Universität der Bundeswehr München
Fakultät für Luft- und Raumfahrttechnik

Thema der Dissertation:

Towards object-related navigation for mobile robots

Verfasser: André Roskopf, M.Sc.

Promotionsausschuss:

Vorsitzender:	Prof. Mag. Dr. habil. Thomas Pany
1. Berichterstatter:	Prof. Dr.-Ing. Hans-Joachim Wünsche
2. Berichterstatter:	Prof. Dr. Dr. h.c. Raúl Rojas

Tag der Prüfung: 03.07.2019

Mit der Promotion erlangter akademischer Grad:

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

Neubiberg, den 26.05.2020

Keywords: Navigation, Motion Modelling, Autonomous Robots, Object Related Navigation

Acknowledgement

First and foremost I want to thank Prof. Wuensche for giving me the opportunity to work on such a fascinating topic. Furthermore, I want to thank Prof. Wuensche and Felix von Hundelshausen for their initial ideas on the concept of the object-related navigation approach. Without them this thesis might not have been written. A huge thanks goes to my colleagues at TAS for their input, feedback and advice. In particular I shared ideas with Michael Himmelsbach and was influenced by his work on LiDAR object perception. From Alois Unterholzner, Thorsten Luettel and Falk Hecker I learned about the fundamentals of robot motion control. Michael Manz, Michael Schweitzer and Sebastian Schneider gave me insights into the challenging topic of computer vision. You were amazing colleagues!

Last but certainly not least I want to thank my wonderful wife Claudia. During the creation of this thesis, you were always there to support me, although I did not have much to give in return, but love.

Abstract

Outdoor robot navigation describes the task of finding a drivable route to a predefined goal under consideration of uncertain and only partially observable environment information. Typically, the long-term destination is defined by coordinates in a global map. A trajectory planner carries out local obstacle avoidance in order to reach that goal. To this end, a simplification of the environment, such as a Cartesian occupancy grid, is generated, which represents each entity of the environment as an obstacle.

As a result each object, regardless of its type, such as tree, person or car, is abstracted as an obstacle. Thus, relevant information about the object is lost in the process. During map generation, data association and mapping routines constantly evaluate every object in the environment. I.e. a large amount of processing power is spent to map all obstacles, independent of their relevance for the actual planning task. Local planning and obstacle avoidance mechanisms then operate on those simplified environment representations. Due to this simplification they generate paths without an understanding of the environment they are meant to navigate in, which results in the decoupling of perception and action planning.

The presented approach, which is called object-related navigation, tries to amend some of the weaknesses that accompany global, map-based navigation approaches. Instead of incorporating the entire environment in the form of a heavily abstracted environment map into the planning routines, classified objects are considered, whose relevance for the task at hand is determined. They are directly incorporated into the navigation and reasoning algorithms in order to narrow down the gap between perception, planning and robot control. Relative spatial information, which is gained by exploiting the local sensor data of the robot is directly conveyed to the planning and control routines instead of transforming the information into a global representation. By training motion models per driving maneuver, such as *"overtake"* or *"turn left"*, robot motion generation and object motion prediction in relation to other objects is achieved. These models allow to form plans of the sort: *"follow lane, then turn left on next crossroad, ..."*, which simplifies the reasoning routines and allows plans to be directly conveyed to a human observer and vice versa. The results showed that trained maneuver models were able to steer the autonomous robot Munich Cognitive Autonomous Robot Car 3rd Generation (MuCAR-3), when a route description of the form: *"follow lane, then turn left on crossroad, ..."* was provided. Furthermore, the ability of the trained models to predict the motion of the robot and other traffic participants and their ability to predict real world percepts was evaluated.

Zusammenfassung

In der Robotik beschreibt der Begriff „Navigation“ die Aufgabe eines mobilen Roboters einen befahrbaren Pfad zu einem gegebenen Ziel unter Einbezug unsicherer und nur teilweise beobachtbarer Umgebungsinformationen zu finden. Das Ziel ist oft durch Koordinaten in einer globalen Karte (z.B. GPS-Koordinaten) vorgegeben. Ein lokaler Trajektorienplaner, dessen primäre Aufgabe darin besteht lokal wahrgenommene Hindernisse sicher zu umfahren, verwendet diese Zielkoordinaten als Richtungsinformation. Hindernisse werden oft mithilfe einer vereinfachten Repräsentation der Umgebung, zum Beispiel in Form einer kartesischen Rasterkarte, dargestellt. Jedes statische Objekt der Umgebung des Roboters wird hierbei als Menge von belegten Kartenzellen repräsentiert, unabhängig von Typ (z.B. Mensch, Fels) oder Wichtigkeit. Dynamische Objekte werden in auf Karten basierenden Navigationsansätzen gesondert behandelt, da diese oft in Form von objektumschließenden Quadern vorliegen und nur unter großem Aufwand in die kartierte Umgebung des Roboters überführt werden können.

Mit dem hier vorgestellten Navigationsansatz, genannt „objekt-relationale Navigation“, werden einige Schwächen von globalen, Karten-basierten Navigationsansätzen adressiert. Um nicht das gesamte Umfeld des Roboters zu abstrahieren werden lediglich klassifizierte Objekte, deren Relevanz für die aktuelle Navigationsaufgabe bewertet wurde, in die Planungsroutinen mit einbezogen. Somit muss nicht die gesamte Umgebung berücksichtigt werden, sondern nur die für die aktuelle Aufgabe relevanten Objekte. Statt in globalen Räumen zu planen werden egozentrische, räumliche Beziehungen zu den umgebenden Objekten direkt zwischen Wahrnehmung, Planung und Planausführung über erlernte Bewegungsmodelle ausgetauscht. Hiermit wird eine größtmögliche Verzahnung zwischen diesen, üblicherweise getrennten Bereichen sichergestellt.

Die Ergebnisse zeigen, dass unser Forschungsfahrzeug MuCAR-3 unter Verwendung von Routenbeschreibungen der Art: *“folge Spur, biege links an der Kreuzung ab, ...”* effizient autonom gesteuert werden konnte. Darüber hinaus wurde die Fähigkeit der erlernten Bewegungsmodelle gezeigt, zukünftige Objektpositionen zu prädizieren.

Contents

Symbols **XIII**
Abbreviations . . . XIII
Conventions . . . XIV
Indices . . . XV

Glossary **XVI**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Structure of the thesis . . . 3

2 Robot navigation - the three main paradigms **5**
2.1 Approaches to reactive robot navigation . . . 5
2.2 The deliberative paradigm . . . 8
2.3 The hybrid deliberative/reactive paradigm . . . 11

3 Outdoor robot navigation **15**
3.1 DARPA Grand Challenge 2004 / 2005 . . . 16
3.2 DARPA Urban Challenge 2007 . . . 18
3.3 "Stadtpilot" - project . . . 21
3.4 Grand Cooperative Driving Challenge . . . 22
3.5 European Land Robot Trial . . . 23
3.6 Our robot: MuCAR-3 . . . 25

4 Object-related navigation **29**
4.1 Human principles of navigation . . . 30
4.2 Principles of robot navigation - SLAM by example . . . 31
4.3 Problem statement . . . 36
4.4 Object perception mechanisms . . . 37
4.5 Environment representation . . . 40
4.5.1 Relation . . . 41
4.5.2 Data approximation . . . 48
4.6 Learning behavior models of relative motion . . . 56
4.6.1 Sequence generation . . . 57
4.6.2 The descriptive power of a relative representation . . . 62
4.6.3 Maneuver identification . . . 65
4.6.4 Approximation errors and measurement uncertainties . . . 73
4.6.5 Motion prediction . . . 74
4.7 Object related reasoning . . . 87
4.7.1 Maneuvers and plans . . . 89
4.7.2 Object-action-map . . . 90
4.7.3 Plan generation and execution . . . 99

4.7.4 Robot control 104
4.8 Results 110

5 Summary **125**

A Poses and homogeneous transformation matrices **127**
A.1 Homogeneous transformation matrices 127
A.2 Pose to homogeneous transformation matrix 127
A.3 Homogeneous transformation matrix to pose 128

B Bayes filter derivation **131**

C Maneuver sequences **133**

Bibliography **135**

Symbols

Abbreviations

MuCAR-3	Munich Cognitive Autonomous Robot Car 3rd Generation
ABE	autonomous benthic explorer
ACC	adaptive cruise control
AI	artificial intelligence
AuRA	autonomous robot architecture
AUV	automated underwater vehicle
C-ELROB	Civilian European Land Robot Trial
CAN	controller area network
CMU	Carnegie Mellon University
CPU	central processing unit
DARPA	Defense Advanced Research Projects Agency
DBN	dynamic bayesian network
DBW	drive-by-wire
DFG	Deutsche Forschungs-Gemeinschaft
DGPS	differential global positioning system
DOF	degrees of freedom
EKF	extended Kalman filter
ELROB	European Land Robot Trial
EOD	explosive ordnance disposal
ESP	electronic stability control
GCDC	Grand Cooperative Driving Challenge
GPS	global positioning system
HMI	human machine interface
HMM	hidden Markov model
HTAS	High Tech Automotive Systems
IMU	inertial measurement unit
INS	inertial navigation system
IPC	inter-process communication
KF	Kalman filter
KogMo-RTDB	KogniMobil Real Time Data Base
LiDAR	light detection and ranging
LM	Levenberg-Marquardt
LS	Least Squares
MarVEye-8	Multifocal active / reactive Vehicle Eye 8th Generation
MEMS	micro-electro-mechanical system
MPC	model predictive control
Navlab	navigation laboratory
OxTS	Oxford Technical Solutions
PF	particle filter

Pkw	passenger car, *german: Personenkraftwagen*
POMDP	paritially observable Markov decision process
psd	positive semidefinite
PWM	pulse width modulation
RADAR	radio detection and ranging
RDDF	route definition data format
RDT	rapidly exploring dense trees
RLS	recursive least squares
RNDF	road network definition file
SE(2)	special euclidean group of dimension 2
SLAM	simultaneous localization and mapping
SSD	solid state drive
STRIPS	stanford research institute problem solver
TNO	Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (Netherlands Organisation for Applied Scientific Research)
TOF	time-of-flight
UAV	unmanned aerial vehicle
UKF	unscented Kalman filter
UniBwM	University of the Bundeswehr Munich
USA	United States of America
V2I	vehicle-to-infrastructure
V2V	vehicle-to-vehicle
VaMoRs	test vehicle for autonomous mobility and computer vision, *german: Versuchsfahrzeug für autonome Mobilität und Rechnersehen*
VaMP	VaMoRs-Pkw

Conventions

$\mathbf{S}$	Cartesian coordinate system
$\mathbf{K}$	Kalman gain
L	arc length of a clothoid
$\hat{\mathbf{P}}$	updated (a posteriori) error covariance
$\mathbf{P}^*$	predicted (a priori) error covariance
$\mathbf{P}$	error covariance
$\mathbf{Q}$	process noise covariance
$\mathbf{R}$	measurement noise covariance
$\boldsymbol{l_k}$	landmark associations at time step k
${}^{O_j}A_{O_i}$	action A of object O_i relative to object O_j
$\mathbf{L_k}$	set of landmark associations up to time step k
χ	lane azimuth
c_0	curvature at the start of a lane segment
c_1	change of curvature of a lane segment
c	curvature
$\mathcal{N}(\boldsymbol{s} \mid \mu, \boldsymbol{\Sigma})$	gaussian probability density with mean μ and covariance $\boldsymbol{\Sigma}$ for variable $\boldsymbol{s}$
$\mathfrak{r}^{-1}$	unary operation with respect to a relation to get its inverse
ξ	location of a landmark
λ	steering angle

Ξ	environment map
η	normalizing constant
O_i	object with index i
π	mathematical constant to expresses the ratio of the circumference of a circle to its diameter
Π_{O_i}	plan Π of object O_i
$\oplus$	binary operation between two relations that realizes a concatenation
α	angle at which an object is observed
β	relative heading of an object with respect to α
$\ominus$	binary operation between two relations that realizes the computation of their difference
d	distance between two objects
$\mathcal{R}$	sequence of relations
$\mathfrak{r}(\alpha, \beta, d)$	relation between two objects
$\mathcal{E}$	the robot
$\boldsymbol{s}$	state of a system
$sgn(\cdot)$	the signum function
$\mathcal{T}$	trajectory in euclidean space
$\hat{(\cdot)}$	estimated values
$\mathbf{H}$	homogeneous transformation matrix
Θ	pitch angle
$\boldsymbol{p}$	point in 3D
Φ	roll angle
Ψ	yaw angle
$\boldsymbol{y}$	measurement vector
$(\cdot)^*$	predicted values
$\mathbf{\Phi}$	state transition matrix
$\mathcal{C}$	observation matrix
$\mathcal{S}$	innovation covariance
$\mathbf{Y}$	set of measurement vectors
k	time index
$\boldsymbol{u_k}$	scalar input of dynamic system and/or controller output at time step k
$\mathbf{U_k}$	set of input vectors of a dynamic system up to time step k
$\boldsymbol{x_k}$	state vector at time step k
$\hat{\boldsymbol{x}}$	corrected state vector
$\boldsymbol{x}^*$	predicted state vector
$\boldsymbol{y_k}$	measurement vector at time step k
$\boldsymbol{y}^*$	predicted measurement vector
$\mathbf{Y_k}$	set of measurement vectors up to time step k
$\boldsymbol{x}$	state vector
$\mathbf{X}$	matrix
$\mathbf{P}$	pose
s	scalar
$\boldsymbol{v}$	vector

Indices

t	time

Glossary

Defense Advanced Research Projects Agency

An agency of the ministry of defence of the United States of America (USA), which conducts research projects for the armed forces.

FireWire

FireWire, i.LINK or IEEE1394 is a bus system for serial data transfer. There are two versions to be named: IEEE1394a with up to 400 Mbit/s and IEEE1394b with up to 800 Mbit/s.

Grand Challenge

Competition for autonomous vehicles in 2004 and 2005, which was hosted by the Defense Advanced Research Projects Agency (DARPA).

H1ghlander

Autonomous vehicle of the Red Team, built to participate in the Grand Challenge 2005.

High Tech Automotive Systems

The Dutch Automotive Innovation Programme

Levenberg-Marquardt

Numerical optimization procedure to solve non-linear problems according to the LS-method, named after Kenneth Levenberg and Donald Marquardt.

Red Team

Team founded for the participation on the Grand Challenge, comprising members of Carnegie Mellon University, Caterpillar, General Motors and others.

Sandstorm

Autonomous vehicle of the Red Team, built to participate in the Grand Challenge 2005.

Stanford Racing Team

Team founded for the participation on the Grand Challenge, comprising members of Stanford University, Volkswagen Electronics Research Laboratory (ERL) and others.

Stanley

Autonomous vehicle of the Stanford Racing Team, built to participate in the Grand Challenge 2005.

Urban Challenge

Competition for autonomous vehicles in 2007, which was hosted by the DARPA.

Velodyne HD-LiDAR

high definition LiDAR produced by Velodyne

1 Introduction

The results of this thesis might be described by Albert Einstein's saying: „Nothing happens until something moves". Although Einstein referred to his research in particle physics, his words apply to the present thesis as well, which was aimed at the development of a navigation approach for ground robot mobility in outdoor scenarios. It has been proposed that the motion of an object is influenced by the motion of adjacent objects, but, reciprocally, the motion of an object itself influences the motion of others as well. The objective of the present thesis was to investigate how this behavior can be incorporated into a navigation approach in robotics and how it can become its guiding principle.

In this thesis, a navigation approach reflecting the principles of human reasoning and circumnavigation behavior is presented. Instead of operating in global environment maps, in which obstacles are represented by means of some kind of occupancy value, detected real world objects with a detailed description of their path of motion and behavior relative to one another are the basis for the reasoning and planning process. It is shown how a local ego-centered object-related representation helps to simplify interaction modelling. Furthermore, the incorporation of the interaction models into the prediction and action selection procedure is described. Finally, a control strategy to robustly steer an autonomous robot based on the object-related environment representation is introduced.

1.1 Motivation

State-of-the-art navigation approaches in mobile robotics create and update maps of the local environment, see e.g. Smith et al. [1990]. A trajectory planner (see e.g. Fassbender et al. [2014]) then generates a local, collision free route in these maps in order to satisfy global navigation constraints. These constraints typically contain a global navigation goal, as well as restrictions with respect to the robot motion to increase driving comfort and safety.

During map creation and update, objects from local sensor data are transformed in a global reference coordinate frame. In this process the uncertainties of the objects are mixed up with the position uncertainty of the robot in the global map. An uncertain robot pose would thus lead to highly occupied global maps, although the real environment is mainly unoccupied. The approach that will be presented in this thesis aims to directly convey object measurements to the reasoning and control procedures in order to remove the dependency between the position uncertainty of object measurements and the uncertainty of the robot position in a global map.

Input to the mapping procedure is the whole set of measurements, gained from the various sensors of the robot. I.e. measurements that do not relate to the actual navigation task, e.g. bushes that are separated from the robot by a metal rail, are fed into the mapping procedure as well. As a result the computational effort for data association and mapping increases, although the mapped information is not relevant for the task at hand. The approach of this thesis aims at the separation of relevant and irrelevant objects by evaluating the importance of an object with respect to the current navigation task. This evaluation is achieved by linking the motions of the objects in the robot environment.

While state-of-the-art approaches to robot navigation separate the motion of the entities of the environment, the approach of this thesis explicitly links them. Thus, interaction models that describe the relative motion of one object with respect to another are created from observations. By evaluating these interaction models not only a means to separate relevant and irrelevant data is created but also a way to robustly predict the future motion of an object. Additionally, the interaction models create a link between perception, planning and plan execution, in order to directly convey perceived information to the planning and control mechanisms of the robot.

Finally, the set of interaction models creates a means to ease human machine interaction. Paths of the sort $(x_1, y_1, \ldots, x_n, y_n)$ are typical outputs of path planners as in e.g. Fassbender et al. [2014]. Their numerical representation cannot be understood by humans and their graphical representation changes constantly, although the general task remains unchanged. However, by allowing for plans based on interactions such as *"Follow lane X, then turn right on crossroad Y, ..."*, a means to convey a human readable plan to the robot as well as a way for the robot to communicate with a human observer is created.

1.2 Structure of the thesis

To provide an overview of the current state of research, chapters 2 and 3 are dedicated to the main planning paradigms as well as state-of-the-art methods in outdoor robot navigation.

The chapter 4 then introduces the navigation approach by first outlining the differences between robot and human navigation principles. In this context, the immanent problem of mapping algorithms is shown.

The chapter continues with a description of the object perception mechanism and the manner in which objects are represented. It introduces an environment representation that matches the requirements of the local sensory input. Based on this representation the creation of interaction models is shown that are subsequently used to predict the motion of other objects as well as to steer the robot. To this end, the reasoning and control algorithms that make use of the interaction models are described in detail. A first implementation of the approach was already described in Mueller and Wuensche [2012], which builds the foundation of this thesis. Notably, all papers I contributed to carry my birth name Mueller.

Finally, chapter 5 summarizes the results of this thesis and gives an outlook into possible future research directions. One of these possibilities is already outlined in Deutsche Forschungs-Gemeinschaft (DFG) proposal Wuensche [2013]. While this thesis focuses on the navigation in relation to perceived object geometries, DFG proposal Wuensche [2013] extends this idea in order to incorporate salient object features into the reasoning and control procedures of the robot.

2 Robot navigation - the three main paradigms

The basic steps of navigation (Latin: "navis" means "ship" and "agere" means "to move") were already defined as early as 1840 by [Raper, 1840] and included:

- Determine the current position of the ship with respect to one or more landmarks on a chart.
- Determine the position of the goal and of potential hindrances on the chart.
- Determine an obstacle free route and set sail to move the ship according to the planned course. Observe the progress continuously or at least frequently.

The existing techniques from nautical science were adapted to fit the requirements for navigation in the three dimensional space, but the basic methodology remained unchanged. As described by Levitt and Lawton [1990] or shown by Thrun et al. [2000], the questions: "Where am I?", "Where are other places in relation to me?" and "How can I get there?", still arise. To answer these questions in the robotic context, it is necessary to relate uncertain landmark measurements [Smith et al., 1990] to one another. The aim is to create a static environment map, in which the robot can locate its own position based on its current perception. Furthermore, the robot has to compute and follow a path between a start and a goal position that circumvents obstacles based on this map.

The following sections provide an overview of the three main paradigms in robot navigation. Their aim is to introduce the principles of each paradigm alongside its advantages, disadvantages and preferred area of application.

2.1 Approaches to reactive robot navigation

Almost everyone can recall a situation, in which they tried to sit at a table and banged their knee against the table leg in the process. Aside from some crying

or cursing to canalize the pain, the overstimulated nerves probably also induced a backup action away from the table leg. That is a reaction.

Without a global world model and just by means of the current perception (*e.g. the intense stimulus of the nerves in the former example*) an almost instantaneous action can be elicited, whereby the subject tries to enter into a safer state (*e.g. somewhere away from the table leg*) without thinking about past conditions or future consequences. In robotics, the paradigm of reactive navigation strongly depends on current perception. Control commands are directly derived from possibly preprocessed sensor data to allow for quick responses paired with low computational demands. Figure 2.1 provides a simplified view of the interconnection between perception (*SENSE*) and action (*ACT*) in the reactive paradigm.

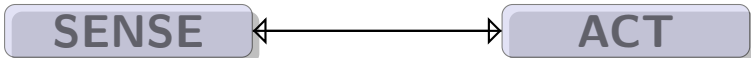
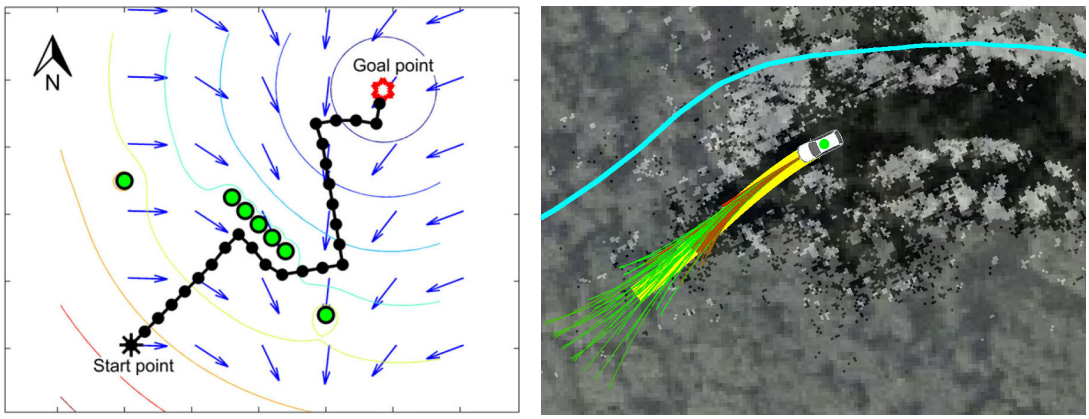


Figure 2.1: The schematic representation of the reactive paradigm.

The efficiency of reactive navigation is achieved by creating actions, which process a certain facet of the input data only and combine them to enable complex robot behavior. Each action can be developed independently and has no influence on the outcome of other actions. This was one of the reasons for the paradigm's success in the 1980's. The parallel execution of multiple behaviors is another advantage that allows adding or removing actions during execution. The most prominent examples are potential field methods. They were introduced by Andrews [1983] and applied to mobile robots by Arkin [1987]. The number of approaches that utilize potential field methods is vast. The work by Pêtrès et al. [2012], for example, showed how boats are able to navigate autonomously in an obstacle rich environment while taking the wind direction and goal position into consideration (see figure 2.2a).

Potentials for upwind, downwind (*blue arrows*), obstacle (*green dots*) and goal position (*red*) were set according to the current obstacle constellation and wind condition. A similar approach was pursued by Yoerger et al. [1998] that enabled the autonomous benthic explorer (ABE) to avoid obstacles during autonomous submerged states. To generate a steering force, a synthetic elliptical force field was applied to the map of distance measurements. An approach to autonomous ground robot mobility that applied reactive navigation was presented by Shimoda et al. [2005]. Its foundation was a potential field over a trajectory space that mapped to actuator commands (steering and throttle, break) directly.



(a) **(b)**

Figure 2.2:
Two examples of reactive navigation approaches. Figure 2.2a provides some insights about how boats can be enabled to steer reactively [Pêtrès et al., 2012], while figure 2.2b shows how the "tentacle"-approach of von Hundelshausen et al. [2008] finds its way through an obstacle rich environment.

Although potential field methods offer an easy way to implement robot behavior, their inherent limitations [Koren and Borenstein, 1991] led researchers to pursue other approaches, including the subsumption architecture [Brooks, 1986]. Instead of summing up behavior outputs, the most prioritized behavior is utilized for robot control. Behaviors are arranged in vertical layers, with each layer having access to the current perception and the output of lower layers. Behaviors that are situated in higher levels subsume the capabilities of lower level behaviors. As a result more complex tasks can be accomplished. Two principles, i.e. suppression and inhibition, are applied, which either replace the input or block the output of lower level behaviors. The subsumption architecture gained popularity after it was described that small insect-like robots were able to avoid obstacles, climb stairs or follow corridors with on-board hardware only.

To yield more agile behavior that addresses the dynamic constraints of the robot directly, control theory principles also found their way into robot navigation. In the approach by von Hundelshausen et al. [2008], the geometric equivalents of feasible control commands are projected into the preprocessed sensor data of the robot (see figure 2.2b). Similar to the mechanical realization of Nilsson [1984], these "tentacles" enabled the robot MuCAR-3 to drive autonomously in unknown, rough and hilly terrain with narrow passages and steep slopes.

With the introduction of fuzzy sets [Zadeh, 1965], a basis for the incorporation of the uncertainty of sensor measurements into robot navigation was developed. Numerous

publications deal with fuzzy logic, for example the work by Dickmanns [2007], which describes how a vehicle was able to drive autonomously from Munich to Denmark in public traffic with speeds of up to 180 kph. Fuzzy set theory was applied to various facets of robot navigation, i.e. to analyze drivable terrain [Seraji, 1999] or to build and localize the robot in environment maps [Saffiotti, 1997].

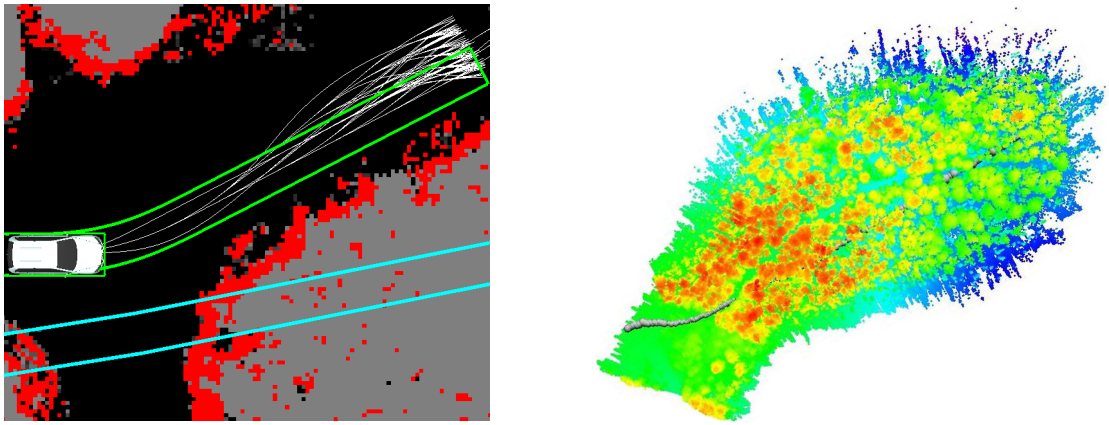
These approaches showed how the potential of reactive mechanisms can be used with respect to mobile robot navigation. Without the need of prior knowledge of the environment, quick responses are guaranteed. However, the publication of Koren and Borenstein [1991] had already revealed some limitations with regard to potential fields, which were applicable to reactive mechanisms as well. Potential fields are able to mislead the robot into a dead end. Forces of closely spaced obstacles are able to repel the robot in a way that prevents it from passing and the presence of obstacles causes oscillations in any case. Additionally, the publication of Arkin [1987] describes that aside from the common potential field disadvantages, their system was also unable to navigate, when strict real-time deadlines were exceeded or when the robot velocity dropped below a certain level. Just like potential fields, reactive behaviors may lead to poor decisions, since neither past states nor future consequences are evaluated. Furthermore, reactive methods often tend to overshoot (comparable to the oscillating behavior of potential fields), which limits their robustness and thus favours their application to low velocity robots. While the above paragraphs propagate the simplicity of creating reactive robot systems, increasing functionality boosts their complexity as well. Eventually such systems are no longer straightforward, though their low complexity was one of the original advantages.

2.2 The deliberative paradigm

Instead of creating a motion force directly from perception, the deliberative paradigm relies on the consideration of prior knowledge and the estimation of future events to create a long lasting plan. Figure 2.3 provides a schematic representation of this process.

Figure 2.3: The schematic representation of the deliberative paradigm.

In the robotic domain to act according to the deliberate paradigm usually means to search for a feasible trajectory from a start to a goal position in a certain configuration space. The configuration space is typically modelled with regard to the degrees of freedom (DOF) of the robot and provides information about the valid spatial configurations the robot is able to use. Two examples are shown in figure 2.4: Figure 2.4a illustrates how, according to Fassbender et al. [2014], a feasible trajectory in the space made up of the special euclidean group of dimension 2 (SE(2)) is planned. The SE(2) represents a configuration (x, y, Ψ) by means of a 2-dimensional Cartesian coordinate (x, y) and a rotation / yaw-angle Ψ. A 3-dimensional configuration space (x, y, z) is presented in figure 2.4b. A tunnel is created through the 3D space by the concatenation of spheres of free space (*grey*).



(a) **(b)**

Figure 2.4:
Figure 2.4a shows how obstacle information provided by the LiDAR data of MuCAR-3 can be used to plan collision free trajectories (see [Fassbender et al., 2014]). Figure 2.4b presents a 3D-path (*grey*) in the environment representation of an aerial vehicle [Vandapel et al., 2005].

Methods of deliberative navigation primarily differ in their manner of representing the environment. Methods such as rapidly exploring dense trees (RDT) [LaValle and Kuffner, 2001], Probabilistic Roadmaps [Kavraki et al., 1996] or Visibility Graphs [Nissoux et al., 1999] are used to explore and represent the free space, while taking robot constraints into consideration. The nodes in these representations embody obstacle free configurations that are utilized as input for graph-search algorithms, for example the A*-algorithm [Hart et al., 1968], which computes a path from start to goal configuration under consideration of optimality constraints.

The Stanford cart Moravec [1980] as well as Shakey [Nilsson, 1984] are prominent representatives of the application of the deliberative paradigm. While the Stanford cart planned a trajectory in a map of circles that represented possible obstacles,

Shakey was able to reason about its actions by evaluating axioms that described the current robot state. I.e. the robot solved the problem of how to get from one room to another by evaluating the consequences of its actions. A world model, comprising axiomatic descriptions of doors, faces, objects, etc. was input to the stanford research institute problem solver (STRIPS). Vision-based localization through visual landmarks allowed Shakey to form the preconditions of actions. I.e. instead of planning and optimizing a trajectory in a Cartesian space, Shakey reasoned about a set of actions such as: "go through door", "block door" or "unblock door" with a logic-based problem solver at its core.

Mobile robot navigation, however, has to cope with uncertainties, for example due to the measurement noise in sensor data or due to imprecise moving object predictions. To compensate for these uncertainties, principles from graph theory are adapted to be used in information spaces as well. The probabilistic version of the Dijkstra shortest path algorithm [Dijkstra, 1959] for instance, assigns probabilities to its edges that represent the possibility of a certain state transition. With policy iteration [Bellman, 1957] the space of plans is iteratively searched rather than utilizing one action space only. By starting with a possibly suboptimal plan, the sequence of actions is refined iteratively until the most optimal plan is derived. Similarly, methods that optimize a high-order parametrized geometric primitive are utilized, which explore the concept of differential flatness [Fliess et al., 1995], or generate a path with cubic spirals [Kanayama and Hartman, 1997]. More sophisticated methods that account for vehicle dynamics as well as environmental restrictions - model predictive control (MPC) [Kwon et al., 1983] is an example - also found their way into robot navigation. They optimize a current time step under consideration of boundary conditions, while future time steps are kept in account concurrently.

When applied to robot navigation problems in any given configuration space, approaches to the hierarchical paradigm create optimal results. Since past, present and future robot and environment states are evaluated, robust motion behavior is achieved. However, optimality comes at a cost as large configuration spaces that represent complex environments typically yearn for large computational resources. In addition, to generate feasible solutions in case of unforeseen events, for example when a person appears right in the way of the robot, deliberative methods are required to obey temporal bounds. Dynamic objects in general require special attention in a complicated procedure [Rohrmuller et al., 2008]. To allow the deduction of possible future object states, uncertainties with respect to the motion progress need to be incorporated into the planning procedure in addition to the estimated motion.

Thus, each deliberative approach needs to define the type of information, that is required to represent the environment properly [Hayes, 1971], in order to keep the planning procedure computationally tractable. This often leads to heavily abstracted models, which are able to represent portions of the real world only. Grid representations [Elfes, 1989], for instance, gained popularity due to their regular structure, their ability to adjust the resolution according to the problem statement and their ability to store data from multiple data sources. The path that is yielded from planning in discrete spaces is optimal according to the discretization only, i.e. the poorer the resolution of the grid, the less optimal the resulting path. But essential to all deliberative approaches is their measure of optimality - aside from runtime, complexity of the environment or completeness of a procedure – because it influences the result to a large extent. Unfortunately, defining an optimality measure is more of an art than science due to its subjectivity and often requires a lot of parameter tuning in a multitude of real world scenarios.

When resorting to deliberative methods only, consistent information of the environment needs to be at hand. I.e. information that holds "no surprises" and follows the closed world assumption has to be available. However, new information is gathered with each sensor measurement during the exploration phase of a robot and this may trigger continuous re-planning steps. Consequently, purely deliberative methods are only utilized in robotics to navigate maps, which either contain complete knowledge of the environment or which need to update small portions on rare occasions only.

2.3 The hybrid deliberative/reactive paradigm

To allow for both, long lasting plans that incorporate prior knowledge as well as quick responses to unforeseen events, the hybrid paradigm combines the strengths of the reactive and the deliberative paradigms. A schematic representation is shown in figure 2.5. It clearly shows how the information from the environment is used for plan generation, but also effects the action generation directly.

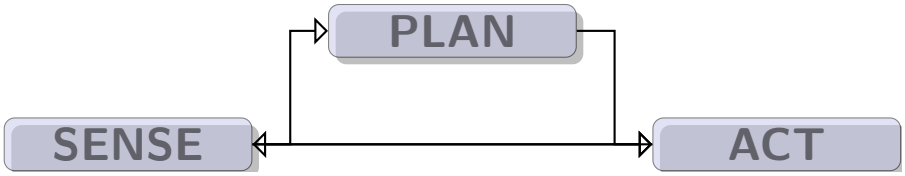
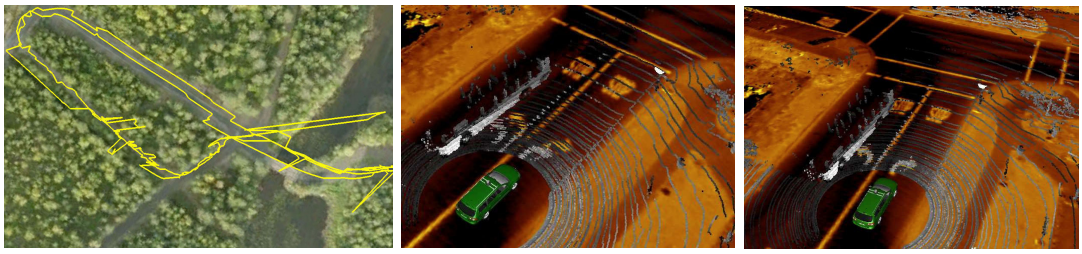


Figure 2.5: The schematic representation of the hybrid paradigm.

Robots that make use of the hybrid navigation paradigm require predefined and updated map information as well as information about the whereabouts of the robot in the map. Dead reckoning, which happens upon integration of robot velocity and steering rate from a reference position, is the most trivial form of localization. Although modern measurement methods, such as laser-ring-gyroscopes, allow for high accuracy, an apparent drift is still present. Therefore, methods for ego-motion estimation were developed, for instance by Smith et al. [1990] or Nister et al. [2004], which incorporate data from camera devices or laser range finders, for example, to cope with the effects of drift motion. While the work by Nister et al. [2004] focuses on the determination of robot motion based on visual features, Smith et al. [1990] formulated the more general problem of creating an environment map from uncertain sensor data with simultaneous robot localization, nowadays known as the SLAM-problem.

Map-based mechanisms became popular that allowed for robot localization in predefined maps or that were able to create new or to update existing maps. Absolute robot localization is carried out by resorting either to deterministic techniques such as triangulation [Sugihara, 1988] or to Montecarlo methods [Dellaert et al., 1999] or to procedures that are based on the Markov assumption [Thrun, 2000, Spangenberg et al., 2016]. Using the Markov assumption, a position distribution on a global map is estimated over time, based on the constellation of perceived landmarks and robot motion.



(a) **(b)** **(c)**

Figure 2.6:
Figure (a) shows the impact of erroneous global positioning system (GPS)-signals on the position estimate of the robot (see [Luettel et al., 2009]). The resulting deviation from the true robot position (b) can be diminished by incorporating local sensor data into the mapping procedure, as shown by Levinson and Thrun [2010].

For instance, Levinson and Thrun [2010] presented how to localize a robot in high-resolution probabilistic infra-red maps. They were able to account for offsets caused by erroneous GPS signals (one example is shown in figure 2.6a) by first creating a high-resolution map of the environment and subsequently evaluating the discrepancy between current sensor data and previously stored map information (see figure 2.6b).

Due to the discrepancies, e.g. between the perceived (*white*) and mapped (*orange*) arrow signs on the street, Levinson and Thrun [2010] were able to correct for the errors caused by GPS and drift motion (see 2.6c for the corrected position estimate, where the vehicle position is slightly more to the right). In [Morris et al., 2005] or [Mueller et al., 2011], the environment was explored by evaluating the topological relation between detected crossroads, while in Tomatis et al. [2003] an approach to localize a robot based on the topological map of an indoor environment was presented.

Robot navigation is then carried out by planning an obstacle free path between current robot position and goal location. The navigation progress is frequently observed. Typically, path generation is achieved by applying methods from section 2.2 to a generated map that models certain facets of the environment, ranging from pure obstacle driven [Elfes, 1989] to object based representations [Limketkai et al., 2005, Vasudevan et al., 2007, Thomas et al., 2015]. But in contrast to the deliberative paradigm, an established plan is executed until either the goal is reached or an event occurs that hinders its completion. Plan tracking is accomplished by updating the location of the robot, i.e. by combining the methods described above with a reactive mechanism (see section 2.1) that is responsible for path following.

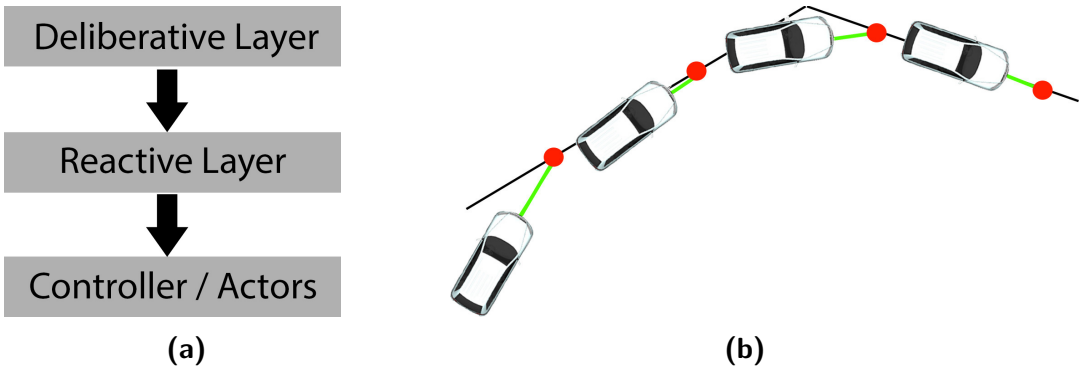


Figure 2.7:
Principles of hybrid navigation approaches. Figure (a) shows the hybrid navigation hierarchy where the deliberative layer outranks the reactive layer. Figure (b) then presents how a plan in the hybrid navigation paradigm is executed. The path is planned by a deliberative approach, but executed reactively, as proposed by Coulter [1992].

One of the most elementary approaches in hybrid robot navigation was presented by Coulter [1992] with the pure pursuit method (see figure 2.7b). The current robot position is projected onto a previously computed path. In a certain look-ahead distance, the goal position is determined and transformed in the robot coordinate frame. In that fashion, the procedure aligns to human driving behavior: during

standard lane follow behavior, human drivers tend to observe a point in a certain look-ahead distance that serves as guidance (see, e.g. [Deml et al., 2008]).

Arkin et al. [1987] presented an architecture with distinct functional layers for planning and plan execution when they introduced their autonomous robot architecture (AuRA). The deliberative portion was responsible for map management, provided functionality for a human machine interface (HMI) as well as path planning and behavior generation capabilities. Based on defined goal information, subtasks, like: *"follow floor, enter room X"* were generated. For each subtask a set of behaviors was created. In accordance with the nature of a behavior a potential field approach was executed. At the very core, an illustration as shown in figure 2.7a remains. Essentially all approaches that implement the hybrid paradigm, including automated underwater vehicle (AUV)'s [Yoerger et al., 2007], ground robots [Fassbender et al., 2014] and unmanned aerial vehicle (UAV)'s [Pettersson and Doherty, 2004], resort to this methodology.

Most robot platforms that operate in high dynamic environments implement some sort of hybrid navigation strategy, since fast reactions are combined with anticipatory behavior. In order to create admissible plans, however, a highly accurate localization is mandatory. The application of SLAM-algorithms did indeed prove to be of value in practical implementations [Thrun et al., 2000], but slight theoretical contradictions were demonstrated [Julier and Uhlmann, 2001]. Furthermore, SLAM-algorithms commonly create uncertain map material based on a static world assumption. Dynamic objects thus require special attendance during the mapping process and later during possible planning procedures [Rohrmuller et al., 2008]. In practice, global metric trajectories are the sole source of information that describes how to navigate in a certain environment. Dynamic objects that are treated incorrectly or imprecise maps, which result from inaccurate localization procedures, may influence the creation of these trajectories. In such a case, permanent re-planning is the worst case scenario.

3 Outdoor robot navigation

Although great advances have been achieved since the early attempts to artificial intelligence (AI) in mobile robotics [Walter, 1950, Braitenberg, 1984, Nilsson, 1984], real world applications still pose a large obstacle to all the facets of robotic systems. Mobile robots need to cope with the complexity, dynamics and uncertainties of the real world, while they are concomitantly limited by processing power and the demand for action in real-time. A robot has to evaluate error-prone sensor data, to create hypotheses and to make predictions. Limited sensor range and resolution may hinder the perception process additionally. Moreover, perception processes are influenced by drastically changing lighting conditions, blurring effects due to rapid robot motion and distortions caused by changing weather conditions. Unforeseen events, caused by high environment dynamics, can force a robot into a complete re-evaluation of the current environment interpretation, which in turn forces it to revise its initial plan for safe navigation.

In order to obtain a truly cognitive system, i.e. mechanisms that allow perception, reasoning and action in accordance with their environment, the situated agent paradigm emerged. The paradigm's ultimate goal is to create an embodied artificial agent that provides cognitive functions in the real world. Thanks to the pioneering work of Dickmanns [2007], the research at the Carnegie Mellon University (CMU) navigation laboratory (Navlab) [Stentz, 1990] and the suggestion of [Mackworth, 1992] to establish a league of soccer-playing robots, the situated agent approach became one of the main driving forces towards cognitive systems.

In the work of Dickmanns [2007] for instance, the situated agent was embodied by a vehicle that was able to drive autonomously from Munich to Denmark in public traffic with speeds of up to 180 kph in 1994. The car drove autonomously for 95% of the time. The system contained a fuzzy rule-base that mapped model-driven perception to robot actions. In the 1980's Dickmanns pioneered autonomous driving with his vehicles, which were capable of driving with high velocities in structured terrain [Dickmanns and Zapp, 1988, Dickmanns, 1988].

To highlight the challenges of outdoor robot navigation (see also e.g. [Mueller and Wuensche, 2013]), major recent events and projects are briefly described in the following sections. In addition, the hardware set-up of MuCAR-3 is presented in order to introduce one possible sensor configuration that allows to tackle the challenges of outdoor robot navigation.

3.1 DARPA Grand Challenge 2004 / 2005

To push innovation with respect to ground vehicle navigation, the DARPA proposed and organized the Grand Challenge. Its goal was to enable autonomous robots to navigate in off-road terrain with as little previous knowledge about the environment as possible. The only information provided by the organizers was a GPS track that consisted of some thousand GPS-way-points. The first competition on an approximately 240 km course through the Mojave Desert was held in 2004. Within 10 hours, 15 contestants out of 107 applicants had to navigate along the given GPS track and avoid obstacles. Although the harsh environment posed a huge obstacle, all participants failed due to a simpler problem: possible errors from maps based on inaccurate GPS data or erroneous GPS position measurements were dismissed and instead every robot drove off the destined track, because it obstinately followed given way-points, which even lead to heavy accidents. Unsurprisingly, none of the participants reached the goal as a result. In fact, no more than 5% of the track was completed.

In a further strive towards the above mentioned goal, the second Grand Challenge [Buehler et al., 2007] was organized in 2005. A track of approximately 212 km through the Mojave Desert had to be completed within a 10 hour time slot. The race specific track was provided 2 hours prior to the race in a DARPA-specified route definition data format (RDDF) format. Files in RDDF format contain a list of longitudes, latitudes, corridor widths that define the outer limits of the course and speed limits associated with each coordinate. The RDDF in the 2005 race comprised 2,935 GPS-way-points with velocities varying between 8 and 80 kph.

In the course of the race different vehicle capabilities such as high-speed road finding and driving, obstacle detection and avoidance were tested. Although the robots had to navigate in an entirely static environment, the terrain conditions still posed a threat to the autonomous vehicles. Basically every terrain type had to be mastered, ranging from paved and dirt roads to rocky mountain passes and rugged

or steep terrain. Furthermore, three underpasses as well as a narrow winding decent through the Beer Bottle pass were included in the course. Although the robots had to maneuver primarily across wide straight dirt roads, difficulties arose, which encompassed dispersed dust that created fake obstacles in the LiDAR sensor data, sun glares that prohibited the correct detection of the road in camera data, and localization errors caused by corrupted GPS-measurements. Further, a large planning horizon was required as the robots were able to drive with high velocities. I.e. distant obstacles and the associated uncertainties had to be incorporated into the planning process and a long-term plan had to be established. Among other things, this plan then had to incorporate terrain characteristics (flat, bumpy, etc.), maximum lateral forces, and the distance from the obstacle into the determination of an optimal trajectory.

From 195 registered teams, only 23 were selected to participate in the final race. Having addressed the problems from 2004, 5 teams were able to complete the competition, but only 4 teams in the given time frame. "Stanley" from the Stanford Racing Team (see [Thrun et al., 2006]) was declared to be the winner with a travel time of 6 h 53 min 58 s. On second and third place followed "Sandstorm" and "H1ghlander" of the Red Team from CMU (see [Urmson et al., 2006]).

Given the GPS-way-points, the approach of [Thrun et al., 2006] created a smoothed global trajectory, to allow for high velocities and low lateral accelerations while concomitantly keeping the prescribed road corridor. A local path planner evaluated driving primitives called "nudges" and "swerves" that allowed for lateral offset changes in case of obstacles in the driving corridor. Similar to [Thrun et al., 2006], the approach of [Urmson et al., 2006] features a global path planning routine that produces a spline interpolation of the given way-points. Additionally, human editors, who were reviewing the resulting path, ensured the safety of the path by e.g. widening possible turns. Local path planning was based on the global path by performing an A*-search of configurations on line segments that lay normal to the preplanned track. By following the local path, obstacles were thus avoided while in parallel the route was kept in relation to the global path. The most astounding result, however, was delivered by Team Gray with their robot Kat-5 [Trepagnier et al., 2006], finishing the race as forth place only 16 min after "H1ghlander" of the Red Team. It was equipped with a cubic spline planner that kept the vehicle within the route boundaries and enabled the robot to evade obstacles. At the end of the race it was found that a bug hindered the vehicle to reach a higher rank. It caused the vehicle to drive slowly in wider segments of the race, where other participants drove at high speed.

3.2 DARPA Urban Challenge 2007

The Grand Challenge 2005 had proven that robots were able to move robustly in static environments over large distances with relatively high velocities. Therefore DARPA officials proposed a new scenario to advance autonomous ground robot research. For the Urban Challenge 2007 [Buehler et al., 2009] robots were required to drive long distances in urban environments, to interact with other moving vehicles and to abide by traffic rules, specifically by those from the California Drivers' Handbook. The large number of encounters between robots and other traffic participants was a unique feature of the Urban Challenge. It was carried out on the site of the former George Air Force Base in Victorville, California, which provided a variety of roads, intersections and parking lots to test the robots. Initially 89 teams from around the world applied for the challenge. After a series of tests, only 11 teams received permission to participate in the final event.

In order to attain a start slot in the final event, each robot had to master a number of tests during the National Qualification Event. Three courses were prepared by the officials. The first area A tested the abilities of the robots to safely merge into dense moving traffic. One difficulty in this context was the correct estimation of the motion of other vehicles as well as the gap in between them. In order to choose the best moment to start the merging maneuver, multiple moving object estimates had to be taken into consideration during the reasoning process. Another difficulty arose from the detection of vehicles that were occluded by others and which had to be incorporated into the reasoning process also. The merging maneuver as such had to be safely executed without any excessive delay. In a relatively long course in area B it was tested, whether the participating robots were able to cope with parked cars, construction areas and other static road obstacles. Each robot had to find the optimal route to the goal within a pre-defined road network while safely avoiding obstacles. Lastly the ability of each robot to safely cross four-way intersections in moving traffic was tested in area C. Moreover, the robots' behavior at unexpectedly blocked roads was observed. The test focused on the detection and tracking of objects and road-blockages and the correct incorporation of these obstacles into the reasoning process while concurrently abiding by traffic rules.

A complete road network definition file (RNDF) was provided two days before the final event by DARPA, which included 629 lane way-points. It contained geometric lane information, lane markings, stop signs, parking lots, and velocity limitations (up to 48 kph, but with lower velocity limits in many places). A second file was provided

5 minutes before the event, which contained special checkpoints that served as mission goal points. Each robot was required to visit each checkpoint in the given order. Additionally, each team was provided with an aerial image of the competition area to further enhance the road network, if necessary.

"Boss" [Urmson et al., 2008] of the Tartan Racing Team was declared to be the winner after 4h 10min 20s and after a distance of 90km had been covered. "Boss" was followed by "Junior" of the Stanford Racing Team and "Odin" of Team Victor Tango on second and third place, respectively (see figure 3.1). [1]

(a) **(b)** **(c)**

Figure 3.1:
Three participants of the Urban Challenge 2007, the robot Junior from the Stanford Racing Team [Montemerlo et al., 2008] (a), the robot Boss from Tartan Racing [Urmson et al., 2008] (b) and the robot of Team AnnieWAY [Kammel et al., 2008] (c).

The system of Urmson et al. [2008] comprised a layered software architecture with layers for sensor data processing, sensor fusion, decision making, etc. The highest level of reasoning consisted of a context based behavior system. The behaviors are clustered into groups with some behaviors depending on the output of others. These groups include road driving or intersection handling, for instance. The behaviors for each context are able to access the results of other behaviors that either reside in the same context or in other contexts. The motion planner is triggered via the behavior layer, wherein the input is always a goal position that is determined by the currently active behavior. The motion planner comprises the results from an MPC trajectory generator, which is constrained by counting regulations such as road boundaries, and optimizes its trajectories under consideration of curvature, curvature rate, maximum acceleration and deceleration, and control input latency. In the case of zone navigation, a lattice planner provides an obstacle free path that keeps the robot away from undesired areas, such as curbs or areas where other dynamic objects may appear. The resulting path is then further processed like a road with the aforementioned methodology.

[1] UniBw Munich participated as a member of team AnnieWay, which was one of only two non-American teams to reach the finals (but it was not able to finish the race).

A hierarchical finite state machine was used by the robot Junior [Montemerlo et al., 2008] of the Stanford Racing Team to switch between different behaviors such as driving, locating or turning. With a smoothed global lane network, which was derived from RNDF data as a foundation, a search algorithm provided the best route to the next mission checkpoint. To avoid local obstacles in the path of the robot, two motion planners were implemented, one for on-road obstacle avoidance and one free-form planner for parking lots or large areas without road information. While the on-road planner resembled the approach of [Thrun et al., 2006], the free-form planner implemented a search strategy on a local map of the environment, called hybrid A*. Instead of planning in a discrete space from one grid cell to another, the geometric equivalent of admissible robot control commands are applied to a grid structure to yield continuous transitions from cell to cell and thus a smooth robot motion.

Team AnnieWAY's approach to autonomous navigation [Kammel et al., 2008] is of particular interest to the present work. Our research team contributed methods for robot navigation [von Hundelshausen et al., 2008] that were already tested and had proven to be reliable in the autonomous navigation scenario of the Civilian European Land Robot Trial (C-ELROB) 2007 (see section 3.5). Global mission planning was carried out by first creating a graph based geometrical representation based on the provided RNDF data. Next, a route containing all checkpoints was determined using a shortest path search and taking the current robot position into consideration. The search procedure was applied at the start of the competition as well as in the event of road blockages that forced the vehicle to take detours. Just like the robot Junior, a hierarchical finite state machine was utilized for reasoning about robot maneuvers. The states comprised, among others, capabilities for driving, merging into traffic, intersection handling and zone navigation. On-road navigation was carried out by directly utilizing the smoothed global geometric representation for robot steering. When road information was not available a free-form planner based on an A*-search in a control command space was applied. But to allow for maximum safety, a reactive fall-back layer was implemented according to the approach by von Hundelshausen et al. [2008]. This reactive fall-back layer showed its usefulness in the case of narrow passages or free-form objects that reached into the road, such as rocks and pavement edges.

3.3 “Stadtpilot” - project

To transfer the knowledge, which was gathered by Team CarOLO, the second non-American team in the DARPA Urban Challenge 2007 (see section 3.2), to real traffic at Braunschweig's inner ring road, the “Stadtpilot”-project [Saust et al., 2011] was devised at the Technical University of Braunschweig. Using the robot car “Leonie”, the project was aimed at fully autonomous driving on public roads in conjunction with the regular traffic flow and in accordance with current traffic regulations. In contrast to the artificial, strongly simplified situations at the Urban Challenge, “Leonie” had to cope with a more chaotic and highly dynamic environment at higher speeds. The densely populated area of Braunschweig's inner ring, for example, prevented the use of GPS as sole positioning system. Large buildings or trees shielded the receivers on the car from GPS-signals or deflected them and thus hampered correct position estimation. Aside from the denser population, roads and intersections were larger in comparison to the Urban Challenge. Multi-lane roads and intersections that connected these roads had to be mastered.

The entire environment of the “Stadtpilot”-project was more complex compared to the one in the Urban Challenge: the environment in the Urban Challenge had been fully controlled by DARPA and stunt drivers had performed a rather defensive driving style. In the “Stadtpilot”-project, however, the drivers steered their vehicles out of their own volition and did not necessarily obey all the traffic rules in detail. Overall, “Leonie” was required to interact with other traffic participants, while behaving according to traffic regulations and the current traffic flow in order to not harm any other road users. Due to the dense traffic at the two-lane ring road, the distances in between other vehicles were very short, which required highly precise maneuver execution. The behavior of “Leonie” at intersections and during merging maneuvers into moving traffic had to adapt to the traffic flow. Resultantly, the robot had to deal with strong velocity changes of other road users, while performing safe and comfortable acceleration and deceleration maneuvers itself. Every maneuver executed by “Leonie” had to abide by traffic rules, which needed to be modelled into the behavior system. They had to enable the vehicle to not obstruct the traffic flow or irritate other traffic participants in order to prevent hazardous situations, which might arise from the reactions of the vehicle.

3.4 Grand Cooperative Driving Challenge

In May of 2011 High Tech Automotive Systems (HTAS) and Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (Netherlands Organisation for Applied Scientific Research) (TNO) organized the first Grand Cooperative Driving Challenge (GCDC) [van Nunen et al., 2012]. The GCDC provided a testing and evaluation ground for cooperatively driving vehicles. It was also meant to facilitate further developments in the area of cooperative driving, and was hoped to significantly alleviate traffic problems and to improve traffic safety. Nine international teams had the opportunity to compete in both, urban and highway scenarios in order to compare and prove the robustness of their longitudinal control systems with regard to criteria such as platoon length, traffic light throughput and string stability. The team that responded most quickly to heavy acceleration and deceleration effects and required the least amount of road space for that purpose was declared to be the winner. After 15 runs (called heaps) with vehicles, which were randomly assigned to two neighboring lanes, Team AnnieWAY [Geiger et al., 2012] won the first GCDC as its performance conformed best to the judging criteria.

In contrast to the DARPA Grand Challenge or Urban Challenge, one major aspect of the GCDC was the transmission of vehicle state information between the participants instead of estimating vehicle states based on sensor data, i.e. camera or LiDAR. Cameras, vehicle-to-infrastructure (V2I), and vehicle-to-vehicle (V2V) connectivity (WiFi hotspots, GPS transmitters) allowed the estimation of position, velocity and acceleration of each vehicle on the test site as well as the communication of these parameters among the participants. Thus, the prediction of the behavior of occluded or distant vehicles became possible. This allowed for upcoming traffic situations, such as possible hazards or incidents to be judged. As a result, robust data transmission and error handling in case of transmission distortions or outages was a core prerequisite. However, according to [Geiger et al., 2012] the sole reliance on communicated states proved to be error-prone, because inaccurate GPS measurements and errors in communicating vehicle states resulted in false vehicle estimates. Therefore information from camera and LiDAR sensors was also incorporated into the estimation process to enhance vehicle state estimates.

The urban scenario was composed of two traffic lights, whose states were communicated via roadside units and which were connected via a two-lane, uni-directional road. At each traffic light the platoon of vehicles waited until the traffic light turned green. When the traffic lights turned green, the participants were required

to cooperatively accelerate in order to maximize the throughput at this traffic light and to maintain a stable platoon. During each run the vehicles from the platoon upstream had to join vehicles from the platoon downstream while maintaining string stability. The urban part of each run came to an end when the last vehicle of either side crossed the finish line.

The highway scenario was carried out on the A-270 highway between the cities of Helmond and Eindhoven. To test the participants' behavior at varying speeds, including the reaction to disturbances such as heavy deceleration and acceleration maneuvers as well as their ability to maintain a stable platoon, the GCDC lead vehicle introduced disturbances to the platoon. Typically even a short braking maneuver introduces a disturbance in a platoon that may cause standstills or even collisions further downstream. A follower vehicle was employed with the aim to dampen these disturbances and thus to maintain a coherent and string stable platoon with an overall length that was minimized.

3.5 European Land Robot Trial

Since 2006 the European Land Robot Trial (ELROB) with both, civilian and military focus, is held once a year to demonstrate and compare the abilities of autonomous robotic systems in realistic scenarios (see e.g. [Himmelsbach et al., 2009, Fries et al., 2017]). A multitude of scenarios is provided, which range from reconnaissance and surveillance tasks over camp security and transportation to autonomous navigation and even explosive ordnance disposal (EOD). Each scenario requires specialized robots, as most robots designed for autonomous navigation, for example, lack the specialized tools that are required to dispose of bombs in EOD scenarios.

In past ELROB competitions it was attempted to imitate real world situations as accurately as possible for each scenario in order to push participating robotic systems to their limits and beyond. In contrast to the DARPA Grand Challenge or Urban Challenge, for example, the organizers did not modify the environment with artificial barricades or curbs to construct some sort of driving corridor. Instead each participant had to cope with the complexity that the real world offers. Furthermore, each participant only received a few GPS points (about 3 to 20) with general driving directions rather than an entire GPS network. These conditions in combination with the rather vague description of what to expect from the scenario demanded a high

level of autonomy as well as a considerable amount of cognitive abilities from the participating robots.

Usually the participants were confronted with a non-urban scenario. Only a minority of scenarios took place in actual houses, on tarred roads or on concrete. Most terrains included vegetation (trees, grass, and bushes), sand, water, stones, difficult access paths and dirt roads. Quite frequently, wooded or hilly terrain with steep hillsides or narrow passages restricted the motion options of a robot, which could even result in a side way roll-over if driven wrongly or lead to a complete standstill. Additionally, large trees and house fronts interfered with or prevented the reception of GPS signals. A rather rough position estimate with errors of up to 50 meters had to suffice in the competition. Since the regulations of the ELROB do not prohibit the use of publicly available environmental data, road network information could additionally be incorporated into the localization process. However, their degree of detail was lacking in comparison to the maps used in the DARPA competitions or in the "Stadtpilot"-project.

However, the correct interpretation of sensor data in completely unknown terrain at day and night was a far greater challenge than these localization issues. Moreover, obstacles were not merely defined by their size above ground. Instead overhanging structures that could be passed by had to be detected and evaluated. A reliable vegetation classification, which is still subject to intense research had to be implemented to allow the detection of flexible obstacles. Since surveillance tasks were also in the range of the problem statements, robust object detection and tracking mechanisms, which were able to cope with a multitude of objects, had to be available to each robotic system. Dead ends, sharp turns or road blockages also belonged to the majority of scenarios in order to test the reasoning capabilities of the participants.

At the onset of a scenario the respective robot was placed somewhere in a marked-off area. After determining its own position via its localization procedures, the robot had to either complete a route that traversed along given target points, or it had to monitor an area, which was defined by a boundary polygon. Distances and dimensions varied with the respective scenarios. During reconnaissance and surveillance routes, for example, about 3 km had to be covered, while autonomous navigation typically required the robots to move along a 7 to 20 km track. Dynamic objects were announced for every trial, but they rarely entered a scene. Time limits of 60 minutes were introduced for each scenario as about 8 scenarios with 5 to 10 participants had to be completed within four days.

Completion of each level qualified the participants for the next higher one. The first level typically included wide dirt roads with almost no obstacles on the road and the highest level contained sharp turns and narrow roads, which were delimited by bushes, trees, overhanging structures or artificial obstacles that had been placed into the path of the robot directly. The mule-scenario, for instance, demanded the detection of a moving object. This "moving object" was usually a person (see e.g. figure 3.2b), who "pulled" the robot (see for instance [Mueller et al., 2009]) along a path that the robot had to memorize in order to successfully shuttle between two locations.

(a) **(b)**

Figure 3.2:
Figure (a) shows our robot MuCAR-3 with roof mounted 360 degree Velodyne HD Laser Scanner, GPS antennas and a pitch and yaw-able camera platform behind the front window. Figure (b) then provides a snapshot of MuCAR-3 in action during an autonomous object following maneuver.

3.6 Our robot: MuCAR-3

Munich Cognitive Autonomous Robot Car 3rd Generation (MuCAR-3), see Figure 3.2a, is a VW Touareg vehicle with full drive-by-wire (DBW) modifications. The third generation vehicle succeeds test vehicle for autonomous mobility and computer vision, *german: Versuchsfahrzeug für autonome Mobilität und Rechnersehen* (VaMoRs) and VaMoRs-Pkw (VaMP), which were devised by Prof. Ernst D. Dickmanns at the University of the Bundeswehr Munich (UniBwM) from the early 1980s to the mid 1990s. The standard equipment of the VW Touareg comprises an electronic accelerator system. During autonomous drives the interface to this system is triggered by the on-board computer to command a desired velocity. Additionally, the active brake booster, which is usually used for the electronic stability control (ESP), adaptive cruise control (ACC) or the electronic brake assist, is actuated by a PWM-signal from the on-board computer. Furthermore, an auxiliary direct current

motor is mounted close to the steering column in order to provide access to the vehicle's steering actuator. Similar changes had to be applied to the automatic gearbox as well as to the foot-operated parking brake. In this manner, basic vehicle controls that are available to a human driver are made available to the on-board computer as well. However, a human safety driver may override them at any time by actuating the brake pedal or the steering wheel or by pushing the emergency stop buttons.

Various sensors provide data to allow for autonomous driving. These comprise the vehicle's built in velocity and steering angle measurement units, as well as the supplementary installed Velodyne HD-LiDAR, camera platform MarVEye-8 or the Oxford Technical Solutions (OxTS) RT3003 inertial navigation system (INS). In addition, an absolute rotary encoder is linked to the steering column. Compared to the production vehicle's encoder, it allows for higher resolution as well as a higher sampling frequency. The measurements of each sensor are then made available by the controller area network (CAN)-bus, which serves as an interface between vehicle and on-board computer.

An Inertial Navigation System, type OxTS RT3003, (see Figure 3.3c) allows for the measurement of inertial pose and referenced positions in the world coordinate frame. This INS comprises an inertial measurement unit (IMU) with three gyros and three accelerometers (build from micro-electro-mechanical system (MEMS)) for 6-DOF measurements, coupled with a high-precision differential global positioning system (DGPS) receiver, which provides position and velocity information. An integrated Kalman filter is utilized for the fusion of measured data. The system is configured as dual antenna model, which makes it possible to determine the vehicle's direction/orientation/heading after power-up without moving it. The position accuracy can be as exact as 0.02 m (using DGPS), the velocity accuracy as exact as 0.05 kph, the acceleration accuracy as exact as 0.01 m/s^2 , the orientation accuracy as exact as 0.03° in pitch/roll axes and 0.1° in yaw(=heading)-axis, and finally the angular rate accuracy as exact as 0.01 °/s.

A state-of-the-art high-resolution 360° LiDAR system (Figure 3.3a), type Velodyne HD-LiDAR (HDL64SE2)[2], examines the robot's environment with its 64 laser diodes. The active depth sensor measures distances according to the time-of-flight (TOF) principle. A short pulse of long-wave (905 nm) laser light, invisible to the human eye, is emitted, then reflected by environmental structures and received at photo diodes. The diodes cover a horizontal resolution between 0.0886° and 0.3544°. Vertical field

[2]For more information see the developers homepage: *http://www.velodynelidar.com*

(a)
Velodyne 360 degree HD Laser Scanner.

(b)
Pitch- and yaw-able camera platform MarvEye-8.

(c)
Inertial Navigation System Oxford RT3003.

Figure 3.3: The sensors of our test platform.

of view is $\approx 2°$ to $\approx -24°$ at a resolution of 0.3° for the upper 32 diodes and 0.5° for the lower 32 diodes. Measurement precision is specified with $\sigma < 0.02\,\text{m}$ at 25 m, i.e. 99.7 % of the measurements of an object in 25 m distance are located in the interval $25\,\text{m} \pm 0.06\,\text{m}$. The maximum distance is about 120 m. At a frequency of rotation of 10 Hz, 130000 distance measurements are accumulated per rotation. The sensor block is mounted on the vehicles roof (see Figure 3.2a) in a manner so as to keep the occluded area around the vehicle as small as possible.

MarVEye-8 (Figure 3.3b), an 8th generation camera platform, is equipped with up to four wide-angle and one inertially stabilized tele-lens camera. The design of the platform is based on the human eye with a wide view angle of approximately 100 degree and a high resolution fovea in the center of the image. This is achieved by combining two wide angle cameras (8 mm lenses) and a tele-camera. To achieve fast saccadic vehicle eye motions, the platform yaw axis is driven by a hollow shaft torque motor. Since changes to the camera configuration influence platform dynamics, the results of an RLS estimator (see [Unterholzner et al., 2012]) are used to adapt the parameters of the control law that governs the saccadic motion. The tele-camera is placed in the hollow shaft of the torque axis with an adjustable mirror reflecting the horizontal view towards the camera. As a result, only the mirror has to be adjusted (between $-12°$ and 12°) to allow for active gaze stabilization.

A multi-processor system with dual hexacore Intel Xeon L5640 central processing unit (CPU), 12 GB working memory and shock resistant SSD-hard drives evaluates sensor data and communicates vehicle commands by means of FireWire, CAN, RS-232 and network interfaces. The CAN-bus provides the connection between the on-board computer and two real-time computer systems of type dSpace Autobox and MicroAutobox. These two real-time computer systems trigger the actuators of the vehicle, such as throttle, brake and MarVEye-8.

At the heart of the software-architecture of MuCAR-3 is a real-time database called KogniMobil Real Time Data Base (KogMo-RTDB), see [Goebl, 2009]. It provides inter-process communication (IPC) functionality, which is specifically adapted to the needs of cognitive autonomous vehicles. KogMo-RTDB acts as middle-ware, which enables independent system processes to securely and asynchronously access the shared memory and object information. Further, the KogMo-RTDB provides functionality to record subsequent snapshots of its database objects and to replay once captured snapshots. This functionality enables repeated development and testing of algorithms on the same dataset and the more rapid tracing of errors during autonomous drives.

4 Object-related navigation

The principles guiding "object-related navigation" are based on human knowledge representation and reasoning that are cognition-dependent. This includes the processing of information gathered by the organs of perception of humans and how this processing is connected to the resulting actions. In 1976 Ulrich Neisser, who introduced the term "cognitive psychology", already stated that perception and action are not mutually exclusive. Neisser [1976] and Gibson [1977] speculated that there must be two perceptual systems in the brain, namely direct perception and recognition.

Gibson [1977] proposed and proved that certain perceivable stimuli in the environment, which he called affordances cause certain actions. For example, if someone throws a ball right at you, in order to not get hit the first action would be to avoid the ball. Affordances are merely potentials provided by the environment, but in order to execute a certain behavior all other conditions must be met as well. The action to avoid the ball may not be carried out, when a ball is thrown and one does not care about getting hit. Affordances provide a means to directly perceive the environment without the need for memory, inference and interpretation. I.e. Gibson's perspective shifts the view from structural to functional models, because objects have a certain function. Cars, for example, can either be overtaken or followed, crossroads afford left and right turns and so forth. Gibson believed that perception and action are interconnected and that it is counter-productive to separate perception of the environment and the behaviors of an agent.

Affordances help to respond quickly to environmental conditions, but they do not help to make distinctions between e.g. "one's own car" from "the neighbor's car". Accordingly, Neisser [1976] and most contemporary cognitive scientists postulated that there must be one or more portions of the brain responsible for problem solving, recognition and other cognitive processes. These parts of the brain were expected to require memory in order to discern objects, inference to anticipate future events and interpretation to evaluate the current environment. According to his theories perception and action are tightly coupled by providing two essential components, a

plan of how to perform an action and a selective attention strategy of how to get more information about the environment to support this action. Hence, perception is defined as a constructive process, in which an agent tries to predict information about the environment to either be ready to accept them or to cope with them when they occur. According to Neisser, these perceptive mechanisms are arranged in cognitive maps (see e.g. Tolman [1948], O'Keefe and Nadel [1978], Eden [1988]) in order to direct certain actions and to form plans, which navigate a human through a certain environment.

4.1 Human principles of navigation

How do humans perceive their environment? According to [Russ, 2006] the maximum theoretical resolution of the human eye, which can be thought to be the most important sense with respect to navigation, is approximately 50 cycles per degree at a distance of 1 m. One cycle is defined as an adjacent pair of black and white lines, i.e the human eye has a maximum theoretical resolution of 100 pixel per degree. Such a high resolution is available in the central fovea region only, which covers about 2° of the visual field. But due to the saccadic motion of the eye this small area can be extended over time (scanning saccade). In order to give an impression of the amount of data that needs to be processed, let us assume a piece of DIN A4 paper (29.7 cm $\times$ 21 cm) is located at a distance of 1 m in front of the human eye. It would thus cover an angular area of 16.89° $\times$ 11.99°. Relating this to the maximal resolution of 0.01° per pixel, we get $(16{,}89/0{,}01 = 1689) \times (11{,}99/0{,}01 = 1199) = 2025111$ pixel. I.e. approximately 2 million pixels need to be processed to recognize a piece of DIN A4 paper at a distance of 1 m.

In order to then process the huge amount of visual information, a small fraction of the human brain (see Grill-Spector et al. [2001]) creates an abstraction by means of classified object instances. But interestingly, no more than nine objects can be processed in parallel without losing attention (see Miller [1956]). Given the number of perceived objects, the human brain continuously updates an egocentric representation of the environment [Wang and Spelke, 2000]. Its cognitive map [Tolman, 1948, Eden, 1988], built from egocentric information only, contains local representations of the environment without any reference to a global world frame. According to Gothard et al. [1996], only a small part of the brain - the Hippocampus - is involved in the spatial sense of human beings.

Due to updates of an ego-centric map, humans rarely ask "Where am I?", when searching for a destination. Instead they ask: "How can I get to ...?" A natural response would be for instance: "Take the next exit on the right side, follow the corridor ..." These instructions do not contain a detailed representation of the whole environment, but describe salient landmarks that are sufficient for the purpose of human navigation. The landmarks are structures (e.g. exit, corridor, ...), which can be discriminated by perception, and they have actions (such as: turn right, follow ...) associated with them. The concatenation of such object-action-pairs leads to the creation of a plan, which is independent of the actual scene context. Even when the scene context is uncovered during the execution of the plan, the action sequence remains unchanged as long as there are no major discrepancies (i.e. the corridor is blocked). While the large scale orientation and navigation of humans appear to be arranged thoughtfully, the local behavior actually adapts to perception, while bearing the global plan in mind.

4.2 Principles of robot navigation - SLAM by example

To properly navigate in any given environment, i.e. to be able to make use of the planning approaches of section 2.3, a map of the environment must be created. Creating valid sensor maps requires the incorporation of the robot's ego-motion, including potential errors, which may happen during the estimation step, and landmark measurements with their respective uncertainties. One exemplary map creation process is shown in figure 4.1, in which obstacles O_1 till O_4 are mapped and robot position (coordinate frames with number 1 to 7) is estimated. In this example the robot might follow a curved road, while passing objects to the left and to the right.

Starting at an initial position in some global coordinate frame (*the coordinate frame numbered "1" in figure 4.1*), the robot state estimate $\boldsymbol{x_k}$ at step k (*coordinate frames numbered "2" to "7"*) is updated given motion model $p(\boldsymbol{x_k} \mid \boldsymbol{x_{k-1}}, \boldsymbol{u_{k-1}})$ (*blue arrows of figure 4.1*) and measurement model $p(\boldsymbol{y_k} \mid \boldsymbol{x_k}, \Xi, \boldsymbol{l_k})$ (*red arrows of figure 4.1*). With $\boldsymbol{x_{k-1}}$ being the state at step $k-1$, $\boldsymbol{u_{k-1}}$ the control input at step $k-1$, $\boldsymbol{y_k}$ the measurement at step k, Ξ the current map and $\boldsymbol{l_k}$ being the landmark association at step k. With each new measurement, the global robot position as well as global landmark positions are estimated in relation to the artificial reference point of the global coordinate frame. In addition, associated error covariances (*blue-gray ellipses*) are updated according to motion and measurement noise. Each landmark

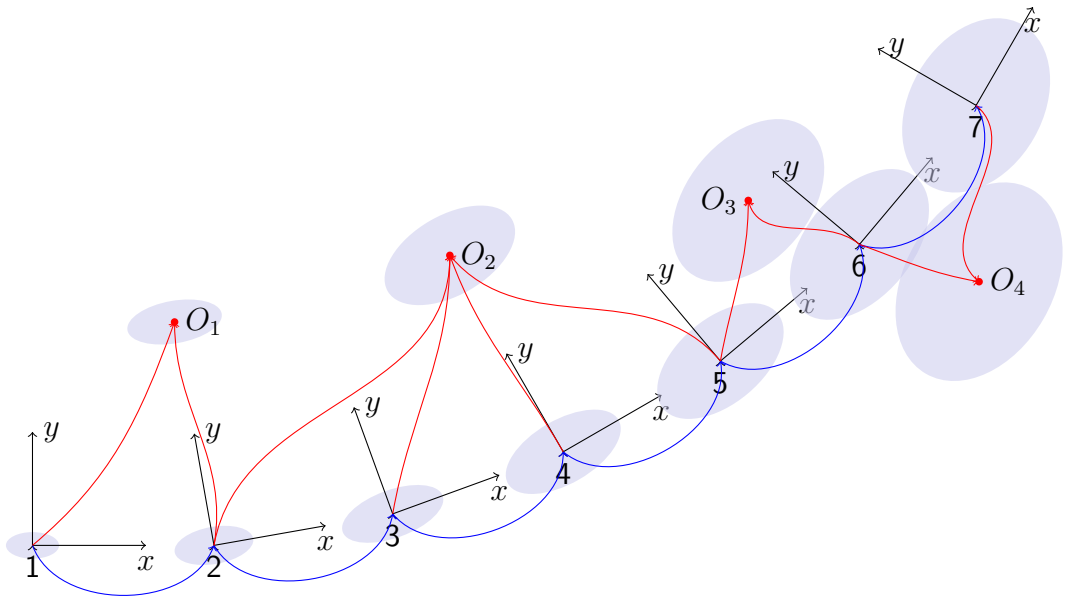


Figure 4.1:
The SLAM-problem. Each coordinate frame represents a robot pose in the SE(2). The uncertainties of the global robot pose as well as the global position uncertainty of each landmark (*red dots*) is represented by ellipses (*blue-grey*). With each motion the global position uncertainty of the robot pose as well as the global position uncertainties of the landmarks increase.

percept needs to be transformed with respect to an already uncertain global robot position, which in turn adds to the landmark uncertainties during the mapping process. I.e. expected robot position $\boldsymbol{x_k}$ and landmark positions Ξ are conditioned on all previous robot motions $\mathbf{U_{k-1}}$, landmark measurements $\mathbf{Y_k}$ and landmark associations $\mathbf{L_k}$. In virtue of Montemerlo et al. [2002] the SLAM posterior at time k is then formulated as Bayes filter as shown in equation 4.1 with η being the normalizing constant and $p(\boldsymbol{x_{k-1}}, \Xi \mid \mathbf{Y_{k-1}}, \mathbf{U_{k-2}}, \mathbf{L_{k-1}})$ the SLAM posterior at time step $k-1$

$$\begin{aligned} p(\boldsymbol{x_k}, \Xi \mid \mathbf{Y_k}, \mathbf{U_{k-1}}, \mathbf{L_k}) = \\ \eta \cdot p(\boldsymbol{y_k} \mid \boldsymbol{x_k}, \Xi, \boldsymbol{l_k}) \\ \cdot \int p(\boldsymbol{x_k} \mid \boldsymbol{x_{k-1}}, \boldsymbol{u_{k-1}}) \cdot p(\boldsymbol{x_{k-1}}, \Xi \mid \mathbf{Y_{k-1}}, \mathbf{U_{k-2}}, \mathbf{L_{k-1}}) d\, \boldsymbol{x_{k-1}} \end{aligned} \tag{4.1}$$

To estimate the SLAM posterior $p(\boldsymbol{x_k}, \Xi \mid \mathbf{Y_k}, \mathbf{U_{k-1}}, \mathbf{L_k})$, some sort of Kalman filter (KF) (e.g. extended Kalman filter (EKF), unscented Kalman filter (UKF), to name the most prominent) or particle filter (PF) are often utilized. Typically, the

posterior is then represented by a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{s} \mid \mu, \boldsymbol{\Sigma})$, with system state $\boldsymbol{s_k} = \{\boldsymbol{x_k}, \xi_{1k} \dots \xi_{Nk}\}$ at time step k, which is defined by the location of the robot $\boldsymbol{x_k}$ and the landmarks ξ_{1k} to ξ_{Nk}.

As equation 4.1 constitutes, current robot location as well as the location of every single landmark $p(\boldsymbol{x_k}, \Xi \mid \mathbf{Y_k}, \mathbf{U_{k-1}}, \mathbf{L_k})$ is dependent on the observation probability of each visible landmark $p(\boldsymbol{y_k} \mid \boldsymbol{x_k}, \Xi, \boldsymbol{l_k})$ as well as on the probability of residing in state $\boldsymbol{x_k}$ and observing map Ξ while all previous measurements and motions are given (when the integral part of 4.1 is dissolved, $p(\boldsymbol{x_k}, \Xi \mid \mathbf{Y_{k-1}}, \mathbf{U_{k-1}}, \mathbf{L_{k-1}})$ is retrieved). I.e. mistaken landmark associations or noisy sensor data from both, landmark measurements and ego-motion measurements impact the global position estimate of the robot.

Now lets assume, that the measurement of landmark O_2 in figure 4.1 is corrupted, by either sensor noise, wrong landmark associations, changes in shape, or by the landmark being itself a moving object (although dynamic objects are removed before or during the mapping process, small object motions are still able to remain undetected). To show the impact on the map as well as on the robot position, the definition of conditional probability (equation 4.2) is used to expand the SLAM posterior of equation 4.1, as shown in equation 4.3.

$$p(A, B) = p(A \mid B) \cdot p(B) \tag{4.2}$$

$$\begin{aligned} p(\boldsymbol{x_k}, \Xi \mid \mathbf{Y_k}, \mathbf{U_{k-1}}, \mathbf{L_k}) &= p(\Xi \mid \boldsymbol{x_k}, \mathbf{Y_k}, \mathbf{U_{k-1}}, \mathbf{L_k}) \\ &\quad \cdot p(\boldsymbol{x_k} \mid \mathbf{Y_k}, \mathbf{U_{k-1}}, \mathbf{L_k}) \end{aligned} \tag{4.3}$$

The SLAM posterior (left hand side of the equal sign) of equation 4.1 can be substituted by equation 4.3 to yield:

$$\begin{aligned} p(\Xi \mid \boldsymbol{x_k}, \mathbf{Y_k}, \mathbf{U_{k-1}}, \mathbf{L_k}) \cdot p(\boldsymbol{x_k} \mid \mathbf{Y_k}, \mathbf{U_{k-1}}, \mathbf{L_k}) = \\ \eta \cdot p(\boldsymbol{y_k} \mid \boldsymbol{x_k}, \Xi, \boldsymbol{l_k}) \\ \cdot \int p(\boldsymbol{x_k} \mid \boldsymbol{x_{k-1}}, \boldsymbol{u_{k-1}}) \cdot p(\boldsymbol{x_{k-1}}, \Xi \mid \mathbf{Y_{k-1}}, \mathbf{U_{k-2}}, \mathbf{L_{k-1}}) d\, \boldsymbol{x_{k-1}} \end{aligned} \tag{4.4}$$

If landmark O_2 drifts due to one of the aforementioned corruptions, measurement model $p(\boldsymbol{y_k} \mid \boldsymbol{x_k}, \Xi, \boldsymbol{l_k})$ of equation 4.4 may now favour a robot position $p(\boldsymbol{x_k} \mid$

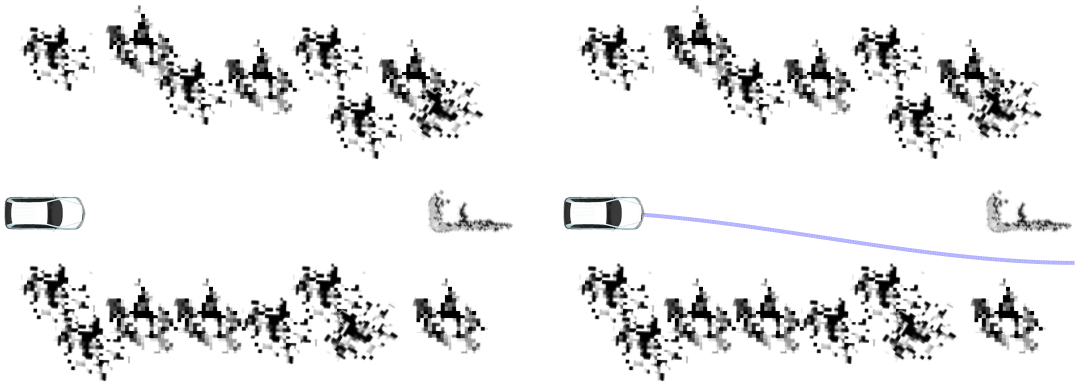
$\mathbf{Y_k}, \mathbf{U_{k-1}}, \mathbf{L_k})$ that is slightly shifted in accordance with the displacement of the landmark. From the left part of equation 4.4 it can further be derived that the shift in position then influences the update of each landmark position $p(\Xi \mid \boldsymbol{x_k}, \mathbf{Y_k}, \mathbf{U_{k-1}}, \mathbf{L_k})$ instead of influencing the position of the displaced one only. I.e. the uncertainty of the entire map increases because one landmark moved rather than being static. Moreover, as a result of the recursive design of the estimation process as described by equation 4.1, the estimate of time step $k-1$ impacts the estimate of time step k, which in turn propagates errors that were made in early estimation steps through the entire estimation process over time. Hence, huge position uncertainties may be associated to obstacles O_3 and O_4 and robot positions 2 to 7, which in turn might render subsequent planning steps unable to pass through obstacles O_3 and O_4.

Julier and Uhlmann [2001] already showed the negative impact of measurement noise on the global position estimation of a robot in 2001. Highly dynamic environments or domains that contain numerous distinct moving objects pose conceptual challenges. Foremost among these challenges is the fact that environment maps are static by definition. Therefore dynamic objects are a misfit requiring special attention, because they violate the static obstacle assumption, see Wang et al. [2003], for example. Consequently environment maps may converge very slowly to a true static environment representation, if it is feasible at all.

Figures 4.2 and 4.3 show a comparison of both, the principles of robot navigation as well as human principles of navigation, in an everyday traffic scenario. The scenario is comprised of the robot, oncoming traffic, a bend between the robot and the oncoming vehicle and obstacles (trees) to the left and to the right of the street.

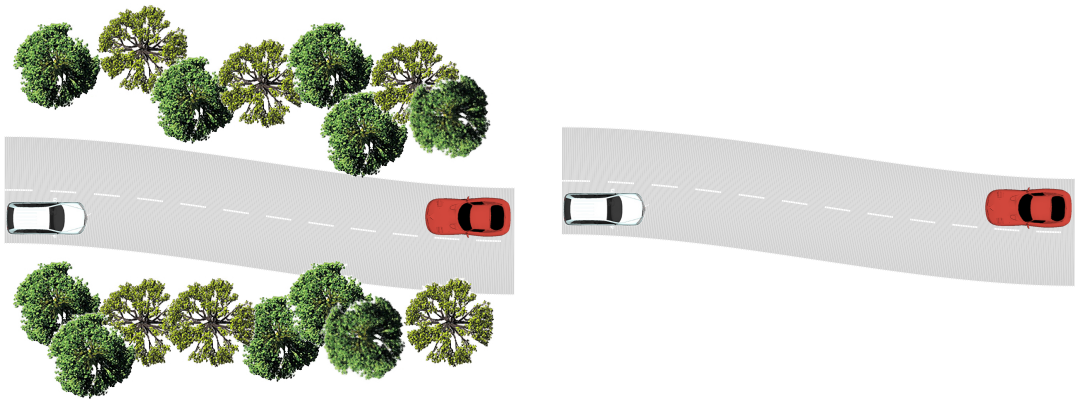
For state-of-the-art robot navigation approaches, an obstacle map of the real environment needs to be created (see figure 4.2a). The moving obstacle in front of the robot (*white vehicle*) has to be distinguished from the set of static obstacles. Furthermore, a path (figure 4.2b *blue* polyline) has to be planned, based on the current snapshot of the environment and on the current motion of the moving obstacle. Thus, the entire environment, including objects with no impact on the final plan such as the trees, is incorporated into the reasoning process in order to merely drive along a lane.

In contrast, human reasoning gathers information on the relevant objects only (e.g. as depicted in figures 4.3a and 4.3b), instead of incorporating every piece of information about the environment into the reasoning process. Static obstacles are



(a) **(b)**

Figure 4.2:
A traffic scenario represented by a grid based approach (a), which is then used to generate a trajectory (*blue line sequence* in (b)) so that the robot (*white vehicle*) is able to drive safely.



(a) **(b)**

Figure 4.3:
The traffic scenario of figure 4.2 represented by means of objects and their properties. The object-based representation exploits the properties of the objects to narrow down the set of required objects (red car and road in (b)) to get to a conclusion about the further motion.

not distinguished from dynamic objects, since the required action, e.g. avoidance, might be the same. When an object does not move according to the expectation (the oncoming car in figure 4.3a for instance), the behavior is only adapted with respect to that object and not with respect to the entire scene. Finally, prior knowledge can be brought into the reasoning process. Prior knowledge in the example of illustration 4.3, for instance, may lead to the conclusion that the two vehicles are not required to interact, if they follow their respective lanes. As a result, dynamic objects, alongside

the trees to the left and to the right, might be cancelled out of the considerations regarding future robot motions.

4.3 Problem statement

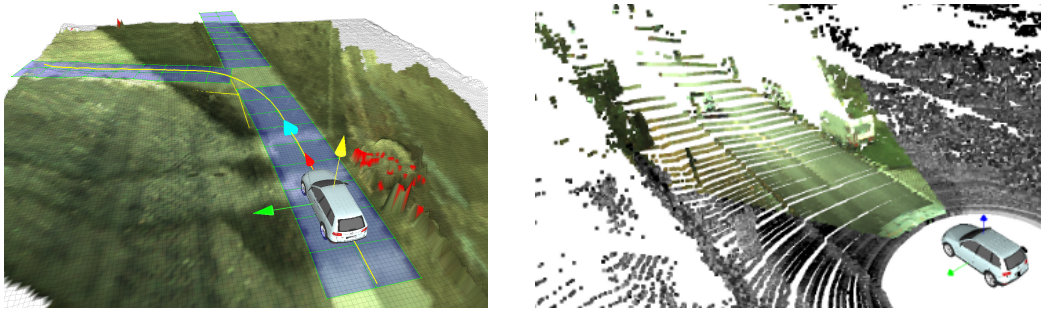
Based on the above considerations, the question arises whether simple object-based principles of human navigation can be transferred to navigation approaches in robotics and whether a simplification of the whole navigation process can be accomplished. This question is at the core of an approach termed "object-related navigation", which is aimed at shifting the view of modern approaches to navigation from a global, metric, trajectory-based representation to a more human-like, local, object-based one.

In contrast to global, metric navigation, the approaches presented in this thesis incorporate relative spatial information, which is gained by exploiting the local sensor data of the robot, directly into the planning and control routines. To allow for quick reactions to unforeseen events, a spatial ego-centric representation has to be created that enables a tight coupling between perception and action on the one hand and the generation of robust robot control commands on the other hand. Based on the ego-centric representation as well as its inherent geometric transformations a means to model and estimate object motion is required, which exploits the relative instead of the global motion between objects. By exploiting the relative motion of objects to each other, new motion primitives, like: *"follow"*, *"overtake"* or *"turn right"* become available. A method of how these relative motion primitives are utilized in the steering command generation process must be created to allow for robust robot control as well as quick reactions to unforeseen events. Finally, a way of how to generate and execute long term plans in the manner of human comprehension such as *"Follow lane X, then turn right on crossroad Y, ..."* must be implemented.

At first, the internal representation of objects like crossroads, lanes or vehicles is described, to provide a general understanding of the perception system. After the introduction of the ego-centric environment representation it is clarified how these object representations are transformed in order to fit the needs of the motion prediction and reasoning methodologies of this thesis.

4.4 Object perception mechanisms

The navigation methods proposed in this thesis rely on data provided by the object detection and tracking routines of our robot. For more information, refer to [Himmelsbach and Wuensche, 2012] and [Manz, 2013]. Figure 4.4 shows a view of the environment and objects, which were detected via the different perception mechanisms, from the perspective of the MuCAR-3 sensors.



(a) Fused environment representation of MuCAR-3, including a detected crossroad *(blue structures)* (see Manz [2013])

(b) LiDAR point cloud, colored with information of the camera system (see Himmelsbach et al. [2011])

Figure 4.4:
Different environment representations, given sensor information of LiDAR and camera.

The perception system of MuCAR-3 is able to detect objects like lanes, crossroads or vehicles. Each object is represented in the robot's egocentric coordinate frame by a pose ${}^{\mathrm{O}}\mathbf{P}_{\mathrm{E}}$ that describes the location of an object's origin by means of Cartesian coordinates x, y, z as well as roll angle Φ, pitch angle Θ, and yaw angle Ψ. In addition, each object description contains parameters to describe its geometric properties, such as curvature c_0 and change in curvature c_1 in case of lanes or crossroads, or length and height in case of vehicles, bicycles or pedestrians. Poses represent the transformation between coordinate systems, e.g. from coordinate frame $\mathbf{S}_1$ to coordinate frame $\mathbf{S}_2$. Poses are extracted from homogeneous 4×4 transformation matrices, as described in appendix A.2 and A.3. Homogeneous transformation matrices are composed of a rotation part $\mathbf{R}$ and a translation part $\mathbf{T}$, as shown in 4.5.

$$ {}^{\mathrm{S2}}\mathbf{H}_{\mathrm{S1}} = \begin{pmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix} \tag{4.5} $$

To describe the transformation between two coordinate frames $\mathbf{S}_1$ and $\mathbf{S}_3$ while taking the joined coordinate frame $\mathbf{S}_2$ into consideration, the homogeneous matrices ${}^{2}\mathbf{H}_{1}$ and ${}^{3}\mathbf{H}_{2}$ are chained together by a matrix multiplication to yield the transformation

$$ {}^{3}\mathbf{H}_{1} = {}^{3}\mathbf{H}_{2} \cdot {}^{2}\mathbf{H}_{1} \tag{4.6} $$

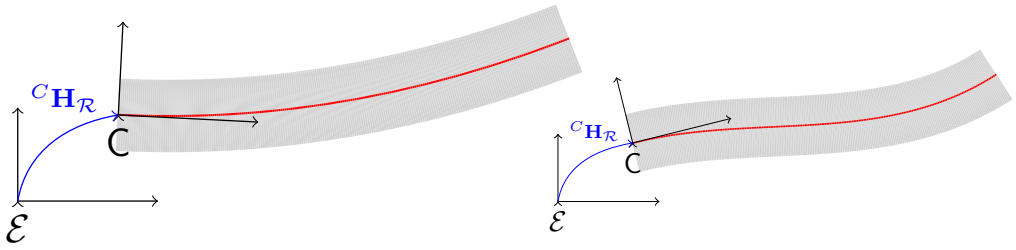
To represent a 3D point $\begin{pmatrix} x & y & z \end{pmatrix}^T$ when a homogeneous transformation matrix $\mathbf{H}$ in another coordinate frame is given, its Euclidean coordinates must be transformed into homogeneous coordinates $\begin{pmatrix} x & y & z & 1 \end{pmatrix}^T$:

$$ \begin{pmatrix} x' & y' & z' & w \end{pmatrix}^T = \mathbf{H} \cdot \begin{pmatrix} x & y & z & 1 \end{pmatrix}^T . \tag{4.7} $$

To keep the notation as simple as possible 3D points are directly multiplied with a homogeneous transformation matrix. The conversion to homogeneous coordinates as well as the conversion from homogeneous coordinates back to Euclidean coordinates of the transformed 3D point is thus implicit, i.e. when two 3D points $\boldsymbol{p}_E$ and $\boldsymbol{p}_O$ in ego-centric and object-centric coordinates are given:

$$ \boldsymbol{p}_O = {}^{O}\mathbf{H}_{E} \cdot \boldsymbol{p}_E, \;\; \boldsymbol{p}_E = {}^{O}\mathbf{H}_{E}^{-1} \cdot \boldsymbol{p}_O \tag{4.8} $$

In addition to the transformation of a lane, its course is represented by clothoid segments, as shown in figure 4.5.



(a) **(b)**

Figure 4.5:
Distinctively curved clothoid samples *(red line sequences)* with different transformations relative to robot $\mathcal{E}$, (a): clothoid C with constant, positive curvature c_0 and negative rotation with respect to the yaw axis, (b): negative curvature c_0 with positive change in curvature c_1 resulting in an S-shaped center line.

Clothoids are characterized by a start curvature c_0 as well as a change of curvature c_1. To get the lane azimuth χ at a given arc length L equation 4.9 needs to be solved

$$\chi(L) = \Psi + \int_0^L c(l)dl = \Psi + \left(c_0 + \frac{c_1 L}{2}\right) L, \tag{4.9}$$

$$\text{with } c(l) = c_0 + c_1 l, \tag{4.10}$$

with l being a control variable with respect to the arc length along the clothoid segment and Ψ being the yaw angle at the origin of the lane. Given origin $\begin{pmatrix} x_C(0) & y_C(0) & 0 \end{pmatrix}^T$, each coordinate $x_C(l_i)$, $y_C(l_i)$ of the skeleton line $\boldsymbol{p}_C(l) = \begin{pmatrix} x_C(l) & y_C(l) & 0 & 1 \end{pmatrix}^T$ of the clothoid segment results from integrating the lane azimuth over the arc length of the segment and is approximated by the sum:

$$x_C(l_i) = x_C + \int_0^{l_i} \cos(\chi(l))dl \approx x_C + \sum_{j=0}^{j=i} \cos(\chi(l_j))\Delta l \tag{4.11}$$

$$y_C(l_i) = y_C + \int_0^{l_i} \sin(\chi(l))dl \approx y_C + \sum_{j=0}^{j=i} \sin(\chi(l_j))\Delta l, \tag{4.12}$$

with l_i being a discrete sample of the arc length along the clothoid segment and Δl being the distance between subsequent samples. Then by considering lane width b_C and lane azimuth $\chi(L)$ the borders of the lane $\boldsymbol{p}_b(l)$ are described as follows.

$$\boldsymbol{p}_b(l) = \boldsymbol{p}_C(l) \pm \frac{1}{2} b_C \begin{bmatrix} -\sin(\chi(L=l)) \\ +\cos(\chi(L=l)) \\ 0 \\ 1 \end{bmatrix} \tag{4.13}$$

Hence, in addition to origin $\begin{pmatrix} x_C(0) & y_C(0) & 0 \end{pmatrix}^T$ a clothoid C is described by the parametric description $[\Psi_C, c_{0C}, c_{1C}, b_C]^T$.

Instead of just one clothoid segment, crossroads are represented by n segments, conditioned on the number of branches. Again, the origin of the crossroad $\begin{pmatrix} x_{Cr}(0) & y_{Cr}(0) & 0 \end{pmatrix}^T$, which is typically the intersection point of the branches, defines the relative spatial

relation to the robot. Each branch k is then represented by a parameter description $[\Psi_k, c_{0k}, c_{1k}, b_k]^T$, which equals the former description of a clothoid.

The object detection and tracking mechanisms of our robot provide information about relative position and orientation as well as information about the spatial dimensions of each moving object, such as vehicles, bicycles or pedestrians in form of an object enclosing bounding box. The center of an object is represented by a transformation ${}^{\mathbf{S}_O}\mathbf{H}_{\mathbf{S}_E}$ from the ego-centric robot coordinate frame $\mathbf{S}_E$ to the object coordinate frame $\mathbf{S}_O$ (see figure 4.6). I.e. in addition to origin $\begin{pmatrix} x_O & y_O & 0 \end{pmatrix}^T$ an object O is described by the parametric description $[\Psi_O, w_O, l_O, h_O]^T$, with Ψ_O being the yaw-angle, w_O the width, l_O the length and h_O being the height of the object.

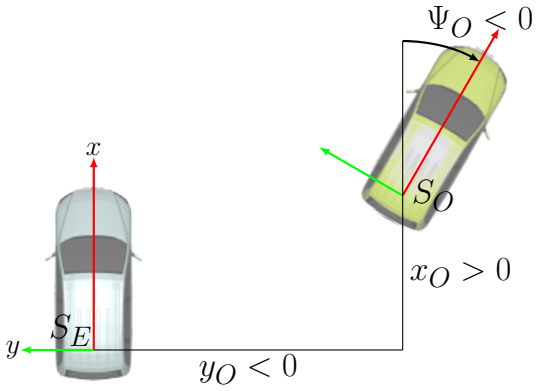


Figure 4.6:
Representation of position and orientation of a perceived object O in the ego-coordinate frame S_E.

The geometric information about the various objects in the robot environment builds the foundation for later processing steps that need to model spatial relationships as well as relative motion over time.

4.5 Environment representation

The environment representation that is proposed in this chapter constitutes the core of the reasoning and control system. Sensor data is ego-centric by definition and perceived object information, be it from camera, LiDAR or radio detection and ranging (RADAR), is natively represented via angles and distances. To best fit this information, the author chose an ego-centric representation, which leads to expressing the motion of the environment in the robot's view, instead of representing the motion of the robot in its environment. This may seem unusual upon first glance.

But the ego-centric representation helps to directly convey perceived, local object data to the robot's reasoning and control systems and to process static and dynamic objects in equal measure in the long term. Throughout the following sections the reasons for choosing this environment representation are clarified and operations between the entities of the representation are introduced. In the latter course of this thesis it will also be shown how these operations help to exploit relative object-object motions, if necessary.

4.5.1 Relation

In approaches to map-based navigation the planning reference is always some artificial reference point in the global map frame (often its origin), which is neither related to any perception nor to a possible robot trajectory. Taking this artificial reference into consideration, obstacle and robot positions are registered in the map and global trajectories are related to the environment. Upon stringent inspection, plans appear to be completely unrelated to the actual perceptions, because trajectories are merely linked by one reference to a map that represents the true environment with potentially large uncertainties. Thus a core design principle was to allow for exchangeable real world references instead of artificial ones in order to provide a more robust foundation for robot navigation as well as motion prediction. The author proposes that these references are the perceived objects themselves as long as they are within sensor range, as they provide a visible reference and a certain connection to the actual robot environment. Instead of linking the behavior of the robot to a global trajectory without any relation to the avoidable objects, the representation described here is aimed at connecting the behavior of the robot closely to the perceived objects.

Thus, the representation aims to express object-object relations from an observer-dependent viewpoint, in order to allow a robot-centered or an object-centered perspective. The object-centered perspective, in particular, is expected to become meaningful in the context of observing and evaluating object-object interactions, e.g. between another vehicle and the road, in order to then apply these interactions (e.g. how another vehicle followed a lane) to the robot itself.

Another core design principle was to represent sensor data as accurate as possible and to enable the direct information transfer to the robot control law, which governs the motion of the robot, without the need for further transformations. Via this link perception is directly coupled to robot control to allow for quick robot motion adaptations in the case of unforeseen object behavior. Hence, the term *relation* is

proposed as presented in definition 1. A *relation* is composed by two angles and a distance, which - in combination - represent detected objects in the environment in a human-like, ego-centric fashion (see figure 4.7 for a visual representation).

Definition 1. A *relation* $\mathfrak{r}(\alpha, \beta, d)$ is a triple $\mathfrak{r} = (\alpha, \beta, d)$ with:

1. α being a scalar that represents the angle to an object with respect to the observers heading with $-\pi < \alpha \leq \pi$,

2. β being a scalar that embodies the relative heading of an object with respect to the vector $\overrightarrow{S_E S_O}$ with $-\pi < \beta \leq \pi$,

3. d being a scalar that corresponds to the distance between observer and object in question with $d \in \mathbb{R}^+$

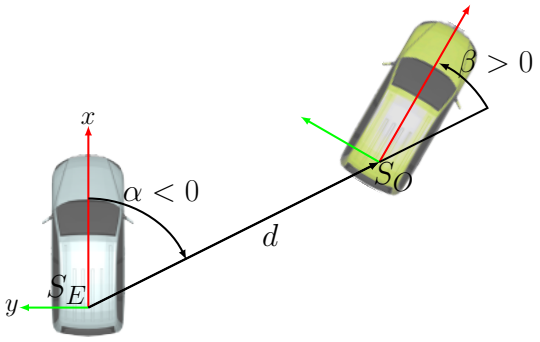


Figure 4.7:
Two objects with their respective poses and the relation between them. Here the relation of an object (*yellow vehicle*) relative to the observer (*white vehicle*) is shown, with positive yaw-angles representing a counter-clockwise rotation.

Consequently, a *relation* shifts the reference frame from a global to a local perspective. In essence, the definition of a *relation* substitutes the elements of the SE(2) (2D object position x_O, y_O, and object orientation Ψ_O) to get the configuration space: $\mathbb{T}^2 \times \mathbb{R}$ with an entity described by (α, β, d). While α and d represent consistent sensor measurements, because they are typically ego-centric with angle and distance information as spatial reference, angle β meets the demands of successive reasoning and robot control steps. With respect to later reasoning steps angle β provides information on the orientation of an object toward an observer, independent of the other two entities, which facilitates faster decisions. For instance, an object with $\beta = \pi$ will always be oriented towards the observer, independent of angle α or distance d. In contrast, to obtain similar information in the configuration space of the SE(2), all entities (x, y, Ψ) need to be evaluated. Moreover, angle β has a large impact on robot control. To steer the robot correctly in relation to another object

the objects orientation needs to be evaluated, because a vehicle following maneuver, for example, requires the robot to stay behind ($\beta \approx 0$) another object.

Although, the representation of the perceived environment via relations according to definition 1 and via elements of the SE(2) are isomorphic (one relation can be exactly mapped to one pose in the SE(2) and vice versa), a relation already provides more useful object information at a single point in time. By exploiting the characteristics of relations over time shows even more significant differences. To this end, the overtaking maneuver of figure 4.8 is examined in both representations, as illustrated in figures 4.9 and 4.10.

Figure 4.8:
General representation of an overtaking maneuver (*blue*), with the white vehicle overtaking the green vehicle.

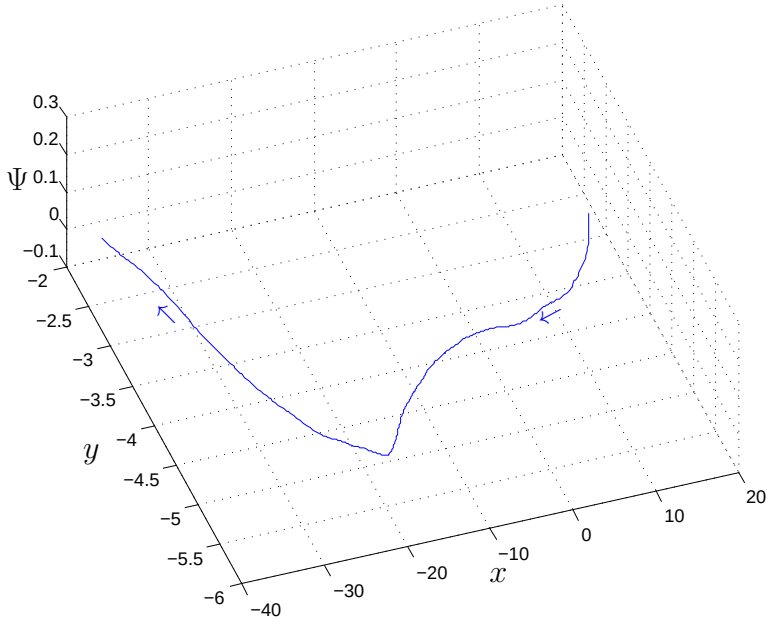


Figure 4.9:
Representation of an overtaking maneuver from an ego-centric perspective (ego being the overtaking vehicle) by means of a sequence of poses (x, y, Ψ) in the SE(2). The start of the maneuver is located at the top right of the figure. The y-coordinate starts with a negative value, since the record started when the ego-vehicle moved slightly to the left already in order to overtake.

Figure 4.9 illustrates the progress of an overtaking maneuver from a robot-centered perspective in the SE(2), with the robot being the overtaking vehicle. The maneuver starts on the right hand side, with a positive x-coordinate of the perceived object,

with the y-coordinate being slightly negative and yaw angle Ψ being close to zero. During the overtaking maneuver, the value of the x-coordinate decreases, since the longitudinal position of the perceived object changes from the front of the robot to the rear of the robot. The value of the y-coordinate decreases up to its peak in the middle of the maneuver, when the robot and the perceived object are right next to each other. It starts to increase again, as the robot positions itself in front of the overtaken vehicle. Only small variations are observed with respect to the yaw-angle Ψ, since the vehicles are aligned to each other almost every time during the maneuver.

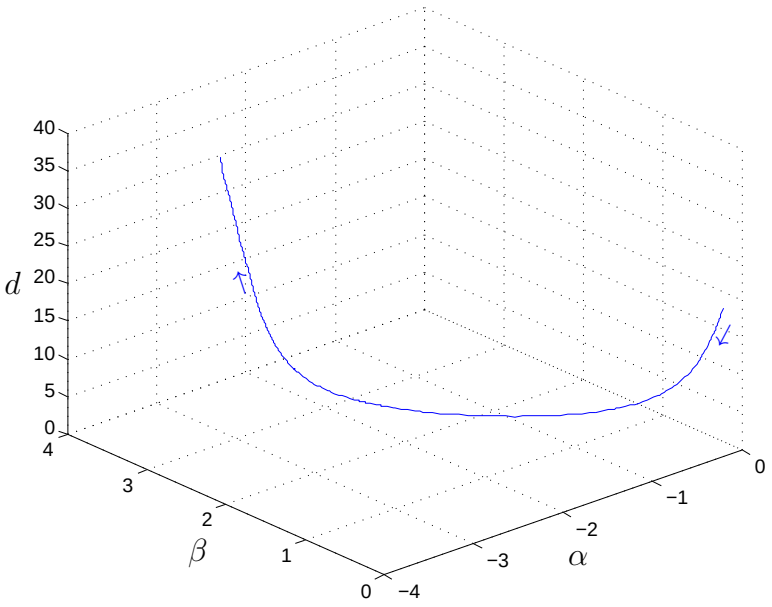


Figure 4.10:
Representation of the overtaking maneuver of figure 4.8 by means of a sequence of relations (α, β, d). The maneuver starts on the right hand side.

The progression of a relation-based representation with respect to an overtaking maneuver (see figure 4.10, start of the maneuver is on the right hand side) seems to mimic the perception of a human observer in a natural fashion. It is apparent that angle α starts at value $\alpha \approx 0$ (i.e. the object resides in front of the robot) and finishes at $\alpha \approx \pi$ (i.e. the other object resides to the rear of the robot). Angle β behaves in a similar fashion to provide information on the relative orientation of the object. At first the rear side of the vehicle is visible ($\beta \approx 0$), then its left side ($\beta \approx \pi/2$), then its front ($\beta \approx \pi$). Finally, distance d becomes smaller upon catching up to the vehicle, remains almost constant during the passing phase (during the overtaking maneuver the relative velocity was low) and becomes larger after the vehicle was passed.

By evaluating each entity of a *relation* independent of the other two, some conclusions can be derived: The evaluation of angle α, for example, may imply that the observer passed another object, while angle β carries information on visible object features or its general heading (away from the observer, towards the observer). Moreover, by solely evaluating distance d the importance of the object may be derived by formulating rules such as: large distances are less important than short distances, for instance.

To enable transformations between relations in order to e.g. exploit the relation between two objects or to allow assumptions about future object states, three operations are defined. Their intended use is to concatenate relations as well as to explore the differential behavior between them. Therefore, inverse relation $\mathfrak{r}^{-1}$ according to definition 2, concatenation $\oplus$ according to definition 3 and difference $\ominus$ according to definition 4 are introduced.

Definition 2. Given relation $\mathfrak{r}(\alpha, \beta, d)$, inverse relation $\mathfrak{r}^{-1}(\alpha^{-1}, \beta^{-1}, d^{-1})$ is defined by:

$$\mathfrak{r}_{\alpha}^{-1} = \pi - \mathfrak{r}_{\beta}$$

$$\mathfrak{r}_{\beta}^{-1} = \pi - \mathfrak{r}_{\alpha}$$

$$\mathfrak{r}_{d}^{-1} = \mathfrak{r}_{d}$$

Inverse relation $\mathfrak{r}^{-1}$ allows to change the viewpoint of a percept. I.e. given an object percept from the viewpoint of the robot, the application of the inverse relation expresses how the robot is perceived from the viewpoint of the object. In figure 4.8 for example, the trajectory of the overtaking maneuver is represented from the viewpoint of the overtaken vehicle, instead of the viewpoint of the robot.

Definition 3. To get the concatenated relation $\mathfrak{r}_{\oplus}$, given two relations $\mathfrak{r}_1$ and $\mathfrak{r}_2$, the binary $\oplus$ operation is defined with $\mathfrak{r}_1 \oplus \mathfrak{r}_2 = \mathfrak{r}_{\oplus}$ as follows:

$$\theta = -\pi - \mathfrak{r}_{1_\beta} - \mathfrak{r}_{2_\alpha}, \ with \ -\pi < \theta \leq \pi$$

$$\mathfrak{r}_{\oplus_d} = \sqrt{\mathfrak{r}_{1_d}^2 + \mathfrak{r}_{2_d}^2 - 2 \cdot \mathfrak{r}_{1_d} \cdot \mathfrak{r}_{2_d} \cdot \cos(\theta)}$$

$$\mathfrak{r}_{\oplus_\beta} = \arccos\left(\frac{\mathfrak{r}_{1_d}^2 - \mathfrak{r}_{2_d}^2 - \mathfrak{r}_{c_d}^2}{-2 \cdot \mathfrak{r}_{2_d} \cdot \mathfrak{r}_{c_d}}\right) \cdot sgn(\theta) + \mathfrak{r}_{2_\beta}$$

$$\mathfrak{r}_{\oplus_\alpha} = \mathfrak{r}_{2_\beta} + \theta + \mathfrak{r}_{1_\alpha} - \mathfrak{r}_{\oplus_\beta}$$

In this context $\mathfrak{r}_{\oplus_\alpha}, \mathfrak{r}_{\oplus_\beta}, \mathfrak{r}_{\oplus_d}$ represent the respective entities of relation $\mathfrak{r}_\oplus$, $sgn(\cdot)$ the signum function and θ the angle between the sides of the triangle in figure 4.11 that are marked as $\mathfrak{r}_{1_d}$ and $\mathfrak{r}_{2_d}$. Value π has to be incorporated into the calculation of angle θ so that the correct angle value and the correct sign are obtained.

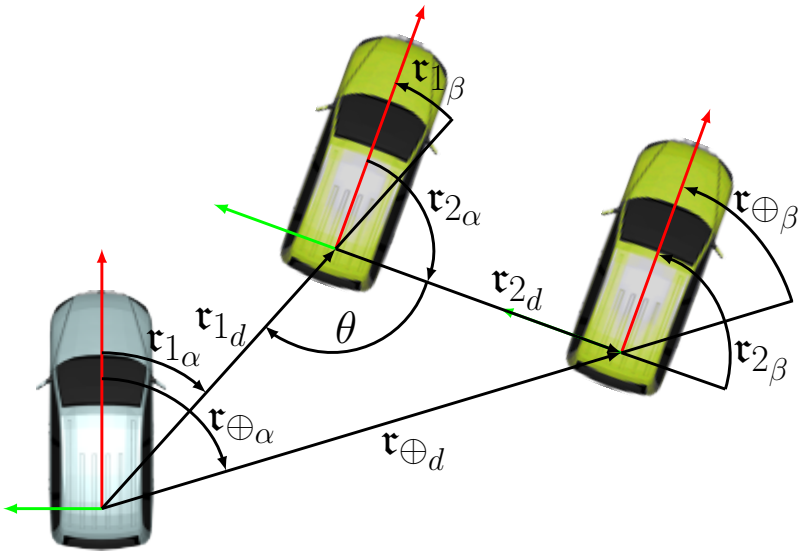


Figure 4.11:
Visualization of the result $(\mathfrak{r}_{\oplus_\alpha}, \mathfrak{r}_{\oplus_\beta}, \mathfrak{r}_{\oplus_d})$ of applying concatenating operation $\oplus$ to relations $\mathfrak{r}_1$ and $\mathfrak{r}_2$.

The concatenation operation $\oplus$ allows for the transformation of a relation according to a given relative motion, as shown in figure 4.11. In this manner future object states may, for example, be derived, when a differential relative motion over time is given.

Definition 4. The binary $\ominus$ operation to compute the difference relation $\mathfrak{r}_\ominus$ given two relations $\mathfrak{r}_1$ and $\mathfrak{r}_2$ with $\mathfrak{r}_2 \ominus \mathfrak{r}_1 = \mathfrak{r}_\ominus$ is defined as follows:

$$\theta = \mathfrak{r}_{2_\alpha} - \mathfrak{r}_{1_\alpha}$$

$$\mathfrak{r}_{\ominus_d} = \sqrt{\mathfrak{r}_{1_d}^2 + \mathfrak{r}_{2_d}^2 - 2 \cdot \mathfrak{r}_{1_d} \cdot \mathfrak{r}_{2_d} \cdot \cos(\theta)}$$

$$\mathfrak{r}_{\ominus_\beta} = \arccos\left(\frac{\mathfrak{r}_{1_d}^2 - \mathfrak{r}_{2_d}^2 - \mathfrak{r}_{d_d}^2}{-2 \cdot \mathfrak{r}_{2_d} \cdot \mathfrak{r}_{d_d}}\right) \cdot -sgn(\theta) + \mathfrak{r}_{2_\beta}$$

$$\mathfrak{r}_{\ominus_\alpha} = \mathfrak{r}_{2_\beta} + \theta - \mathfrak{r}_{1_\beta} - \mathfrak{r}_{d_\beta}$$

The $\ominus$-operation (see figure 4.12) enables the evaluation of the relation between two objects. Its application over time enables the exploitation of the relative motion between two objects, which impacts object motion prediction in particular. Both

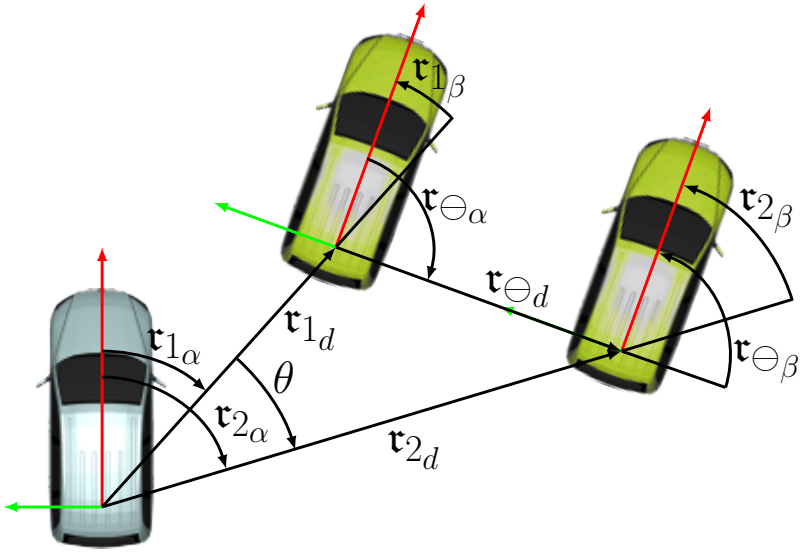


Figure 4.12:
Visualization of the result $(\mathfrak{r}_{\ominus_\alpha}, \mathfrak{r}_{\ominus_\beta}, \mathfrak{r}_{\ominus_d})$ of applying the difference operation $\ominus$ to $\mathfrak{r}_1$ and $\mathfrak{r}_2$.

operations are meaningful in the context of environment modelling, as they can represent environment dynamics.

At this point one of the most crucial differences to common approaches to robot navigation emerges: typically, a robot moves in a map built from static obstacles. This procedure facilitates an update step where one entity (the robot) needs to be transformed only, while the possibly numerous obstacles remain untouched. But as a result, dynamic objects need to be considered separately from the rest of the obstacles. By representing the whole environment by means of relations from an ego-centric perspective, however, each object (static as well as dynamic) is updated in relation to the robot. I.e. instead of moving the robot through a static environment, the entire environment is transformed and the ego-centric perspective is updated.

Another aspect of the relational nature of the representation is that the relative behavior of objects can be exploited over time. In that manner, objects that maintain their relative spatial position and orientation, i.e. objects that move in the same way, can be grouped together. This would include all static objects or even dynamic objects that move in accord (e.g. a car and its trailer), which offers the opportunity to update them in conjunction. I.e. instead of updating each object separately, these operations could be applied to object clusters and thus the ego-centric representation would be updated more effectively. By considering the entire environment as being dynamic, because each object moves in the egocentric perspective of the robot

independent of the object's own motion, the unification of static and dynamic objects without the need for any specialized operations is enabled.

4.5.2 Data approximation

It is one of the central ideas of the object-related navigation approach to describe relative object motion over time in order to loop perceived object relations directly through to the robot control law on the one hand and in order to allow assumptions about relative behavior over time on the other hand. The latter may then be used to either predict the motion of arbitrary objects relative to the robot or to predict the relative motion between objects in the robot environment.

One example of a relative motion over time was already shown in figure 4.10, where an overtaking maneuver was represented by means of relations. Many such sequences can be captured for different maneuvers, like e.g. *"follow road"*, *"turn left"* or *"follow vehicle"*, to name some prominent examples. Each such motion may contain up to many thousand data points, depending on the duration of the motion. To store, evaluate and re-use (e.g. for motion prediction or robot control) such vast amounts of data consumes time and processing power.

Hence, each motion sequence will be approximated to reduce the load of stored and processed information, which in turn allows for quicker data availability. Typically, regression methods like the least squares [Kariya and Kurata, 2004] approximation, Bayesian methods [Carlin and Louis, 2009] or neural networks [McCulloch, 1943] are used to approximate a given data set. In essence, these methods approximate given data via some sort of base function, e.g. higher order polynomials, trigonometric functions, Gaussians, etc.

The general form of an approximation $\mathcal{A}$ given measurements $\mathbf{Y_k} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_k)$ is represented by:

$$\mathcal{A}(\mathbf{Y_k}) = \sum_{j=1}^{m} \omega_j h_j(\mathbf{Y_k}) \tag{4.14}$$

with $h_j(\cdot)$ being some sort of basis function and ω_j its weight in the final approximation. The number m of basis functions as well as the function itself must be

previously defined. Weights ω_1 till ω_m, as well as internal function parameters will be adapted during the approximation process.

In the following section it is discussed, how well sequences of relational data can be expressed by possible approximations. The choice of the base functions is explained, as well as their internal parameters. This section also compares how well one relative motion over time can be approximated in two different spatial representations. It will be examined how well the motion can be approximated by means of relations according to definition 1 in comparison to an approximation by means of the entities in the SE(2).

For that purpose the exemplary overtaking maneuver of the previous section (compare figures 4.9 and 4.10) is examined. In order to use simple base functions, the sequence of relations as well as the sequence of poses are decomposed into their respective components. Figure 4.13 (*black dots*) provides an insight, of what an overtaking maneuver looks like over time t, when only angle β of a relation and the y-coordinate of the SE(2) are considered.

The goal is to find a function that best fits given data characteristics. In the example of figure 4.13a for instance, the inverse tangent function $\tan^{-1}(t)$ seems to be a good fit, since the progress of the data (*black dots*) aligns to the form of the function (see figure 4.14) very well.

In order to be able to adapt the form of the function to the characteristics of the data, additional parameters are introduced. In order to shift and stretch the function along the y-axis the function, parameters o_β and s_β are added to the equation of the inverse tangent function to yield: $s_\beta \cdot \tan^{-1}(t) + o_\beta$. Furthermore parameters o_t and s_t are introduced to shift and shear the function along the t-axis. This results in the base function to approximate angle β of a sequence of relations, as shown in equation 4.15.

$$\beta(t) = s_\beta \cdot \tan^{-1}(s_t(t + o_t)) + o_\beta \tag{4.15}$$

Two examples of a successful data approximation can be seen in figure 4.13. Alongside the captured data (*black*), both figures provide the approximations of each entity (*blue*). In the same fashion the most fitting functions for each element of a relation

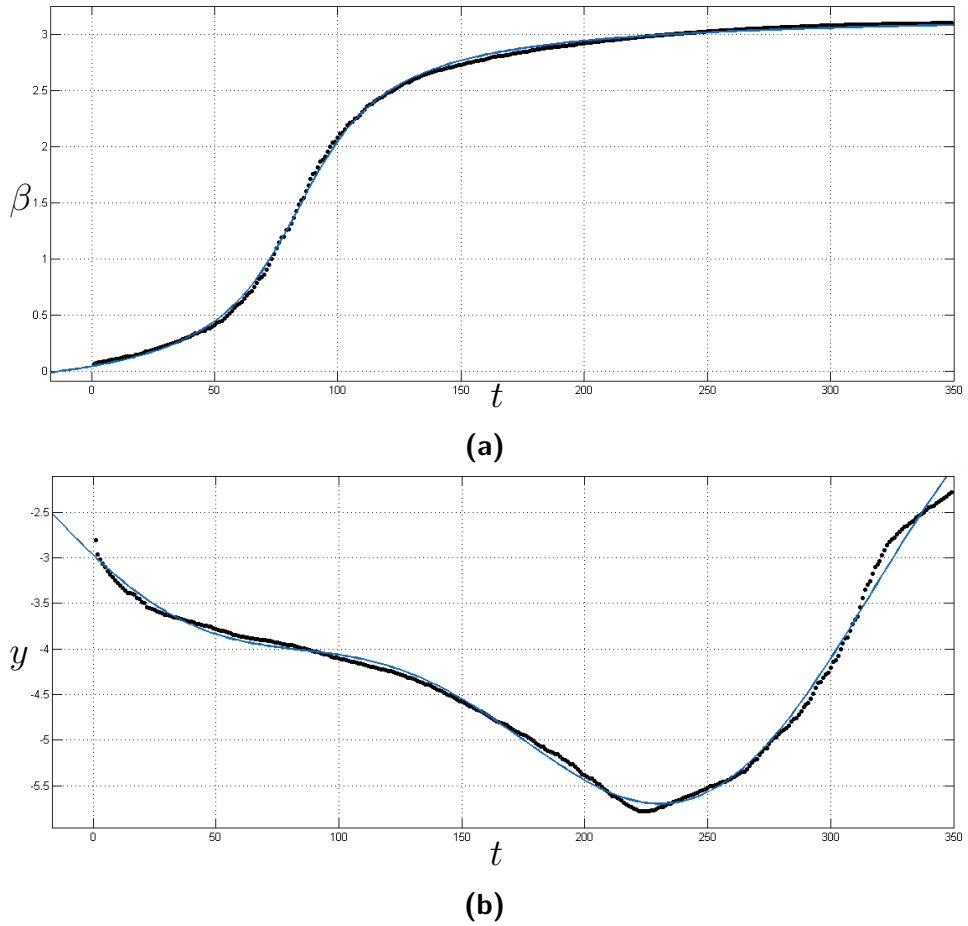


Figure 4.13:
Figure 4.13a shows the progress of angle β during an overtaking maneuver (*black samples*). Figure 4.13b then shows the progress along the y-axis, when the same maneuver is represented in the SE(2)-group (*black samples*). The curves (*blue*) provide an approximation of the respective data.

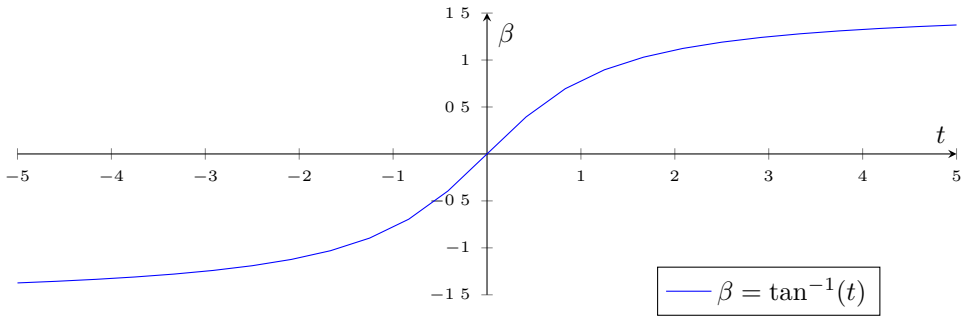


Figure 4.14: The inverse tangent function in range -5 to 5.

and for each element of a pose in the SE(2) can be determined. To this end, different approximation functions were evaluated with respect to the given data.

In addition to the inverse tangent function tan^{-1}, the four approximation functions, Fourier series, forth, sixth and eighth-order polynomial turned out to be the most promising fits, given the data of an overtaking maneuver. They are defined, as shown in equations 4.16 and 4.17, with $f(t)$ being the general representation of a Fourier series and $p(t)$ the representation of a polynomial of order N.

$$f(t) = a_0 + \sum_{n=1}^{N} a_n \cos(n \cdot t) + \sum_{n=1}^{N} b_n \sin(n \cdot t) \tag{4.16}$$

$$p(t) = p_0 + \sum_{n=1}^{N} p_n \cdot t^n \tag{4.17}$$

To add the possibility to shift and shear the given functions, t of equations 4.16 and 4.17 is substituted by $s_t \cdot (t + o_t)$ with s_t being the parameter responsible for shearing and o_t being responsible for shifting along the t-axis. This results in equations 4.18 and 4.19.

$$f(t) = a_0 + \sum_{n=1}^{N} a_n \cos(n \cdot s_t(t + o_t)) + \sum_{n=1}^{2} b_n \sin(n \cdot s_t(t + o_t)) \tag{4.18}$$

$$p(t) = p_0 + \sum_{n=1}^{N} p_n \cdot (s_t \cdot (t + o_t))^n \tag{4.19}$$

During the approximation process, parameters a_n, b_n, p_n, s_t and o_t are adapted to fit the given curve characteristics. To now constitute the ability of a function to represent given data properly, the coefficient of determination R^2, according to equation 4.20 is considered.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} w_i (y_i - f_i)^2}{\sum_{i=1}^{n} w_i (y_i - \overline{y})^2} \tag{4.20}$$

R^2 represents the ratio between the sum of squares of residuals and the total sum of squares. The sum of squares integrates data set values y_i and model values f_i, while the total sum of squares incorporates data set values y_i and the mean of observed data $\overline{y}$. The importance of a sample is represented by weight w_i. In the evaluation

of this thesis, each sample has the same weight $w_i = \frac{1}{n}$, with n being the number of samples. I.e. the residuals are normalized with respect to the variance. Hence, R^2 can be used as a measure for the quality of fit of a model across experimental sample values and also as a measure for the reliability of predicted future outcomes. Table 4.1 provides on overview of how well the mentioned approximation functions are able to represent the given data, based on the coefficient of determination.

Component	tan^{-1}	Fourier	x^4	x^6	x^8
α	0,999	-	-	-	-
β	0,9995	-	-	-	-
d	-	0,9994	0,9976	0,9993	0,9996
x	-	1	1	1	1
y	-	0,9935	0,9545	0,9964	0,9969
Ψ	-	0,9897	0,984	0,9952	0,9967

Table 4.1:
Coefficient of determination R^2 given the approximations of the inverse tangent (tan^{-1}), Fourier series ($Fourier$), forth, sixth and eighth-order polynomials (x^4, x^6 x^8) and the data represented by figure 4.13.

Each row of table 4.1 represents one of the dimensions (α, β, d) of a relation and one of the dimensions of a pose (x, y and Ψ) in the SE(2). Each column represents the type of approximation function, namely the inverse tangent (tan^{-1}), the Fourier series ($Fourier$) a forth- (x^4), sixth- (x^6) or eight-order (x^8) polynomial. Each value of table 4.1 represents the respective coefficient of determination. With a coefficient of determination above $0{,}999$ the inverse tangent is able to approximate angles α and β with high accuracy. In the same fashion the Fourier series approximation is able to reproduce the samples of distance d quite accurately. While the x-component of the SE(2) can be reproduced exactly with a forth-order polynomial, the quality of approximation of the y and Ψ components drops, even by further increasing the order of the polynomial.

During the evaluation procedure the degree of the Fourier series was limited to 2, since a higher degree was not able to explain the given data with higher precession. In accordance to the evaluation of table 4.1 and the findings during various approximation attempts (including motion sequences of evasive and yield maneuvers as shown in appendix C), sequences of relative object-object motion will thus be approximated by:

$$\mathcal{A}(\mathbf{Y_k}) = \Big(\alpha(\mathbf{Y_k})\ \beta(\mathbf{Y_k})\ d(\mathbf{Y_k})\Big)^T \text{ with:} \tag{4.21}$$

$$\alpha(t) = s_\alpha \cdot \tan^{-1}(s_{t_\alpha}(t + o_{t_\alpha})) + o_\alpha \tag{4.22}$$

$$\beta(t) = s_\beta \cdot \tan^{-1}(s_{t_\beta}(t + o_{t_\beta})) + o_\beta \tag{4.23}$$

$$d(t) = a_0 + \sum_{n=1}^{2} a_n \cos(n s_t(t + o_t)) + \sum_{n=1}^{2} b_n \sin(n s_t(t + o_t)) \tag{4.24}$$

, where $\mathbf{Y_k} = (y_1, y_2, \ldots, y_k)$ is a set of relative object-object motions, represented by means of relations, according to definition 1 and $o_{(\cdot)}$,$s_{(\cdot)}$, $a_{(\cdot)}$ and $b_{(\cdot)}$ are the parameters to be optimized.

The set of parameters is estimated by applying a Levenberg-Marquardt (LM)-solver [Levenberg, 1944] to the optimization problem stated by equation 4.21. The LM-algorithm minimizes error metrics by iteratively computing a numerical solution for non-linear regression problems taking into account the Least Squares (LS). Like other approaches to iterative minimization, the LM-algorithm requires an initial guess, which in turn influences the final solution in case of multiple local minima. The LM-solver of this thesis resorts to the implementation of the publicly available Eigen3 library for linear algebra [Guennebaud et al., 2010].

With respect to the quality of the data approximation, two experiments were conducted. Their aim is to foster the numbers of table 4.1 and to show how changes in the coefficient of determination R^2 impact the quality of the approximation. The first experiment relates to how missing data influences the final data approximation. Data from the middle of an overtaking maneuver was removed. Figure 4.15 shows the missing data approximation in the relation-based space. Information (*green points*) in the middle of the maneuver was removed from the original data (*blue points*). Although the data was removed, the approximation based on this reduced data set (*red curve*) closely relates to the approximation applied to the full data set (*magenta*), as similar curve characteristics can be observed.

In the same fashion, data was approximated that resulted from representing the very same overtaking maneuver in the SE(2). Figure 4.16 shows both, the original data set (*blue*) and the information that was removed (*green*). In contrast to the approximations in the relation based representation, huge deviations can be observed. The approximation (*red*) that relies on the reduced data set alone strongly deviates

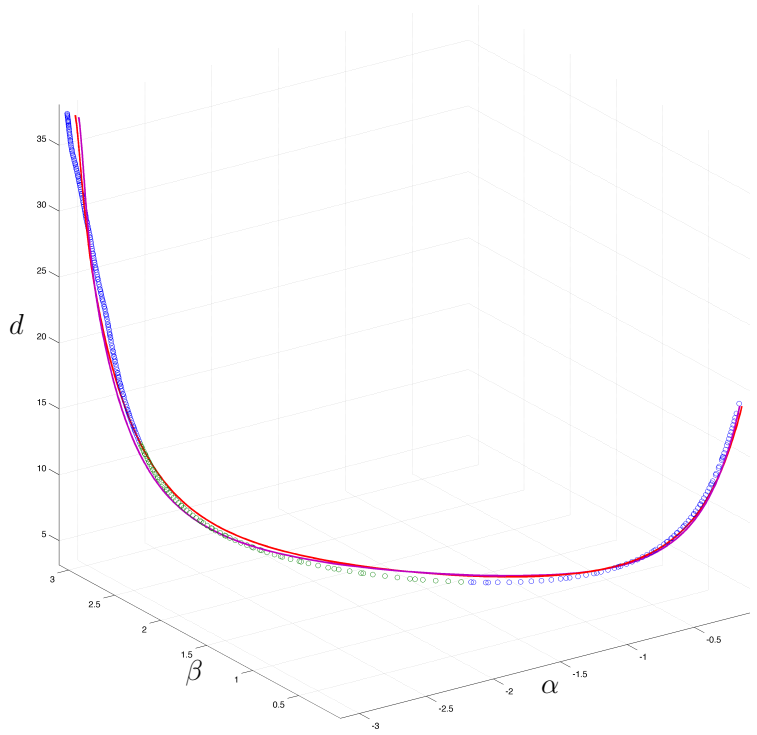


Figure 4.15:
Missing data approximation in the relation-based space. Information (*green circles*) in the middle of the maneuver was removed from the original data (*blue circles*). Both approximations (*red: data with missing information, magenta: original data*) in the relation-based space show a similar progress.

from the full data approximation (*magenta*), especially in the region, where the data was removed.

The second experiment examines how well an approximation is able to extrapolate given data. The complete set of raw data of figures 4.15 and 4.16 was used in order to provide an approximation. Figure 4.17 shows this process, with the points colored red being the original data and the points colored blue being samples drawn from the approximation. The transition between the original data and the samples drawn from the approximation is fluent (magnification of figure 4.17), since the approximation catches the data characteristics very well.

In contrast, by applying the same process to the complete data set of figure 4.18 huge deviations between real data (*red*) and samples drawn from the approximation (*blue*) are observable. Especially at the start and the end of the maneuver, where extrapolations are typically meaningful.

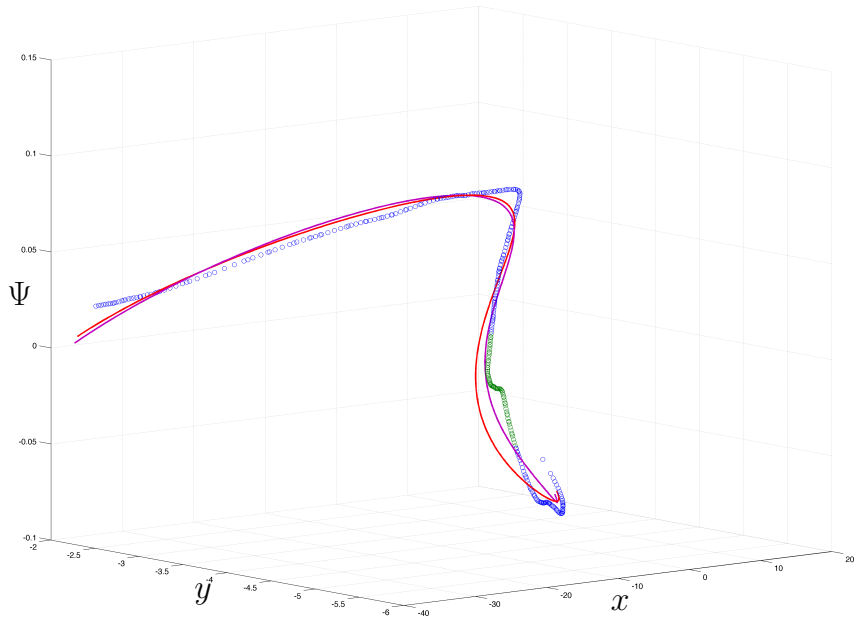


Figure 4.16:
Missing data approximation in the SE(2)-group. Information (*green circles*) in the middle of the maneuver was removed from the original data (*blue circles*). The approximation of the reduced data set (*red line sequence*) shows huge deviations from the approximation based on the full data (*magenta line sequence*), especially close to the area with the information removed.

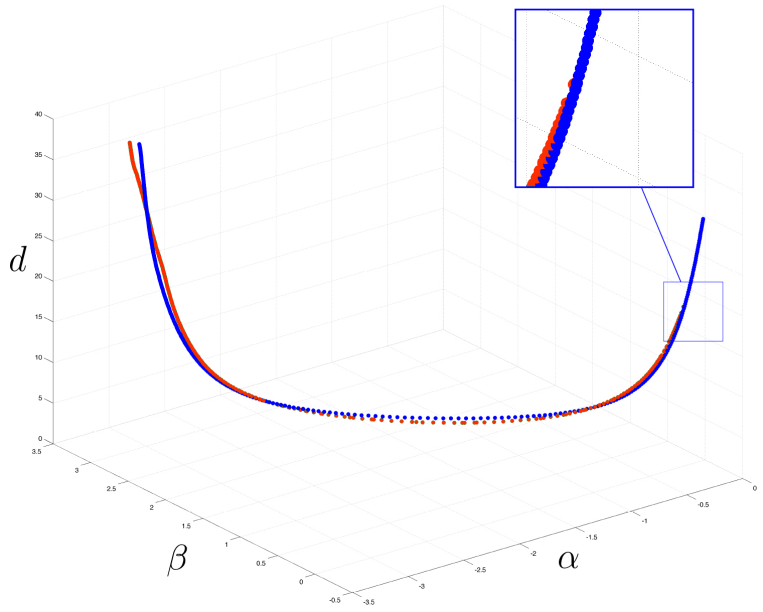


Figure 4.17:
Extrapolation of a maneuver in the relation based space. The sampled data from the extrapolation (*blue points*) shows a fluent transition between original (*red*) and extrapolated data (see magnification).

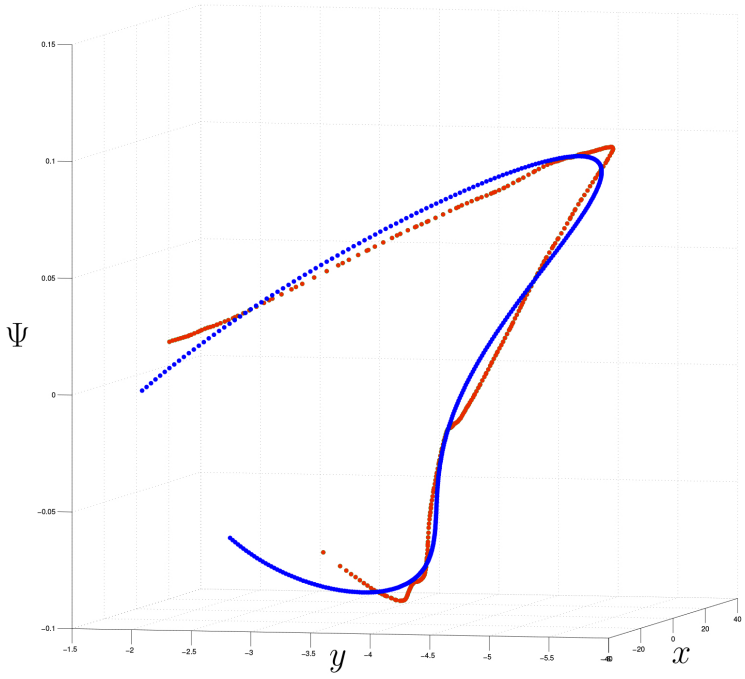


Figure 4.18:
Extrapolation of a maneuver in the SE(2)-group. The sampled data from the extrapolation (*blue points*) shows strong deviations from the original data (*red points*).

These simple experiments show how even small differences in the coefficient of determination of table 4.1 impact the quality of the final approximation. Furthermore, while data drawn from a relation based representation can be approximated very well, sequences of poses in the SE(2) are difficult to characterize. Based on the above considerations a relation based representation seems to benefit data approximation, when ego-centric environment data is considered, compared to a representation in the SE(2). Thus, the relation based representation and its approximation will be used in the following section to generate a more general motion representation: an action.

4.6 Learning behavior models of relative motion

One fundamental ingredient of the object-related navigation approach are actions. An action ${}^{O_j}A_{O_i}$ represents a certain behavior between an object O_i relative to object O_j and serves as input for the reasoning as well as the control system of the robot. Examples are *"follow"* or *"overtake"*, where object O_i and O_j might be two vehicles in the vicinity of the robot or *"turn right"*, where O_i might be the robot and O_j a crossroad. Each action is composed of a name, a link to its objects (e.g.

vehicle - vehicle or robot - lane) and at least one sequence of relations that describes the action in a geometrical fashion. The sequence of relations of figure 4.15 that describes an overtaking maneuver is one such example.

The following sections will describe how such sequences are captured, combined and stored in order to provide meaningful actions. Furthermore, their capability to predict the motion of other traffic participants is shown.

4.6.1 Sequence generation

A relative motion between two objects O_1 (observer) and O_2 (observed) can be represented by a sequence ${}^{O_2}\mathcal{R}_{O_1} = (\mathfrak{r}(t_1), \mathfrak{r}(t_2), \ldots, \mathfrak{r}(t_n), v(t_1), v(t_2), \ldots, v(t_n))$. It is composed of relations $\mathfrak{r}(t_k)$ and relative velocities $v(t_k)$ for each step k. These sequences are generated by exploiting the set of objects provided by the perception mechanisms of our robot (see section 4.4) and their relative motion behavior.

Relations to objects, like vehicles or pedestrians, are obtained by evaluating the reference pose ${}^{O}\mathbf{P}_{\mathcal{E}}$ of object O with respect to robot $\mathcal{E}$. How poses are derived from the homogeneous transformation matrices of section 4.4 is presented in appendix A.3. Pose ${}^{O}\mathbf{P}_{\mathcal{E}}$ is represented in the robot coordinate frame $\mathbf{S}_{\mathcal{E}}$, which represents the base coordinate frame with respect to every position and relation in later chapters and is located at the center of the rear axis of our robot, as shown in figure 4.19a.

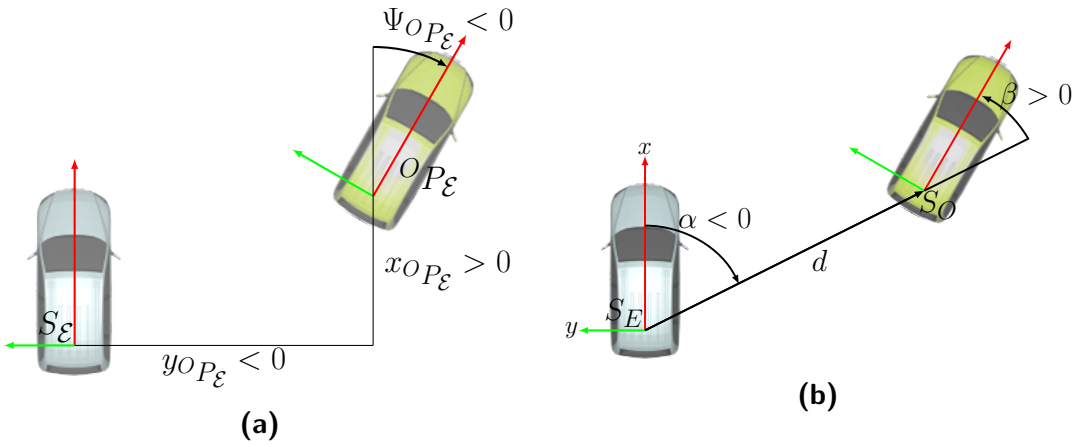


Figure 4.19:
Different location representations. In 4.19a the position and orientation of an object (*green*) is represented by means of the SE(2). In figure 4.19b the same location is represented by a relation.

The rear axis frame is also the reference system for the control law governing the motion of the robot. By linking both, a close connection of perception and robot control is facilitated. Since every coordinate is defined in $\mathbf{S}_{\mathcal{E}}$, the transformation described by pose ${}^{O}\mathbf{P}_{\mathcal{E}}$ acts in concert with the transformation represented by a relation. By extracting the Cartesian coordinates $xo_{\mathbf{P}_{\mathcal{E}}}$ and $yo_{\mathbf{P}_{\mathcal{E}}}$ as well as yaw angle $\Psi o_{\mathbf{P}_{\mathcal{E}}}$ of an object O, the entities of a relation $\mathfrak{r}(\alpha, \beta, d)$ can be determined as follows:

$$\alpha = atan2(yo_{\mathbf{P}_{\mathcal{E}}}, xo_{\mathbf{P}_{\mathcal{E}}}) \tag{4.25}$$

$$\beta = \Psi o_{\mathbf{P}_{\mathcal{E}}} - \alpha \tag{4.26}$$

$$d = \sqrt{x^2_{O_{\mathbf{P}_{\mathcal{E}}}} + y^2_{O_{\mathbf{P}_{\mathcal{E}}}}} \tag{4.27}$$

A motion sequence ${}^{O}\mathcal{R}_{\mathcal{E}} = (\mathfrak{r}(t_1), \mathfrak{r}(t_2), \ldots, \mathfrak{r}(t_n), v(t_1), v(t_2), \ldots, v(t_n))$ of robot $\mathcal{E}$ in relation to object O is created by stringing the individual relations of a motion together. Relative velocities $v(t)$ are obtained by subtracting object velocity at time step t from the robot velocity at time step t.

Relations to a lane are represented by evaluating its skeleton line, which is composed of points $\boldsymbol{p}_C(l_k)$ of a clothoid C at run length l_k, as described by equations 4.11 and 4.12 of section 4.4. Like an object pose ${}^{O}\mathbf{P}_{\mathcal{E}}$, the set of skeleton points $(\boldsymbol{p}_C(l_1), \boldsymbol{p}_C(l_2), \ldots, \boldsymbol{p}_C(l_n))$ is represented in the robot coordinate frame $\mathbf{S}_{\mathcal{E}}$. If two points $\boldsymbol{p}_C(l_k)$ and $\boldsymbol{p}_C(l_{k+1})$, each composed of a x and a y-coordinate, out of a sequence of points $(\boldsymbol{p}_C(l_1), \ldots, \boldsymbol{p}_C(l_n))$ are given, a transformation into the space made up of relations is performed for each point, according to equations 4.28 to 4.31. The process is similar to the transformation of object poses to relations, but with one additional step. The heading Ψ of a point (direction towards its successor) must be computed by determining the difference to its successor in each dimension (x, y) and applying the arctangent function to it (see equation 4.28).

$$\Psi = atan2(y_{\boldsymbol{p}_C(l_{k+1})} - y_{\boldsymbol{p}_C(l_k)}, x_{\boldsymbol{p}_C(l_{k+1})} - x_{\boldsymbol{p}_C(l_k)}) \tag{4.28}$$

$$\alpha_{\boldsymbol{p}_C(l_k)} = atan2(y_{\boldsymbol{p}_C(l_k)}, x_{\boldsymbol{p}_C(l_k)}) \tag{4.29}$$

$$\beta_{\boldsymbol{p}_C(l_k)} = \Psi - \alpha_{\boldsymbol{p}_C(l_k)} \tag{4.30}$$

$$d_{\boldsymbol{p}_C(l_k)} = \sqrt{x^2_{\boldsymbol{p}_C(l_k)} \cdot y^2_{\boldsymbol{p}_C(l_k)}} \tag{4.31}$$

The result of the complete process is a list $(\mathfrak{r}_1, \mathfrak{r}_2, \ldots, \mathfrak{r}_n)$ of relations $\mathfrak{r}_k$ that represent the skeleton line of a lane.

Instead of storing the whole set of relations of a lane for each time step t_k of a motion sequence, only the perpendicular relation $\mathfrak{r}_\perp$ of robot $\mathcal{E}$ with respect to lane L (see figure 4.20) is stored.

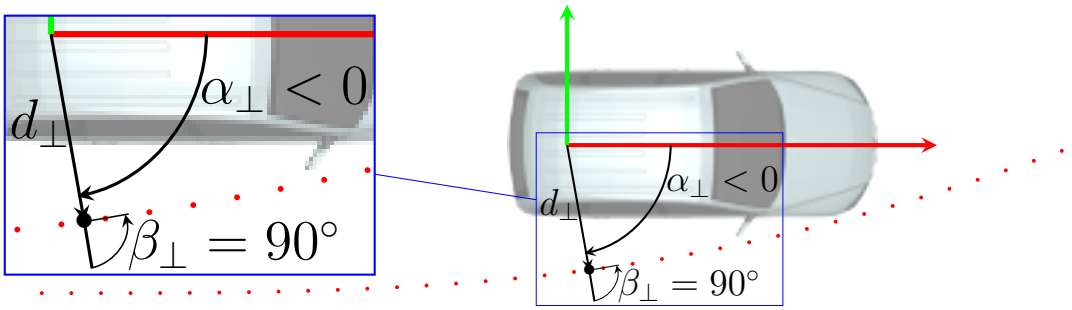


Figure 4.20:
The perpendicular relation $\mathfrak{r}_\perp(\alpha_\perp, \beta_\perp, d_\perp)$ (*black dot*) of the robot with respect to a given point sequence, that describes the center skeleton of a lane.

To safe computational effort, the determination of the perpendicular relation is subdivided into two parts. First, the two points (relations) of the sequence that limit the perpendicular relation to the left and to the right (see e.g. the *red* points of figure 4.20 right next to the perpendicular relation) are determined. For each successive pair of relations $\mathfrak{r}_k$ and $\mathfrak{r}_{k+1}$ in the list of relations, the system checks, whether any interior angle of the triangle made up of relations $\mathfrak{r}(0,0,0)$ (robot reference), $\mathfrak{r}_k$ and $\mathfrak{r}_{k+1}$ is larger than 90° (see figure 4.21 for visual confirmation).

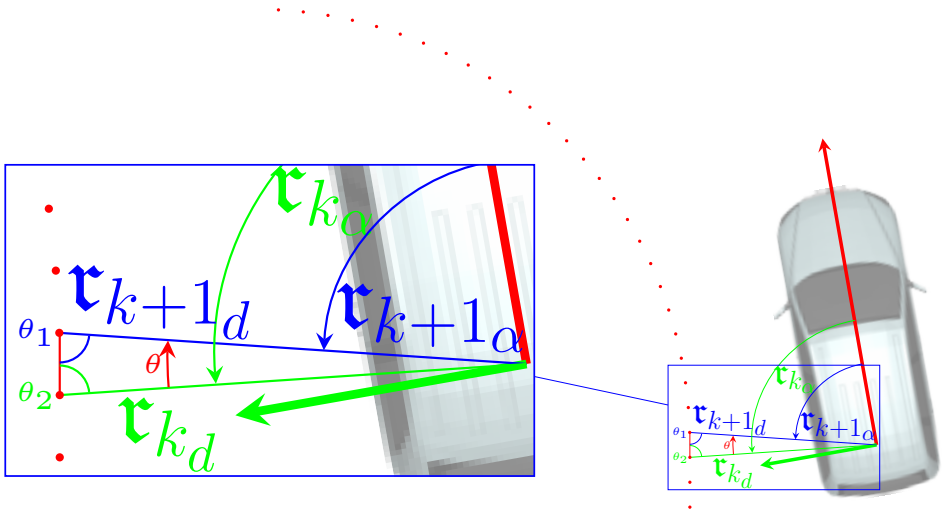


Figure 4.21:
Check whether any interior angle (θ_1, θ_2) of the triangle made up of relations $\mathfrak{r}(0,0,0)$ (robot reference), $\mathfrak{r}_k$ and $\mathfrak{r}_{k+1}$ is larger than 90°.

Equations 4.32 to 4.37 describe the respective mathematical statements. First, the interior angle between the two relations $\mathfrak{r}_k$ and $\mathfrak{r}_{k+1}$ is determined (equation 4.32). Based on this information the distance between the two relations (*red* line in figure 4.21) is computed (equation 4.34). Finally, both interior angles θ_1 and θ_2 can be determined, according to equations 4.35 and 4.36 and the result can be checked according to equation 4.37.

$$\theta = \mathfrak{r}_{k+1_\alpha} - \mathfrak{r}_{k_\alpha} \tag{4.32}$$

$$a = \mathfrak{r}_{k_d} \cdot \mathfrak{r}_{k+1_d} \cdot \cos(\theta) \tag{4.33}$$

$$d = \sqrt{\mathfrak{r}_{k_d}^2 + \mathfrak{r}_{k+1_d}^2 - 2a} \tag{4.34}$$

$$\theta_1 = \arccos\left(\frac{\mathfrak{r}_{k_d}^2 - \mathfrak{r}_{k+1_d}^2 - d^2}{-2 \cdot \mathfrak{r}_{k+1_d} \cdot d}\right) \tag{4.35}$$

$$\theta_2 = \arccos\left(\frac{\mathfrak{r}_{k+1_d}^2 - \mathfrak{r}_{k_d}^2 - d^2}{-2 \cdot \mathfrak{r}_{k_d} \cdot d}\right) \tag{4.36}$$

$$\mid \theta_1 \mid \leq \frac{1}{2}\pi \text{ and } \mid \theta_2 \mid \leq \frac{1}{2}\pi \tag{4.37}$$

In case of a successful outcome of the former check, perpendicular relation $\mathfrak{r}(\alpha_\perp, \beta_\perp, d_\perp)$, or $\mathfrak{r}_\perp$ for short, with respect to two relations $\mathfrak{r}_k$ and $\mathfrak{r}_{k+1}$ is then computed according to equations 4.38 to 4.40. Values a, d and θ_2 of equations 4.33, 4.34 and 4.36 are re-used to determine $\alpha_\perp$ and $d_\perp$ given the law of cosines. The value of entity β is dependent on the sign of $\alpha_\perp$, as shown in equation 4.39. The entirety of values that is involved in the computation is illustrated in figure 4.22.

$$\alpha_\perp = \mathfrak{r}_{k_\alpha} + (\frac{1}{2}\pi - \theta_2) \tag{4.38}$$

$$\beta_\perp = \begin{cases} \frac{1}{2}\pi & \text{if } \alpha_\perp \leq 0 \\ -\frac{1}{2}\pi & \text{if } \alpha_\perp > 0 \end{cases} \tag{4.39}$$

$$d_\perp = \mathfrak{r}_{k_d} \cdot \sqrt{1 - \mathfrak{r}_{k_d}^2 \cdot (\mathfrak{r}_{k_d}^2 - a)^2 \cdot (\mathfrak{r}_{k+1_d} - 2 \cdot a + \mathfrak{r}_{k_d}^2)} \tag{4.40}$$

Each time a lane is measured, the relation $\mathfrak{r}_\perp$ with minimal distance entry $d_\perp$ is used as reference. I.e. for each relative motion sequence between robot $\mathcal{E}$ and lane L the perpendicular relations are stored in the sequence ${}^L\mathcal{R}_\mathcal{E} = (\mathfrak{r}_\perp(t_1), \mathfrak{r}_\perp(t_2), \ldots, \mathfrak{r}_\perp(t_n), v(t_1), v(t_2), \ldots, v(t_n))$ accompanied by the relative ve-

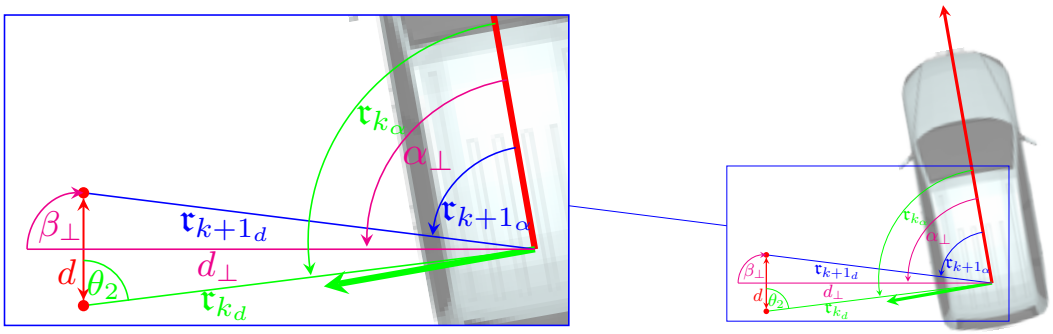


Figure 4.22:
Derivation of the entities of perpendicular relation $\mathfrak{r}_\perp$ when two relations $\mathfrak{r}_k$ (*green*) and $\mathfrak{r}_{k+1}$ (*blue*) are given .

locities $v(t_1), v(t_2), \ldots, v(t_n)$. These are acquired together with the respective lane measurements. Relative velocities at $v(t)$ correspond to the velocity of the vehicle at time step t.

Similar to the acquisition of motion sequences between robot and lane, the relations to a crossroad are examined by evaluating the spatial relation of the robot to a multitude of lanes. Crossroads are defined by number and position of the branches of the respective crossroad. As stated in section 4.4, each crossroad provides n branch descriptions in addition to its origin $\begin{pmatrix} x_{Cr} & y_{Cr} & 0 \end{pmatrix}^T$, which are parametrized by $[\Psi_k, c_{0k}, c_{1k}, b_k]^T$, with $1 \leq k \leq n$. This representation facilitates the generation of n separate lane objects that connect each branch to the one the robot travels on, as shown in figure 4.23.

Each branch is transformed into $\mathbf{S}_\mathcal{E}$ and sampled in a similar fashion to a lane. Instead of connecting the different branches at the center of the crossroad (x_{Cr}, y_{Cr}), however, samples of each branch up to a distance of 6 meters from the center of the crossroad are removed and replaced by the samples of a cubic spline that connects the respective branches, as shown in figure 4.23 (*blue* paths in the inner *white* area of the crossroad). To capture a sequence of relations with respect to a crossroad, the respective branch, which the maneuver aims for, has to be chosen. For instance, in order to create a sequence that represents a left turn action at a crossroad, the leftmost branch has to be chosen (see highlighted path of figure 4.23). The relations to the lane that is composed of the ingoing (the branch the robot is located on) and outgoing branch are determined as described by equations 4.38 to 4.40, which is similar to how relations with respect to a single lane are captured.

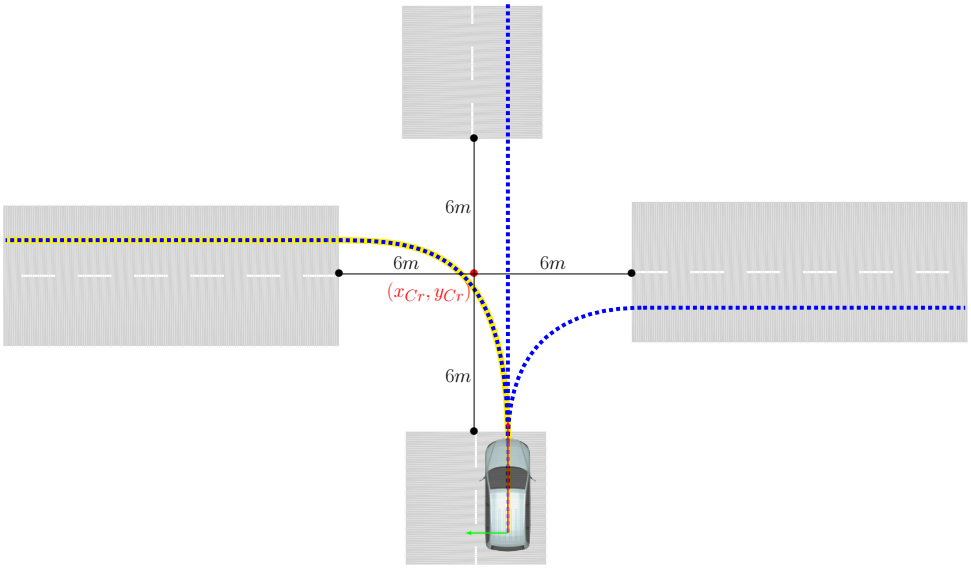


Figure 4.23:
Given center coordinate (x_{Cr}, y_{Cr}) a number of branches can be created taking into account their yaw angle, width, curvature and change in curvature.

4.6.2 The descriptive power of a relative representation

For each action, e.g. *"overtake"* or *"turn left"*, hundreds of sequences ${}^{O_j}\mathcal{R}_{O_i}$ between two objects O_i and O_j can be captured, each differing in length of the action, relative velocity or geometry of the road (e.g. curved, straight). To show the impact of varying velocities and lane geometries on the relative representation on the one hand and global representations on the other hand, two experiments were conducted.

The aim of the first experiment was to show the impact of different absolute robot velocities on the progress of an action that was either represented in relative or in global coordinates. To this end, an overtaking maneuver was repeatedly captured on a straight lane with different absolute robot velocities, but with similar relative velocities between the robot (overtaking vehicle) and the overtaken vehicle. Two global trajectories of those maneuvers are shown in figure 4.24, where the *red* trajectory was captured with a robot velocity of ca. 30 kph and the *blue* trajectory was captured with a robot velocity of ca. 40 kph. The driving direction is from left to right.

The trajectories differ from each other in length and shape, although they describe the same action. If the four key points, 1: steer to the left, 2: drive straight, 3: steer to the right and 4: drive straight, of an overtaking maneuver are considered,

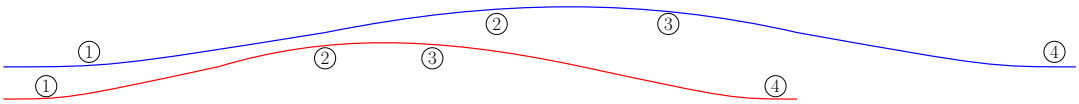


Figure 4.24:
Global trajectories of two overtaking maneuvers at different absolute speeds, but similar relative velocities (driving direction from left to right). The overtaking maneuver with lower absolute speed (*red*) shows a different course in comparison to the overtaking maneuver with higher speed (*blue*), which is highlighted by the diverging spatial occurrence of four key points of the maneuver.

both maneuvers in figure 4.24 show completely different spatial occurrences of the respective steps. In this context, figure 4.25 provides a visual representation of these dissimilarities. From the set of recorded trajectories the two most dissimilar trajectories (at 30kph *red* and at 60kph *magenta*) are chosen and the area (error) between them is computed (*green*).

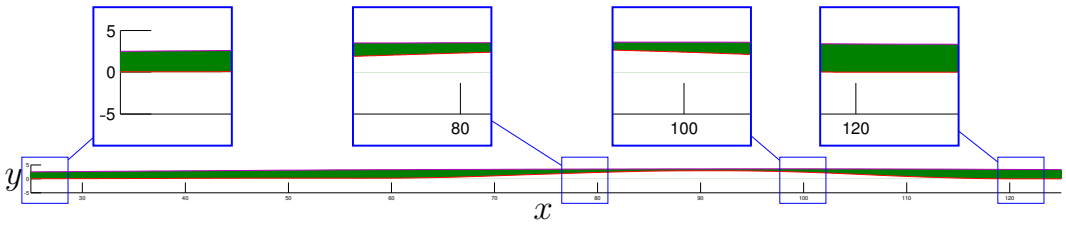


Figure 4.25:
Dissimilarity (*green*) between the two trajectories of figure 4.24. The magnification (*blue box*) of a closer perspective (*top*) provides a hint toward their numerical differences.

In order to compute the dissimilarity, a LM solver is applied first to minimize the squared Euclidean distance $\| \boldsymbol{p}_{\mathcal{T}_1} - \boldsymbol{p}_{\mathcal{T}_2} \|^2$ between closest points $\boldsymbol{p}_{\mathcal{T}_1}$ and $\boldsymbol{p}_{\mathcal{T}_2}$ of the trajectories $\mathcal{T}_1$ (*red*) and $\mathcal{T}_2$ (*magenta*) that represent the overtaking maneuvers of figure 4.25. I.e. both trajectories are moved closer together in order to minimize their spatial distance. Second, the space between them is integrated to get the error value. Each magnification of figure 4.25 shows the error at the key-points that were defined in figure 4.24. The error reaches its climax during the approach / avoidance behavior and when the robot moves towards the front of the overtaken vehicle, i.e. when highly precise steering is required.

In contrast, figure 4.26 shows the equivalent of these maneuvers represented by the relative description that is proposed in this thesis. The viewpoint of the visualized line sequences is the overtaken vehicle. I.e. each sequence shows how the overtaken vehicle would perceive an overtaking maneuver, with the start of the maneuver located at the left hand side of figure 4.26.

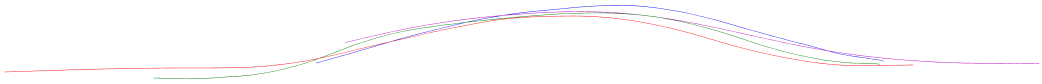


Figure 4.26:
The global trajectories of an overtaking maneuver are represented by means of relation sequences. Driving direction is from left to right.

Right away these sequences seem to be quite similar to each other, when compared to the sequences of figure 4.24. Upon applicaton of the dissimilarity measure that was used on the trajectories of figure 4.25 to the relative representation as shown in figure 4.27, only minor deviations are observable.

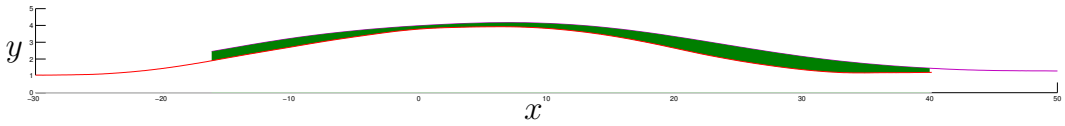


Figure 4.27:
The dissimilarity (*green*) between two of the sequences of figure 4.26 is shown.

Their similarity originates from the incorporation of relative motions as well as relative velocities. Since each overtaking maneuver was driven with almost identical relative velocities, they look quite similar from the perspective of an ego-centric observer. Global representations may add additional information where there is none.

A second experiment was conducted to investigate the effect of different relative velocities on sequences composed of relations. Furthermore, overtaking maneuvers on differently curved roads were investigated instead of utilizing spatial data gained from observing overtaking maneuvers on straight lanes. Figure 4.28 shows the geometric representation of the overtaking maneuvers at different speeds in the global Cartesian space. They were observed in a left bend, in a right bend and on a straight lane. Notably, each trajectory deforms depending on actual velocity and course of the road with virtually no possibility to deduce one trajectory based on another. Their dissimilarity increases during the traversal of the left and right turn.

Figure 4.29 shows the corresponding maneuvers represented by the relational description. In contrast to the trajectories in figure 4.28 the sequences resemble each other quite well. Their resemblance becomes clear, when the execution of an overtaking maneuver is considered from an ego-centric perspective. The overtaking vehicle starts at the rear of the vehicle to pass, then traverses to its left and completes the maneuver at the front of the overtaken vehicle. These steps are applicable in every

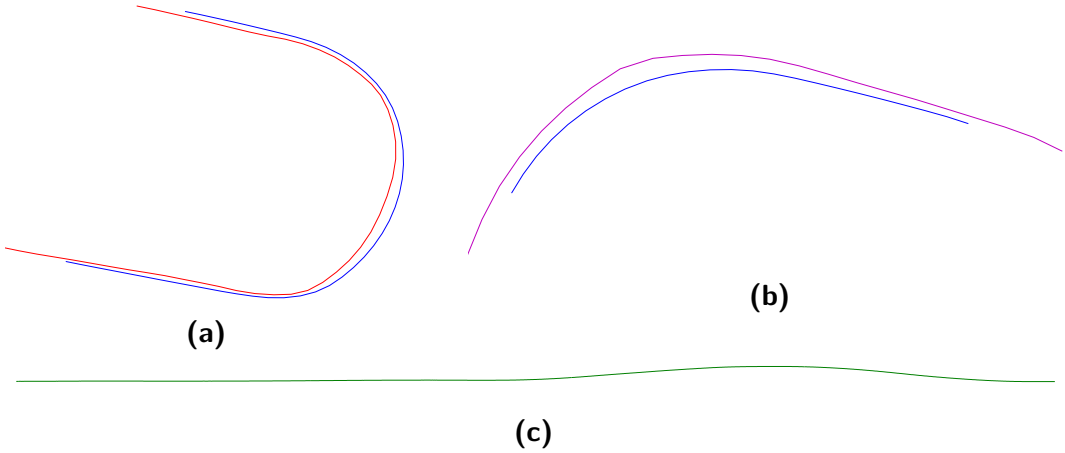


Figure 4.28:
Global trajectories of various overtaking maneuvers given different relative speeds and different curved roads. The blue trajectory in (a) and (b) represents the overtaken vehicle during a left bend and a right bend. In (c) an overtaking maneuver on a straight lane is represented.

situation, independent of the shape of the road and thus of the motion of the object being overtaken.

Figure 4.29:
The global trajectories of figure 4.28 from a relative perspective, with the overtaken vehicle being the observer. The colors are chosen in accordance to the colors of the trajectories of figure 4.28.

4.6.3 Maneuver identification

The previous section showed, how one maneuver, like e.g. *"overtaking"* can be represented by many trajectories. But while no meaningful connection between the trajectories can be found in the global space (see figure 4.28), a correlation between trajectories of one maneuver becomes obvious in an object relative representation (see figure 4.29). Hence, the aim of the following section is to describe, how a set of trajectories that describe one maneuver, like e.g. *"overtaking"* can be represented by one parametric description.

Figure 4.30 shows what the trajectories of figure 4.28 look like, when just the α-entity of a relation is considered. Upon closer examination of figure 4.30, the graph (*red*) that represents the overtaking maneuver of figure 4.28a (left bend) seems to be stretched and shifted with respect to the graph (*magenta*) that represents

the overtaking maneuver of figure 4.28b (right bend). A similar conclusion can be derived, when both graphs are compared to the one that describes the overtaking maneuver at a straight lane (*green*).

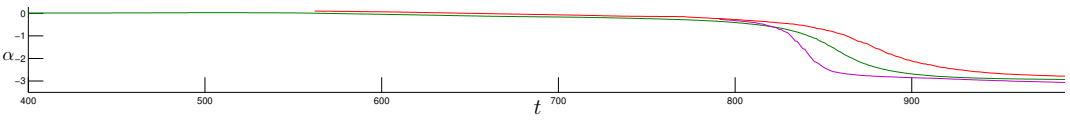


Figure 4.30:
The progress of the α-entity in the relation-based representation of the overtaking maneuvers of figure 4.28 with colors chosen accordingly. The *red* graph shows the progress of the α-entity at a relative velocity of ca. 10kph, the *green* graph at ca. 15 kph and the *magenta* graph at ca. 20 kph. The graphs are already shifted along the t-axis to increase visual clarity.

Thus it is important to check whether a mapping $f : {}^{O_2}\mathcal{R}_{O_1} \rightarrow {}^{O_2}\mathcal{R}'_{O_1}$ between sequences ${}^{O_2}\mathcal{R}_{O_1}(\mathfrak{r}(t_1), \mathfrak{r}(t_2), \ldots, \mathfrak{r}(t_n), v(t_1), v(t_2), \ldots, v(t_n))$ (e.g. the *red* trajectory of figure 4.29) and ${}^{O_2}\mathcal{R}'_{O_1}(\mathfrak{r}(t_1)', \mathfrak{r}(t_2)', \ldots, \mathfrak{r}(t_n)', v(t_1)', v(t_2)', \ldots, v(t_n)')$ (e.g. the magenta trajectory of figure 4.29) exists that transforms a sequence of relations ${}^{O_2}\mathcal{R}_{O_1}$ in a manner so that it fits a set of relations ${}^{O_2}\mathcal{R}'_{O_1}$. In figure 4.30 this would mean that the *magenta* graph, for example, is shifted and stretched, so that it resides exactly on the *red* graph. If such a mapping existed a more general representation of a maneuver could be derived that is able to express a whole set of sequences with as little information as possible.

In this context, the parametric sequence description $(o_{(\cdot)}, s_{(\cdot)}, a_{(\cdot)}, b_{(\cdot)})$ of section 4.5.2 is utilized to represent a sequence ${}^{O_2}\mathcal{R}_{O_1}(\mathfrak{r}(t_1), \ldots, \mathfrak{r}(t_n), v(t_1), v(t_2), \ldots, v(t_n))$ by means of a parametric representation ${}^{O_2}\mathcal{R}_{O_1}(o_{(\cdot)}, s_{(\cdot)}, a_{(\cdot)}, b_{(\cdot)}, v(t_1), v(t_2), \ldots, v(t_n))$, with $o_{(\cdot)}, s_{(\cdot)}, a_{(\cdot)}, b_{(\cdot)}$ being the approximated parameters of functions 4.22 to 4.24. Given the parametric representation of one sequence, a transformation of the parameters has to be determined that is able to explain the data of another sequence as well. It is assumed that $v(t_1) = v(t_2) = \cdots = v(t_n)$ to simplify matters with respect to the basic mapping procedure. This constraint will be dropped later, when the mapping procedure is applied to sequences that contain varying velocities. To simplify the explanation of the mapping procedure, we first concentrate on the α-entity of the mapping and then try to apply the same principles to the remaining entities.

When the graphs in figure 4.30 are examined, for instance, their vertical alignment seems to fit quite well, while they are misaligned horizontally. Parameters s_α and o_α of equation 4.22 are responsible for the vertical shape (stretch and shift) of the tangent function. I.e. when parameters s_α with s'_α and o_α with o'_α of the two

approximated sequences ${}^{O_2}\mathcal{R}_{O_1}$ and ${}^{O_2}\mathcal{R}'_{O_1}$ in figure 4.30 are compared, only small deviations are observed. In addition to the vertical transformation, parameters s_{t_α} and o_{t_α} are able to stretch and shift the tangent function horizontally. Thus they describe how the tangent function is transformed in order to represent the temporal progress, along the t-axis. After examination of figure 4.30 the horizontal alignment needs to be adapted.

To now align estimated parameters s_{t_α} with s'_{t_α} and o_{t_α} with o'_{t_α} of two approximated sequences ${}^{O_2}\mathcal{R}_{O_1}$ and ${}^{O_2}\mathcal{R}'_{O_1}$, the two parameters δo and δs are introduced. They describe the difference in temporal shift and stretch between the two sequences. The mapping f between two sequences with respect to the α-entity can now be described by:

$$f(\delta o, \delta s): \\ s_\alpha \cdot \tan^{-1}((s_{t_\alpha} + \delta s)(t + o_{t_\alpha} + \delta o)) + o_\alpha \rightarrow s'_\alpha \cdot \tan^{-1}(s'_{t_\alpha}(t + o'_{t_\alpha})) + o'_\alpha \tag{4.41}$$

By applying equation 4.41, the temporal differences between two sequences with respect to their α-entity can be described. In this context, figure 4.31 provides a visual representation, in which one data approximation (*red*) is first stretched (*green*) and then shifted to fit another data approximation (*blue*).

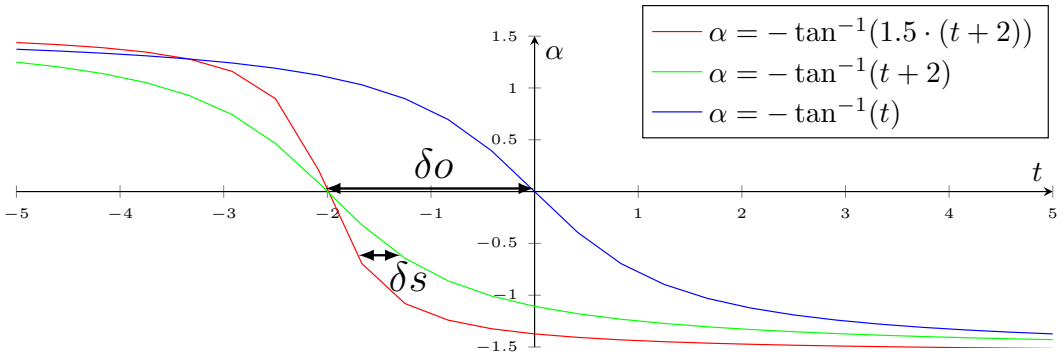


Figure 4.31:
One data approximation (*red*) is first stretched with parameter δs (*green*) and then shifted with parameter δo to fit another data approximation (*blue*).

Mapping f of equation 4.41 allows to project from any one sequence to another of the α-entity with respect to an overtaking maneuver. I.e. by taking mapping f into account it is possible to define one parametric representation that is able to represent many possible trajectories of an overtaking maneuver, such as the ones shown in figure 4.30.

For instance, the parametric description $(s_\alpha, o_\alpha, s_{t_\alpha}, o_{t_\alpha})$ of the *green* graph of figure 4.30 can be defined to be the parametric representation of the α-entity of an overtaking maneuver. Now, mapping f of equation 4.41 can be applied to transform the approximation of every other sequence $\mathcal{R}_k(\mathfrak{r}(t_1), \ldots, \mathfrak{r}(t_n))$, with $1 \leq k \leq n$, onto the parametric maneuver representation. I.e. for each sequence only two values (δo and δs) need to be stored, instead of a whole parametric sequence description $(s_\alpha, o_\alpha, s_{t_\alpha}, o_{t_\alpha})_k$. This results in n transformations $f(\delta o, \delta s)_k$, whose parameters δo_k and δs_k describe how the approximation of a sequence must be shifted and stretched to fit the parametric maneuver representation.

With hundreds or thousands of possible sequences per maneuver, a tremendous amount of mappings f needs to be stored. To reduce this huge amount of information, the impact of relative velocity $v(t)$, that is stored with each sequence $\mathcal{R}$, on the shape of the graph is examined. When the graphs of figure 4.30 are examined, for instance, it becomes clear that relative velocity $v(t)$ has a huge impact on the shearing factor s_{t_α} of an approximation. Under closer examination of figure 4.30 it seems that the higher the relative velocity, the steeper the graph.

I.e. with higher relative velocities the graph of a motion sequence is less stretched. However, the influence of the velocity is not linear. While low relative velocities already result in a relatively high shearing factor s_{t_α}, this effect is dampened for higher relative velocities, until a certain limit is reached. Hence, the natural logarithm is used to model the relationship between relative velocity $v(t)$ and shearing factor s_{t_α} according to equation 4.42.

$$g : \mathbb{R} \to \mathbb{R},\ g(v(t)) : v_1 \cdot \ln(v(t) + v_2) + v_3 \tag{4.42}$$

Parameters v_1 and v_3 are responsible for shearing and shifting the function horizontally, while parameter v_2 allows for a shift in the vertical direction. Given function $g(v(t))$ it is now possible to map from any relative velocity $v(t)$ to any shearing factor s_{t_α}. Effectively, shearing parameter s_{t_α} of the data approximation of equation 4.22 can be substituted to yield:

$$\alpha(t) = s_\alpha \cdot \tan^{-1}((v_1 \cdot \ln(v(t) + v_2) + v_3) \cdot (t + o_{t_\alpha})) + o_\alpha \tag{4.43}$$

Similar to parameters s_α, o_α and o_{t_α}, parameters v_1 to v_3 can be estimated by applying a LM-solver to find the best function fit, when recorded data points and

relative velocities of a relative motion are given. Furthermore, upon application of equation 4.43 the precondition $v(t_1) = v(t_2) = \cdots = v(t_n)$ that was introduced to simplify the approximation procedure can now be dropped. The condition was previously introduced, when only one parameter (s_{t_α}) was available to define the shearing of an approximation. With the introduction of equation 4.43 the shearing can now be adapted based on different relative velocities of a relative motion sequence.

With the above considerations, parameter δs of a mapping f can also be dropped, which further reduces the amount of information that needs to be stored per maneuver. When two sequences ${}^{O_2}\mathcal{R}_{O_1}$ and ${}^{O_2}\mathcal{R}'_{O_1}$ are given in their parametric description, parameter δs of function f that maps ${}^{O_2}\mathcal{R}_{O_1}$ onto ${}^{O_2}\mathcal{R}'_{O_1}$ is determined according to equation 4.44.

$$\delta s = s'_{t_\alpha} - s_{t_\alpha}, \text{ with } s'_{t_\alpha} \in {}^{O_2}\mathcal{R}'_{O_1} \text{ and } s_{t_\alpha} \in {}^{O_2}\mathcal{R}_{O_1} \tag{4.44}$$

Parameters s'_{t_α} and s_{t_α} can be substituted by equation 4.42 to yield:

$$\delta s = v_1 \cdot (\ln(v(t)' + v_2) + v_3 - (v_1 \cdot \ln(v(t) + v_2) + v_3) \tag{4.45}$$

$$\delta s = v_1 \cdot \ln\left(\frac{v(t)' + v_2}{v(t) + v_2}\right) \tag{4.46}$$

with $v(t)'$ and $v(t)$ being the relative velocities of the respective motion sequence. Finally, the initial mapping $f(\delta s, \delta o)$ of equation 4.41 can be rewritten, by substituting parameters s_{t_α}, s'_{t_α} and δs with the equations deduced above.

$$\begin{aligned} &f(\delta o) : s_\alpha \cdot \tan^{-1}((s_{t_\alpha} + \delta s)(t + o_{t_\alpha} + \delta o)) + o_\alpha \\ &\quad \rightarrow s'_\alpha \cdot \tan^{-1}((v_1 \cdot \ln(v(t)' + v_2) + v_3)(t + o'_{t_\alpha})) + o'_\alpha \\ &\text{with:} \\ &(s_{t_\alpha} + \delta s) = (v_1 \cdot \ln(v(t) + v_2) + v_3) + (v_1 \cdot \ln\left(\frac{v(t)' + v_2}{v(t) + v_2}\right)) \end{aligned} \tag{4.47}$$

If the mathematical statement $(s_{t_\alpha} + \delta s)$ is now simplified, according to equations 4.48 to 4.50 and then re-substituted into equation 4.47, mapping f reduces to the identity transformation with respect to parameter δs, since the shearing now depends

on relative velocities alone, without the need for a specific shearing parameter s_{t_α} per approximation.

$$(s_{t_\alpha} + \delta s) = v_1 \cdot (\ln(v(t) + v_2) + \ln\left(\frac{v(t)' + v_2}{v(t) + v_2}\right)) + v_3 \quad (4.48)$$

$$(s_{t_\alpha} + \delta s) = v_1 \cdot \ln((v(t) + v_2) \cdot \left(\frac{v(t)' + v_2}{v(t) + v_2}\right)) + v_3 \quad (4.49)$$

$$(s_{t_\alpha} + \delta s) = v_1 \cdot \ln(v(t)' + v_2) + v_3 \quad (4.50)$$

With parameter s_{t_α} replaced and parameter δs removed, the data approximation with respect to the α-entity is formulated according to equation 4.51 and mapping f is formulated according to equation 4.52.

$$\alpha(t) = s_\alpha \cdot \tan^{-1}((v_1 \cdot \ln(v(t) + v_2) + v_3) \cdot (t + o_{t_\alpha})) + o_\alpha \quad (4.51)$$

$$f(\delta o) : s_\alpha \cdot \tan^{-1}((v_1 \cdot \ln(v(t) + v_2) + v_3)(t + o_{t_\alpha} + \delta o)) + o_\alpha \quad (4.52)$$

$$\rightarrow s'_\alpha \cdot \tan^{-1}((v_1 \cdot \ln(v(t)' + v_2) + v_3)(t + o'_{t_\alpha})) + o'_\alpha$$

Figure 4.32 shows how the set of approximations in figure 4.30 is projected onto one approximation given equations 4.51 and 4.52. Each input data sequence is approximated according to equation 4.51 and then shifted (with the help of mapping $f(\delta o)$) along the t-axis, so that the center of the maneuver resides at $t = 0$.

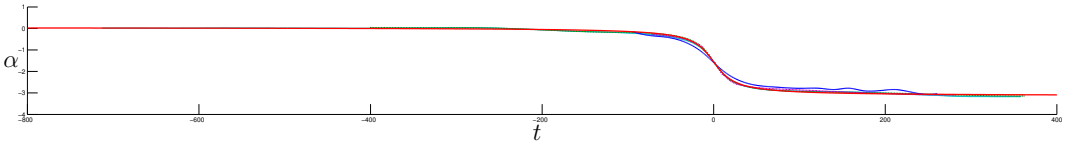


Figure 4.32:
Progress of the approximation of the α-entity (*red line-sequence*) that describes the set of all overtaking maneuvers of figure 4.30.

Similar to the approximation of the α-entity of a relation, equations 4.23 and 4.24 that represent the initial approximation functions of the β and d-entity of a relation can be reformulated to replace the shearing parameters. Thus, the complete approximation of a motion sequence can be formulated as shown in equations 4.54 to 4.56.

$$\mathcal{A}(\mathbf{Y_k}) = \Big(\alpha(\mathbf{Y_k})\ \beta(\mathbf{Y_k})\ d(\mathbf{Y_k})\Big)^T \text{ with:} \tag{4.53}$$

$$\alpha(t) = s_\alpha \cdot \tan^{-1}((v_{1_\alpha} \cdot \ln(v(t) + v_{2_\alpha}) + v_{3_\alpha}) \cdot (t + o_{t_\alpha})) + o_\alpha \tag{4.54}$$

$$\beta(t) = s_\beta \cdot \tan^{-1}((v_{1_\beta} \cdot \ln(v(t) + v_{2_\beta}) + v_{3_\beta}) \cdot (t + o_{t_\beta})) + o_\beta \tag{4.55}$$

$$\begin{aligned} d(t) = {} & a_0 + \sum_{n=1}^{2} a_n \cos(n(v_1 \cdot \ln(v(t) + v_2) + v_3) \cdot (t + o_t)) \\ & + \sum_{n=1}^{2} b_n \sin(n(v_1 \cdot \ln(v(t) + v_2) + v_3) \cdot (t + o_t)) \end{aligned} \tag{4.56}$$

Furthermore, mapping $f(\delta o)$ can be formalized, as shown in equations 4.57 to 4.59.

$$\begin{aligned} & s_\alpha \cdot \tan^{-1}((v_{1_\alpha} \cdot \ln(v(t) + v_{2_\alpha}) + v_{3_\alpha})(t + o_{t_\alpha} + \delta o)) + o_\alpha \\ & \rightarrow s'_\alpha \cdot \tan^{-1}((v_{1_\alpha} \cdot \ln(v(t)' + v_{2_\alpha}) + v_{3_\alpha})(t + o'_{t_\alpha})) + o'_\alpha \end{aligned} \tag{4.57}$$

$$\begin{aligned} & s_\beta \cdot \tan^{-1}((v_{1_\beta} \cdot \ln(v(t) + v_{2_\beta}) + v_{3_\beta})(t + o_{t_\beta} + \delta o)) + o_\beta \\ & \rightarrow s'_\beta \cdot \tan^{-1}((v_{1_\beta} \cdot \ln(v(t)' + v_{2_\beta}) + v_{3_\beta})(t + o'_{t_\beta})) + o'_\alpha \end{aligned} \tag{4.58}$$

$$\begin{aligned} & a_0 + \sum_{n=1}^{2} a_n \cos(n(v_1 \cdot \ln(v(t) + v_2) + v_3) \cdot (t + o_t + \delta o)) \\ & \quad + \sum_{n=1}^{2} b_n \sin(n(v_1 \cdot \ln(v(t) + v_2) + v_3) \cdot (t + o_{t+\delta o})) \\ & \rightarrow a'_0 + \sum_{n=1}^{2} a'_n \cos(n(v_1 \cdot \ln(v(t)' + v_2) + v_3) \cdot (t + o'_t + \delta o)) \\ & \quad + \sum_{n=1}^{2} b'_n \sin(n(v_1 \cdot \ln(v(t)' + v_2) + v_3) \cdot (t' + o_{t'} + \delta o)) \end{aligned} \tag{4.59}$$

In consequence, mapping $f(\delta o)$ jointly shifts the approximations $\alpha(t), \beta(t), d(t)$ of all entities of a relation along the t-axis, so that one approximation is able to explain all sequences of a maneuver. Notably, this temporal shift results from different data recording triggers, i.e. from the differing points in time when the capture of the sequence was started. Since the parametric description of a maneuver is not required to model these differences, parameter δo can be ignored after the completed approximation procedure.

Given approximation 4.53 and mapping $f(\delta o)$ it is now possible to describe a set of relative motions and their associated relative velocities, that represent one maneuver, like e.g. *"overtake"* or *"swerve"* by a simple set of functions. I.e. when the equations of 4.53, alongside estimated parameters $s_{(\cdot)}$, $o_{(\cdot)}$, $a_{(\cdot)}$, $b_{(\cdot)}$ and $v_{(\cdot)}$ are given, a valid relative motion sequence for any given velocity $v(t)$ at time t can be generated.

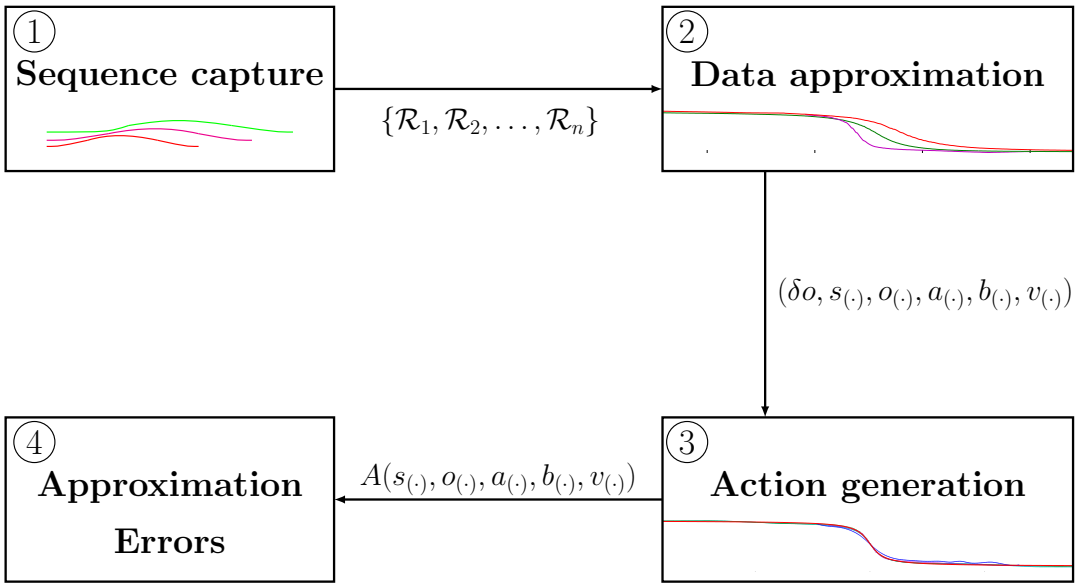


Figure 4.33:
Overview of the different steps of how to get from a set of sequences to a maneuver A.

The outline of the procedure following these considerations can by formulated as shown in figure 4.33. The procedure enables the computation of a maneuver representation, when a set of sequences $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n\}$ is given that describe the temporal progress of one maneuver, like e.g. *"overtake"* or *"swerve"*. One maneuver is captured multiple times (see figure 4.33, step 1) by evaluating the respective object percepts over time. This process can be repeated to gain multiple sequences for different maneuvers. The set of sequences per maneuver are then approximated given equations 4.54 to 4.56 and mapping $f(\delta o)$ by applying a LM-solver (see figure 4.33, step 2). During this estimation step all parameters are jointly estimated over all sequences, except δo. Parameter δo of mapping f is estimated per sequence, since it represents the sequence specific temporal shift. After the approximation procedure of all sequences is completed, the sequence specific temporal shift δo can be neglected (see figure 4.33, step 3). The final approximation of a maneuver A is then used to determine the inherent approximation errors (see figure 4.33, step 4), which is described in the next section.

4.6.4 Approximation errors and measurement uncertainties

The maneuver representation encodes the progress of a maneuver, but does not take approximation errors into account or the measurement noise that is inherent to the data provided by the various sensors of the robot. Hence, sequences generated from such maneuver approximations may represent the progress of a maneuver inadequately. To account for possible approximation errors as well as measurement uncertainties an error expression accompanies the maneuver representation. It captures possible approximation errors and the remaining random noise. A given maneuver approximation $A(s_{(\cdot)}, o_{(\cdot)}, a_{(\cdot)}, b_{(\cdot)}, v_{(\cdot)})$ (see equation 4.53) is used to determine sample covariance $\mathbf{P}_A(t)$ with respect to time step t. In principle, the approximation of a maneuver is compared to the recorded sequences the approximation is based upon, for each point in time t. But since the samples of different sequences are not equally distributed, i.e. a sample of sequence $\mathcal{R}$ at time t may not have a corresponding sample at sequence $\mathcal{R}'$ and vice versa, the time axis is subdivided into n equidistant intervals. All residuals falling into one of these intervals are utilized to determine the error covariance per interval. Figure 4.34 provides an overview of the procedure.

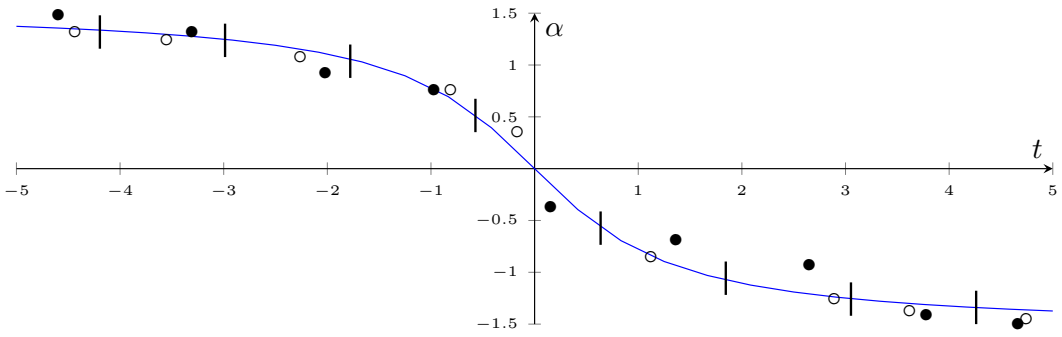


Figure 4.34:
The approximation of a maneuver (*blue*) is subdivided into n parts, so that at least two samples of different data sequences (*filled*, *unfilled* points) are contained in one such interval.

The algorithm succeeds to the steps of figure 4.33 and is fed with the maneuver representation $A(s_{(\cdot)}, o_{(\cdot)}, a_{(\cdot)}, b_{(\cdot)}, v_{(\cdot)})$ and the individual sequences. For each interval (exemplary marked by the black strokes in figure 4.34) the residuals (difference between approximation and measured data points) of the data that falls into the respective interval are determined. Given the set of residuals, one sample covariance

can be determined per interval. The set $\Omega_{\mathbf{P}_A}$, that contains the error covariances of each interval is then added to a maneuver description, as shown in equation 4.60.

$$A = ((s_{(\cdot)}, o_{(\cdot)}, a_{(\cdot)}, b_{(\cdot)}, v_{(\cdot)}), \Omega_{\mathbf{P}_A}) \tag{4.60}$$

I.e. a given maneuver description A contains a set of error covariance matrices, that describe the model uncertainty in equidistantly distributed regions in addition to its parametric representation. A maneuver description, such as "overtake" or "follow", can now be utilized to either predict the motion of other objects or to steer the robot. The approaches with respect to object prediction and motion planning are described in the following sections.

4.6.5 Motion prediction

The approach of object-related navigation models and predicts the behavior of objects with respect to one another. I.e. objects are coupled to one another: vehicles are bound to roads, to crossroads or even to other vehicles (for instance in the cases of overtaking, swerve or following maneuvers), while cyclists or pedestrians are bound to their respective pathways, for example. Hence, relative object-object motion prediction is required. Instead of evaluating objects according to their global motion, which is defined by their speed, direction and potentially by their change in direction as proposed by Hermes et al. [2009] or Large et al. [2004], their relative motion needs to be evaluated.

Relating the motion of objects to one another was already introduced by e.g. Lefevre et al. [2012], Foka and Trahanias [2010], Meyer-Delius et al. [2009] or Firl et al. [2012], to name just a few. The work of Lefevre et al. [2012] focused on risk assessment at road junctions. Dangerous situations were detected by evaluating conflicts between what drivers intend to do and what is expected of them. Each participating vehicle was equipped with V2V wireless communication devices that communicated position, orientation and speed to jointly estimate the most probable maneuver for each vehicle and a possible collision risk in a dynamic bayesian network (DBN). The DBN models the interaction between each vehicle as well as the interaction between vehicle and road on a maneuver level. Maneuvers such as "stop" or "give way" were modelled geometrically by an exemplary path. The DBN estimated the probability of a vehicle performing a certain maneuver, which is called

"intention". The expected behavior is modelled by jointly evaluating "intention" and current percepts. I.e. if the current percept does not match the intended behavior the collision risk increased. In contrast to other works, a collision risk was provided without prediction of the possible collision site.

To navigate in uncertain environments, the work by Foka and Trahanias [2010] incorporated information on possible future robot positions, static obstacle structures and possible future obstacle configurations into a hierarchical paritially observable Markov decision process (POMDP). To enable the estimation of future obstacle configurations, a method that allowed for long-term dynamic object prediction was implemented. The methodology connected the observed path of a dynamic object to a set of destination positions, called "hot points", that a human might intend to reach. Either a pre-defined or a trained map of "hot points" was input to a prediction procedure, that estimated the probability of a certain "hot point" being the true destination of an obstacle, based on the observed obstacle path. By means of the estimate, a trajectory from the last observed obstacle position to the estimated destination was computed to allow for the prediction of future obstacle states.

Instead of just utilizing pre-defining models, Meyer-Delius et al. [2009] as well as Firl et al. [2012] trained a number of Hidden Markov Models (HMM's), each representing a certain maneuver, such as "overtaking", "following", etc. Similar to the methodology of this thesis, their approach incorporated egocentric, relative object data like relative bearing, distance or relative velocity into the estimation and prediction of object states. During actual maneuvers, observations of current relative behaviors were fed to each model to determine the most fitting one. Due to the generative nature of HMM's object prediction became possible by evaluating the most probable model with respect to future time steps.

The need to exploit object-object behavior or at least the need to evaluate object intentions to allow for more robust object predictions was clearly recognized by all previously mentioned authors. Human navigation behaviour is indeed characterized by clinging to continuous objects such as lanes, curbs, bushes, or walls. In a building, for example, humans typically follow corridors towards their destination, instead of a thoroughly planned trajectory. In the same fashion the methodology of object-related navigation aims at more robust object motion predictions taking into account object-object relations. For instance, the motion of a vehicle is commonly influenced by the course of the road. While it is difficult to deduce future object states from its current motion alone, since an upcoming bend of the road may alter its motion

behavior, the course of the road may provide viable information with respect to possible future object motions, even at large distances.

Furthermore, relating objects with relative motion behavior will proof to be favorable with respect to object predictions in real world scenarios. Sensor measurements are typically error-prone and thus provide noisy object data, especially in case of objects residing far away from the sensor. While the sensor gets closer to the object in question, measurement uncertainty decreases with more accurate object information, such as center coordinate, length, width, etc. Predicting object states at great distances without a link to the environment thus often results in fluctuating object positions, because large uncertainties dominate the prediction step. In contrast, by accounting for the relative motion between objects, more accurate predictions are rendered possible, as the state of an object is updated according to the state of another object with smaller uncertainties.

To provide proof for this statement, a simple scenario with two objects moving side by side towards the observer is considered. In a first attempt the objects' states are estimated independently of one another to show the impact on the object uncertainties. In a second attempt it is shown how the object uncertainties develop when the object states are linked. Both attempts are formulated with the help of a KF, since it allows assumptions regarding the development of uncertainties during the respective steps. We are interested in the development of the uncertainties of the respective object state only, hence a complete description of the KF is omitted (see e.g. Bishop and Welch [2001] for an introduction). Equations 4.61 to 4.64 show how prediction covariance $\mathbf{P}^*(k)$, innovation covariance $\mathcal{S}(k)$, Kalman gain $\mathbf{K}(k)$ and estimation covariance $\hat{\mathbf{P}}(k)$ are determined at time step k with respect to the linear KF.

$$\mathbf{P}^*(k) = \mathbf{\Phi}(k) \cdot \hat{\mathbf{P}}(k-1) \cdot \mathbf{\Phi}^T(k) + \mathbf{Q}(k) \tag{4.61}$$

$$\mathcal{S}(k) = \mathcal{C}(k) \cdot \mathbf{P}^*(k) \cdot \mathcal{C}^T(k) + \mathbf{R}(k) \tag{4.62}$$

$$\mathbf{K}(k) = \mathbf{P}^*(k) \cdot \mathcal{C}^T(k) \cdot \mathcal{S}^{-1}(k) \tag{4.63}$$

$$\hat{\mathbf{P}}(k) = \mathbf{P}^*(k) - \mathbf{K}(k) \cdot \mathcal{S}(k) \cdot \mathbf{K}^T(k) \tag{4.64}$$

In this context $\mathbf{\Phi}(k)$ and $\mathcal{C}(k)$ represent the state transition and observation matrices, while $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ represent the process and measurement noise covariance matrices. State transition and observation matrix may be substituted by their Jacobians in case of the EKF without any impact on further mathematical statements. To simplify matters the states of two objects of equal type are estimated in order to omit possibly challenging dependencies between state variables, e.g. between state variables modelled in a Cartesian and others modelled in a cylindrical coordinate frame.

For the first attempt, both object states are estimated in one KF with uncorrelated object measurements. Transition and observation matrices as well as the complete covariance $\mathbf{P}$ are thus defined by:

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_{O_1} & \mathbf{\Phi}_{O_1,O_2} \\ \mathbf{\Phi}^T_{O_1,O_2} & \mathbf{\Phi}_{O_2} \end{bmatrix} \quad \mathcal{C} = \begin{bmatrix} \mathcal{C}_{O_1} & 0 \\ 0 & \mathcal{C}_{O_2} \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{O_1} & \mathbf{P}_{O_1,O_2} \\ \mathbf{P}^T_{O_1,O_2} & \mathbf{P}_{O_2} \end{bmatrix}$$

Further, the uncertainties attached to the initial states are defined with $\mathbf{P}_{O_1}(0) < \mathbf{P}_{O_2}(0)$, either because object O_1 is larger than object O_2 and thus provides more accurate position estimates, or because object O_1 is closer to the sensor than object O_2.

Since each object state is estimated in an independent fashion, we are able to further define $\mathbf{\Phi}_{O_1,O_2} = 0$ and $\mathbf{\Phi}^T_{O_1,O_2} = 0$. Likewise, process and measurement noise matrices $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ contain zeros in their upper right and lower left blocks. If we now follow the development of the uncertainties as described by equations 4.61 to 4.64 we get:

$$\mathbf{P}^*(k) = \begin{bmatrix} \mathbf{\Phi}_{O_1}(k) \cdot \hat{\mathbf{P}}_{O_1}(k-1) \cdot \mathbf{\Phi}^T_{O_1}(k) + \mathbf{Q}_{O_1}(k) & 0 \\ 0 & \mathbf{\Phi}_{O_2}(k) \cdot \hat{\mathbf{P}}_{O_2}(k-1) \cdot \mathbf{\Phi}^T_{O_2}(k) + \mathbf{Q}_{O_2}(k) \end{bmatrix}$$

...

$$\hat{\mathbf{P}}(k) = \begin{bmatrix} \mathbf{P^*}_{O_1}(k) - \mathbf{K}_{O_1}(k) \cdot \mathcal{S}_{O_1}(k) \cdot \mathbf{K}_{O_1}^T(k) & 0 \\ 0 & \mathbf{P^*}_{O_2}(k) - \mathbf{K}_{O_2}(k) \cdot \mathcal{S}_{O_2}(k) \cdot \mathbf{K}_{O_2}^T(k) \end{bmatrix}$$

Since the objects are of equal type, i.e. $\mathcal{C}_{O_1} = \mathcal{C}_{O_2}$ and move in the same fashion, their overall motion can be considered to be equal , i.e. $\mathbf{\Phi}_{O_1} \approx \mathbf{\Phi}_{O_2}$. As a result, the uncertainty $\hat{\mathbf{P}}_{O_2}$ of object O_2 never decreases to a value equal $\hat{\mathbf{P}}_{O_1}$ even after $k \to \infty$ iterations.

However, when some kind of model between the two objects is assumed, the uncertainty of one object can be updated according to the uncertainty of the other one. This reduces the larger uncertainty. We thus define the relation $\mathbf{r}_{O_1,O_2}$ between the two objects O_1 and O_2 by:

$$\mathbf{r}_{O_1,O_2} = \mathcal{R}_{O_1,O_2} \cdot \boldsymbol{x}_{O_1,O_2}$$

with $\boldsymbol{x}_{O_1,O_2}$ being the joint object state to estimate and $\mathcal{R}_{O_1,O_2}$ a matrix to map state variables to relations. Notably, $\mathbf{r}_{O_1,O_2}$ is not required to satisfy definition 1 of a relation as defined in this thesis. It just serves as a representation of the connection between the two objects and may also be represented by means of Cartesian coordinates, distances, etc. Moreover, we are able to define the covariance $\mathbf{P}_{\mathbf{r}_{O_1,O_2}}$ of $\mathbf{r}_{O_1,O_2}$ by:

$$\mathbf{P}_{\mathbf{r}_{O_1,O_2}} = \mathcal{R}_{O_1,O_2} \cdot \mathbf{P} \cdot \mathcal{R}_{O_1,O_2}^T = \mathbf{P}_{O_1} + \mathbf{P}_{O_2} - \mathbf{P}_{O_1,O_2} - \mathbf{P}_{O_1,O_2}^T \tag{4.65}$$

When we assume a relative model between objects O_1 and O_2, the relation $\mathbf{r}_{O_1,O_2}$ between both objects can be estimated with complete certainty. I.e. covariance $\mathbf{P}_{\mathbf{r}_{O_1,O_2}}$ converges against zero. To fulfil $\mathbf{P}_{\mathbf{r}_{O_1,O_2}} = 0$ each covariance from 4.65 is required to be equal, i.e.

$$\mathbf{P}_{O_1} = \mathbf{P}_{O_2} = \mathbf{P}_{O_1,O_2} = \mathbf{P}_{O_1,O_2}^T \tag{4.66}$$

Of course the uncertainties themselves do not converge against zero, but the uncertainties of related objects converge against each other. We now apply 4.66

to our example. As each object moves in the same fashion, prediction covariances ${\mathbf{P}^*}_{O_1}$ and ${\mathbf{P}^*}_{O_2}$ increase by similar magnitudes. This results in

$$\det {\mathbf{P}^*}_{O_1} \leq \det {\mathbf{P}^*}_{O_2}. \tag{4.67}$$

A KF is initialized with a positive semidefinite (psd) state covariance matrix $\mathbf{P}(0)$. Matrices $\mathbf{Q}(k)$ as well as $\mathbf{R}(k)$ are both psd as are matrices $\mathbf{P}^*(k)$, $\mathcal{S}(k)$, $\mathbf{K}(k) \cdot \mathcal{S}(k) \cdot \mathbf{K}^T(k)$ and $\hat{\mathbf{P}}(k)$ according to equations 4.61 to 4.64. Thus the determinant of the estimation covariance, which provides a measure for the area covered by a matrix, is smaller or equal to the determinant of the prediction covariance.

$$\det \hat{\mathbf{P}}(k) = \det[\mathbf{P}^*(k) - \mathbf{K}(k) \cdot \mathcal{S}(k) \cdot \mathbf{K}^T(k)] \leq \det \mathbf{P}^*(k) \tag{4.68}$$

In order to fulfil 4.66, 4.67 and 4.68 the larger object uncertainty has to converge towards the lower one and vice versa, i.e. $\mathbf{P}_{O_2}$ is required to converge against $\mathbf{P}_{O_1}$, effectively decreasing the uncertainty of object O_2. With $\mathbf{P}_{O_2}$ fixed and $\mathbf{P}_{O_1}$ converging against $\mathbf{P}_{O_2}$, $\det \hat{\mathbf{P}} > \det \mathbf{P}^*$, which contradicts with 4.68.

Hence, by linking the estimate of an object to its environment, more accurate estimates are achievable. E.g. relating the motion of a vehicle to a previously known course of a lane may improve the accuracy of the vehicle estimate, while the estimation of a previously unknown road may be enriched by the estimate of other traffic participants.

To further exemplify the benefits of an object-related motion prediction, the initial scenario as presented in figure 4.3 is utilized, but it is now enriched by temporal information as shown in 4.35.

Typically, a recursive Bayesian estimator is set up to incorporate measurable object information into the process of determining future object positions, as shown in figure 4.35. In order to introduce the notation, but first and foremost to provide a mathematical comparison between object prediction based on global trajectories and object-related prediction, each prediction step is formalized. Equation 4.69 outlines

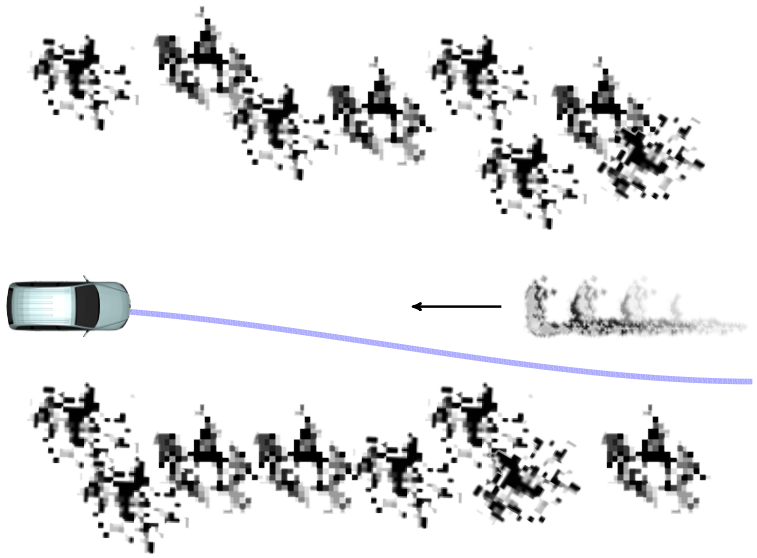


Figure 4.35:
Traffic scenario of figure 4.3 enriched by temporal information. The dynamic object's motion in the map is estimated (*black arrow*) over subsequent measurements, which enables motion prediction in relation to the map.

the basic structure of a recursive Bayesian estimator which enables multi-step forward prediction based on current object information.

$$p(\boldsymbol{x}^*_{V_{k+n}} \mid \mathbf{Y}_{V_k}) = p(\boldsymbol{x}^*_{V_{k+n}} \mid \hat{\boldsymbol{x}}_{V_k}) \cdot p(\hat{\boldsymbol{x}}_{V_k} \mid \mathbf{Y}_{V_k}) \tag{4.69}$$

In this context, corrected state $\hat{\boldsymbol{x}}_{V_k}$ is estimated by utilizing k measurements $\mathbf{Y}_{V_k}$ of vehicle V_k. These measurements may – amongst others – include global position, vehicle velocity, yaw angle and yaw rate. Referring to the example in figure 4.35, each measurement $\boldsymbol{y}_{V_k}$ of the vehicle may include constant yaw angles and yaw rates close to zero, since the dynamic object moves straight towards the robot (*white vehicle*). State $\hat{\boldsymbol{x}}_{V_k}$ can be estimated, given k measurements $\mathbf{Y}_{V_k}$ by conditioning the rightmost term of 4.69 on previous measurements $\mathbf{Y}_{V_{k-1}}$ and states $\hat{\boldsymbol{x}}_{V_{k-1}}$. The full derivation of probabilities $p(\hat{\boldsymbol{x}}_{V_k} \mid \mathbf{Y}_{V_k})$ and $p(\boldsymbol{x}^*_{V_{k+n}} \mid \mathbf{Y}_{V_k})$ is omitted here but thoroughly explained in appendix B.

The model $p(\hat{\boldsymbol{x}}_{V_k} \mid \mathbf{Y}_{V_k})$, which is estimated at step k can be utilized to predict state $\hat{\boldsymbol{x}}_{V_k}$ n steps in the future. Thus, by applying the information of estimated state $\hat{\boldsymbol{x}}_{V_k}$ n times we get probability $p(\boldsymbol{x}^*_{V_{k+n}} \mid \hat{\boldsymbol{x}}_{V_k})$ of being in state $\boldsymbol{x}^*_{V_{k+n}}$ after n steps. When the estimated information of state $\hat{\boldsymbol{x}}_{V_k}$ is applied n times to the example in figure 4.35, this may result in no or only slight changes of steering angle

and steering rate. I.e. since a straight object motion was observed at time step k, the estimator may come to the conclusion that the object proceeds along this way further. To obtain the measurement estimate ${\boldsymbol{y}^*}_{V_{k+n}}$, observation probability $p({\boldsymbol{y}^*}_{V_{k+n}} \mid {\boldsymbol{x}^*}_{V_{k+n}})$ has to be combined with the state prediction of equation 4.69 to yield:

$$p({\boldsymbol{y}^*}_{V_{k+n}} \mid \mathbf{Y}_{V_k}) = p({\boldsymbol{y}^*}_{V_{k+n}} \mid {\boldsymbol{x}^*}_{V_{k+n}}) p({\boldsymbol{x}^*}_{V_{k+n}} \mid \mathbf{Y}_{V_k}) \tag{4.70}$$

Equation 4.70 constitutes the most probable measurement estimate, when all previous measurements and predictions are given. However, applying equation 4.70 to our example, i.e. only information that is associated with the object in question is taken into account (*simplified to the black arrow in figure 4.35*), may result in an estimated object position in the immediate pathway of the robot. This may trigger further reasoning steps in order to prevent a possible collision, which is depicted in figure 4.35 by the curve originating from the robot. Thus utilizing the motion of a dynamic object in a global world frame as the sole reference to predict its motion may provide falsified predicted information, which may trigger dispensable processing steps.

Current research in the area of long term motion prediction diverges in its implementation of prediction $p({\boldsymbol{x}^*}_{V_{k+n}} \mid \hat{\boldsymbol{x}}_{V_k})$ and estimation $p(\hat{\boldsymbol{x}}_{V_k} \mid \mathbf{Y}_{V_k})$. Hermes et al. [2009], for example, captured different vehicle trajectories at crossroads. While the estimation step descended a binary tree that linked to a trajectory database to get the trajectories that best explain actual perception, the prediction step utilized the spatial information of these trajectories to predict future object states. In a similar fashion Large et al. [2004] captured vehicle trajectories to model different vehicle motions. In contrast to Hermes et al. [2009] these trajectories were clustered to yield different vehicle maneuvers. Each maneuver was subsequently described by a representative trajectory. Via the estimation process, the likelihood of a partially observed trajectory belonging to one of the clusters was afterwards determined by evaluating the dissimilarity between the observed trajectory and each representative. The prediction step then utilized the most probable cluster to forecast possible future object positions.

However, object maneuvers are often connected to some environmental structures. In fact, most maneuvers that are performed by an object are dictated by its environment. For instance, roads are meant to be followed, while other objects may require some kind of evasive action. In the example in figure 4.36 the courses of both the robot and

the dynamic object are influenced by the shape of the road. Instead of referencing the motion of an object to itself only, the influence of inter-object relations with respect to object motion prediction is therefore examined to gain more precise and more robust estimates. For this purpose an object-based representation, e.g. the example in figure 4.3b, can be enhanced by some temporal information (as shown in figure 4.36).

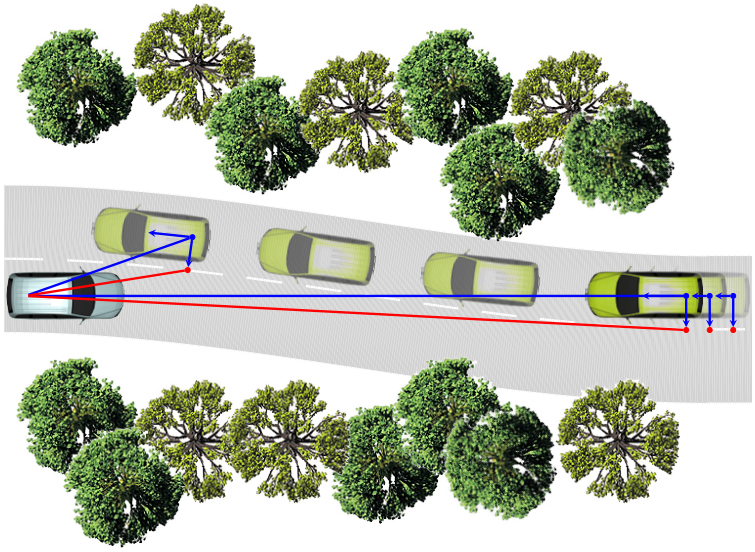


Figure 4.36:
Traffic scenario of figure 4.3b enriched by temporal information. Instead of estimating a vehicle's motion in the global coordinate frame without a relation to the real world, its dynamic behavior is estimated in relation to the lane.

To incorporate the relative behavior between two objects into the prediction of their relative motion n steps into the future, equation 4.69 is modified to include a second set of measurements $\mathbf{Y}_{L_{k+n}}$ into the estimation procedure. In order to represent the example in figure 4.36 adequately, each variable from equation 4.71 is indexed in a consistent manner. Sets $\mathbf{Y}_{V_k}$ and $\mathbf{Y}_{L_{k+n}}$ provide vehicle and lane measurements at steps k and $k+n$, respectively. ${\boldsymbol{x}^*}_{V_{k+n}}$ represents the predicted relative state at time step $k+n$ between lane and vehicle. In this example, we assume that the lane is measured up to step $k+n$ as well, either via existing digital road data or via a perception module that is able to perceive the course of the road. To evaluate probability $p({\boldsymbol{x}^*}_{V_{k+n}} \mid \mathbf{Y}_{V_k}, \mathbf{Y}_{L_{k+n}})$ of observing predicted state ${\boldsymbol{x}^*}_{V_{k+n}}$, when measurements $\mathbf{Y}_{V_k}$ and $\mathbf{Y}_{L_{k+n}}$ are given, Bayes rule is applied.

$$p({\boldsymbol{x}^*}_{V_{k+n}} \mid \mathbf{Y}_{V_k}, \mathbf{Y}_{L_{k+n}}) = \frac{p({\boldsymbol{x}^*}_{V_{k+n}}, \mathbf{Y}_{L_{(k+1):n}} \mid \mathbf{Y}_{L_k}, \mathbf{Y}_{V_k})}{p(\mathbf{Y}_{L_{(k+1):n}} \mid \mathbf{Y}_{L_k}, \mathbf{Y}_{V_k})} \tag{4.71}$$

Equation 4.71 decomposes the set of lane measurements $\mathbf{Y}_{L_{k+n}}$ to obtain the set of measurements $\mathbf{Y}_{L_k}$ up to time step k as well as the set of measurements $\mathbf{Y}_{L_{(k+1):n}}$ ranging from $k+1$ to n. In order to incorporate estimated state $\hat{\boldsymbol{x}}_{V_k}$, the nominator is dependent on $\hat{\boldsymbol{x}}_{V_k}$. Furthermore, the denominator can be simplified, as the set of future measurements $\mathbf{Y}_{L_{(k+1):n}}$ is conditionally independent of the sets of past measurements $\mathbf{Y}_{L_k}$ and $\mathbf{Y}_{V_k}$. In fact all measurements are conditionally independent from one another, because one cannot deduce one measurement from just another given measurement without the help of a model, i.e. $p(\mathbf{Y}_{L_{(k+1):n}} \mid \mathbf{Y}_{L_k}, \mathbf{Y}_{V_k}) = p(\mathbf{Y}_{L_{(k+1):n}})$.

$$p({\boldsymbol{x}^*}_{V_{k+n}} \mid \mathbf{Y}_{V_k}, \mathbf{Y}_{L_{k+n}}) = \frac{p({\boldsymbol{x}^*}_{V_{k+n}}, \mathbf{Y}_{L_{(k+1):n}} \mid \hat{\boldsymbol{x}}_{V_k}, \mathbf{Y}_{L_k}, \mathbf{Y}_{V_k}) \cdot p(\hat{\boldsymbol{x}}_{V_k} \mid \mathbf{Y}_{L_k}, \mathbf{Y}_{V_k})}{p(\mathbf{Y}_{L_{(k+1):n}})} \tag{4.72}$$

The leftmost term of the nominator is expanded using the definition of conditional probability and incorporating the fact that predicted state ${\boldsymbol{x}^*}_{V_{k+n}}$ and measurements $\mathbf{Y}_{L_{(k+1):n}}$ are independent of past measurements $\mathbf{Y}_{L_k}$ and $\mathbf{Y}_{V_k}$, i.e. $p({\boldsymbol{x}^*}_{V_{k+n}}, \mathbf{Y}_{L_{(k+1):n}} \mid \hat{\boldsymbol{x}}_{V_k}, \mathbf{Y}_{L_k}, \mathbf{Y}_{V_k}) = p({\boldsymbol{x}^*}_{V_{k+n}}, \mathbf{Y}_{L_{(k+1):n}} \mid \hat{\boldsymbol{x}}_{V_k})$.

$$p({\boldsymbol{x}^*}_{V_{k+n}}, \mathbf{Y}_{L_{(k+1):n}} \mid \hat{\boldsymbol{x}}_{V_k}) = p({\boldsymbol{x}^*}_{V_{k+n}} \mid \mathbf{Y}_{L_{(k+1):n}}, \hat{\boldsymbol{x}}_{V_k}) \cdot p(\mathbf{Y}_{L_{(k+1):n}} \mid \hat{\boldsymbol{x}}_{V_k}) \tag{4.73}$$

Then equation 4.73 is re-substituted into 4.72, considering that measurements $\mathbf{Y}_{L_{(k+1):n}}$ ranging from $k+1$ to n are conditionally independent of state $\hat{\boldsymbol{x}}_{V_k}$, i.e. $p(\mathbf{Y}_{L_{(k+1):n}} \mid \hat{\boldsymbol{x}}_{V_k}) = p(\mathbf{Y}_{L_{(k+1):n}})$.

$$p({\boldsymbol{x}^*}_{V_{k+n}} \mid \mathbf{Y}_{V_k}, \mathbf{Y}_{L_{k+n}}) = p({\boldsymbol{x}^*}_{V_{k+n}} \mid \mathbf{Y}_{L_{(k+1):n}}, \hat{\boldsymbol{x}}_{V_k}) \cdot p(\hat{\boldsymbol{x}}_{V_k} \mid \mathbf{Y}_{L_k}, \mathbf{Y}_{V_k}) \tag{4.74}$$

Estimating state $\hat{\boldsymbol{x}}_{V_k}$ now depends on the measurements of both lane $\mathbf{Y}_{L_k}$ and object $\mathbf{Y}_{V_k}$ up to time-step k, while state prediction ${\boldsymbol{x}^*}_{V_{k+n}}$ is now dependent on state $\hat{\boldsymbol{x}}_{V_k}$ as well as on the measurements of the lane $\mathbf{Y}_{L_{(k+1):n}}$ up to time-step n. In a similar fashion to B.2 state $\hat{\boldsymbol{x}}_{V_k}$ can be estimated, as shown in equation 4.75.

But instead of relating the object state estimate on past object measurements only, the relation to a measured lane is incorporated into the estimation process also.

$$
\begin{aligned}
&p(\hat{\boldsymbol{x}}_{V_k} \mid \mathbf{Y}_{L_k}, \mathbf{Y}_{V_k}) = \\
&\quad \eta \cdot p(\boldsymbol{y}_{V_k}, \boldsymbol{y}_{L_k} \mid \hat{\boldsymbol{x}}_{V_k}) \\
&\quad \cdot \int p(\hat{\boldsymbol{x}}_{V_k} \mid \hat{\boldsymbol{x}}_{V_{k-1}}) \cdot p(\hat{\boldsymbol{x}}_{V_{k-1}} \mid \mathbf{Y}_{L_{k-1}}, \mathbf{Y}_{V_{k-1}}) d\, \hat{\boldsymbol{x}}_{V_{k-1}}
\end{aligned}
\tag{4.75}
$$

By now referring to the example in figure 4.36, state $\hat{\boldsymbol{x}}_{V_k}$, which may include object specific parameters such as steering rate and steering angle as well as parameters that represent connections to other objects in the environment, is estimated by evaluating the relation between measurements $\mathbf{Y}_{L_k}$ (*red arrows in figure 4.36*) and $\mathbf{Y}_{V_k}$ (*blue arrows in figure 4.36*). I.e. an estimator is now enabled to derive conclusions about the connection between object behavior and the course of the lane. Upon integration of state estimation (equation 4.75) into state prediction (equation 4.74), the thus gained knowledge about relative object behavior in form of state $\hat{\boldsymbol{x}}_{V_k}$ becomes available for further object prediction steps. In contrast to the state estimate based on object measurements only (B.3), state prediction $p(\boldsymbol{x}^*{}_{V_{k+n}} \mid \mathbf{Y}_{L_{(k+1):n}}, \hat{\boldsymbol{x}}_{V_k})$ is now influenced by the relative behavior modelled in $\hat{\boldsymbol{x}}_{V_k}$ as well as measurements $\mathbf{Y}_{L_{(k+1):n}}$ that describe the course of the lane. I.e. by utilizing the further course of the lane, the predicted object state $\boldsymbol{x}^*{}_{V_{k+n}}$ is influenced to change in an appropriate manner. Hence, the course of the road in example 4.36 may have an impact on certain parameters of state prediction $\boldsymbol{x}^*{}_{V_{k+n}}$, such as steering rate and steering angle, although these parameters did not change during the whole observation sequence. I.e. while the object traverses the bend, steering angle as well as steering rate may change in order to keep the observed relation to the lane, thus following the lane instead of driving straight through the bend.

Predicted object measurement $\boldsymbol{y}^*{}_{V_{k+n}}$ can now be provided by evaluating predicted state $\boldsymbol{x}^*{}_{V_{k+n}}$ in relation to the $k+n$'th measurement of the lane $\mathbf{Y}_{L_{k+n}}$.

$$
\begin{aligned}
&p(\boldsymbol{y}^*{}_{V_{k+n}} \mid \mathbf{Y}_{L_{k+n}}, \mathbf{Y}_{V_k}) = \\
&\qquad p(\boldsymbol{y}^*{}_{V_{k+n}} \mid, \mathbf{Y}_{L_{k+n}}, \boldsymbol{x}^*{}_{V_{k+n}}) \cdot p(\boldsymbol{x}^*{}_{V_{k+n}} \mid \mathbf{Y}_{V_k}, \mathbf{Y}_{L_{k+n}})
\end{aligned}
\tag{4.76}
$$

In contrast to 4.70, $\boldsymbol{y}^*{}_{V_{k+n}}$ is dependent on $\boldsymbol{x}^*{}_{V_{k+n}}$ as well as on the $k+n$'th lane measurement $\mathbf{Y}_{L_{k+n}}$. I.e. the geometrical sequence of vehicle measurements will be

similar to the course of the lane, since the lane provides a real world reference, which the observed relative motion model is able to use for long term object prediction.

The maneuvers described in section 4.6.3 express relative behavior that couples the objects from the environment. They can be used to determine both, the current state estimate as well as the future behavior of an object, which will be clarified in the next paragraphs. According to equation 4.76, the most probable state prediction ${\boldsymbol{x}^*}_{O_{k+n}}$ (for the purpose of generality, vehicle V is substituted by object O) is required, in order to derive the most probable object measurement ${\boldsymbol{y}^*}_{O_{k+n}}$ of an object O n steps in the future. The most probable state prediction is derived by accounting for the current state estimate $\hat{\boldsymbol{x}}_O$ as well as the past measurements of the two related objects (see equation 4.74). If the state vector $\boldsymbol{x} = \{{}^{O_j}A_{O_i}, t, \dots\}$ is now comprised by the current maneuver ${}^{O_j}A_{O_i}$ between two objects O_i and O_j as well as its progress t, amongst other state variables, then the state prediction of equation 4.74 can be reformulated to yield:

$$p({}^{O_j}A^*_{O_i,k+n}, t^*_{k+n} \mid \mathbf{Y}_{O_{i,k}}, \mathbf{Y}_{O_{j,k+n}}) = \\ p({}^{O_j}A^*_{O_i,k+n}, t^*_{k+n} \mid \mathbf{Y}_{O_{j,(k+1):n}}, {}^{O_j}\hat{A}_{O_i,k}, \hat{t_k}) \cdot p({}^{O_j}\hat{A}_{O_i,k}, \hat{t_k} \mid \mathbf{Y}_{O_{j,k}}, \mathbf{Y}_{O_{i,k}}) \tag{4.77}$$

and equation 4.76 that formalizes the measurement prediction can be reformulated to yield:

$$p({\boldsymbol{y}^*}_{O_{i,k+n}} \mid \mathbf{Y}_{O_{j,k+n}}, \mathbf{Y}_{O_{i,k}}) = \\ p({\boldsymbol{y}^*}_{O_{i,k+n}} \mid \mathbf{Y}_{O_{j,k+n}}, {}^{O_j}A^*_{O_i,k+n}, t^*_{k+n}) \cdot p({}^{O_j}A^*_{O_i,k+n}, t^*_{k+n} \mid \mathbf{Y}_{O_{i,k}}, \mathbf{Y}_{O_{j,k+n}}) \tag{4.78}$$

In the object-related navigation approach, state estimation $p({}^{O_j}\hat{A}_{O_i,k}, \hat{t_k} \mid \mathbf{Y}_{O_{j,k}}, \mathbf{Y}_{O_{i,k}})$ of equation 4.77, given the measurements $\mathbf{Y}_{O_{j,k}}$ and $\mathbf{Y}_{O_{i,k}}$ of two objects up to the current step k is implemented by comparing whether the past motion between the respective objects can be explained by a maneuver. Figure 4.37 shows the principles with respect to this comparison.

Assume a sequence of relations $\mathcal{R}_{O_1} = (\mathfrak{r}(t_1)_{O_1}, \dots, \mathfrak{r}(t_k)_{O_1}, v(t_1)_{O_1}, \dots, v(t_k)_{O_1})$ up to current step k between two objects (for instance, the robot $\mathcal{E}$ and an object O_1) is given (see figure 4.37 step 1). Assume further that the set of maneuvers

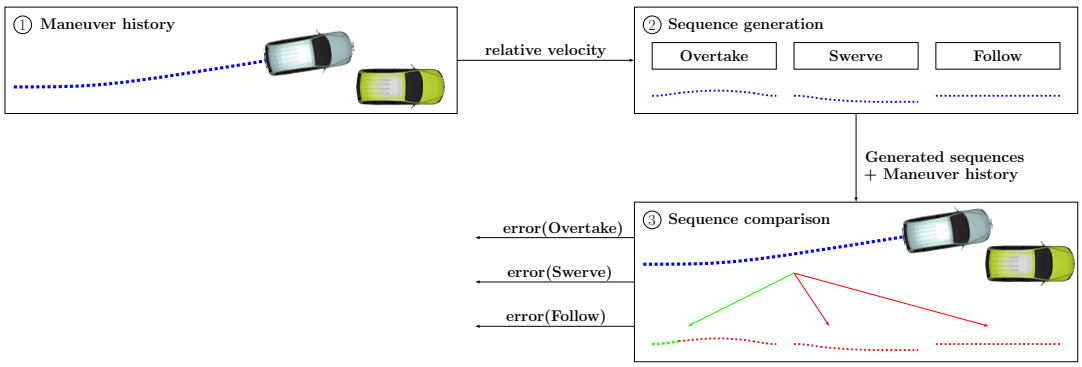


Figure 4.37:
Computation of the error between an observed motion history between two objects and a set of maneuvers that can be performed between the two respective objects.

that the robot is able to apply with respect that object is given (see figure 4.37 step 2). Then we are able to compute the accumulated error between the observed sequence and the sequence of relations that are generated from applying a certain maneuver A (see figure 4.37 step 3). To generate a sequence of relations $\mathcal{R}_A = (\mathfrak{r}(t_1)_A, \ldots, \mathfrak{r}(t_k)_A, v(t_1)_A, \ldots, v(t_k)_A)$ of a maneuver, the maneuver approximation of equation 4.53 is employed. I.e. for each time step t a maneuver A is able to produce a relation $\mathfrak{r}(\alpha(t)_A, \beta(t)_A, d(t)_A)$ that is comprised of the approximated values $\alpha(t)_A$, $\beta(t)_A$ and $d(t)_A$ at time t. The error ϵ between two sequences $\mathcal{R}_{O_1}$ and $\mathcal{R}_A$ (see equation 4.79 and figure 4.37) can be used to determine state estimate $p({}^{O_j}\hat{A}_{O_i,k}, \hat{t_k} \mid \mathbf{Y}_{O_{j,k}}, \mathbf{Y}_{O_{i,k}})$, according to equation 4.80.

$$\epsilon = \min_t \sum_{j=1}^{k} d(\mathfrak{r}(t_j)_{O_1} \ominus \mathfrak{r}(t_j + t)_A) \tag{4.79}$$

$$p({}^{O_j}\hat{A}_{O_i,k}, \hat{t_k} \mid \mathbf{Y}_{O_{j,k}}, \mathbf{Y}_{O_{i,k}}) = \mathrm{e}^{-\epsilon} \tag{4.80}$$

In essence, equation 4.79 accumulates the difference between the two relations $\mathfrak{r}(t_j)_{O_1}$ and $\mathfrak{r}(t_j + t)_A$ with parameter t chosen such, that the overall sum is minimal. The progress $\hat{t_k}$ of maneuver A is then simply $\hat{t_k} = t_j + t$. The more suitable a maneuver seems with respect to current and past object relations the higher is its probability. E.g. if the robot detects an object in front, which is directed towards the robot, the probability of a swerve maneuver increases while the probability of an overtaking maneuver drops.

The left hand side of the state prediction of equation 4.77 is then computed according to equation 4.81. I.e. if the current state is composed of maneuver ${}^{O_j}A_{O_i}$ and t

only, then the most probable future state will be composed of the same maneuver ${}^{O_j}A_{O_i}$ and time step $t+n$ that adds n number of steps.

$$p({}^{O_j}A^{*}_{O_i,k+n}, t^{*}_{k+n} \mid \mathbf{Y}_{O_{j,(k+1):n}}, {}^{O_j}\hat{A}_{O_i,k}, \hat{t_k}) = \begin{cases} 1 & \text{if } {}^{O_j}A^{*}_{O_i,k+n} = {}^{O_j}\hat{A}_{O_i,k} \\ & \quad \text{and } t^{*}_{k+n} = \hat{t_k} + n \\ 0 & \text{otherwise} \end{cases} \tag{4.81}$$

Finally, given the most probable future state $p({}^{O_j}A^{*}_{O_i,k+n})$ at step $k+n$, the most probable measurement prediction ${\boldsymbol{y}^{*}}_{O_{i,k+n}}$ can be derived according to equation 4.82.

$$p({\boldsymbol{y}^{*}}_{O_{i,k+n}} \mid \mathbf{Y}_{O_{j,k+n}}, {}^{O_j}A^{*}_{O_i,k+n}, t^{*}_{k+n}) = \begin{cases} 1 & \text{if } {\boldsymbol{y}^{*}}_{O_{i,k+n}} = \boldsymbol{y}_{O_{j,k+n}} \oplus \\ & \quad \mathfrak{r}(\alpha(t_{k+n})_A, \beta(t_{k+n})_A, d(t_{k+n})_A) \\ 0 & \text{otherwise} \end{cases} \tag{4.82}$$

I.e. the implementation of the prediction computes relation $\mathfrak{r}(\alpha(t_{k+n})_A, \beta(t_{k+n})_A, d(t_{k+n})_A)$ at future time step t_{k+n} when the approximation of maneuver A is given. The computed relation is then added to the already measured relation $\boldsymbol{y}_{O_{j,k+n}}$ of object O_j at time step t_{k+n}. Figure 4.38 provides a visual representation of this process. The most probable future trajectory of the robot (*blue*) is predicted, based on the past relative motion of the two objects (*red*) and the predicted future motion of the overtaken vehicle (*green*).

In the example in figure 4.38 the future trajectory of the overtaken vehicle is represented by measurements $\boldsymbol{y}_{O_{j,k+n}}$ although they might also result from a prediction step that evaluates the motion of the vehicle relative to a lane, for instance.

4.7 Object related reasoning

Based on the results from the previous section, the following section describes how maneuver predictions can be used to allow the robot to reason about the state of the environment. This is achieved by either predicting the most probable state of

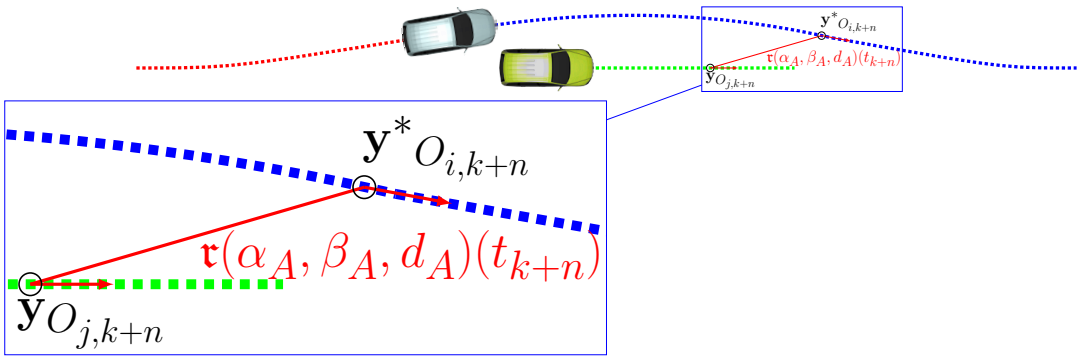


Figure 4.38:
The most probable future trajectory (comprised of future measurements $\boldsymbol{y}^*_{O_{i,k+n}}$) of the robot (*blue*) is predicted, based on the past relative motion of the two objects (*red*) and the future trajectory (comprised of measurements $\boldsymbol{y}_{O_{j,k+n}}$) of the overtaken vehicle (*green*).

other objects or of the robot itself. It will further be clarified how these models enable robust and reliable robot control. Notably, the application of these models to control the robot differs from previous work by e.g. Meyer-Delius et al. [2009], Firl et al. [2012] or Foka and Trahanias [2010]].

Initially, the concepts of maneuver and plan are explained with respect to the reasoning system at hand. The explanation includes how maneuvers are integrated into a long term plan. The section about plan creation tackles the questions of how plans are created and altered, either by a human observer or by the reasoning system itself. The proposed methods also aim at providing simple strategies to create long term plans by humans, which in turn facilitates human-machine interaction. While a plan that is readable for humans is input into the robot reasoning system, the current maneuver as well as the further progress of a plan can be understood by a human observer.

The purpose of the reasoning system is two-fold: firstly, it is responsible for the continuous monitoring of the feasibility of the current plan and, secondly, it aims to change a plan in order to safely reach the predefined goal. Notably, a plan represents a logic concatenation of simple maneuvers, instead of a long term geometric path that is the typical result of the majority of planning approaches described in section 2.2. In general, the conception of the system focuses on presenting the capabilities of the object-related navigation approach in everyday traffic scenarios, such as turning and overtaking and does not aim at the solution of complex scenarios with large amounts of objects.

Therefore, the realization of vehicle control is outlined, which is required in order to steer the robot towards a predefined goal by processing a previously planned sequence of maneuvers. In this context it is shown how the geometric representation of a maneuver is utilized to steer the robot and how the relative velocity with respect to an object is incorporated into the initialization of a maneuver. Furthermore, the robot control strategy is described in detail. In spite of possible modelling errors, as described in section 4.6.4, and of possible measurement uncertainties, robust robot control will be shown. Finally, various applications of a maneuver are described in order to demonstrate the robustness of the representation with respect to different situations.

4.7.1 Maneuvers and plans

Maneuvers are able to represent elementary robot motions (e.g. *"steer left"* or *"steer right"*) that are described by some kind of geometric representation such as lines, arcs or clothoids. Plans either concatenate such geometries or omit this concatenation by creating long term plans that comprise a number of supporting points. However, for this thesis the definition for actions and plans by Nilsson [1984] is employed that was outlined in section 2.2. I.e. instead of basic motions, a maneuver may describe complex motions, such as *"overtake"* or *"turn left"*, which are provided to a hybrid logic-probabilistic reasoner. The reasoner is responsible for generating the optimal sequence of maneuvers that form the plan with respect to the objects of the environment. Each maneuver is given a name with the aim of a more natural human-machine-interaction. On the one hand this allows a human to create long-term plans such as: "Follow lane, then turn right on next crossroad, then ..." and on the other hand it allows the system to explain its current and future actions to a human observer.

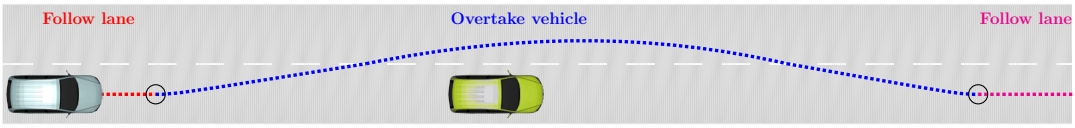


Figure 4.39:
The geometric representation of the plan: *"follow lane"*, *"overtake vehicle"*, *"follow lane"*. The *black* circles represent the locations where two maneuvers connect.

Hence, a maneuver ${}^{O_j}A_{O_i}$ of object O_i relative to object O_j comprises the parametric representation from section 4.6.3 as well as a name that captures the matter of the maneuver verbally. Resultantly a plan Π_{O_i} defines a n-tuple $({}^{O_1}A_{O_i}, {}^{O_2}A_{O_i}, \ldots, {}^{O_n}A_{O_i})$, which contains a sequence of n maneuvers of object O_i relative to a set of objects

O_1 to O_n. One example is shown in figure 4.39, with the associated plan: $\Pi_{O_i} =$ (^{lane}follow$_{robot}$, ^{vehicle}overtake$_{robot}$, ^{lane}follow$_{robot}$) (verbally: *"robot, first follow the lane, then overtake the vehicle, then follow the lane"*). Each maneuver defines the course of a relative motion from an initial relation to a goal relation. The goal relation of a maneuver relative to object O_j is the initial relation of its successor relative to object O_{j+1}, as shown in figure 4.39 (*black* circles). Notably, the objects of a plan do not have to be mutually exclusive, i.e. object O_1 may be equal to object O_2. For instance, driving multiple times in a roundabout implies that each traffic sign is recognized multiple times and is thus included a number of times in a respective plan.

These conceptions allow the system to represent the behavior of the robot in a manner that is comprehensible to humans. They enable the generation of a coarse plan with regard to a set of objects in the environment without the need to plan exact motions. Furthermore, they allow for quick plan alterations and adaptations by the reasoning system to possible errors made by the perception or the prediction system. E.g. if an overtaken vehicle drifts to the left and thus provokes a collision with the robot during an overtaking maneuver, the plan does not have to be rejected and planned anew. Merely an adaptation of the steering behavior of the robot is needed e.g. by either increasing or decreasing the velocity or by following the motion of the overtaken vehicle. The plan has to be altered only, if this adaptation leads to a collision with another object, e.g. the guard railing. But instead of rejecting the old plan completely, merely minor adaptations of the plan are required, e.g. by inserting a maneuver that resolves the conflict, such as: *"follow vehicle"*.

4.7.2 Object-action-map

How maneuvers are generated was described in section 4.6.3. Once they are generated, maneuver representations are stored in a data structure called object-action-map, which maps maneuvers to their primary object class. The primary object class is the one that performs the maneuver, e.g. the robot itself is an instantiation of a primary object class (vehicle) during turn, overtake or evasion maneuvers. Similarly a bicycle may be the primary object during a maneuver that describes its traversal of a road. Each primary object class in the object-action-map is connected to the secondary object class, i.e. the object that the maneuver is applied to. In the example plan above, the *robot* was always the primary object, while *lane* and *vehicle* were the secondary objects. The secondary object class is allowed to be a

primary object class as well; during an overtaking maneuver, for example, the primary object class "vehicle" is also the secondary object class. An example graph of an object-action-map is shown in figure 4.40, where the blue boxes describe possible maneuvers between two objects.

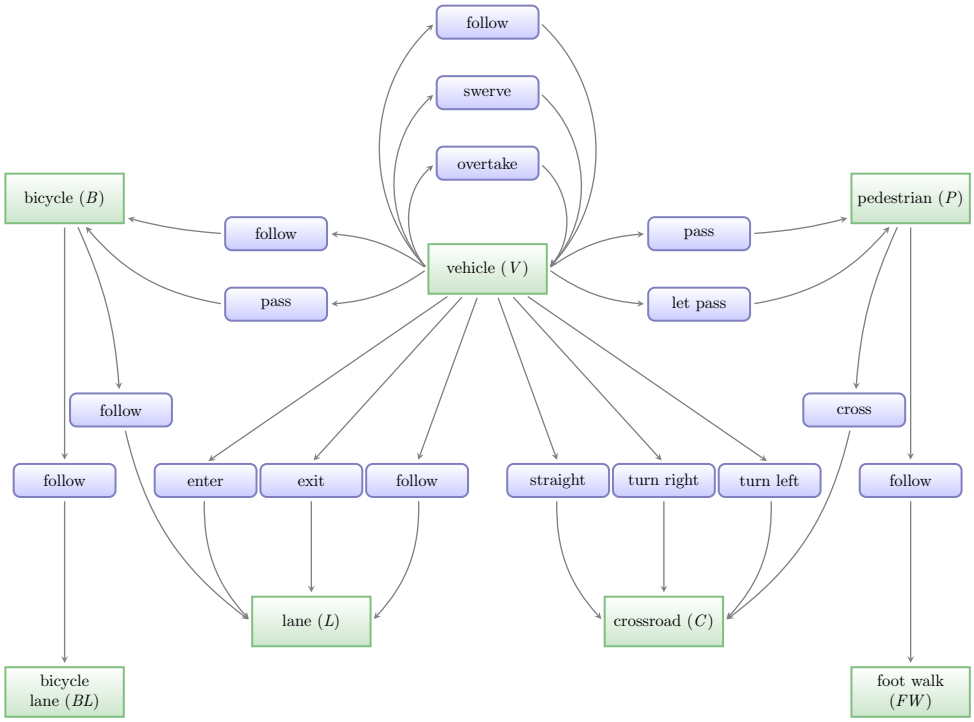


Figure 4.40:
The visual representation of a possible object-action-map, with the *blue* boxes describing possible maneuvers between two object classes (*green*).

The aim of this data structure is to quickly access the range of maneuvers that an object class is able to execute. Either in order to access the driving capabilities of the robot or to gain quick access to possible maneuvers of other objects in the environment. Every maneuver that is captured and approximated according to the methods in the previous sections can thus be stored for later re-use. The object-action-map is implemented in a concurrent way to enable the parallel evaluation of possible maneuvers of a perceived object. In this manner every maneuver of an object in the environment is tracked and evaluated to allow the system to make assumptions about possible future behaviors.

If the set of perceived objects in the environment of the robot is given, one instance of the object-action map can be created. One instance contains only those maneuvers that are applicable to the given objects. I.e. an instance of the map represents the current state of the environment at a given point in time with the root node representing the observer. During an instantiation, the map organizes

itself automatically to obtain a representation of the environment that expresses the dependencies between the perceived objects. Figure 4.41, for instance, provides the state of the environment of the initial scenario from figure 4.3b (two vehicles that follow their respective lane) given the complete set of maneuvers that is represented in the object-action-map from figure 4.40.

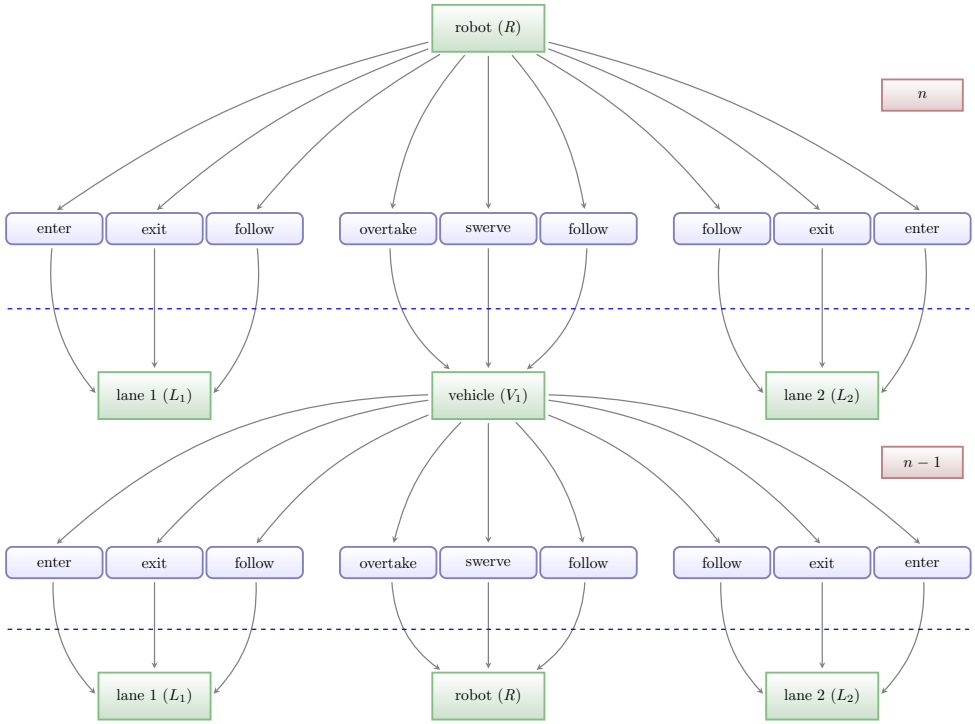


Figure 4.41:
An instantiation of the object-action-map from figure 4.40 with respect to scenario represented in figure 4.3b. The dashed lines represent the boundaries between the different reasoning levels, with the bottom level being indexed with 0.

Notably, there is no instantiation of the class nodes for bicycle or pedestrian (compare figure 4.40). I.e. instead of evaluating the complete set of maneuvers the robot is able to perform, only those maneuvers that are assigned to the perceived objects are evaluated. Furthermore, the object-object dependencies of one instantiation of the object-action-map form a tree-like structure with the robot being the root node. In that fashion a sequence is created that defines in which order the objects have to be processed. For instance, in order to evaluate the driving possibilities of the robot, the driving possibilities of vehicle V_1 have to be evaluated first.

To avoid recurring object instantiations due to cyclic dependencies (e.g. objects of class vehicle depend on each other and may thus force an infinite forking of the tree), each object in an instantiation of the object-action-map is allowed to be primary object in its branch of the tree once only. I.e. since the robot is the root node of

the graph of figure 4.41, it is allowed to be secondary object only in the remains of the graph (see figure 4.41 bottom center node). If the robot was allowed to be primary object again, the graph would grow limitless. To clarify this decision, we assume the following example: two vehicles drive in opposing directions towards each other. The respective drivers are not able to tell what the other will be doing in the future (will the other driver avoid the collision or not?). Hence each driver must decide independently what maneuver to apply (typically each driver should choose an action to avoid the collision, independent of the choice of the other driver). The decision to not allow cyclic dependencies simplifies the reasoning process, but still allows the incorporation of each object-object interaction.

In order to derive an optimal maneuver for the root node (i.e. the robot) in a given situation, the tree is evaluated from the bottom to the top, so that maneuvers closer to the root node depend on the maneuvers that are farther away. For instance, the robot maneuvers "swerve", "follow" or "overtake" in figure 4.41 depend on the future actions of vehicle V_1 relative to lane L_1, lane L_2 or the robot R in the situation representation. We assume that all maneuvers A of an instantiation of the object-action-map that are k steps away from the root node are put in the same set or level Ω_A^k. We further assume that all primary objects of those maneuvers are put into set Ω_O^k (the dashed lines in figure 4.41 represent the boundaries of the differently indexed sets). The set that contains all maneuvers farthest away from the root node gets index 0 and the index increases with every step closer to the root node. Now, the most probable set of maneuvers Ω_A^n at level n, which is dependent on all other maneuvers and objects in sets $n-1$ down to level 0, can be computed by applying Bayes rule as shown in equation 4.83.

$$p(\Omega_A^n \mid \Omega_A^{n-1}, \ldots, \Omega_A^1, \Omega_O^{n-1}, \ldots, \Omega_O^0) = \frac{p(\Omega_A^{n-1}, \ldots, \Omega_A^1, \Omega_O^{n-1}, \ldots, \Omega_O^0 \mid \Omega_A^n) \cdot p(\Omega_A^n)}{p(\Omega_A^{n-1}, \ldots, \Omega_A^1, \Omega_O^{n-1}, \ldots, \Omega_O^0)}, \tag{4.83}$$

In the example from figure 4.41 set Ω_A^n is comprised of the maneuvers of the robot R (*"swerve"*, *"follow"*, *"exit"*, ...), set Ω_O^{n-1} contains objects vehicle V_1, lane L_1 and lane L_2 and set Ω_A^{n-1} is comprised of the maneuvers of the vehicle V_1. At the bottom level, just the set of objects Ω_O^0 remains, with its entities lane L_1, lane L_2 and robot R.

The denominator of equation 4.83 expresses the joint probability of observing all maneuvers Ω_A^{n-1} and all objects Ω_O^{n-1} of level $n-1$ and below. This represents a high dimensional distribution over all maneuvers and object states that cannot be gathered. But the denominator can be reformulated by conditioning the set of maneuvers Ω_A^{n-1} of level $n-1$ on all other levels. By applying Bayes rule the denominator of equation 4.83 can be reformulated according to equation 4.84.

$$\begin{aligned} &p(\Omega_A^{n-1}, \ldots, \Omega_A^1, \Omega_O^{n-1}, \ldots, \Omega_O^0) = \\ &\quad p(\Omega_A^{n-1} \mid \Omega_A^{n-2}, \ldots, \Omega_A^1, \Omega_O^{n-2}, \ldots, \Omega_O^0) \cdot p(\Omega_A^{n-2}, \ldots, \Omega_A^1, \Omega_O^{n-2}, \ldots, \Omega_O^0) \end{aligned} \tag{4.84}$$

Since the set of maneuvers Ω_A^{n-1} is independent of the set of objects Ω_O^{n-1} of level $n-1$ the dependence on the object set can be neglected for the conditional probability of equation 4.84. Furthermore, by evaluating the right hand side of equation 4.84 in the same manner, equation 4.84 can be further simplified to yield:

$$\begin{aligned} &p(\Omega_A^{n-1}, \ldots, \Omega_A^1, \Omega_O^{n-1}, \ldots, \Omega_O^0) = \\ &\qquad \prod_{k=n}^{2} p(\Omega_A^{k-1} \mid \Omega_A^{k-2}, \ldots, \Omega_A^1, \Omega_O^{k-2}, \ldots, \Omega_O^0) \end{aligned} \tag{4.85}$$

Each factor of the product of equation 4.85 can then be evaluated in a recursive fashion by applying equation 4.83. I.e. in order to evaluate the maneuvers of level n, the maneuvers of level $n-1$ must be evaluated first.

Further, the numerator of equation 4.83 can be simplified by evaluating the conditional dependencies of probability $p(\Omega_A^{n-1}, \ldots, \Omega_A^1, \Omega_O^{n-1}, \ldots, \Omega_O^0 \mid \Omega_A^n)$. According to the graph in figure 4.41 maneuvers of level $n-1$ are independent of maneuvers of level n and maneuvers of level n are independent of objects of level n. Thus the numerator of equation 4.83 can be simplified according to equation 4.86.

$$\begin{aligned} &p(\Omega_A^{n-1}, \ldots, \Omega_A^1, \Omega_O^{n-1}, \ldots, \Omega_O^0 \mid \Omega_A^n) \cdot p(\Omega_A^n) = \\ &\qquad \prod_{k=n}^{1} p(\Omega_O^{k-1} \mid \Omega_A^k) \cdot p(\Omega_A^n) \end{aligned} \tag{4.86}$$

By then re-substituting equations 4.85 and 4.86 into 4.83 and absorbing probability $p(\Omega_A^n)$ into a normalization constant, because all maneuvers are equally likely to

occur, equation 4.87 is derived. It describes how to determine the most probable robot maneuver.

$$p(\Omega_A^n \mid \Omega_A^{n-1}, \ldots, \Omega_A^1, \Omega_O^{n-1}, \ldots, \Omega_O^0) \propto \frac{\prod_{k=n}^{1} p(\Omega_O^{k-1} \mid \Omega_A^k)}{\prod_{k=n}^{2} p(\Omega_A^{k-1} \mid \Omega_A^{k-2}, \ldots, \Omega_A^1, \Omega_O^{k-2}, \ldots, \Omega_O^0)} \tag{4.87}$$

Equation 4.87 represents the simplification of probability $p(\Omega_A^n \mid \Omega_A^{n-1}, \ldots, \Omega_A^1, \Omega_O^{n-1}, \ldots, \Omega_O^0)$ after all independencies have been considered. It consists of two parts. The denominator of 4.87 can be evaluated in a recursive fashion by applying equation 4.87 to maneuvers of the next lower level. The nominator, i.e. the probability $p(\Omega_O^{k-1} \mid \Omega_A^k)$ of observing a set of object descriptions Ω_O^{k-1} given a set of maneuvers Ω_A^k, can be determined by applying the considerations from section 4.6.5 that describes how future object states can be predicted by evaluating the relative behavior of objects. This methodology will be clarified in the next paragraph.

First, probability $p(\Omega_O^{k-1} \mid \Omega_A^k)$ will be expanded, as shown in equation 4.88.

$$p(\Omega_O^{k-1} \mid \Omega_A^k) = p(O_1, \ldots, O_n \mid A_1, \ldots, A_m), \text{ with: } O_i \in \Omega_O^{k-1} \text{ and } A_j \in \Omega_A^k \tag{4.88}$$

I.e. the objects O_i are conditionally dependent on some of the maneuvers A_j, since maneuvers can be applied to them or interfere with their future motion. E.g. if the robot applies maneuver *"overtake"* to an object *"vehicle 1"* that is located to the front of the robot, the robot might interfere with the future motion of an oncoming object *"vehicle 2"*, which would decrease the probability of *"vehicle 2"* moving in the predicted way. Vice versa, the probability of applying maneuver *"overtake"* will decrease as well.

However, all maneuvers and objects within a set are conditionally independent, therefore equation 4.88 can be further simplified to yield:

$$p(\Omega_O^{k-1} \mid \Omega_A^k) = \prod_i \prod_j p(O_i \mid A_j) \tag{4.89}$$

Computing $p(O_i \mid A_j)$ of equation 4.89 thus means to compute, whether the current relation of an object, as well as its most probable future path can be explained by a maneuver. The maneuver approximation from section 4.6.3 as well as the prediction of future object relations from section 4.6.5 now help to realize this computation.

Apart from its geometric properties such as length and width or its classification, an object description O contains the set $\mathbf{Y}$ of current and past object measurements and its most probable future path. The most probable future path is a series of the most probable future measurements $\boldsymbol{y}^*$. I.e. conditional probability $p(O_i \mid A_j)$ can be reformulated, according to equation 4.90.

$$p(O_i \mid A_j) = p({\boldsymbol{y}^*}_{O_{i_{k+n}}}, \ldots, {\boldsymbol{y}^*}_{O_{i_{k+1}}}, \mathbf{Y}_{O_{i_k}} \mid A_j) \tag{4.90}$$

Equation 4.90 essentially expresses, whether given past and predicted future object measurements are likely, if maneuver A_j is applied. The outcome of applying a maneuver was already clarified in section 4.6.5, when the methodology of object prediction was introduced. According to equation 4.82 the future object measurement ${\boldsymbol{y}^*}_{O_{k+n}}$ of an object O at time step $k+n$ can be determined by applying a maneuver A. Above equation 4.90 can thus be implemented via an intersection test between sequences $\mathcal{R}_{O_i} = (\mathfrak{r}(t_k)_{O_i}, \ldots, \mathfrak{r}(t_{k+n})_{O_i}, v(t_k)_{O_i}, \ldots, v(t_{k+n})_{O_i})$ of object O_i and $\mathcal{R}_{A_j} = (\mathfrak{r}(t_k)_{A_j}, \ldots, \mathfrak{r}(t_{k+n})_{A_j}, v(t_k)_{A_j}, \ldots, v(t_{k+n})_{A_j})$ of maneuver A_j, which verifies that the future motions of two objects (represented by means of relations) do not interfere with one another. Figure 4.42 provides a visual representation of the procedure.

As figure 4.42 shows, probability $p({\boldsymbol{y}^*}_{O_{i_{k+n}}}, \ldots, {\boldsymbol{y}^*}_{O_{i_{k+1}}}, \mathbf{Y}_{O_{i_k}} \mid A_j)$ of equation 4.90 is lower, the more two sequences intersect, because two objects cannot occupy the same space at the same time. The error that results from the intersection test can be formalized as shown in equation 4.91. It can then be used to determine probability $p(O_i \mid A_j)$ according to equation 4.92.

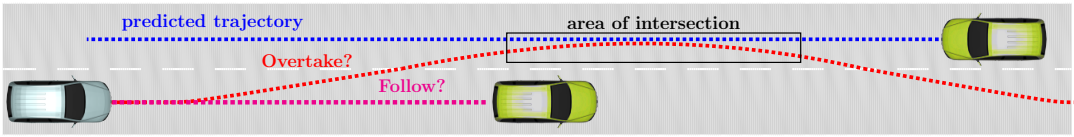


Figure 4.42:
Intersection test of different maneuvers the robot (*white vehicle*) is able to apply. The black rectangle shows an area, where an already predicted trajectory of an oncoming object (*blue*) intersects with the locations maneuver *"overtake"* (*red*) would occupy.

$$\epsilon = \sum d(\mathfrak{r}(t_k)_{O_i}, \mathfrak{r}(t_k)_{A_j}) \tag{4.91}$$

$$p({\boldsymbol{y}^*}_{O_{i_{t+n}}}, \ldots, {\boldsymbol{y}^*}_{O_{i_{t+1}}}, \mathbf{Y}_{O_{i_t}} \mid A_j) = \mathrm{e}^{-\epsilon} \tag{4.92}$$

Equation 4.91 accumulates the error of an intersection test $d(\cdot, \cdot)$, when two relations are given. Input to the testing method are two relations $\mathfrak{r}(t_k)_{O_i}$ and $\mathfrak{r}(t_k)_{A_j}$ as well as geometries $\mathcal{B}_{O_i}$ and $\mathcal{B}_{A_j}$. The geometries describe the bounding box of the respective objects. A geometry $\mathcal{B}$ represents any convex polygon by means of a set of ordered relations (see equation 4.93). Each relation $\mathfrak{r}$ describes the geometric location of a boundary point in relation to the center point of object O_i as shown in figure 4.43.

$$\mathcal{B}_{O_i} = \{\mathfrak{r}_{O_{i,1}}, \mathfrak{r}_{O_{i,2}}, \ldots, \mathfrak{r}_{O_{i,n}}\} \tag{4.93}$$

By evaluating the relation ${}^{A_j}\mathfrak{r}_{O_i}(t_k) = \mathfrak{r}_{A_j}(t_k) \ominus \mathfrak{r}_{O_i}(t_k)$ between objects O_i and O_j (the object which maneuver A_j is applied to) subsequently, the intersection of two geometries reduces to one triangle-triangle intersection (see figure 4.43 *blue* and *magenta* triangle).

When relation ${}^{A_j}\mathfrak{r}_{O_i}(t_k)$ and its inverse relation ${}^{A_j}\mathfrak{r}^{-1}_{O_i}(t_k)$ are given, the angle-values of their respective α-entities (see figure 4.43, *blue* and *magenta* arcs) are used to determine the two adjacent boundary points that enclose the respective angle. In figure 4.43 the enclosing boundary points are coloured *blue* and *magenta*. The boundary points that enclose the α-entity of relation ${}^{A_j}\mathfrak{r}_{O_i}(t_k)$ can be named ${}^{\alpha_+}\mathfrak{r}_{O_i}$ and ${}^{\alpha_-}\mathfrak{r}_{O_i}$ and the boundary points that enclose the α^{-1}-entity of relation ${}^{A_j}\mathfrak{r}^{-1}_{O_i}(t_k)$ can be named ${}^{\alpha^{-1}_+}\mathfrak{r}_{A_j}$ and ${}^{\alpha^{-1}_-}\mathfrak{r}_{A_j}$. By further adding relations $\mathfrak{r}_{O_i}$ and ${}^{A_j}\mathfrak{r}_{O_i}$, i.e.

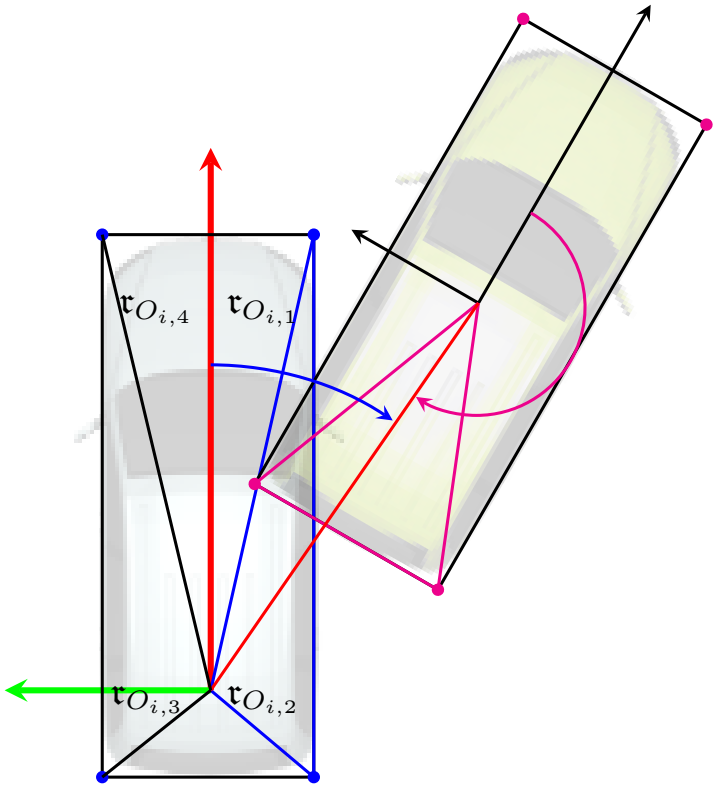


Figure 4.43:
Intersection test between two objects that are represented by a convex geometry, where the colored triangles need to be checked only.

the reference points of the objects, to these boundaries, we obtain the triangular representation of equation 4.94 that is also shown in figure 4.43.

$$\triangle_{O_i} = \{\mathfrak{r}_{O_i}, {}^{\alpha_+}\mathfrak{r}_{O_i}, {}^{\alpha_-}\mathfrak{r}_{O_i}\} \text{ and } \triangle_{A_j} = \{{}^{A_j}\mathfrak{r}_{A_i}, {}^{\alpha_+^{-1}}\mathfrak{r}_{A_j}, {}^{\alpha_-^{-1}}\mathfrak{r}_{A_j}\} \tag{4.94}$$

The intersection of both triangles $\triangle_{O_i}$ and $\triangle_{A_j}$ is then realized by cross-checking whether any line-segment of $\triangle_{O_i}$ intersects with any line-segment of $\triangle_{A_j}$. Figure 4.43 shows an example of how the intersection test is applied. Any line-segment of the *blue* triangle is tested against any line-segment of the *magenta* triangle.

In general, if one line is made up of two relations $\mathfrak{r}_1$ and $\mathfrak{r}_2$ and one line is made up of relations $\mathfrak{r}_3$ and $\mathfrak{r}_4$, the intersection test between both lines verifies that the points described by relations $\mathfrak{r}_3$ and $\mathfrak{r}_4$ are separated by the line described by relations $\mathfrak{r}_1$ and $\mathfrak{r}_2$ and vice versa. To this end, the difference relation ${}^j\mathfrak{r}_i = \mathfrak{r}_j \ominus \mathfrak{r}_i$ from section 4.5.1 is used that determines the location of a relation $\mathfrak{r}_j$ relative to a relation $\mathfrak{r}_i$. Given two line segments, the difference operation is used to determine relations ${}^2\mathfrak{r}_1$,

${}^3\mathfrak{r}_1$, ${}^4\mathfrak{r}_1$, i.e. of how all relations are located relative to relation $\mathfrak{r}_1$, as well as ${}^1\mathfrak{r}_3$, ${}^2\mathfrak{r}_3$, ${}^4\mathfrak{r}_3$, i.e. of how all relations are located relative to relation $\mathfrak{r}_3$. The α-entity is used from each such relation (${}^{\mathfrak{r}_j}\alpha_{\mathfrak{r}_i}$ defines the α-entity of relation ${}^j\mathfrak{r}_i$) and evaluated according to equation 4.95 that verifies the above mentioned separation.

$$\cap(\mathfrak{r}_1,\ldots,\mathfrak{r}_4) = \begin{cases} true & \text{if } {}^{\mathfrak{r}_3}\alpha_{\mathfrak{r}_1}(\leq / \geq)^{\mathfrak{r}_2}\alpha_{\mathfrak{r}_1} \text{ and } {}^{\mathfrak{r}_4}\alpha_{\mathfrak{r}_1}(\geq / \leq)^{\mathfrak{r}_2}\alpha_{\mathfrak{r}_1} \\ & \text{and } {}^{\mathfrak{r}_4}\alpha_{\mathfrak{r}_3}(\leq / \geq)^{\mathfrak{r}_1}\alpha_{\mathfrak{r}_3} \text{ and } {}^{\mathfrak{r}_4}\alpha_{\mathfrak{r}_3}(\geq / \leq)^{\mathfrak{r}_2}\alpha_{\mathfrak{r}_3} \\ false & \text{otherwise} \end{cases} \quad (4.95)$$

Given the test from equation 4.95 two predicted object trajectories can be evaluated with respect to their degree of intersection. The more the trajectories and the object boundaries of the respective objects intersect, the lower is the probability $p(\boldsymbol{y}^*{}_{O_{i_{t+n}}},\ldots,\boldsymbol{y}^*{}_{O_{i_{t+1}}},\mathbf{Y}_{O_{i_t}} \mid A_j)$ from equation 4.92, which reduces the probability that maneuver A_j is a good choice.

Taking probability $p(\boldsymbol{y}^*{}_{O_{i_{t+n}}},\ldots,\boldsymbol{y}^*{}_{O_{i_{t+1}}},\mathbf{Y}_{O_{i_t}} \mid A_j)$ into account, the most probable set of maneuvers Ω_A^n at level n can now be determined according to equation 4.87. Thus, by evaluating all given dependencies of the object-action-map, the most probable maneuver the robot needs to apply in a certain situation can be determined.

By abiding by the object-action-map, the most probable maneuver at a given point in time can be determined. How this methodology will be used to create long-term plans will be described in the next section.

4.7.3 Plan generation and execution

The object-action-map provides the most probable maneuver, given the current situation. But by always choosing the most probable maneuver an arrival at a desired goal is not guaranteed. Thus, long-term plans need to be created, that allow the definition of a route, which ultimately leads to the desired goal. Instead of utilizing sequences of e.g. GPS-points, as most state-of-the-art algorithms do, however, the object-related navigation approach defines a route by a concatenation of maneuvers. Such a concatenation might be verbally described by instructions like: *"follow vehicle X, then turn right on next crossroad, then follow the road, ..."*. In this example,

maneuvers like *"follow"* or *"turn right"* are applied to different objects, that might or might not be visible already. Object *"vehicle X"* might be the vehicle in front of the robot, which allows the direct application of the specified maneuver. But object *"next crossroad"* specifies not a tangible object, but rather an abstract class, which might be perceived in the future. Thus, a distinction is made between these two object types, that can make up a plan. Symbol O will be used for already perceived objects in the environment of the robot $\mathcal{E}$ (e.g. the car in front), while symbol C represents an abstract object class, such as *"vehicle"*. The example plan above can now be expressed by the following mathematical statement:

$$({}^{O_1}A_{\mathcal{E}}, {}^{C_1}A_{\mathcal{E}}, {}^{C_2}A_{\mathcal{E}}), \tag{4.96}$$

where O_1 represents *"vehicle X"*, C_1 any upcoming crossroad object, C_2 any upcoming road object and A the named maneuvers. If symbol X is used as a place holder for either a tangible object O or an object class C a more general mathematical statement can be created that describes an initial plan $\Pi^I_{\mathcal{E}}$ with respect to the robot $\mathcal{E}$ (see equation 4.97).

$$\begin{aligned}\Pi^I_{\mathcal{E}} = ({}^{X_1}A_{\mathcal{E}}, {}^{X_2}A_{\mathcal{E}}, \ldots, {}^{X_n}A_{\mathcal{E}}) \\ \text{,with } X_j \in \{O_1, \ldots, O_N, C_1, \ldots, C_M\} \text{ and } j = 1, \ldots, n\end{aligned} \tag{4.97}$$

In this manner, plans that incorporate actual percepts such as *"the red car in front"* as well as abstract descriptions such as *"the next crossroad"* into the reasoning process are rendered possible.

But for some plans it might not be possible to apply maneuver ${}^{X_2}A_{\mathcal{E}}$ directly after maneuver ${}^{X_1}A_{\mathcal{E}}$. E.g. when the plan: *"turn left"*, *"follow lane"* cannot be executed, because a lane blockage occurred. Thus, for each maneuver ${}^{X_j}A_{\mathcal{E}}$ of an initial plan 4.97, the reasoning system tries to find an optimal sequence of maneuvers to fulfil this particular goal, according to the Dijkstra shortest path algorithm [Dijkstra, 1959]. Thus, a reward function $r({}^{O_j}A_{\mathcal{E}})$ is created that evaluates whether the application of a maneuver ${}^{O_j}A_{\mathcal{E}}$ leads either to a collision (low reward) or to a safe robot motion (high reward). The function is defined according to equation 4.98.

$$r({}^{O_j}A_{\mathcal{E}}) = p({}^{O_j}A_{\mathcal{E}} \mid \Omega_A^{n-1}, \ldots, \Omega_A^1, \Omega_O^{n-1}, \ldots, \Omega_O^0) \tag{4.98}$$

It employs equation 4.83 from the previous section and thus the object-action-map, which is able to provide a probability for each robot maneuver, given the current set of objects and their future paths. I.e. the higher the probability of a robot maneuver, the higher the reward.

Furthermore, a function is required that terminates the Dijkstra algorithm. Termination function $r_F({}^{O_j}A_{\mathcal{E}})$ provides information on whether sub-goal ${}^{O_j}A_{\mathcal{E}}$ of initial plan $\Pi^I_{\mathcal{E}}$ is achieved (see equation 4.99).

$$r_F({}^{O_j}A_{\mathcal{E}}) = \begin{cases} 1 & \text{if } {}^{O_j}A_{\mathcal{E}} \equiv {}^{X_j}A_{\mathcal{E}} \\ 0 & \text{if } {}^{O_j}A_{\mathcal{E}} \neq {}^{X_j}A_{\mathcal{E}} \end{cases} \tag{4.99}$$

In essence, equation 4.99 checks whether the current maneuver ${}^{O_j}A_{\mathcal{E}}$ of a partial plan is already the desired goal maneuver ${}^{X_j}A_{\mathcal{E}}$ or not.

Finally, the estimated reward $R^*_k({}^{X_j}A_{\mathcal{E}})$ of a k-step plan with respect to the specific goal maneuver ${}^{X_j}A_{\mathcal{E}}$ is computed by maximizing the reward of the sequence of maneuvers ${}^{O_1}A_{\mathcal{E}}, \ldots, {}^{O_k}A_{\mathcal{E}}$ given reward function $r(\cdot)$ (equation 4.98) and termination function $r_F(\cdot)$ (equation 4.99).

$$R^*_k({}^{X_j}A_{\mathcal{E}}) = \max_{{}^{O_{(\cdot)}}A^1_{\mathcal{E}}, \ldots, {}^{O_{(\cdot)}}A^k_{\mathcal{E}}} \{ \prod_{l=1}^{k} r({}^{O_{(\cdot)}}A^l_{\mathcal{E}}) \cdot r_F({}^{O_{(\cdot)}}A^l_{\mathcal{E}}) \} \tag{4.100}$$

Essentially, equation 4.100 describes a path through k different instantiations of the object-action-map. At each step l the set of objects including their object classes and relations is evaluated to obtain optimal maneuver ${}^{O_{(\cdot)}}A^l_{\mathcal{E}}$ at the respective point in time. Object constellations of future time steps, i.e. for $l \geq 2$ are determined by predicting the motion of each object taking into account its most likely maneuver. Finally, by maximizing the product over all sub-goals of initial plan $\Pi^I_{\mathcal{E}}$ according to equation 4.101, the optimal sequence of maneuvers that comprises the entire initial plan is generated.

$$R^*_n(\Pi^I_{\mathcal{E}}) = \prod_{j=1}^{n} R^*_k({}^{X_j}A_{\mathcal{E}}) \tag{4.101}$$

The algorithmic steps of equations 4.100 and 4.101 are visualized in figures 4.44 and 4.45. Algorithm 4.100 provides the most suitable sequence of k maneuvers

${}^{O_{(\cdot)}}A^1_{\mathcal{E}}, \ldots, {}^{O_{(\cdot)}}A^k_{\mathcal{E}}$ to reach a certain sub-goal. The input to the algorithm is sub-goal maneuver ${}^{X_j}A_{\mathcal{E}}$ at planning step j (see figure 4.44 between steps 1 and 2). Sub-goal maneuver ${}^{X_j}A_{\mathcal{E}}$ is provided by the plan creation procedure (figure 4.44 step 1). E.g. during manual user input, the plan *"Overtake red vehicle in front, turn left on next crossroad"* might have been created. Both maneuvers are sub-goals that are fed, one by one, to step 2 of figure 4.44. In this context, $O_{(\cdot)}$ represents an object with undefined index $(\cdot)$, which can take on any integer and $X_j, j = 1, \ldots, n$ represents an object that may be either an object in the environment of the robot (e.g. *"red vehicle in front"*) or an abstract object class (e.g. *"crossroad"*). The maneuver evaluation step (figure 4.44 step 2) iterates over all maneuvers ${}^{O_{(\cdot)}}A^l_{\mathcal{E}}$ the robot is able to apply to every object $O_{(\cdot)}$ in the environment of the robot at planning horizon l. Planning horizon l defines how many maneuvers are already concatenated to reach the desired sub-goal. I.e. for every planning horizon (up to k) the outcome of each applicable robot maneuver is predicted (figure 4.44, step 3) and reward $r(\cdot)$ of equation 4.98 is determined. The sequence of maneuvers ${}^{O}A^1_{\mathcal{E}}, \ldots, {}^{O}A^l_{\mathcal{E}}$ that maximizes the reward and fulfils the termination condition of equation 4.99 (figure 4.44, step 4), i.e. that the last maneuver of the sequence is the desired sub-goal maneuver, will be returned to the global planning procedure.

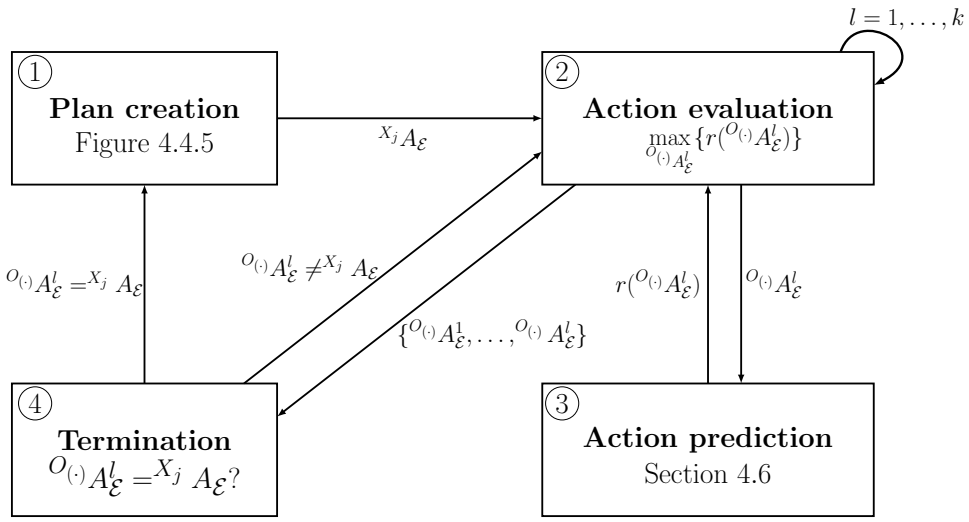


Figure 4.44:
Steps of the sub-goal planning procedure, whose input is a sub-goal ${}^{X_j}A_{\mathcal{E}}$ of initial plan $\Pi^I_{\mathcal{E}}$ and whose output is a sequence of maneuvers to arrive at the respective sub-goal.

The result of algorithm 4.44 is then utilized in the global planning procedure from figure 4.45 that computes the optimal sequence of maneuvers, given a series of sub-goals.

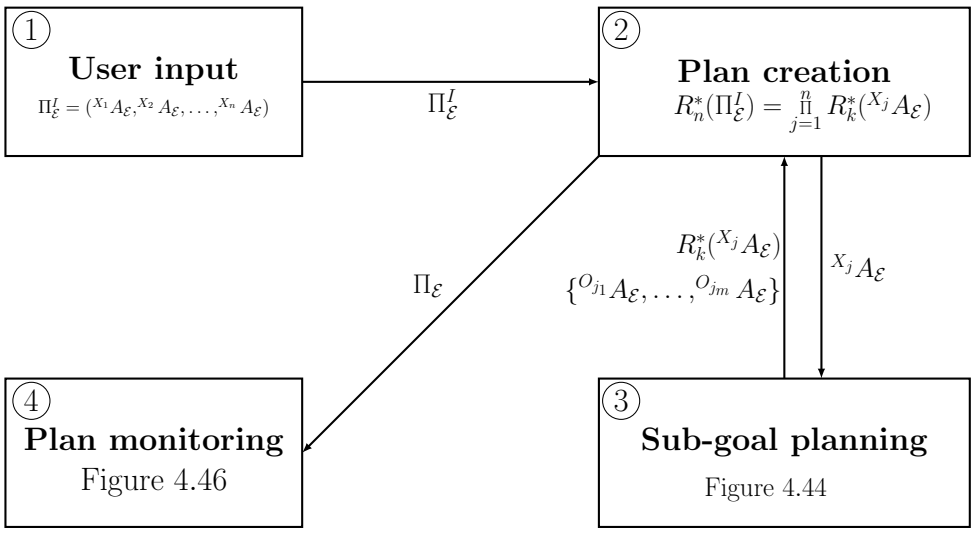


Figure 4.45:
Steps of the global maneuver planning procedure. A user defined plan $\Pi^I_{\mathcal{E}}$ is refined, so that every sub-goal is applicable. The final plan $\Pi_{\mathcal{E}}$ is then provided to the monitoring system.

Given an initial plan $\Pi^I_{\mathcal{E}}$ that might be defined by a human user (figure 4.45, step 1) the algorithm iterates over all its n elements ${}^{X_j}A_{\mathcal{E}}$ (figure 4.45, step 2). In each iteration the respective sub-goal ${}^{X_j}A_{\mathcal{E}}$ is provided to the sub-goal planning procedure (figure 4.45, step 3) to retrieve the optimal sequence of maneuvers that enable the application of sub-goal maneuver ${}^{X_j}A_{\mathcal{E}}$. By retrieving the optimal sequence of maneuvers per sub-goal, an optimal global plan $\Pi_{\mathcal{E}}$, that maximizes equation 4.101 is derived by concatenating the various sub-plans. Plan $\Pi_{\mathcal{E}}$ is then provided to the monitoring procedure, that verifies its correct execution (figure 4.45, step 4).

The maneuvers of plan $\Pi_{\mathcal{E}}$ are processed in order with the first maneuver in the plan being active after the procedure in figure 4.45 is finished (figure 4.46, step 1). The overall progress of the active maneuver is monitored continuously to detect possible hazards (e.g. other objects do not behave as predicted) as quickly as possible. After its successful application, the active maneuver is removed from the plan and its successor is defined as active. Hazard detection as well as progress monitoring are implemented as a continuous model check, i.e. the current maneuvers of other traffic participants are checked for their interference with the current maneuver of the robot. Reward $r(A_{\mathcal{E}})$ of equation 4.98 is utilized to check whether the current maneuver of the robot is still applicable (figure 4.46, step 2). If a hazard occurs (reward $r(A_{\mathcal{E}}) = 0$), the subsequent replanning routine does not disregard the entire plan, but it tries to iteratively reach one of the successive maneuvers (figure 4.46, step 3) by utilizing the sub-goal planning procedure from figure 4.44.

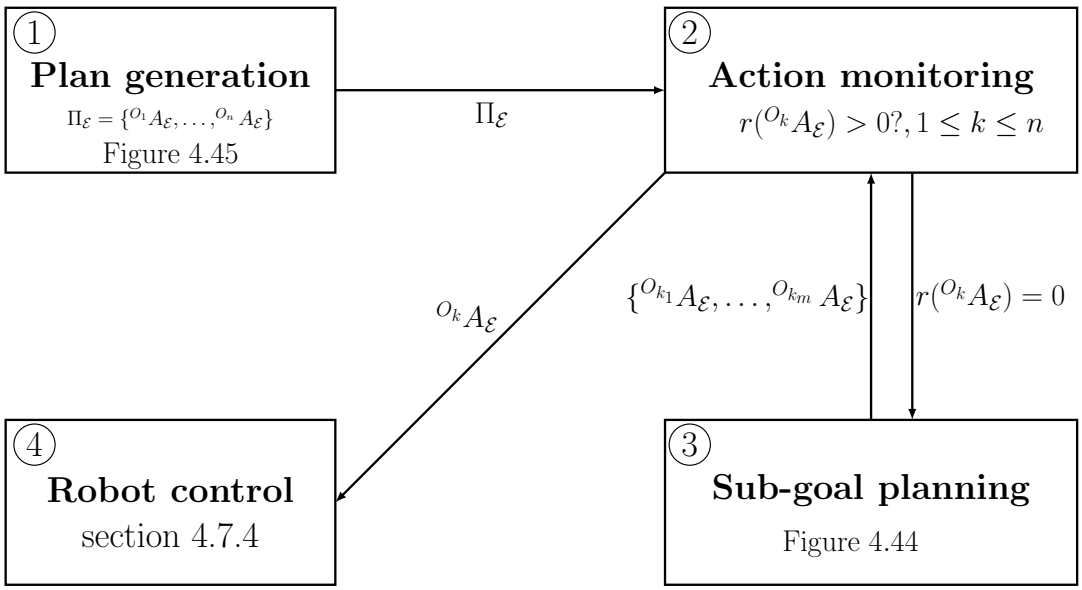


Figure 4.46:
Steps of how each maneuver of a generated plan $\Pi_{\mathcal{E}}$ is monitored and revised.

In general, the sequence of maneuvers of plan $\Pi_{\mathcal{E}}$ provides a means to facilitate steering information to the robot control system. The relative position information of the currently active maneuver is compared to currently measured object relations to allow the generation of an adequate steering command (4.46, step 4), which will be clarified in the next section.

4.7.4 Robot control

The robot control interface is responsible for generating valid parameter inputs (displacement d, yaw Ψ, curvature c_0) that the robot control law is able to process, when the currently active maneuver of the algorithm shown in figure 4.46 is given. To enable correct robot steering behavior a controller design similar to Zapp [1988] was chosen. According to Zapp [1988] the attempt to adjust state values of displacement d and relative yaw angle Ψ of a given lane as well as adjusting the steering angle λ to equal zero results in correct lane execution of straight lanes. Further, the drift in curved lanes is accounted for by determining feed forward control characteristics for the current steering angle given the bicycle model, which is fed by vehicle wheelbase and current curvature c_0. According to Zapp [1988] the controller implementation accounts for smooth steering maneuvers even at heavily curved lanes and thus builds a robust foundation with respect to robot control.

Instead of modifying the controller design to fit the proposed relative object description, the robot control interface leaves the proposed implementation of Zapp [1988] unaffected, but maps the entities of a relation to the parameters that are required by the control law. Thus, the progress estimation of the maneuver monitor (figure 4.46) is utilized. This estimate is considered to be a first hint and is refined further. When a maneuver A and a hint towards the progress of a maneuver are given, the geometric representation of a maneuver in a given range can be sampled, which is determined by the progressed time t and a predefined progress variance δ that is set to 2 seconds. A sequence of relations

$$\mathcal{R}_A = (\mathfrak{r}(t-\delta)_A, \ldots, \mathfrak{r}(t)_A, \ldots, \mathfrak{r}(t+\delta)_A) \tag{4.102}$$

is generated given the maneuver approximation of equation 4.53. I.e. for each time step t a maneuver A is able to produce a relation $\mathfrak{r}(\alpha(t)_A, \beta(t)_A, d(t)_A)$ that is comprised of the approximated values $\alpha(t)_A$, $\beta(t)_A$ and $d(t)_A$ at time t.

For each successive pair of relations $\mathfrak{r}(t_k)_A$ and $\mathfrak{r}(t_{k+1})_A$ in $\mathcal{R}_A$ the perpendicular relation $\mathfrak{r}_\perp$ according to equations 4.38 to 4.40 from section 4.6.1 can be determined (see figure 4.47 for a visual representation). The relation $\mathfrak{r}_\perp(\alpha_\perp, \beta_\perp, d_\perp)$ with minimal distance entry $d_\perp$ then builds the base for the parameters provided to the vehicle control law. Entity $d_\perp$ is mapped to control parameter displacement d and entity $\alpha_\perp + \beta_\perp$ mapped to yaw-angle Ψ.

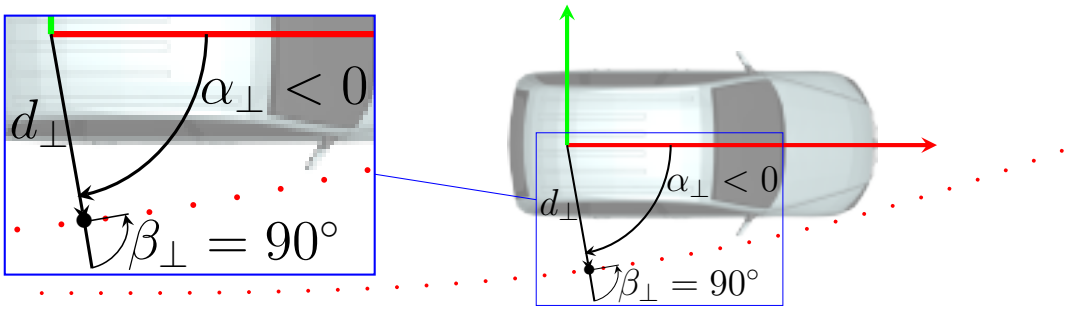


Figure 4.47:
Perpendicular relation $\mathfrak{r}_\perp$ (*black*) of the robot with respect to a sequence of given relations (*red*).

Curvature c_0 cannot be directly computed, but is provided by evaluating those relations of the sequence that the perpendicular relation $\mathfrak{r}_\perp$ was derived from. If relation $\mathfrak{r}_\perp$ is derived from the connection of relations $\mathfrak{r}_t$ and $\mathfrak{r}_{t+1}$, a third relation $\mathfrak{r}_{t+2}$ at time step $t+2$ is sampled. By applying the difference operation $\ominus$ (see

definition 4) to $\mathfrak{r}_t$ and $\mathfrak{r}_{t+1}$ as well as to $\mathfrak{r}_{t+1}$ and $\mathfrak{r}_{t+2}$ the two following relations are obtained:

$$\delta\mathfrak{r}_{t,t+1} = \mathfrak{r}_{t+1} \ominus \mathfrak{r}_t \tag{4.103}$$

$$\delta\mathfrak{r}_{t+1,t+2} = \mathfrak{r}_{t+2} \ominus \mathfrak{r}_{t+1} \tag{4.104}$$

Curvature c_0 is then determined by evaluating the individual components α, β, d of relations $\delta\mathfrak{r}_{t,t+1}$ and $\delta\mathfrak{r}_{t+1,t+2}$ according to equations 4.105 to 4.110. Figure 4.48 provides a visual representation of the determination of the respective values.

$$\gamma = \pi - \mid \alpha_{\delta\mathfrak{r}_{t+1,t+2}} + \beta_{\delta\mathfrak{r}_{t,t+1}} \mid \tag{4.105}$$

$$\gamma_1 = \frac{\pi - \gamma}{2} \tag{4.106}$$

$$\gamma_2 = \frac{\pi}{2} - \gamma_1 \tag{4.107}$$

$$l_1 = \frac{1}{2} d_{\delta\mathfrak{r}_{t,t+1}} \cdot \sin\left(\frac{\gamma}{2}\right) \tag{4.108}$$

$$l = \frac{l_1}{\cos(\gamma_2)} \tag{4.109}$$

$$c_0 = \frac{1}{l} \tag{4.110}$$

Finally, the vector of control parameters cp can be created by the combination of entities $\alpha_\perp$ and $d_\perp$ of $\mathfrak{r}_\perp$ and of curvature c_0 (see equation 4.111).

$$cp = \Big(d,\ \Psi,\ c_0\Big)^T = \Big(d_\perp \cdot -sgn(\beta_\perp),\ \alpha_\perp + \beta_\perp,\ c_0\Big)^T \tag{4.111}$$

The signum function $sgn(\cdot)$ from equation 4.111 guarantees that the distance value becomes negative when the robot is located to the left of the trajectory to follow.

But instead of directly utilizing the control parameters of equation 4.111 as controller input, the control interface makes use of the covariance matrix that is stored with each maneuver A to cope for possible approximation errors. According to equation 4.60 from section 4.6.4 the parametric description of a maneuver $A = ((s_{(\cdot)}, o_{(\cdot)}, a_{(\cdot)}, b_{(\cdot)}, v_{(\cdot)}), \Omega_{\mathbf{P}_A})$ is accompanied by a set $\Omega_{\mathbf{P}_A}$ of covariances, that

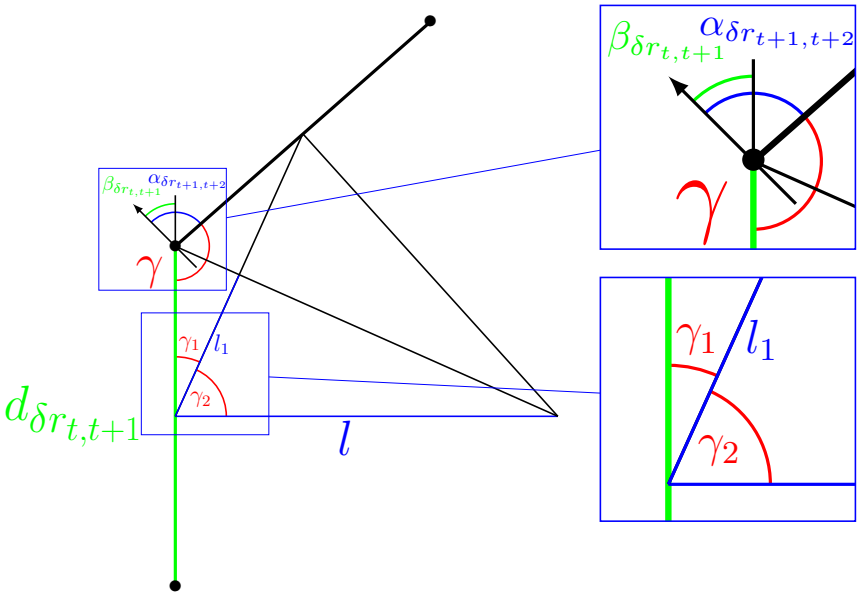


Figure 4.48: Determination of length l.

model the approximation error at each time step t of a maneuver. I.e. for each time step t sample covariance $\mathbf{P}_A(t) \in \Omega_{\mathbf{P}_A}$ can be incorporated into the process of determining the parameters of the control law.

From covariance $\mathbf{P}_A(t)$, entries σ_{d^2}, σ_{α^2} and $\sigma_{d\alpha} = \sigma_{\alpha d}$ can be extracted, that represent the noise of the approximation for the α and d components of a relation. These covariance entries can now be used to define a matrix ω_{cp}, that weights the control parameters cp in order to cope with possible errors in the approximation process (see equation 4.112). I.e. the larger the covariance entry, the less the respective control parameter influences the motion behavior of the robot.

$$\omega_{cp} = \begin{pmatrix} \frac{1}{1+\sigma_{d^2}} & \frac{1}{1+\sigma_{d\alpha}} & 0 \\ \frac{1}{1+\sigma_{\alpha d}} & \frac{1}{1+\sigma_{\alpha^2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.112}$$

As shown in equation 4.112, the weight of parameter c_0 is set to 1, since sample covariance $\mathbf{P}_A(t)$ does not gather information about the correlation between curvature c_0 and α, β, d of a relation. Matrix ω_{cp} is used to weight the entities of the vector of control parameters cp to obtain:

$$cp = \omega_{cp} \cdot cp \tag{4.113}$$

According to [Zapp, 1988], the robot is able to steer relative to heavily curved bends, straight lanes, crossroads or obstacles such as parked cars, when control parameters cp are given. But the controller assumes that the object towards which it controls its relative behavior to is static, which may conflict with most situations the robot encounters. One example is given in figure 4.49, where the robot (*white vehicle*) tries to overtake another vehicle (*yellow*) by taking into account the predicted behavior of the other vehicle (*blue trajectory of the yellow vehicle*). Over time, however, the real trajectory of the vehicle (*red trajectory of the yellow vehicle*) deviates from the predicted trajectory, which forces the robot to adapt its trajectory as well (*red trajectory of the robot*).

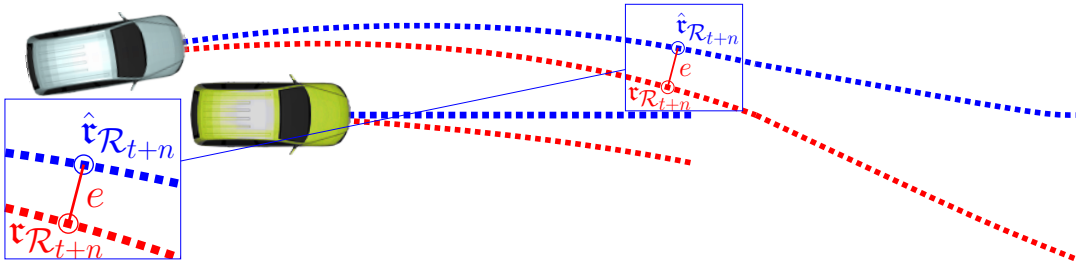


Figure 4.49:
Correction of the robot motion, based on the error e between predicted and measured relation at time step $t + n$.

One possibility for taking into account the dynamic behavior of objects is to track their motion and incorporate their dynamics into the bicycle model. Thereby a possible steering angle and thus a curvature are obtained, which may be included into the the design of the robot controller. But in this thesis, the author chose a more biologically inspired approach instead of introducing more models into the controller design (more model assumptions may even hinder the correct determination of the control parameters, since the bicycle model requires e.g. the length of the respective object, which may be falsely observed). The robot control interface thus introduces a damping vector that accounts for the difference between desired and real change in observation. I.e. the damping vector expresses whether the observation changed in the desired fashion after adjusting the control values.

Each motion of the robot can be expressed by means of a relation. E.g. if relation $\mathfrak{r}_{\mathcal{E}}(t)$ represents the robot location at time step t and a sequence of motions from time step t until time step $t + n$ is applied, goal relation $\mathfrak{r}_{\mathcal{E}}(t + n)$ (location) can be derived by applying the concatenation operation (see definition 3) n-times, as

shown in equation 4.114. Here, relation $\mathfrak{r}_{\mathcal{E}}(t, t+1)$ represents the relative motion of the robot $\mathcal{E}$ between two consecutive points in time.

$$\mathfrak{r}_{\mathcal{E}}(t+n) = \mathfrak{r}_{\mathcal{E}}(t) \oplus \mathfrak{r}_{\mathcal{E}}(t, t+1) \oplus \cdots \oplus \mathfrak{r}_{\mathcal{E}}(n-1, n) \tag{4.114}$$

If the set of relations $\mathcal{R}_t$ that represent the desired future path at current time step t (see e.g. figure 4.49, *blue* trajectory in front of the robot) as well as the robot motion between steps t and $t+n$ is given, a new set of relations $\hat{\mathcal{R}}_{t+n}$ at time step $t+n$ can be determined that is the transformation of $\mathcal{R}_t$ according to the robot motion. Essentially, each relation of sequence $\mathcal{R}_t$ is adjusted according to the relational motion between $\mathfrak{r}_{\mathcal{E}}(t)$ and $\mathfrak{r}_{\mathcal{E}}(t+n)$. I.e each relation $\mathfrak{r}_{\mathcal{R}_t} \in \mathcal{R}_t$ is replaced by a new relation $\hat{\mathfrak{r}}_{\mathcal{R}_{t+n}}$ according to:

$$\hat{\mathfrak{r}}_{\mathcal{R}_{t+n}} = \mathfrak{r}_{\mathcal{R}_t} \ominus \mathfrak{r}_{\mathcal{E}}(t+n) \tag{4.115}$$

To get the set $\hat{\mathcal{R}}_{t+n}$ that represents the corrected desired path according to the robot motion. The entities of $\hat{\mathcal{R}}_{t+n}$ are then exploited according to equations 4.38 to 4.40 from section 4.6.1 in order to get $\hat{\mathfrak{r}}_{\perp}(t+n)$, the corrected perpendicular relation at time step $t+n$. Corrected perpendicular relation $\hat{\mathfrak{r}}_{\perp}(t+n)$ can then be transformed into the parameter space (see equation 4.111) to get a corrected parameter vector $\hat{cp}_{t+n}$.

In case the reference object (e.g. the overtaken vehicle in figure 4.49) moves in the predicted fashion, the parameter vector $\hat{cp}_{t+n}$ that was obtained by adjusting the desired future robot path at time step t by the motion of the robot between steps t and $t+n$, equals parameter vector cp_{t+n} that is gained by recomputing the desired path at time step $t+n$. But often, the reference object does not move according the expectations. Thus, error $e_{cp_{t+n}}$ can be determined according to equation 4.116 that compares $\hat{cp}_{t+n}$ to cp_{t+n}.

$$e_{cp_{t+n}} = cp_{t+n} - \hat{cp}_{t+n} \tag{4.116}$$

Error $e_{cp_{t+n}}$ represents the difference between the corrected control parameters and the control parameters that are measured at time step $t+n$. Figure 4.49 provides a visual representation of the error that arises, when a corrected relation $\hat{\mathfrak{r}}_{\mathcal{R}_{t+n}}$ is

compared to a recomputed relation $\mathfrak{r}_{\mathcal{R}_{t+n}}$ that incorporates the altered motion of the reference object already. The error is then added to the control parameter vector by extending equation 4.113, as shown in equation 4.117.

$$cp_{t+n} = \omega_{cp_{t+n}} \cdot cp_{t+n} + \omega_e \cdot e_{cp_{t+n}} \tag{4.117}$$

In this regard, ω_e represents a weighting matrix with secondary diagonal elements equal zero and main diagonal entries that range from 0 to 1 in order to weight the error values of $e_{cp_{t+n}}$. Thereby, the control parameter vector cp_{t+n} at time step $t+n$ can be corrected to allow the incorporation of the motion of the reference object as well.

4.8 Results

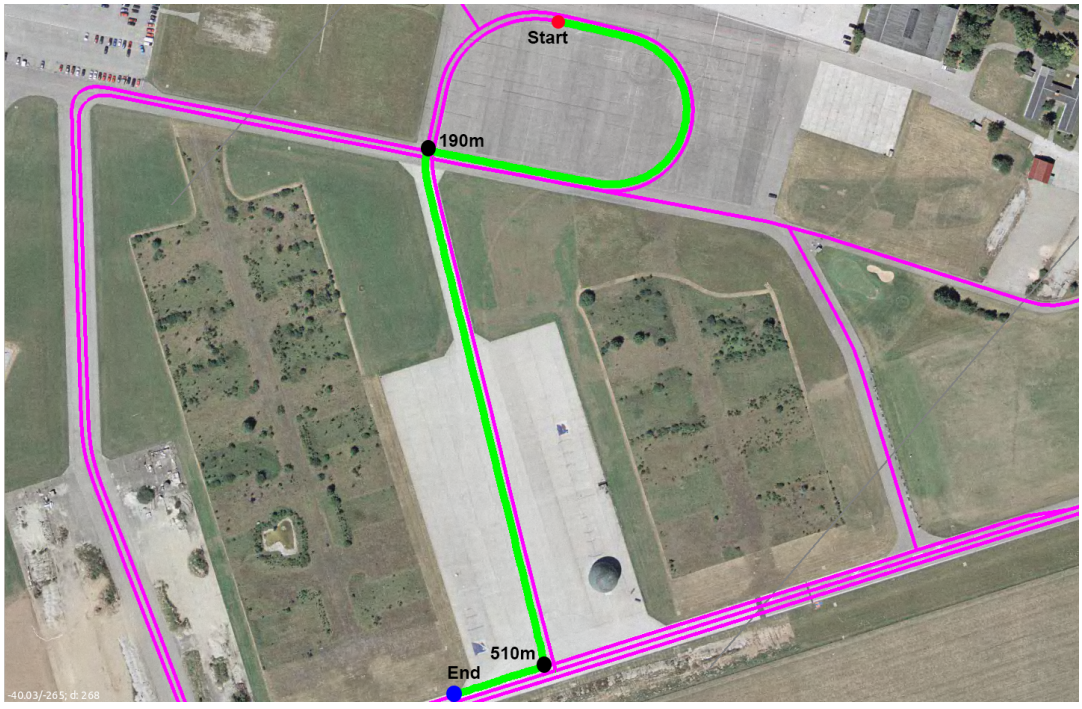


Figure 4.50:
Aerial image of the old air field of the UniBwM enriched with GPS road information (*magenta*). One of the paths that was driven autonomously by MuCAR-3 with the object-related-navigation approach is colored green.

The object-related navigation approach was tested with respect to various objects in different scenarios to evaluate its capabilities with regard to prediction and navigation. They were conducted on the old air field of the UniBwM (see figure 4.50), which

provides a set of possible traffic situations ranging from high speed highway driving to turning on crossroads. The tests are focused on lane following and overtaking maneuvers as well as on crossroad turning maneuvers. Each evaluation provides a visual representation of the maneuver as well as a quantitative error analysis. The error represents the dissimilarity in $[m]$ between the expected and the real position of the robot. Given current time step t_0 and a certain prediction horizon $\delta\, t$, say e.g. 2 seconds, the predicted position of the robot $\boldsymbol{p}(t_1)^*$, with $t_1 = t_0 + \delta\, t$ can be determined. Given the robot's motion, the predicted position is then corrected. At time step t_1 the position $\boldsymbol{p}(t_1)$ is measured and compared to $\boldsymbol{p}(t_1)^*$ to yield error e according to:

$$e = d(\boldsymbol{p}(t_1), \boldsymbol{p}(t_1)^*), \tag{4.118}$$

with $d(\cdot, \cdot)$ being the Euclidean distance between two points.

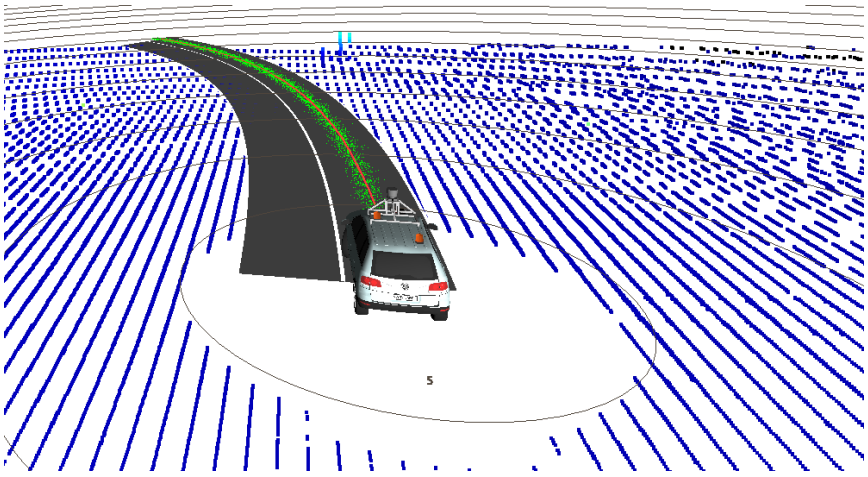


Figure 4.51:
Results of applying the model of maneuver *"follow"* to a curved lane. The result of the prediction is visualized via a line sequence (*red*) that represents the mean of the maneuver as well as sampled points (*green*) that represent the error covariance. The visualization results from a model that was approximated with data that represents typical lane following behavior.

The first test covered scenarios, in which the robot followed differently curved lanes in order to evaluate the accuracy of the *"follow"* maneuver in relation to the object *"lane"*. In figures 4.51 and 4.52 the results are visualized. Each figure shows the robot (*white vehicle*) surrounded by the current point cloud of the LiDAR (*blue to yellow colored dots*), the lane representation (*grey,white*), other objects (*green bounding boxes*) and the predicted maneuver (*green*). The visualization of the predicted maneuver represents the two parts of the maneuver approximation as

described in section 4.6.3, the parametric course of the maneuver supplemented by the samples of the maneuver error covariance (*green dots*).

The maneuver model of figure 4.51 was approximated given data that represents typical lane following behavior, while the maneuver model in figure 4.52 was approximated with lane following maneuvers that describe an oscillation between the left and right border of a lane instead of lane following maneuvers that follow the lane's center, exclusively.

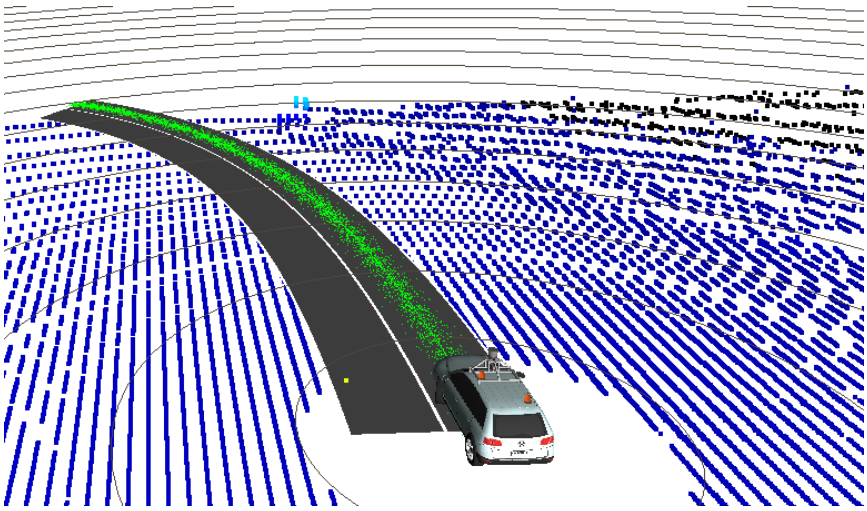


Figure 4.52:
Results of applying the model of maneuver *"follow"* to a curved lane. The approximation had to cope with data that represents an oscillating motion on the lane, which results in large uncertainties.

Thus, the model of figure 4.51 predicts the future motion of the robot to be close to the centre of the lane on average with a relatively small uncertainty, while the visualized model of figure 4.52 contains large uncertainties with respect to the possible future positions of the robot.

Figures 4.53 and 4.54 describe the prediction error (see equation 4.118) of the visualized models. The prediction errors shown in figure 4.53 result from a lane following maneuver at a straight lane that is predicted, given prediction horizons of 0 s (*blue*), 1 s (*green*), 2 s (*black*) and 3 s (*brown*). I.e. the graphs show the difference between the predicted position at the respective time horizon and the real robot position at that time horizon, both transformed in a common coordinate frame. The abscissa describes the driven distance, when a reference point is given at the start of the maneuver. The ordinate describes the difference between the predicted and the real pose of the robot. This difference combines the error that is made by the prediction procedure and the error that is made by the lane curvature estimation.

The curve that describes the prediction error of 0 s (*blue*) provides insight into the general discrepancy between maneuver approximation and the driven maneuver, independent of the error covariance of the approximation. I.e. during the lane following maneuver the manually driven vehicle had a maximum lateral displacement of approximately 0.2 m. However, the predictions at 1 s (*green*), 2 s (*black*) and 3 s provide more meaningful information regarding the prediction capabilities of the model.

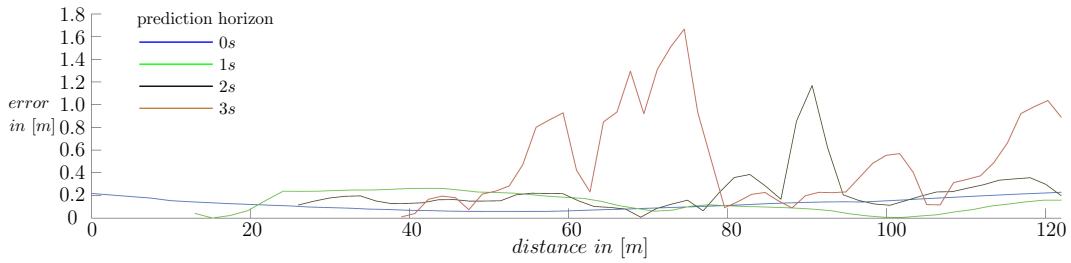


Figure 4.53:
The respective prediction errors at a straight lane with prediciton horizons of 0 s (*blue*), 1 s (*green*), 2 s (*black*) and 3 s (*brown*).

In general, the error became larger when the prediction horizon became larger. While the maximum dissimilarity at a prediction horizon of 1 s was approximately 0.2 m, which is comparable to the dissimilarity at 0 s, the deviation between predicted and measured robot pose at 3 s amounted to approximately 1.8 m. However, these huge dissimilarities were not caused by the motion model alone. As the curve of the prediction error at e.g. 1 s illustrates, its values are even lower than the optimal prediction at 0 s (which represented the closest distance between robot and maneuver center line). The reason was the lane estimation itself. While a small error in e.g. the lane curvature estimation did not have much impact on the predicted position within a short prediction horizon, longer prediction horizons caused the distance between predicted and measured robot pose to deviate significantly.

The impact of the lane estimation was even obvious in case of lane following maneuvers with respect to curved lanes as illustrated in figure 4.54. While the deviation between predicted and measured robot position at a prediction horizon of 0 s (*blue*) was similar to the deviation at straight lanes, prediction horizons of 1 s (*green*), 2 s (*black*) and 3 s (*brown*) caused deviations of up to 6 m.

The maneuvers started in the center of the bend of figure 4.50 (*green path at top right*) with the robot following the lane up to the next intersection. It is clearly visible that the quality of the prediction decreased greatly during the traversal of the bend, when the prediction horizon was larger. In the case of a prediction horizon of

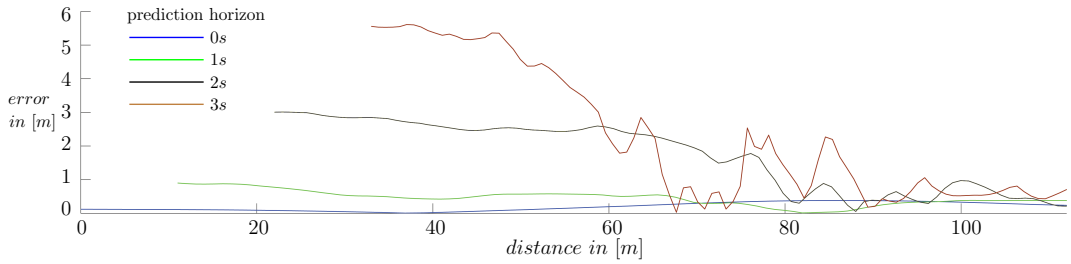


Figure 4.54:
The respective prediction errors at a curved lane with prediciton horizons of 0 s (*blue*), 1 s (*green*), 2 s (*black*) and 3 s (*brown*).

3 s, for instance, future robot positions were estimated based on the curvature and change of curvature of the bend, although the bend already merged into a straight lane within the respective prediction horizon. From the time that the robot entered the straight part of the lane, the prediction error decreased to close to zero again.

In contrast to the partially large prediction errors at curved lanes, the dissimilarity between predicted and measured robot positions at crossroads was comparably small as shown in figures 4.55 to 4.57. Figure 4.55 provides an insight into the approximation for the maneuver *"turn left"* with respect to the object *crossroad* from the viewpoint of the robot.

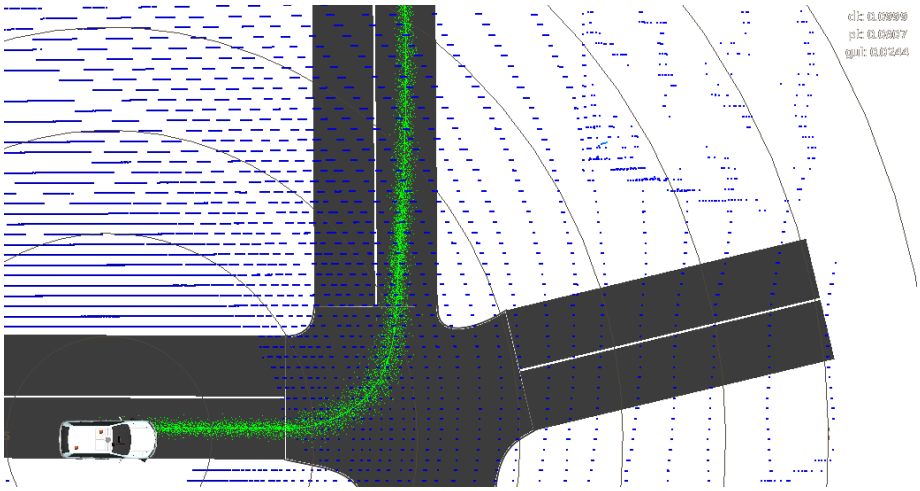


Figure 4.55:
Result of applying maneuver *"turn left"* to a crossroad structure from a robot-centric perspective. The predicted trajectory is visualized via a linesequence (*green*) that represents the mean of the maneuver as well as sampled points (*green*) that represent the error covariance.

The data to be approximated contained various turn maneuvers of the robot with the entry point to the crossroad being one of its branches. The crossroad that was used for the practical evaluation of the robot's behavior, however, was not part of

this data set. As shown in figure 4.55, the approximation represents the uncertainty of a turn maneuver well. While the maneuver uncertainty with respect to the entry and exit lanes is relatively small, the robot positions in the center of the crossroad are uncertain.

As proposed in section 4.7 it is the purpose of the maneuver models to predict the possible behaviour of other objects as well. In this context, figure 4.56 shows the predicted trajectory of maneuver *"turn left"* with respect to a crossroad from the perspective of another traffic participant. I.e. the figure shall merely exemplify the application of a model to another traffic participant without any connection to its true future behavior. The traffic participant is represented in the form of a bounding box model (*green*) with the maneuver prediction ranging from the center of the object to the exit of the respective branch of the crossroad. The predicted object positions behaved in the same manner as the predicted positions of the robot. While significant uncertainties were present in the center of the crossroad during the left turn, the position of the object was predicted with high certainty during the traversal of the branches of the crossroad.

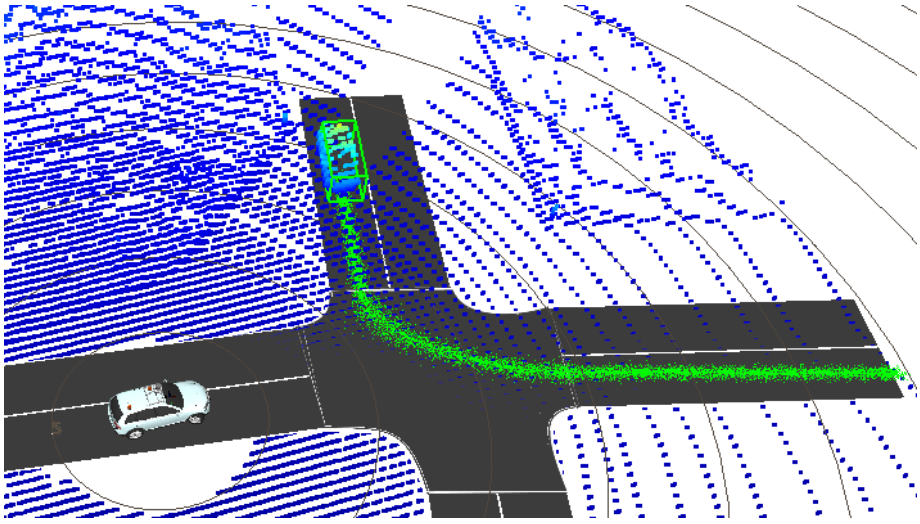


Figure 4.56:
Result of applying maneuver *"turn left"* to a crossroad structure from an object-centric perspective.

With respect to the quality of the prediction, figure 4.57 provides insight into the prediction error of a left turn maneuver at a crossroad. The behaviour of another traffic participant was predicted and evaluated with the same methodology that was employed to determine the dissimilarity of predicted and actual robot position in figures 4.53 and 4.54. The graphs from figure 4.57 represent four prediction horizons at 0 s (*blue*), 2 s (*green*), 4 s (*brown*) and 6 s (*magenta*). As mentioned

previously, the abscissa describes the driven distance with respect to a reference point at the start of the maneuver, while the ordinate describes the difference between the predicted and the real position of the robot. The prediction dissimilarity at a horizon of 0 s represents how well the center trajectory of the maneuver model was able to represent the real trajectory of the car. The largest error occurs in the center of the crossroad, because the tracked vehicle made an even sharper turn than anticipated by the model. Closer to each of the two branches, i.e. the entry and the exit branch, the dissimilarity became smaller, since the vehicle merged into the lane again. As was the case for the prediction results of the lane following maneuver, the dissimilarity between predicted and actual robot position increased with larger prediction horizons. In strong contrast to the results of the lane following maneuver, however, the dissimilarity did not increase significantly due to the more stable crossroad representation. The predictions were close to the optimal model (see e.g. at 11 m when the curve of the prediction horizon of 2 s touches the curve of the prediction horizon of 0 s). The dissimilarities at the end of the maneuver resulted from more moderate acceleration of the vehicle than had been anticipated.

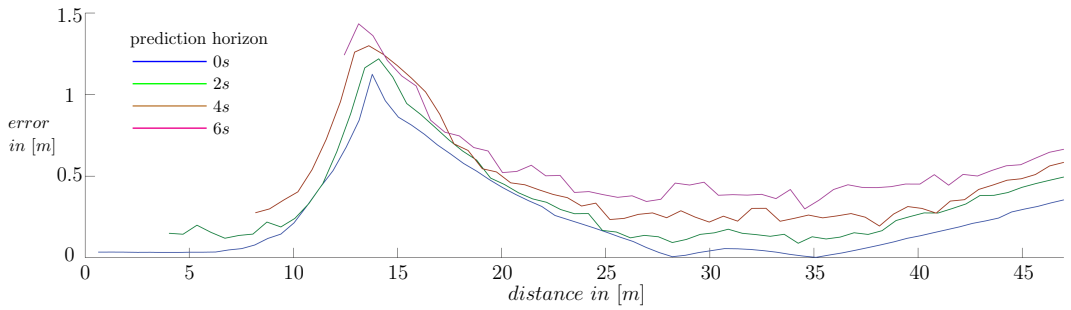


Figure 4.57:
The respective prediction errors with respect to a left turn on the crossroad with prediction horizons of 0 s (*blue*), 2 s (*green*), 4 s (*brown*) and 6 s (*magenta*).

Aside from the prediction of possible future object positions, section 4.7 introduced how a maneuver is able to generate an observation probability. Figures 4.58 and 4.59 represent the application of the methodology from section 4.7 in order to determine the most probable future maneuver of the robot as well as of another traffic participant. The colours of the trajectories indicate the likelihood of a maneuver ranging from green (high probability) to red (low probability).

Figure 4.58 shows how the robot approaches the crossroad. Although the robot is not even close to its centre, the probability of the maneuver *straight* decreases already. This is caused by the temporal evaluation of each maneuver. The data contained velocities of up to $5m/s$ for the *straight* maneuver, while the largest velocities with respect to the turn maneuvers were between $2m/s$ and $3m/s$. Thus,

robot pose predictions of maneuver *straight* that were carried out in previous time steps are not able to explain the measured robot poses. This leads to a decrease in the probability of continuing the maneuver *straight* upon approaching a crossroad. A left turn is as likely to occur as a right turn at this point of the maneuver.

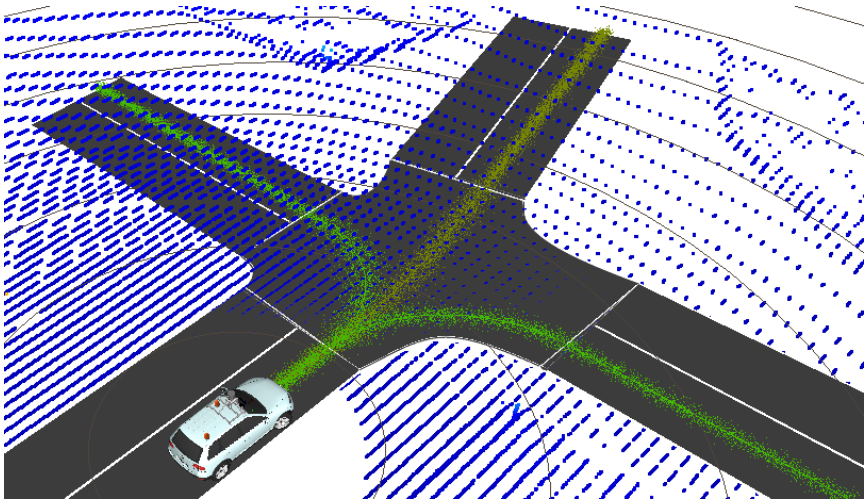


Figure 4.58:
Application of a set of maneuvers to a crossroad from the perspective of the robot in order to obtain the most probable future behavior. The probability of a maneuver is coded by the color of the respective trajectory ranging from red (*low probability*) to green (*high probability*).

Given the same models, figure 4.59 shows the likelihood of the maneuvers with respect to another traffic participant (*green bounding box*). The relations of the tracked vehicle (*green bounding box*) relative to the crossroad are evaluated in the same manner as the relations of the robot. This gives rise to results that were similar to those obtained from the evaluation of the robot's motion. When the vehicle enters the crossroad more information on the possible turn maneuver becomes available. Instantaneously, the probability of the *turn right* maneuver drops, because the real relation with respect to the crossroad deviates strongly from the predicted relation. Even the *straight* maneuver becomes more likely, because its predicted relation resembles the objects' relation relative to the crossroad more closely than the relation of the right turn maneuver.

In a similar way to how the motion between a vehicle and a road structure is described by relations, the motion between two traffic participants can be modelled as shown in figure 4.60. It shows the predicted trajectory of the robot (*green line sequence*), when the most probable motion of the vehicle to the front of the robot (*green bounding box*) is provided. The motion of the vehicle depends on the course of the road, which causes the predicted trajectory to align to it. Figure 4.60 also shows the

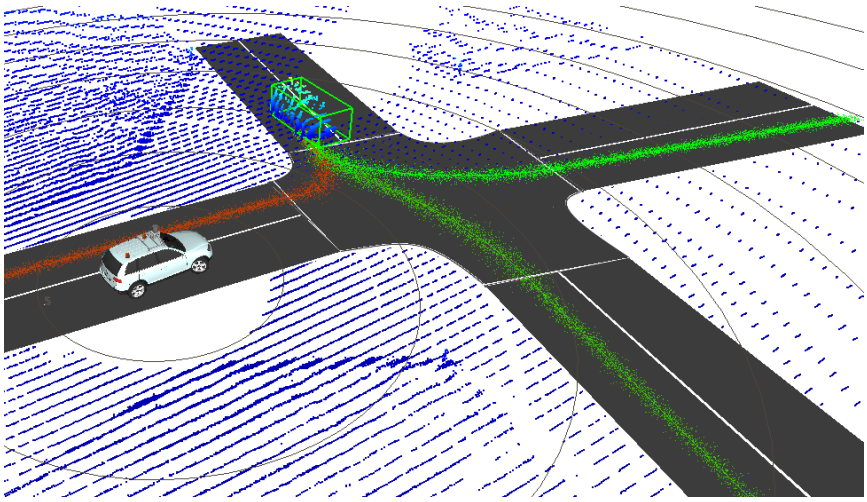


Figure 4.59:
The application of a set of maneuvers to a crossroad from the perspective of another traffic participant in order to obtain its most probable future behavior. The probability of a maneuver is coded by the color of the respective trajectory ranging from red (*low probability*) to green (*high probability*).

dimensions of the trajectory in case of an overtaking maneuver. The final relation lies far beyond the detection horizon of the lane, which added uncertainty to the predicted motion of the robot.

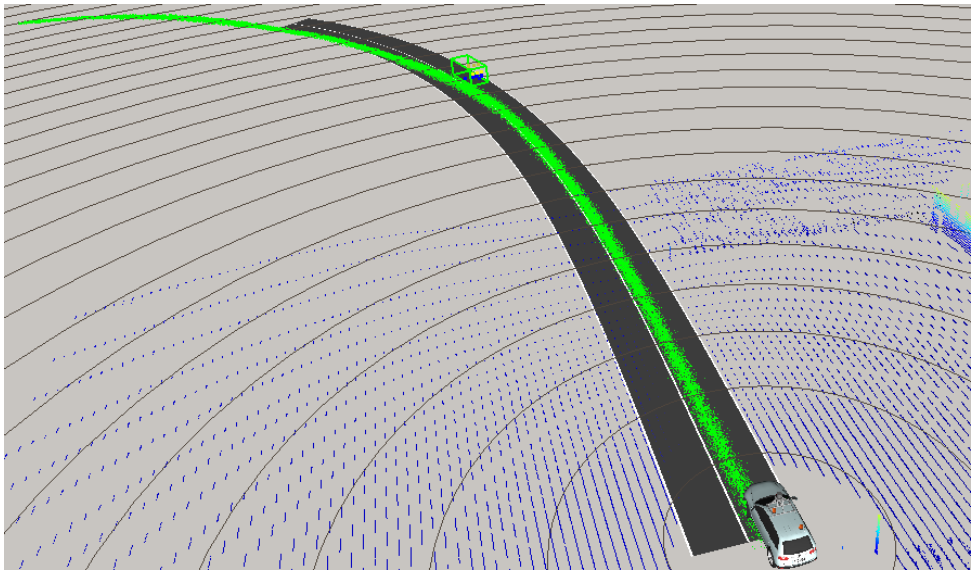


Figure 4.60:
The figure presents a prediction of an overtaking maneuver given the most probable behavior of the vehicle to overtake.

A qualitative and quantitative evaluation of the maneuver *overtake* relative to the ego-vehicle is presented in figures 4.61 to 4.63. The set of figures in 4.61 and 4.62 represent the overtaking maneuver between another traffic participant (*green*

bounding box) and the robot (*white*) from two different perspectives, the local and the global one. The local perspective represents the projection of the relations of a maneuver with respect to the object to overtake, when the motion of the overtaken vehicle is not incorporated. It is particularly important for robot steering as described in section 4.7.4, since the desired relation that is provided by the model is evaluated against the perceived relation provided by the perception systems. The figure shows that the application of the maneuver from an object-centric perspective produces promising results, although the model had been approximated based on robot-centric motion data, i.e. from data that described the motion of the robot relative to an object. The desired position of the maneuver approximation (the fist point of the trajectory) always fits the current position of the vehicle, which implies that the model behaves like a real human driver in the same situation.

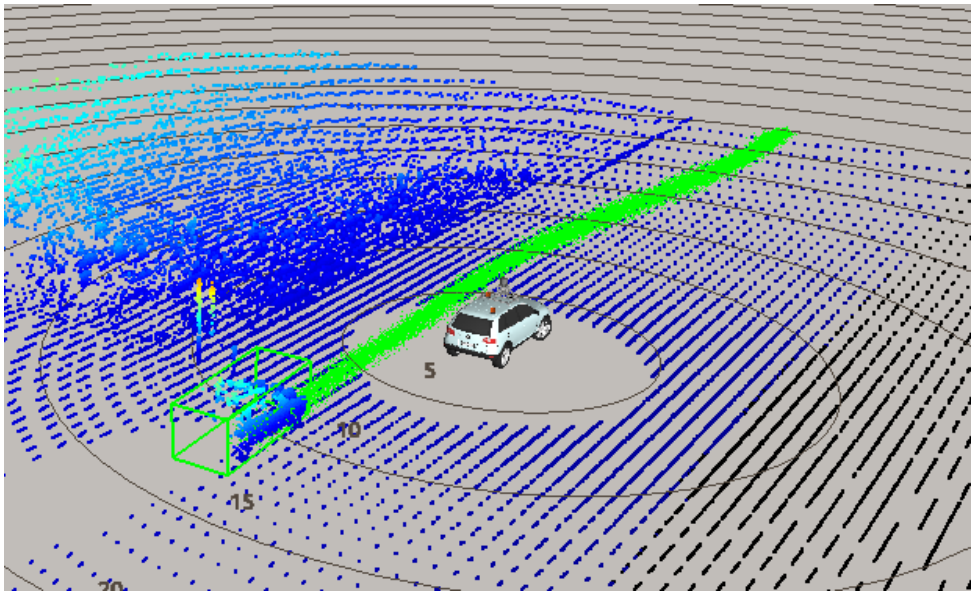


Figure 4.61:
The predicted trajectory of the robot (*green line sequence*) as well as its uncertainty (*green samples*), which describes the relative motion between two objects without the incorporation of the global motion of the overtaken object.

While the local representation was responsible for robot steering, the global perspective allows for the evaluation of predicted object trajectories. Figure 4.62 presents the overtaking maneuver from figure 4.61 when the motion of the robot is taken into consideration. Although the maneuver approximation remains untouched, the shape of the trajectory changes in order to adapt to the predicted motion of the robot as well (road information it not accessible, so the prediction of the robot is based on its past motion). For instance, while in figure 4.61 the trajectory ends at ca. $35m$, the trajectory in figure 4.62 shows a motion to the right at the same distance.

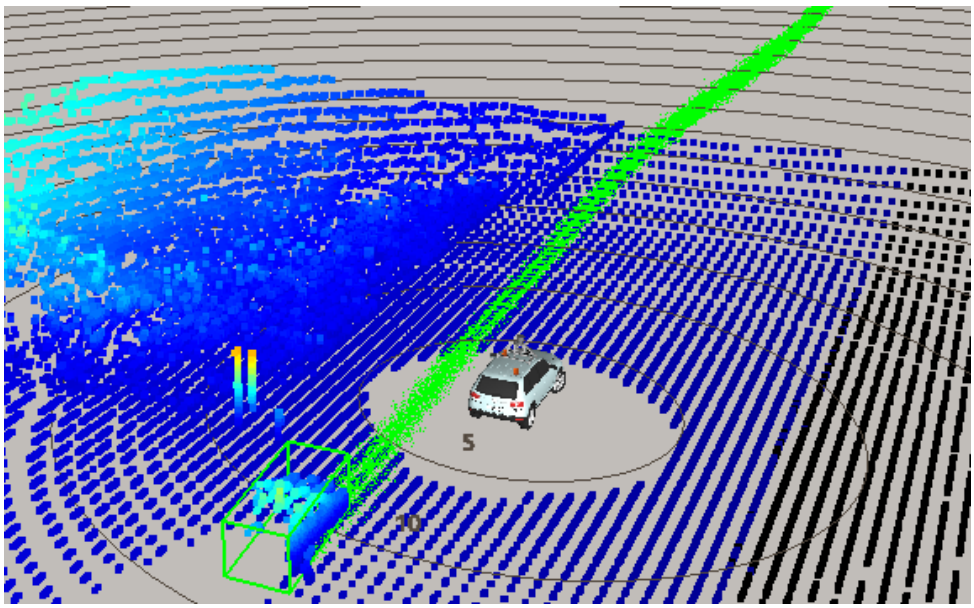


Figure 4.62:
The predicted trajectory of the robot (*green line sequence*) as well as its uncertainty (*green samples*), which describes the relative motion between two objects with the incorporation of the global motion of the overtaken object. The time step is comparable to the one from figure 4.61. But since the motion of the overtaken object is incorporated, the shape of the trajectory changes. For instance, while in figure 4.61 the trajectory ends at ca. $35m$, the trajectory in figure 4.62 shows a movement to the right at the same distance.

The quantitative evaluation of the overtaking maneuver is presented in figure 4.63. Five prediction horizons at 0 s (*blue*), 2 s (*green*), 4 s (*brown*), 6 s (*black*) and 8 s (*magenta*) were evaluated. The abscissa describes the driven distance with respect to a reference point at the start of the maneuver and the ordinate describes the difference between the predicted and the real position of the overtaking vehicle. The curve that represents the error at a prediction horizon of 0 s described the maneuver quite well. During the approach the deviation was relatively small with 0.2 m. Subsequently it increased up to circa 0.4 m when the vehicle steered to the left in order to overtake the robot. Gradually the error decreased again down to 0 m when the vehicle drove side by side with the robot. Since the vehicle steered to the right after it had passed the robot, the error increased again up to a maximum of approximately 0.4 m. The curves of all other prediction horizons follow this course of events with a maximum error of approximately 1.3 m. But each prediction horizon estimated the critical phases of the maneuver at approximately the same time, i.e. that the object would overtake the robot at a distance of about 80 m to 100 m, which shows the robustness of the prediction.

In order to evaluate the steering behavior of the object-related navigation approach, the behaviour of the autonomously driven robot will now be compared to the

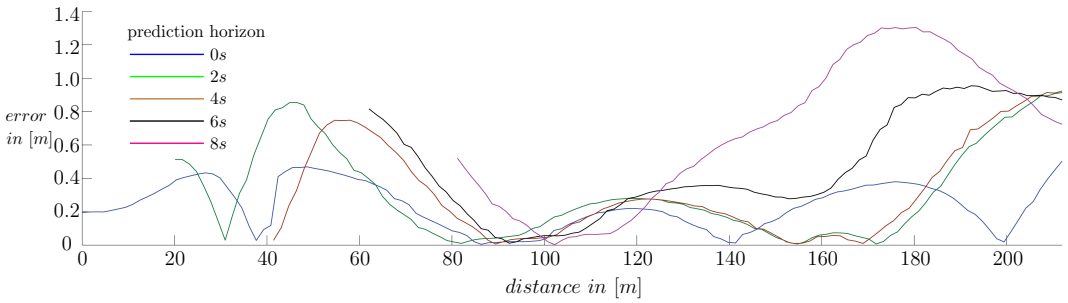


Figure 4.63:
The respective prediction errors with respect to the overtaking maneuver from figure 4.62 with prediction horizons of 0 s (*blue*), 2 s (*green*), 4 s (*brown*), 6 s (*black*) and 8 s (*magenta*).

behaviour of a vehicle that is steered by a human driver. For this purpose, an approximately 600 m long track was driven on the testing grounds of the UniBwM (see figure 4.50 (*green path*)). It contained straight as well as curved lanes and two crossroads. At the first crossroad a left turn had to be executed, followed by a right turn at the second crossroad. In order to enable both drivers (the human and the robot) to complete the track the exact same commands where given to them: *"follow the lane"*, *"turn left on the crossroad"*, *"follow the lane"*, *"turn right on the crossroad"* and finally *"follow the lane"* again. I.e. instead of providing spoken language to the human driver and some kind of coordinate based representation of the track to the robot, both participants received the same information.

Figures 4.64 and 4.65 show the results of the drive. The abscissae of both figures describe the driven distance with respect to a reference point at the start of the track (*red* dot at the top of figure 4.50), while the ordinate describes the distance to a GPS reference track consisting of the center line of the respective lanes. The deviation of the manual drive from the GPS reference track is depicted in figure 4.64. Up until the first crossroad (the first large peak in the graph at ca. 180m) the driving behaviour was somewhat unsteady due to the traversal of the large bend. Often some small adjustments where required in order to stay in the lane properly. However, after the first crossroad had been passed, which caused the deviation to rise greatly, the driving behaviour became more stable, possibly because it was only required to follow a straight lane. Up until the next crossroad, at ca. $510m$ the deviation dropped to $0m$ two times (at ca. $400m$ and $500m$). This can be considered a characteristic behaviour whenever the vehicle traverses the center of the lane. At $500m$, for example, the vehicle steered from the left-hand side of the lane towards the right-hand side of the lane in order to turn at the crossroad. At the end of the track the right turn (second peak at ca $510m$) caused the deviation

to increase again, especially during the traversal of the crossroad's center, while the approach to the subsequent lane center caused the deviation to diminish again.

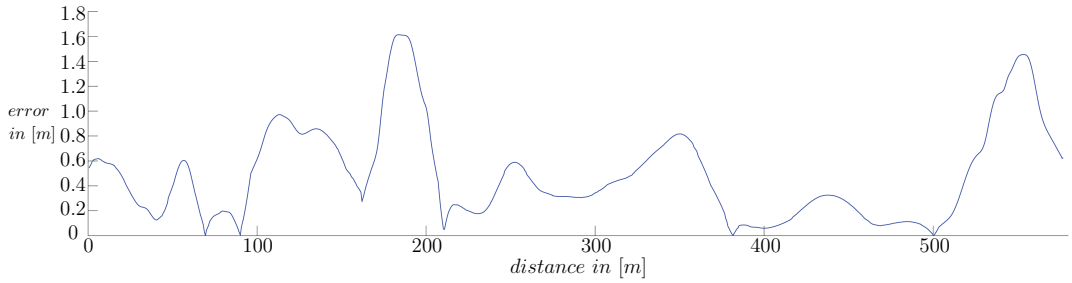


Figure 4.64:
Deviation between a GPS reference track and the track that was driven manually by a human driver. The instructions given to the driver were: "follow the lane, turn left on the crossroad, follow the lane, turn right on the crossroad". The equivalent GPS-path is shown in figure 4.50 (*green*).

In comparison to the manual drive, figure 4.65 provides insight into the progress of the deviation of the autonomous drive. In general, the driving behaviour was more stable and more comfortable. Even in the bend at the start of the track the deviation from the reference track does not change significantly. The same statement can be made with respect to the straight part of the track between the two crossroads. The deviation at the crossroads itself was comparable to the deviation of the manual drive. Altogether, the course of the autonomous drive was comparable to the manual drive, which suggests that the behaviour of the robot was quite similar to the behaviour of the human steered car.

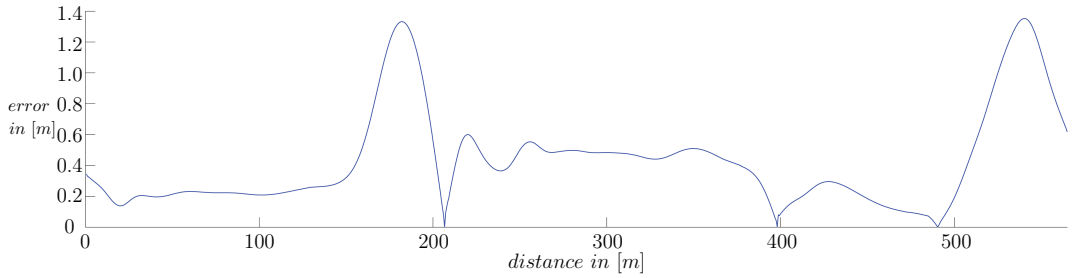


Figure 4.65:
Deviation between a GPS reference track and the track that was driven autonomously by MuCAR-3. The instructions given to the system were: "follow lane, turn left on the crossroad, follow lane, turn right on the crossroad". The equivalent GPS-path is shown in figure 4.50 (*green*).

Far and foremost, the autonomous drive showed how similar the robot behaved in comparison to a human driver. During the drive the robot initiated a new maneuver so that smooth transitions between the maneuvers were executed. The interface to the robot allowed to command the vehicle in a human like way, to e.g. *"turn*

left", while it also enables a human observer to understand the action the robot was currently performing. The approach in general is able to predict the behavior of other traffic participants by observing their relative behavior to their surroundings. In the same way it is able to reason about the consequences of its own actions by predicting them into the future. The object-related navigation approach enables the robot to be steered by sole comparison of the perceived relations between the robot and objects in the environment and desired relations provided by the maneuver approximations. Although the internal approximations are quite simple, they are able to represent most of the required maneuvers, such as *follow lane*, *turn left* or *overtake vehicle*. Those models are quick and easy to evaluate, while providing promising prediction results at the same time.

However, the models do not account for the initial position of the object to predict. Figure 4.58 for example, shows the possible future trajectories of the robot in a crossroad scenario. But the predictions do not start at the rear axis of the robot, but represent a generalisation of all turn motions the maneuver models were approximated with, which typically start in the center of the lane on average. Although the variance stored in the models allows for slight variations from the optimal trajectory of a maneuver, the initial position of the object is not considered as starting point of the predicted trajectory. Large scale prediction and reasoning was affected by the visibility limit of the perception system as only the next object was visible. Prediction in general was influenced by the local nature of the information provided to the system. While e.g. a clothoid allows to steer a robot based on information provided by the perception system it fails to describe a course of a road due to its local assumptions.

5 Summary

In this dissertation an approach for mobile robot navigation and object prediction was presented that is called object-related navigation. In contrast to most state-of-the-art navigation algorithms the frame of reference in object-related navigation is shifted from the global to the local perspective, in which motions do not describe the motion within a map but relative to perceived objects. Plans do not describe a global trajectory but the concatenation of actions in relation to objects. In this manner plans in the form of *"follow lane, turn right on crossroad, follow lane, then turn left on crossroad, ... "* become possible, which facilitate an easier human-machine-interaction. The system is able to comprehend and execute natural language plans and provides feedback regarding possible future actions in natural language form. The reactive nature of each action allowed for the adaptation of the motion behavior in case the reference object did not move according to the expectation. I.e. instead of disregarding the entire plan, small deviations between perceived and modelled motion were covered by the actions without influencing the long-term plan at all. To model the motion relative to other objects a procedure was presented to approximate a general model out of a multitude of motion sequences that describe the respective motion, e.g. *"overtaking"* or *"turn left"*.

The results showed how the approach enabled the robot MuCAR-3 to drive on a test track in accordance with instructions that were provided in natural language only. Furthermore, the prediction capabilities of the approach and their quality were presented in various scenarios.

A short demonstration of the approach is available on youtube:

https://youtu.be/7Uw6Ak45CW8

A Poses and homogeneous transformation matrices

A.1 Homogeneous transformation matrices

A homogeneous transformation matrix is defined according to equation A.1 and describes a 3×3 rotation and 3×1 translation.

$$\mathbf{H} = \begin{pmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix} \tag{A.1}$$

A.2 Pose to homogeneous transformation matrix

A pose $\mathbf{P}(x, y, z, \Phi, \Theta, \Psi)$ contains Cartesian coordinates x, y, z as well as roll angle Φ, pitch angle Θ, yaw angle Ψ. A homogeneous transformation matrix can be generated that represents the transformation between pose $\mathbf{P}(0, 0, 0, 0, 0, 0)$ and pose $\mathbf{P}(x, y, z, \Phi, \Theta, \Psi)$ as follows:

A rotation of Φ radians about the x-axis is defined as:

$$\mathbf{R}_x(\Phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Phi) & -\sin(\Phi) & 0 \\ 0 & \sin(\Phi) & \cos(\Phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

, a rotation of Θ radians about the y-axis is defined as:

$$\mathbf{R}_y(\Theta) = \begin{pmatrix} \cos(\Theta) & 0 & \sin(\Theta) & 0 \\ 1 & 0 & 0 & 0 \\ -\sin(\Theta) & 0 & \cos(\Theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and a rotation of Ψ radians about the z-axis is defined as:

$$\mathbf{R}_z(\Psi) = \begin{pmatrix} \cos(\Psi) & -\sin(\Psi) & 0 & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Furthermore, matrix $\mathbf{T}(x, y, z)$ describes the translation along the axes of the coordinate frame.

$$\mathbf{T}(x, y, z) = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 1 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Homogeneous transformation matrix $\mathbf{H}$ finally describes the sequence of rotations around the z, y and x-axis combined with transformation $\mathbf{T}(x, y, z)$ according to formalization A.2.

$$\mathbf{H} = \mathbf{R}_x(-\Phi) \cdot \mathbf{R}_y(-\Theta) \cdot \mathbf{R}_z(-\Psi) \cdot \mathbf{T}(-x, -y, -z) \tag{A.2}$$

A.3 Homogeneous transformation matrix to pose

The computation of the Euler angles from a rotation matrix is carried out according to [Slabaugh, 1999] (see formalization A.3).

$$\begin{aligned} \Theta &= -asin(\mathbf{H}_{13}) \\ \Phi &= atan2(\frac{\mathbf{H}_{23}}{\cos(\Theta)}, \frac{\mathbf{H}_{33}}{\cos(\Theta)}) \\ \Psi &= atan2(\frac{\mathbf{H}_{12}}{\cos(\Theta)}, \frac{\mathbf{H}_{11}}{cos(\Theta)}) \end{aligned} \tag{A.3}$$

If $\cos(\Theta) = 0$, then $\mathbf{H}_{13}$ is either 1 or -1 and this so called Gimbal lock needs to be addressed. Thus, the yaw angle can be set to any value, for instance $\Psi = 0$. If the matrix entity $\mathbf{H}_{13}$ equals -1 then angles Θ and Φ are determined as follows:

$$\begin{aligned} \Theta &= \frac{\pi}{2} \\ \Phi &= \Psi + atan2(\mathbf{H}_{21}, \mathbf{H}_{31}) \end{aligned} \tag{A.4}$$

Otherwise, if the matrix entity $\mathbf{H}_{13}$ equals 1, then both angles are determined according to equation A.5.

$$\begin{aligned} \Theta &= -\frac{\pi}{2} \\ \Phi &= -\Psi + atan2(-\mathbf{H}_{21}, -\mathbf{H}_{31}) \end{aligned} \tag{A.5}$$

Finally, Cartesian coordinates x, y, z are obtained by transforming origin $\begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^T$ under consideration of matrix inverse $\mathbf{H}^{-1}$ as shown in equation A.6

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix}^T = \mathbf{H}^{-1} \cdot \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^T \tag{A.6}$$

to yield pose $\mathbf{P}(x, y, z, \Phi, \Theta, \Psi)$.

B Bayes filter derivation

The state $\hat{\boldsymbol{x}}_{V_k}$ can be estimated, given k measurements $\mathbf{Y}_{V_k}$ by conditioning the rightmost term of 4.69 on $\mathbf{Y}_{V_{k-1}}$ and applying Bayes rule.

$$p(\hat{\boldsymbol{x}}_{V_k} \mid \mathbf{Y}_{V_k}) = \eta \cdot p(\boldsymbol{y}_{V_k} \mid \hat{\boldsymbol{x}}_{V_k}) \cdot p(\hat{\boldsymbol{x}}_{V_k} \mid \mathbf{Y}_{V_{k-1}}) \tag{B.1}$$

The denominator from Bayes rule is absorbed into a normalizing constant and is written as η. By subsequently applying the law of Total Probability, state $\hat{\boldsymbol{x}}_k$ is conditioned on state $\hat{\boldsymbol{x}}_{k-1}$.

$$p(\hat{\boldsymbol{x}}_{V_k} \mid \mathbf{Y}_{V_k}) = \eta \cdot p(\boldsymbol{y}_{V_k} \mid \hat{\boldsymbol{x}}_{V_k}) \cdot \int p(\hat{\boldsymbol{x}}_{V_k} \mid \hat{\boldsymbol{x}}_{V_{k-1}}) \cdot p(\hat{\boldsymbol{x}}_{V_{k-1}} \mid \mathbf{Y}_{V_{k-1}}) d\,\hat{\boldsymbol{x}}_{V_{k-1}} \tag{B.2}$$

Essentially, state $\hat{\boldsymbol{x}}_{V_k}$ is estimated by accounting for the probability of each predecessor $p(\hat{\boldsymbol{x}}_{V_{k-1}} \mid \mathbf{Y}_{V_{k-1}})$, when $k-1$ measurements, transition probability $p(\hat{\boldsymbol{x}}_{V_k} \mid \hat{\boldsymbol{x}}_{V_{k-1}})$ from state $\hat{\boldsymbol{x}}_{V_{k-1}}$ to state $\hat{\boldsymbol{x}}_{V_k}$, i.e. the dynamic motion model, as well as the probability $p(\boldsymbol{y}_{V_k} \mid \hat{\boldsymbol{x}}_{V_k})$ of observing measurement $\boldsymbol{y}_{V_k}$ at state $\hat{\boldsymbol{x}}_{V_k}$ are given. In this manner, state $\hat{\boldsymbol{x}}_{V_k}$ can be considered to be a representation of any kind of model. By re-substituting equation B.2 into 4.69 predicted state ${\boldsymbol{x}^*}_{V_{k+n}}$ is determined, n steps in the future, when k measurements are given.

$$\begin{aligned} p({\boldsymbol{x}^*}_{V_{k+n}} \mid \mathbf{Y}_{V_k}) = p({\boldsymbol{x}^*}_{V_{k+n}} \mid \hat{\boldsymbol{x}}_{V_k}) \cdot \eta \cdot p(\boldsymbol{y}_{V_k} \mid \hat{\boldsymbol{x}}_{V_k}) \\ \cdot \int p(\hat{\boldsymbol{x}}_{V_k} \mid \hat{\boldsymbol{x}}_{V_{k-1}}) \cdot p(\hat{\boldsymbol{x}}_{V_{k-1}} \mid \mathbf{Y}_{V_{k-1}}) d\,\hat{\boldsymbol{x}}_{V_{k-1}} \end{aligned} \tag{B.3}$$

C Maneuver sequences

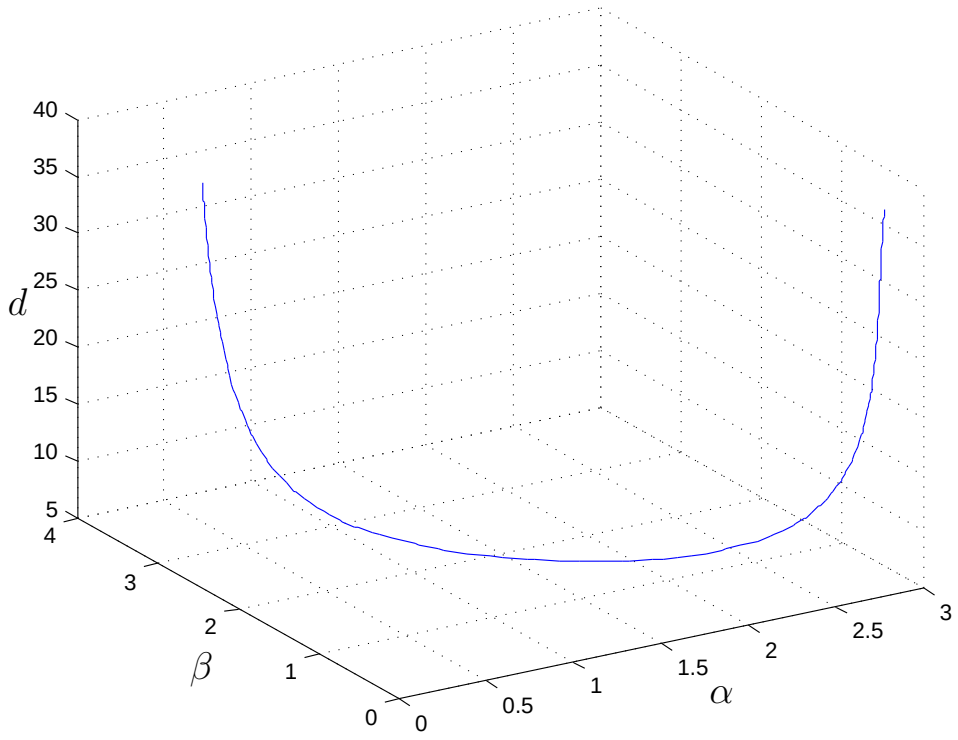


Figure C.1:
Representation of a swerve maneuver from an ego-centric perspective by means of a sequence of relations (α, β, d). The start of the maneuver is located at the top left of the figure. Two vehicles were placed facing each other $(\alpha \approx 0, \beta \approx \pi)$. During the approach both vehicles executed an evasive motion to the right with respect to their driving direction, consequently resulting in both vehicles passing each other $(\alpha \approx \pi/2, \beta \approx \pi/2, d \approx 5m)$. After the vehicles passed each other, a slight left turn motion was executed, so that the rear of the vehicles faced each other $(\alpha \approx \pi/2, \beta \approx 0)$.

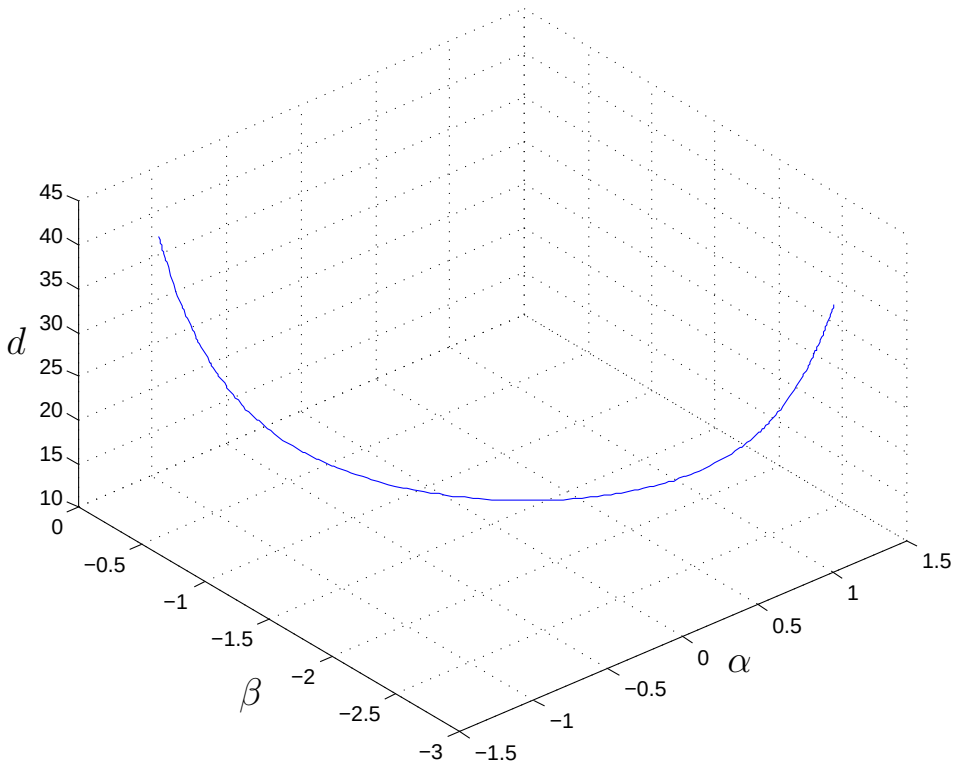


Figure C.2:
Representation of a yield maneuver on a 4-way crossroad from an ego-centric perspective by means of a sequence of relations (α, β, d). The start of the maneuver is located at the top right of the figure. The ego-vehicle waited in order for the second car to pass the crossroad. The second car approached from the left-hand side ($\alpha \approx \pi/2, \beta \approx -\pi$), passed through the center of the crossroad ($\alpha \approx 0, \beta \approx -\pi/2$) and moved away to the right ($\alpha \approx -\pi/2, \beta \approx 0$).

Bibliography

Andrews, J. R. (1983). *Impedance control as a framework for implementing obstacle avoidance in a manipulator*. PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering.

Arkin, R. (1987). Motor schema-based mobile robot navigation. In *IEEE International Conference on Robotics and Automation*, pages 264–271.

Arkin, R. C., Riseman, E. M., and Hanson, A. R. (1987). Aura: An architecture for vision-based robot navigation. In *Image Understanding Workshop: proceedings of a workshop held at Los Angeles, California, February 23-25, 1987*, volume 2, page 417. Morgan Kaufmann.

Bellman, R. E. (1957). Dynamic programming. *Princeton University Press, Princeton, NJ*.

Bishop, G. and Welch, G. (2001). An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8:27599–3175.

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. The MIT Press.

Brooks, R. (1986). A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14–23.

Buehler, M., Iagnemma, K., and Singh, S. (2007). *The 2005 DARPA Grand Challenge: The Great Robot Race*. Springer Publishing Company, Incorporated, 1st edition.

Buehler, M., Iagnemma, K., and Singh, S. (2009). *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Springer Publishing Company, Incorporated, 1st edition.

Carlin, B. P. and Louis, T. A. (2009). *Bayesian methods for data analysis; 3rd ed.* CRC Press.

Coulter, R. C. (1992). Implementation of the Pure Pursuit Path Tracking Algorithm.

Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings.*, volume 2, pages 1322 –1328 vol.2.

Deml, B., Neumann, H., Mueller, A., and Wuensche, H.-J. (2008). Commanding a Simulated Robot Car by a Cognitive Driver Model. In *Proceedings of Intl. Workshop on Distributed Intelligent Systems and Technologies*, St. Petersburg, Russia.

Dickmanns, E. D. (1988). 4d-dynamic scene analysis with integral spatio-temporal models. In *Proceedings of the 4th international symposium on Robotics Research*, pages 311–318, Cambridge, MA, USA. MIT Press.

Dickmanns, E. D. (2007). *Dynamic Vision for Perception and Control of Motion*. Springer Verlag, London.

Dickmanns, E. D. and Zapp, A. (1988). Autonomous high speed road vehicle guidance by computer vision. In *10th IFAC World Congress*, volume 1, pages p. 221–226.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Eden, C. (1988). Cognitive mapping. *European Journal of Operational Research*, 36(1):1–13.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57.

Fassbender, D., Mueller, A., and Wuensche, H.-J. (2014). Trajectory Planning for Car-Like Robots in Unknown, Unstructured Environments. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, Chicago, IL, USA.

Firl, J., Stubing, H., Huss, S., and Stiller, C. (2012). Predictive maneuver evaluation for enhancement of car-to-x mobility data. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 558–564.

Fliess, M., Lévine, J., Martin, P., and Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61(6):1327–1361.

Foka, A. and Trahanias, P. (2010). Probabilistic autonomous robot navigation in dynamic environments with human motion prediction. *International Journal of Social Robotics*, 2:79–94.

Fries, C., Burger, P., Kallwies, J., Naujoks, B., Luettel, T., and Wuensche, H.-J. (2017). How MuCAR Won the Convoy Scenario at ELROB 2016. Yokohama, Japan.

Geiger, A., Lauer, M., Moosmann, F., Ranft, B., Rapp, H., Stiller, C., and Ziegler, J. (2012). Team annieway's entry to the 2011 grand cooperative driving challenge. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):1008–1017.

Gibson, J. J. (1977). *The Theory of Affordances*. Lawrence Erlbaum.

Goebl, M. (2009). *Eine realzeitfähige Architektur zur Integration kognitiver Funktionen*. Dissertation, Technische Universität München, München.

Gothard, K., Skaggs, W., Moore, K., and McNaughton, B. (1996). Binding of hippocampal ca1 neural activity to multiple reference frames in a landmark-based navigation task. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 16(2):823–835.

Grill-Spector, K., Kourtzi, Z., and Kanwisher, N. (2001). The lateral occipital complex and its role in object recognition. *Vision Research*, 41(10):1409–1422.

Guennebaud, G., Jacob, B., Capricelli, T., et al. (2010). Eigen v3. http://eigen.tuxfamily.org.

Hart, P., Nilsson, N., and Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

Hayes, P. J. (1971). The frame problem and related problems in artificial intelligence. Technical report, Stanford University, Stanford, CA, USA.

Hermes, C., Wohler, C., Schenk, K., and Kummert, F. (2009). Long-term vehicle motion prediction. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 652 –657.

Himmelsbach, M., Schneider, S., and Wuensche, H.-J. (2011). A Comparison of Error Metrics for Extrinsic Calibration and Fusion of Camera and Multi-Layer LIDAR. In *Proceedings of IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA)*, Innsbruck, Austria.

Himmelsbach, M., von Hundelshausen, F., Luettel, T., Manz, M., Mueller, A., Schneider, S., and Wuensche, H.-J. (2009). Team MuCAR-3 at C-ELROB 2009. In *Proceedings of 1st Workshop on Field Robotics, Civilian European Land Robot Trial 2009*, ISBN 978-951-42-9176-0, Oulu, Finland. University of Oulu.

Himmelsbach, M. and Wuensche, H.-J. (2012). Tracking and Classification of Arbitrary Objects with Bottom-Up/Top-Down Detection. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 577–582, Alcalá de Henares, Spain.

Julier, S. J. and Uhlmann, J. K. (2001). A counter example to the theory of simultaneous localization and map building. In *In IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4238–4243.

Kammel, S., Ziegler, J., Pitzer, B., Werling, M., Gindele, T., Jagzent, D., Schröder, J., Thuy, M., Goebl, M., Hundelshausen, F. v., Pink, O., Frese, C., and Stiller, C. (2008). Team annieway's autonomous system for the 2007 darpa urban challenge. *Journal of Field Robotics*, 25(9):615–639.

Kanayama, Y. J. and Hartman, B. I. (1997). Smooth local-path planning for autonomous vehicles1. *The International Journal of Robotics Research*, 16(3):263–284.

Kariya, T. and Kurata, H. (2004). *Generalized Least Squares*. Wiley.

Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566 –580.

Koren, Y. and Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1398–1404.

Kwon, W. H., Bruckstein, A. M., and Kailath, T. (1983). Stabilizing state-feedback design via the moving horizon method†. *International Journal of Control*, 37(3):631–643.

Large, F., Vasquez, D., Fraichard, T., and Laugier, C. (2004). Avoiding cars and pedestrians using velocity obstacles and motion prediction. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 375–379.

LaValle, S. M. and Kuffner, J. J. (2001). Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400.

Lefevre, S., Laugier, C., and Ibanez-Guzman, J. (2012). Risk assessment at road intersections: Comparing intention and expectation. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 165–171.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168.

Levinson, J. and Thrun, S. (2010). Robust vehicle localization in urban environments using probabilistic maps. In *Robotics and Automation*, pages 4372 –4378.

Levitt, T. S. and Lawton, D. T. (1990). Qualitative navigation for mobile robots. *Artif. Intell.*, 44(3):305–360.

Limketkai, B., Liao, L., and Fox, D. (2005). Relational object maps for mobile robots. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, pages 1471 – 1476, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Luettel, T., Himmelsbach, M., von Hundelshausen, F., Manz, M., Mueller, A., and Wuensche, H.-J. (2009). Autonomous Offroad Navigation Under Poor GPS Conditions. In *Proceedings of 3rd Workshop On Planning, Perception and Navigation for Intelligent Vehicles (PPNIV), IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA.

Mackworth, A. K. (1992). On seeing robots. In *Computer Vision: Systems, Theory, and Applications*, pages 1–13. World Scientific Press.

Manz, M. (2013). *Modellbasierte visuelle Wahrnehmung zur autonomen Fahrzeugführung*. PhD thesis, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Neubiberg.

McCulloch, W. S. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 7:115–133.

Meyer-Delius, D., Plagemann, C., and Burgard, W. (2009). Probabilistic situation recognition for vehicular traffic scenarios. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 459–464.

Miller, G. A. (1956). The magical number seven plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81–97.

Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., Johnston, D., Klumpp, S., Langer, D., Levandowski, A., Levinson, J., Marcil, J., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., Pflueger, M., Stanek, G., Stavens, D., Vogt, A., and Thrun, S. (2008). Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). Fastslam: A factored solution to the simultaneous localization and mapping problem. In *National Conference on Artificial Intelligence*.

Moravec, H. P. (1980). *Obstacle avoidance and navigation in the real world by a seeing robot rover*. PhD thesis, Stanford, CA, USA.

Morris, A., Silver, D., Ferguson, D., and Thayer, S. (2005). Towards topological exploration of abandoned mines. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.*, pages 2117–2123.

Mueller, A., Himmelsbach, M., Luettel, T., von Hundelshausen, F., and Wuensche, H.-J. (2011). GIS-Based Topological Robot Localization through LIDAR Crossroad Detection. In *Proceedings of International IEEE Conference on Intelligent Transportation Systems*, pages 2001–2008, Washington, DC, USA.

Mueller, A., Manz, M., Himmelsbach, M., and Wuensche, H.-J. (2009). A Model-Based Object Following System. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Xi'an, Shaanxi, China.

Mueller, A. and Wuensche, H.-J. (2012). Object-Related-Navigation for Mobile Robots. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 603–610, Alcalá de Henares, Spain.

Mueller, A. and Wuensche, H.-J. (2013). Navigationsverfahren fuer mobile Roboter – Ein Blick in Vergangenheit, Gegenwart und Zukunft / Navigation Approaches for Mobile Robots – Past, Present and Possible Future Advances. *at - Automatisierungstechnik*, 61(4):290–300. Schwerpunktheft Mobile Robotik.

Neisser, U. (1976). *Cognition and reality : principles and implications of cognitive psychology*. W.H. Freeman San Francisco.

Nilsson, N. J. (1984). Shakey the robot. Technical report, Defense Technical Information Center OAI-PMH Repository.

Nissoux, C., Simeon, T., and Laumond, J.-P. (1999). Visibility based probabilistic roadmaps. In *Intelligent Robots and Systems, 1999. IROS '99. Proceedings.*, volume 3, pages 1316 –1321 vol.3.

Nister, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages 652 – 659.

O'Keefe, J. and Nadel, L. (1978). *The hippocampus as a cognitive map*. Clarendon Press, Oxford.

Pêtrès, C., Romero-Ramirez, M.-A., and Plumet, F. (2012). A potential field approach for reactive navigation of autonomous sailboats. *Robotics and Autonomous Systems*, (0).

Pettersson, P. O. and Doherty, P. (2004). Probabilistic roadmap based path planning for an autonomous unmanned aerial vehicle. In *ICAPS-04 Workshop on Connecting Planning Theory with Practice*, pages 49 – 55.

Raper, H. (1840). *The practice of navigation and nautical astronomy*. R.B. Bate, 21 Poultry.

Rohrmuller, F., Althoff, M., Wollherr, D., and Buss, M. (2008). Probabilistic mapping of dynamic obstacles using markov chains for replanning in dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2504–2510.

Russ, J. C. (2006). *The image processing handbook*. CRC press.

Saffiotti, A. (1997). The uses of fuzzy logic in autonomous robot navigation. *Soft Computing*, 1(4):180–197.

Saust, F., Wille, J., Lichte, B., and Maurer, M. (2011). Autonomous vehicle guidance on braunschweig's inner ring road within the stadtpilot project. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 169–174.

Seraji, H. (1999). Traversability index: a new concept for planetary rovers. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 3, pages 2006–2013 vol.3.

Shimoda, S., Kuroda, Y., and Iagnemma, K. (2005). Potential field navigation of high speed unmanned ground vehicles on uneven terrain. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2828–2833.

Slabaugh, G. G. (1999). Computing euler angles from a rotation matrix.

Smith, R., Self, M., and Cheeseman, P. (1990). Autonomous robot vehicles. chapter Estimating uncertain spatial relationships in robotics, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA.

Spangenberg, R., Goehring, D., and Rojas, R. (2016). Pole-based localization for autonomous vehicles in urban scenarios. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2161–2166.

Stentz, A. J. (1990). *The Navlab system for mobile robot navigation*. PhD thesis, Pittsburgh, PA, USA. UMI Order No. GAX90-33071.

Sugihara, K. (1988). Some location problems for robot navigation using a single camera. *Computer Vision, Graphics, and Image Processing*, 42(1):112 – 129.

Thomas, J., Stiens, K., Rauch, S., and Rojas, R. (2015). Grid-based online road model estimation for advanced driver assistance systems. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 71–76.

Thrun, S. (2000). Probabilistic algorithms in robotics. *AI Magazine*, 21:93–109.

Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Creemers, A., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (2000). Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19(1):972–999.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208.

Tomatis, N., Nourbakhsh, I., and Siegwart, R. (2003). Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 44(1):3 – 14.

Trepagnier, P. G., Nagel, J., Kinney, P. M., Koutsougeras, C., and Dooner, M. (2006). Kat-5: Robust systems for autonomous vehicle navigation in challenging and unknown terrain. *Journal of Field Robotics*, 23(8):509–526.

Unterholzner, A., Himmelsbach, M., and Wuensche, H.-J. (2012). Active Perception for Autonomous Vehicles. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1620–1627, St. Paul, MN, USA.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T. M., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.-W., Singh, S., Snider, J., Stentz, A., Whittaker, W. R., Wolkowicki, Z., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M., and Ferguson, D. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466.

Urmson, C., Ragusa, C., Ray, D., Anhalt, J., Bartz, D., Galatali, T., Gutierrez, A., Johnston, J., Harbaugh, S., "Yu" Kato, H., Messner, W., Miller, N., Peterson, K., Smith, B., Snider, J., Spiker, S., Ziglar, J., "Red" Whittaker, W., Clark, M., Koon, P., Mosher, A., and Struble, J. (2006). A robust approach to high-speed navigation for unrehearsed desert terrain. *Journal of Field Robotics*, 23(8):467–508.

van Nunen, E., Kwakkernaat, R., Ploeg, J., and Netten, B. (2012). Cooperative competition for future mobility. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):1018–1025.

Vandapel, N., Kuffner, J., and Amidi, O. (2005). Planning 3-d path networks in unstructured environments. In *Robotics and Automation, 2005 (ICRA 2005)*, pages 4624–4629.

Vasudevan, S., Gächter, S., Nguyen, V., and Siegwart, R. (2007). Cognitive maps for mobile robots—an object based approach. *Robotics and Autonomous Systems*, 55(5):359 – 371.

von Hundelshausen, F., Himmelsbach, M., Hecker, F., Mueller, A., and Wuensche, H.-J. (2008). Driving with tentacles: Integral structures for sensing and motion. *J. Field Robot.*, 25(9):640–673.

Walter, W. G. (1950). An Imitation of Life. *Scientific American*, 182:42–45.

Wang, C.-C., Thorpe, C., and Thrun, S. (2003). Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 1, pages 842 – 849.

Wang, R. and Spelke, E. (2000). Updating egocentric representations in human navigation. *Cognition*, 77(3):215–250.

Wuensche, H.-J. (2013). Merkmalsbasierte objektrelationale Navigation. DFG proposal.

Yoerger, D. R., Bradley, A. M., Walden, B. B., Singh, H., and Bachmayer, R. (1998). Surveying a subsea lava flow using the autonomous benthic explorer (abe). *International Journal of Systems Science*, 29(10):1031–1044.

Yoerger, D. R., Jakuba, M., Bradley, A. M., and Bingham, B. (2007). Techniques for deep sea near bottom survey using an autonomous underwater vehicle. *Int. J. Rob. Res.*, 26(1):41–54.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338 – 353.

Zapp, A. (1988). *Automatische Strassenfahrzeugführung durch Rechnersehen*. PhD thesis, University of the Bundeswehr Munich.