s to end e[2]. Based on an estimated current state $\hat{\mathrm{x}}_{k_i}$ it might be possible to predict the future state $\hat{\mathrm{x}}_{k_j}$. With error estimation, one can compute an uncertainty area (like the ellipse in Kalman-filter) to compute an area $\hat{\mathcal{X}}_{k_j}$. Following success criteria could help to evaluate the performance of the developed algorithms with customized prioritization:

- Prediction performance: $\hat{\mathcal{X}}_{k_j}$ should capture the real position of the trajectory (like the intersection of the trajectory and the ellipse). This critical criterion ensures safety for all road users.
- Volume reduction: Reduction of the area (2D) or volume compared to reachability analysis. The ratio between the area or volume is predicted with machine learning (belief sets: the areas or volume where the pedestrian will be). The areas computed with classical reachability analysis (plausible sets: all possible physical areas) give us the area reduction.
- Computational complexity: The computational requirements could also be a bottleneck criterion for the realization and application in the vehicles. The computational resources are limited in vehicles.

3.3.2. Group actions and set operations

Instead of using vectors (like $\hat{\mathrm{x}}_{k_i}$), we could use sets $\hat{\mathcal{X}}_{k_i}$. The idea is to increase the complexity to incorporate multiple scenarios (e.g. like in reacha-

[2] For simplicity, the red trajectory does not show the time-labels



(a) Prediction with machine learning and uncertainty quantification

(b) Reachability analysis with zonotopes

Figure 3.11.: Set prediction and reachability analysis with zonotopes

bility analysis). We deform or move the sets by group actions. A group action $(\mathfrak{G}, *)$ can transform a set $\mathcal{X}$. First we start with a single operation, but there are many options available (e.g. multiple operations, hybrid: analytical and data-based, parallel: common operation for parts of the sets). A left group action is $\alpha : \mathfrak{G} \times \mathcal{X} \to \mathcal{X}$ and the right group action is $\alpha : \mathcal{X} \times \mathfrak{G} \to \mathcal{X}$. Two familiar group actions and set-operations are the Minkowski-sum and orthogonal group (SO(n) actions for rotations in n-dimensions) and translation. An example would be iterative application of a constant set operation:

$$\hat{\mathcal{X}}_{k_{i+1}} = \mathfrak{G} * \hat{\mathcal{X}}_{k_i} \tag{3.21}$$

This is a shorthand for:

$$\mathfrak{G} * \hat{\mathcal{X}}_{k_i} := \{\mathfrak{G}(\hat{\mathfrak{x}}) | \mathfrak{G} \in \mathbf{OP}(\dots), \hat{\mathfrak{x}} \in \hat{\mathcal{X}}\} \tag{3.22}$$

This is a mathematically simplified view as $\mathfrak{G}$ "operates" on each element of $\hat{\mathcal{X}}$ with $\mathfrak{G} \in \mathbf{OP}(\dots)$, where $\mathbf{OP}(\dots)$ is a not further explained set of mathematical operations. An example is the Minkowski sum (compare the definition section A.2). Another example for a constant group action is the matrix operation with translation $(t_x, t_y > 0)$ without rotation $(\theta = 0)$. The orange sets from figure 3.11b could be computed by this kind of mathematical iterative procedure. We could use a set $\hat{\mathcal{X}}_{k_0}$ as an initial set.

Also an iterative model with time dependent set-deformations (e.g. group actions) is possible:

$$\hat{\mathcal{X}}_{k_{i+1}} = \mathfrak{G}_{k_i} * \hat{\mathcal{X}}_{k_i} \tag{3.23}$$

$\mathfrak{G}_{k_i}$ is a group action operating on the set $\hat{\mathcal{X}}_{k_i}$ and deforming it at time k_i. We could use familiar group actions (e.g. Minkowski sum or SO(n)). An example might be a reinforcement learning approach which moves a set $\hat{\mathcal{X}}_{k_i}$ with group actions from SO(n) so that it captures a trajectory and a reward function is maximized to capture the unknown trajectory. Target is that the real state is in $\boldsymbol{x}^p_{k_j} \in \hat{\mathcal{X}}_{k_j}\ \forall\ k_j \in \mathbb{T}$ (compare formula 2.19). We could deform this set for the following time-steps $k_i, k_{i+1}, \ldots, k_j$. Each set or each operation on deforming a set has some properties[3], which might be useful for consideration:

- Volume or area: We can compute the volume or area of each set (compare A.2).
- (Non-) convexity: An interested reader can find the definition A.2.2 of a convex set in the appendix.
- Closure property: A set is defined to be closed, if the group action produces a member of the set. An example is the matrix multiplication and a zonotope. The result is again a zonotope. The matrix operates on the generators and the center of the zonotope.
- Topology of the set
- Action on the whole set or a subset. We want to hold some freedom and flexibility in deforming sets. That is why we use methods to deform parts of the sets.
- Set-type: The type of set offers different freedom degrees and also support function, ellipsoid, polyhedron, zonotope (compare figure A.4)
- Computational complexity: The amount of data to represent each sets or the effort for the computation is important for the usage of the algorithms.
- Mathematical concept: Different mathematical disciplines like (multi-) linear algebra, differential geometry, probability theory offer a variety of tools for the deformations.

[3] Some parts of that field are more related to control engineering (e.g. reachability analysis) and other parts from machine learning are introduced in this thesis.

3.3.3. Special case for using group actions: Reachability Analysis

A special case for adaptive set deformation is reachability analysis from section 2.1 where the Minkowski-sum is used to compute the reachable sets of a dynamic system. We want to look if reachability analysis could help to solve the optimization procedures with formula 2.12. An example with reachability analysis and zonotopes is visualized in figure 3.11b, where the approach from [Girard et al., 2006] has been implemented. It is showing the green discrete reachable sets $\mathcal{R}_{k_1}, \dots, \mathcal{R}_{k_4}$ and the orange homogeneous sets $\mathcal{H}_{k_1}, \dots, \mathcal{H}_{k_4}$. It is easy to see that the green sets are getting bigger over time and the orange areas have the same size. The initial zonotope is

$$\begin{aligned} \mathcal{Z}^{\mathcal{X}}_{k_0} = \Big\{ \boldsymbol{c}^{\mathcal{X}} = \begin{bmatrix} 0 & 0 & 10 & 3 \end{bmatrix}^T, \dots \\ \dots \mathcal{G}^{\mathcal{X}} = \{ \boldsymbol{g}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T, \boldsymbol{g}_2 = \begin{bmatrix} -1 & 1 & 0 & 0 \end{bmatrix}^T \} \Big\} \end{aligned} \tag{3.24}$$

and for the control input zonotope

$$\begin{aligned} \mathcal{Z}^{\mathcal{U}} = \Big\{ \boldsymbol{c}^{\mathcal{U}} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T, \dots \\ \dots \mathcal{G}^{\mathcal{U}} = \{ \boldsymbol{g}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T, \boldsymbol{g}_2 = \begin{bmatrix} 0 & 1 & 0 & 3 \end{bmatrix}^T \} \Big\} \end{aligned} \tag{3.25}$$

A variant of formula 2.30 was starting point of the algorithm of [Girard et al., 2006]. The question is if we could use machine learning (**ML**) to adapt the reachability analysis procedure (compare next section 3.4)

$$\mathcal{R}(t, \mathcal{U}(t)) = \mathcal{H}(t) \underbrace{\oplus \mathcal{P}(t, \mathcal{U}(t))}_{\textbf{ML}} \tag{3.26}$$

for only the particular solution or the whole reachable sets

$$\mathcal{R}(t, \mathcal{U}(t)) = \underbrace{\mathcal{H}(t) \oplus \mathcal{P}(t, \mathcal{U}(t))}_{\textbf{ML}} \tag{3.27}$$

The papers in [Hartmann and Watzenig, 2019a,Hartmann and Watzenig, 2019b,Schratter et al., 2019] are examples of reachability analysis enriched

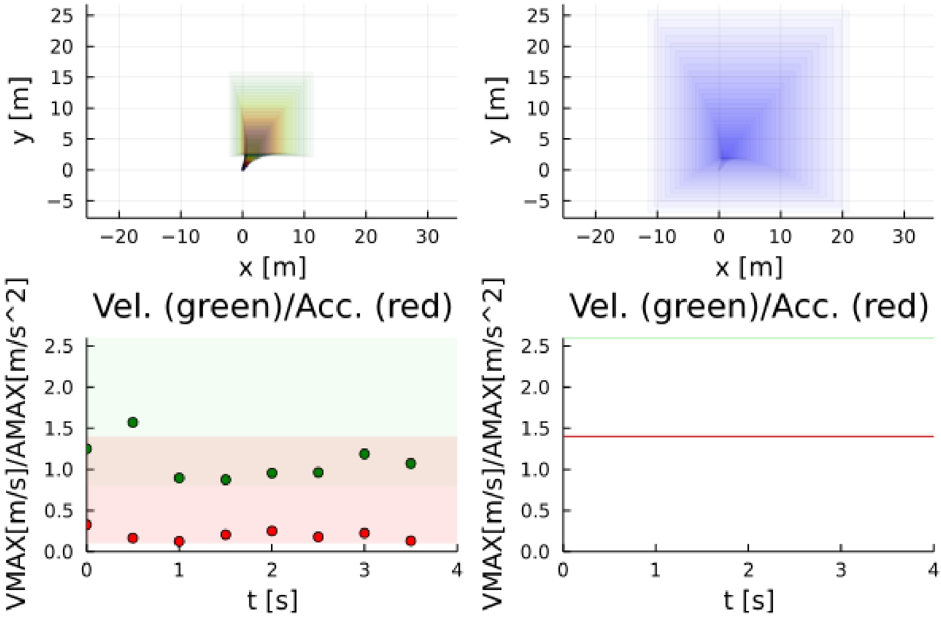


Figure 3.12.: The both top pictures show the result of reachable sets. The two left pictures show a stochastic process where the maximal parameters of velocity (green points) and acceleration (red points) feed the reach set computation. On the right side we have classical parameters for the maximal parameters.

with machine learning. In these approaches the meta-parameters (e.g. maximal velocity and acceleration) of reachability analysis are estimated in each situation (compare section 3.4). The approach in figure 3.12 shows how a stochastic approach is used to sample the maximal velocity and maximal acceleration. The resulting cumulative reach set volume could be huge if the maximal parameters are not well computed. The over-approximation of the set could minimize the difference between both volumes.

3.3.4. Examples for set-based movement prediction

Section 3.4 gives a motivation for set-based deformation and prediction algorithms. This subsection should give a short recap based on two real-world examples. Imagine an elastic ball that is thrown in the air by a catapult or person, and another person is trying to catch the deformed ball with a bag, compare figure 3.13. In physics, one could represent static objects as particles, which is not valid. The ball has a changing volume (area in the current 2D-example) due to the deformations so that it can be represented as changing $\mathcal{X}(t_q),\ t_q \in [t_i, t_j]$ (for the discrete case $\{\mathcal{X}_{k_q}\}_{q=i}^{j}$). It is unknown how the deformed ball is thrown (strength and direction), so the ball's trajectory, deformation, and shape are uncertain. A person or a robot is trying to catch the ball. The person or the robot is using a deformable bag also with time-varying sets $\{\mathcal{Y}_{k_q}\}_{q=i}^{j},\ \forall\, k_q \in \mathbb{T}$. The observed position sets $[t_r, t_i[$ of the ball could help to extrapolate the trajectory to predict the future sets in $[t_i, t_j]$. The bag is catching the ball if $\mathcal{X}(t) \subset \mathcal{Y}(t)$. [4] One could program a robot to predict the future sets $\{\hat{\mathcal{X}}_q\}_{q=i}^{j}$ and formulate a model-predictive set-based optimal-control problem.

The second example is considering an autonomous vehicle on a street network near a bus station in Graz compare figure 3.14. Three layers show the real positions of the road users (Layer 1), the topology (Layer 2), and the maximal velocities (Layer 3). If the autonomous vehicle considers the speed limits of layer 3, this could help consider reachability analysis or set-based prediction. The acceleration is fixed, ensuring the plausible positions on the road-topology with a maximal control input set $\mathcal{U}$ for the vehicle.

3.4. Funnel prediction with belief sets

This section shows concepts for funnel prediction[6], which could lead to risky motion planning. Therefore we give a short motivation to use intelligent funnels in section 3.4.1. Section 3.4.2 gives a detailed view on deforming sets over time, which is an important step on building funnels. Section 3.4.3 shows the incorporation of data-based approaches. Section 3.4.4 gives a short introduction to adaptive reachability analysis from [Hartmann and

[4] One could also define it as $\mathcal{X} \subseteq \mathcal{Y}$, which is mathematically possible but does not match real experiments.

[6] Following sections might useful for a reader to understand the concept of funnel prediction. Section 2.1 shows the concepts of computing worst-case reachability analysis and section 3.3 gives the motivation to deform the areas by methods from machine learning.

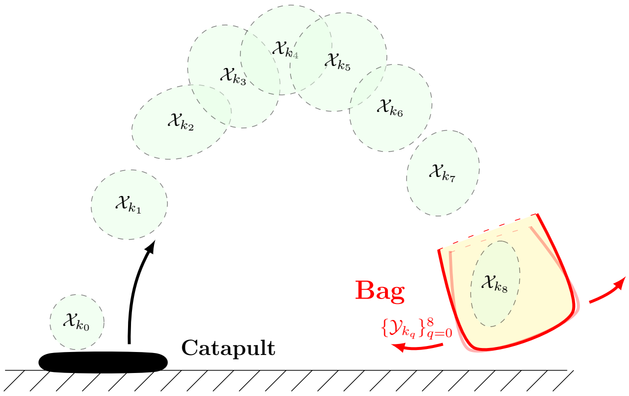


Figure 3.13.: Fictive example as an analogy for the set-based movement prediction[5] A catapult is giving an impulse to a very elastic ball. A person or robot is trying to capture the elastic ball by a bag.

Watzenig, 2019a]. Section 3.4.5 gives an example for a data-based approach and section 3.4.6 an example for a funnel prediction with real-world data.

3.4.1. Motivation for intelligent funnel prediction

This section aims to reduce the very conservative reachable sets to arrive at a less conservative prediction result (at the expense of increasing the risk) and thus give planning more freedom. Classical reachability analysis for dynamic systems relies on the physics (energy) of the system. [Liu et al., 2017] presents an application of reachability analysis for pedestrians to find possible future positions under consideration of the maximal velocity and acceleration. It also gives an extension to incorporate traffic rules[7]. For a deterministic system (vehicle), reachability analysis works quite well (with knowledge and modeling of the manipulated variables and limits). Choosing the right parameters is not trivial for pedestrians (unknown intent and high variability of the influencing factors). This section motivates to incorporate known historical information about pedestrians to predict the future unknown movements and find subsets of reachable sets. This section

[7] If a road user does not obey the rules, and there is an accident, then at least the fault does not lie entirely with the autonomous system.

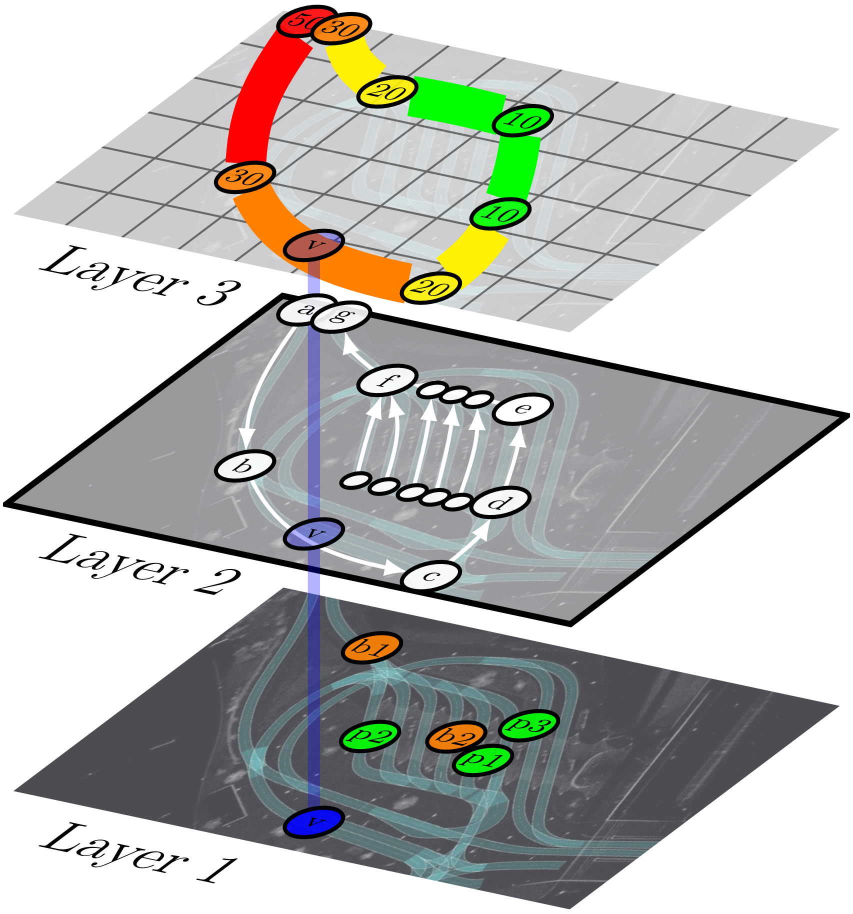


Figure 3.14.: Example of an urban environment in Graz (Austria) for a bus station with an autonomous vehicle. Layer 1 shows the position of the autonomous vehicle (v). Exemplary, some pedestrians (p1-p3) and two busses (b1-b2) are visualized as nodes. The road network is shown in layer 2. The autonomous vehicle drives from a,b,c,d,e,f to g, and the actual position are between b and c. Around the nodes d, e, and f are multiple white nodes that give alternative routes if the busses block the first gateway. The maximum current velocity is 30 km/h for the vehicle, and it switches to 20 km/h for c to d and 10 km/h for d to f (compare layer 3).

proposes the idea to incorporate additional information (compare section 3.3) from the pedestrian and compute funnels (deformed sets over time). These funnels $\{\hat{\mathcal{X}}_{k_q}\}_{q=i}^{j}$ should capture the movements of the pedestrian:

$$\boldsymbol{x}_{k_q}^{p} \in \{\hat{\mathcal{X}}_{k_q}\}_{q=i}^{j} \; \forall \, k_q \in \mathbb{T} \tag{3.28}$$

We could incorporate following information for the funnel prediction:

- Historical information. Information directly before the computation of the funnel. An example [Hartmann and Watzenig, 2019a] gives the idea to predict the maximal velocity and acceleration before a specific situation. [Hartmann et al., 2018b] gives the idea to incorporate historical information based on a cell of a manifold.
- Pedestrian related information: Relevant information influencing the pedestrian (e.g. human body and personal factors, compare chapter B) for a better movement- and set prediction.
- Environment: Information from the (urban) environment to incorporate obstacles and semantic information (compare 3.5 and chapter B)

Historical information can easily be incorporated (this section) and also the urban environment (section 3.5). Pedestrian related infromation can indirectly be incorporated. This section proposes to use machine learning to compute the funnels $\{\hat{\mathcal{X}}_{k_q}\}_{q=i}^{j}$. We could start with homogenous sets $\{\mathcal{H}_{k_q}\}_{q=i}^{j}$ or direct with the reachable sets $\{\mathcal{R}_{k_q}\}_{q=i}^{j}$. Afterwards we could deform these sets to represent the funnel. Irrelevant state space (which is outside of the reachable sets) is ignored by using reachable sets[8]:

$$\hat{\mathcal{X}}_{k_q} \subseteq \mathcal{R}_{k_q} \; \forall \, k_q \in \mathbb{T} \tag{3.29}$$

Methods from differential geometry (e.g. group action) and machine learning are proposed to build or deform the sets. Figure 3.15a shows two illustrations with the green reachable sets and homogeneous sets (yellow

[8] One assumption that the reachability analysis is computed adequate and with correct parameters. In practice a very high success rate should be possible if we assume high maximal velocities and accelerations for the pedestrian. But on the other hand we will get reachable sets will get big areas.

color). Figure 3.15a shows a selection of velocity vectors for different timestamps (a selection of states inside the reachability sets leads to the velocity vectors). Near to the homogeneous sets there are multiple arrows starting from the same situation. The complexity of the dynamics would increase if the dynamics are influenced by the decisions of the pedestrian and the cognition. Figure 3.15b shows the yellow homogeneous sets $\{\mathcal{H}_{k_q}\}_{q=i}^{j}$, the green reachable sets $\{\mathcal{R}_{k_q}\}_{q=i}^{j}$. The approach from [Girard et al., 2006] was used to compute the reachable sets with zonotopes. The reachable sets $\mathcal{R}_{k_q} := \mathcal{Z}_{k_q}^{\mathcal{R}} = \{\boldsymbol{c}_{k_q}^{\mathcal{R}}, \mathcal{G}_{k_q}^{\mathcal{R}}\}$ and homogenous sets $\mathcal{H}_{k_q} := \mathcal{Z}_{k_q}^{\mathcal{H}} = \{\boldsymbol{c}_{k_q}^{\mathcal{H}}, \mathcal{G}_{k_q}^{\mathcal{H}}\}$ are represented by zonotopes. The homogeneous set (yellow) 3.15b are mapped over time with a matrix multiplication (red transformed sets):

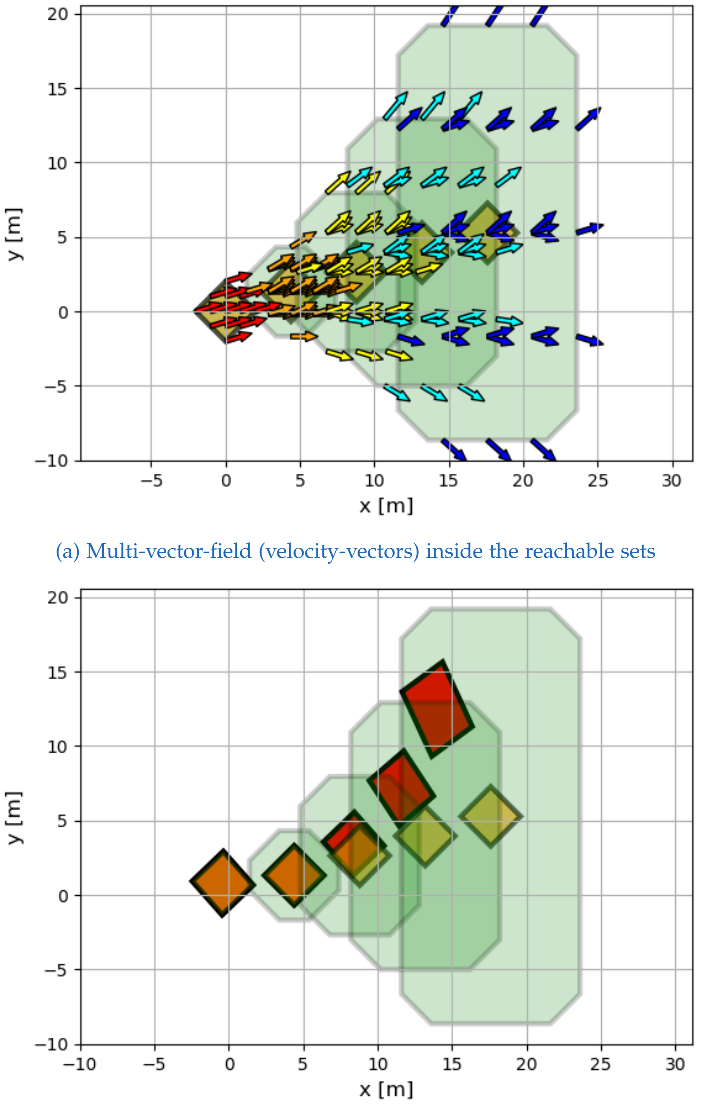
$$\hat{\mathcal{X}}_{k_q} = \boldsymbol{G} \cdot \mathcal{Z}_{k_q}^{\mathcal{H}} \tag{3.30}$$

The matrix $\boldsymbol{G} \in \mathbb{R}^{n \times n}$ is multiplied on the zonotope center and the zonotope generators. We could scale the generators by considering the eigenvalue problem $(\boldsymbol{G} \cdot \boldsymbol{g}_{k_q} = \lambda \cdot \boldsymbol{g}_{k_q})$. At the beginning of the five cycles in figure 3.15b the red and yellow sets overlap together. The transparency of the sets lead to the orange color.

3.4.2. Exemplary view on deforming sets and incorporation of prediction models

This section should show that there exist different approaches for set-based movement prediction.[9] Figure 3.16 shows examples how sets could be deformed and actions operate on these sets. A very common example of a group action is the $SO(3)$ with translation and rotation and the Minkowski sum (compare figure 3.16a and section A.2 for definitions). The sets could also represent by topological structures (e.g., graph theory). Each vertices could mapped to the next step (compare 3.16b) which lead to higher computational complexity. Different sets (e.g., zonotopes, ellipsoids, support vectors, polyhedrons) could lower computational complexity. Figure 3.16c

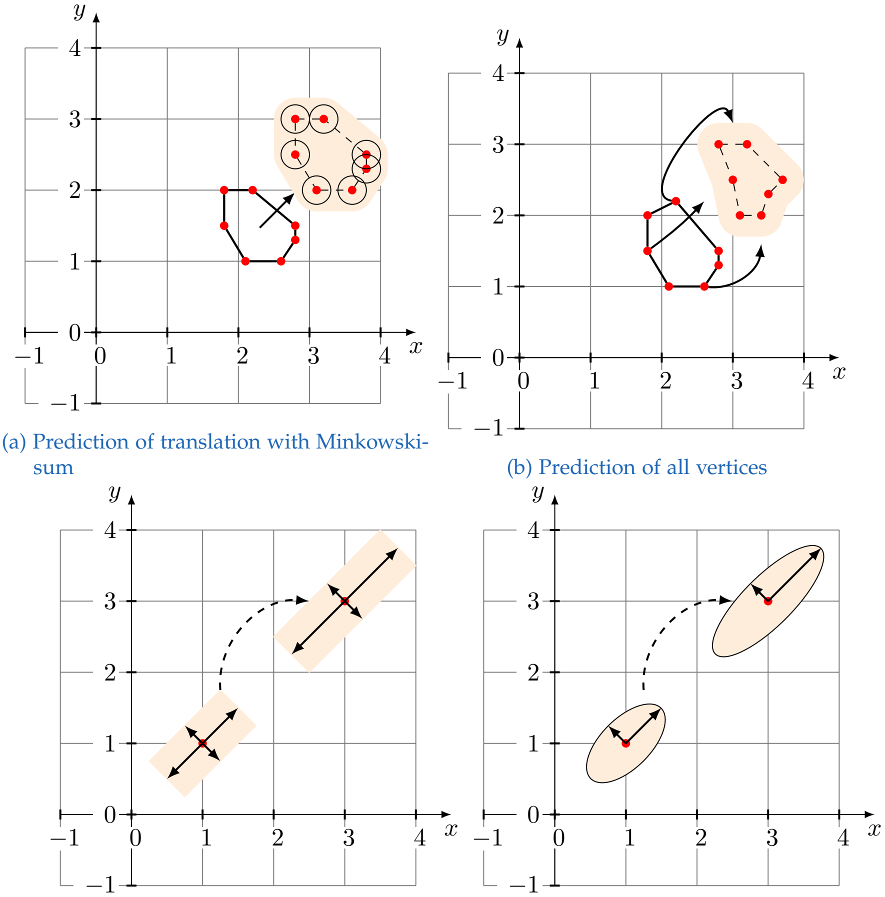
[9] A LSTM is chosen arbitrarily, and any other approach could be used for extrapolation. This section focuses on set-based prediction models.



(a) Multi-vector-field (velocity-vectors) inside the reachable sets

(b) Transformation of the homogeneous sets

Figure 3.15.: Multi-vector-field and deforming homogeneous sets



(a) Prediction of translation with Minkowski-sum

(b) Prediction of all vertices

(c) Prediction with support functions

(d) Prediction with ellipses

Figure 3.16.: Deforming sets with (group) actions

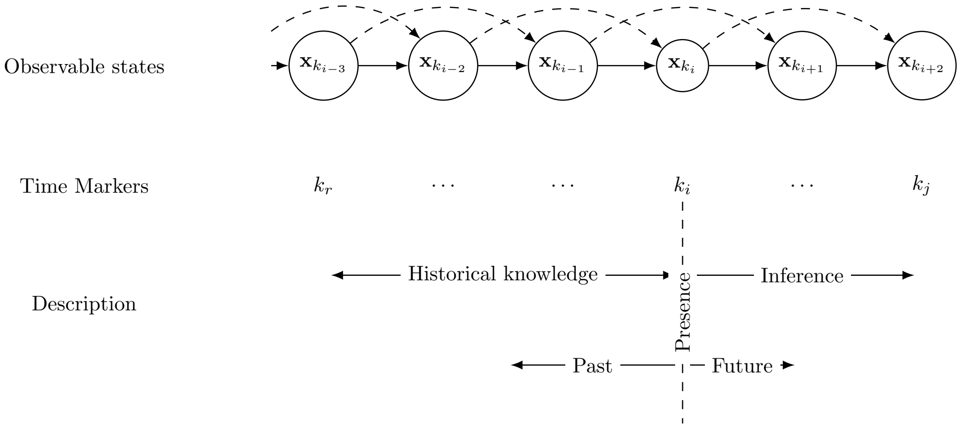


Figure 3.17.: First-order (without dashed arrows) or second-order (with dashed arrows) Markov-Chains with observable states. For this example we use r=i-3 and j=i+2 as a shorthand. The timesteps from k_r to k_i define historical timesteps for prediction to the k_j.

and figure 3.16d shows two similar mappings. Very common in machine learning are Markov-Chains like in figure 3.17, where the state is observable (white circle). Historical knowledge from $[k_r, k_{i-1}]$ can help to predict the future states by statistical inference. Dependent on the dynamic system, we have the First-order Markov property (solid lines), Second-order Markov property (with dashed arrows), or other dynamic principles (e.g., higher-order Markov properties). A stochastic process adds randomness to the state evolution so that the states are hidden, and the perturbed measurements in figure 3.18 are the only available information for the system. In technical systems, it might be sufficient to use a form of a Bayes filter where a single measurement in each iteration is sufficient for the next prediction (Kalman- or particle-filter). We use sets instead of a single particle to capture the future state of dynamic systems with aleatoric and epistemic uncertainty (nonreducible and reducible uncertainty by measurements). Figure 3.19 shows observable state sets changing over time.

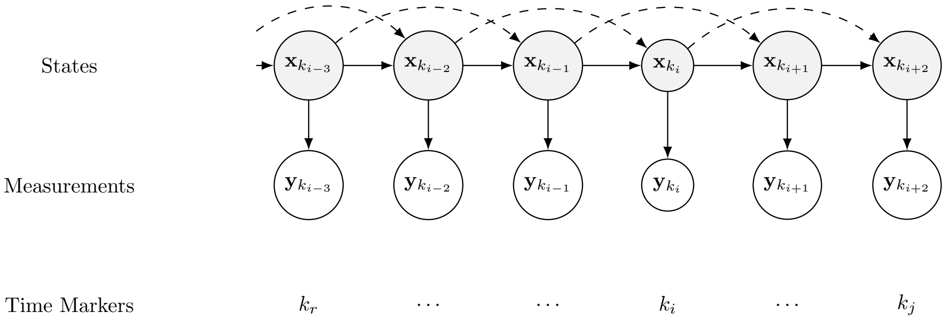


Figure 3.18.: Inference with sequential Markov-Chains. Hidden states (grey) and observable measurements (white)

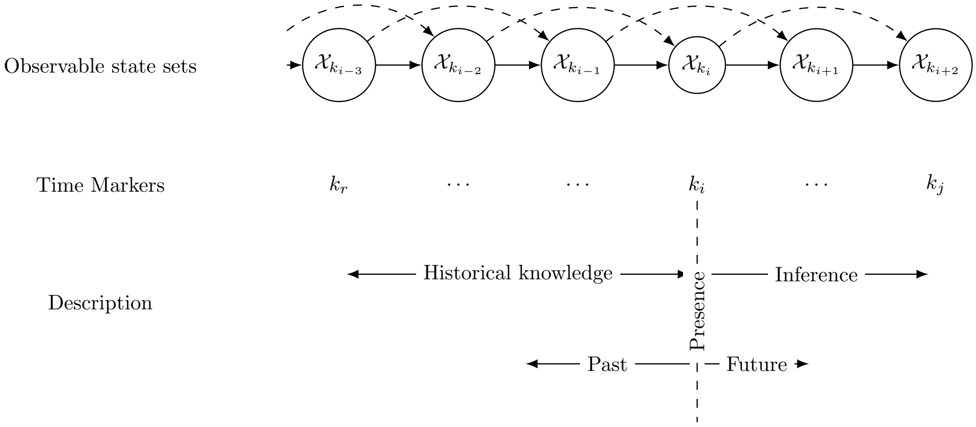


Figure 3.19.: Observable state sets

3.4.3. Data-based approaches

This section compares worst-case reachability analysis, conventional movement prediction and shows also the perspective to funnel prediction. Figure 3.20 shows a red trajectory with "captured" with reachability analysis and a funnel computed with machine learning. It also highlights the difficulties in computing adequate sets. We assume now a data-based approach with historical measurements

$$\mathcal{Y}_{r,i} := \{\mathbf{y}_{k_r}, \mathbf{y}_{k_{r+1}}, \ldots, \mathbf{y}_{k_{i-1}}, \mathbf{y}_{k_i}\} \tag{3.31}$$

as our knowledge about the system states $\mathcal{K}_{r,i} := \{\mathbf{x}_{k_r}, \mathbf{x}_{k_{r+1}}, \ldots, \mathbf{x}_{k_{i-1}}, \mathbf{x}_{k_i}\}$ without any system-model. For the timesteps we assume: $k_r < k_i < k_j,\ r, i, j \in \mathbb{N}$. The task is to predict next expected future measurement. For the single value prediction we compute the belief state:

$$\mathbf{b}_j := \hat{\mathbf{x}}_j = \mathbb{E}_{\mathbf{x}_{k_j} \sim \mathrm{P}(\mathbf{x}_{k_j} | \mathcal{K}_{r,i})}[\mathbf{f}(\mathbf{x}_{k_j})] \tag{3.32}$$

Instead of predicting N future states over a time horizon $\{\boldsymbol{x}_{k_i}, \boldsymbol{x}_{k_{i+1}} \ldots, \boldsymbol{x}_{k_j}\}$ this thesis give some models to predict future belief sets $\{\mathcal{B}_{k_i}, \mathcal{B}_{k_{i+1}} \ldots, \mathcal{B}_{k_j}\}$ of a person.

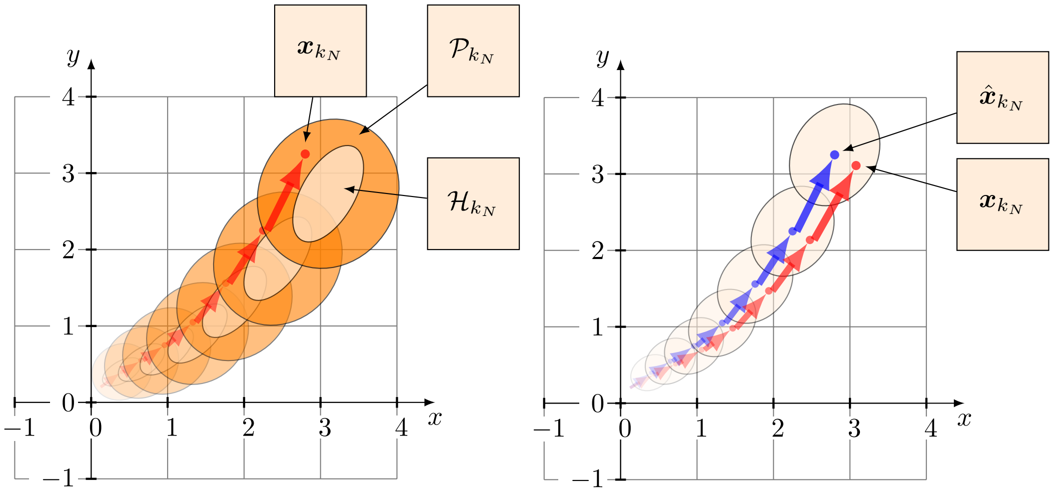
Definition 3.4.1. The belief set $\mathcal{B}_{k_j}$ is computed by a machine learning model at discrete time k_i. It predicts all belief-states at:

$$\mathcal{B}_{k_j} := \hat{\mathcal{X}}_{k_j} = \{\boldsymbol{b}_j | \mathbb{E}_{\boldsymbol{x}_{k_j} \sim \mathrm{P}(\boldsymbol{x}_{k_j} | \mathcal{K}_{r,i})}[\mathbf{f}(\boldsymbol{x}_{k_j})],\ k_r < k_i < k_j\} \tag{3.33}$$

If someone tries to model the human locomotion as a dynamic system, at least three different types of states are necessary to model the dynamic system - the measurable body state $\boldsymbol{s}_{k_i}$ and the unknown cognitive state $\boldsymbol{c}_{k_i}$ of the brain activity and perception $\boldsymbol{o}_{k_i}$ (the perception $\boldsymbol{o}_{k_i}$ is part of section B.1. It represents the information that a person perceives by sense-organs. [10].

$$\boldsymbol{x}^h_{k_i} := \begin{bmatrix} \boldsymbol{c}_{k_i} \\ \boldsymbol{s}_{k_i} \end{bmatrix} \begin{array}{c} \text{hidden} \\ \text{observable}^{11} \end{array} \tag{3.34}$$

[10] Section B.1 gives also details on the causal relationship between the body state $\boldsymbol{s}_{k_i}$, the cognitive state $\boldsymbol{c}_{k_i}$, the perceived information $\boldsymbol{o}_{k_i}$. The perceived information $\boldsymbol{o}_{k_i}$ might be some measurements made by the human. We could describe a state as a measurable quantity describing the evolution of the system



(a) Capturing the real trajectory (red) with Reachability Analysis $\{\mathcal{R}_{k_i}\}_{i=1}^{N}$

(b) Movement prediction of the real trajectory (red) with the predicted trajectory.

Figure 3.20.: Reachability analysis and machine learning

The body state $\boldsymbol{s}_{k_i}$ is sometimes observable by sensors (if obstacles do not hide the body). The cognitive state $\boldsymbol{c}_{k_i}$ is often unknown[12] (chapter B discusses also measurement technologies which are capable to measure brain activity. There exist also wearable brain sensors, compare section B.3). Also, the perception of a person is not observable. Another problem is the high dimensionality of the vector (curse of dimensionality). Suppose all voxels of the brain images and body states are captured in a single vector and substantial simplifications of pedestrian models. To model and measure the cognitive state is not trivial, and chapter B in the appendix gives more details on this topic. The hidden state of the intention and brain state (under the assumption of free will) might lead to an uncertain prediction, which might be fatal in the case of an autonomous vehicle. We assume there is a nonreducible (aleatoric-) uncertainty. One assumption (with good reasons for that) is that it is not possible to predict the future states $\boldsymbol{x}_{k_j}$ with absolute certainty if the intention of the person is unknown. The person has the ability and possibility to move freely. This thesis aims to use state set prediction (funnel-prediction) algorithms to use them for

[12] That is the reason why many movement-prediction algorithms assume a fixed intention.

human locomotion. Figure 3.21 shows the general approach with figure 3.22 as a fictive example. The concrete algorithms with an extended evaluation are presented in the next chapter 4. The actual situation (actually observed trajectories of the pedestrian) is used to predict the funnels with machine learning and evaluated when the funnel is capturing the trajectory like in figure 3.22. Methods to deform sets (e.g., SO(3), Minkowski-sum or vertices prediction) with different set types (e.g., support function, ellipsoids, or reachable sets with homogenous sets can be a reference). Figure 3.22 shows the black dotted trajectory with multiple arrows. The funnels $\{\hat{\mathcal{X}}_{k_q}\}_{q=i}^{j}$ is capturing single points of the trajectory.

3.4.4. Adaptive reachability analysis

This section discusses approaches for adaptive reachability analysis (adapted from [Hartmann and Watzenig, 2019a]) instead of only using set-prediction approaches (which rely only on historical data). In adaptive reachability analysis (compare figure 3.23) we use the approach from worst-case reachability analysis (compare section 2.1) and predict their meta-parameters (e.g. maximal values for jerk, acceleration, velocity), which changes the form of the control input set $\mathcal{U}(t)$. The vehicle set $\mathcal{X}^v(t)$ cannot intersect with the pedestrian set $\mathcal{X}^h(t)$ or any other object or area which might lead to collisions. One can use the physical limitations of the human body to compute reachable sets. The papers of [Liu et al., 2017,Hartmann and Watzenig, 2019a] were starting points for the following approach (compare figure 3.24). Instead of using only the maximal acceleration and velocity for conservative motion planning, the algorithm predicts the jerk in each situation for riskier motion planning. Gaussian processes based on current observations predict the jerk. Pedestrians can have typical movement patterns constrained by a static environment in a fixed environment. The movements on the street are different compared to the sidewalk[13]. We use three models, where the maximal parameters (e.g., jerk, acceleration, velocity) are set and predicted for each situation. The jerk-constrained model leads to the reachable sets $\mathcal{R}_{e,k_q}$ under consideration of all three input-variables $\boldsymbol{v}(t)$ (velocity), $\boldsymbol{a}(t)$

[13] A database could provide information about physical limitations as prior knowledge for the human movement prediction with reachability analysis

(acceleration) and jerk $\boldsymbol{e}(t)$. Only the maximal value for the jerk constraints the control-input $\mathcal{U}_e$:

$$\begin{aligned} \dot{p}_x(t) = v_x(t),\ \dot{p}_y(t) = v_y(t), \\ \dot{v}_x(t) = a_x(t),\ \dot{v}_y(t) = a_y(t) \\ \dot{a}_x(t) = e_x(t),\ \dot{a}_y(t) = e_y(t) \\ \mathcal{U}_e = \{(e_x(t), e_y(t)) \in \mathbb{R} \times \mathbb{R} | e_x(t)^2 + e_y(t)^2 \leq e_{\text{max}}^2\} \end{aligned} \tag{3.35}$$

The acceleration-constrained model leads to the reachable sets $\mathcal{R}_{a,k_q}$ under consideration of all two input-variables $\boldsymbol{v}(t)$ (velocity) and $\boldsymbol{a}(t)$ (acceleration). Only the maximal value for the acceleration constrains the control-input $\mathcal{U}_a$:

$$\begin{aligned} \dot{p}_x(t) = v_x(t),\ \dot{p}_y(t) = v_y(t), \\ \dot{v}_x(t) = a_x(t),\ \dot{v}_y(t) = a_y(t) \\ \mathcal{U}_a = \{(a_x(t), a_y(t)) \in \mathbb{R} \times \mathbb{R} | a_x(t)^2 + a_y(t)^2 \leq a_{\text{max}}^2\} \end{aligned} \tag{3.36}$$

The velocity-constrained model leads to the reachable sets $\mathcal{R}_{v,k_q}$ under consideration of the input-variable $\boldsymbol{v}(t)$ (velocity). Only the maximal value for the velocity constrains the control-input $\mathcal{U}_v$:

$$\begin{aligned} \dot{p}_x(t) = v_x(t),\ \dot{p}_y(t) = v_y(t), \\ \mathcal{U}_v = \{(v_x(t), v_y(t)) \in \mathbb{R} \times \mathbb{R} | (v_x(t))^2 + (v_y(t))^2 \leq (v_{\text{max}})^2\} \end{aligned} \tag{3.37}$$

$\mathcal{U}_e$, $\mathcal{U}_a$, $\mathcal{U}_v$ are three dimensional circles (balls in multiple dimensions) $\mathcal{B}_{u_{\text{max}}} = \{\boldsymbol{u} \in \mathbb{R}^2 |\ ||\boldsymbol{u}||_2 \leq u_{\text{max}}\}$. The following formula can approximate the control input space defined as a circle:

$$\begin{aligned} \mathcal{B}_{u_{\text{max}}} \approx \mathcal{P}_{u_{\text{max}}} := \{(u_x(t), u_y(t)) \in \mathbb{R} \times \mathbb{R} | \ldots \\ u_x = u_{\text{max}} \cdot \sin(\frac{2 \cdot \pi \cdot n}{n_P}), \ldots \\ u_y = u_{\text{max}} \cdot \cos(\frac{2 \cdot \pi \cdot n}{n_P}),\ n = 0, \ldots, n_P - 1\} \end{aligned} \tag{3.38}$$

The approximation quality can be adjusted by the parameter n_P. The result is a polygon. Over the complete time horizon $k_q \in k_i, k_{i+1} \ldots, k_j$ the reachable sets of the jerk-, acceleration- and velocity-constrained model are computed

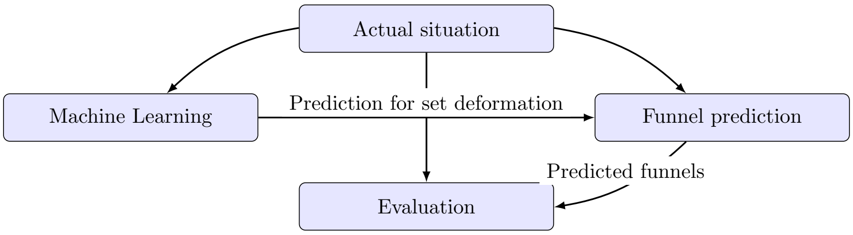


Figure 3.21.: Concept for funnel prediction

$\mathcal{R}_{e,k_q}, \mathcal{R}_{a,k_q}, \mathcal{R}_{v,k_q}$. Like [Liu et al., 2017] we can combine the reachable sets of the acceleration- and velocity-constrained model and compute the worst-case reachable sets $\mathcal{R}_{k_q}$ at time-step k_q:

$$\mathcal{R}_{k_q} := \mathcal{R}_{a,k_q} \cap \mathcal{R}_{v,k_q} \ \forall \ k_q \in \mathbb{T} \tag{3.39}$$

The combination of all constrained models lead to prediction sets $\hat{\mathcal{X}}_{k_q}$:

$$\hat{\mathcal{X}}_{k_q} := \mathcal{R}_{e,k_q} \cap \mathcal{R}_{a,k_q} \cap \mathcal{R}_{v,k_q} \ \forall \ k_q \in \mathbb{T} \tag{3.40}$$

The prediction sets are again a subset of the worst-case reachable sets $\hat{\mathcal{X}}_{k_q} \subseteq \mathcal{R}_{k_q} \ \forall \ k_q \in \mathbb{T}$. The advantage of this approach is that the maximal parameters $e_{\max}, a_{\max}, v_{\max}$ have to be estimated by the current observed trajectory $\mathcal{T}$. In [Hartmann and Watzenig, 2019a]we showed how to predict the change of only the maximal values $\boldsymbol{e}(t)$ with two Gaussian processes with a similar approach like [Ellis et al., 2009].

$$p(\Delta \boldsymbol{e}_{k+1} | x_k, y_k, \mathcal{T}) \approx \mathcal{N}(\bar{\mathbf{f}}, \mathbb{V}) \tag{3.41}$$

$\bar{f}$ is the predicted mean value with the variance $\mathbb{V}$. One can compute $\bar{f}$ and $\mathbb{V}$ from the following posterior distribution obtained by the conditioning of the observed data:

$$f_* | \mathbf{f} \sim \mathcal{N}(\underbrace{K_* K^{-1} \mathbf{f}}_{\bar{f}}, \underbrace{k_{**} - K_* K^{-1} K_*^T}_{\mathbb{V}}) \tag{3.42}$$

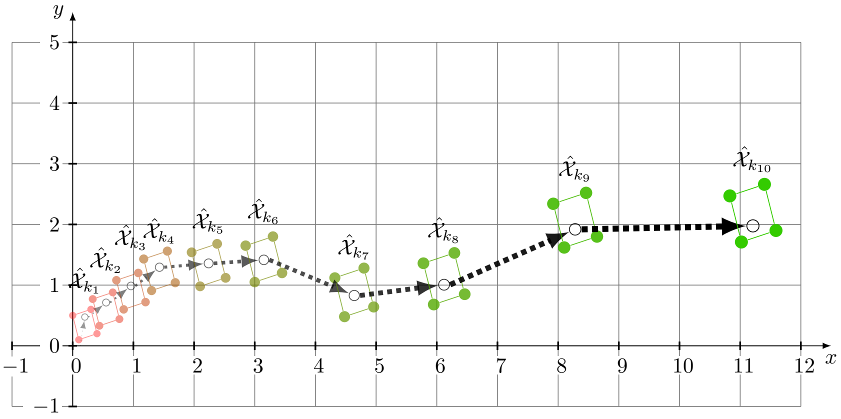


Figure 3.22.: Example of set based movement prediction with $\{\hat{\mathcal{X}}_{k_q}\}_{q=i}^{j}$[14]

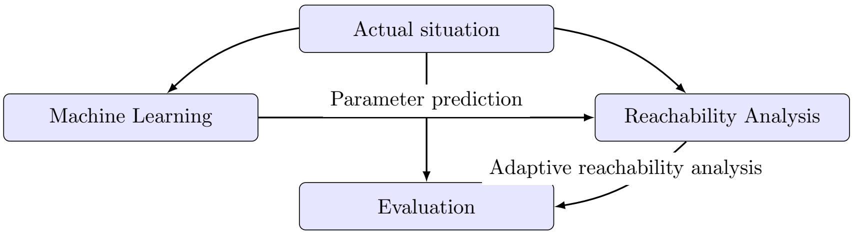


Figure 3.23.: Concept for adaptive reachability analysis

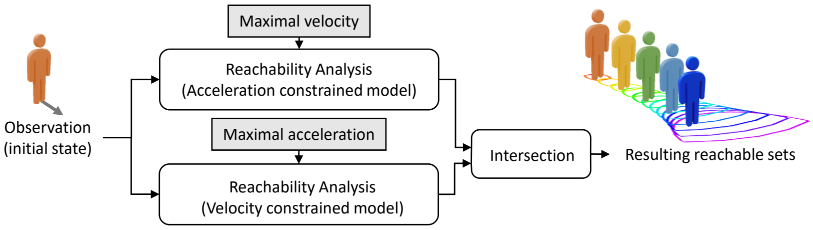


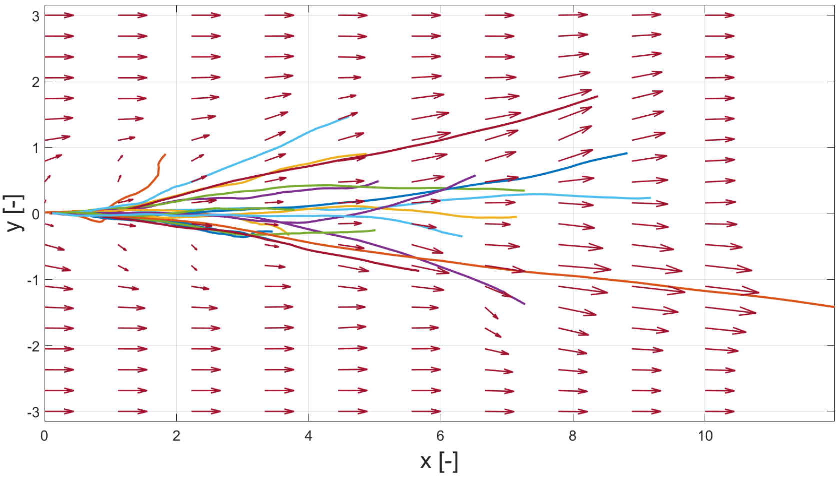
Figure 3.24.: Worst-case-reachability analysis as a reference for adaptive reachability analysis. A prediction method in [Hartmann and Watzenig, 2019a,Schratter et al., 2019] predicts the maximal values for the jerk, acceleration and velocity (Picture source: [Schratter et al., 2019])

3.4.5. Example with a data-based approach

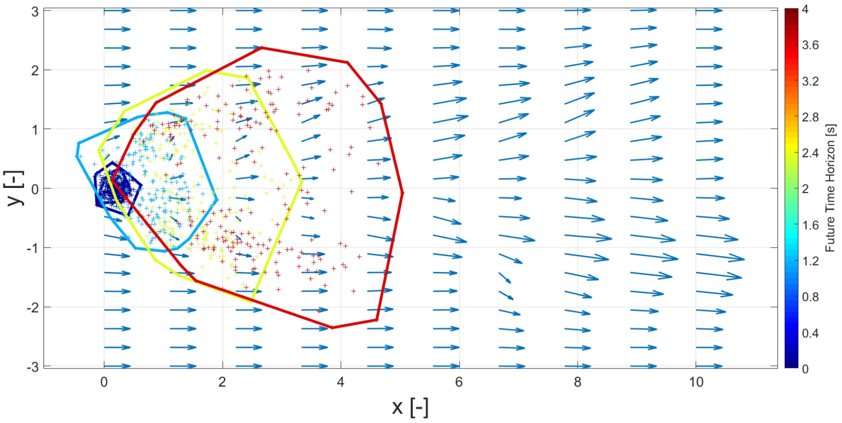
This section shows an example of funnel prediction. It is a counter-example to adaptive reachability analysis. This example shows the flexibility of predicting all the vertices with (multiple) predictors. On the other hand, is many vertices have to be propagated. This prediction complexity might be problematic for the computational complexity. Adaptive reachability analysis has only three parameters that have to be computed, which is a strong advantage. This section should show that we can compute the sets $\{\hat{\mathcal{X}}_{k_q}\}_{q=i}^{j}$ only by considering historical trajectories and the vertices of the set. It might be helpful to understand the approach [Hartmann et al., 2018b], where each local environment (represented here as the nearest vector in the vector field) can have an independent predictor, which is also interesting in section 3.5. We introduce the concept by a simple but very inefficient algorithm (there might be improvements in computational efficiency when the particles are predicted). As a starting point we use the stochastic process [De Nicolao et al., 2007b] (formula 2.31-2.38) to generate synthetic trajectories of pedestrians (compare figure 3.25a). The approach proposed in [Ellis et al., 2009] helps to compute the vector-field $\mathsf{Vect}(\mathcal{M})$ (compare formula A.1.6). Figure 3.25b uses this vector-field with Monte Carlo simulations to compute four different polygons (hull of the particles in red, yellow, cyan, and blue color) for 4 seconds. If a set of N initial particles $\mathcal{P}_{k_0} := \{\boldsymbol{p}_{q,k_0} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)\}_{q=1}^{N}$ are placed around the origin $\boldsymbol{\mu} \approx \begin{bmatrix} 0 & 0 \end{bmatrix}^T$. The closure of the particles are defined as convex hull $\overline{\mathcal{P}_{k_0}} := \mathsf{CH}(\mathcal{P}_{k_0})$. With Monte-Carlo simulation, all the particles are propagated in the direction of the nearest vector of the vector field. Afterwards we compute again the convex hull $\overline{\mathcal{P}_{k_1}}$. The particles are propagated with the Monte-Carlo simulation and with the nearest vector.

3.4.6. Example of a funnel prediction with real pedestrian movements

This section shows an approach for funnel prediction with real-world data. In figure 3.26 trajectories were predicted with the Long short-term memory



(a) Stochastic process with Gaussian processes (computed vectorfield) [Ellis et al., 2009]



(b) Monte-Carlo Simulation and resulting sets

Figure 3.25.: Stochastic process and data-based belief sets

(LSTM) neural networks. These LSTMs are used to predict the time-series signals. Only the x, y-positions from the last historical time-stemp k_r to the actual time-stemp k_i ($\{k_r, k_{r+1} \ldots, k_i\}$) are considered by the LSTMs as training dataset. The future timesteps are test set $\{k_i, k_{i+1} \ldots, k_j\}$. In the example we used the first $2.5s$ as a training dataset and the time interval between $2.5 - 3.5s$ as the test set. Like [Ellis et al., 2009] we predicted the position change in x- and y-direction. Another LSTM was used to predict the uncertainty measure (hull around the red and the cyan line in figure 3.26a and coloured rectangles in figure 3.26b). The example shows how machine learning (in this case a arbitrarily chosen deep learning model LSTM to extrapolate data over time) was used to predict a funnel $\{\hat{\mathcal{X}}_{k_q}\}_{q \in \mathcal{I}_{\hat{\mathcal{X}}}}$ (compare section 3.4) with real-world data.

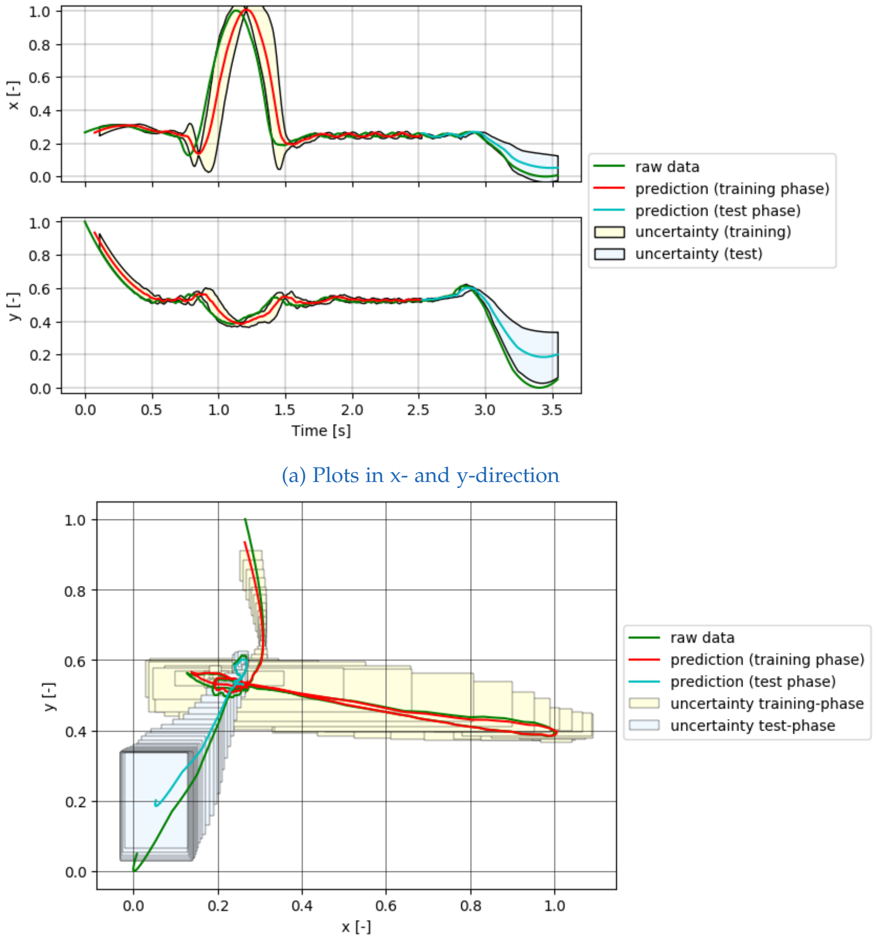
3.5. Pedestrians walking on Manifolds

This section 3.5 focuses on the environment and is a cut to the previous sections. It should provide the mathematical fundamentals to model movements. [15] of pedestrians and also a theoretical outlook for the human locomotion research community. Subsection 3.5.2 compares mathematical concepts for human locomotion. Subsection 3.5.3 focuses on manifolds and differential geometry to describe the urban environment. Subsection 3.5.4 focuses on an excursus to optimal control on manifolds. Subsection 3.5.5 discusses an use case for the university campus of the University of Technology Graz in Austria. Subsection 3.5.6 gives another example with Stanford Pedestrian dataset from [Robicquet et al., 2016b].

3.5.1. Motivation for topological spaces

The motivation for using topological spaces and differential geometry is not that intuitive. Differential geometry is mathematically demanding. There are

[15] Synthetic movement data could produce critical and unusual test-data without considering real test-persons.



(a) Plots in x- and y-direction

(b) Movement prediction with rectangular funnel

Figure 3.26.: Movement prediction with the Stanford Drone dataset [Robicquet et al., 2016a]

also many prerequisites to understanding human locomotion: In appendix-chapter B it is mentioned that the brain has an enormous capacity and complexity. Each region of the brain has different tasks and responsibilities for cognitive processing. The motor and sensory cortex are responsible for performing actions and sensing body movements[16]. It is a highly complex processing network also with different semantic tasks [Huth et al., 2016a]. Nowadays, many machine learning models are used to learn and understand the patterns of brain activity and the functionality of each region (e.g., [Naselaris et al., 2011a]) In chapter B we also present a cognitive MDP (compare figure 3.57) model developed out from the literature survey where we distinguish between the cognitive state of the brain $\boldsymbol{c}(t) \in \mathcal{C}$ (activity levels of human brain measuring devices) and the state of the human body $\boldsymbol{s}(t)$ (body skeleton). We summarize these two different states in a single vector representing the state of the human and compare formula 3.34. We can represent the cognitive state and the brain state in a Euclidean coordinate system and abstract the brain activity to human-understandable concept (compare figure 3.57 and 3.58) a kind of meta state $\boldsymbol{m}(t) \in \mathcal{M}$. This meta state should translate the complex brain activity to what the subject is thinking and doing (e.g., the person walks to the supermarket and hears music). The current research community in Neuroinformatics are trying to encode and decode brain activity ([Naselaris et al., 2011a]) with a machine learning model to represent a function $f : \mathcal{C} \rightarrow \mathcal{M}$ and also the inverse function. The abstraction of the human can also think of a tree as an obstacle, and creativity and intelligence could also relate it to other concepts (e.g., treehouse). These concepts are semantically connected, and it can be shown that there exist different brain activity patterns for different concepts (e.g., [Huth et al., 2016a]). The realization that "a person is walking, the other person is dancing" could differentiate in the current implementation (a wide variety in gaits and dance moves is possible). If we would change the problem to "a person is walking on the sidewalk," we could assume that the pace of a person is something around the walking speed. The brain activity patterns in walking on a sidewalk have repeated cycles ([Brantley et al., 2018]) which could be described in a certain policy $\pi^h(t)$. The policy of the

[16] We assume a healthy person for the example. There must be some assumption that the spinal cord, the whole nervous system, and other functionality or human systems are functioning well.

human brain is trying to fulfill a certain task with a certain goal. Each action could of a human be described in the form of a certain controller policy. This policy exists to fulfill tasks by getting feedback from the perception and sensing the body movements (compare figure B.3 for a deep learning model to learn to walk). The problem in analyzing these concepts has to be cast into mathematical concepts. The following ideas assume that the brain activity patterns in a certain semantic urban state environment $\mathcal{X}$ are the same. The controller set $\mathcal{X}$ (e.g., sidewalk) is defined as a controller performing a force on a particle. We propose a "Pedestrian in the Loop" test environment in 3.7 for further basic research. Figure 3.30 shows the task to model human behavior. The sensing, cognition, and action cycle have to be modeled to compute the trajectories on a topological space. We will use the concept of manifolds as a theoretical concept to structure complex urban environments. Someone unfamiliar with differential geometry might ask, why not use clothoid (or other analytical functions) to model curved curves. Clothoids are a special case of differential geometry ([Schulz, 2014]), and a point mass that is driven with force as a function of time is also a case that fits into the concept. We could use a cell with a dynamic system (e.g. $\dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)$ or probabilistic descriptions) and implement it into a cell $\mathcal{X}_i$ of the manifold. We could change the coordinate system or the dynamic system in the next cell. What still fits well into the concept is a piece of work that describes that work from the subgoal concept cites ikeda2013modeling, but they used spatial graphs to describe the movements. Also, [Vasquez, 2010a,Kitani et al., 2012] present the movements of pedestrians in semantic networks, which strengthen the application of differential geometry in human locomotion. Figure 3.27 shows the path on a manifold. Figure 3.28 shows a curved trajectory which is represented from a manifold. In a fictive example, a person must complete three sub-goals of a task. The subject has to walk to the first pylon (orange circle with number 1) and then has to run (subgoal 3) to the second pylon, which is the fictive target (circle with number 2). The path is labeled with colored and alphabetic letters. Our brain can abstract some targets and also imagine curved trajectories for locomotion. Figure 3.29 shows three pictures[17] from a real scenario where pedestrians might

[17] Pictures are from `https://www.flickr.com/photos/infomastern/28662408986/`, `https://www.flickr.com/photos/infomastern/34486288613/`, `https://www.flickr.com/photos/infomastern/28076991283/`. Downloaded at 4. August). More pictures of the

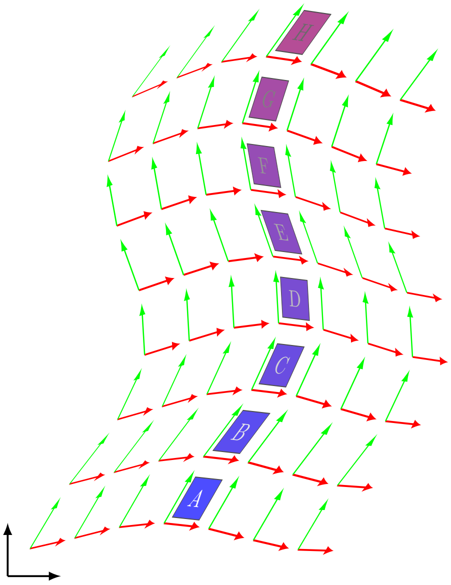


Figure 3.27.: Transformed path (A, B,...,H) in a manifold

walk this kind of curved path. Instead of the pylon, there are white masts, and the bridge has a circular structure and layout, so that is named circle bridge.

3.5.2. Mathematical concepts for human locomotion

This section compares different mathematical concepts for the urban environment and a motivation to use concepts for differential geometry for modeling movements of humans. The modelling of the urban environment as a manifold has the advantage to structure large and complex urban environments and also with higher accuracy than spatial graphs. On the other hand it is mathematically more complex. This section should give a kind of visual introduction to these concepts. For data inference and modelling pedestrian movements, it is essential to represent the environment

bridge and layout at https://miesarch.com/work/3393.

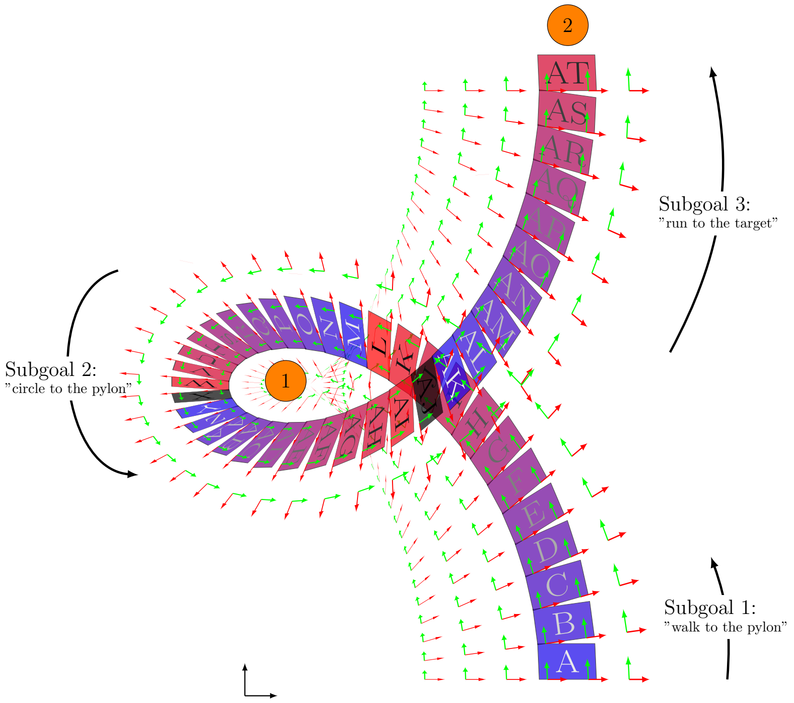


Figure 3.28.: Curved path (A to AT) with different subgoals sub-goal

(a) Walking pedestrians

(b) Bird view perspective

(c) Rounded barrier

Figure 3.29.: Circle bridge (Cirkelbroen) in Copenhagen (Images and ownership from Susanne Nilsson, with licence: CC BY-SA 2.0)

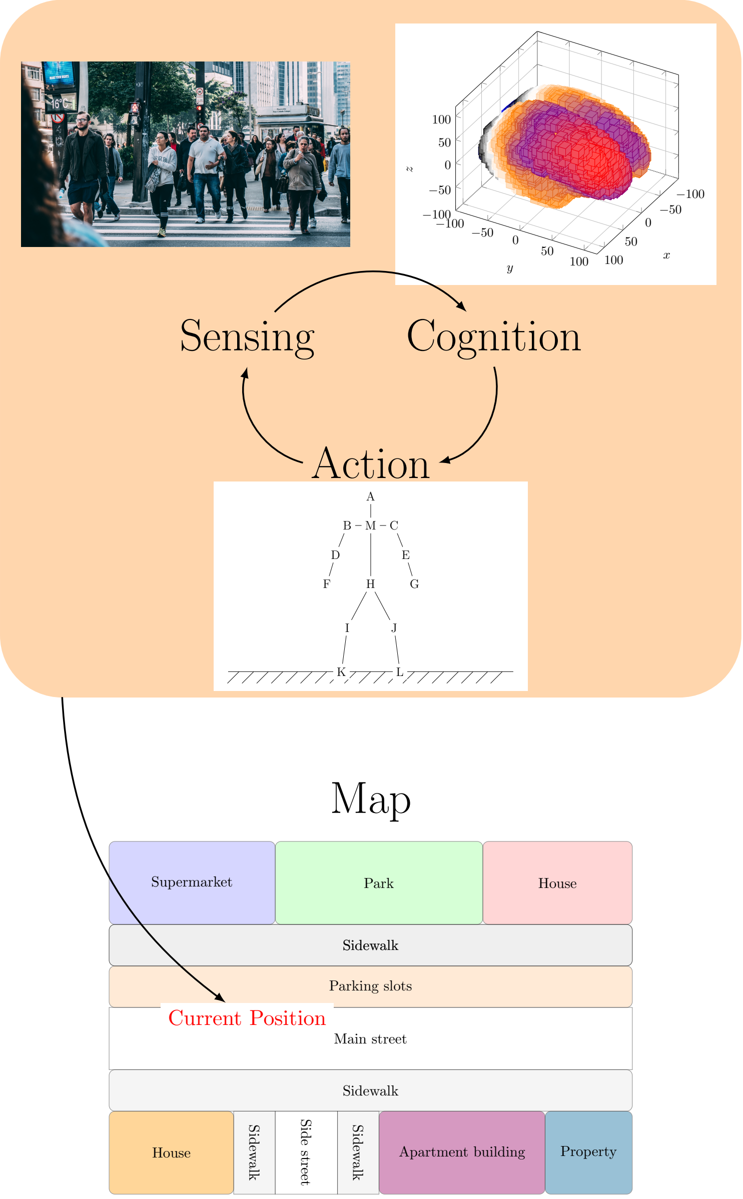
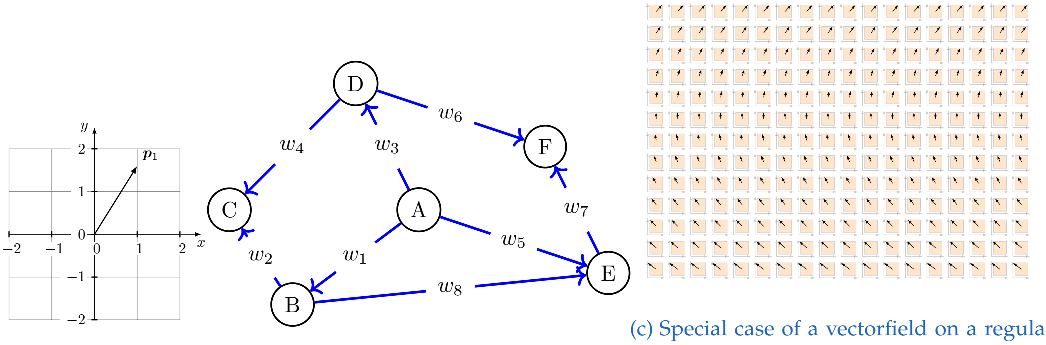


Figure 3.30.: Human locomotion



(a) Euclidean space

(b) Graph

(c) Special case of a vectorfield on a regular grid

Figure 3.31.: Euclidean space, graph representation and vector-field (manifold)

and data adequately. Figure 3.31 shows three relevant representations for this thesis[18]. All three representations (Euclidean space, spatial graph and vector-field) have in common that the state space $\mathcal{X}$ have a structure in form of a topology $\mathcal{T}$. An urban environment can be described in form of a topological space $(\mathcal{X}, \mathcal{T})$ (compare section A.1). The whole state space $\mathcal{X}$ is represented by a set of disjoint cells $\mathcal{X}_i$ (subsets of $\mathcal{X}_i \subseteq \mathcal{X} \ \forall\ i \in \mathcal{I}_\mathcal{X}$):

$$\mathcal{X} := \bigcup_{i \in \mathcal{I}_\mathcal{X}} \mathcal{X}_i \tag{3.43}$$

$$\mathcal{X}_i \cap \mathcal{X}_j = \emptyset \ \forall\ i, j \in \mathcal{I}_\mathcal{X}, i \neq j \tag{3.44}$$

The first representation (compare figure 3.31a) is an Euclidean coordinate system. Orthogonal basis vectors span the spatial space (conventional vector space) of the Euclidean coordinate system and with the special case. Another representation is shown in figure 3.31b with a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and vertices-set $\mathcal{V} = \{v_1, \dots, v_{N_\mathcal{V}}\} = \{v_q\}_{q \in \mathcal{I}^\mathcal{V}}$ and the set with edges $\mathcal{E} = \{e_1, \dots, e_{N_\mathcal{V}}\}$. The label set of the nodes is $\mathcal{I}_\mathcal{V} = \{1, \dots, N_\mathcal{V}\}$. In combination with a coordinate system (each node gets a position) one could describe movements in a spatial graph (e.g. approach in [Vasquez, 2010a]). We need therefore a mapping function for the node label set $\mathcal{I}_\mathcal{V}$ to the cell label set $\mathcal{I}_\mathcal{V}$:

$$f : \mathcal{I}_\mathcal{V} \rightarrow \mathcal{I}_\mathcal{X} \tag{3.45}$$

[18] There are also other representations possible.

A single position $\boldsymbol{p}_i \in \mathcal{X}_i$ from one cell can represent the whole set as a node $\boldsymbol{v}_i$ (from graph theory) for all labels $i \in I$ [19] Figure 3.31c shows the third representation of spatial problems, a manifold with the corresponding vector-field[20]. Figure 3.31c shows many coordinate systems (transparent grey arrows), which are only representing a local environment. The black arrows show the direction of the velocity vector on a transparent orange patch. This illustration is unusual because it is not common to visualize many coordinate systems. It might help an interested reader who is not familiar with differential geometry to understand the concepts (compare A.1). Graph and vector fields have in common that their mathematical objects are structurally connected via a topology (nodes in graphs via edges and vector fields often on a regular grid). A manifold is also a kind of mesh.

Definition 3.5.1. Definition from [Asarin et al., 2003]. A mesh $\mathcal{M}$ of the set $\mathcal{X}$ is a finite set of full-dimensional convex polyhedra in $\mathbb{R}^n$, called cells, satisfying the following conditions:

- The union of all cells $\bigcup_q \mathcal{X}_q = \mathcal{X}$
- If $\mathcal{X}_q$ and $\mathcal{X}_r$ are cells with non-empty intersection, then their intersection lies within the boundaries of both; we say that $\mathcal{X}_q$ and $\mathcal{X}_r$ are adjacent and we denote their intersection by $\partial(\mathcal{X}_q, \mathcal{X}_r)$

This section compared different mathematical expressions to represent movements of pedestrians: Euclidean space, graph, vector-field, Manifold (regular- and transformed grid). In the next subsection, we focus on manifolds to describe human locomotion.

[19] With this approach, it is possible to represent the spatial graph, compute the adjacency matrix $\boldsymbol{A}$, and use linear algebra and graph theory concepts. From that representation it is common to predict spatial movements with Hidden Markov Models [Bishop, 2006,Vasquez, 2010a].

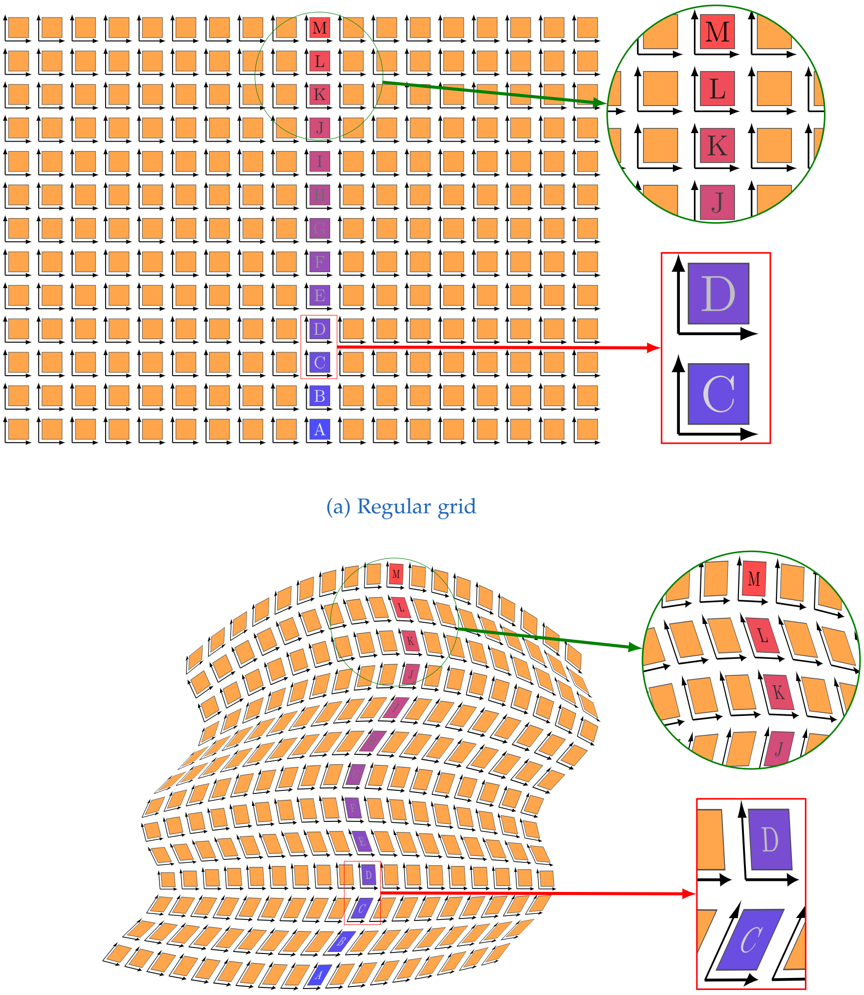
[20] A manifold can have a vector-field. For mathematical correct definitions of a vector-field and manifold compare section A.1)

3.5.3. Manifolds from differential geometry describing the underground for human locomotion

A general way to describe topological (spatial) problems is manifolds from differential geometry. Each patch (spatial set $\mathcal{X}_i$) of the regular and nonregular grid has its euclidean coordinate system. Figure 3.32a and figure 3.32b show colored patches with letters (A to M) representing a path. The path is on a smooth manifold (see Appendix for definition of a smooth manifold). The manifold in Figure 3.32b is transformed with figure 3.32a as a reference. The transformation of the spatial space offers the possibility to describe straight lines as curved paths and vice versa. A simple example is that pedestrians could walk in a city center around a fountain, whose trajectory could be described easily by polar coordinates. Polar coordinates are a particular case for differential geometry. Pedestrians can walk in rather complex urban environments which have a topological structure. A networked system can describe the walkable areas for pedestrians.

3.5.4. Optimal control on manifolds (Interconnected Euclidean Systems)

This section should give a short digression to an example with optimal control. The main goal is to emphasize that human locomotion is often referenced to an imaginary coordinate system in the subconscious, changing over time and the (urban) environment. The cognitive processing of a pedestrian plays an important role in performing movements. A human can change his coordinate system by perceiving objects and spatial distance measurements and adapting to new situations. The optimal control in interconnected coordinate systems (Manifold) should introduce a technical and concrete example for modeling pedestrian movements. This idea will find a place in other technical algorithms. Figure 3.33 shows an example with four interconnected sets $\{\mathcal{X}_i\}_{i\in\mathcal{I}_{\mathcal{X}}}$ (with four colours). The yellow set is adjacent to all other sets and coordinate systems. We can use optimal control (similarly to section 3.2) to compute a trajectory in interconnected Euclidean systems. The target state $\boldsymbol{x}_{k_j}^{A}$ in coordinate system A with basis



(a) Regular grid

(b) Nonregular transformed grid

Figure 3.32.: Transformation of a regular grid (manifold) to new manifold representation

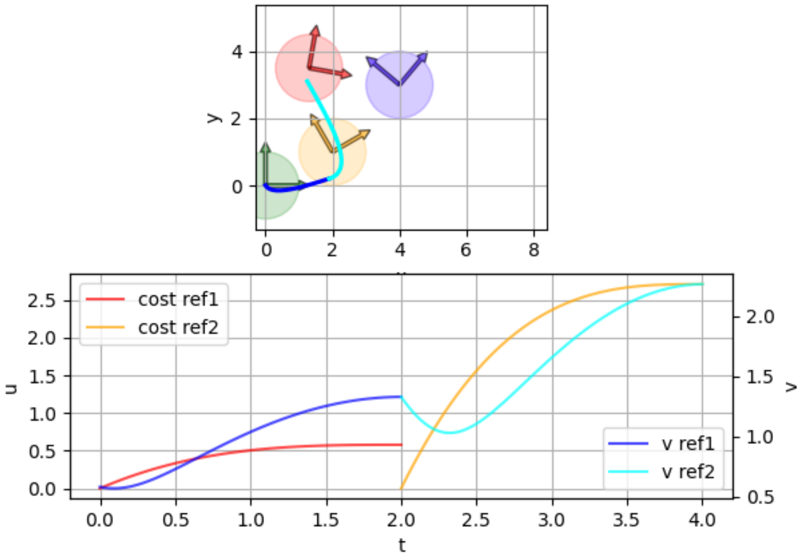


Figure 3.33.: Optimal control on a manifold

vectors (e_1^A, e_2^A) is the initial state $x_{k_{j+1}}^B$ in the new coordinate system B (e_1^B, e_2^B). The concepts of coordinate system change can simply computed with the concepts of tensor calculus (compare section A.1). From [0, 2s], the optimal control system brings the agent from the green coordinate system to the yellow coordinate system (blue trajectory). The bottom subfigure shows the velocity (blue curve) and the cost function (red curve). The cyan trajectory shows the optimal control from the yellow coordinate system to the red coordinate system with the cost function and the velocity. This example should show that the controller is having a change in the reference coordinate system. This example is analogous to how a pedestrian changes his reference system by changing his perspective and perception.

3.5.5. Use case for the University Campus

This section shows some previous works representing movements on spatial graphs. Spatial graphs could be placed on a topological space $(\mathcal{X}, \mathcal{T})$. Figure

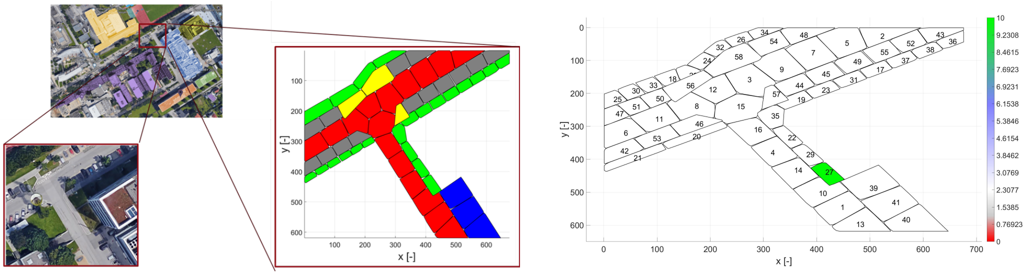
3.34a shows an intersection of the Technical University Graz in Austria described as a topological space. The red patches represent the street, light green the sidewalk, and grey parking slots. With reinforcement learning, one could model an optimal policy of an agent trying to maximize the expected reward. [21]. The patch with label 27 gets the highest reward[22] (green color) in Figure 3.34b. Value iteration from reinforcement learning runs on a topological space with a reward on patch with label 27 (compare figure 3.35a). Figure 3.35b shows the algorithm's resulting optimal paths from different initial positions. This example shows how a virtual agent acts optimally with Markov Decision Process (MDP). This example also shows the drawbacks of spatial graphs for representing the movements of pedestrians. The brain activity of the person is not modeled. It assumes that pedestrians follow optimal paths and have perfect rational behavior. We show in the appendix a theoretical extension in the form of a cognitive MDP from section B.1. However, there is further research necessary to model brain activity adequately. Therefore the thesis proposes a test environment for the research of the human decision making in the presence of autonomous vehicles (compare section 3.7). Figure 3.36 shows further approaches from [Hartmann and Watzenig, 2019b] in a larger scale. This example also shows the importance of high-precision maps and the importance of getting a relationship between the decision-making of pedestrians in different areas.

3.5.6. Use-case: Stanford dataset

An example of a real-world dataset also confirms the motivation to describe the movements with topological spaces. Figure 3.37a shows all the initial- and end-positions of each subject in the Stanford drone dataset [Robicquet et al., 2016b]. The Stanford drone dataset [Robicquet et al., 2016b] is a dataset where pedestrians' movements are tracked by flying drones observing spatial movements. In Figure 3.37b one has selected a certain individual

[21] There are many reasons why optimal movements of the pedestrian are often realistic. The behavior of pedestrian is not rational and optimized (compare section B.1)

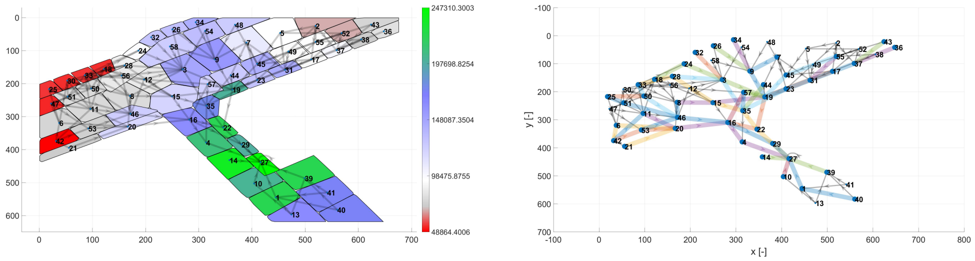
[22] The label 27 is randomly picked and shows only that the agent of the Markov Decision Process is generating an optimal trajectory on the topological space.



(a) Intersection of the University in Graz

(b) Reward on patch 27

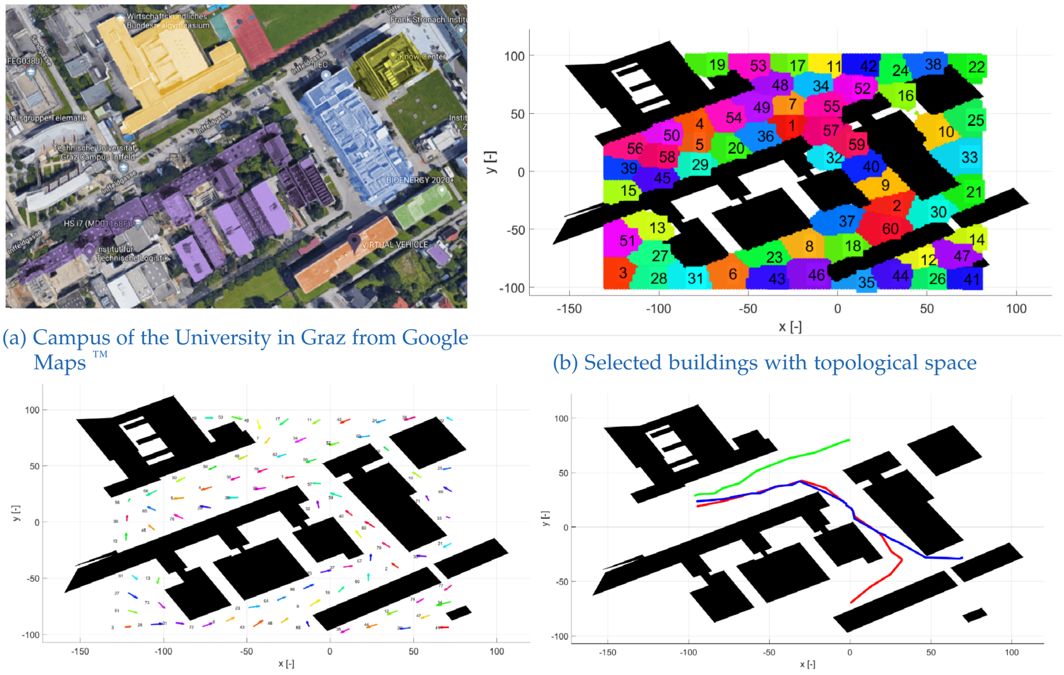
Figure 3.34.: Intersection in Graz. Value iteration in classical non-regular grid and resulting paths



(a) Value Iteration after some iterations

(b) Several trajectories

Figure 3.35.: Intersection in Graz. Value iteration in classical non-regular grid and resulting paths



(a) Campus of the University in Graz from Google Maps ™

(b) Selected buildings with topological space

(c) Vectorfield

(d) Sample trajectories generated from the vector-field

Figure 3.36.: Campus of the University in Graz [Hartmann and Watzenig, 2019b]

who appears in the left top corner. Figure 3.37b shows all initial positions of each individuals who appeared also in the top left corner as blue dots. The end positions are shown with orange dots, three clusters (computed with K-means-clustering), and red ellipses. The orange points in the middle are outliers, so there are only two groups for these specific cases.

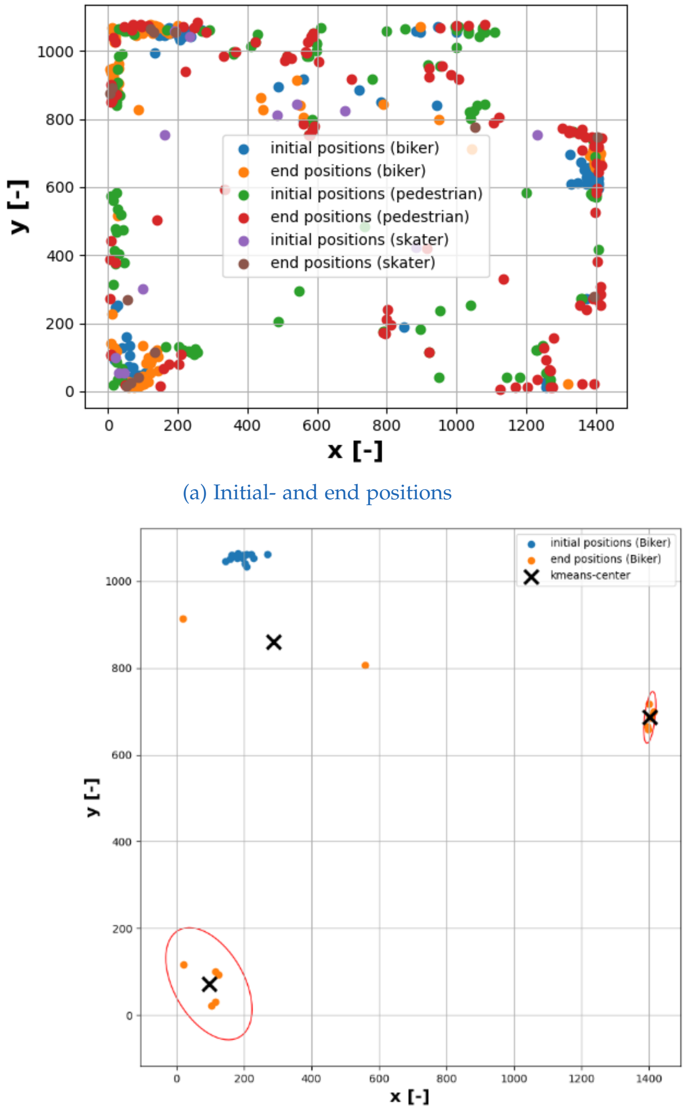
Figure 3.38b shows a selection of trajectories with two groups (blue and red trajectories). As in the beginning of this section described we could group the whole state space $\mathcal{X}$ in disjoint connected cells $\mathcal{X}_i$ (here with voronoi-diagram compare figure 3.38a). The algorithm used in Figure 3.38b computes the velocities. Principal component Analysis (PCA) computed an orthogonal basis vectors for each cell $\mathcal{X}_i$, which are visualized as black orthogonal lines in Figure 3.38b. This example shows topological spaces could help to model many real-world trajectories, which is considered in the next sections.

The advantages of describing the urban environment a manifold:

- Generalization of Euclidean systems, spatial graphs, and vector fields.
- Representing complex movements: It is possible to deform the state space $\mathcal{X}$ and describe complex movements. A particular case of differential geometry is polar coordinates which are often used for circular movements. Manifolds could adequately describe lanelets which are very common in the automotive industry.
- Semantics: Each cell $\mathcal{X}_i$ can have a semantic meaning (sidewalk, street) where the pedestrian might change his behavior or on a transition[23].
- Data-inference and computations: The generalization of Euclidean systems, spatial graphs, and vector fields offers many possibilities for predicting and modeling human movements.
- Combination with other concepts: The combination of set theory and reachability analysis goes well.

The disadvantages are:

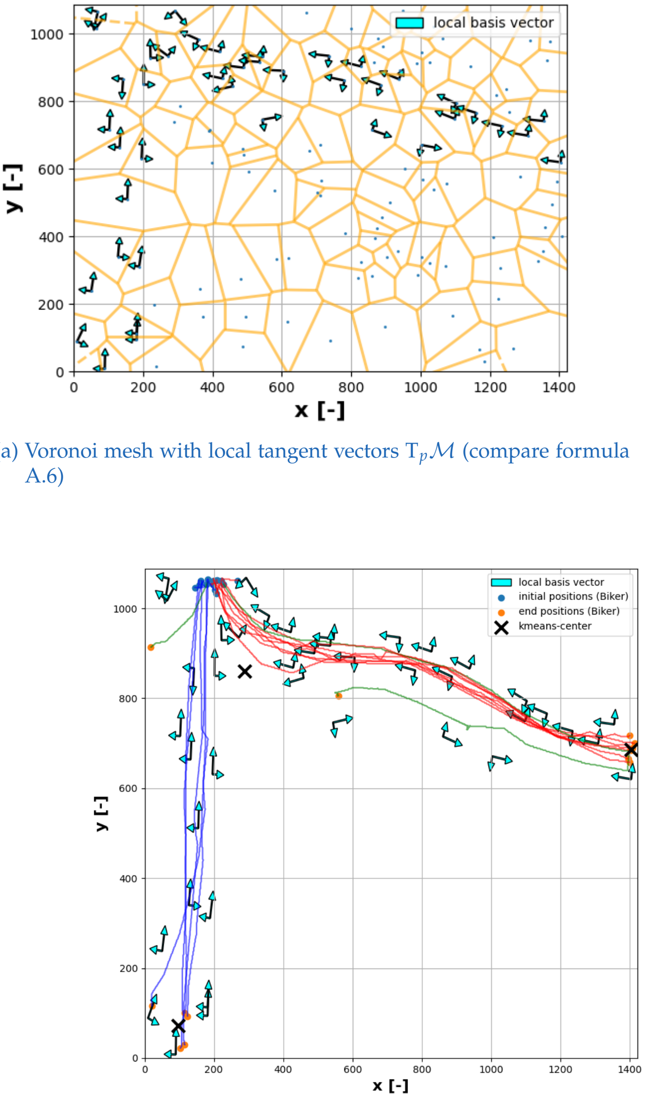
[23] In the example on the University campus (section 3.5.5 it does not change its behavior. However, this is a problem in modeling the behavior in a computer model. The computer model does not capture all possible behaviors and relies on simplifications and assumptions. In section 3.6 we will present more details.



(a) Initial- and end positions

(b) Selection of initial- and end-positions

Figure 3.37.: Starting- and end-positions of pedestrians from the Stanford drone dataset [Robicquet et al., 2016b]



(a) Voronoi mesh with local tangent vectors $T_p\mathcal{M}$ (compare formula A.6)

(b) Two selected groups (blue and red) with local basis vectors

Figure 3.38.: Tangency space with voronoi diagram

- Mathematical complexity: The field of differential geometry is vast and mathematical complex [24]. With tensor calculus, it is possible to represent changes in coordinate systems.
- Not all information on the human locomotion of pedestrians is currently explored. The decision-mapping of pedestrians could be mapped to the manifold cells, and further research on the decision-making is needed.
- Lack of algorithms: The mathematical complexity of manifolds does not offer many algorithms that are successfully applied in movement prediction.

Subsection 3.5.7 gives the motivation to generate synthetic movement data.

3.5.7. Motivation to generate synthetic movement data with concepts from causal inference, differential geometry and set-based methods

This section should give some perspectives to model pedestrian movements for testing autonomous vehicles and combine the ideas of causal reasoning, differential geometry and set-based methods from section 2.1, 3.5 and 3.6. The reasons to use causal inference to model movements and incorporate interventions and intention changes into the framework for generating synthetic movement data. Differential geometry is to describe the movements in a sub-goal framework [Ikeda et al., 2013] and using the structure of the environment. Set-based methods could incorporate physical limitations. As mentioned in section 2.2.1 there exist datasets from observational-studies for real pedestrians like [Robicquet et al., 2016a]. In each situation in observing human locomotion by an autonomous vehicle we have a lack of knowledge, personal- and situational aspects which are specific for the certain situation. The intention, personal aspects (e.g. current feelings and emotions, thoughts, some environmental influences) are not captured in a dataset. A observational study has some bottlenecks, which make a randomized controlled experiment necessary. In this thesis a randomized controlled experiment is proposed in section 3.7. The advantages of synthetic movement data are:

[24] In many engineering disciplines, it is often not in the curriculum

- Synthetic movement data might ease the generation of training and test data and analyze the performance results to generalize prediction algorithms to new situations. It is also necessary to consider that, on the other hand, synthetic movement data is a simplification of the body- and decision dynamics of the pedestrian. For a depth analysis on each factors influencing the pedestrian and prediction a pedestrian-in-the-loop environment might give more insight (see section 3.7).
- Untypical and dangerous behavior of agents in simulated environments in collision avoidance maneuvers with autonomous vehicles could improve safety. The amount of improvements also depends on the quality of the simulations, which is not trivial for realistic modeling of human behavior.
- The generation of the huge amount of movement data is a benefit compared to the expense for the measurements in observational studies. It is also not always possible to observe pedestrians (e.g., day and night).

Some disadvantages are:

- Physical bias: In general, it is easier to measure the physical movements of the human body than the cognitive processes inside the human brain. Therefore it is more convenient to model the body movements in system models. Detailed processes for the dynamics for the intention-change of a person are not well researched yet. The modeling of exclusively physical aspects of human locomotion (e.g., without considering brain functionality) might lead to a substantial simplification. The complexity of the human brain is very complex. There is research ongoing for the simulation of human processes with supercomputers (e.g., Human brain project [Salles et al., 2019]). However, for the application in the automotive industry, simple models (simple physical rules) are often used to test collision avoidance algorithms with pedestrians.
- Simplifications and realistic behavior: The consideration of all complexities in the dynamics of the physical body and the cognitive aspects of thinking is very complex. There exist numerical software projects like [Delp et al., 2007] to approximately human body dynamics. However, we consider substantial simplifications for the human body and its intentions.

- There is the need for (high-quality) maps for movements in realistic environments to produce synthetic movement data for natural environments. The quality of the maps also depends on the modeling of the synthetic movement data. Suppose the perception of a pedestrian is modeled. There might be the requirement to model the environment with the photorealistic or semantic structure depending on the cognitive abilities of the virtual agent. Some substantial simplifications only consider the perception of the environment and the decision-making by an agent.

We want to incorporate causal models and figure 3.39 gives some reasons to use causal models. As described in [Peters et al., 2017] we could use observations from an experiment:

$$(x_1, y_1), \ldots, (x_n, y_n) \tag{3.46}$$

These observations (x_q, y_q) $(q \in \mathbb{N})$ come from random variables (X_q, Y_q), with input-variable X_q and output-variable Y_q:

$$(X_1, Y_1), \ldots, (X_n, Y_n) \tag{3.47}$$

In dynamic settings $k_i, k_j \in \mathbb{T}$, we might consider time t (and there discrete values) with the variables:

$$(x_{k_i}, y_{k_i}), \ldots, (x_{k_j}, y_{k_j}) \tag{3.48}$$

With the time-dependent random variables:

$$(X_{k_i}, Y_{k_i}), \ldots, (Y_{k_j}, Y_{k_j}) \tag{3.49}$$

A very essential assumption of doing experiments and classical statistical learning is that the datasets are independent and identically distributed (i.i.d.). Afterwards we could use classical techniques from statistical-learning, machine learning and deep learning to compute the expectation of the output given the input to form a regression:

$$\mathbf{f}(x) = \mathbb{E}[Y|X = x] \tag{3.50}$$

The function $\mathbf{f} : \mathcal{X} \to \mathcal{Y}$ maps $\mathcal{X}$ to $\mathcal{Y}$, which are both metric spaces. We assume that the density $p_{X,Y}$ of $P_{X,Y}$ are static and independent and identically distributed (i.i.d.). In statistical learning, we learn the density from

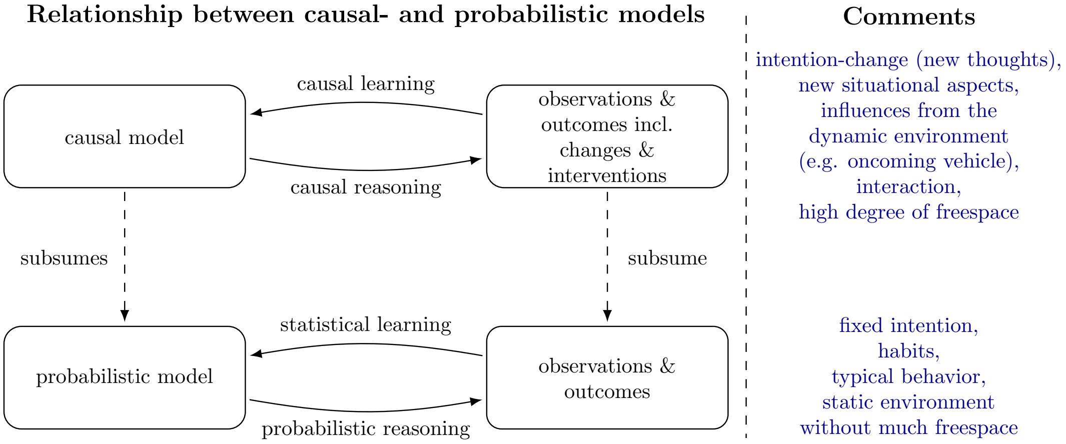

Relationship between causal- and probabilistic models
Comments
causal model
causal learning
causal reasoning
observations & outcomes incl. changes & interventions
subsumes
subsume
probabilistic model
statistical learning
probabilistic reasoning
observations & outcomes
intention-change (new thoughts), new situational aspects, influences from the dynamic environment (e.g. oncoming vehicle), interaction, high degree of freespace
fixed intention, habits, typical behavior, static environment without much freespace

Figure 3.39.: This figure shows causal models and causal reasoning and relates to probabilistic models and observations and outcomes. The structure comes from the book [Peters et al., 2017] and is extended by some of the usages of automotive scenarios (blue). Using causal models in situations with changing probabilities (e.g., intention-change or external influences).

observations and outcomes from an experiment. In probabilistic reasoning, we generate data from a density. There are different challenges in classical machine learning to find suitable classes and functions $\mathbf{f}(x)$. The prediction results from classical machine learning algorithms might vary quantitatively on different real-observed human movement data. Especially if there are new situational aspects (e.g., change of weather conditions), new intentions (e.g., U-turn), irrational behavior, interaction, and other new factors causing the pedestrian's behavior. Causal reasoning allows us to analyze the effect of interventions or distribution changes [Peters et al., 2017]. The inverse problem of causal learning (structure learning and causal discovery) to learning from observations with interventions is a new field but not the focus of the thesis. Therefore we want to propose set-based prediction methods to include the effect of interventions in set-based methods. Set-based methods and reachability analysis could also incorporate worst-case scenarios or situations where the dynamics of a pedestrian are restricted to some physical conditions. The description as a manifold could incorporate the idea presented in section 3.5. It describes the movements with subgoals [Ikeda et al., 2013] and uses the structure of the urban environment (different semantic environments, e.g., sidewalk, street, park) into a mathematical framework.

3.5.8. Quantification of computational complexity

Table 3.1 shows the mathematical complexity of some operations necessary for the reachable sets, and this table will be considered in the next chapter for the complexity of some algorithms. It shows the Minkowski sum of two zonotopes. Also, the matrix multiplication to a zonotope and other mathematical operations. In a single-vector space we have a coordinate system with orthogonal basis-vectors. In this coordinate system we could have a set of a subject $\mathcal{X}_{k_i}$ or a represented as a particle $\boldsymbol{x}_{k_i}$. We could ensure a movement by using a mathematical operation. If we have a time horizon $\{k_i, k_{i+1}, \ldots, k_j\}$ and p steps. The system might be described by a nonlinear system $\boldsymbol{x}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$, but the trajectory with the position of a person $\boldsymbol{p}_{k_i} = [p_{x,k_i}, p_{y,k_i}]^T$. $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}$ $\boldsymbol{A} \in \mathbb{R}^{n \times n}, \boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$ has a computational cost of $\mathrm{O}(n^2)$. If we have a recursive application of the matrix-vector operation in the time horizon $\{k_i, k_{i+1}, \ldots, k_j\}$ we have $\mathrm{O}(p \cdot n^2)$ (iterated over p steps).

$$\begin{bmatrix} p_{x,k_{i+1}} \\ p_{y,k_{i+1}} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}^{\theta,t}} \begin{bmatrix} p_{x,k_i} \\ p_{y,k_i} \\ 1 \end{bmatrix} \tag{3.51}$$

If we have a set $\mathcal{X}_{k_i}$ with the closure $\mathrm{CL}(\mathcal{X}_{k_i})$. We can have a single-vector-space or multiple vector spaces. In the case of multiple vector spaces, we would have a set of coordinate systems. The set traverses over multiple coordinate systems. Figure 3.40 shows a dynamic system with rotation matrix, reachable sets $\mathcal{R}$ and corresponding single trajectory. If an additional mathematical operation operates on the whole reachable set, it can be translated and rotated. Also, the Minkowski sum or difference could be operating on these sets. Figure 3.41a shows a simulation with the Julia Reach package [Bogomolov et al., 2019] on a simple street. Instead of computing from one state a trajectory, we could compute the reachable sets. With a different solver, one could compute a single trajectory figure 3.41b.

Computational complexity of some mathematical operations	
Scalar-vector-multiplication $a \cdot \boldsymbol{x} = \boldsymbol{b}$, $a \in \mathbb{R}, \boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$	O(n)
Vector-vector multiplication $\boldsymbol{a}^T \cdot \boldsymbol{b} = c$, $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n, \boldsymbol{c} \in \mathbb{R}$	O(mp)
Vector-Matrix multiplication $\boldsymbol{a}^T \cdot \boldsymbol{B} = \boldsymbol{c}^T$, $\boldsymbol{a} \in \mathbb{R}^m, \boldsymbol{B} \in \mathbb{R}^{m \times p}, \boldsymbol{c} \in \mathbb{R}^p$	O(mp)
Matrix-Matrix multiplication $\boldsymbol{A} \cdot \boldsymbol{B} = \boldsymbol{C}$, $\boldsymbol{A} \in \mathbb{R}^{n \times m}, \boldsymbol{B} \in \mathbb{R}^{m \times p}$	O(nmp)
$\boldsymbol{b} + \mathcal{Z}_1 = \{\boldsymbol{b} + \boldsymbol{c}, \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_k\}\}$	O($2mn(k+1)$)
$\mathcal{Z}_1 \oplus \mathcal{Z}_2 = \{\boldsymbol{M} \cdot \boldsymbol{c}, \{\boldsymbol{M}\boldsymbol{v}_1, \dots, \boldsymbol{M}\boldsymbol{v}_k\}\}$	O(n) ([Bogomolov et al., 2019])
$\boldsymbol{M} \cdot \mathcal{Z}_1 = \{\boldsymbol{M}\boldsymbol{c}, \{\boldsymbol{M}\boldsymbol{v}_1, \dots, \boldsymbol{M}\boldsymbol{v}_k\}\}$	O($2mn(k+1)$) ([Bogomolov et al., 2019])

Table 3.1.: Mathematical complexity for some operations for computing reachable sets

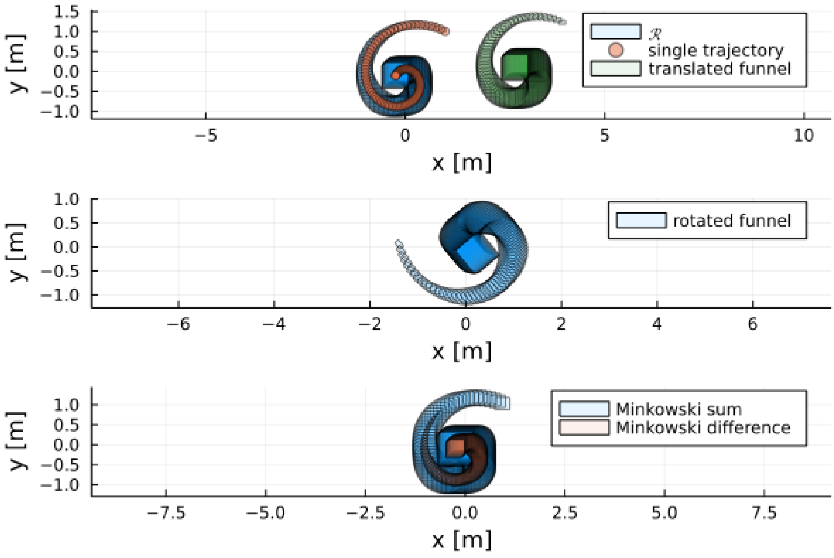


Figure 3.40.: Spiral reach set (programmed with [Bogomolov et al., 2019])

3.5.9. Decision making of pedestrians

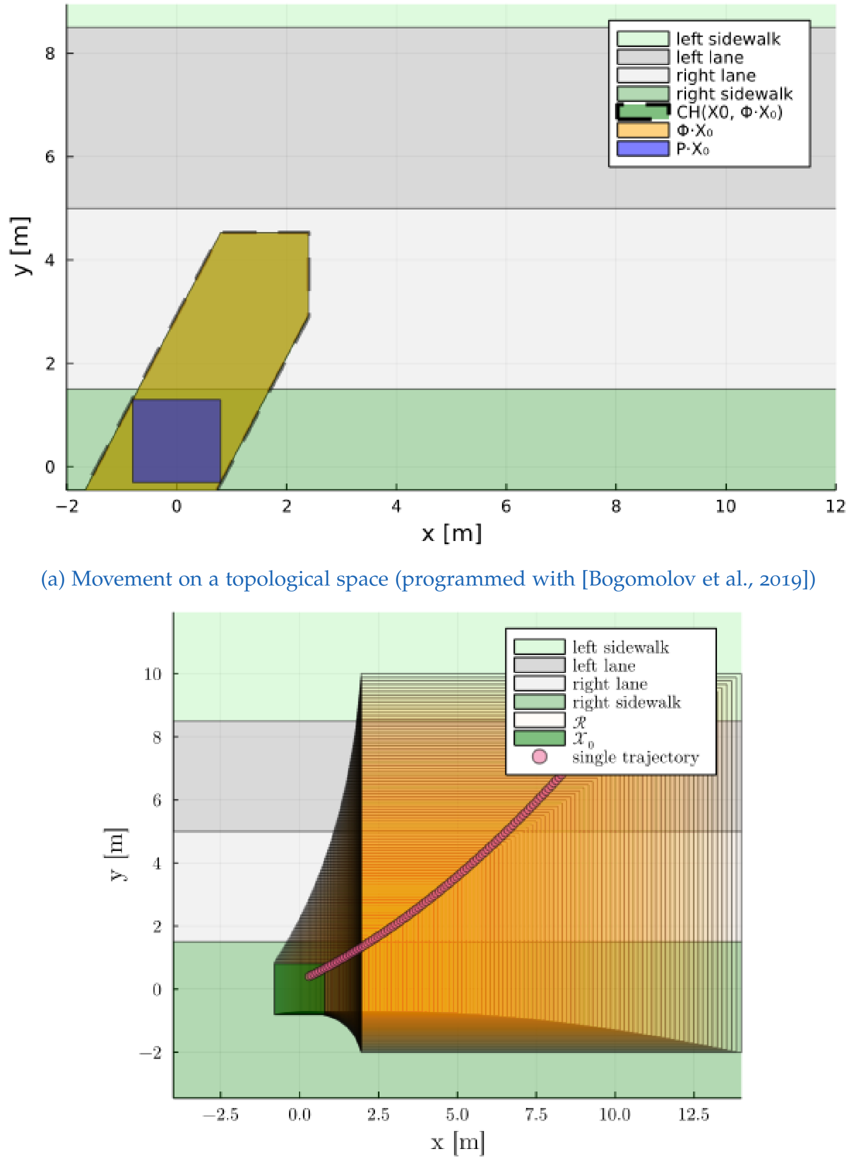
This section should motivate the use of set-based prediction algorithms for pedestrian movement algorithms on manifolds. Figure 3.42a shows the synthetic movement data of a pedestrian in a topological space $\mathcal{X} = \{\mathcal{X}_q\}_{q \in \mathcal{I}}$ and topology set $\mathcal{T}$[25]. The index set is $\mathcal{I} = \{0, \ldots, 39\}$.[26] The pedestrian movements in this simple example are programmed as a fixed deterministic function from the current position $\boldsymbol{p}(t)$:

$$v_x(t) = \mathbf{f}(\boldsymbol{p}(t)) \tag{3.52}$$
$$v_y(t) = \mathbf{f}(\boldsymbol{p}(t)) \tag{3.53}$$

The algorithm 2 could be used to generate pedestrian movements (blue trajectory with multiple decisions on each manifold cell 1, 2, 12, 13, 14, 15) or a black trajectory with single decision in cell 0. Each manifold cell describes

[25] $\mathcal{X}_{11}$ is connected with $\mathcal{X}_{10}, \mathcal{X}_{21}, \mathcal{X}_{12}, \mathcal{X}_1$. Formally every set $\mathcal{X}_i$, which is adjacent to another set $\mathcal{X}_j$ for $i \neq j$ is element of the topology set $\mathcal{T}$. Exact mathematical definitions can be found in [Cavalieri, 2007].

[26] For visualization reason the boundaries of each cell is not visualized.



(a) Movement on a topological space (programmed with [Bogomolov et al., 2019])

(b) Reachability Analysis with package and single trajectory (programmed with [Bogomolov et al., 2019])

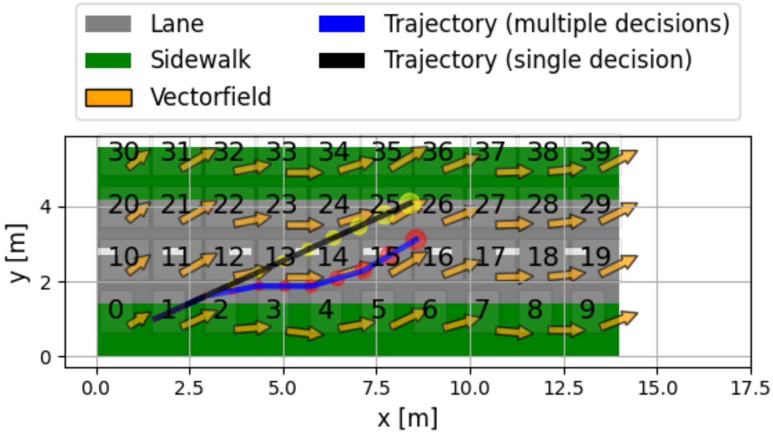
Figure 3.41.: Reachability analysis

a single decision (brain) where a pedestrian follows a certain action (going in a fixed direction).[27] Each cell could also have more complex nonlinear behaviors in form $\dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x}, \boldsymbol{u})$ valid for each cell. [28]. It might also be probable that another person or the same person in a different situation, mood, or personal situation behaves differently. There might be also another vector-field plausible, compare the multi-vector field in figure 3.42b.

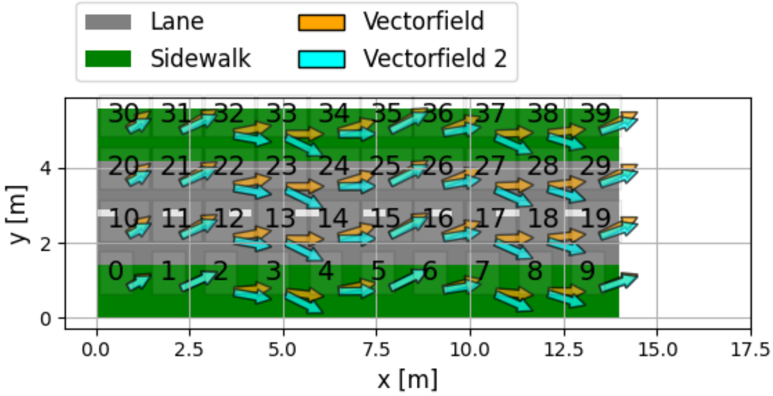
There exists the problem of knowledge vs. lack of knowledge (reducible and nonreducible uncertainty, compare figure 3.43). There is the decision dilemma for an external perception system where a person's decision-making is not observed. There is the problem of distinguishing (reducible and nonreducible uncertainty). The classification of knowledge and uncertainty is part of the uncertainty quantification for the mind-body problem (how to predict the decision-making of the human mind and its resulting actions for the body). Besides that, we have to distinguish the set of plausible decisions in an area that might not be the same as the set of probable decisions. It is problematic to quantify certainty in a prediction without knowledge. It is not known how a foreign person chooses decisions. The hidden cognitive dynamics and the influences of the decisions on the movements are unknown. It is necessary to distinguish between the correlation and causation of mechanisms of action. Understanding the cognitive dynamics (pedestrian in the loop) and their causal relationships is necessary. Experienced drivers can have causal relationships when a child crosses the street to catch the ball. A randomized controlled experiment is needed to go deeper into the ability to decode the causal relationships. From a mathematical point of view, we could tackle the problem in the form of a set-based propagation system on a manifold. Instead of using a single vector field in the tangency space, using a multi-vector field or set-based approach on manifolds might be better. Figure 3.44a shows on each cell three yellow vectors not representing the mean value but rather the closure of the velocity set. The set-based movement prediction system might

[27] This simple example shows only that a particle is moved in the direction of the certain arrow of the vector field. It is not observed how the brain performs decision-making in detail to get the sub-goals.

[28] In this simple example, we also used other simplifications. We used regular grids, and the trajectory is made with low variance. For the realization of the trajectory, we used $dot(\boldsymbol{x}) = \boldsymbol{v}$ the numerical integration of the x-y-components of the velocity



(a) Comparison of two trajectories with multiple subgoals and one decision. The yellow and red points should highlight the direction of the pedestrian.



(b) Instead using a single vector-system the same person could behave differently (compare figure 3.42a). This is also a consequence of the free will, the cognitive decision-making and the possibility to choose differently than mass (depending only on energy).

Figure 3.42.: Modeling of pedestrian movements

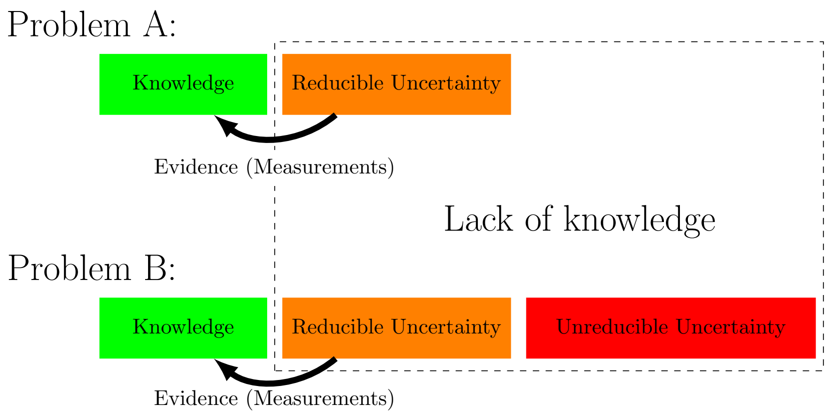
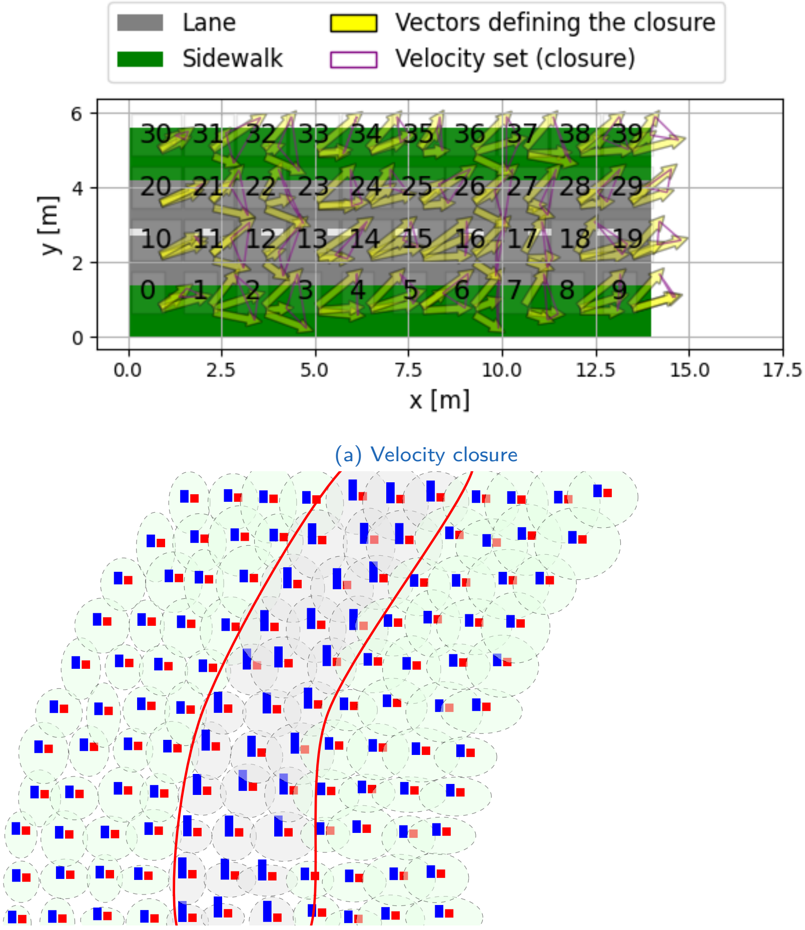


Figure 3.43.: Knowledge and the lack of knowledge in form of two sources of uncertainty (epistemic equals reducible uncertainty and aleatory equals unreducible uncertainty)

over-approximate the velocity set to cover all possible movements. Instead of describing the closure of the velocity set, one might try to estimate the maximal velocity and acceleration in each cell-like in figure 3.44b. Figure 3.44b is visualizing a selection of the street (grey cells with red border) and sidewalk (green cells) from figure C.5.

3.5.10. Computation of set-based movement prediction

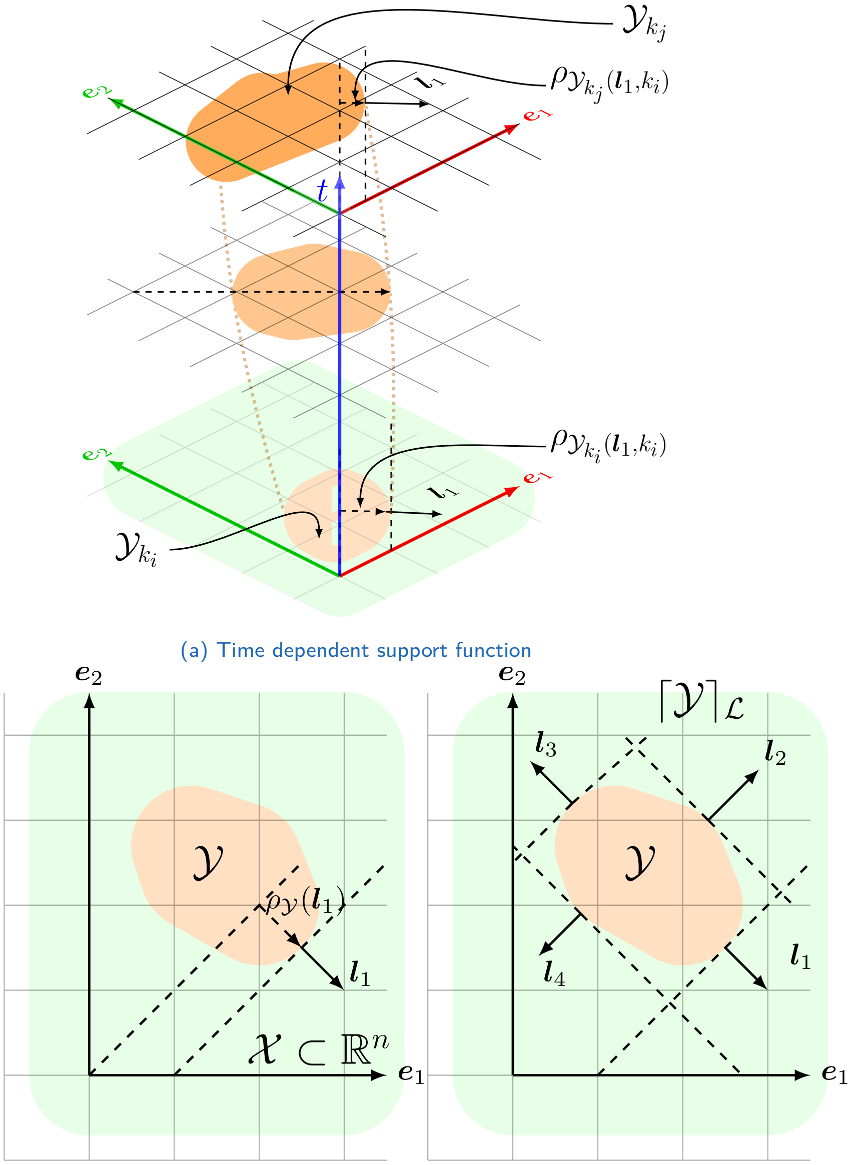
This section presents computation methods for set-based-prediction. Figure 3.26 presented a simulation for funnel prediction for real-pedestrian movements. It can be seen as a special case to over-approximate time-varying rectangle with a set of support functions $\hat{\rho}_{\mathcal{Y}}$ over time t. Figure 3.45a shows the three dimensional expansion (coordinates x, y for spatial expansion and time t) in time-state-space. The red and green basis vectors (often also as x-y-coordinates) describe the spatial movements over time (blue basis vector). The closure (compare [Cavalieri, 2007] for an exact mathematical definition) of a (convex) set is an important topological property. The boundaries of a (convex) set $\mathcal{Y}$ could be approximated by support functions. A support function $\rho_{\mathcal{Y}}(\boldsymbol{l})$ of a compact set $\mathcal{Y} \subset \mathcal{X}$ is defined as [Althoff and Frehse,



(a) Velocity closure

(b) Maximal-velocity (represented as blue bar) and acceleration (represented as red bar) on each cell $\mathcal{X}_q$ of a topological space

Figure 3.44.: Reachable sets instead of Multivectorsystem

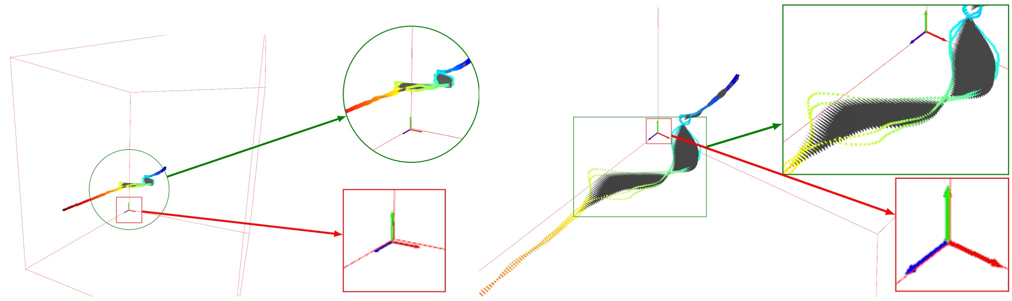


(a) Time dependent support function

(b) Support-function

(c) Over-approximation

Figure 3.45.: Support function and over-approximation (adapted from [Althoff and Frehse, 2016])



(a) Three dimensional illustration of a funnel

(b) Image detail of the funnel

Figure 3.46.: Three dimensional funnel

2016], compare figure 3.45b.

$$\rho_{\mathcal{Y}}(\boldsymbol{l}) = \max\{\boldsymbol{l}^T\boldsymbol{x}|\boldsymbol{x} \in \mathcal{Y}\} \tag{3.54}$$

The halfspace for a given direction $\boldsymbol{l}_1$:

$$\mathcal{H}_l = \{\boldsymbol{x} \in \mathbb{R}^n|\boldsymbol{l}^T\boldsymbol{x} \leq \rho_{\mathcal{X}}(\boldsymbol{l})\} \tag{3.55}$$

The over-approximation by a set of support functions can be described as [Althoff and Frehse, 2016]:

$$\lceil\mathcal{Y}\rceil = \bigcap_{l\in\mathcal{L}}\{\boldsymbol{x} \in \mathbb{R}^n|\boldsymbol{l}^T\boldsymbol{x} \leq \rho_{\mathcal{Y}}(\boldsymbol{l})\} \tag{3.56}$$

The normal-vectors of the halfspaces in figure 3.45b are collected in the set $\mathcal{L} = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_4\}$.

If the set $\mathcal{Y}(t)$ is time-dependent, one could use a time-dependent support function for the time interval $[t_{\text{min}}, t_{\text{max}}]$.

$$\rho_{\mathcal{Y}(t)}(\boldsymbol{l}, t) = \max\{\boldsymbol{l}^T\boldsymbol{x}(t)|\boldsymbol{x}(t) \in \mathcal{Y}(t), t \in [t_{\text{min}}, t_{\text{max}}]\} \tag{3.57}$$

Figure 3.45a shows the support function $\rho_{\mathcal{Y}_{k_i}}(\boldsymbol{l}, k_i)$ and $\rho_{\mathcal{Y}_{k_j}}(\boldsymbol{l}, k_j)$ for the discrete time-step $k_i, k_j \in \mathbb{T}$:

$$\rho_{\mathcal{Y}_{k_q}}(\boldsymbol{l}, k_q) = \max\{\boldsymbol{l}^T\boldsymbol{x}_{k_q}|\boldsymbol{x}_{k_q} \in \mathcal{Y}_{k_q}, k_q \in \mathbb{T}\} \tag{3.58}$$

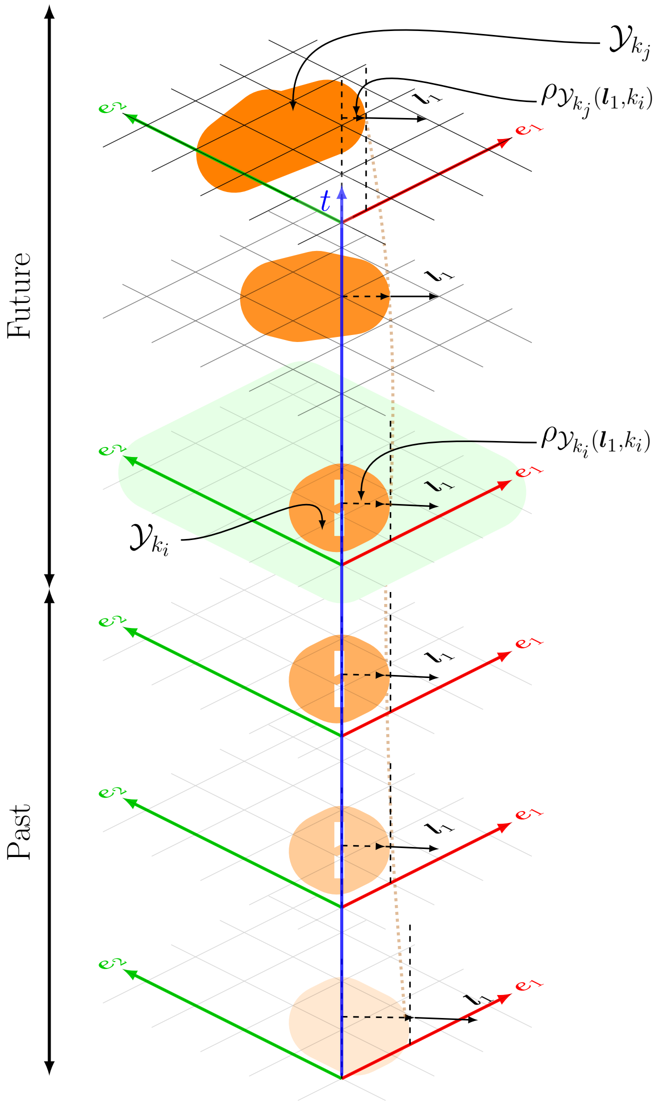


Figure 3.47.: Extrapolation of the support function

The three dimensional illustration in figure 3.46 shows a part of the overapproximation with the rectangular funnel in simulations from figure 3.26.

We formulate the time-dependent halfspace for a given direction $\boldsymbol{l}_1$ as following:

$$\mathcal{H}_{\boldsymbol{l}}(t) = \{\boldsymbol{x}(t) \in \mathbb{R}^n | \boldsymbol{l}^T \boldsymbol{x}(t) \leq \rho_{\mathcal{X}}(\boldsymbol{l}, t)\} \tag{3.59}$$

one could overapproximate the funnel of the set $\mathcal{Y}_{k_q}(t)$:

$$\lceil \mathcal{Y}(t) \rceil = \bigcap_{\boldsymbol{l} \in \mathcal{L}} \{\boldsymbol{x}(t) \in \mathbb{R}^n | \boldsymbol{l}^T \boldsymbol{x}(t) \leq \rho_{\mathcal{Y}}(\boldsymbol{l}, t)\} \tag{3.60}$$

One could overapproximate the funnel of the set $\{\mathcal{Y}_{k_q}\}_{q=i}^{j}$:

$$\{\lceil \mathcal{Y}_{k_q} \rceil\}_{q=i}^{j} = \bigcap_{\boldsymbol{l} \in \mathcal{L}} \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{l}^T \boldsymbol{x} \leq \rho_{\mathcal{Y}}(\boldsymbol{l})\} \tag{3.61}$$

Instead of knowing the real funnel $\mathcal{Y}(t) \; \forall \; t \in [t_i, t_j]$ we could estimate the funnel $\hat{\mathcal{Y}}(t) \; \forall \; t \in [t_i, t_j]$. Suppose we observe a dynamic system with aleatory uncertainty. We might not wish to predict the future position (which might lead to fatal consequences). Based on our knowledge $\mathcal{K}$, we would rather over-approximate the reachability sets (for fixed or time-varying control input sets).

$$\mathcal{R}(t_q) \subseteq \lceil \hat{\mathcal{Y}}(t_q) \rceil \; \forall \; t_q \in [t_i, t_j] \tag{3.62}$$

or for the discrete case:

$$\mathcal{R}_{k_q} \subseteq \lceil \hat{\mathcal{Y}}_{k_q} \rceil \; \forall \; k_q \in \{k_i, \ldots, k_j\} \subset \mathbb{T} \tag{3.63}$$

Usually, this is an ill-posed problem because only the realizations of the trajectories are known.

3.5.11. Algorithms for set-based movement prediction

We observe trajectories from a dynamic system with aleatory uncertainty. If we do not have information about the system model $\mathbf{f}(\boldsymbol{x}, \boldsymbol{u})$ or guessing the system model $\widehat{\mathbf{f}}(\boldsymbol{x}, \boldsymbol{u})$. we would have a model-based approach. We do not know the function of the dynamic system $\mathbf{f}(\boldsymbol{x}, \boldsymbol{u})$. We rely on a data-based approach with machine learning. The control input set $\mathcal{U}(t)$ and the resulting reachable sets $\mathcal{R}(t)$ are not known either. As we know the initial state set $\mathcal{X}_0$, we could compute a funnel to over-approximate the unknown reachable sets $\mathcal{R}(t)$ at least the observing trajectories. However, there might be the risk of not capturing the whole reachability sets and therefore accepting the consequences (e.g., accident). An example was given in section 3.4 with estimating the maximal velocity $\boldsymbol{v}_{\max}$ and acceleration $\boldsymbol{a}_{\max}$, which has the advantage of estimating only two parameters. The maximal velocity $\boldsymbol{v}_{\max}$ and acceleration $\boldsymbol{a}_{\max}$ are only a special case for state-vectors considering positions and velocities like $\boldsymbol{x}(t) = [x(t), y(t), v_x(t), v_y(t)]^T$ and representing $\widehat{\dot{\mathcal{X}}}(t)$ by a two dimensional ball and the two parameters. A more general approach would be to collect all possible state-changes in a set $\dot{\mathcal{X}}(t)$:

$$\dot{\mathcal{X}}(t) = \bigcup_{\boldsymbol{u}(t) \in \mathcal{U},\, \boldsymbol{x}(t) \in \mathcal{X}(t)} \dot{\boldsymbol{x}}(t) = \bigcup_{\boldsymbol{u}(t) \in \mathcal{U},\, \boldsymbol{x}(t) \in \mathcal{X}(t)} \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{3.64}$$

Similar to $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + T_S \cdot \mathbf{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ we could approximate the change of possible states from $\mathcal{X}_k$ to $\mathcal{X}_{k+1}$ for all possible control-inputs $\boldsymbol{u}_k$ in constant control input set $\mathcal{U}$. The increase of possible states could be described with:

$$\begin{aligned} |\mathcal{X}_{k+1} \setminus \mathcal{X}_k| = | \bigcup_{\boldsymbol{x}_{k+1} \in \mathcal{X}_{k+1}} \boldsymbol{x}_{k+1} \setminus \bigcup_{\boldsymbol{x}_k \in \mathcal{X}_k} \boldsymbol{x}_k | = \\ | \bigcup_{\boldsymbol{x}_k \in \mathcal{X}_k, \boldsymbol{u}_k \in \mathcal{U}} \boldsymbol{x}_k + T_S \cdot \mathbf{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \setminus \bigcup_{\boldsymbol{x}_k \in \mathcal{X}_k} \boldsymbol{x}_k | \end{aligned} \tag{3.65}$$

A fictive example is an agent on position $\boldsymbol{x}_{k_0} = 0$ with control input set $\mathcal{U} = \{-1, 0, 1\}$ and system model $\mathbf{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_k + \boldsymbol{u}_k$. In next step $k = 1$ we have the state sets $\mathcal{X}_1 = \{-1, 0, 1\}$, $\mathcal{X}_2 = \{-2, -1, 0, 1, 2\}$, $\mathcal{X}_n = \{-n, -n + 1, \ldots, n-1, n\}$. In each iteration we get two more states. [29] After that one

[29] In higher dimensional problems we get additional complexity because of the curse of dimensionality.

could try to over-approximate $\dot{\mathcal{X}}(t)$ with $\lceil\dot{\mathcal{X}}\rceil(t)$. One important part of the safety is that we assume problem B (compare figure 3.43). for the pedestrian, we have to consider the lack of knowledge, which cannot be reduced. This lack of knowledge is tricky, and one could add a set (e.g., by Minkowski sum) to enlarge the funnel. It is an open issue to quantify that for pedestrians because of the complexity of the problem (compare B).

3.6. Causal inference

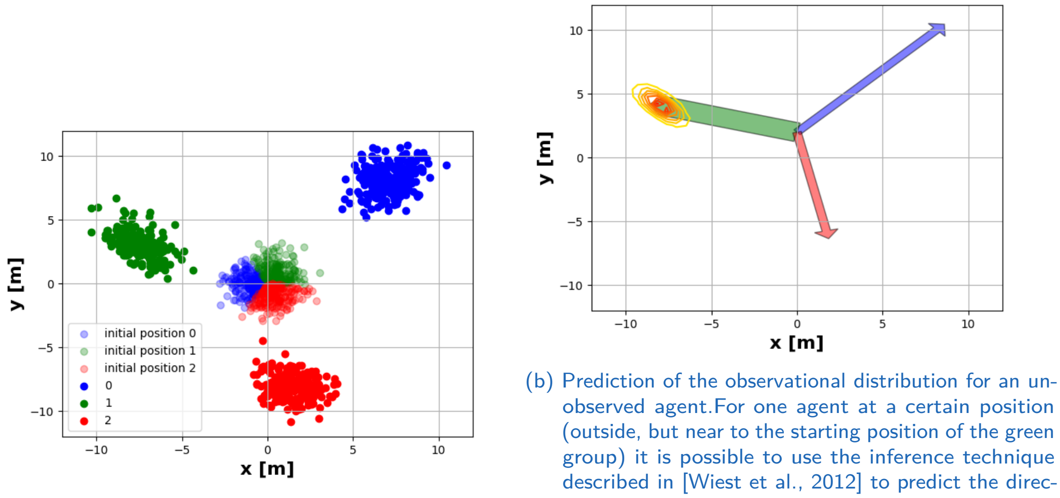
Subsection 3.6.1 describes the motivation for causal inference. Section 3.6.2 offers a hypothetical example for the intervention in non-cyclic models and section 3.6.3 proposes some ideas for cyclic models.

3.6.1. Motivation for Causal Inference

Many external and internal factors influence a human (compare section B.2). A causal model (e.g., pedestrian with the oncoming vehicle) might be interesting for human locomotion. In section 2.2.1 movement prediction algorithms were used based on observational studies (historical experiment). The experiment is only successful when the data has some quality and quantity to make conclusions about the behavior. The inference of causal relationships is of great interest to performing experiments in (ideal- and realistic) environments, where the factors and parameters could be changed to analyze the effect of some interventions. A random-controlled experiment is an environment for experiments where the causal relationships could be inferred by changing only a few parameters. In observational studies, this is often not the case. Enormous effort is often necessary to get datasets from observational studies for pedestrian movements. Machine learning models only could find patterns in datasets, where correlations might lead to suspect conclusions. A random-control experiment for the analysis of collision avoidance scenarios is proposed in section 3.7. A test environment should help to analyze the behavior of pedestrians with oncoming vehicles (e.g., use of virtual- and augmented reality, brain scanner devices, and non-movable walking platforms). To perform experiments in reality in the form of a randomized controlled experiment to analyze collision avoidance systems with real pedestrians is not ethical. It is not realistic to perform dangerous and real experiments with high-speed collision avoidance maneuvers with real pedestrians. It might be difficult to analyze the behavior of pedestrians in dangerous situations (e.g., accidents). The causal relationships might interest the movement prediction algorithms in critical situations. The causal relationships between the behavior of a pedestrian and the oncoming vehicle are complicated. However, some relationships could be modeled by causal inference. The field of causal inference describes settings for interventions and counterfactuals. It might provide an adequate framework for modeling the change of intentions to new targets or decisions. Interventions could happen in the current situation, where the exact time is important (cyclic models), or in non-cyclic models where the intervention affects a population of pedestrians. Non-cyclic models could be of interest to city planners. A new building or construction site (intervention) could affect the typical movements, habits, and decisions of a population of pedestrians. An example of cyclic models would be the effect of the intervention on an oncoming vehicle on a walking person, where the exact time for the influences is essential. Also, a phone call or a new intention could be interpreted as an intervention. Section 3.6.2 gives a hypothetical example for non-cyclic models which might affect a population of pedestrians. This example could be applied to the example shown in figure 3.37, where the initial starting- and end-positions are shown. The question "How would the typical starting and end positions be changed after placing warning signs or advertising pillars, making street lettering or a construction site" might be interesting.

3.6.2. Example for non-cyclic interventions

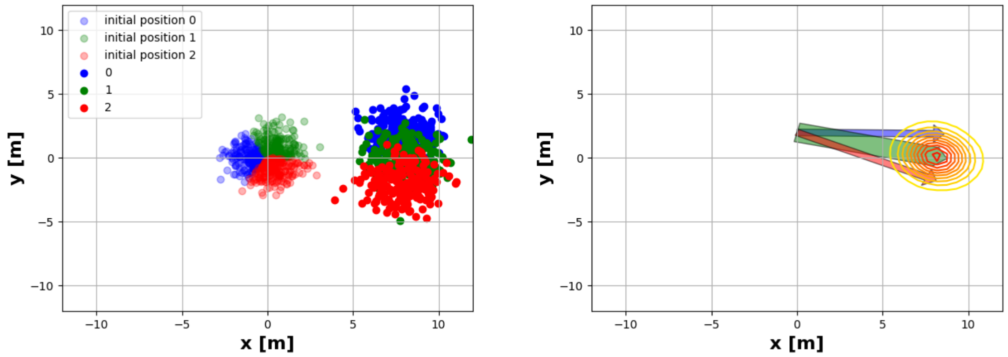
Figure 3.48 (observational distribution), 3.49 (intervention distribution) and 3.50 (counterfactual distribution) shows a fictive experiment with a population of particles representing a group of persons. It should describe an experiment which might show the effect of an intervention to a group of people in a room. The dots transparent dots show initial positions and the end positions with opaque color. The group is represented by the color (blue, green, red). We use an example with non-cyclic interventions. There might



(a) Observational distribution. The agents are all in the middle of the room (transparent colors) and walking to different directions dependent on their group (green, red, blue). The opaque colors show their end position.

(b) Prediction of the observational distribution for an unobserved agent.For one agent at a certain position (outside, but near to the starting position of the green group) it is possible to use the inference technique described in [Wiest et al., 2012] to predict the direction of the unobserved agent. The algorithm predicts that the agent is walking in the direction of the end-positions of the green group (thick green arrow). The ellipses show the contours of the posterior probability.

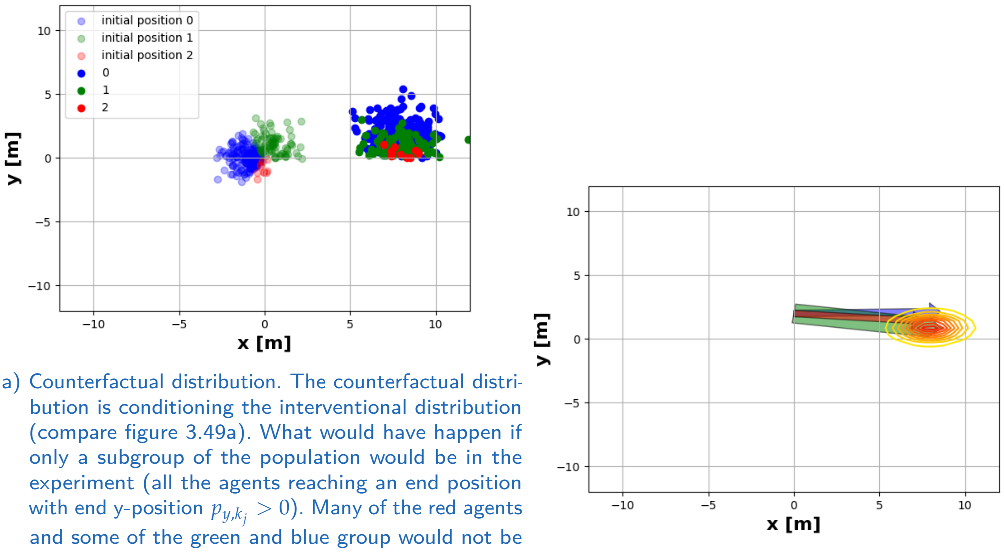
Figure 3.48.: Results of the hypothetical example (observational setting)



(a) Interventional distribution. An external intervention (e.g. a fire alarm) might affect the behavior of all groups to move in direction of the emergency exit.

(b) An unobserved agent might also be affected by the intervention, so that the arrows (predicted means) are directed to the right direction.

Figure 3.49.: Results of the hypothetical example (interventional setting)



(a) Counterfactual distribution. The counterfactual distribution is conditioning the interventional distribution (compare figure 3.49a). What would have happen if only a subgroup of the population would be in the experiment (all the agents reaching an end position with end y-position $p_{y,k_j} > 0$). Many of the red agents and some of the green and blue group would not be considered in the experiment with the intervention.

(b) Counterfactual cond. distribution

Figure 3.50.: Results of the hypothetical example (counterfactual setting)

be the observational probability distribution between the historical state $\boldsymbol{x}_h$ (transparent colors) and the future state $\boldsymbol{x}_f$ (opaque colors) described with the Gaussian mixture distribution [Wiest et al., 2012]:

$$p(\boldsymbol{x}_f, \boldsymbol{x}_h) = \sum_{i=1}^{I} \pi_i \mathcal{N}(\boldsymbol{x}_h, \boldsymbol{x}_f | \mu_i, \Sigma_i) \tag{3.66}$$

To perform a prediction from $\boldsymbol{x}_h$ to $\boldsymbol{x}_f$ the conditional mixture density [Wiest et al., 2012] is:

$$p(\boldsymbol{x}_f | \boldsymbol{x}_h) = \frac{p(\boldsymbol{x}_h, \boldsymbol{x}_f)}{\int p(\boldsymbol{x}_h, \boldsymbol{x}_f) d\boldsymbol{x}_f} = \sum_{i=1}^{I} \mathcal{N}(\boldsymbol{x}_f | \boldsymbol{x}_h, \tilde{\mu}_i, \tilde{\Sigma}_i) \tag{3.67}$$

with the parameters

$$\tilde{\pi}_i = \frac{\pi_i p(\boldsymbol{x}_h | \mu_{i,\boldsymbol{x}_h}, \Sigma_{i,\boldsymbol{x}_h,\boldsymbol{x}_h})}{\sum_{j=1}^{K} \pi_j p(\boldsymbol{x}_h | \pi_{j,\boldsymbol{x}_h}, \Sigma_{j,\boldsymbol{x}_h \boldsymbol{x}_h}} \tag{3.68}$$

$$\tilde{\mu}_i = \mu_{i,\boldsymbol{x}_f} + \Sigma_{k,\boldsymbol{x}_f,\boldsymbol{x}_h} \Sigma^{-1}_{i,\boldsymbol{x}_h,\boldsymbol{x}_h} (\boldsymbol{x}_h - \mu_{i,\boldsymbol{x}_h}) \tag{3.69}$$

$$\tilde{\Sigma}_i = \Sigma_{k,\boldsymbol{x}_f \boldsymbol{x}_f} - \Sigma_{i,\boldsymbol{x}_f \boldsymbol{x}_h} \Sigma^{-1}_{i,\boldsymbol{x}_h \boldsymbol{x}_h} \Sigma_{i,\boldsymbol{x}_h \boldsymbol{x}_f} \tag{3.70}$$

$$\mu_i = \begin{bmatrix} \mu_{i,\boldsymbol{x}_h} \\ \mu_{i,\boldsymbol{x}_f} \end{bmatrix} \tag{3.71}$$

$$\Sigma_i = \begin{bmatrix} \Sigma_{i,\boldsymbol{x}_h\boldsymbol{x}_h} & \Sigma_{i,\boldsymbol{x}_h\boldsymbol{x}_f} \\ \Sigma_{i,\boldsymbol{x}_f\boldsymbol{x}_h} & \Sigma_{i,\boldsymbol{x}_f\boldsymbol{x}_f} \end{bmatrix} \tag{3.72}$$

If the future state $\boldsymbol{x}_f$ is independent from $\boldsymbol{x}_h$ this would apply that $p(\boldsymbol{x}_f|\boldsymbol{x}_h) = p(\boldsymbol{x}_f)$ and the causal relationship between these two variables would disappear. Otherwise their is a causal relationship. We assume the initial distribution $\boldsymbol{x}_h$:

$$p(\boldsymbol{x}_h) = \mathcal{N}(\mu_h, \Sigma_h) \tag{3.73}$$

Depending on the sample position for $\boldsymbol{x}$ for $\mathcal{N}(\boldsymbol{x}, \mu_h, \Sigma_h)$ a KMeans algorithm clusters the initial distribution [Murphy, 2012] in three groups. Each person gets a label representing a color based on the initial position. Depending on the label $i \in \{red, green, blue\}$ the person goes in another direction. The end-position is dependent on the label of each person. Assuming there is an intervention from the environment (alarm, signpost, oncoming cars, or something else). Also, a consultation would influence the behavior of the subjects to go a certain direction, compare Figure 3.49a. As in section 2.2.3 the do-operator introduced by J. Pearl [Pearl and Mackenzie, 2018] describes an interventional setting. This intervention could be described with:

$$p(\boldsymbol{x}_f|\boldsymbol{x}_h) = \sum_{i=1}^{I} \mathcal{N}(\boldsymbol{x}_f|\boldsymbol{x}_h, \mathrm{do}(\tilde{\mu}_i := \widehat{\mu}_i, \tilde{\Sigma}_i := \widehat{\Sigma}_i)) \forall\, i \ \in \{1, \ldots, I\} \tag{3.74}$$

In this thought experiment only $\tilde{\mu}_i$ has changed to $\widehat{\mu}_i$ and $\tilde{\Sigma}_i$ has changed to $\widehat{\Sigma}_i$ for all $i \in \{1, \ldots, I\}$. Figure 3.50a shows the counterfactual distribution when the effect is conditioned. The conditional distributions for a certain point (from the green label) is shown in Figure 3.48b, 3.49b and 3.50b. The figure also visualizes the contours of the ellipses. Intervening in the effect could break the causal relationship between the cause and effect. This situation would mean there would be no $\boldsymbol{x}_h$ in formula 3.74. It does not seem important where the subject is initially coming from in this case. The intervention would lead to a certain end-position no matter from the initial position.

3.6.3. Interventions on cyclic models

The effect of an intervention in a situation with time-dependency is of interest in cyclic models. The influence of the oncoming vehicle on the pedestrian is of great interest. The pedestrian's physical parameters (velocity, acceleration, jerk) might be an exciting example that could be applied to the motion prediction and new pedestrian models. In physical systems (e.g., vehicle), there exist a causal relationship between the actuation of forces and a resulting movement, compare figure 3.51. Suppose a vehicle is accelerating by the drivetrain and assuming no steering and optimal conditions (e.g., flat, clean, and dry streets without obstacles). In that case, we could assume a causal relationship between the acceleration and the locomotion. The causal relationship might be disconnected if the road is slippery (e.g., black ice). The ability of a person to walk or run fast depends on several factors (e.g., physical-, medical, and psychological conditions): Fitness, injuries, age, situation, traffic situation, conventions of society, and also cultural factors. Also, technical tools could help to reach fast velocities (e.g., skateboards, electric scooters). As in the models of cognitive MDPs of section B.2 described, there exist (causal-) relationships between human cognition and the human body. The decisions of a healthy person would lead to the choice of the walking speed $v(t)$ [30]. Figure 3.51 shows the simple principle that the actuation of a mass with forces causes a movement on a macroscopic level[31]. Figure 3.52 shows that this kind of physical force (e.g., in movement prediction, the principle of social forces was introduced) does not tell the whole story. This model shows that human locomotion is more complex and has different subsystems interacting. A healthy person with heart-pumping blood in the body (cardiovascular) and a functioning nerve system are prerequisites of this model. The interested reader might read appendix chapter B for more details. The model has three subgroups (cognition from the brain, the actuation of forces by muscles, and the interior sensing of the human body). This model is very simplified and should only motivate the introduction of the cognitive-MDP and cognitive-POMDP in figure 3.53. In current research on the brain regions, there

[30] The causal relationship between the brain control signals and the actuation to the muscles is broken in a paralyzed person.

[31] Without considering friction effects and only considering Newton-laws.

is much more complexity in abstract thinking in the human brain. The brain regions interact with each other to solve complex cognitive tasks (compare B). The cognitive MDP (figure 3.53a) and POMDP (figure 3.53b) in section B complete the models. The cognitive MDP is introducing two states $\boldsymbol{c}(t)$ (cognitive state) and $\boldsymbol{x}(t)$ (body state), compare figure 3.53. The cognitive POMDP is an extension by adding the observational state $\boldsymbol{o}(t)$ and influences the cognitive state $\boldsymbol{c}(t)$. The observational state $\boldsymbol{o}(t)$ can be represented by the pictures sensed by the eyes and audio signals by the ears. The cognitive state can be measured with brain scanners. They allow us to represent brain activities in three-dimensional pictures. This illustration represents the brain with voxels (pixels in 2D images and voxels in 3D images), and the image was produced synthetically.

3.6.4. Pedestrian movements with causal do-operator

Figure 3.54 shows a simple modeling strategy of a system with state $\boldsymbol{x} = [p_x, p_y, v_x, v_y, a_x, a_y]^T$ and control input $\boldsymbol{u} = [u_x, u_y]^T$. We take the model from [Liu et al., 2017] and [Hartmann and Watzenig, 2019a] for acceleration and velocity constrained models for the simulation. For simplicity we only showing this example for the acceleration constrained model. It also holds for the jerk- and velocity constrained model.

$$\dot{\boldsymbol{x}}(t) = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{x}(t) + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{B}} \boldsymbol{u}(t) \tag{3.75}$$

The control input is constrained $\boldsymbol{u}(t) \in \mathcal{B}(a_{\max})$ where $\mathcal{B}(a_{\max})$ is a two-dimensional ball parameterized with the maximal acceleration $a_{\max}$. For the jerk- constrained and velocity constrained model it is the same procedure with different $\mathcal{B}(v_{\max})$ and $\mathcal{B}(e_{\max})$. We could compute the reachable sets from these different models $\mathcal{R}^v$, $\mathcal{R}^a$ and $\mathcal{R}^e$ and combine it by geometrical intersection $\mathcal{R}(t) = \mathcal{R}^v \cap \mathcal{R}^a \cap \mathcal{R}^e$. Now we could assume that we know the maximal parameters for $v_{\max}, a_{\max}, e_{\max}$. This assumption is often not the case, and the reachable sets have to be estimated $\mathcal{R}(t)$. One way to estimate the reachable sets is to estimate the parameters $\hat{\theta}$ of the corresponding $\mathcal{R}(t)$. In [Hartmann and Watzenig, 2019a] the maximal parameters for the velocity and acceleration were estimated by Gaussian processes and extrapolation for a specific situation. We do not know the real system dynamics, control inputs, and parameters for real-world situations with pedestrians. We could assume a linear system (for simplicity) and the parameter and control input set sampled from two different random processes to handle this difficulty. These random processes change their probability distributions by many different causes, where causal inference might take advantage. A maximal velocity and acceleration constrain the system. The control input is getting a signal from a probability distribution in this case from a normal distribution $u_x \sim \mathcal{N}(\mu_{x_k}, \Sigma_x), u_y \sim \mathcal{N}(\mu_{y_k}, \Sigma_y)$. μ_{x_k}, μ_{y_k} are the first ten seconds not constant and also sampled by a normal distribution. After $10s$ the mean values of the normal distributions gets constant μ_x, μ_y. This intervention leads to a u-turn red dot on the trajectory (Figure 3.54 a shows the projection on the two-dimensional spatial ground). Yellow areas visualize the reachable sets on the same picture. The velocity acceleration, control inputs, and causal intervention on μ_{x_k}, μ_{y_k} have two sections. The first 10 seconds have time-variation on μ_{x_k}, μ_{y_k} (on the left side of the vertical red line) and the right side with constant parameters. This simple example shows the movements of a dynamic system with random control inputs and causal interventions. We could ask where the dynamic system would move if the probability distribution would not have changed (counterfactual). In this example, it probable would move somewhere in the south direction.

3.7. Cognitive Decision Models

This thesis handles a topic with high interdisciplinarity. The analysis of human behavior is of interest in fields like psychology, biomechanics, and neuroscience. Environmental analysis is a geoscience topic, with geospatial

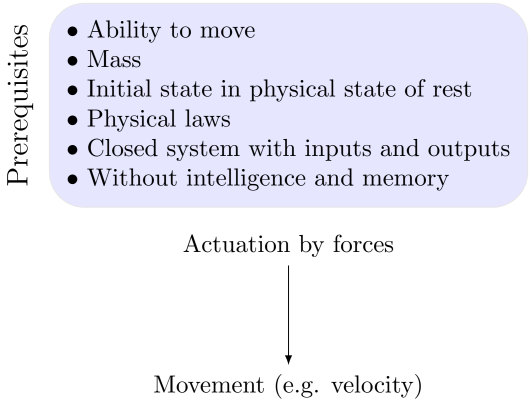

Prerequisites
• Ability to move
• Mass
• Initial state in physical state of rest
• Physical laws
• Closed system with inputs and outputs
• Without intelligence and memory
Actuation by forces
Movement (e.g. velocity)

Figure 3.51.: Causal relationship between the forces and movement of physical systems with selected prerequisites

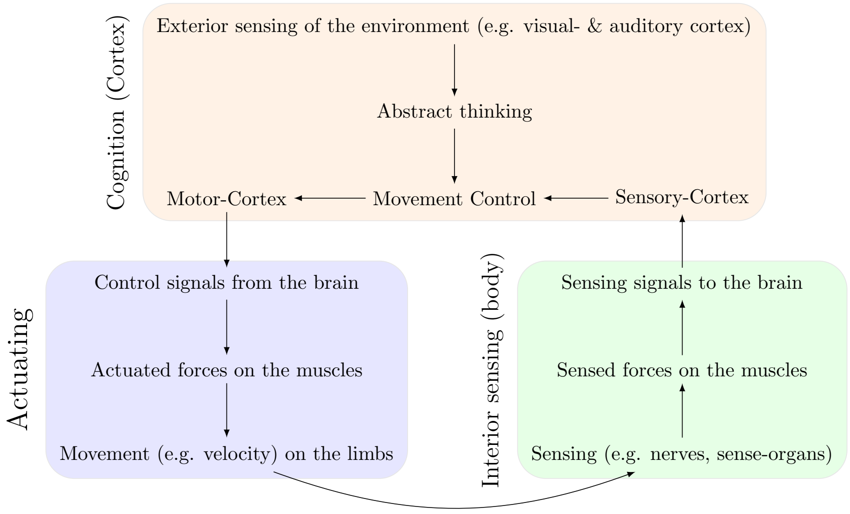
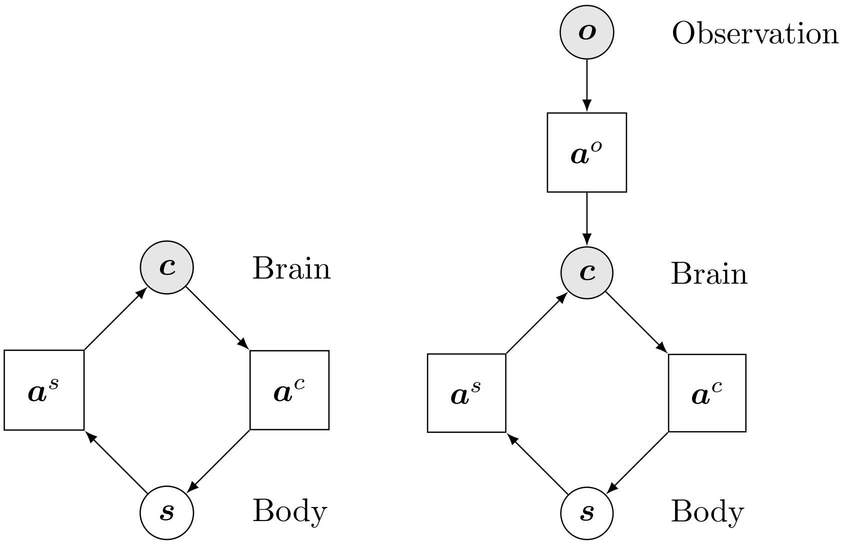

Cognition (Cortex)
Exterior sensing of the environment (e.g. visual- & auditory cortex)
Abstract thinking
Motor-Cortex
Movement Control
Sensory-Cortex
Actuating
Control signals from the brain
Actuated forces on the muscles
Movement (e.g. velocity) on the limbs
Interior sensing (body)
Sensing signals to the brain
Sensed forces on the muscles
Sensing (e.g. nerves, sense-organs)

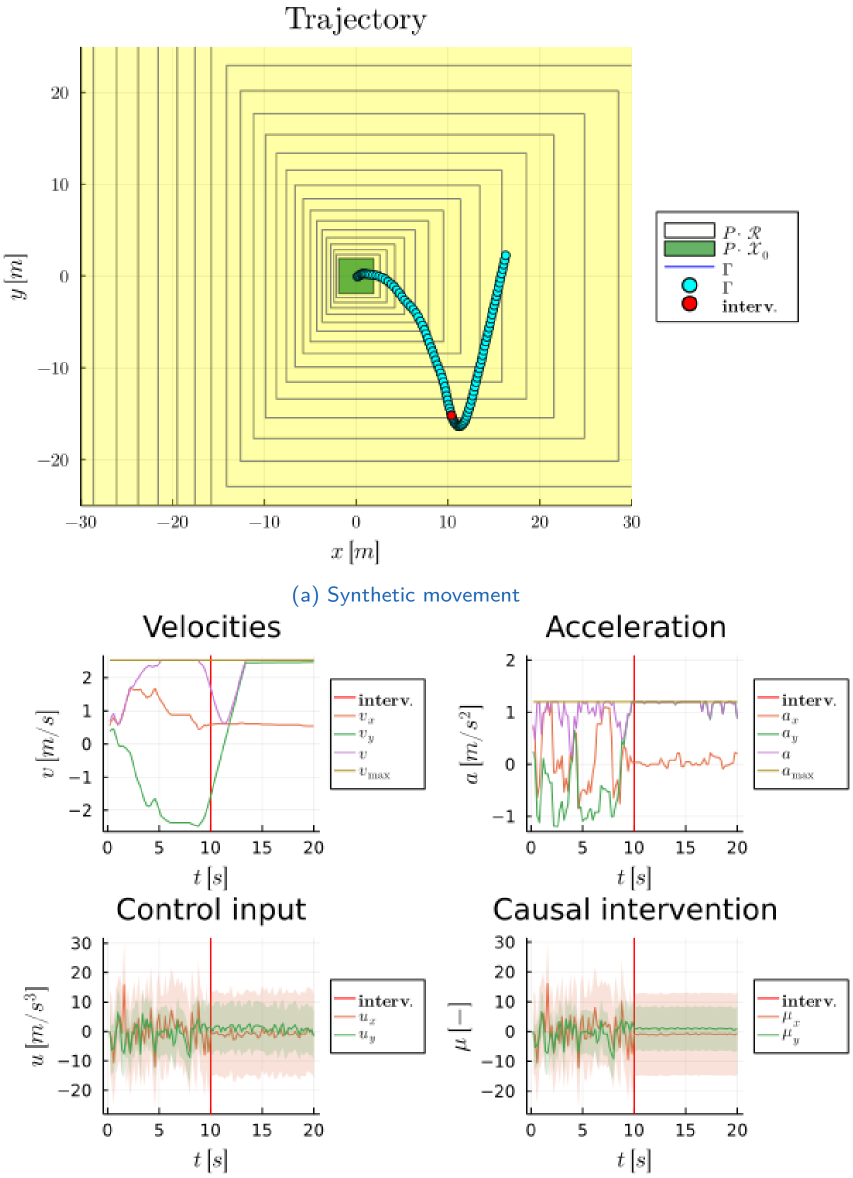
Figure 3.52.: Causal relationship between the control signals (actuating) from the brain, actuated forces on the muscles, and the resulting movements. The interior sensing (body) detects movements, and these signals come to the human brain via the peripheral nervous system. The sensory cortex (after the Brodman-atlas) is processing these signals. Via a movement control unit, the signals from the brain are translated into actuation signals in the motor cortex. Furthermore, the inner cycle of the human (sensing and actuating) is repeated. The human brain is more complex than this simple model. It offers the possibility of perceiving information from the environment (e.g., visual and auditory) and classifying it into concepts.



(a) Cognitive MDP

(b) Cognitive POMDP

Figure 3.53.: Cognitive MDP and POMDP. The cognitive MDP and cognitive POMDP should further simplify the concept from figure 3.52 and should represent an abstraction of the complex process into a computer model with use of markov-models. It should also highlight the simplifications of the model in section 3.5.5 where a simple MDP was used for the generation of synthetic movement data. The introduction from the cognitive MDP and cognitive POMDP can be find in the appendix chapter B. For the further use and application in automotive settings it is necessary to do more basic research on the human locomotion processes and measurements like in 3.7)

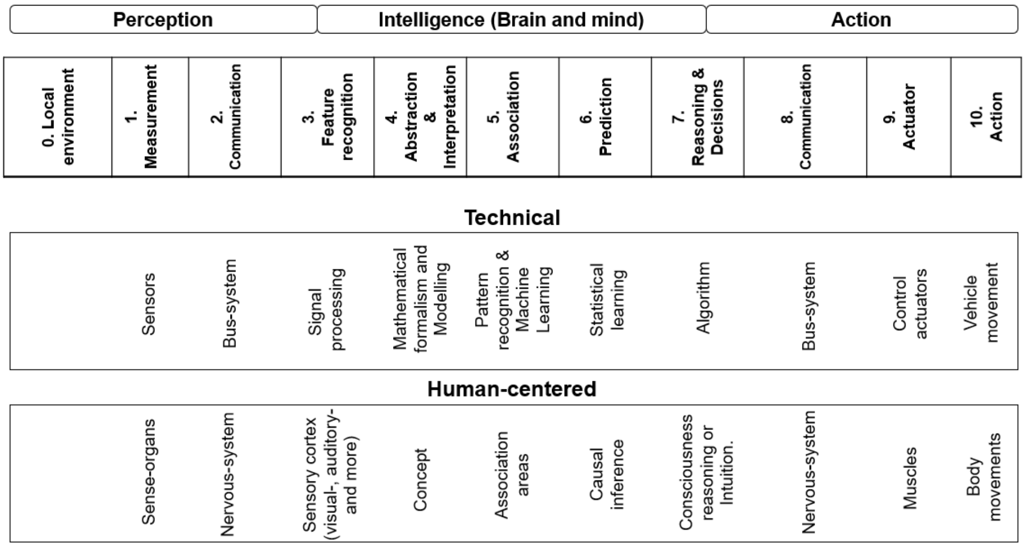


(a) Synthetic movement

(b) Velocities, Acceleration, Control input and intervention

Figure 3.54.: Synthetic movement with causal intervention

analysis and autonomous vehicles' behavior related to machine learning, computer vision, control engineering (vehicle dynamics), and information technology. For the decision-making of safe motion planning of autonomous vehicles, the thesis proposes a new test environment for basic research. It might be necessary to validate human prediction models. The decision models via a new randomized controlled experiment, which offers some benefits compared to the current state of the art with observational analysis [Hartmann et al., 2017b]. For example, in technical applications in macroscopic mechanical systems (e.g., vehicles), it is often sufficient to model classical Newton physics. Moreover, where the models describe the energy flow well after successful modeling and parameterization steps. Taking some measurements, building a dynamic system model, and testing the model with reality is standard practice. For technical systems, it is a common approach. Nevertheless, for modeling the behavior of humans, observational studies are the typical approach. Only taking observational studies has some problems. Into account, means presupposes that the underlying data describes the behavior of pedestrians well and generalize it to every situation. An additional problem compared to technical systems with a state-flow on the energy principle is that humans are constrained only by physical laws. The cognitive processes of human decision-making require that a person follow an intention. Observational studies can have an enormous amount of data describing human behavior. However, it is hard to guarantee that the data fulfills the requirements for building good models representing human locomotion. It might work in a specific environment but might fail in another context. The same problem exists with people from different cultures in different urban environments. The next emotion, the change of the mood or simply an observation of the person, could affect a new intention, leading to drastic consequences when an autonomous vehicle reaches the person. A randomized controlled experiment could help understand the insights of human intention making and help explore the causal relationships of human behavior. A randomized controlled trial wants to answer why a person is doing something—observational studies answer how a person acts based on past observations. The development of autonomous vehicles must guarantee safety. both experiment environments have their validity. However, there is no complete experiment to understand the entire cybernetic cycle (a mind-body problem in complex situations) [Hartmann et al., 2017b]. The current state of the art shows a broad spectrum of different technologies



Perception			Intelligence (Brain and mind)					Action		
0. Local environment	1. Measurement	2. Communication	3. Feature recognition	4. Abstraction & Interpretation	5. Association	6. Prediction	7. Reasoning & Decisions	8. Communication	9. Actuator	10. Action
Technical										
	Sensors	Bus-system	Signal processing	Mathematical formalism and Modelling	Pattern recognition & Machine Learning	Statistical learning	Algorithm	Bus-system	Control actuators	Vehicle movement
Human-centered										
	Sense-organs	Nervous-system	Sensory cortex (visual-, auditory- and more)	Concept	Association areas	Causal inference	Consciousness reasoning or Intuition.	Nervous-system	Muscles	Body movements

Figure 3.55.: Steps for technical and human-centered elements. An informal comparison between technical and human-centered processing steps.

available. In the introduction, we have shown that many different factors influence the behavior of pedestrians. A randomized control experiment could help analyze the interaction between human intelligence, the human body, and the environment. The automotive industry's current trend is to build models on observational studies and use (deep) learning models to model pedestrians' behavior.

Figure 3.55 shows a comparison between technical and human-centered systems and classifies technical and human-centered systems by perception, intelligence, and action, and there exist similarities between the function of biological and technical systems. Figure 3.56 shows a flow-chart with the computational intelligence and the human brain as the cause for decision making. Unobservable help systems are necessary to perform actions with the body of the vehicle and human. An example is the peripheral nervous system for the communication process and the bus systems in the vehicle. Alternatively, the pump systems for hydraulic systems in vehicles as a pendant to the heart pump blood and oxygen in the cardiovascular system. Figure 3.57 shows a simple model for the human locomotion. Action signals $\boldsymbol{a}_k^s$ coming from the peripheral nervous system change the body state

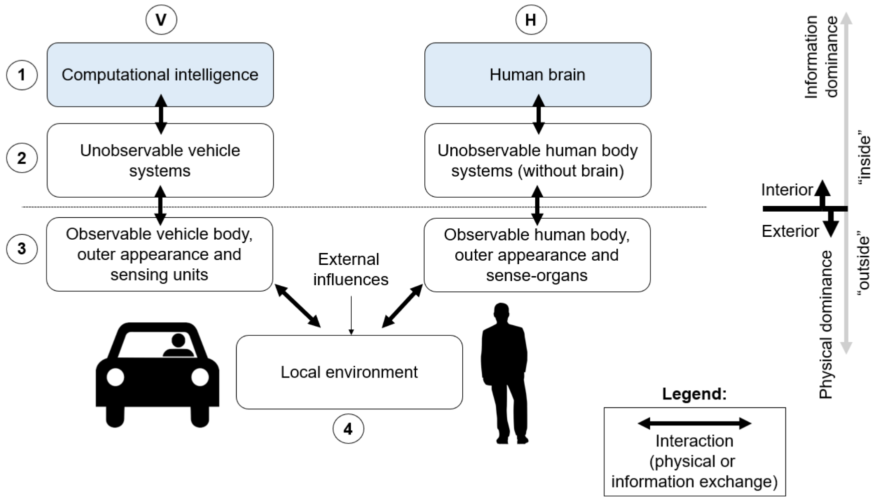

V
H
1
Computational intelligence
Human brain
2
Unobservable vehicle systems
Unobservable human body systems (without brain)
3
Observable vehicle body, outer appearance and sensing units
External influences
Observable human body, outer appearance and sense-organs
Local environment
4
Information dominance
"inside"
Interior
Exterior
"outside"
Physical dominance
Legend:
Interaction (physical or information exchange)

Figure 3.56.: The computational intelligence and the human brain are relevant for the decisions (first layer). There exist unobservable systems. For driving and locomotion dynamics (second layer). The observable body parts and sensing units are present, and information can occur via a local environment.

s_k by an action signal a_k^s formed originally from brain activity (Motor Cortex). There is also a feedback signal from the control transmitted via the peripheral nervous system, which sends signals from the human body to the brain (sensory cortex). The sequence of states c_k describes the state of the human brain. The action signals a_k^c influence brain activity. Physical quantities describe the states. Positions in a three-dimensional orthogonal coordinate system represent body states, and the different technologies (EEG, fMRI) measure brain activity, compare section B.3. A vehicle measures only the observable body states (white nodes). The other nodes are not observable (grey nodes). In [Hartmann et al., 2017b, Hartmann et al., 2018d, Hartmann et al., 2018a] we proposed a Pedestrian in the Loop environment to place a test person in a virtual/augmented environment with measuring the body state. Suppose someone wants to decode the nature and principles of human locomotion. In that case, it is necessary to extend the pedestrian in the Loop environments by decoding how observations affect the brain state and how cognitive intelligence controls the body.

Figure 3.58 extends Figure 3.57 by observation o_k of the sense organs from the environment. The observations influence the brain state by perception.

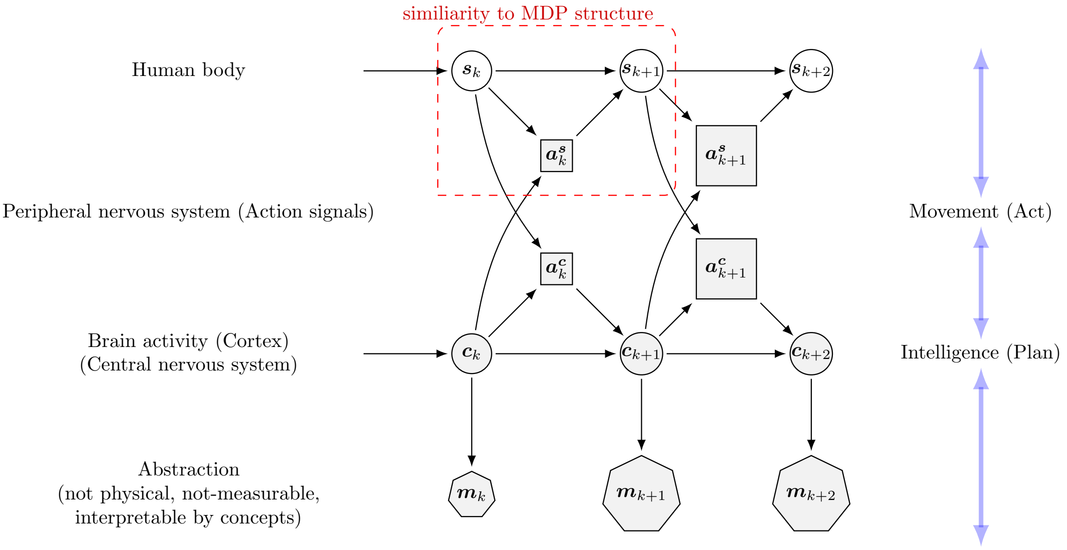


Figure 3.57.: Cognitive MDP

Figure 3.57 and 3.58 are models made from knowledge out from the literature. The models have to be validated by experiments. Especially brain activity has enormous complexity. Like images, the granularity of pixels affects the quality of the picture—the same for the voxel size in 3D images. Another drawback validating the models is the different sampling times for measuring the body state and brain activity. The sampling rate for measuring the body states is higher (lower sample time) than the measurements of the brain state. Figure 3.59a shows the legend of states, meta-states, actions, and rewards. The state is a measurable physical quantity. Brain activity and physical positions, and forces of the body state are measurable by electroencephalography (EEG), magnetic resonance imaging (MRI), functional magnetic resonance imaging (fMRI), and other techniques. The physical positions of the body state Meta-State is a nonphysical concept that gives a state an understandable meaning. When a proband is thinking of a chair, the brain activity patterns are measurable, and the concept of a chair is describable. The imagination is subjective, and therefore it is harder to find mathematical representations. Stimuli of the sense organs are often physically described by images or audio sequences. Action is a process of acting, changing a state. A reward is a quantity that an agent tries to maximize (reinforcement learning). $\mathcal{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{N_\mathcal{S}}\}$ is set of body states (agent). It is representable by the limbs' positions (leg, feet, hands, and more). $\mathcal{A} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{N_\mathcal{A}}\}$ is set of actions (agent). Represented by an action (clapping the hands, opening the door). $\mathcal{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{N_\mathcal{C}}\}$ is set of cortex states (agent). Brain activity patterns are measured by EEG, fMRI, or other technologies. $\mathcal{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{N_\mathcal{O}}\}$ is set of observations (agent). Stimuli of the sense organs (represented by pictures or audio signals). $\mathcal{E} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{N_\mathcal{E}}\}$ is set of environmental states. There exists a huge diversity of representing the environment. Very common is to represent the environment by regular grids or graphs. It is complex to represent the environment because the environment's semantic meaning plays an essential role [Vasquez, 2010b,Ziebart, 2010,Kitani et al., 2012]. $\mathcal{M} = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{N_\mathcal{M}}\}$ is set of meta states. Meta-State is a nonphysical concept that gives a state an understandable meaning. It describes the concept behind a physical representation and its semantics. Figure 3.59b illustrates an experimental platform for further research on human locomotion. The pedestrian in the Loop environment stimulates the sense organs with a virtual reality scene in an urban environment. The test person is walking on a non-movable walking platform. Sensors take cycle measurements of the brain (EEG, fMRI) and body state. This platform would be a quantitive way to extract and validate models of human locomotion. In this thesis, this experimental platform's realization was not realizable due to project reasons and not the main scientific focus. However, it shows that the lack of testing makes it difficult to ensure vulnerable road users' safety. The easiest way is to use observational studies and their datasets. However, this leads to models that might be acceptable only for a particular situation and might not be general enough for other situations in urban environments. [Hartmann et al., 2017b,Hartmann et al., 2018d,Hartmann et al., 2018a] gives additional details. This section presents models formed by reading the literature in fields unrelated to engineering topics (Neuroscience, Biomechanics) and computer science. The Markov property might not be valid for the cognitive state. Not only the past state might influence the new state. A randomized control experiment is necessary to do primary research exploring a model.

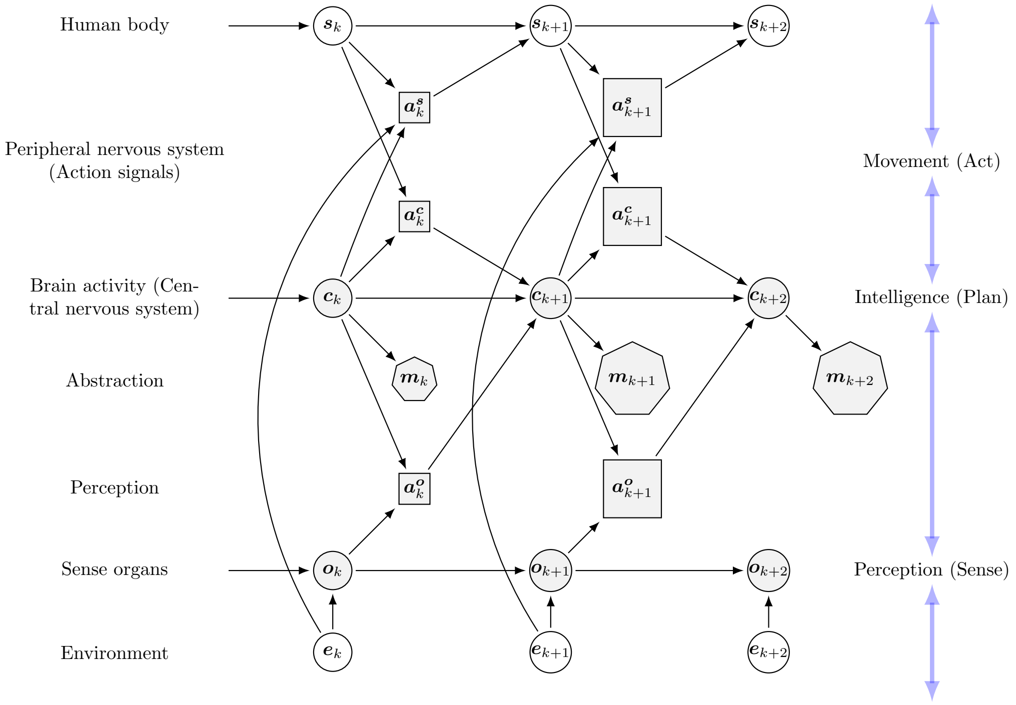
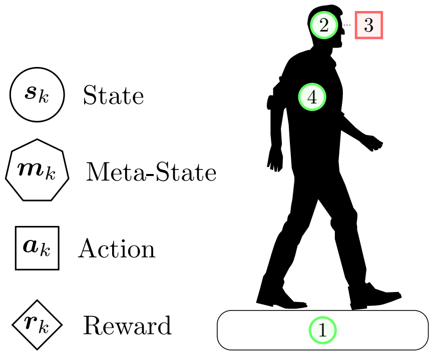


Figure 3.58.: Cognitive POMDP



(a) Legend (b) Testing platform

Figure 3.59.: 1. Measuring the the relative position in a virtual environment. A non-movable walking platform might be used. 2. Measuring of the brain state. 3. Stimuli of the sense organs by a virtual- or augmented- reality. 4. Measuring of all body movements. (Silhouette from a person [Silhouette of a Person, 2017])

4. Evaluation

The idea of this chapter is to give new models for pedestrian movements, especially embedding a dynamic system $\boldsymbol{x}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ on a manifold $\mathcal{M}$. The models are shown in simulations programmed in the programming language Julia© [Bezanson et al., 2017] for adaptive hull computations of pedestrians. Section 4.1 gives mathematical details for the implementation of pedestrian movements. Section 4.2 visualization of exemplary results with the simulator, model-predictive control (MPC) of the vehicle, and pedestrian models modeled by causal inference and with Markov Decision Processes (MDP)s. The focus is not on optimizing the MPC (convexity and nonlinear programming) but rather on presenting new modeling strategies for pedestrian movements. Section 4.3 discusses the causal inference for the goal intention. Section 4.4 offers the possibility to run reachable sets on topological spaces and connect the reach sets for complex urban environments. Section 4.5 compares different pedestrian models for twenty simulation runs to evaluate the performance of each simulation with other simulations. The information on used packages is in the appendix.

4.1. Mathematical details for the implementation in Julia programming

This section aims to combine the mathematical concepts of the previous chapter. The aim is to get a new framework for pedestrian movements consisting of Causal Inference (CI), Reachability Analysis (RA), and Markov Decision Processes (MDP) embedded on a manifold $\mathcal{M}$. Previous section 3.5 describes pedestrian movements in different mathematical expressions. A very common way is to describe it as a dynamic system $\dot{\boldsymbol{x}}(t) = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$

on a single euclidean system with orthogonal basis vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_D$ or with a spatial graph $G = (\mathcal{V}, \mathcal{E})$ ($\mathcal{V}$ for node-set and $\mathcal{E}$ edges set and corresponding mathematical graph theory). One could describe the movements on the spatial graph with a finite Markov chain and stochastic matrices $\boldsymbol{P}$ and the element $P_{i,j} = P(\boldsymbol{s}_i, \boldsymbol{s}_j)$ (see [Sargent, J. Thomas et. al., 2022] for mathematical concept). Figure 4.1 shows four pictures with a yellow non-regular grid. The environment could be described as regular- (transparent lines with grey color) or non-regular grid (yellow grid) with disjoint patches $\mathcal{X}_i \cap \mathcal{X}_j = \forall i \neq j$. A regular grid might be easier for mathematical application (because of equally placed patches). For example for a set $||\boldsymbol{x}||_\infty < L/2$ and with ∞-norm and radius $L/2$ (from traversing the borders length L) and a regular grid with equally placed nodes inside set ($n \times n$ columns and rows). We would get a small patch with length $l = \frac{L}{2*n}$. If we assume a square with $L = 20m$ where a person could move, we will get a length for $2m$ ($n \times n = 5 \times 5$ equally 25 placed nodes). This parameterization is inaccurate, so we would place more nodes to get a more dense spatial network and increase the n. Depending on n we might change the quality of the representation by increasing n by getting more complex stochastic matrices $\boldsymbol{P}$. The resulting probability matrix increases with mathematical complexity $O(n^2)$ (compare table 3.1 and considering Markov property). Each node would represent a subset of the initial set (cell representation). However, on the other side, it might not be adequate for real urban environments. That is why we use a non-regular grid (top left picture). We connect the patches via a topology $\mathcal{T}$ and project all patches on the plane so that the atlas $\mathcal{A}$ completes the manifold description $\mathcal{M} = (\mathcal{X}, \mathcal{T}, \mathcal{A})$. The grid could be dependent on the environmental structure. The two trajectories (blue and red) show some pieces of evidence of how pedestrians could walk in the environment. A problem with the spatial graph is that it does not fit well to describe both trajectories adequately. Elements of the trajectory $\tau_{k_r} \in \mathcal{T}$ (tuple of timestamp k_r and position $\boldsymbol{p}_{k_r} = (p_{x,k_r}, p_{y,k_r})$) might be classified to a node i if it is element of the patch $\tau_{k_r} \in \mathcal{X}_i$. Two trajectories $\mathcal{T}_1$ (blue) $\mathcal{T}_2$ (red) could be represented on the top right picture by a spatial graph (bottom left picture). With a set of nodes $\mathcal{V} = \{\boldsymbol{v}_A, \boldsymbol{v}_B, \boldsymbol{v}_C, \ldots, \boldsymbol{v}_I\}$ and index set $\mathcal{I} = \{'A', 'B', 'C', \ldots, 'I'\}$ on the bottom right picture, and the corresponding patches $\mathcal{X} = \{\mathcal{X}_0, \mathcal{X}_1, \ldots \mathcal{X}_8\}$ we can represent the spatial graph on the bottom left picture. This example in figure 4.1 shows some interesting issues in encoding spatial movements with spatial graphs. The underlying mesh is not very dense, so that both trajectories might have the same indexes (figure 4.1). The blue one has a similar order with "A, D, E, B, C, F, I" as the red one, "A, D, E, B, C, F, E, F, I." Some local information about the coding approach with spatial gets lost. For example, the (U-turn of the red trajectory) inside will not be decoded. The coded information suggests that both trajectories are similar. The error could be immense for a spatial graph by falsely placing the nodes. The computational complexity increase with the node size. For a regular grid by $n \times n$ for a ball with patch as a hyper ball $\mathcal{B}_n(r) = \{\boldsymbol{x} | ||\boldsymbol{c} - \boldsymbol{x}||_n < r, \boldsymbol{x} \in \mathbb{R}^D\}$. Instead of using a spatial graph, we want to embed a dynamic system inside a manifold. We adapt the formulation of formula 3.22 for a local version reachability analysis ([Althoff, 2010]) and embed it on a manifold $\mathcal{M}$ for a certain patch with label q:

$$\begin{gathered} \mathfrak{R}_{\mathcal{X}_q} * \mathcal{X}_{k_i} := \\ \{\boldsymbol{x}(r) = \int_0^r \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \theta) dt | \\ \boldsymbol{x}_{k_i} \in \mathcal{X}_{k_i} \subset \mathcal{X}_q, \boldsymbol{u}([0,r]) \in \mathcal{U}_q, \theta([0,r]) \in \mathcal{P}_q\} \end{gathered} \tag{4.1}$$

If we have knowledge $\mathcal{K}$ from measurements, we could use it to estimate and predict the most probable values for $\hat{\boldsymbol{x}}_{k_i}, \hat{\boldsymbol{u}}([0,r]), \hat{\theta}([0,r])$ for the parameter set and the form and structure of $\hat{\mathcal{X}}_q, \hat{\mathcal{U}}_q, \hat{\mathcal{P}}_q$. We convert the approach to a local and adaptive funnel (hull, or adaptive RA) prediction approach. We could also embed the probabilistic version with the computation of the Chapman-Kolmogorov equation, if we would assume for simplicity Markov property on a local path:

$$\begin{gathered} \mathfrak{P}_{\mathcal{X}_q} * \mathcal{X}_{k_i} := \\ \{p(\mathrm{x}_{k_i} | \mathrm{x}_{k_{i-2}}) = \int_{-\infty}^{\infty} p(\mathrm{x}_{k_i}, \mathrm{x}_{k_{i-1}}) p(\mathrm{x}_{k_{i-1}}, \mathrm{x}_{k_{i-2}}) d\mathrm{x}_{k_{i-1}}\} \end{gathered} \tag{4.2}$$

Figure 4.2 shows the simplified and programmed Julia simulator with different modules. The simulator was programmed with a signal-slot architecture that offers discrete events for different modules to ensure high flexibility and performance. The interested reader might take a look on the implementation in section C.1 for additional details. The theoretical focus is on the adaptive hull computation, causal inference for intention change, and the

Markov-Decision Process (MDP). The main focus is on handling pedestrian trajectories. However, interaction with the motion planning of the vehicle plays a role in this section. The model predictive control (MPC) algorithm for the vehicle is not the main focus of the thesis (see section 1.4). The cost-function is simply set the p-norm distance $J = ||\boldsymbol{x}_k^v - \boldsymbol{x}_k^v||_p$. It offers a simple way to compute planned trajectories for the vehicle. Also, the control input could be incorporated for the discrete time-steps $\{k_i, k_j\}$:

$$\boldsymbol{x}_{k_r}^*, \boldsymbol{u}_{k_r}^* = \arg\min \sum_{r=i}^{j} J(\boldsymbol{x}_{k_r}, \boldsymbol{u}_{k_r}) \text{ subject to} \tag{4.3}$$

$$\text{a) Vehicle dynamics (kinematic single-track), see section 2.3.1} \tag{4.4}$$

$$\text{b) Static obstacle avoidance (lane keeping)} \tag{4.5}$$

$$\text{c) Dynamic obstacle avoidance (pedestrians)} \tag{4.6}$$

The single-track model has some nonlinearities (steering- and acceleration constraints) with $\dot{\boldsymbol{x}}_{k_r} = \mathbf{f}(\boldsymbol{x}_{k_r}, \boldsymbol{u}_{k_r})$. The complexity of the environment for static and dynamic obstacle avoidance could lead to some difficulties. From which side should the pedestrian be bypassed. With a Mixed Integer Linear programming approach, one could introduce other variables. Simple environments like a straight street could be incorporated by some inequalities and define some hyperplanes ($\boldsymbol{x} <= \boldsymbol{Ab}$). We could incorporate obstacle avoidance constraints for a pedestrian like in figure 4.3. Seven sets $\mathcal{X}^p 1 : 7_{k_r}$ represented by support functions (see arrows). The support function $\rho(\boldsymbol{m}, \mathcal{X}_{k_r}^p)$ for the first pedestrian is visualized by a blue hyperplane in direction to the vehicle (by vector $\boldsymbol{m}$). Further difficulties could be that the solver cannot find a local/global optimal solution. There could be a lot of different situations where the solver cannot find a solution. Especially when there are many non-convex constraints. One could linearize vehicle dynamics around a knot-point $\boldsymbol{x}_{k_r}^{v,*}$ (see 2.3.1) with the Jacobi-Approximation based on the current state of the vehicle $\boldsymbol{x}_k^v$ to get a linear system on the knot point or reformulate the vehicle model compare [Pek, 2020]. With the linearization we get a linear system $\boldsymbol{x}_{k_r} = \boldsymbol{A}\boldsymbol{x}_{k_r} + \boldsymbol{B}\boldsymbol{u}_{k_r}$. Similar to the extended Kalman filter we use this linearization we compute all possible future sets by computing the reachability analysis $\mathcal{R}_{k_r}^{v,*}(\mathcal{X}_{k_r}^{v,*}, \mathcal{U}_{k_r}^{v,*})$, $\boldsymbol{x}_{k_r}^{v,*} \in \mathcal{X}_{k_r}^{v,*}$ for the current knot point. For example, the structure of the urban environment could be incorporated. In a basic example, on a straight street, the allowed configuration space of the vehicle is divided by two hyperplanes representing the dimensions of the street. The reachability analysis could be incorporated into the optimization. The mathematical constraints hold the vehicle on the lane (static road leads to static constraint). The car does not collide with the future pedestrian sets (walking pedestrian leads to dynamic constraints). The computation of a model predictive control approach and underlying optimization leads to future vehicle states and control inputs. The algorithm does not reach the global optima because of the nonlinearities and the approximation errors. It performs optimally in only some subregions of the control state-space. We used the package [Dunning et al., 2017] which gives the optimality status in each moment. It is not the goal of the simulator to enhance the optimization strategy of the motion planning. It is not trivial to incorporate dynamic obstacle avoidance with the pedestrian. One way is to use the current distance between the vehicle and pedestrian $d_{k_r}^{v,p}$ and incorporate it into the optimization process. The focus of the thesis is to propose new pedestrian models for dynamic obstacle avoidance (pedestrians). We highlight only some selected details and skip further mathematical details and refer to the actual implementation on Github™. To see the principle idea compare figure 4.3. Between the current state of the vehicle $\boldsymbol{x}_{k_r}^{v}$ one could define a dashed blue line for the linear constraint to the observed pedestrian with label $p1$ at k_r. The seven pedestrian sets $\mathcal{X}_{k_r}^{p1:7}$ are defined by support vectors defining the border of the set. In order to get an optimization, an approach of convex optimization with realtime-capability compares [Pek, 2020] with reachability analysis. Nevertheless the simulation results offer realistic behavior in figure 4.5 and compare section C.1 for more simulations and further details on the used packages. The first point is to enhance classical reachability analysis with capabilities to adapt to the current situation. Similar to extrapolation of the future trajectory. One could predict the future state sets of a pedestrian $\hat{\mathcal{X}}_{k_i}^{p}$ (for example by computing the future support functions compare figure 3.47). Classical reachability analysis could be used for worst-case scenarios of computing all plausible physical future positions. In formula 2.29 and 2.30 we showed the recursive formulation of reachability analysis for a linear system:

$$\mathcal{R}_{k_{i+1}} = \underbrace{\textbf{state-dependent-operation}(\mathcal{R}_{k_i})}_{\text{fixed group action on homogeneous solutions}} \underbrace{\oplus\ \textbf{control-input-dependent set}}_{\text{fixed group action on particular solutions}} \tag{4.7}$$

The package of [Bogomolov et al., 2019] offers different algorithms for solvers for the basic reachability analysis computation. The reach state set volume for conventional computation might get very large for a worst-case scenario. Instead of using a maximal velocity and acceleration for all situations, one could use machine learning (compare [Hartmann and Watzenig, 2019a]) to make adaptive belief set computations. The motion planning of a vehicle gets less conservative, and it might be the case that the vehicle gets faster to the target. However, on the other hand, it is also riskier. The implementation of MDPs is done with the package [Egorov et al., 2017]. The locomotion of pedestrians could be modeled with Markov Decision Models (MDPs) with 4.2. The package [Egorov et al., 2017] offers different solvers, and for the performance of two algorithms, we computed the training time. Table 4.2 shows a benchmark evaluation for the simulation time for a simple Markov Decision Process (MDP) approach with the package [Egorov et al., 2017]. The MDP tries to maximize the expected reward with a continuous system $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$. As described one could model the pedestrian movement as a MDP or POMDP, compare figure 3.57 and 3.58. The Julia-POMDPs package [Egorov et al., 2017] offers a different set of solvers for MDP and POMDPs. The simple benchmark example of a six-dimensional dynamic system compare formula 3.75 should learn to get the highest rewards to reach a certain area. This benchmark example shows that the Monte Carlo Tree Search (MCTS) solver (original paper [Kocsis and Szepesvári, 2006]) with static arrays. This algorithm is faster than the costly local approximation value iteration solver with grid interpolation. Both solvers are adequate for continuous state spaces, but the MCTS solver is adequate for online optimization problems. Only the random and stochastic policies are faster than the MPC algorithms. A simple double integrator $\boldsymbol{x}_{k_{i+1}} = \boldsymbol{x}_{k_i} + T_S(\boldsymbol{A}\boldsymbol{x}_{k_i} + \boldsymbol{B}\boldsymbol{u}_{k_i})$ is used to sample the control input $\boldsymbol{u}_{k_i} \sim U(\boldsymbol{l}, \boldsymbol{u}) \subset \mathcal{B}(r)$. The $U(\boldsymbol{l}, \boldsymbol{u})$ uniform stochastic process produces samples between the lower $\boldsymbol{l}$ and upper $\boldsymbol{u}$ level vector. The agent travels in different random directions with random velocity in the random policy. The acceleration vector is sampled from a fixed distribution in a stochastic policy.

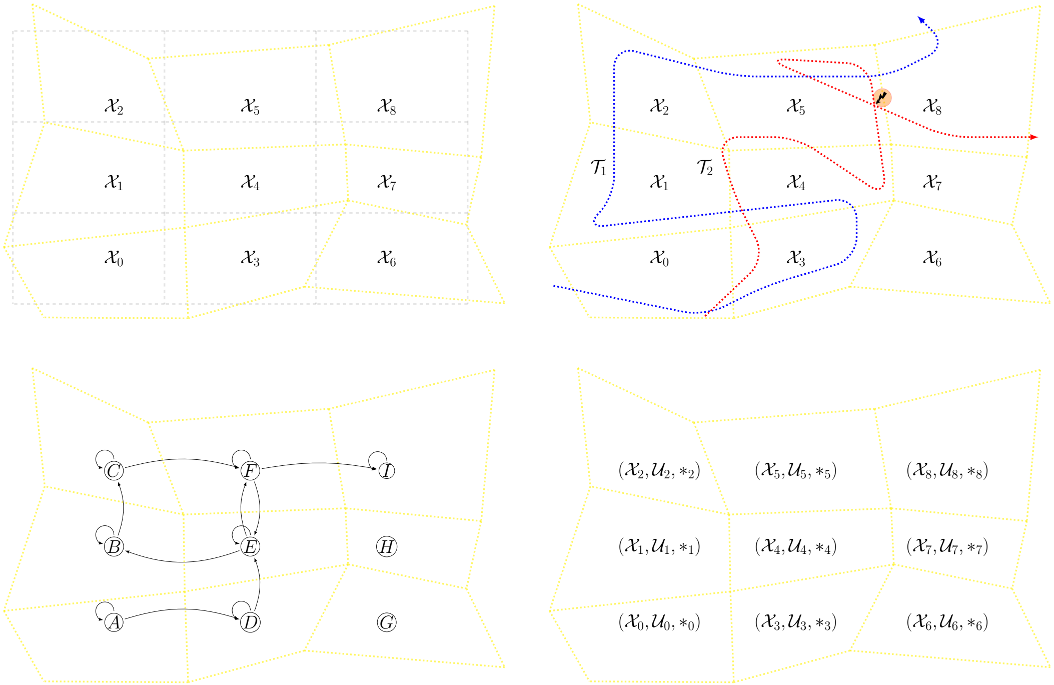

$\mathcal{X}_2$
$\mathcal{X}_5$
$\mathcal{X}_8$
$\mathcal{X}_1$
$\mathcal{X}_4$
$\mathcal{X}_7$
$\mathcal{X}_0$
$\mathcal{X}_3$
$\mathcal{X}_6$
$\mathcal{T}_1$
$\mathcal{T}_2$
C
F
I
B
E
H
A
D
G
$(\mathcal{X}_2, \mathcal{U}_2, *_2)$
$(\mathcal{X}_5, \mathcal{U}_5, *_5)$
$(\mathcal{X}_8, \mathcal{U}_8, *_8)$
$(\mathcal{X}_1, \mathcal{U}_1, *_1)$
$(\mathcal{X}_4, \mathcal{U}_4, *_4)$
$(\mathcal{X}_7, \mathcal{U}_7, *_7)$
$(\mathcal{X}_0, \mathcal{U}_0, *_0)$
$(\mathcal{X}_3, \mathcal{U}_3, *_3)$
$(\mathcal{X}_6, \mathcal{U}_6, *_6)$

Figure 4.1.: Trajectories on a manifold and graph

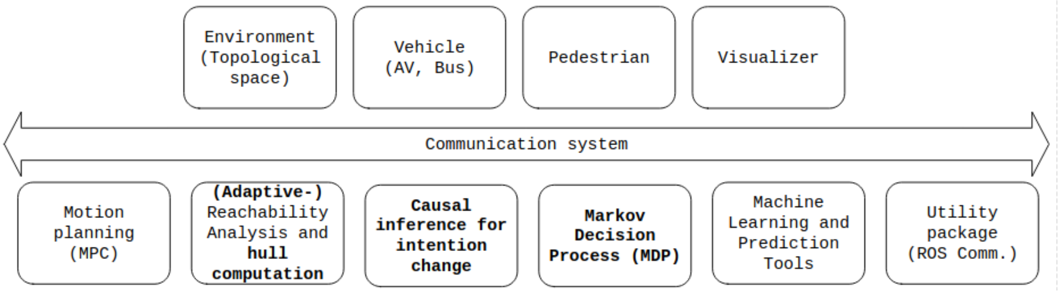

Environment (Topological space)
Vehicle (AV, Bus)
Pedestrian
Visualizer
Communication system
Motion planning (MPC)
(Adaptive-) Reachability Analysis and hull computation
Causal inference for intention change
Markov Decision Process (MDP)
Machine Learning and Prediction Tools
Utility package (ROS Comm.)

Figure 4.2.: Simulation program in Julia programming

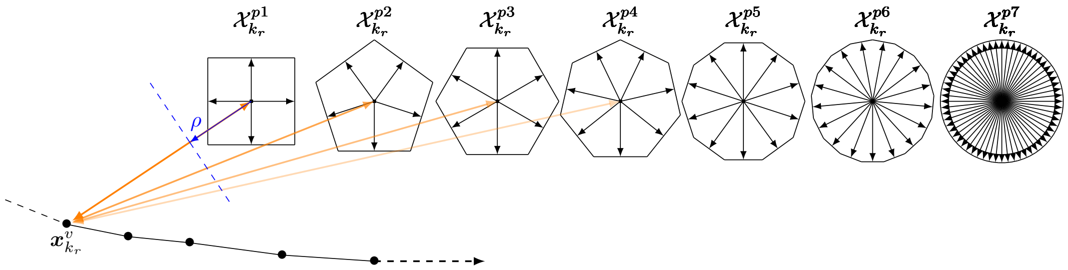


Figure 4.3.: Obstacle avoidance constraint for a current situation

Simulation	Value	Unit
MCTS Solver with Static Arrays (SA)	371.313	ms
Local Approximation Value Iteration Solver (grid interpolation)	21.378	s
Random policy	47.931	ms
Stochastic policy	97.985	ms

Table 4.2.: Benchmark simulation with Julia Programming [Bezanson et al., 2017] for some MDP solvers from the Julia-POMDPs package [Egorov et al., 2017]. The training time for local approximation Value Iteration solver (grid interpolation) is bigger than MCTS solver for a simple example. The reward function is set $R = y$ to positive y coordinate for a double-integrator in a constrained state space. Solvers are MDPs for continous systems. This example should only show the difference in training time for a very simple example.

4.2. Visualization of simulations

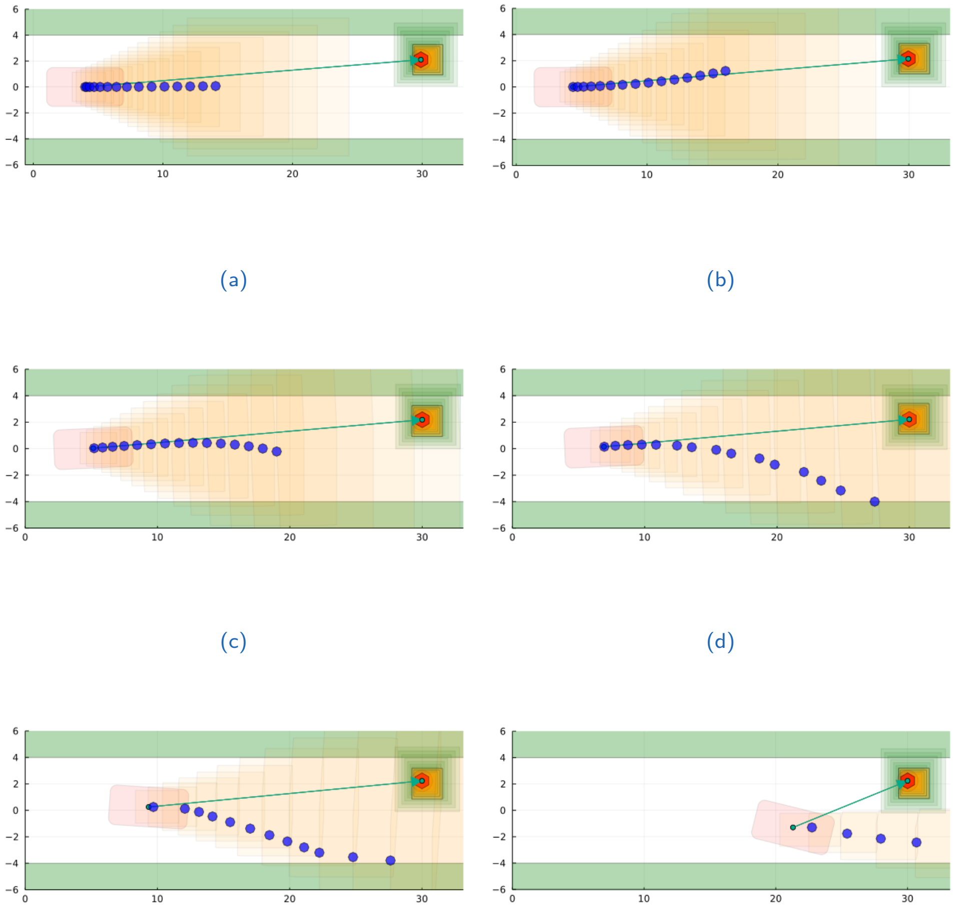
Figure 4.4, C.1 and figure C.2 shows some visualization from the simulator. The pink vehicle set (compare figure 2.16 and figure 4.4) represent a vehicle driving to a certain target state $\boldsymbol{t}$ minimizing the cost function $\mathrm{J} = ||\boldsymbol{x}(t) - \boldsymbol{t}(t)||_2^2$. In [Schratter et al., 2019] we showed that for certain circumstances, the motion planning could drive faster by adapting the reachable sets (pedestrian on the sidewalk). A vehicle is driving to a fixed goal state (behind the pedestrian) optimized with MPC. A pedestrian is waiting (short random walk because small movements can also be measured (noise) when a person is waiting) in a certain area or walking. The reachable sets for the vehicle and the pedestrian are computed and visualized. The reachable sets of the vehicle and the planned trajectories from the motion planning are visualized together. The collision avoidance constraint is activated when the reachable set of the vehicle intersects with the reachable set of the pedestrian $\mathcal{R}^v_{k_r} \cap \mathcal{R}^p_{k_r} \neq$. That is why the vehicle first drives in the direction of the pedestrian. After recognizing the pedestrian, it bypasses it. The green arrow simply shows the constraints for the MPC algorithm (compare 4.3). The constraint is not optimized to the reachable set, which could be done in future research. There are some further issues in optimizing the MPC approach, which was not the focus of this thesis. The performance is not always optimal. There are many reasons for that. The unintended movements of the pedestrian might influence the possible state of space. For example, a fast oncoming vehicle might not find any optimization result when the pedestrian jumps on the street, where a collision is not avoidable. Fortunately, the results have an acceptable and reasonable performance like in figure 4.5, but these problems have to be tackled in the future (finding robustness and trustable of the MPC approach). The used packages for this simulation and their references can be found in the yml-files for each simulation run. The vehicle model from section 2.3.1 was programmed in Julia. For each simulation cycle k, we approximate the nonlinear system to a linear system by computing the Jacobian using the tool [Robot Exploration Lab at Carnegie Mellon University, 2022]. Other important packages are [Egorov et al., 2017] for MDP simulation, [Bogomolov et al., 2019] for reachability analysis, [Tavares et al., 2021] for causal inference and [Dunning et al., 2017] for the MPC computation of the vehicle. The states of the vehicle and the pedestrian are also visualized in figure 4.5 (and figure C.4 for the simulation in the appendix). The vehicle and pedestrian position, velocity, and acceleration are also presented with the Euclidean distance between the vehicle state and pedestrian state. A better approach would be to present the minimal distance between these two sets over time or intersection computation to ensure collision avoidance (compare figure C.6). The status signal figure C.4 is set to 1.00 because an intervention influenced the pedestrian behavior with the causal do-operator (compare section 4.3).

4.3. Causal inference and goals for the agents

In many prediction models of pedestrians, one assumes a pedestrian fixed intention. The intention might change by an event. Therefore causal inference seems adequate to model superficial causal relationships. This section shows causal inference (adapted with chapter 1 from [Peters et al., 2017]) as a technique to model intention change for pedestrians. We present two options of dynamic systems with causal inference. The first approach is when the causal inference sets the goal state in the initial step. The second approach is when the goal state (or the control input when a dynamic system is modeled) and the intention change over time. The state of the art is discussed in section 2.2.3. We have four different distributions for causal inference. The initial-, conditional-, interventional-, counterfactual distribution. For example we might have the observed data of pedestrians of the current position $(x_{k_r}, y_{k_r})\ \forall r \in \{i, i+1, \dots, j\}$ and future position after N steps:

$$(\underbrace{x_{k_i}, y_{k_i}}_{\mathsf{x}_i}, \underbrace{x_{k_{i+N}}, y_{k_{i+N}}}_{\mathsf{y}_i}), \dots, (\underbrace{x_{k_j}, y_{k_j}}_{\mathsf{x}_j}, \underbrace{x_{k_{j+N}}, y_{k_{j+N}}}_{\mathsf{y}_j}) \tag{4.8}$$

$\mathsf{x}_r, \mathsf{y}_r$ are realisations of the random variables $\mathsf{X}, \mathsf{Y}\ \forall r \in \{i, i+1, \dots, j\}$ If we are looking for the expectation $\mathbf{f}(\boldsymbol{x}) = \mathbb{E}[\mathsf{Y}|\mathsf{X} = x]$ in form a regression model. The future position might give a piece of measurable information that could lead to inferring knowledge to encode the intention of a pedestrian. This approach might have some drawbacks because the cognitive nature of the decision-making process is hidden. Nevertheless, the idea might



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.4.: A vehicle (represented by a pink rectangle) with planned trajectory (blue dots) and orange reachable sets is driving to a fixed target state. A pedestrian is waiting on the right, represented by a red hexagon. The pedestrian makes small movements modeled with noise so that the resulting reachable sets are not symmetric. The goal state of the vehicle is behind the pedestrian at the position, so that vehicle cannot drive directly to the goal state. The reachable sets are over-approximated as rectangles. Other forms like octagons or sets by support functions could further be tested. In this example, only a classical reachable set computation was used by setting maximal acceleration, and velocity [Liu et al., 2017].

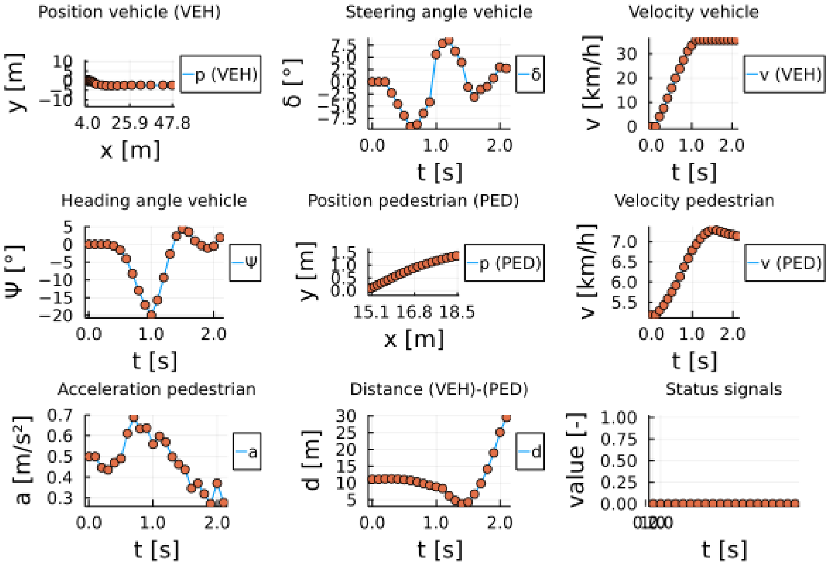


Figure 4.5.: Each simulation run has the state-change of the pedestrian and vehicle (constrained position, velocity and acceleration and control inputs)

bring some simple tools for designing pedestrian models. We might take a selection from the dataset to query a set of realizations for specific conditions. For example, if we observe the movements of a specific area, where the pedestrian may come. If we use a force to change the dataset by intervention with the do−operator and the resulting interventional distribution $\mathrm{P}(\mathrm{do}(\cdot))$. We could formulate the counterfactual probability distribution with causal inference by conditioning the interventional distribution. If we took the force to change the reality, what kind of observations might have taken place. The lack of knowledge and the diverse and complex nature of human decision making might make the formulation $\mathbf{f}(\boldsymbol{x}) = \mathbb{E}[\mathrm{Y}|\mathrm{X} = x, \mathcal{H}$[1]$]$ nontrivial. There might be no single solution if not all information is measurable. Rather the must be a set of possible directions where a pedestrian might go. We might adapt the tools for causal inference for modeling the random target for a group of virtual agents without any dependency on time. Alternatively, suppose we select a single agent. In that case, we compute different targets, control inputs, or other models to model the change in relationship with

[1] $\mathcal{H}$ might represent unknown/hidden and dynamic information which are not independent and identically distributed and change the resulting distribution

time. Figure 4.6 shows an example of different target distributions. On the side, the marginal distributions for x- and y-values are plotted for further understanding. The initial distribution $P(\mathsf{X}|\mathsf{Ber})$ is set to define four different targets with different probabilities. We use two Bernoulli distributions $\mathsf{Ber}(\cdot)$ to model the side-selection of the person $\mathsf{Ber}(\rho_{\text{side}})$ (if it is true the agent goes to the upper sidewalk otherwise bottom sidewalk and the ρ_{side} as the parameter of the Bernoulli function) and the x-position (left or right). The Bernoulli-distributions has two binary outputs. From the distribution we sample the target position with uniform distributions $\mathsf{Uniform(a,\ b)}$. The probability density function for a Bernoulli function is [Bishop, 2006]:

$$f(x|\rho) = \begin{cases} \rho^x (1-\rho)^{1-\rho} \text{ for } x = 0, 1,\ \rho \in [0,1], \\ 0 \text{ otherwise} \end{cases} \tag{4.9}$$

For the continuous uniform distribution the probability density function [Bishop, 2006] is:

$$f(x) = \begin{cases} \frac{1}{b-a} \text{for } a \leq x \leq b, \\ 0 \text{ otherwise} \end{cases} \tag{4.10}$$

We could have used other distributions to model the target selection of a pedestrian like normal distributions or others. It is reasonable that such simple distributions could model real human behavior (causal structural learning). Nevertheless, one could use this approach to generate data to test the functionality of autonomous vehicles. Moreover, the approach benefits from interpreting the data for real urban environments. One could model the distribution for the likelihood that a pedestrian has a preference for the supermarket, the park, and other areas (compare section 3.5). We could sample the target position $\boldsymbol{p}^{\text{target}}$ with x, y coordinates from the target distribution $\boldsymbol{p}^{\text{target}} = (x, y) \sim (\mathsf{X}_{\text{pos}}, \mathsf{X}_{\text{side}})$. For the x coordinate we get:

$$x \sim \mathsf{X}_{\text{pos}} = \begin{cases} \mathsf{Uniform(8.0,\ 12.0)} \text{ if } b \sim \mathsf{Ber}(\rho_{\text{pos}}) \text{ is true (left-position)} \\ \mathsf{Uniform(18.0{,}22.0)} \text{ otherwise (right-position)} \end{cases} \tag{4.11}$$

And for the y coordinate we get:

$$y \sim \mathsf{X}_{\text{side}} = \begin{cases} \mathsf{Uniform}(4.0,\ 6.0) \text{ if } b \sim \mathsf{Ber}(\rho_{\text{side}}) \text{ is true (upper sidewalk)} \\ \mathsf{Uniform}(\text{-6.0,-4.0}) \text{ otherwise (bottom sidewalk)} \end{cases} \tag{4.12}$$

The conditional distribution is inferred from the initial distribution. We select a set of samples with certain conditions to answer the question based on the observed data. For example in figure 4.6b we take all the sample to ask how is the probability changed if we observe from the initial distribution only the agents, who have their goal on the upper sidewalk (green) $P(\mathsf{X}_{\text{pos}}, \mathsf{X}_{\text{side}} | \mathsf{X}_{\text{side}} > 0.0)$. The interventional distribution manipulates the initial distribution so that we can set new targets (and situations which could not performed in reality, e.g. testing a collision avoidance algorithm with real pedestrians). Figure 4.6c changes the x position

$$\begin{gathered} x \sim \mathsf{X}_{\text{pos}}(\text{do}(Uniform(8.0, 12.0) => 3.0)) = \\ \begin{cases} \text{do}(3.0) \text{ if } b \sim \mathsf{Ber}(\rho_{\text{pos}}) \text{ is true (left-position)} \\ \mathsf{Uniform}(18.0,22.0) \text{ otherwise (right-position)} \end{cases} \end{gathered} \tag{4.13}$$

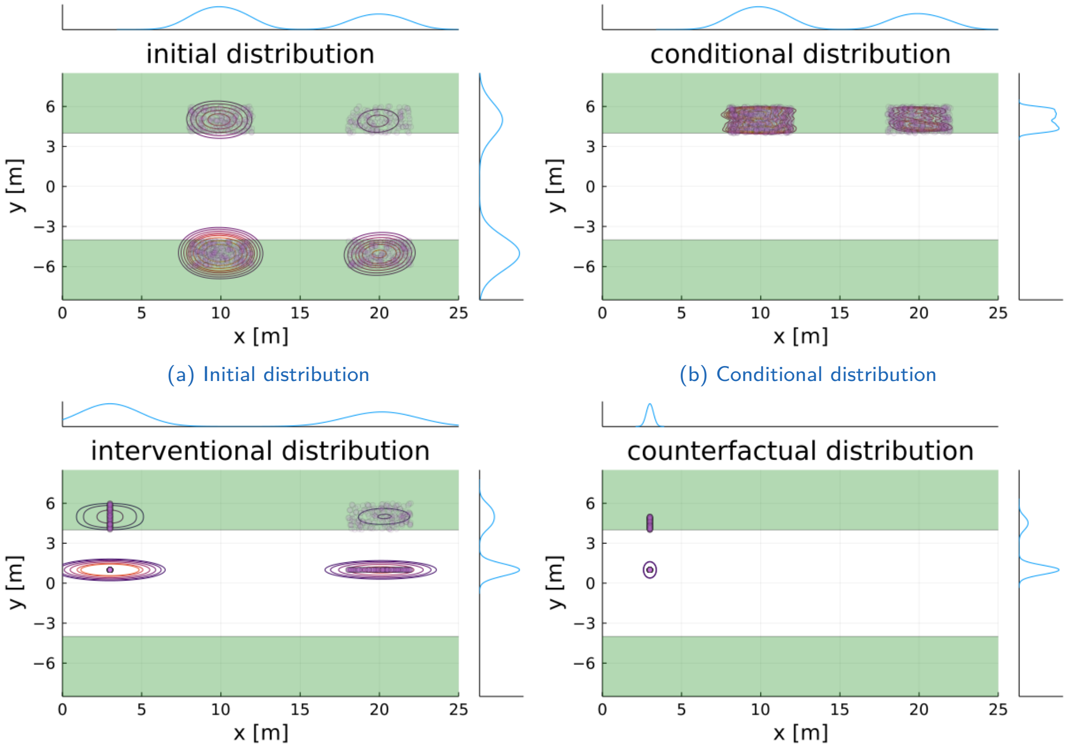
If the Bernoulli distribution is true, the agents are "forced" to go to the $x = 3$ position. Otherwise, the distribution is unchanged. Another intervention is done with the y coordinate:

$$\begin{gathered} y \sim \mathsf{X}_{\text{side}}(\text{do}(Uniform(-6.0, -4.0) => 1.0)) = \\ \begin{cases} \mathsf{Uniform}(4.0,\ 6.0) \text{ if } b \sim \mathsf{Ber}(\rho_{\text{side}}) \text{ is true (upper sidewalk)} \\ \text{do1.0 otherwise (bottom sidewalk)} \end{cases} \end{gathered} \tag{4.14}$$

The counterfactual distribution takes the interventional distribution as a reference (conditioning). It is the conditional distribution of the interventional distribution. Figure 4.6d shows the probability:

$$P(\mathsf{X}_{\text{side}}(\text{do}(Uniform(-6.0, -4.0) => 1.0)) | \mathsf{X}_{\text{side}} < 5.) \tag{4.15}$$

We observe the date from the interventional distribution and select all samples for y to coordinate smaller than the value 5.0. The same for the x-coordinate. The probability $\mathsf{X}_{\text{pos}}(\text{do}(Uniform(8.0, 12.0) => 3.0) | \mathsf{X}_{\text{pos}} <=$



(a) Initial distribution

(b) Conditional distribution

(c) Interventional distribution

(d) Counterfactual distribution

Figure 4.6.: Example for initial-, conditional, interventional- and counterfactual distribution to sample the target of an agent

8.). We observe only the two left intervened data clusters. See how the marginal distributions for the x-coordinate have changed, while the small mode for they changed only a bit. These distributions will be used (under different configuration, other parameters and distributions but the same concept) in the simulator in section 4.5. The initial-, conditional, interventional and counterfactual distribution will generate the target position. A Markov Decision Process will find the policy and how to find the optimal policy to come to the target.

4.4. Connected reachable sets topological spaces

Figure 4.7 shows the reduction for different values set for the maximal velocity $v_{\max}$ and acceleration $a_{\max}$. Figure 4.7 presents the cumulative area A_c of reach sets projected on the two-dimensional ground by the projection matrix $\boldsymbol{P}$. Each timestamp k_i the projected reachable sets on the two-dimensional plane, maximal velocity $v_{\max}$ and acceleration $a_{\max}$ are computed (in form of a function with shorthand $\mathcal{R}(v_{\max}, a_{\max}, k_r)$) for the time interval $\{k_i, \ldots, k_j\}$:

$$A_c := \sum_{r=i}^{j} \mathbf{area}(\boldsymbol{P} \cdot \mathcal{R}(v_{\max}, a_{\max}, k_r)) \tag{4.16}$$

This chapter aims to show to connect reachable sets (successive computation of reachable sets over time). A reachable set $\mathcal{R}_{a,k_i,k_r}$ for a time interval $[k_i, k_r]$ is connected $\mathcal{R}_{b,k_r,k_j}$ if there is subset $\mathcal{R}_{a,k_r} \cap \mathcal{R}_{b,k_r} \neq$ at time k_r. In section 3.5 it was mentioned that we could use different techniques for movements and incorporate the environment. Figure 4.9 shows connected reachable sets starting in the center. The green reachable sets show the first time horizon between $[t_1, t_2]$, the second time horizon $[t_2, t_3]$ in red, the third $[t_3, t_4]$ in yellow and the fourth reachable set in blue $[t_4, t_5]$. The last set of each flow pipe (set of reachable sets over time) has 30 sample points. One sample was selected randomly instead of connecting the whole flow pipe to another flow pipe. At the next point, the next flow pipe starts. The first two reachable sets in green and red in the right picture have nearly the same appearance as the left picture. The third and the fourth reachable sets have half of the allowed maximal velocity and acceleration. It shows that the cumulative areas of yellow and blue flow pipes are much smaller than the first two flow pipes. Compare green and red flow pipes in appearance with the yellow and blue in the amount of area. This simple example should show that the maximal velocity and acceleration parameters might enormously impact the resulting area, which is forbidden for the autonomous vehicle. Be aware that the sampling process on the left and right sides is not the same. The velocity direction could also be different (the sampling process on the left picture

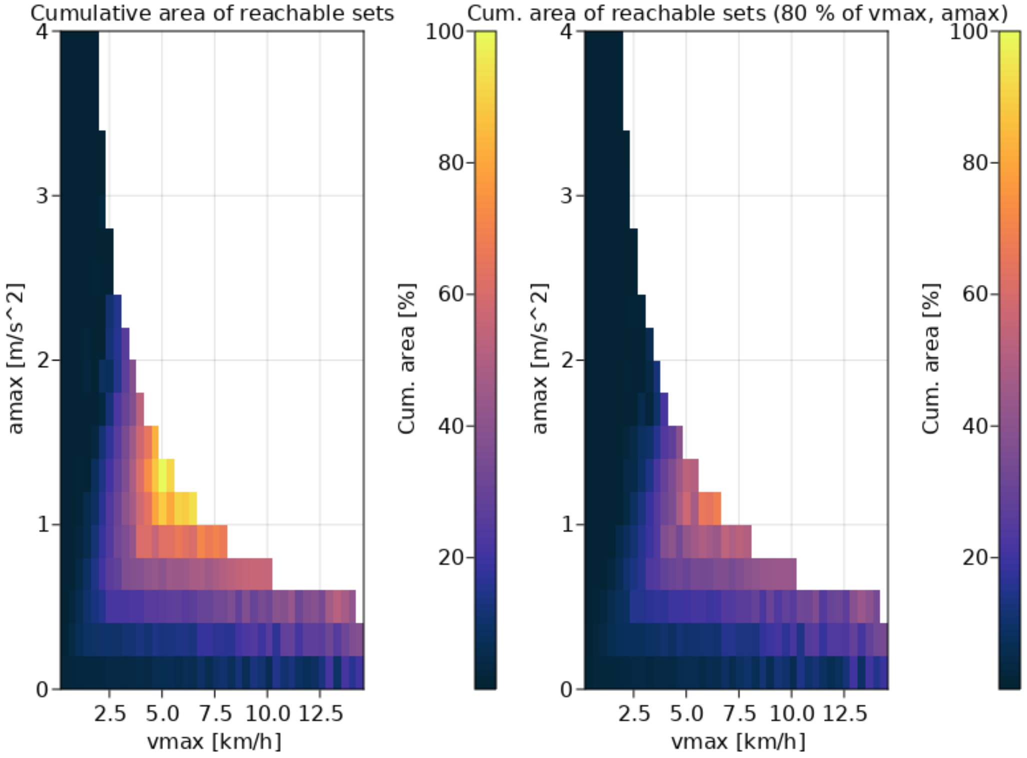


Figure 4.7.: Cumulative area of reachable sets for a specific time horizon. A reduction of the maximal parameters could lead to a significant reduction in the cumulative area. Only for small velocities, a bigger acceleration was computed. This example is not validated with experiments from real persons. Persons might have different physical conditions. Also, the curve is computed randomly. It should highlight that persons could accelerate in the same way under high velocities with the same conditions. It makes no sense that a person has high acceleration and velocity without technical tools (electro-scooter).

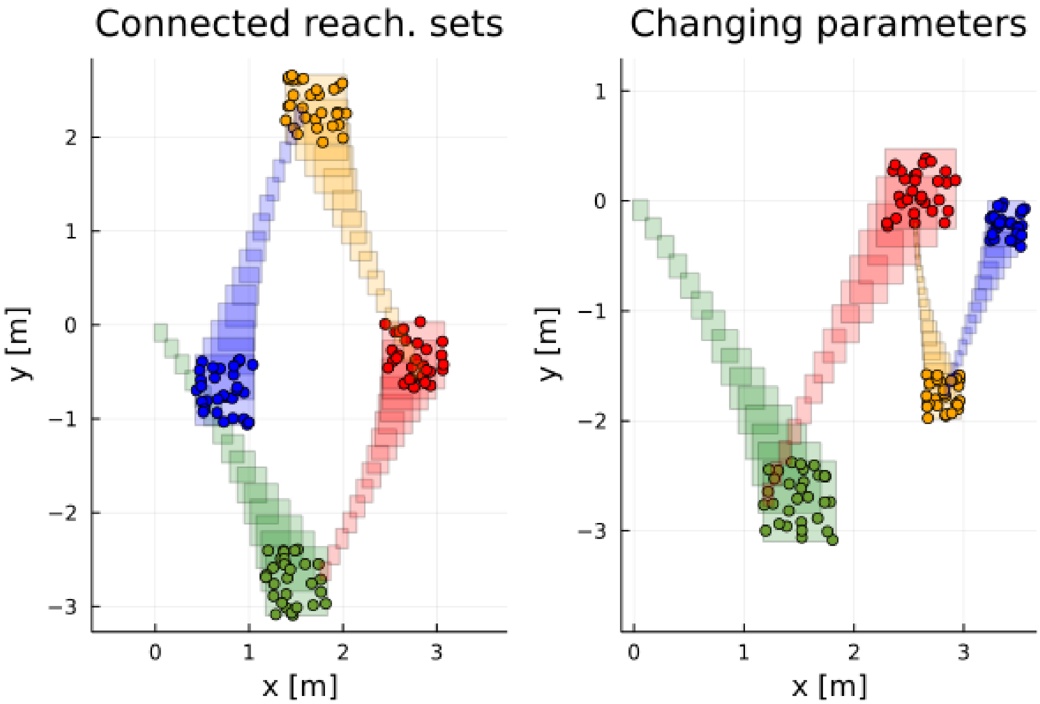


Figure 4.8.: Connected reachable sets starting at the center point (0,0) with the green reachable sets. The red reachable set is connected with the green one. From there it goes to the yellow and blue reachable sets. The sampling process on the right side is different on both pictures.

Figure 4.9.: Connected reachable sets

is different from the right picture, so the form and direction are different). This change is why all the flow pipes of the same color have different forms, and the yellow and blue ones on the right side also have different directions. In [Hartmann and Watzenig, 2019a] prediction models were used to predict the meta-parameters. This example from figure 4.9 shows also possibility to model reachable set in topological spaces $\mathcal{X}_i \; \forall \; i \; \in \; \mathcal{I}$ compare figure 3.44b. In the street, the pedestrian might walk faster to traverse the street (higher maximal velocity and acceleration) than on the sidewalk. On the other hand, a vehicle might have a causal influence that the pedestrian stops or walks faster, depending on the individual's reaction. This behavior might be modeled by a stochastic process, Bayes Graph, or causal model on the meta-parameters of the reachable sets.

4.5. Evaluation over different pedestrian models

The simulator has different pedestrian models. Model a) and b) are solvers for optimizing MDPs, where a) is faster in learning a new policy than b). c) and d) are stochastic policies where c) has a random white noise and d) is a fixed stochastic policy. Model e) is an affine translation. Model f) is a stochastic policy with a causal intervention on random time k. The random time is modeled with a Bernoulli process. Model g) has random meta parameters on reachable sets (maximal velocity and acceleration). Model h-k combines causal inference and MDP with an MDP MCTS solver. We used the implementation of following packages [Egorov et al., 2017, Tavares et al., 2021] and the combination of the packages in table C.1 for the whole simulator:

a) MDP Monte Carlo Tree Search (MCTS) Solver (fast) [Kocsis and Szepesvári, 2006]
b) MDP Local Approximation Value Iteration (slow)
c) Random policy (random walk for the control input of a double integrator)
d) Fixed stochastic policy (similar to [De Nicolao et al., 2007b], like in formula 2.36)
e) Simple translation with fixed translation vector $\boldsymbol{t}$ and $\boldsymbol{x}_{k_{i+1}} = \boldsymbol{A}_t \boldsymbol{x}_{k_i}$

f) Stochastic policy with causal intervention
g) Random meta parameters reachability sets
h) Causal inference and MDP-MCTS Solver - initial distribution
i) Causal inference and MDP-MCTS Solver - conditional distribution
j) Causal inference and MDP-MCTS Solver - interventional distribution
k) Causal inference and MDP-MCTS Solver - counterfactual distribution

The models from a-e are from the literature. The models e-j are models by considering new kinds of models for pedestrians. The models g-j are using the Markov Decision model from a). The concepts of causal inference (initial-, conditional, interventional- and counterfactual distribution) for modeling of target distributions in section 4.3. For each simulation model, we use 20 different simulations and different initial states for the pedestrian. This way of the experiment should be a kind of stress test for the simulator and different initial positions of the pedestrian. All figures from 4.10, 4.11a, 4.12 have boxplots (salmon-colored) and densities (blue) for each algorithm (a-k). Figure 4.10 shows the evaluation of different pedestrian models (a-k) and their maximal values for velocity and acceleration in a single simulation run. It shows the densities for 20 simulations for each pedestrian model (a-k). Figure 4.11a shows the minimal distance between the state of the vehicle and the pedestrian state (for the sake of simplicity, the euclidean distance was used). Figure 4.11b shows how far the vehicle has come for different situations (x-value). Figure 4.12 shows the different simulation times in total comparison to the maximal simulation time with the Local Approximation Value Iteration (LAVI) solver for MDP. As in table 4.2 shows also that the 4.12 MDP with Local Approximation Value Iteration is very slow. All other algorithms are less than 10 percent faster than algorithm b). The algorithms are solvers for classical MDPs and continuous dynamic systems, where a reward is placed randomly in the pedestrian area. The solvers find an optimal policy for each cycle to compute the next acceleration. A random policy is sampling the next acceleration vector by a random process, whereas the fixed stochastic policy has a fixed probability distribution for sampling (compare formula 2.36, illustration 3.25a from [De Nicolao et al., 2007b] as a reference). The simple translation is only iterative computation of formula 3.51 and angle $\theta =$ constant. In model f, the distribution for generating the control input of the system is changed (changing the intention of the intention by do-operator in causal inference) after a random

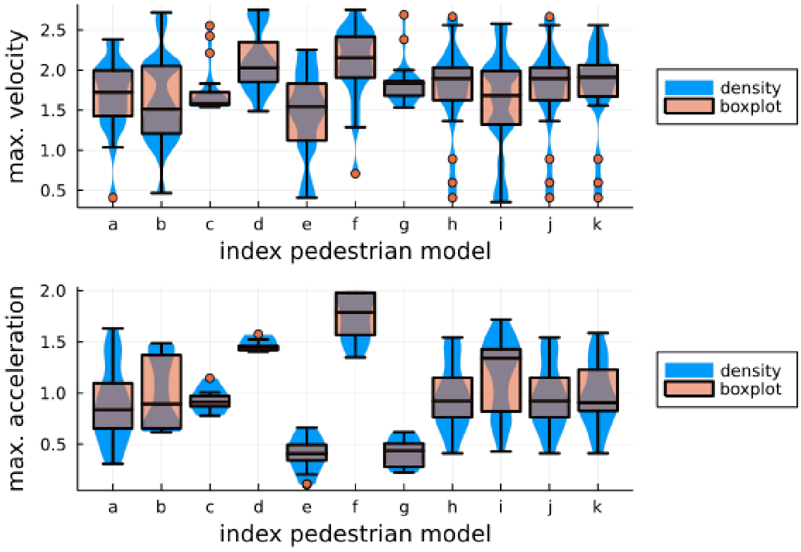
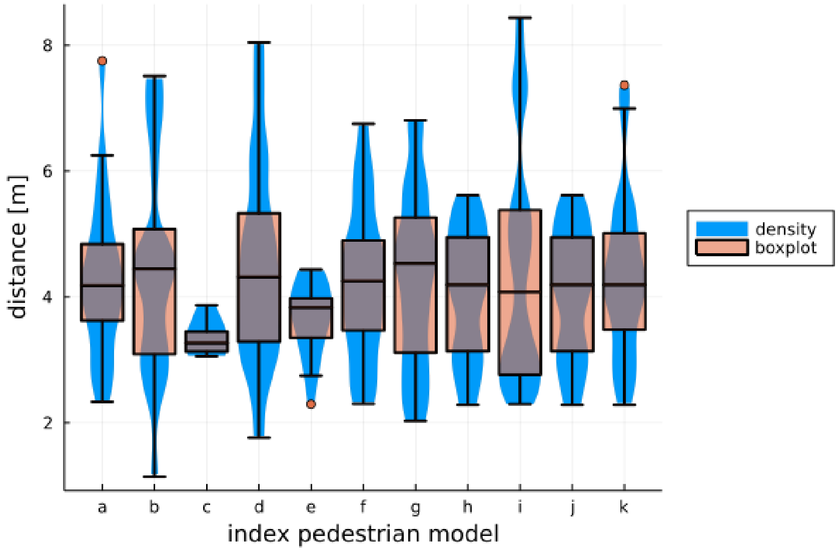
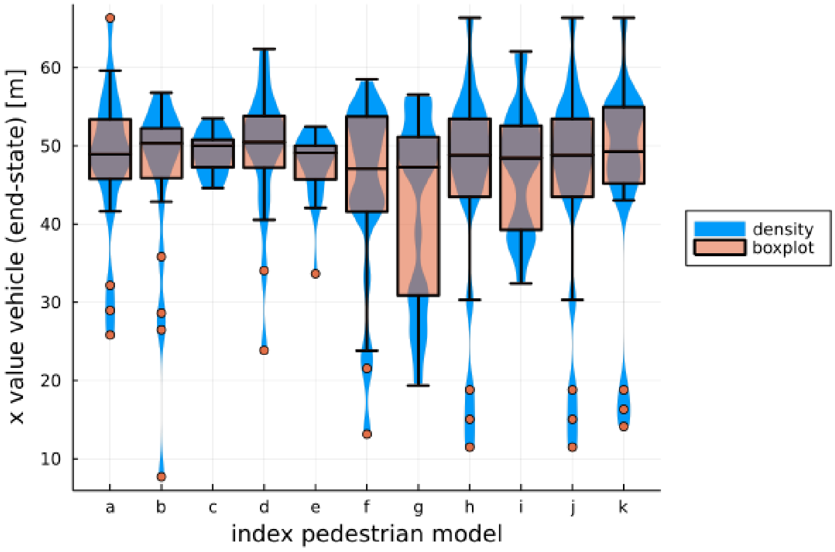


Figure 4.10.: Maximal velocity and acceleration for different pedestrian models (a-k)

amount of time (modeled with a Bernoulli distribution, compare figure 3.54). The meta-parameters are set randomly in model g by a stochastic process (compare figure 4.9). The models, "h-k," model a) with the MDP-MCTS solver were used to reach a certain goal with a Markov Decision Process. The sampling of the target goal (offline, sampling before the use of MDP) was done with the initial distribution in "h." Followed by the sampling process of the conditional distribution in "i." After that, the interventional distribution "j" sampled a target. This process was followed by the counterfactual distribution "k." The Euclidean distance between the vehicle and the pedestrian state is visualized in figure 4.11b. For each model, it varies a lot dependent on the initial position of the vehicle and the pedestrian. The amount of parameters influencing the simulation is not that small, so further research is needed. In model g, the MPC has problems bypassing pedestrian movements.



(a) Minimum distance between the pedestrian and vehicle for different pedestrian models



(b) Vehicle is driving to x-direction. For the end value after a simulation run the x-value of the vehicle is visualized for different pedestrian models.

Figure 4.11.: Simulation results for different pedestrian models

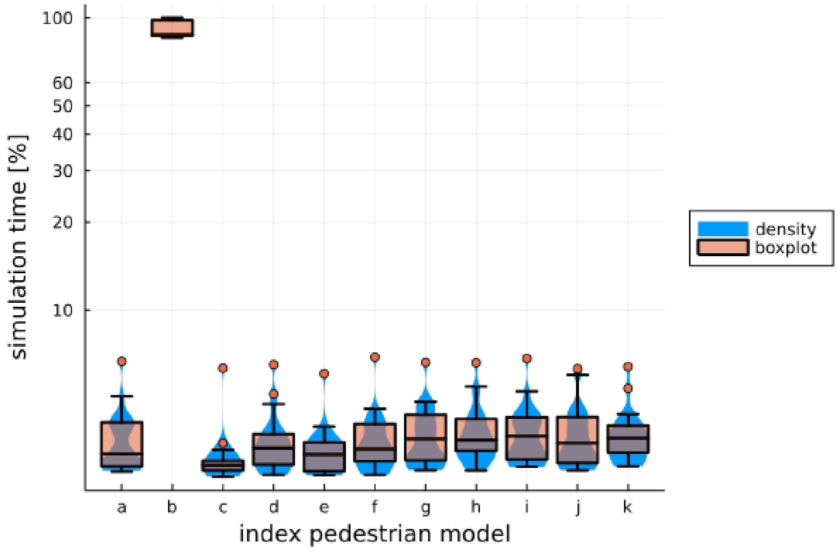


Figure 4.12.: The simulation time for different pedestrian models. The simulation time for the costly local approximation value iteration (LAVI) is referenced near to 100 percent of the simulation time. All other pedestrian models have less than ten percent of the simulation in comparison to the LAVI.

5. Conclusion

This chapter gives a discussion and contribution in section 5.1. Section 5.2 gives limitations on the proposed approach and section 5.3 gives an outlook.

5.1. Discussion and contribution

Typically encoding the principles of the energy flow in dynamic systems is sufficient to predict future states in model-based approaches or computing the reachable sets for all possible future state sets. On the other hand, in knowledge systems, evidence in the form of data builds the knowledge basis of the dynamic system in data-based systems. Sometimes the manifold assumption could help a predictor predict the state flow based on the geometrical structure of data. If there is not much data available, predicting the future state might be challenging. Human locomotion has three measurable sources influencing human behavior and dynamics: the human body, the cognitive system, and the environment. Natural laws constrain the human body as an energy system. The complex information system of the brain communicating with the nervous system affects the actions of the human body. The interaction between the human brain, body, and environment is complex. Many details and more basic research are necessary to understand the dynamics. It might be a question of future research if the human behavior is predictable (Laplace demon) if all sources of information are available. This thesis tries to capture the uncertainty by proposing adaptive belief sets. Predicting and guaranteeing future positions might be unknown for a trustable autonomous vehicle. The lack of knowledge of the intention leads to conflict in the decision-making process. Computing conservative reachable sets for pedestrians might lead to unnecessarily large areas by setting significant conservative parameters for the maximal velocity and acceleration. Therefore, this thesis proposes estimating the reachable sets for a specific situation to form an adaptive belief set. One way of handling this difficulty is to estimate the maximal velocity and acceleration or predict the future from geometrical sets. This thesis describes pedestrian models with causal inference between the brain and the body. A contribution of this thesis is to use machine learning to adapt conservative reachable sets to current situations. This thesis presents a simulation platform in Julia programming for cooperative motion planning and adaptive hull computation for new pedestrian models. These new pedestrian models combine data-based approaches with reachability analysis and causal inference. Meta-parameters for reachability analysis are often unknown in real-world applications and change over time. A pedestrian might have different maximal velocities and accelerations in different situations. This thesis proposes to combine machine learning for the adaption of reachable sets. Another contribution is that the simulator also has some models from causal inference to model intention changes for pedestrians. The target distribution consists of the initial-, conditional, interventional- and counterfactual distributions.

5.2. Limitations

A crucial part of safe autonomous vehicles in urban environments with pedestrians is understanding the nature of human beings and their decision-making. It is nearly impossible to describe all aspects of human behavior in an exact mathematical way and a simulation model. Some aspects lead to a lack of knowledge of an autonomous vehicle. In advance, the vehicle cannot know all relevant factors (future intention of a pedestrian) of decision making. This problem will lead to non-reducible uncertainty in the movement prediction of humans. The complexity of influencing factors that affect human behavior is vast. Therefore there exists a risk for the human movement prediction to fail and the risk of fatal decisions made by the motion planning. It is impossible to predict a person's movements with absolute certainty and high precisions if their intention is unknown. The complexity and diversity of human individuals alone make it difficult. It is impossible to interpret the human being as a closed system alone. Environment, social norms, culture, and individual decisions influence a person. It is all ready for human drivers sometimes a challenge to interpret and predict the decisions of humans. How can you get a machine to correctly interpret human concepts, social norms, or urban city interpretation? The development is dependent on computational capabilities and concepts, and each individual might also have individual preferences and irr-/rational decision-making (free will). This thesis cannot give a mathematical model which ensures safe predictions. Therefore human movements are too complex because cognitive mechanisms are not observable, and the intention is unknown. The proposed causal models represent only a tiny part of human behavior. The causal models simplify the relationships between the human brain and the body movements, where the application to real-world examples might deliver drawbacks.

5.3. Outlook

Safe, robust movement prediction algorithms for the autonomous system and driving efficiently in urban environments in the presence of pedestrians will be a hot topic for the vehicle research community. Absolute safety with the best efficiency might be the key challenge for building autonomous vehicles. Safety for pedestrians and vehicles by finding efficient motion plans in urban environments is crucial for accepting autonomous vehicles. There are further improvements needed in the perception part. Not all sensors recognize pedestrians and other vehicles in advance and only in some specific conditions. Perception is the basis for prediction, where the context and situational influences might affect a person's behavior. The causal relationships between the information perceived and encoding of the causal information chains are an essential step towards concept learning as humans do. Which factors could lead to an intentional change, and how is the vehicle influencing the behavior of the pedestrian. What is other information affecting the pedestrian? The causal models are fundamental. Structural causal learning is an active research field in finding causal relationships from data. This field only has a few practical solutions. It is necessary to get more robust set predictions by finding causal relationships between the environment, human cognition, and body movements. It is also possible to research predicting the meta-parameters robust and safe to make the adaptive belief sets capture complex situations. New optimization strategies or nonlinear reachability analysis models would be another exciting field. Ensuring optimality in nonlinear vehicle dynamics could be another feature for sophisticated scenarios. For databased-learning approaches, deep reinforcement learning is another active field where a learning model learns from data. For safety issues, it would be necessary to ensure robustness and learning result is, in some sense, dependent on the gathered dataset. It would be interesting that deep reinforcement learning learns causal structural relationships for prediction. These ideas should be validated to ensure pedestrian movement safety in some scenarios. One might also be interested in robust and set-based prediction algorithms for more complex scenarios where many different pedestrians are walking and interacting. A structural causal pattern recognition might be hard to get in real-world scenarios, but this would be a powerful tool for predicting all behaviors. Also, in scenarios with oncoming vehicles, the pedestrian's intention might be affected. Getting large datasets in the form of observational studies might not be enough. It might be necessary to find the effect in interventional settings. External influences might change the behavior of pedestrians, and this is part of causal structural learning approaches. A proposed Pedestrian in the Loop test environment might offer the possibility for experimental basic research to understand the whole cycle of human locomotion. For cooperative autonomous vehicles, safety and efficiency for vulnerable road users are contrary targets, and the development of decision-making algorithms is not trivial. Existing testing and robot systems reduce a pedestrian's intelligence due to the cognitive complexity of human intelligence. Therefore new "Pedestrian in the Loop" [Hartmann et al., 2017b, Hartmann et al., 2018d, Hartmann et al., 2018a] test environments place a test person in a virtual- or augmented test environment or by replacement by a drone. This new "Pedestrian in the Loop" offers the automotive industry new testing possibilities with natural test persons without any safety risks. It might be interesting to feed a computational intelligence with data to learn how a pedestrian would behave, like in inverse- and deep reinforcement learning.

Numbers and Arrays

a	A scalar (integer or real)
$\boldsymbol{a}$	A vector
$\boldsymbol{A}$	A matrix
$\mathsf{\mathbf{A}}$	A tensor
$\boldsymbol{I}_n$	Identity matrix with n rows and n columns
$\boldsymbol{I}$	Identity matrix with dimensionality implied by context
$\boldsymbol{e}^{(i)}$	Standard basis vector $[0, \dots, 0, 1, 0, \dots, 0]$ with a 1 at position i
$\text{diag}(\boldsymbol{a})$	A square, diagonal matrix with diagonal entries given by $\boldsymbol{a}$
a	A scalar random variable
$\mathbf{a}$	A vector-valued random variable
$\mathbf{A}$	A matrix-valued random variable

Sets and Graphs

$\mathcal{A}$	A set
$\mathbb{R}$	The set of real numbers
$\{0,1\}$	The set containing 0 and 1
$\{0,1,\dots,n\}$	The set of all integers between 0 and n
$[a,b]$	The real interval including a and b
$(a,b]$	The real interval excluding a but including b
$\mathcal{A}\backslash\mathcal{B}$	Set subtraction, i.e., the set containing the elements of $\mathcal{A}$ that are not in $\mathcal{B}$
$\mathbb{G}$	A graph
$Pa_{\mathbb{G}}(\mathrm{x}_i)$	The parents of x_i in $\mathbb{G}$
a_i	Element i of vector $\boldsymbol{a}$, with indexing starting at 1
a_{-i}	All elements of vector $\boldsymbol{a}$ except for element i
$A_{i,j}$	Element i,j of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{i,:}$	Row i of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	Column i of matrix $\boldsymbol{A}$
$\mathsf{A}_{i,j,k}$	Element (i,j,k) of a 3-D tensor $\boldsymbol{\mathsf{A}}$
$\boldsymbol{\mathsf{A}}_{:,:,i}$	2-D slice of a 3-D tensor
a_i	Element i of the random vector $\mathbf{a}$

Calculus

$\frac{dy}{dx}$	Derivative of y with respect to x
$\frac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\boldsymbol{x}} y$	Gradient of y with respect to $\boldsymbol{x}$
$\nabla_{\boldsymbol{X}} y$	Matrix derivatives of y with respect to $\boldsymbol{X}$
$\nabla_{\textsf{\textbf{X}}} y$	Tensor containing derivatives of y with respect to $\textsf{\textbf{X}}$
$\frac{\partial f}{\partial \boldsymbol{x}}$	Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$
$\nabla_{\boldsymbol{x}}^2 f(\boldsymbol{x})$ or $\boldsymbol{H}(f)(\boldsymbol{x})$	The Hessian matrix of f at input point $\boldsymbol{x}$
$\int f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral over the entire domain of $\boldsymbol{x}$
$\int_{\mathcal{S}} f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral with respect to $\boldsymbol{x}$ over the set $\mathcal{S}$

Probability and Information Theory

$\mathrm{P}(\mathsf{a})$	A probability distribution over a discrete variable
$p(\mathsf{a})$	A probability distribution over a continuous variable, or over a variable whose type has not been specified
$\mathsf{a} \sim P$	Random variable a has distribution P
$\mathbb{E}_{\mathsf{x}\sim P}[f(x)]$ or $\mathbb{E}[f(x)]$	Expectation of $f(x)$ with respect to $P(\mathsf{x})$
$\mathrm{Var}(f(x))$	Variance of $f(x)$ under $P(\mathsf{x})$
$\mathrm{Cov}(f(x), g(x))$	Covariance of $f(x)$ and $g(x)$ under $P(\mathsf{x})$
$H(\mathsf{x})$	Shannon entropy of the random variable x
$D_{\mathrm{KL}}(P\|Q)$	Kullback-Leibler divergence of P and Q
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Functions

$f : \mathcal{A} \to \mathcal{B}$	The function f with domain $\mathcal{A}$ and range $\mathcal{B}$
$f \circ g$	Composition of the functions f and g
$f(\boldsymbol{x}; \boldsymbol{\theta})$	A function of $\boldsymbol{x}$ parametrized by $\boldsymbol{\theta}$. (Sometimes we write $f(\boldsymbol{x})$ and omit the argument $\boldsymbol{\theta}$ to lighten notation)
$\log x$	Natural logarithm of x
$\lvert\lvert \boldsymbol{x} \rvert\rvert_p$	L^p norm of $\boldsymbol{x}$
$\lvert\lvert \boldsymbol{x} \rvert\rvert$	L^2 norm of $\boldsymbol{x}$
x^+	Positive part of x, i.e., $\max(0, x)$
$\mathbf{1}_{\text{condition}}$	is 1 if the condition is true, 0 otherwise

Differential geometry

$\mathcal{M}$	Manifold
$\mathcal{U}$	Patch
$\mathcal{T}$	Topology
$\mathcal{A}$	Atlas
ϕ	Coordinate chart
$\mathrm{CH}(\cdot)$	Convex hull
$\mathrm{CL}(\cdot)$	Closure

Computational Geometry

$\boldsymbol{e}_i$	Basis vector
$+$ or $\oplus$	Minkowski-Sum
conv	Convex Hull

Appendix

Appendix A.

Mathematical Background

This chapter should give a mathematical background for the whole thesis. It should give a brief recap of some common mathematical principles. Section A.1 introduces mathematical concepts from differential geometry. Section A.2 discusses concepts from computational geometry.

A.1. Differential Geometry

The papers of [Doolin and Martin, 2013,Robbin and Salamon, 2011] give a good introduction to differential geometry which is an essential part of understanding manifolds. The environment of a pedestrian can be described as a manifold (compare [Hartmann and Watzenig, 2019b]), and this is an essential part of the thesis. This is important for section 3.5. Figure A.1 shows a topological space. A set $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ of patches, which are connected via a topology $\mathcal{T}$. This topological space is a tuple $(\mathcal{X}, \mathcal{T})$.

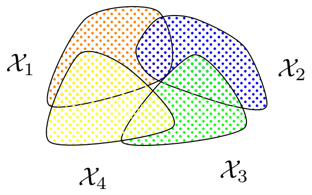


Figure A.1.: Topological space

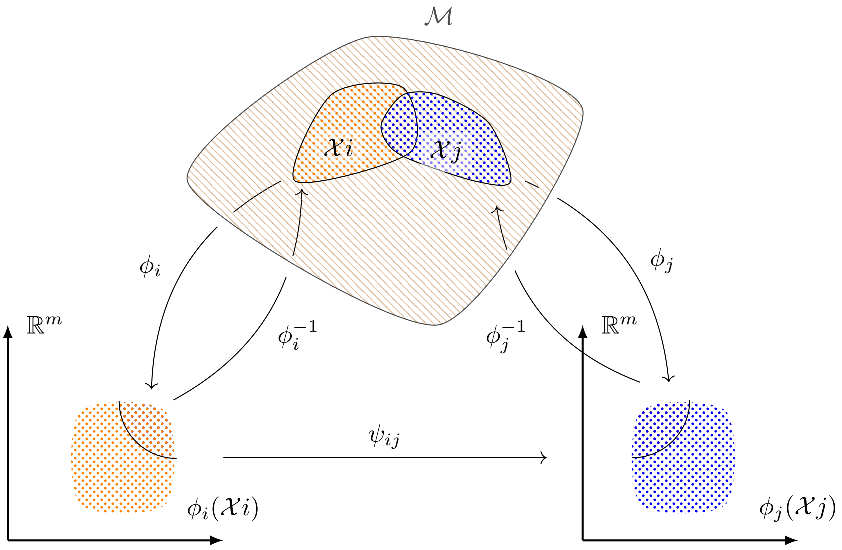


Figure A.2.: Manifold

In figure A.2 we see some mathematical illustration. Interconnected patches $\mathcal{X}_i$ and $\mathcal{X}_j$ could be mapped (chart) into two different euclidean spaces (which do not have orthogonal basis vectors). The formal definition of the chart is:

Definition A.1.1. Definition from [Robbin and Salamon, 2011]: A chart on a set $\mathcal{M}$ is a pair $(\phi, \mathcal{U})$ where $\mathcal{U}$ is a subset of M and $\phi : \mathcal{U} \rightarrow \phi(\mathcal{U})$ is a bijection from U to an open set $\phi(\mathcal{U})$ in $\mathbb{R}^m$. An atlas on $\mathcal{M}$ is a collection:

$$\mathcal{A} = \{(\phi_\alpha, \mathcal{U}_\alpha)\}_{\alpha \in A} \tag{A.1}$$

of charts such that the domains $\mathcal{U}_\alpha$ cover $\mathcal{M}$, i.e.

$$\mathcal{M} = \bigcup_{\alpha \in A} \mathcal{U}_\alpha \tag{A.2}$$

$\phi(p) = (x_1(p), \ldots, x_m(p))$ for $p \in \mathcal{U}$ form local coordinates defined on the subset $\mathcal{U}$. The definition of a topological manifold is:

Definition A.1.2. Definition from [Robbin and Salamon, 2011]: A topological manifold is a topological space $\mathcal{M}$. Each point $p \in \mathcal{M}$ has an open neighborhood $\mathcal{U}$, which is homeomorphic to an open subset of Euclidean space.

The intersection between two connected sets $\mathcal{X}_i$, $\mathcal{X}_j$ could be transformed in the euclidean space. The two charts are topologically compatible as follows.

Definition A.1.3. Definition from [Robbin and Salamon, 2011]: Let $\mathcal{M}$ be a set. Two charts $(\phi_1, \mathcal{U}_1)$ and $(\phi_2, \mathcal{U}_2)$ on $\mathcal{M}$ are said to be topologically compatible if $\phi_1(\mathcal{U}_1 \cap \mathcal{U}_2)$ and $\phi_2(\mathcal{U}_1 \cap \mathcal{U}_2)$ are open subsets of $\mathbb{R}^m$ and the transition map:

$$\phi_{21} = \phi_2 \circ \phi_1^{-1} : \phi_1(\mathcal{U}_1 \cap \mathcal{U}_2) \to \phi_2(\mathcal{U}_1 \cap \mathcal{U}_2) \tag{A.3}$$

is a homeomorphism. An atlas is said to be a topological atlas if any two charts are topological compatible.

The definition of a smooth manifold could be defined with a smooth atlas, where every chart is smoothly compatible with its members:

Definition A.1.4. Definition for Smooth Manifold from [Robbin and Salamon, 2011]: Let $\mathcal{M}$ be a set. A chart on $\mathcal{M}$ is a tuple $(\phi, \mathcal{U})$ where $\mathcal{U} \subset \mathcal{M}$ and ϕ is a bijection from $\mathcal{U}$ to an open subset $\phi(\mathcal{U}) \subset \mathbb{R}^m$ of some euclidean space. Two charts $(\phi_1, \mathcal{U}_1)$ and $(\phi_2, \mathcal{U}_2)$ are said to be smoothly compatible if $\phi_1(\mathcal{U}_1 \cap \mathcal{U}_2)$ and $\phi_2(\mathcal{U}_1 \cap \mathcal{U}_2)$ are both open $\mathbb{R}^m$ and the transition map

$$\phi_{21} = \phi_2 \circ \phi_1^{-1} : \phi_1(\mathcal{U}_1 \cap \mathcal{U}_2) \to \phi_2(\mathcal{U}_1 \cap \mathcal{U}_2) \tag{A.4}$$

is a diffeomorphism. A smooth atlas on $\mathcal{M}$ is a collection $\mathcal{A}$ of charts on $\mathcal{M}$ any two of which are smoothly comaptible and such that the set $\mathcal{U}$, as $(\phi, \mathcal{U})$ ranges over the elements of $\mathcal{A}$, cover $\mathcal{M}$ (i.e. for every $p \in \mathcal{M}$ there is a chart $(\phi, \mathcal{U}) \in \mathcal{A}$ with $p \in \mathcal{U}$). A maximal smooth atlas is an atlas which contains every chart which is smoothly compatible with each of its members. A smooth manifold is a pair consisting of a set $\mathcal{M}$ and a maximal smooth atlas $\mathcal{A}$ on $\mathcal{M}$.

The tangent vector for a manifold is:

Definition A.1.5. Definition from [Robbin and Salamon, 2011]: Let $\mathcal{M} \subset \mathbb{R}^k$ be a smooth m-dimensional manifold and fix a point $p \in \mathcal{M}$. A vector $v \in \mathbb{R}^k$

is called a tangent vector of $\mathcal{M}$ at p if there exists a smooth curve $\gamma : \mathbb{R} \to \mathcal{M}$ such that

$$\gamma(0) = p, \dot{\gamma}(0) = v \tag{A.5}$$

The set

$$\mathrm{T}_p\mathcal{M} := \{\dot{\gamma}(0) | \gamma : \mathbb{R} \to \mathcal{M} \text{ is smooth }, \gamma(0) = p\} \tag{A.6}$$

of tangent vectors of $\mathcal{M}$ at p is called the tangent space of $\mathcal{M}$ at p.

The tangent bundle of a manifold $\mathcal{M}$ is:

Definition A.1.6. Definition from [Robbin and Salamon, 2011]: Let $\mathcal{M} \subset \mathbb{R}^k$ be a smooth m-manifold. A (smooth) vector field on $\mathcal{M}$ is a smooth map $X : \mathcal{M} \to \mathbb{R}^k$ such that

$$X(p) \in \mathrm{T}_p\mathcal{M} \tag{A.7}$$

for every $p \in \mathcal{M}$. The set of smooth vector fields on $\mathcal{M}$ will be denoted by

$$\text{Vect}(\mathcal{M}) := \{X : \mathcal{M} \to \mathbb{R}^k | X \text{ is smooth}, X(\boldsymbol{p}) \in \mathrm{T}_{\boldsymbol{p}}\mathcal{M} \; \forall \; \boldsymbol{p} \; \in \mathcal{M}\} \tag{A.8}$$

The tangent bundle of $\mathcal{M}$ is as following:

Definition A.1.7. Definition from [Robbin and Salamon, 2011]: Let $\mathcal{M} \subset \mathbb{R}^k$ be a smooth m-manifold. The set

$$\mathrm{T}\mathcal{M} := \{(\boldsymbol{p}, \boldsymbol{v}) | \boldsymbol{p} \in \mathcal{M}, \boldsymbol{v} \in \mathrm{T}_p\mathcal{M}\} \tag{A.9}$$

is called the tangent bundle of $\mathcal{M}$.

Figure A.3 shows the earth as a sphere and a local area (with a red grid) which could be described as a euclidean space. In [Hartmann and Watzenig, 2019b] walkable areas in urban environments were described as topological spaces.

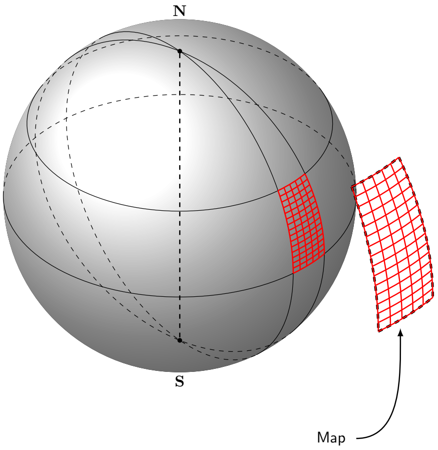


Figure A.3.: Earth as a manifold (Latex template adapted from [Miani, 2009])

A.2. Computational geometry

This section should describe some concepts from computational geometry. We first start with some properties for sets. The following definition describes an affine set:

Definition A.2.1. Definition adapted from [Boyd et al., 2004]: A set $\mathcal{C} \subseteq \mathbb{R}^n$ is affine if the line through any two distinct points in $\mathcal{C}$ lies in $\mathcal{C}$.

Another important definition is the convex set:

Definition A.2.2. Definition adapted from [Boyd et al., 2004]: A set $\mathcal{C}$ is convex if the line segment between any two points in $\mathcal{C}$ lies in $\mathcal{C}$, i.e., if for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$ and any θ with $0 \leq \theta \leq 1$.

Figure A.4 shows examples for different convex sets, represented by different mathematical objects. The convex hull is defined as:

$$\mathrm{CH}(\mathcal{V}_1, \mathcal{V}_2) = \{\alpha_1 \cdot s_1 + \alpha_2 \cdot s_2 | s_1 \in \mathcal{V}_1, s_2 \in \mathcal{V}_2, \alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_1 + \alpha_2 = 1\} \tag{A.10}$$

The Minkowski-Sum between two sets $\mathcal{V}, \mathcal{U}$ (compare the illustration in figure A.5):

$$\mathcal{V} \oplus \mathcal{U} = \{\boldsymbol{v} + \boldsymbol{u} | \boldsymbol{v} \in \mathcal{V}, \boldsymbol{u} \in \mathcal{U}\} \tag{A.11}$$

The definition of a zonotope is:

Definition A.2.3. A zonotope $\mathcal{Z}$ is a set such that (modified from: [Girard et al., 2006]):

$$\mathcal{Z} = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} = \boldsymbol{c} + \sum_{i=1}^{i=p} x_i g_i, -1 \leq x_i \leq 1 \right\} \tag{A.12}$$

where $\boldsymbol{c}, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_p$ are vectors of $\mathbb{R}^n$. Notation: $\mathcal{Z} = (\boldsymbol{c}, < \boldsymbol{g}_1, \ldots, \boldsymbol{g}_p >)$

The volume or area of two or three vectors $\boldsymbol{x}_i$ $i = 1, \ldots, 3$ can be computed by the determinant. The minmin-function is [Grégoire and Bouillot, 1998] could give us the minimal distance between two sets:

$$\mathrm{D}(\mathcal{X}, \mathcal{Y}) = \min_{\boldsymbol{x} \in \mathcal{X}}(\min_{\boldsymbol{y} \in \mathcal{Y}}(\mathrm{D}(\boldsymbol{x}, \boldsymbol{y}))) \tag{A.13}$$

with the euclidean distance (which could replaced by another norm).

$$\mathrm{D}(\boldsymbol{x}, \boldsymbol{y}) = |\boldsymbol{x} - \boldsymbol{y}|_2 \tag{A.14}$$

We want to estimate the computational complexity for different variants of sets. Two compare two sets $\mathcal{A}$ (for a reachable set $\mathcal{R}$) and $\mathcal{B}$ (catcher set $\hat{\mathcal{Y}}$). The intersection is defined as:

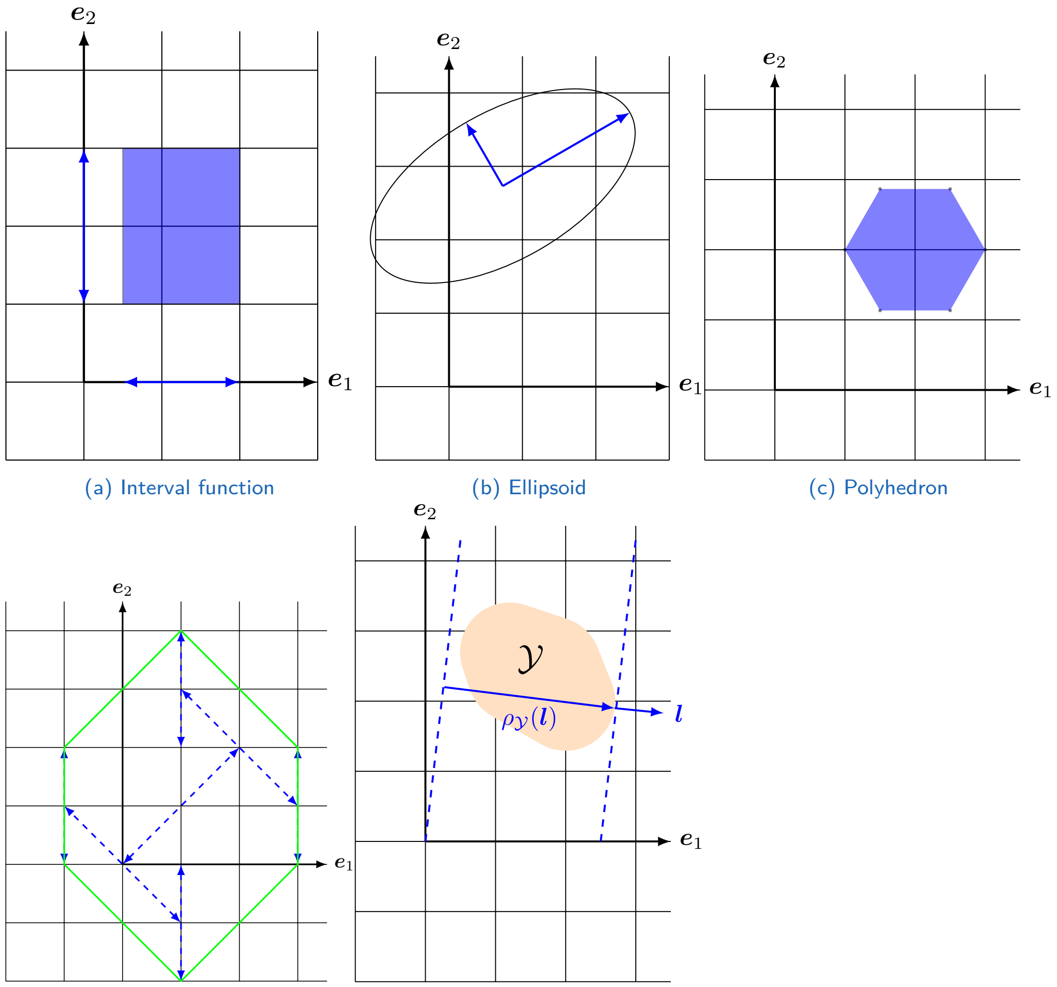
$$\mathcal{C} = \mathcal{A} \cap \mathcal{B} \tag{A.15}$$

The union is defined as:

$$\mathcal{C} = \mathcal{A} \cup \mathcal{B} \tag{A.16}$$

The Jaccard-distance is defined as:

$$J(\mathcal{A}, \mathcal{B}) = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|} \tag{A.17}$$



(a) Interval function

(b) Ellipsoid

(c) Polyhedron

(d) Zonotope

(e) Support function

Figure A.4.: Different set representations

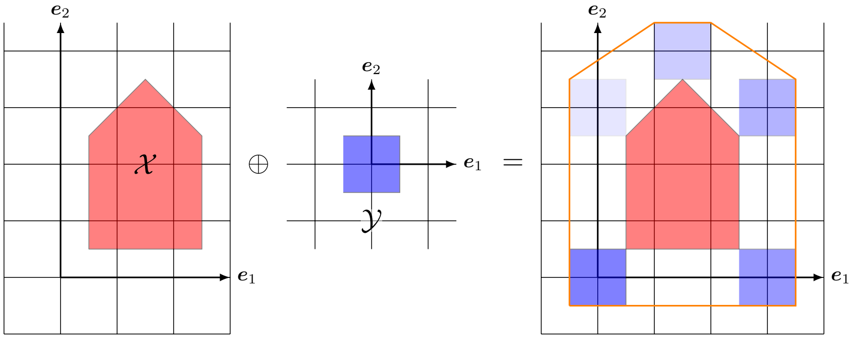


Figure A.5.: Minkowski-Sum

It might be a possible risk to capture not parts of the reachable sets by a catcher

$$\mathcal{C}(t) = \mathcal{R}(t) \setminus (\mathcal{R}(t) \cap \hat{\mathcal{Y}}(t)) \tag{A.18}$$

Another distance for comparing two sets is the Hausdorff distance. Table A.6 shows different representations of different convex set representations.

(Convex-)set representations	
Ellipsoid	1.1 Center $\boldsymbol{c} \in \mathbb{R}^n$ 1.2 Directional scaling $\{\boldsymbol{s}_q\}_{q=1}^n \in \mathbb{R}^n$ 1.3 Quadratic form $Q(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$, $\boldsymbol{A} = \begin{bmatrix} a & b \\ b & d \end{bmatrix} = \boldsymbol{A}^T$(symmetric)
Polytope (three-dimensional: Polyhedron, two-dimensional: Polygon)	2.1a Vertices (V-Representation) $\mathcal{V} = \text{conv}(\{\boldsymbol{v}_q\}_{q=1}^n)$[a] 2.1b Half-spaces (H-Representation) $\cap\{\{\boldsymbol{x}_q \| \boldsymbol{A}\boldsymbol{x}_q \leq \boldsymbol{b}, \boldsymbol{x}_q \in \mathcal{X}\}\}_{q=1}^p$ [a] Instead of computing the convex hull, one could give the topology of the vertices, which might lead to non-convex polyhedrons.
Zonotope	3.1 Center $\boldsymbol{c} \in \mathbb{R}^n$ 3.2 Generator-Set $\mathcal{G} = \{\boldsymbol{g}_q\}_{q=1}^n \in \mathbb{R}^n$
Over-approximation with support-functions	3.1 Direction vectors $\mathcal{L} = \{\boldsymbol{l}_q\}_{q=1}^p$ 3.2 Support function $\rho_{\mathcal{Y}}(\boldsymbol{l}) = \max\{\boldsymbol{l}^T \boldsymbol{x} \| \boldsymbol{x} \in \mathcal{Y}\}$ 3.3 $\mathcal{H}_{\boldsymbol{l}} = \{\boldsymbol{x} \in \mathbb{R}^n \| \boldsymbol{l}^T \boldsymbol{x} \leq \rho_{\mathcal{X}}(\boldsymbol{l})\}$ 3.4 $\lceil \mathcal{Y} \rceil = \bigcap_{\boldsymbol{l} \in \mathcal{L}} \{\boldsymbol{x} \in \mathbb{R}^n \| \boldsymbol{l}^T \boldsymbol{x} \leq \rho_{\mathcal{Y}}(\boldsymbol{l})\}$

Figure A.6.: Table for different (convex-)set representations

Appendix B.

Cognitive systems

This chapter describes some aspects of human locomotion that describe the current state of the art. Partially, some aspects cannot be verified within the scope of the thesis. Measurements, for example, with consideration of human cognitions, were not possible within the scope of the thesis but would be an interesting component to investigate more detailed results. For the interested reader, a part of the current literature is offered in this section. Section B.1 presents some challenges for modelling human behavior. Section B.2 gives an overview of human locomotion's complexity with aspects from biomechanics and neuroscience. Section B.3 summarizes the measurement and simulation technologies and section B.4 introduces the spatial description of the environment.

B.1. Challenges for modelling human behavior

The script of [Knill, 2005] presents different types of mathematical descriptions of dynamic systems (with advanced definitions of semi-groups and different mathematical properties). [Earman et al., 1986] presents a philosophical perspective to determinism. An introduction to uncertainty quantification [Sullivan, 2015, Zio and Pedroni, 2013]. System theory describes a system with a boundary where external influences change its state over time. In physical dynamic systems, the state evolves based on principles and depending on the energy flow (contrary to pure mathematical systems, which describe physical systems). Many technical systems can be described with ordinary differential equations, fluids by partial differential equations, and weather phenomena by chaotic systems, depending on the system's nature. In physics, many phenomena could not be described in mathematical models adequately. The predictions could not have a good performance. For example, the prediction of weather phenomena had a low performance as long the models did not consider the chaos theory and the computational resources. Scientists in the 19 century thought about a world formula, where all states in the cosmos are predictable, and there would be determinism, which could lead to prediction models (Laplace demon). Due to theories from relativity theory, chaos theory, and uncertainty relation in quantum theory, it is known that there exists some non-reducible uncertainty. This problem might also be the case in human movements. One big problem in this thesis is that the time-evolution of a human brain's cognitive intelligence state is complex. It is a philosophical question of whether the brain's state is predictable and how this would affect a person's free will. There is also a demand for further improvements in the measurements of brain functionality. The measurement devices are often costly, not adequate for testing in automotive tests and the time-space resolution is improvable. This complexity of human locomotion has consequences for the prediction models. We assume a non-reducible uncertainty of human behavior prediction exists, and there are determinism and uncertainty in human decision-making. Algorithms can learn patterns in a dataset, but the problem is that the algorithms cannot decode the nature of causal relationships in human decision-making. Some essential variables might not be available. Other factors like the environment also play a crucial part and affect pedestrians. There exist different representations of the spatial environment of the walkable areas of the pedestrian. If historical data is available, machine learning can guess the most probable position in spatial space. However, how trustable could the prediction model make decisions when an unknown human appears. However, when there is uncertainty due to randomness (aleatoric uncertainty, e.g., due to the lack of knowledge), the task is more guessing than a prediction. For better predictions, it is necessary to decode the state flow's underlying principles. The nature of the biological cybernetic system in human locomotion is not sufficiently developed. There exists a need for basic research (for advanced models in human locomotion and the causal relationships between perception, decision-making, and movement).

B.2. Complexity of human locomotion

Figure B.1 shows the taxonomy of information necessary for the movement prediction of pedestrians. The interior perspective captures all information inside the pedestrian, the human body, and psychological aspects caused by the human brain. The mind with its cognitive activities, including different concepts of consciousness, imagination, perception, thinking, judgment, language, and memory. Historical aspects might influence the information structure of the human brain and personality. The human brain can learn from sensorial inputs. The complex structure of the human brain can change in time and its connectivity of neurons. The exterior perspective concerns the environment, the current situation, and social constraints affecting human behavior. There is also other information caused by the interaction process between the environment and the consciousness of the human mind.

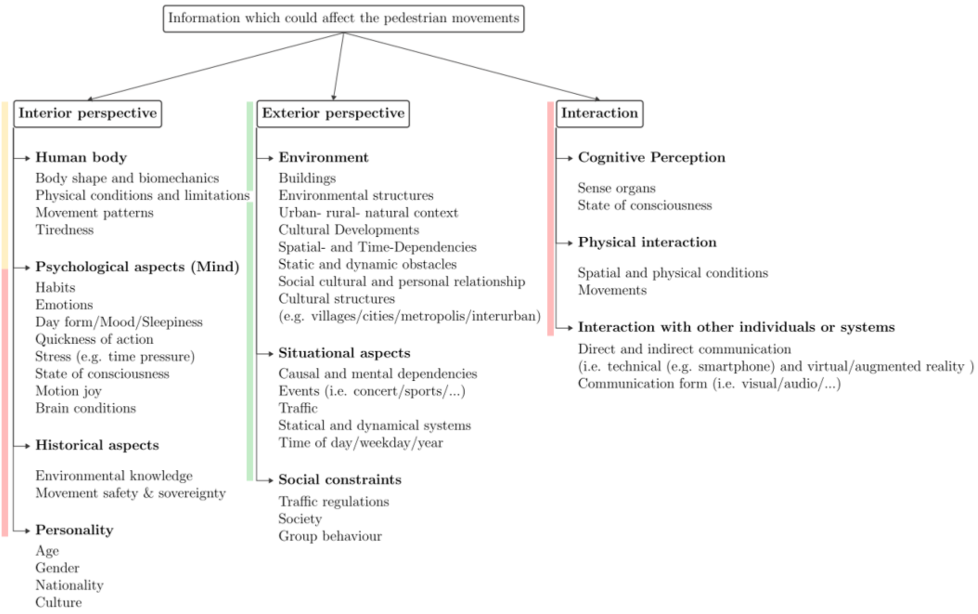

Information which could affect the pedestrian movements
Interior perspective
Human body
Body shape and biomechanics
Physical conditions and limitations
Movement patterns
Tiredness
Psychological aspects (Mind)
Habits
Emotions
Day form/Mood/Sleepiness
Quickness of action
Stress (e.g. time pressure)
State of consciousness
Motion joy
Brain conditions
Historical aspects
Environmental knowledge
Movement safety & sovereignty
Personality
Age
Gender
Nationality
Culture
Exterior perspective
Environment
Buildings
Environmental structures
Urban- rural- natural context
Cultural Developments
Spatial- and Time-Dependencies
Static and dynamic obstacles
Social cultural and personal relationship
Cultural structures
(e.g. villages/cities/metropolis/interurban)
Situational aspects
Causal and mental dependencies
Events (i.e. concert/sports/...)
Traffic
Statical and dynamical systems
Time of day/weekday/year
Social constraints
Traffic regulations
Society
Group behaviour
Interaction
Cognitive Perception
Sense organs
State of consciousness
Physical interaction
Spatial and physical conditions
Movements
Interaction with other individuals or systems
Direct and indirect communication
(i.e. technical (e.g. smartphone) and virtual/augmented reality)
Communication form (i.e. visual/audio/...)

Figure B.1.: The taxonomy shows the classification of information that might affect the quality of movement prediction.

This section highlights the state of the art of the apparatus of human locomotion. It consists of the cognitive cycle of observations to the reasoning (section B.2.1 and B.2.2) to the performing of human actions with the human body B.2.3, which is relevant for the interaction between vehicle and pedestrian. This section gives insights and a basic tutorial into topics (neuroscience, biomechanics, and others) relevant for developing simulation models from a technical engineer's perspective.

B.2.1. Human perception

A person observes the environment with his sense organs, mainly his eyes, and catches visual signals. Eye muscles can stretch or relax the eye lenses to focus on an object. The visual information reaches the eyes and the retina. Some photoreceptors on the retina convert the visual signal into electrical impulses [Hubel and Wiesel, 1979]. Optic nerves transfer this information to the primal visual cortex. Both brain halves share some visual information due to the optic chiasm. Each half of the brain gets visual signals from the visual fields. There exist two pathways after the visual cortex processes the information. The dorsal path answers spatial questions like "where is something" (object location, spatial perception, working memory is active). The ventral path answers what the brain sees (object identification, fine visual details are detected, long-term memory is activated). The visual cortex computes some scene features (objects, depth-information) [Yamins et al., 2014a]. After transferring the electrical impulses to the visual cortex, the person is reasoning about the incoming information, mainly in the association cortex [Yamins et al., 2014b]. This fact offers the possibility to detect objects, classify them, and associate them with historical experiences. The process of identifying objects and recognizing them is part of a network of hierarchically related brain areas [Botvinick, 2007,Yamins et al., 2014a]. The isolated left hemisphere (left halve of the brain) is concerned with abstract thinking, symbolic relationships, and logical analysis of details [Hubel and Wiesel, 1979]. It is the leading hemisphere for the control of the nervous system. The left hemisphere cannot exist without the right hemisphere. The right hemisphere is responsible for concrete thinking, spatial consciousness, compression of complex relationships, and auditory tasks. [Hubel and Wiesel, 1979] gave a detailed interpretation of the visual system and the signal processing of images in the visual cortex. Each layer in the visual cortex works on different abstraction levels. Encoding and decoding [Naselaris et al., 2011b,Naselaris et al., 2011a] is a current approach to reading the patterns of brain activity. The method is done by signaling and measurement devices of the human brain (EEG, fMRI, and others) and decoding visual information or inferring brain activity from visual input (encoding). It is a complex problem to make realistic simulations of mental functions and cognitive tasks in an exact and complete mathematical model. [Chen, 2017] gives an introduction to these measurement technologies. The motor cortex is mainly responsible for transferring signals from the brain to the body (first image) combined with the premotor cortex and supplementary motor cortex: the somatosensory cortex, Brodman Nr. 3, 1 and 2 are mainly responsible for the measurements of the body movements—the visual cortex for the sensing of visual information and the auditory cortex for audio signals. The Brodman area 7 processes information relevant to the relative position of an object to the human body. The Brodman atlas is an older classification of the human brain, and there exist newer models which give better spatial representations of the human brain [Naselaris et al., 2011a], [Huth et al., 2016a].

B.2.2. Reasoning for locomotion

The brain processes incoming information by a vast number of neurons [Salles, 2019] in a very complex topological structure. Each region is responsible for different mental tasks [Huth et al., 2016a]. A current challenge is the low time-space resolution, the test's complexity, and generalizability of experiments. Different experiments exist to understand the functionality of the human brain [Essen and Glasser, 2016,Markram, 2006,Salles, 2019,Bargmann, 2014]. The Human-Connectome project offers new brain visualizations [Essen and Glasser, 2016] that visualize the information highways and their topological structure. In the Blue-Brain project [Markram, 2006], researchers built a detailed simulation of neuronal activity levels. In the Human-Brain project, [Salles, 2019] extended their research. The Brain-Initiative [Bargmann, 2014] tries to build realistic simulations of the human brain with supercomputers. Researchers use machine learning approaches in [Yamins et al., 2014a] to encode and decode brain measurements to classify brain activity. [Huth et al., 2016b] showed that it is possible to classify the brain regions by the semantic meaning and introduce a semantic atlas for a specific task (hearing a radio show) for the subjects. Harvard University developed a program called "Freesurfer" [Fischl, 2012] to analyze brain images. Many research networks are trying to reproduce the human brain's functionality in a simulation framework [Markram, 2006,Essen and Glasser, 2016,Salles, 2019]. There are several difficulties because there is a huge demand for computing power [Salles, 2019] and measurement technologies [Chen, 2017].

B.2.3. Human movements

The Motor- and Sensory-Cortex is a kind of interface between the brain and the rest of the body, which actuates and receives signals to perform movements [Botvinick, 2007]. The peripheral nervous system's communication system transfers the electric impulses to the muscles to perform the movements. A human skeleton has a topological structure [Drake et al., 2009]. The joints (vertexes/nodes) connect bones (edges) in a physical multibody system. Different types of joints exist, but the synovial joints are freely movable. The body's mass density (parameters for a physical model) — the brain's communication procedures and the communication in the peripheral nervous system can influence action quickness [Delp et al., 2007]. Synovial joints have different movement leeway freedoms. This freedom ensures complex skeleton movements. [Li and Yang, 2012] discusses different dynamic models for human locomotion. [Nigg and Kuntze, 2012] gives an introduction to several topics in biomechanics.

B.2.4. Consequences

If someone modeled the human locomotion of a person, it would be necessary to model the state of the brain $\boldsymbol{c}(t)$ and the human body state $\boldsymbol{s}(t)$. If the brain's activity patterns represent the cognitive state $\boldsymbol{c}(t)$ and the positions of the human skeleton are $\boldsymbol{s}(t)$. If only the skeleton of a person is measurable one could model a nonlinear function $\dot{\boldsymbol{s}} = \mathbf{f}(\boldsymbol{s}(t), \boldsymbol{a}^s(t))$ with a control input $\boldsymbol{a}^s(t)$. The dimensions of the brain state and body skeleton are immense. Only the three-dimensional pictures of the human brain bring an immense of information. Also, the skeleton information would bring several dimensions. For machine learning approaches, this would mean using a prediction model which could handle the curse of dimensionality [Bishop, 2006, Murphy, 2012]. The complexity of the brain and some lack of measurement- and computer technology is a huge problem for this thesis. It might be necessary to encode the brain's activity patterns leads to the problem that no exact computer model for human locomotion is available. In automotive engineering, scientists model human behavior as a black box or inspired by physical laws (like social force models). The whole complexity of human locomotion often leads to oversimplification. Another consequence is the non-reducible uncertainty of predicting the actual movements of a person.

B.3. Measurement- and simulation technologies of body movements

If there is the approach to find an exact human locomotion model (white- or grey box model) and find a formulation $\dot{\boldsymbol{s}} = \mathbf{f}(\boldsymbol{s}(t), \boldsymbol{a}^s(t))$ for the state-flow of the body skeleton. Measurements for the influences from the brain and their control inputs would be essential. This section presents some measurement technologies for the body- $\boldsymbol{s}(t)$ and brain state $\boldsymbol{c}(t)$. There are different technical challenges for tracking body movements, simulation, and learning. Tracking the body is an essential discipline in the measurement community, and there are different body-tracking sensors available. The authors in [Pellegrini et al., 2009] present measurement and simulation technologies for human movements. Also, the computer vision community is very active in learning the movements from camera images. [Xia et al., 2012] present laboratory measurements of the movement of the joints of a test person. The test person's joint positions were tracked with a Microsoft Kinect camera. The test person performed actions in a laboratory (e.g., stand up, wave hands, and more). The reduction to a single skeleton trajectory leads to loss of information, where the person's activities are not known anymore (walking, waving hands). [Tompson et al., 2014], [Jain et al., 2013] proposed one of the first results of human pose estimation with deep learning — a training procedure with convolutional networks and a graphical model for human pose estimation. [Eigen and Fergus, 2015] also predicts depth, surface normals, and semantic labels with a typical multi-scale convolutional architecture. [Wei et al., 2016] use a multi-stage CNN that can localize body parts. [Pellegrini et al., 2009] use bird view measurements of pedestrians. [Janai et al., 2017] present segmentation learning for specific driving situations with pedestrian detection. [Charles et al., 2017] focuses on human body parsing in images and videos. [Shotton et al., 2013] proposes an approach for human body parsing from a single depth image and performing 3D human pose estimation. [Urtasun et al., 2006] proposes an approach for human skeleton tracking. [Toshev and Szegedy, 2014] present an approach for a cascade of Convolution Neural Networks for human pose estimation: at each stage of the cascade, the predicted pose is updated to get closer to the ground truth pose. [Wei et al., 2016] also uses CNNs that produce a heat map of a localized body part. [Bogo et al., 2017] presents dynamic movement datasets of persons with time-varying meshes. [Peng et al., 2018] presents an approach with deep reinforcement learning to imitate human movements and simulate an avatar. The algorithm can perform new movements with the avatar based on a few experiences. The deep learning algorithm of [Jaśkowski et al., 2018] learns to perform the tasks (running and walking) by repetition. It imitates control of the motor and sensory cortex. Often the body dynamics of a pedestrian are simplified and approximated. Numerical software [Delp et al., 2007] could help compute the human body forces. [Jaśkowski et al., 2018] used algorithms to show that Deep Learning Algorithms can imitate some parts of the human brain to perform different tasks (e.g., running). The winning solution [Jaśkowski et al., 2018] of the Run Challenge [Kidziński et al., 2018] is used for the illustrations' simulations. Stanford University developed the simulation tool "OpenSim," which computes a skeleton's forces with a multi-body simulation [Delp et al., 2007]. The skeleton consists of different bones and joints. There is a high degree of freedom to control the whole multi-body system, making the task difficult. There are only a few experiments where scientists measured body movements and brain signals. [He et al., 2018] developed an experiment on a treadmill and EEG measurements and of body movements. [Brantley et al., 2018] proposes wearable EEG measurements for locomotion research in an indoor hall for movement research. [Perez et al., 2019] proposes a test environment with a test person and driver on a computer with virtual reality. For the body state $\boldsymbol{s}_k$, we model N positions $\boldsymbol{p}_{i,k} \in \mathbb{R}^3$ (e.g., position of left hand, head or right knee) of the human skeleton $\boldsymbol{s}_k := [\boldsymbol{p}_{1,k}, \ldots, \boldsymbol{p}_{N,k}]^T \in \mathbb{R}^{3 \cdot N}$. This thesis use only the center of gravity $\overline{\boldsymbol{p}}_k$ also because of the high-dimensionality and the projection to the ground $\boldsymbol{x}_k = \boldsymbol{P} \cdot \overline{\boldsymbol{p}}_k \in \mathbb{R}^2$. Figure B.2 shows the Brodman Atlas. It is a classification of regions of the brain (a manifold) by different action patterns. The Motor Cortex is responsible for the movements of the person, while the sensory cortex is for sensing. The visual cortex for the vision and many other regions. Now there exist further advanced approaches like the Pragmatic Atlas in figure B.3a for semantic mapping of brain regions [Huth et al., 2016a]. There exist approaches to classify the brain regions in voxel-level B.3a mainly with machine- and deep learning. The relation between the cognitive states of the brain $\boldsymbol{c}(t)$ and the body movements like in B.3b is an open basic research problem. For that, a Pedestrian in the Loop is proposed in the next section.

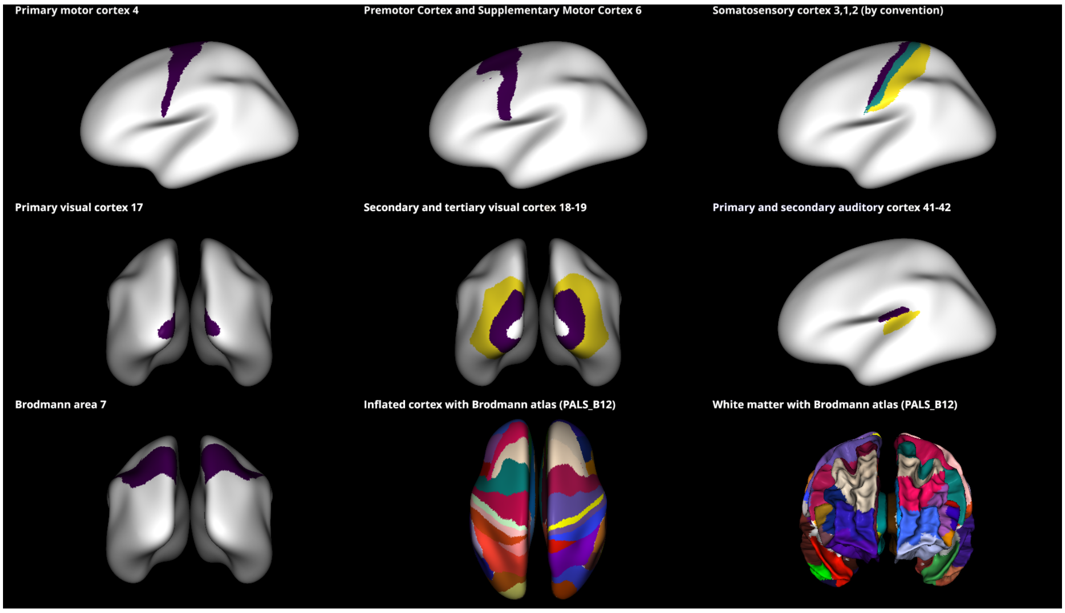
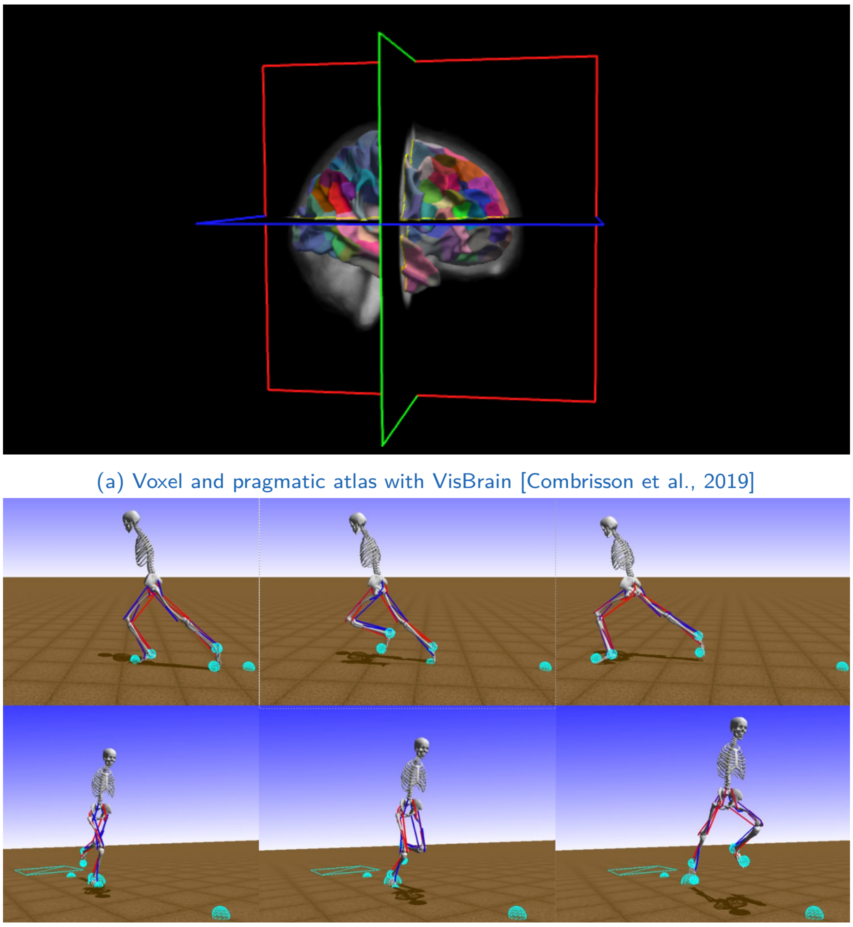

Primary motor cortex 4
Premotor Cortex and Supplementary Motor Cortex 6
Somatosensory cortex 3,1,2 (by convention)
Primary visual cortex 17
Secondary and tertiary visual cortex 18-19
Primary and secondary auditory cortex 41-42
Brodmann area 7
Inflated cortex with Brodmann atlas (PALS_B12)
White matter with Brodmann atlas (PALS_B12)

Figure B.2.: Brodman atlas and selection of different activation regions

B.4. Simulation of urban environments

The environment influences and constrains the movements of the pedestrian body state $\boldsymbol{s}(t)$ and affects the cognitive state $\boldsymbol{c}(t)$. There exist different mathematical descriptions of the environment: euclidean space, Spatial graph, manifold, regular grid, lanelets, and nonregular grid. There also exist different measurement technologies for the environment. The paper in [Hackel et al., 2017] presents a semantic point cloud and the work of [Cignoni and Callieri, 2008] could be used for Mesh data. Drones can collect data from the ground as a 3D point cloud, compared to the bottom right picture. [Armeni et al., 2016, Armeni et al., 2017] presented a point cloud dataset of buildings with corresponding 3D Mesh, 2D and 3D semantics, Depth information, and surface normals. [Savva et al., 2019] offers a rendered virtual environment "Habitat project" with a photorealistic virtual environment as meshes. [Armeni et al., 2019] give also an interesting hierarchical structure of environments [Armeni et al., 2019]. Interactive Gibson environment [Xia et al., 2018] is a new environment for developing reinforcement learning algorithms for agents in different virtual buildings. A humanoid virtual



(a) Voxel and pragmatic atlas with VisBrain [Combrisson et al., 2019]

(b) Open Sim Simulation [Kidziński et al., 2018]

Figure B.3.: Brain and Body simulation technologies

agent can learn to move in a virtual environment. Simulated physical forces are affecting the agent. [Straub et al., 2019] proposes the replica dataset showing a virtual environment and the corresponding segmentation [Straub et al., 2019]. The geometric structure of the environment constrains the movements and the physical touch. The environment's geometric structure is not the only kind of information that is important for human locomotion. Also, the semantic information [Kitani et al., 2012,Ziebart, 2010,Vasquez, 2010b] is an important factor in answering "why" a person intends to go to a certain place (goes to the supermarket to buy something).

Appendix C.

Additional information

Section C.1 gives more information on the programmed simulator and further simulation examples.

C.1. Simulator

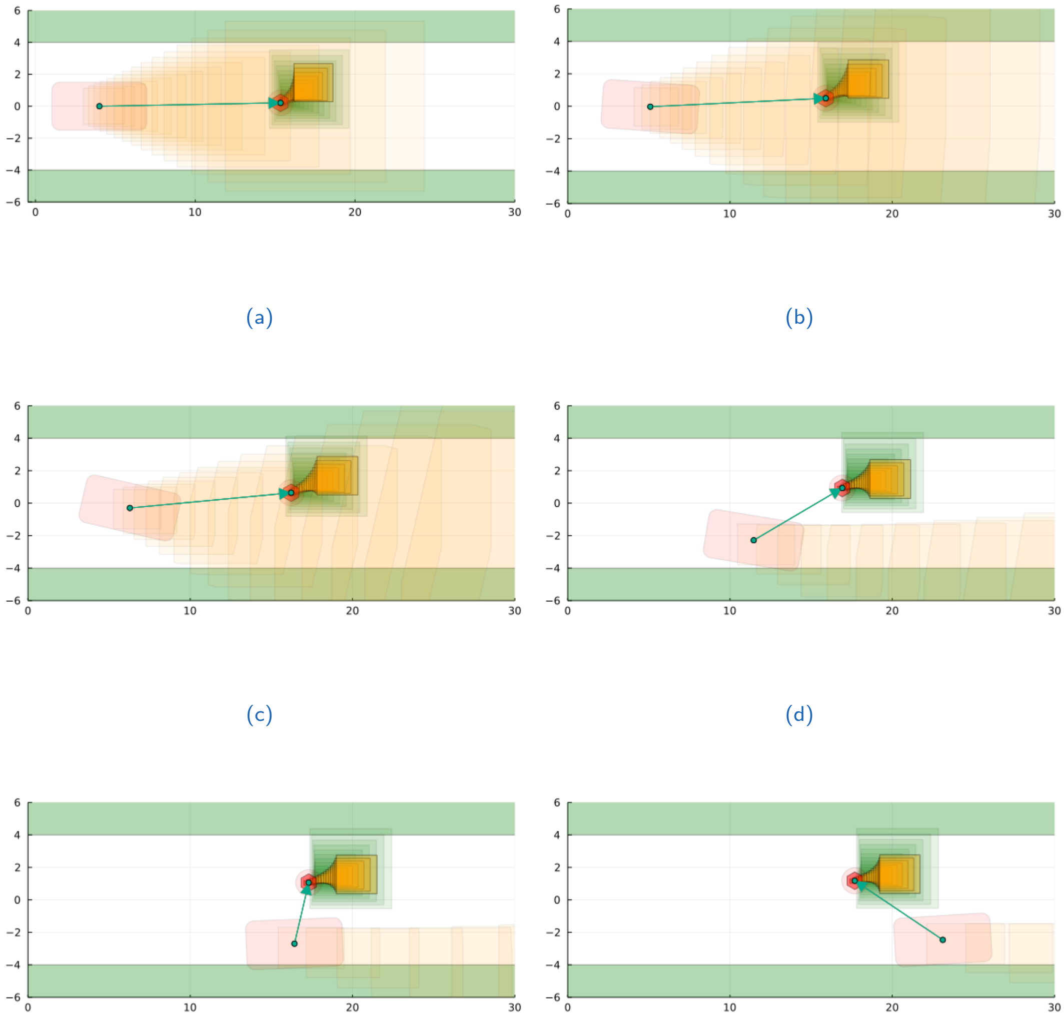
The simulator was programmed with Julia Programming [Bezanson et al., 2017]. It should have similarities to ROS [et al., 2018] by sending messages and having a signal slot architecture for event-based programming. Table C.1 shows a selection of installed packages for the use of the simulator with some parameters like in table C.2. Figure C.1, C.2, C.3 show further visual representations of the simulations and figure C.4 shows the resulting states.

```
Installed packages: |
[6e4b80f9] BenchmarkTools v0.7.0
[a81c6b42] Compose v0.9.3
[a93c6f00] DataFrames v1.3.2
[b4f34e82] Distances v0.9.2
[31c24e10] Distributions v0.23.11
[c87230d0] FFMPEG v0.4.1
[5789e2e9] FileIO v1.13.0
[587475ba] Flux v0.11.3
[f6369f11] ForwardDiff v0.10.25
[c91e804a] Gadfly v1.3.4
[4b11ee91] Gaston v1.0.5
[bb4c363b] GridInterpolations v1.1.2
[f67ccb44] HDF5 v0.16.5
[c601a237] Interact v0.10.4
[b6b21f68] Ipopt v0.7.0
[682c06a0] JSON v0.21.3
[4076af6c] JuMP v0.21.5
[b4f0291d] LazySets v1.56.1
[a40420fb] LocalApproximationValueIteration v0.4.2
[db97f5ab] LocalFunctionApproximation v1.1.0
[e12ccd36] MCTS v0.4.7
[b8f27783] MathOptInterface v0.9.22
[283c5d60] MeshCat v0.13.2
[b8a86587] NearestNeighbors v0.4.9
[510215fc] Observables v0.4.0
[1af16e33] Omega v0.1.1 https://github.com/zenna/Omega.jl.gitmaster
[bac558e1] OrderedCollections v1.4.1
[08074719] POMDPModelTools v0.3.12
[182e52fb] POMDPPolicies v0.4.2
[e0d0a172] POMDPSimulators v0.3.13
[a93abf59] POMDPs v0.9.4
[d96e819e] Parameters v0.12.3
[67491407] Polyhedra v0.6.17
[92933f4c] ProgressMeter v1.7.2
[8af83fb2] QuickPOMDPs v0.2.13
[1e97bd63] ReachabilityAnalysis v0.16.5
[38ceca67] RobotDynamics v0.3.4
[74be38bb] RobotZoo v0.2.3
[6038ab10] Rotations v1.3.0
[90137ffa] StaticArrays v1.4.2
[2913bbd2] StatsBase v0.33.16
[f3b207a7] StatsPlots v0.14.30
[0c5d862f] Symbolics v3.2.3
[7770976a] TrajOptPlots v0.2.0
[c79d492b] TrajectoryOptimization v0.5.0
[ddb6d928] YAML v0.4.7
```

Table C.1.: Installed packages for the simulator

Table C.2.: Vehicle parameters (Small vehicle adapted from [Althoff and Wuersching, 2020])

Parameters	Symbol	Value	Unit
Parameters for the geometric dimensions			
$\theta^{\text{geom}} = [l, w, l_{wb}, l_h, l_b]^T$			
Vehicle length	l	4.298	$[m]$
Vehicle width	w	1.674	$[m]$
Wheelbase	l_{wb}	2.391	$[m]$
Vehicle height	l_h	1.582	$[m]$
Wheel spacing	l_b	1	$[m]$
Driving parameters $\theta^{\text{driv}} = [a_{\max}, \underline{v_\delta}, \overline{v_\delta}, \underline{\delta}, \overline{\delta}, \underline{v}, \overline{v}]^T$			
Minimal steering angle	$\underline{\delta}$	-0.910	$[rad]$
Maximal steering angle	$\overline{\delta}$	0.910	$[rad]$
Minimal steering velocity	$\underline{v_\delta}$	-0.4	$[rad/s]$
Maximal steering velocity	$\overline{v_\delta}$	0.4	$[rad/s]$
Minimal velocity	$\underline{v}$	-2.222	$[m/s]$
Maximal velocity	$\overline{v}$	13.889	$[m/s]$
Switching velocity	v_S	4.755	$[m/s]$
Maximal acceleration	$a_{\max}$	11.5	$[m/s^2]$
Artificial geometric parameters $\theta^{\text{art}} = [r_m, l_{vp}]$			
Radius for Minkowksi-Sum	r_m	0.3	$[m]$
Minimal distance between vehicle and pedestrian	$\underline{l_{vp}}$	0.8	$[m]$



(a)

(b)

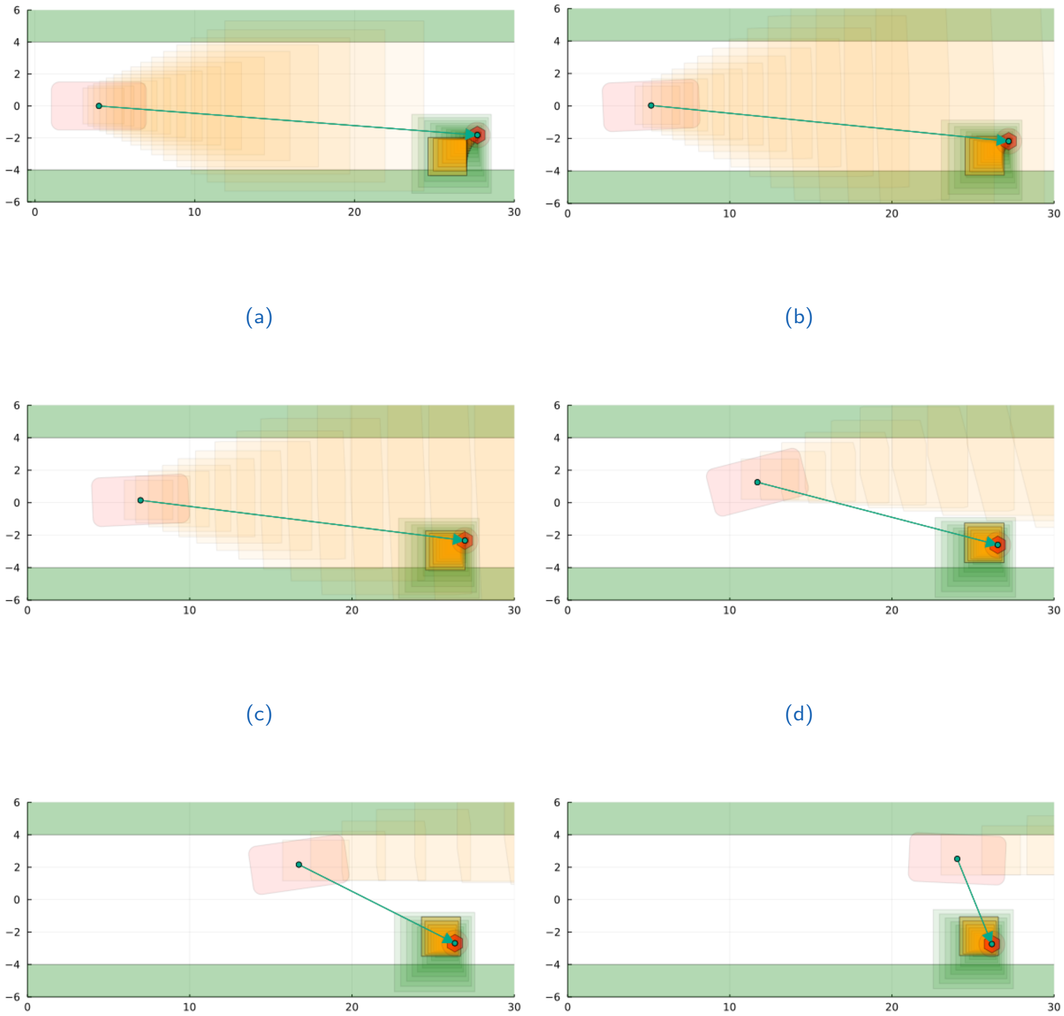
(c)

(d)

(e)

(f)

Figure C.1.: A vehicle (represented by a pink rectangle) with planned trajectory (blue dots) and orange reachable sets is driving to a fixed target state. A pedestrian is walking in the right direction without changing his/her intention.



(a)

(b)

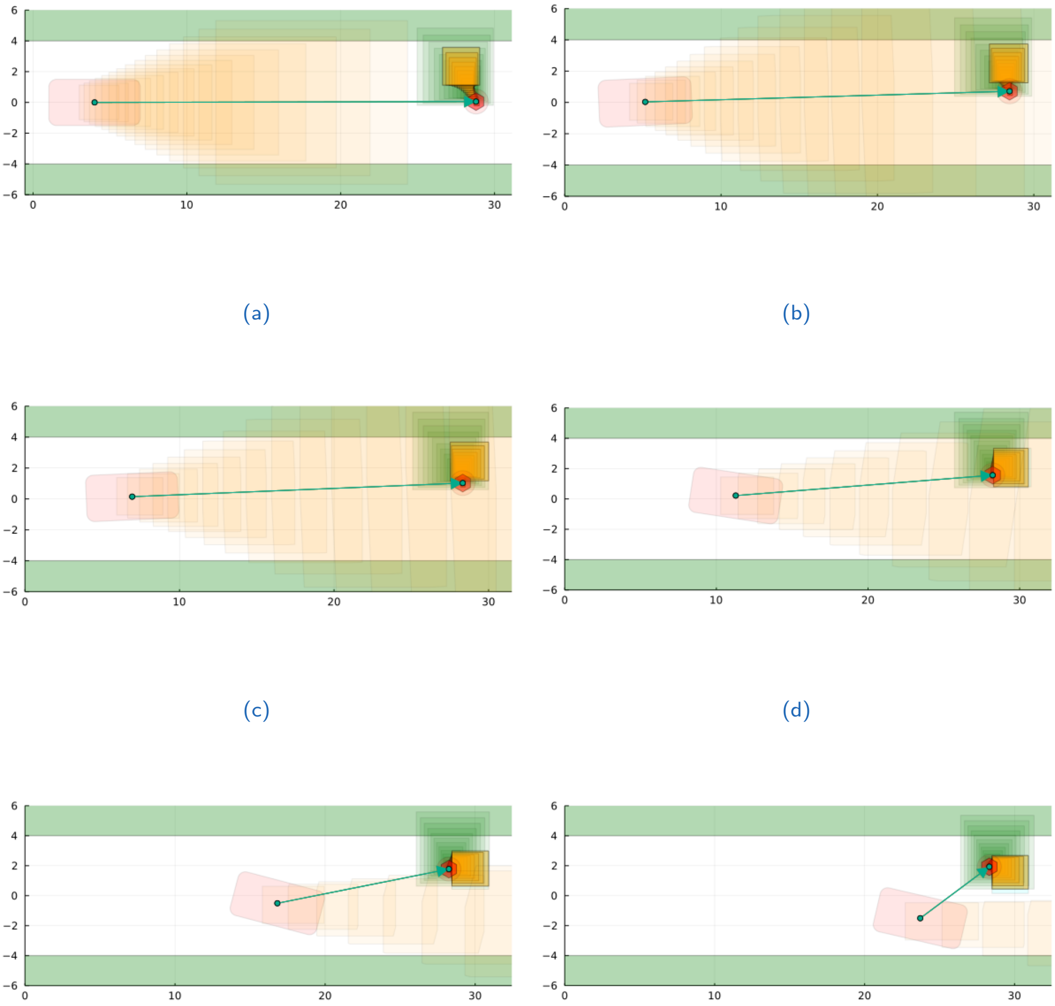
(c)

(d)

(e)

(f)

Figure C.2.: A vehicle (represented by a pink rectangle) with planned trajectory (blue dots) and orange reachable sets is driving to a fixed target state. A pedestrian walks to the bottom sidewalk (represented as a green area). After some time, it has an intention change.



(a)

(b)

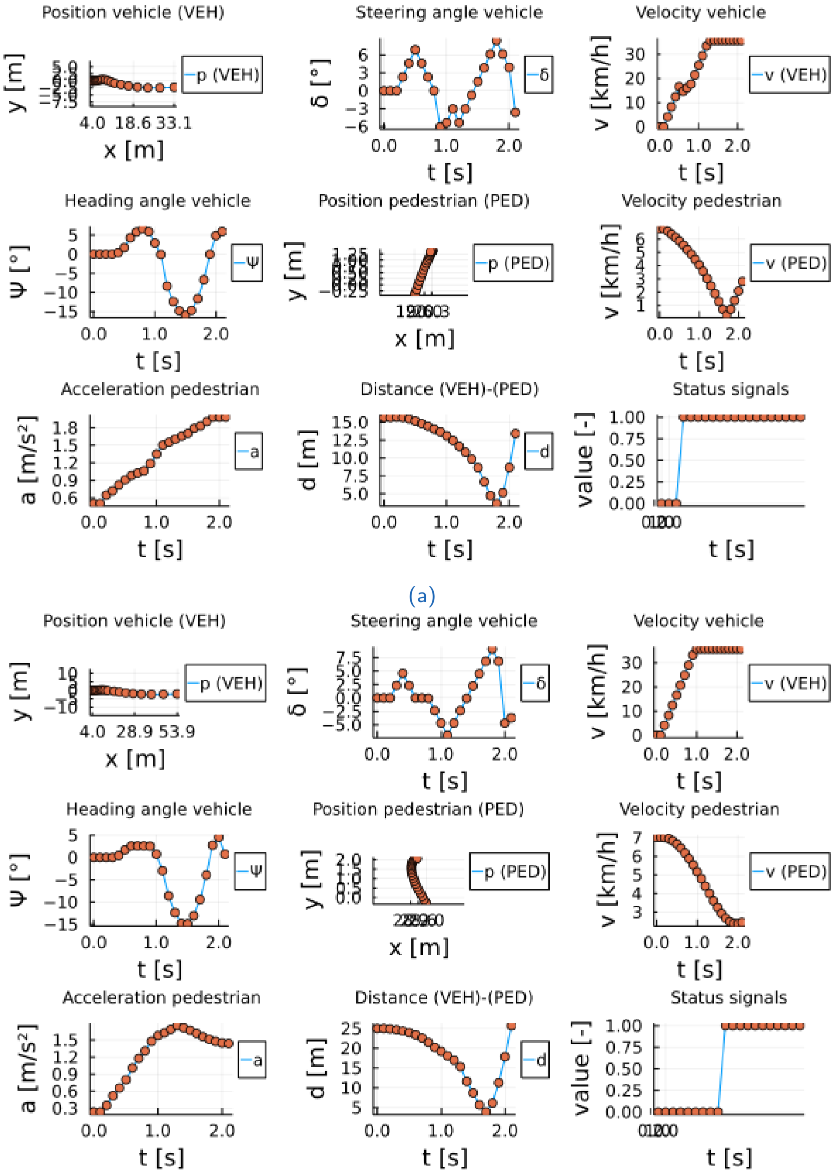
(c)

(d)

(e)

(f)

Figure C.3.: A vehicle (represented by a pink rectangle) with planned trajectory (blue dots) and orange reachable sets is driving to a fixed target state. A pedestrian is walking to the bottom sidewalk (represented as a green area). After some time, it has an intention change.



(a)

(b)

Figure C.4.: Each simulation run has the state-change of the pedestrian and vehicle (constrained position, velocity and acceleration and control inputs)

C.2. Algorithms

This section should give some further ideas to sample get funnels on manifolds. Figure C.5 gives an example of a curved street. The green sidewalk consists of green ellipses (cells) and the curved street with grey ellipses. The topology $\mathcal{T}$ is describing which of the cells $\mathcal{X}_i, \mathcal{X}_j$ are adjacent. For example $\mathcal{X}_5$ is adjacent with $\mathcal{X}_4$ (overlapping). Each cell has its own coordinate system (compare section A.1). If an agent with position $\boldsymbol{p}_{k_i} = \begin{bmatrix} p_{x,k_i} & p_{y,k_i} \end{bmatrix}^T$ and state $\boldsymbol{x}_{k_i} = \begin{bmatrix} p_{x,k_i} & p_{y,k_i} & v_{x,k_i} & v_{y,k_i} \end{bmatrix}^T$ is in one cell of the manifold $\mathcal{X}_q$ with $\boldsymbol{p}_{k_i} \in \mathcal{X}_q$ we could use a matrix-multiplication with matrix $\boldsymbol{A}^{\theta,t}$ for a translation with vector $\boldsymbol{t} = \begin{bmatrix} t_x & t_y \end{bmatrix}^T$:

$$\begin{bmatrix} p_{x,k_{i+1}} \\ p_{y,k_{i+1}} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}^{\theta,t},\ \theta=0} \cdot \begin{bmatrix} p_{x,k_i} \\ p_{y,k_i} \\ 1 \end{bmatrix} \tag{C.1}$$

If we have in each cell $\mathcal{X}_q$ a translation $\boldsymbol{t}$ by a constant matrix multiplication, we could represent the manifold with a vector-field (arrows in each cell). Instead of using a translation vector $\boldsymbol{t}$ we could use a probabilistic operator on each cell $\mathcal{X}_q$ (e.g. depending on nearest mean value), with the conditional probability $p(\boldsymbol{x}_{k_{i+1}}|\boldsymbol{x}_{k_i})$ between the current state $\boldsymbol{x}_{k_i}$ and the following state $\boldsymbol{x}_{k_{i+1}}$ with Markov property:

$$p(\boldsymbol{x}_{k_{i+1}}|\boldsymbol{x}_{k_i}) = \frac{p(\boldsymbol{x}_{k_{i+1}}, \boldsymbol{x}_{k_i})}{p(\boldsymbol{x}_{k_i})} \tag{C.2}$$

We could marginalize the probability $p(\boldsymbol{x}_{k_{i+1}}|\boldsymbol{x}_{k_i})$ to get the probability of the future state $p(\boldsymbol{x}_{k_{i+1}})$:

$$p(\boldsymbol{x}_{k_{i+1}}) = \int p(\boldsymbol{x}_{k_{i+1}}|\boldsymbol{x}_{k_i}) p(\boldsymbol{x}_{k_i}) d\boldsymbol{x}_{k_i} \tag{C.3}$$

If we assume a model like in [Ellis et al., 2009]:

$$\boldsymbol{x}_{k_{i+1}} = \boldsymbol{x}_{k_i} + \underbrace{\mathbf{f}(\boldsymbol{x}_{k_i})}_{\text{state change } \Delta\boldsymbol{x}_{k_{i,i+1}}} + \text{noise} \tag{C.4}$$

The state change $\Delta\boldsymbol{x}_{k_{i,i+1}} = \boldsymbol{x}_{k_{i+1}} - \boldsymbol{x}_{k_i}$ is modelled by a nonlinear function $\mathbf{f}(\boldsymbol{x}_{k_i})$. We could assume that the $\Delta\boldsymbol{x}_{k_{i,i+1}} = \mathcal{N}(\boldsymbol{\mu}_q, \Sigma_q)$ is modelled by a (multivariate) normal distribution with mean value $\boldsymbol{\mu}_q$ and covariance-matrix Σ_q. This operator $\Delta\boldsymbol{x}_{k_{i,i+1}} = \mathcal{N}(\boldsymbol{\mu}_q, \Sigma_q)$ is only valid for the cell $\mathcal{X}_q$. With Monte-Carlo-Sampling one could use the algorithm 2 for a cell to compute the state trajectory inside a cell. Like in [Ellis et al., 2009] this approach could be used with vector-fields and this approach was used in figure 3.25 for computing the next particles. It stops when the future state is outside the boundaries of the cell. The algorithm 2 was only for a single vector $\boldsymbol{x}_{k_i}$ representing the state-change in a manifold-cell. But we could extend the ideas to more general algorithms. We could use sets $\mathcal{X}_{k_i}$ and set-deformation-techniques from section 3.3 instead of vectors $\boldsymbol{x}_{k_i}$ in an urban environment represented as a manifold $\mathcal{M}$ (section 3.5) to compute funnels (section 3.3 and 3.4). We could also use probabilistic or group operations only valid for a particular cell. Algorithm 3 shows the brute-force algorithm to compute all motion plans in the presence of the movement prediction of pedestrians or $\{\hat{\mathcal{Y}}^p_{k_q}\}_{k_q \in \{k_i, k_{i+1} \ldots, k_j\}}$ other road users (vehicles or busses). We assume a constant countable control input set $\mathcal{U}^v = \{\boldsymbol{u}^v_1, \ldots, \boldsymbol{u}^v_n\}$. The trajectory could be sampled afterward to get a trajectory from the funnel (set of sets). Algorithm 4 could define reachability analysis in a manifold cell to ensure physical valid trajectories. We could have in each cell $\mathcal{X}_q$ of the manifold different control input sets $\mathcal{U}$ (compare section 2.1 for reachability analysis and modelling the environment section 3.5). For the reachability analysis with zonotopes we have the algorithm 5. We have the input and initial state set $\mathcal{X}_{k_i} := \mathcal{Z}_{k_i} = \{\boldsymbol{c}_{k_i}, \mathcal{G}_{k_i}\}$ and the control-input set $\mathcal{U} := \mathcal{Z}_{k_i} = \{\boldsymbol{c}_{k_i}, \mathcal{G}_{k_i}\}$ which is constant and only defined in a manifold cell. A general algorithm for set deformation in a manifold cell is proposed in algorithm 6. The human behavior could be described locally with some decisions, which are specific for a local environment and done by sub-goals. The behavior could be described by a sum of local decisions lead to a global realization of the observed trajectory. Algorithm 7 is proposing a pseudo-algorithm for predicting the funnel. Algorithm 8 is applying it to

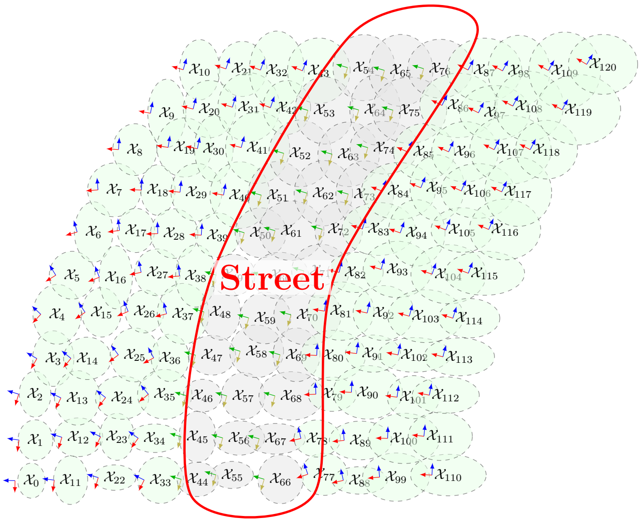

$\mathcal{X}_{10}$ $\mathcal{X}_{21}$ $\mathcal{X}_{32}$ $\mathcal{X}_{43}$ $\mathcal{X}_{54}$ $\mathcal{X}_{65}$ $\mathcal{X}_{76}$ $\mathcal{X}_{87}$ $\mathcal{X}_{98}$ $\mathcal{X}_{109}$ $\mathcal{X}_{120}$
$\mathcal{X}_{9}$ $\mathcal{X}_{20}$ $\mathcal{X}_{31}$ $\mathcal{X}_{42}$ $\mathcal{X}_{53}$ $\mathcal{X}_{64}$ $\mathcal{X}_{75}$ $\mathcal{X}_{86}$ $\mathcal{X}_{97}$ $\mathcal{X}_{108}$ $\mathcal{X}_{119}$
$\mathcal{X}_{8}$ $\mathcal{X}_{19}$ $\mathcal{X}_{30}$ $\mathcal{X}_{41}$ $\mathcal{X}_{52}$ $\mathcal{X}_{63}$ $\mathcal{X}_{74}$ $\mathcal{X}_{85}$ $\mathcal{X}_{96}$ $\mathcal{X}_{107}$ $\mathcal{X}_{118}$
$\mathcal{X}_{7}$ $\mathcal{X}_{18}$ $\mathcal{X}_{29}$ $\mathcal{X}_{40}$ $\mathcal{X}_{51}$ $\mathcal{X}_{62}$ $\mathcal{X}_{73}$ $\mathcal{X}_{84}$ $\mathcal{X}_{95}$ $\mathcal{X}_{106}$ $\mathcal{X}_{117}$
$\mathcal{X}_{6}$ $\mathcal{X}_{17}$ $\mathcal{X}_{28}$ $\mathcal{X}_{39}$ $\mathcal{X}_{50}$ $\mathcal{X}_{61}$ $\mathcal{X}_{72}$ $\mathcal{X}_{83}$ $\mathcal{X}_{94}$ $\mathcal{X}_{105}$ $\mathcal{X}_{116}$
$\mathcal{X}_{5}$ $\mathcal{X}_{16}$ $\mathcal{X}_{27}$ $\mathcal{X}_{38}$ Street $\mathcal{X}_{82}$ $\mathcal{X}_{93}$ $\mathcal{X}_{104}$ $\mathcal{X}_{115}$
$\mathcal{X}_{4}$ $\mathcal{X}_{15}$ $\mathcal{X}_{26}$ $\mathcal{X}_{37}$ $\mathcal{X}_{48}$ $\mathcal{X}_{59}$ $\mathcal{X}_{70}$ $\mathcal{X}_{81}$ $\mathcal{X}_{92}$ $\mathcal{X}_{103}$ $\mathcal{X}_{114}$
$\mathcal{X}_{3}$ $\mathcal{X}_{14}$ $\mathcal{X}_{25}$ $\mathcal{X}_{36}$ $\mathcal{X}_{47}$ $\mathcal{X}_{58}$ $\mathcal{X}_{69}$ $\mathcal{X}_{80}$ $\mathcal{X}_{91}$ $\mathcal{X}_{102}$ $\mathcal{X}_{113}$
$\mathcal{X}_{2}$ $\mathcal{X}_{13}$ $\mathcal{X}_{24}$ $\mathcal{X}_{35}$ $\mathcal{X}_{46}$ $\mathcal{X}_{57}$ $\mathcal{X}_{68}$ $\mathcal{X}_{79}$ $\mathcal{X}_{90}$ $\mathcal{X}_{101}$ $\mathcal{X}_{112}$
$\mathcal{X}_{1}$ $\mathcal{X}_{12}$ $\mathcal{X}_{23}$ $\mathcal{X}_{34}$ $\mathcal{X}_{45}$ $\mathcal{X}_{56}$ $\mathcal{X}_{67}$ $\mathcal{X}_{78}$ $\mathcal{X}_{89}$ $\mathcal{X}_{100}$ $\mathcal{X}_{111}$
$\mathcal{X}_{0}$ $\mathcal{X}_{11}$ $\mathcal{X}_{22}$ $\mathcal{X}_{33}$ $\mathcal{X}_{44}$ $\mathcal{X}_{55}$ $\mathcal{X}_{66}$ $\mathcal{X}_{77}$ $\mathcal{X}_{88}$ $\mathcal{X}_{99}$ $\mathcal{X}_{110}$

Figure C.5.: Curved street (grey) and sidewalk (light green) as a manifold

the set-based-movement prediction with static support functions, which are getting used in next section. Before we get to an evaluation we also have to define the performance metrics for the set-based prediction. We model the urban environment as a topological space $(\mathcal{X}, \mathcal{T})$[1] (compare section 3.5).

[1] See definitions and short introduction to differential geometry (section A.1)

Algorithm 2: Algorithm for Monte-Carlo Sampling in a manifold cell

```
Input: $\boldsymbol{p}_{k_i}, \mathcal{X}_q, \boldsymbol{\mu}_q, \Sigma_q,\ \boldsymbol{p}_{k_i} \in \mathcal{X}_q$
Output: $\mathcal{L} = \{\boldsymbol{p}_{k_w}\}_{w=i}^{j},\ k_i < k_j\ \forall\ k_i, k_j \in \mathbb{T}$
$\mathcal{L} = \{\}$                                          ▷ empty list for collecting states[a]
$\boldsymbol{p}_{k_w} \leftarrow \boldsymbol{p}_{k_i}$                                  ▷ $\boldsymbol{p}_{k_w}$ as a dummy variable[b]
while $\boldsymbol{p}_{k_w} \in \mathcal{X}_q$                                               ▷ [c]
    $\mathcal{L} \leftarrow \boldsymbol{p}_{k_w}$
    $\boldsymbol{p}_{k_{w+1}} = \boldsymbol{p}_{k_w} + \mathcal{N}(\boldsymbol{\mu}_w, \Sigma_w)$
    $w \leftarrow w + 1$                                   ▷ $w$ as a counting variable
end while
$j \leftarrow w$                                          ▷ $j$ defining the future index
```

[a] $\mathcal{L} = \{\}$ is defining a list. $\mathcal{L} \leftarrow a$ would mean the list is getting the variable a like $\mathcal{L} = \{a\}$. $\mathcal{L} \leftarrow b$ would extend the list $\mathcal{L} = \{a, b\}$

[b] $\boldsymbol{p}_{k_w}$ is a new defined dummy-variable and the name of the variable could be changed. The variable $\boldsymbol{p}_{k_w}$ is not defined as a list like $\mathcal{L}$ beforehand, so that the dummy variable gets the position value of $\boldsymbol{p}_{k_i}$ like the statement $\boldsymbol{p}_{k_w} = \boldsymbol{p}_{k_i}$

[c] Is the position of the agent $\boldsymbol{p}_{k_w}$ at time-step $k_w \in \mathbb{T}$ still in the manifold cell $\mathcal{X}_q$

Algorithm 3: Brute force algorithm for computing all motion plans with movement prediction of road-users

Input:

Pedestrian:

- Movement prediction sets of pedestrian $\{\hat{\mathcal{Y}}^p_{k_q}\}_{k_q \in \{k_i, k_{i+1} \ldots, k_j\}}$

Vehicle:

- Current vehicle state $\boldsymbol{x}^v_{k_i}$
- Constant and countable control input set $\mathcal{U}^v = \{\boldsymbol{u}^v_1, \ldots, \boldsymbol{u}^v_n\}$
- Target state $\boldsymbol{x}^{v,\text{target}}_{k_i}$
- Driving parameters $\theta^{\text{driv}} = [a_{\max}, \underline{v}_\delta, \overline{v}_\delta, \underline{\delta}, \overline{\delta}, \underline{v}, \overline{v}]^T$
- Geometrical parameters $\theta^{\text{geom}} = [l_{wb}, r_m, l_b]^T$

Output:

Vehicle geometrical state sets: $\{\boxed{\mathcal{X}}^v_{k_q}\}_{k_q \in \{k_i, k_{i+1}, \ldots, k_j\}}$

```
for $q = i \rightarrow j$:
    for $\boldsymbol{u}^v_{k_q} \in \{\boldsymbol{u}^v_1, \ldots, \boldsymbol{u}^v_n\}$:
        $\boldsymbol{x}^v_{k_{q+1}} = \mathbf{f}(\boldsymbol{x}^v_{k_q}, \boldsymbol{u}^v_{k_q})$        ▷ Use system model to compute next state
        $\boxed{\mathcal{X}}^v_{k_{q+1}} = \mathbf{g}(\boldsymbol{x}^v_{k_{q+1}}, \theta)$        ▷ Compute next vehicle set
        $\boldsymbol{j} = \mathrm{J}(\boxed{\mathcal{X}}^v_{k_{q+1}})$        ▷ Evaluate cost function
```

Algorithm 4: Algorithm for reachability analysis in a manifold cell

Input: $\mathcal{X}_{k_q}, \mathcal{U}$

Output: $\mathcal{L} = \{\mathcal{X}_{k_q}\}^j_{q=i}, \; k_i < k_j, k_i, k_j \in \mathbb{T}$

```
$\mathcal{L} = \{\}$        ▷ empty list for collecting state sets
$\mathcal{L} \leftarrow \mathcal{R}(k_j, k_i, \mathcal{X}_0)$        ▷ Compare algorithm 1
```

Algorithm 5: Algorithm for set-deformation of a zonotope in a manifold cell

```
Input: $\mathcal{X}_{k_i} = \mathcal{Z}_{k_i}(\boldsymbol{c}_{k_i}, \mathcal{G}_{k_i}), \mathcal{X}_q,\ \boldsymbol{c}_{k_i} \in \mathcal{X}_q$
Output: $\mathcal{L} = \{\boldsymbol{x}_{k_w}, \mathcal{Z}_{k_w}\}_{w=i}^{j},\ k_i < k_j, k_i, k_j \in \mathbb{T}$
$\mathcal{L} = \{\}$                                        ▷ [a]
$\boldsymbol{x}_{k_w} \leftarrow \boldsymbol{c}_{k_i}$
$\mathcal{X}_{k_w} \leftarrow \mathcal{X}_{k_i}$            ▷ All sets $\mathcal{X}_{k_w}, \mathcal{X}_{k_i}$ here are zonotopes [b]
$\mathcal{L} \leftarrow (\boldsymbol{x}_{k_w}, \mathcal{X}_{k_w})$
while $\boldsymbol{x}_{k_w} \in \mathcal{X}_q$               ▷ [c]
    $\mathcal{X}_{k_w} \leftarrow \mathfrak{G} * \mathcal{X}_{k_w}$        ▷ Closed set deformation $(\mathfrak{G}, *)$ [d]
    $\boldsymbol{x}_{k_w} \leftarrow \boldsymbol{x}_{k_w} \sim \mathcal{X}_{k_w}$   ▷ getting a sample point $\boldsymbol{x}_{k_w} \in \mathcal{X}_{k_w}$
    $\mathcal{L} \leftarrow (\boldsymbol{x}_{k_w}, \mathcal{X}_{k_w})$
end while
```

[a] empty list for collecting tuples of state vectors and zonotopes

[b] For a certain number of mappings the zonotope is closed. An example is the matrix multiplication $\mathcal{Z}_{k_{i+1}} = \boldsymbol{A} \cdot \mathcal{Z}_{k_i}\ \boldsymbol{A} \in \mathbb{R}^{n \times n}$

[c] To prevent an infinite loop, the sampling of $\boldsymbol{x}_{k_w}$ should be possible outside of $\mathcal{X}_q$. This depends from the set-deformation-operator $\mathfrak{G}$ (operation on a zonotope) and the set $\mathcal{X}_q$.

[d] Here it is defined as a local and constant deformation operator $\mathfrak{G}$ and it is only operating in the manifold cell. Set-deformations like the Minkowski-sum, rotations, translations of sets might be possible options. Another option is to use a time-dependent set operation $\mathfrak{G}_{k_w}\ k_w \in \mathbb{T}$. The set operation could be updated afterwards in a separated step.

Algorithm 6: General pseudo-algorithm for local set-deformations in a manifold-cell

```
Input: $\boldsymbol{x}_{k_i} \in \mathcal{X}_{k_i}, \boldsymbol{x}_{k_i} \in \mathcal{X}_q, \mathfrak{G}, \ \mathcal{X}_{k_i} \subseteq \mathcal{X}_q$ [a]
Output: $\mathcal{L} = \{(\boldsymbol{x}_{k_w}, \mathcal{X}_{k_w})\}_{w=i}^{j}, \ k_i < k_j \ \forall \ k_i, k_j \in \mathbb{T}$
$\mathcal{L} = \{\}$                                              ▷ [b]
$\boldsymbol{x}_{k_w} \leftarrow \boldsymbol{x}_{k_i}$
$\mathcal{X}_{k_w} \leftarrow \mathcal{X}_{k_i}$
$\mathcal{L} \leftarrow (\boldsymbol{x}_{k_w}, \mathcal{X}_{k_w})$
while $\boldsymbol{x}_{k_w} \in \mathcal{X}_q$                                  ▷ [c]
    $\mathcal{X}_{k_w} \leftarrow \mathfrak{G} * \mathcal{X}_{k_w}$                  ▷ Set deformation $(\mathfrak{G}, *)$ [d]
    $\boldsymbol{x}_{k_w} \leftarrow \boldsymbol{x}_{k_w} \sim \mathcal{X}_{k_w}$        ▷ getting a sample point $\boldsymbol{x}_{k_w} \in \mathcal{X}_{k_w}$
    $\mathcal{L} \leftarrow (\boldsymbol{x}_{k_w}, \mathcal{X}_{k_w})$
end while
```

[a] There might different possibilities to define a membership-function, e.g. in sets like ellipsoids, zonotopes the position of the center-point might define the membership-function $\boldsymbol{c} \in \mathcal{X}_q$. Another approach would be to define the membership with the amount of area intersecting with another manifold cell.

[b] empty list for collecting tuples of state vectors and state sets

[c] To prevent an infinite loop, the sampling of $\boldsymbol{x}_{k_w}$ should be possibly outside of $\mathcal{X}_q$. This depends from the set-deformation-operator $\mathfrak{G}$ and the set $\mathcal{X}_q$.

[d] Here it is defined as a local and constant deformation operator $\mathfrak{G}$ and it is only operating in the manifold cell. Set-deformations like the Minkowski-sum, rotations, translations of sets might be possible options. Another option is to use a time-dependent set operation $\mathfrak{G}_{k_w}$ $k_w \in \mathbb{T}$. The set operation could be updated afterwards in a separated step.

Algorithm 7: General set-based-movement-prediction

```
Input: Structure flexibility of the set, Machine learning model
Output: Predicted future state sets
for $q = i \rightarrow j$:
    Get data from measurements
    Predict one representation point $\hat{\boldsymbol{y}}_{k_q} \in \hat{\mathcal{Y}}_{k_q}$
    Predict the structure of the set $\hat{\mathcal{Y}}_{k_q}$
    Enlargement of $\hat{\mathcal{Y}}_{k_q}$ by considering an irreducible uncertainty
```

Algorithm 8: Set-based-movement-prediction with static support function

```
Input: Measurements $\{\boldsymbol{x}_{k_m}\}_{m=r}^{j}$, Vectors for the support function
    $\mathcal{L} = \{\boldsymbol{l}_1, \dots, \boldsymbol{l}_N\}$
Output: $\{\hat{\mathcal{Y}}_{k_q}\}_{q=i}^{j}$
    for $q = i \rightarrow j$:
        Get data from measurements with moving-time-window
        Predict one representation point $\hat{\boldsymbol{y}}_{k_q} \in \hat{\mathcal{Y}}_{k_q}$.
        Often the mean-value is predicted $\hat{\boldsymbol{y}} \approx \hat{\boldsymbol{x}} = \mathbb{E}[\boldsymbol{x}]$.
        Predict the structure of the set $\hat{\mathcal{Y}}_{k_q}$
        Enlargement of $\hat{\mathcal{Y}}_{k_q}$ by considering an irreducible uncertainty
        Enlargement by Minkowski-sum)
```

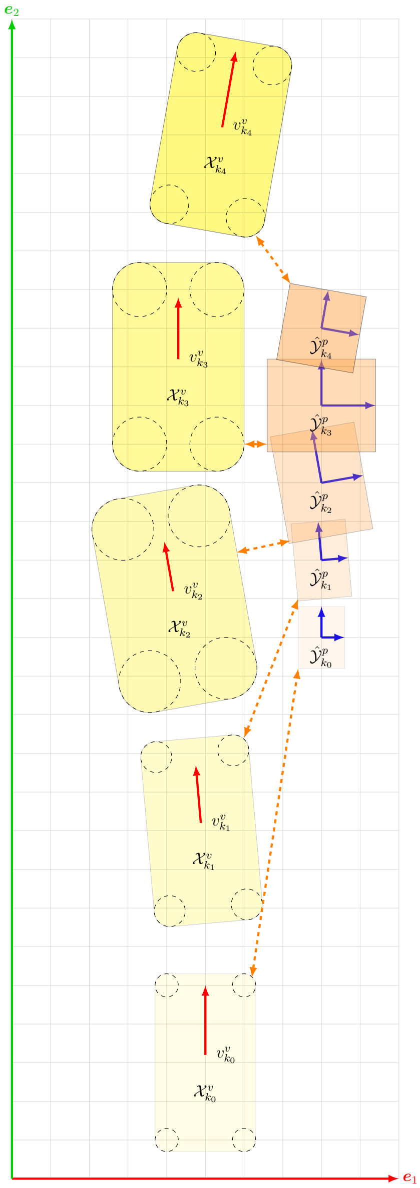


Figure C.6.: The vehicle sets $\{\mathcal{X}_{k_q}\}_{k_q\in\{k_0,k-4\}}$ and the pedestrian prediction sets $\{\mathcal{Y}_{k_q}\}_{k_q\in\{k_0,k-4\}}$. The minimal distance between $\min(\mathrm{D}(\mathcal{X}^v_{k_q},\hat{\mathcal{Y}}^p_{k_q}))\ \forall\ \boldsymbol{x}^v_{k_q}\in\mathcal{X}^v_{k_q},\hat{\boldsymbol{y}}^v_{k_q}\in\hat{\mathcal{Y}}^v_{k_q}$ the two sets are represented as orange double arrows.

Bibliography

[Aksjonov et al., 2019] Aksjonov, H., Beglerovic, H., Hartmann, M., Jugade, S., and Vaseur, C. (2019). On driver–vehicle–environment integration for multi–actuated ground vehicles safety advancement. *IEEE ICCVE 2019*, page 7.

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

[Alanwar et al., 2021] Alanwar, A., Koch, A., Allgöwer, F., and Johansson, K. H. (2021). Data-driven reachability analysis using matrix zonotopes. In *Learning for Dynamics and Control*, pages 163–175. PMLR.

[Allen and Pavone, 2016] Allen, R. and Pavone, M. (2016). A real-time framework for kinodynamic planning with application to quadrotor obstacle avoidance. In *AIAA Guidance, Navigation, and Control Conference*, page 1374.

[Althoff, 2010] Althoff, M. (2010). *Reachability analysis and its application to the safety assessment of autonomous cars*. PhD thesis, Technische Universität München.

[Althoff and Frehse, 2016] Althoff, M. and Frehse, G. (2016). Combining zonotopes and support functions for efficient reachability analysis of linear systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7439–7446. IEEE.

[Althoff et al., 2017] Althoff, M., Koschi, M., and Manzinger, S. (2017). Commonroad: Composable benchmarks for motion planning on roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 719–726. IEEE.

[Althoff and Wuersching, 2020] Althoff, M. and Wuersching, G. (2020). Commonroad:vehicle models (version 2020a).

[Armeni et al., 2016] Armeni, I. et al. (2016). 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Armeni et al., 2017] Armeni, I. et al. (2017). Joint 2d-3d-semantic data for indoor scene understanding. preprint, arXiv.

[Armeni et al., 2019] Armeni, I. et al. (2019). 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE International Conference on Computer Vision*.

[Asarin et al., 2000] Asarin, E., Bournez, O., Dang, T., and Maler, O. (2000). Approximate reachability analysis of piecewise-linear dynamical systems. In *International workshop on hybrid systems: Computation and control*, pages 20–31. Springer.

[Asarin et al., 2003] Asarin, E., Dang, T., and Girard, A. (2003). Reachability analysis of nonlinear systems using conservative approximation. In *International Workshop on Hybrid Systems: Computation and Control*, pages 20–35. Springer.

[Athans and Falb, 2013] Athans, M. and Falb, P. L. (2013). *Optimal control: an introduction to the theory and its applications*. Courier Corporation.

[Bargmann, 2014] Bargmann, C. e. a. (2014). *BRAIN 2025*. National Institutes of Health.

[Beal et al., 2018] Beal, L., Hill, D., Martin, R., and Hedengren, J. (2018). Gekko optimization suite. *Processes*, 6(8):106.

[Bengler et al., 2014] Bengler, K., Dietmayer, K., Farber, B., Maurer, M., Stiller, C., and Winner, H. (2014). Three decades of driver assistance systems: Review and future perspectives. *IEEE Intelligent transportation systems magazine*, 6(4):6–22.

[Bertsekas et al., 1995] Bertsekas, D. P. et al. (1995). Dynamic programming and optimal control. *Vol.*, 1.

[Bertsekas et al., 2000] Bertsekas, D. P. et al. (2000). *Dynamic programming and optimal control: Vol. 1*. Athena scientific Belmont.

[Bezanson et al., 2017] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

[Bock et al., 2019] Bock, J., Krajewski, R., Moers, T., Runde, S., Vater, L., and Eckstein, L. (2019). The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1929–1934. IEEE.

[Bogo et al., 2017] Bogo, F., Romero, J., Pons-Moll, G., and Black, M. J. (2017). *Dynamic FAUST: Registering Human Bodies in Motion (CVPR 2017)*. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR.

[Bogomolov et al., 2019] Bogomolov, S., Forets, M., Frehse, G., Potomkin, K., and Schilling, C. (2019). Juliareach: a toolbox for set-based reachability. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 39–44.

[Bonnin et al., 2014] Bonnin, S., Weisswange, T. H., Kummert, F., and Schmüdderich, J. (2014). Pedestrian crossing prediction using multiple context-based models. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 378–385. IEEE.

[Botvinick, 2007] Botvinick, M. M. (2007). Multilevel structure in behaviour and in the brain: a model of fuster's hierarchy. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1485):1615–1626.

[Boyd et al., 2004] Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[Brantley et al., 2018] Brantley, J. A. et al. (2018). Full body mobile brain-body imaging data during unconstrained locomotion on stairs, ramps, and level ground. *Scientific data*, 5.

[Braziunas, 2003] Braziunas, D. (2003). Pomdp solution methods. *University of Toronto*.

[Brunetti et al., 2018] Brunetti, A., Buongiorno, D., Trotta, G. F., and Bevilacqua, V. (2018). Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300:17–33.

[Cavalieri, 2007] Cavalieri, R. (2007). Lecture notes in "introduction to topology".

[Charles et al., 2017] Charles, J., Budvytis, I., and Cipolla, R. (2017). Real-time factored convnets: Extracting the x factor in human parsing. *British Machine Vision Conference 2017*.

[Chen, 2017] Chen, Z. (2017). A primer on neural signal processing. *IEEE Circuits and Systems Magazine*, 17(1):33–50.

[Cignoni and Callieri, 2008] Cignoni, P. and Callieri, M. (2008). *and Massimiliano Corsini and Matteo Dellepiane and Fabio Ganovelli and Guido Ranzuglia. "MeshLab: an Open-Source Mesh Processing Tool"*. Eurographics Italian Chapter Conference.

[Combrisson et al., 2019] Combrisson, E., Vallat, R., O'Reilly, C., Jas, M., Pascarella, A., Saive, A.-l., Thiery, T., Meunier, D., Altukhov, D., Lajnef, T., et al. (2019). Visbrain: a multi-purpose gpu-accelerated open-source suite for multimodal brain data visualization. *Frontiers in Neuroinformatics*, 13:14.

[Crivellari and Beinat, 2020] Crivellari, A. and Beinat, E. (2020). Lstm-based deep learning model for predicting individual mobility traces of short-term foreign tourists. *Sustainability*, 12(1):349.

[De Nicolao et al., 2007a] De Nicolao, G., Ferrara, A., and Giacomini, L. (2007a). Onboard sensor-based collision risk assessment to improve pedestrians' safety. *IEEE transactions on vehicular technology*, 56(5):2405–2413.

[De Nicolao et al., 2007b] De Nicolao, G., Ferrara, A., and Giacomini, L. (2007b). Onboard sensor-based collision risk assessment to improve pedestrians' safety. *IEEE transactions on vehicular technology*, 56(5):2405–2413.

[Delp et al., 2007] Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., and Thelen, D. G. (2007). Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering*, 54(11):1940–1950.

[Dolgov et al., 2008] Dolgov, D. et al. (2008). Practical search techniques in path planning for autonomous driving. *Ann Arbor*, 1001(48105):18–80.

[Dollár et al., 2007] Dollár, P., Rabaud, V., and Belongie, S. (2007). Learning to traverse image manifolds. *Advances in neural information processing systems*, 19:361.

[Doolin and Martin, 2013] Doolin, B. F. and Martin, C. F. (2013). *Introduction to differential geometry for engineers*. Courier Corporation.

[Drake et al., 2009] Drake, R., Vogl, A. W., and Mitchell, A. W. (2009). *Gray's anatomy for students E-book*. Elsevier Health Sciences.

[Dunning et al., 2017] Dunning, I., Huchette, J., and Lubin, M. (2017). Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320.

[Earman et al., 1986] Earman, J. et al. (1986). *A primer on determinism*, volume 37. Springer Science & Business Media.

[Egorov et al., 2017] Egorov, M., Sunberg, Z. N., Balaban, E., Wheeler, T. A., Gupta, J. K., and Kochenderfer, M. J. (2017). POMDPs.jl: A framework for sequential decision making under uncertainty. *Journal of Machine Learning Research*, 18(26):1–5.

[Eigen and Fergus, 2015] Eigen, D. and Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*.

[Eilbrecht et al., 2017] Eilbrecht, J. et al. (2017). Model-predictive planning for autonomous vehicles anticipating intentions of vulnerable road users by artificial neural networks. *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017.

[Ellis et al., 2009] Ellis, D., Sommerlade, E., and Reid, I. (2009). Modelling pedestrian trajectory patterns with gaussian processes. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1229–1234. IEEE.

[Essen and Glasser, 2016] Essen, D. and Glasser, M. (2016). *The Human Connectome Project: Progress and Prospects*. Cerebrum.

[et al., 2018] et al., S. A. I. L. (2018). Robotic operating system.

[Fahad et al., 2018] Fahad, M., Chen, Z., and Guo, Y. (2018). Learning how pedestrians navigate: A deep inverse reinforcement learning approach. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

[Fischl, 2012] Fischl, B. (2012). Freesurfer. *Neuroimage*, 62(2):774–781.

[Flohr et al., 2015] Flohr, F., Dumitru-Guzu, M., Kooij, J. F., and Gavrila, D. M. (2015). A probabilistic framework for joint pedestrian head and body orientation estimation. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1872–1882.

[Föllinger, 1985] Föllinger, O. (1985). *Regelungstechnik Hüthig Buch Verlag*. Heidelberg.

[Friedman et al., 2001] Friedman, J., Hastie, T., Tibshirani, R., et al. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

[Geiger et al., 2013] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237.

[Geronimo et al., 2009] Geronimo, D. et al. (2009). Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 7:1239–1258.

[Girard, 2005] Girard, A. (2005). Reachability of uncertain linear systems using zonotopes. In *International Workshop on Hybrid Systems: Computation and Control*, pages 291–305. Springer.

[Girard et al., 2006] Girard, A., Le Guernic, C., and Maler, O. (2006). Efficient computation of reachable sets of linear time-invariant systems with inputs. In *International workshop on hybrid systems: Computation and control*, pages 257–271. Springer.

[Goldhammer et al., 2013] Goldhammer, M., Gerhard, M., Zernetsch, S., Doll, K., and Brunsmann, U. (2013). Early prediction of a pedestrian's trajectory at intersections. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 237–242. IEEE.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

[Govea, 2010] Govea, A. D. V. (2010). *Incremental learning for motion prediction of pedestrians and vehicles*, volume 64. Springer, Science & Business Media.

[Grégoire and Bouillot, 1998] Grégoire, N. and Bouillot, M. (1998). Hausdorff distance between convex polygons. `http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html`. Accessed: 2021-04-22.

[Hackel et al., 2017] Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., and Pollefeys, M. (2017). *A new large-scale point cloud classification benchmark*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences.

[Hartmann, 2021] Hartmann, M. (2021). Python package reachab. `https://github.com/ga74kud/reachab`. Accessed: 2021-04-21.

[Hartmann, 774A] Hartmann, M. (GB000002569774A). Method for virtual testing of real environments with pedestrian interaction and drones.

[Hartmann et al., 2018a] Hartmann, M., Abaunza, H., Castillo, P., Stolz, M., and Watzenig, D. (2018a). Pedestrian in the loop: An approach using flying drones. In *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6. IEEE.

[Hartmann et al., 2018b] Hartmann, M., Ferrara, A., and Watzenig, D. (2018b). Data-based reachability analysis for movement prediction of pedestrians and motion planning. In *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–7. IEEE.

[Hartmann et al., 2017a] Hartmann, M., Patelli, E., and De Angelis, M. (2017a). Uncertainty quantification for motion planning. *NAFEMS World Congress 2017*.

[Hartmann et al., 2018c] Hartmann, M., Stolz, M., and Watzenig, D. (2018c). Movement prediction hypotheses for pedestrians and trajectory planning for cooperative driving systems. *SAE International Journal of Connected and Automated Vehicles*, 2(12-02-01-0002):17–26.

[Hartmann et al., 2017b] Hartmann, M., Viehweger, M., Desmet, W., Stolz, M., and Watzenig, D. (2017b). "pedestrian in the loop": An approach using virtual reality. In *2017 XXVI International Conference on Information, Communication and Automation Technologies (ICAT)*, pages 1–8. IEEE.

[Hartmann et al., 2018d] Hartmann, M., Viehweger, M., Stolz, M., Watzenig, D., Spitzer, M., and Desmet, W. (2018d). "pedestrian in the loop": An approach using augmented reality. Technical report, SAE Technical Paper.

[Hartmann and Watzenig, 2019a] Hartmann, M. and Watzenig, D. (2019a). Optimal motion planning with reachable sets of vulnerable road users. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 891–898. IEEE.

[Hartmann and Watzenig, 2019b] Hartmann, M. and Watzenig, D. (2019b). Pedestrians walking on reachable sets and manifolds. In *2019 IEEE International Conference on Mechatronics (ICM)*, volume 1, pages 562–569. IEEE.

[Hartmann, 049A] Hartmann, M. e. a. (GB000002562049A). Improved pedestrian prediction by using enhanced map data in automated vehicles.

[Hartmann, 400A] Hartmann, M. e. a. (GB000002563400A). Method and process for co-simulation with virtual testing of real environments with pedestrian interaction.

[Hartmann, 897A] Hartmann, M. e. a. (GB000002564897A). Method and process for motion planning in (un-)structured environments with pedestrians and use of probabilistic manifolds.

[He et al., 2018] He, Y. et al. (2018). A mobile brain-body imaging dataset recorded during treadmill walking with a brain-computer interface. *Scientific data*, 5.

[Hirakawa et al., 2018] Hirakawa, T., Yamashita, T., Tamaki, T., and Fujiyoshi, H. (2018). Survey on vision-based path prediction. In *International Conference on Distributed, Ambient, and Pervasive Interactions*.

[Huang et al., 2018] Huang, X. et al. (2018). The apolloscape open dataset for autonomous driving and its application. preprint, arXiv.

[Huang et al., 2019] Huang, X., Wang, P., Cheng, X., Zhou, D., Geng, Q., and Yang, R. (2019). The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2702–2719.

[Hubel and Wiesel, 1979] Hubel, D. H. and Wiesel, T. N. (1979). Brain mechanisms of vision. *Scientific American*, 241(3):150–163.

[Hug et al., 2018] Hug, R., Becker, S., Hübner, W., and Arens, M. (2018). Particle-based pedestrian path prediction using lstm-mdl models. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2684–2691. IEEE.

[Huth et al., 2016a] Huth, A., de Heer, W., Griffiths, T., et al. (2016a). Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532:453.

[Huth et al., 2016b] Huth, A., Lee, T., Nishimoto, S., Bilenko, N., Vu, A., and Gallant, J. (2016b). *Decoding the Semantic Content of Natural Movies from Human Brain Activity*. Frontiers in Systems Neuroscience.

[Ikeda et al., 2013] Ikeda, T., Chigodo, Y., Rea, D., Zanlungo, F., Shiomi, M., and Kanda, T. (2013). Modeling and prediction of pedestrian behavior based on the sub-goal concept. *Robotics*, 10:137–144.

[Imbens and Rubin, 2015] Imbens, G. W. and Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

[Jain et al., 2013] Jain, A. et al. (2013). Learning human pose estimation features with convolutional networks. preprint, arXiv.

[Janai et al., 2017] Janai, J. et al. (2017). Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. preprint, arXiv.

[Jaśkowski et al., 2018] Jaśkowski, W. et al. (2018). Reinforcement learning to run... fast. In *The NIPS'17 Competition: Building Intelligent Systems.*, pages 155–167. Springer.

[Jurafsky and Martin, 2020] Jurafsky, D. and Martin, J. H. (2020). Hidden markov models. *Speech and Language Processing*, page 1024.

[Karaman and Frazzoli, 2011] Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894.

[Karasev et al., 2016] Karasev, V., Ayvaci, A., Heisele, B., and Soatto, S. (2016). Intent-aware long-term prediction of pedestrian motion. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2543–2549. IEEE.

[Kato et al., 2018] Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monrroy, A., Ando, T., Fujii, Y., and Azumi, T. (2018). Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296. IEEE.

[Keller and Gavrila, 2013] Keller, C. G. and Gavrila, D. M. (2013). Will the pedestrian cross? a study on pedestrian path prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):494–506.

[Kidziński et al., 2018] Kidziński, Ł., Mohanty, S. P., Ong, C., Hicks, J., Francis, S., Levine, S., Salathé, M., and Delp, S. (2018). Learning to run challenge: Synthesizing physiologically accurate motion using deep reinforcement learning. In Escalera, S. and Weimer, M., editors, *NIPS 2017 Competition Book*. Springer, Springer.

[Kidziński et al., 2018] Kidziński, Ł., Mohanty, S. P., Ong, C. F., Huang, Z., Zhou, S., Pechenko, A., Stelmaszczyk, A., Jarosik, P., Pavlov, M., Kolesnikov, S., et al. (2018). Learning to run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 121–153. Springer.

[Kirk, 2004] Kirk, D. E. (2004). *Optimal control theory: an introduction*. Courier Corporation.

[Kitani et al., 2012] Kitani, K. M. et al. (2012). Activity forecasting. In *European Conference on Computer Vision. , , Heidelberg*, Berlin. Springer.

[Knill, 2005] Knill, O. (2005). Lecture notes in "dynamical systems". Harvard University.

[Kochenderfer, 2015] Kochenderfer, M. J. (2015). *Decision making under uncertainty: theory and application*. MIT press.

[Kocsis and Szepesvári, 2006] Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer.

[Kooij et al., 2014] Kooij, J. F., Schneider, N., and Gavrila, D. M. (2014). Analysis of pedestrian dynamics from a vehicle perspective. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1445–1450. IEEE.

[Kords, 2021] Kords, M. (2021). Straßenverkehrsunfälle in deutschland bis 2020.

[Koschi and Althoff, 2017a] Koschi, M. and Althoff, M. (2017a). Spot: A tool for set-based prediction of traffic participants. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1686–1693. IEEE.

[Koschi and Althoff, 2017b] Koschi, M. and Althoff, M. (2017b). Spot: A tool for set-based prediction of traffic participants. *IEEE Intelligent Vehicles Symposium (IV). IEEE*, 2017.

[Koubâa et al., 2017] Koubâa, A. et al. (2017). *Robot Operating System (ROS).*, volume 1. Springer.

[Kouvaritakis and Cannon, 2016] Kouvaritakis, B. and Cannon, M. (2016). Model predictive control. *Switzerland: Springer International Publishing*, page 38.

[Krajewski et al., 2018] Krajewski, R. et al. (2018). The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE.

[Kurzhanski and Varaiya, 2002] Kurzhanski, A. B. and Varaiya, P. (2002). On ellipsoidal techniques for reachability analysis. part i: external approximations. *Optimization methods and software*, 17(2):177–206.

[Lavalle, 1998] Lavalle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State University.

[LaValle, 2006] LaValle, S. M. (2006). *Planning algorithms*. University press, Cambridge.

[Lefèvre et al., 2014] Lefèvre, S., Vasquez, D., and Laugier, C. (2014). *A survey on motion prediction and risk assessment for intelligent vehicles*. Springer, Robomech Journal.

[Lewis et al., 2012] Lewis, F. L., Vrabie, D., and Syrmos, V. L. (2012). *Optimal control*. Wiley, John & Sons.

[Li and Yang, 2012] Li, Q. and Yang, X.-S. (2012). New walking dynamics in the simplest passive bipedal walking model. *Applied Mathematical Modelling*, 36(11):5262–5271.

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

[Lutz and Wendt, 2007] Lutz, H. and Wendt, W. (2007). *Taschenbuch der Regelungstechnik: mit MATLAB und Simulink*. Harri Deutsch Verlag.

[Manh and Alaghband, 2018] Manh, H. and Alaghband, G. (2018). Scene-lstm: A model for human trajectory prediction. *arXiv preprint arXiv:1808.04018*.

[Markram, 2006] Markram, H. (2006). The blue brain project. *Nature Reviews Neuroscience*, 7(2):153–160.

[Miani, 2009] Miani, M. (2009). Example: Spherical and cartesian grids. https://texample.net/tikz/examples/spherical-and-cartesian-grids/. Accessed: 2021-04-22.

[Mohamed et al., 2020] Mohamed, A., Qian, K., Elhoseiny, M., and Claudel, C. (2020). Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14424–14432.

[Müller and Guido, 2016] Müller, A. C. and Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.".

[Murphy, 2012] Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

[Naselaris et al., 2011a] Naselaris, T., Kay, K., Nishimoto, S., and Gallant, J. (2011a). *Encoding and decoding in fMRI*. Elsevier, NeuroImage.

[Naselaris et al., 2011b] Naselaris, T., Kay, K. N., Nishimoto, S., and Gallant, J. L. (2011b). Encoding and decoding in fmri. *Neuroimage*, 56(2):400–410.

[Neal, 2020] Neal, B. (2020). Lecture notes in introduction to causal inference from a machine learning perspective.

[Neogi et al., 2017] Neogi, S., Hoy, M., Chaoqun, W., and Dauwels, J. (2017). Context based pedestrian intention prediction using factored latent dynamic conditional random fields. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE.

[Nigg and Kuntze, 2012] Nigg, B. and Kuntze, G. (2012). *Human locomotion biomechanics*, pages 1–480. Eolss Publishers, Oxford, UK.

[Observatory, 2018] Observatory, E. R. S. (2018). Traffic safety basic facts 2018.

[Papageorgiou et al., 2015] Papageorgiou, M., Leibold, M., and Buss, M. (2015). *Optimierung*, volume 4. Springer.

[Pearl, 2003] Pearl, J. (2003). Statistics and causal inference: A review. *Test*, 12(2):281–345.

[Pearl, 2009] Pearl, J. (2009). *Causality*. Cambridge university press.

[Pearl et al., 2009] Pearl, J. et al. (2009). Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146.

[Pearl et al., 2016] Pearl, J., Glymour, M., and Jewell, N. P. (2016). *Causal inference in statistics: A primer*. John Wiley & Sons.

[Pearl and Mackenzie, 2018] Pearl, J. and Mackenzie, D. (2018). *The book of why: the new science of cause and effect*. Basic books.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

[Pek, 2020] Pek, C. F. (2020). *Provably Safe Motion Planning for Autonomous Vehicles Through Online Verification*. PhD thesis, Technische Universität München.

[Pellegrini et al., 2009] Pellegrini, S. et al. (2009). You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE.

[Peng et al., 2018] Peng, X. B., Abbeel, P., and Levine, S. (2018). *and van de Panne, Michael. "DeepMimic: Example-guided Deep Reinforcement Learning of Physics-based Character Skills"*. ACM Trans. Graph.

[Perez et al., 2019] Perez, D., Hasan, M., Shen, Y., and Yang, H. (2019). *AR-PED: A framework of augmented reality enabled pedestrian-in-the-loop simulation*. Simulation Modelling Practice and Theory.

[Peters et al., 2017] Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of causal inference*. The MIT Press.

[Rajamani, 2012] Rajamani, R. (2012). *Vehicle Dynamics and Control*. Springer.

[Rasouli et al., 2017] Rasouli, A., Kotseruba, I., and Tsotsos, J. K. (2017). Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 206–213.

[Redmon and Farhadi, 2017] Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[Rehder and Kloeden, 2015] Rehder, E. and Kloeden, H. (2015). Goal-directed pedestrian prediction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 50–58.

[Rehder et al., 2018] Rehder, E., Wirth, F., Lauer, M., and Stiller, C. (2018). Pedestrian prediction by planning using deep neural networks. In *Conference on Robotics and Automation (ICRA)*. IEEE.

[Robbin and Salamon, 2011] Robbin, J. W. and Salamon, D. A. (2011). Introduction to differential geometry. *ETH, Lecture Notes, preliminary version*, page 18.

[Robicquet et al., 2016a] Robicquet, A., Alahi, A., Sadeghian, A., Anenberg, B., Doherty, J., Wu, E., and Savarese, S. (2016a). Forecasting social navigation in crowded complex scenes. *arXiv preprint arXiv:1601.00998*.

[Robicquet et al., 2016b] Robicquet, A., Sadeghian, A., Alahi, A., and Savarese, S. (2016b). Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer.

[Robot Exploration Lab at Carnegie Mellon University, 2022] Robot Exploration Lab at Carnegie Mellon University (2022). RobotDynamics.jl.

[Rudenko et al., 2020] Rudenko, A., Palmieri, L., Herman, M., Kitani, K. M., Gavrila, D. M., and Arras, K. O. (2020). Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935.

[Safetynet, 2009] Safetynet, E. (2009). Pedestrians & cyclists. *EC, Brussels*.

[Salles et al., 2019] Salles, A., Bjaalie, J., Evers, K., Farisco, M., Tyr Fothergill, B., Guerrero, M., Maslen, H., Muller, J., Prescott, T., Stahl, B., Walter, H., Zilles, K., and Amunts, K. (2019). *The Human Brain Project: Responsible Brain Research for the Benefit of Society*. Neuron.

[Salles, 2019] Salles, A. e. a. (2019). *The Human Brain Project: Responsible Brain Research for the Benefit of Society*. Neuron.

[Sargent, J. Thomas et. al., 2022] Sargent, J. Thomas et. al. (2022). Quantecon: Open source code for economic modeling.

[Savva et al., 2019] Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al. (2019). Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347.

[Schilling, 2018] Schilling, C. (2018). *Fundamental techniques for the scalable analysis of systems.* PhD thesis, University of Freiburg, Freiburg im Breisgau, Germany.

[Schouwenaars, 2006] Schouwenaars, T. (2006). *Safe trajectory planning of autonomous vehicles*. PhD thesis, Diss. Massachusetts Institute of Technology.

[Schratter et al., 2019] Schratter, M., Hartmann, M., and Watzenig, D. (2019). Pedestrian collision avoidance system for autonomous vehicles. *SAE International Journal of Connected and Automated Vehicles*, 2(12-02-04-0021):279–293.

[Schulz and Stiefelhagen, 2015] Schulz, A. T. and Stiefelhagen, R. (2015). Pedestrian intention recognition using latent-dynamic conditional random fields. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 622–627. IEEE.

[Schulz, 2014] Schulz, J. (2014). Lecture notes in "angewandte differentialgeometrie".

[Shotton et al., 2013] Shotton, J. et al. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124.

[Silhouette of a Person, 2017] Silhouette of a Person (2017). Shape of walking man. https://tex.stackexchange.com/questions/386903/shape-of-walking-man-in-tikz. Accessed: 2021-04-16.

[Starek et al., 2014] Starek, J. et al. (2014). *Bidirectional fast marching trees: An optimal sampling-based algorithm for bidirectional motion planning*. Workshop on Algorithmic Foundations of Robotics.

[Straub et al., 2019] Straub, J. et al. (2019). The Replica dataset: A digital replica of indoor spaces. arXiv preprint.

[Sullivan, 2015] Sullivan, T. J. (2015). *Introduction to uncertainty quantification*, volume 63. Springer.

[Tavares et al., 2021] Tavares, Z., Koppel, J., Zhang, X., Das, R., and Solar-Lezama, A. (2021). A language for counterfactual generative models. In *International Conference on Machine Learning*, pages 10173–10182. PMLR.

[Thrun, 2002] Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, 45(3):52–57.

[Tompson et al., 2014] Tompson, J. J. et al. (2014). *Joint training of a convolutional network and a graphical model for human pose estimation*. Advances in neural information processing systems.

[Toshev and Szegedy, 2014] Toshev, A. and Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[Urtasun et al., 2006] Urtasun, R., Fleet, D. J., and Fua, P. (2006). 3d people tracking with gaussian process dynamical models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 238–245. IEEE.

[Van Der Maaten et al., 2009] Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13.

[Vasishta et al., 2017] Vasishta, P., Vaufreydaz, D., and Spalanzani, A. (2017). Natural vision based method for predicting pedestrian behaviour in urban environments. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE.

[Vasquez, 2010a] Vasquez, D. (2010a). *Incremental Learning for Motion Prediction of Pedestrians and Vehicles*. Springer.

[Vasquez, 2010b] Vasquez, D. (2010b). *Incremental learning for motion prediction of pedestrians and vehicles*. Springer.

[Vasquez, 2016] Vasquez, D. (2016). Novel planning-based algorithms for human motion prediction. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3317–3322. IEEE.

[Völz et al., 2016] Völz, B., Mielenz, H., Siegwart, R., and Nieto, J. (2016). Predicting pedestrian crossing using quantile regression forests. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 426–432. IEEE.

[Wei et al., 2016] Wei, S.-E. et al. (2016). Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Wiest et al., 2012] Wiest, J., Höffken, M., Kreßel, U., and Dietmayer, K. (2012). Probabilistic trajectory prediction with gaussian mixture models. In *2012 IEEE Intelligent Vehicles Symposium*, pages 141–146. IEEE.

[Wu et al., 2018] Wu, J., Ruenz, J., and Althoff, M. (2018). Probabilistic map-based pedestrian motion prediction taking traffic participants into consideration. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1285–1292. IEEE.

[Xia et al., 2018] Xia, F., Zamir, A. R., He, Z., Sax, A., Malik, J., and Savarese, S. (2018). Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079.

[Xia et al., 2012] Xia, L., Chen, C.-C., and Aggarwal, J. K. (2012). View invariant human action recognition using histograms of 3d joints. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE.

[Xue et al., 2018] Xue, H., Huynh, D. Q., and Reynolds, M. (2018). Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1186–1194. IEEE.

[Yamins et al., 2014a] Yamins, D., Hong, H., Cadieu, C., Solomon, E., Seibert, D., and DiCarlo, J. (2014a). *Performance-optimized hierarchical models predictneural responses in higher visual cortex*. Proceedings of the National Academy of Sciences of the United States of America.

[Yamins et al., 2014b] Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014b). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624.

[Zhan et al., 2019a] Zhan, W. et al. (2019a). Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. preprint, arXiv.

[Zhan et al., 2019b] Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kummerle, J., Konigshof, H., Stiller, C., de La Fortelle, A., et al. (2019b). Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*.

[Zhang et al., 2018] Zhang, Y. et al. (2018). Integrating kinematics and environment context into deep inverse reinforcement learning for predicting off-road vehicle trajectories. preprint, arXiv.

[Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22.

[Ziebart, 2010] Ziebart, B. D. (2010). *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD.

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

[Zio and Pedroni, 2013] Zio, E. and Pedroni, N. (2013). Methods for representing uncertainty. *Foundation for an Industrial Safety Culture, Toulouse, France*.