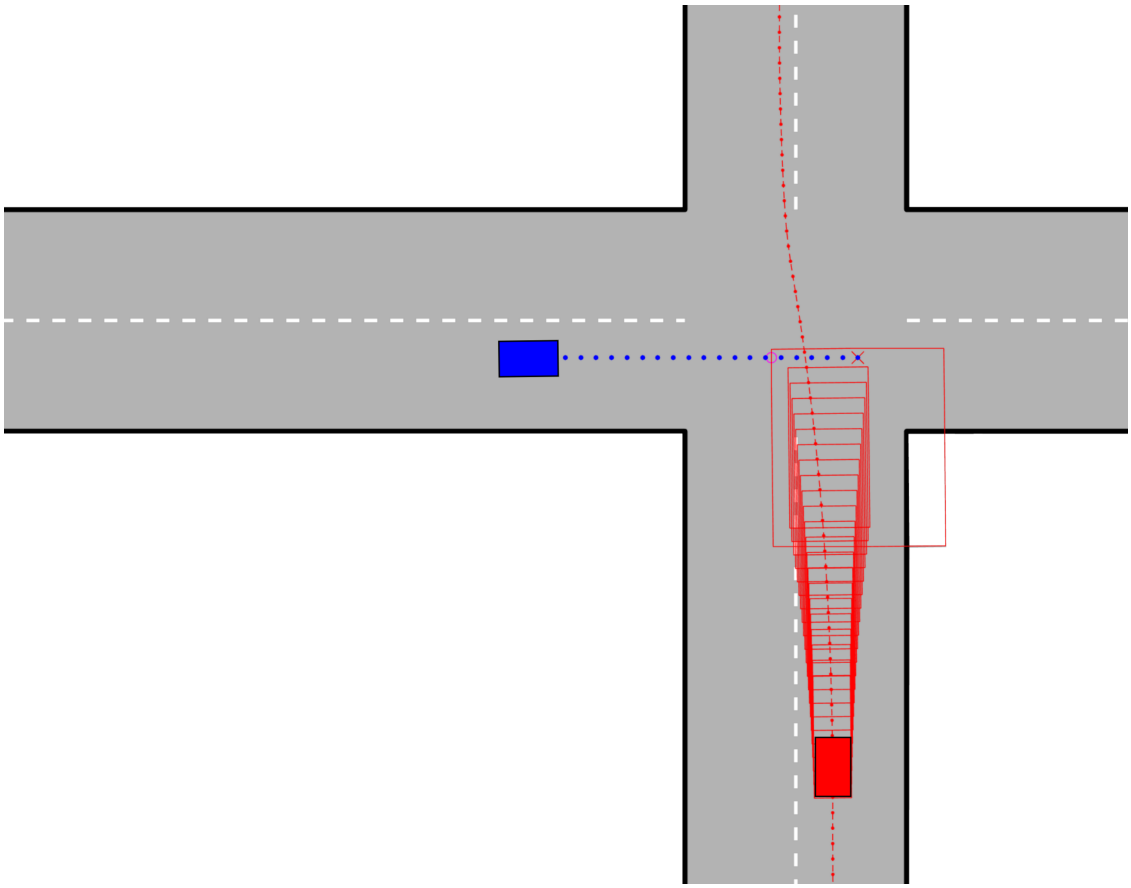
CHALMERS
UNIVERSITY OF TECHNOLOGY



Centralized collision avoidance system for automated vehicles

A safety system utilizing classic reachability analysis and model predictive control

Master's thesis in Master Programme Systems, Control and Meachatronics

Andreas Hahlin
Anton Olsson

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis 2017:EX027

Centralized collision avoidance system for automated vehicles

A safety system utilizing classic reachability analysis and model predictive control

Andreas Hahlin
Anton Olsson

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology
Gothenburg, Sweden 2017

Centralized collision avoidance system for automated vehicles

A safety system utilizing classic reachability analysis and model predictive control

Master's thesis in Master Programme Systems, Control and Meachatronics
Andreas Hahlin
Anton Olsson

© Andreas Hahlin, Anton Olsson, 2017.

Supervisor: Josef Nilsson, Zenuity
Examiner: Jonas Sjöberg, Signals and Systems

Master's Thesis 2017:EX027
Department of Electrical Engineering
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Birdview of an observed collision risk through reachability analysis between a vehicle that can not be controlled from below and a vehicle that can be controlled on the left whose trajectory being altered to be outside the observed dangerous area.

Typeset in LaTeX
Printed by [Name of printing company]
Gothenburg, Sweden 2017

Centralized collision avoidance system for automated vehicles
A safety system utilizing classic reachability analysis and model predictive control
Andreas Hahlin
Anton Olsson
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This master's thesis examines the possibility of utilizing classic Reachability Analysis (RA) together with Model Predictive Control (MPC) in order to create a Centralized Collision Avoidance System (CCAS) for automated vehicles. With minimal intrusion to a preexisting system, the CCAS engage when a collision risk is detected and averts it. Implementation and evaluation were performed in Volvo cars' existing Simulation Platform for Active Safety (SPAS).
The CCAS is designed and evaluated for a traffic intersection scenario consisting of a vehicle with unknown future behaviour and two vehicles with known future behaviour which also could be controlled by the CCAS. The created CCAS utilizes classic RA to predict the future reachable sets of positions of the vehicle with unknown behaviour up to a prediction horizon H_x. If a collision risk is detected, the MPC derives new trajectories for the controllable vehicles in order for them to avoid any collision risks.
Though the CCAS was successfully implemented into SPAS, it was concluded that the vehicle with unknown behaviour must have a restricted uncertainty while using classic RA as its reachable sets grows too rapidly for the CCAS to be minimally intrusive to the preexisting system. It was further derived that the initial velocities at which all vehicles enter the intersection at proved to have a large effect on the CCAS's feasibility to avoid the collision. Though solvable, it further increases the intrusiveness to the preexisting system. It is concluded that a successful method to derive the minimum prediction horizon H_x was implemented based on resulting simulations.

Keywords: Collision avoidance, active safety, safety system, centralized control, reachability analysis, model predictive control, MPC.

Acknowledgements

We would like to acknowledge several individuals and teams that have helped us during the progress of this master's thesis. We like to thank our examiner Jonas Sjöberg for valuable insight and constructive criticism since the start of this project. Josef Nilsson for introducing us to this opportunity and his support and insight as a whole. Gazaleh Panahandeh for a similar role as well as her feedback of the literal aspects of the thesis. Christoffer Wedding for his technical expertise concerning SPAS and its inner workings. Mats Jonasson for helping us with modeling the vehicles and finally Volvo cars and its employees, again for offering us the opportunity for this exciting thesis as well as their warm treatment throughout.

Andreas Hahlin, Anton Olsson
Gothenburg, June 2017

Preliminaries

This preliminary chapter starts by presenting used abbreviations and are then followed by mathematical notations categorized by association.

Abbreviations

CCAS	Centralized Collision Avoidance System
RA	Reachability Analysis
MPC	Model Predictive Control
FMPC	Fast Model Predictive Control
LQR	Linear Quadratic Regulator
ITS	Intelligent Traffic System
iTRANSIT	intelligent Traffic mANagement System based on ITS
SPAS	Simulation Platform for Active Safety
CV	Constant Velocity
BM	Bicycle Model
G-representation	Generator Representation
H-representation	Half Space Representation
I-representation	Interval Representation
V-representation	Vertex Representation
LP	Linear Programming
ODE	Ordinary Differential Equation
CoG	Center of Gravity

General notations

$\mathbb{N}$	Natural numbers
$\mathbb{R}$	Real valued numbers
$\mathbb{I}$	Real valued intervals
$\dot{\square}$	Time derivative
$\bar{\square}$	Maximum value
$\underline{\square}$	Minimum value
$\square^{BM}$	Matrix associated with the BM
$\square^{CV}$	Matrix associated with the CV model

Model notations

n	$\in \mathbb{N}$	Number of state variables
m	$\in \mathbb{N}$	Number of control inputs
n_c	$\in \mathbb{N}$	Number of controlled outputs
$\mathbf{A}$	$\in \mathbb{R}^{n \times n}$	System matrix
$\mathbf{B}$	$\in \mathbb{R}^{n \times m}$	Input matrix
$\mathbf{C}$	$\in \mathbb{R}^{n_c \times n}$	Output matrix
$\mathbf{X}$	$\in \mathbb{R}^{n}$	State vector
$\mathbf{U}$	$\in \mathbb{R}^{m}$	Input vector
f_{uc}	$\in \mathbb{R}^{m}$	Model of uncontrollable vehicle
f_{NL}	$\in \mathbb{R}^{m}$	Nonlinear model
M	$\in \mathbb{R}$	Mass
I_z	$\in \mathbb{R}$	Moment of interia around z-axis
C_f	$\in \mathbb{R}$	Corning stiffness, front
C_r	$\in \mathbb{R}$	Corning stiffness, rear
C	$\in \mathbb{R}$	Length between CoG and front
L	$\in \mathbb{R}$	Length of vehicle
W	$\in \mathbb{R}$	Width of vehicle
l_f	$\in \mathbb{R}$	Length between CoG and front axis
l_r	$\in \mathbb{R}$	Length between CoG and rear axis
δ	$\in \mathbb{R}$	Steering angle
F_x	$\in \mathbb{R}$	Applied force
a	$\in \mathbb{R}$	Applied acceleration
X	$\in \mathbb{R}$	Global position
X_r	$\in \mathbb{R}$	Global reference position
Y	$\in \mathbb{R}$	Global position
Y_r	$\in \mathbb{R}$	Global reference position
V_X	$\in \mathbb{R}$	Global velocity
V_Y	$\in \mathbb{R}$	Global velocity
x	$\in \mathbb{R}$	Local longitudinal position
y	$\in \mathbb{R}$	Local lateral position
V_x	$\in \mathbb{R}$	Local longitudinal velocity
V_x^{const}	$\in \mathbb{R}$	Constant longitudinal velocity
V_y	$\in \mathbb{R}$	Local lateral Velocity
Ψ	$\in \mathbb{R}$	Heading
ω	$\in \mathbb{R}$	Yaw rate

Reachability analysis notations

$\mathcal{Z}$	$\subset \mathbb{R}^{n}$	Zonotope
$\mathbf{c}$	$\in \mathbb{R}^{n}$	Center of zonotope
$\mathbf{g}$	$\in \mathbb{R}^{n}$	Generator
$\mathcal{Z}^{red}$	$\subset \mathbb{R}^{n}$	Reduced Zonotope
$\hat{\mathcal{Z}}$	$\subset \mathbb{R}^{n}$	Unreduced part of zonotope

$\check{\mathcal{Z}}$	$\subset \mathbb{R}^n$	Reduced part of zonotope
ϕ	$\subset \mathbb{R}^n$	Polytope
Π	$\in \mathbb{R}^{n\times n}$	Scaling matrix
v	$\in \mathbb{R}$	Inverse volume
$\mathcal{X}_0$	$\subset \mathbb{R}^n$	Initial set
$\mathcal{U}$	$\subset \mathbb{R}^m$	Input set
$\mathcal{H}$	$\subset \mathbb{R}^n$	Homogeneous solution
V	$\subset \mathbb{R}^n$	Increment of particular solution
$\mathcal{P}$	$\subset \mathbb{R}^n$	Particular solution
$\mathcal{R}$	$\subset \mathbb{R}^n$	Local reachable set
r	$\subset \mathbb{R}^2$	Local reachable set of velocities
$\mathcal{R}^p$	$\subset \mathbb{R}^2$	Global reachable set of positions
$\mathcal{R}^{uc}$	$\subset \mathbb{R}^2$	Global reachable set
$\mathcal{F}$	$\in \mathbb{I}^{n\times n}$	Correction matrix
$\mathcal{E}$	$\in \mathbb{I}^{n\times n}$	Error matrix
p_0	$\subset \mathbb{I}$	Position set
Ψ	$\subset \mathbb{I}$	Heading angle set
M^I	$\in \mathbb{I}^{2\times 2}$	State transformation interval matrix
$\mathbf{M}$	$\in \mathbb{R}^{2\times 2}$	Rotation matrix
$\bar{\mathbf{M}}$	$\in \mathbb{R}^{2\times 2}$	Maximum value of rotation matrix
$\underline{\mathbf{M}}$	$\in \mathbb{R}^{2\times 2}$	Minimum value of rotation matrix
d	$\in \mathbb{I}^{2\times 1}$	Dimension of vehicle
$\mathbf{p}$	$\in \mathbb{R}^2$	Global position
$\boldsymbol{\Delta}_X$	$\in \mathbb{R}^n$	State deviation
Δ_X	$\in \mathbb{R}$	X deviation
Δ_Y	$\in \mathbb{R}$	Y deviation
Δ_{V_x}	$\in \mathbb{R}$	V_x deviation
Δ_{V_y}	$\in \mathbb{R}$	V_y deviation
Δ_Ψ	$\in \mathbb{R}$	Ψ deviation
Δ_ω	$\in \mathbb{R}$	ω deviation
$\boldsymbol{\Delta}_p$	$\in \mathbb{R}^{2\times 1}$	p deviation
$\bar{V}_x$	$\in \mathbb{R}$	Maximum initial V_x in RA
$\underline{V}_x$	$\in \mathbb{R}$	Minimum initial V_x in RA
$\bar{V}_y$	$\in \mathbb{R}$	Maximum initial V_y in RA
$\underline{V}_y$	$\in \mathbb{R}$	Minimum initial V_y in RA
$\bar{\Psi}$	$\in \mathbb{R}$	Maximum initial Ψ in RA
$\underline{\Psi}$	$\in \mathbb{R}$	Minimum initial Ψ in RA
$\bar{\omega}$	$\in \mathbb{R}$	Maximum initial ω in RA
$\underline{\omega}$	$\in \mathbb{R}$	Minimum initial ω in RA
$\bar{\mathbf{p}}$	$\in \mathbb{R}^{2\times 1}$	Maximum initial $\mathbf{p}$ in RA
$\underline{\mathbf{p}}$	$\in \mathbb{R}^{2\times 1}$	Minimum initial $\mathbf{p}$ in RA
$\bar{\Phi}$	$\in \mathbb{R}$	Heading angle amplitude
$\mathbf{C}_H$	$\in \mathbb{R}^{1\times 2}$	Half space weights
$\mathbf{C}_X$	$\in \mathbb{R}$	Individual weights
$\mathbf{C}_Y$	$\in \mathbb{R}$	Individual weights
$\mathbf{d}_H$	$\in \mathbb{R}$	Half space constants

X_H	$\in \mathbb{R}$	Intersection point between half space and reference trajectory
Y_H	$\in \mathbb{R}$	Intersection point between half space and reference trajectory
t	$\in \mathbb{R}$	Time step
η	$\in \mathbb{R}$	Number of Taylor terms
ϵ	$\in \mathbb{R}$	Metric
n_g	$\in \mathbb{R}$	Number of generators

Model predictive control notations

Notations associated with the complete MPC setup:

n_a	$\in \mathbb{N}$	Number of actors
i	$\in \mathbb{N}$	Actor number
H_x	$\in \mathbb{N}$	Prediction horizon
H_u	$\in \mathbb{N}$	Control horizon
n_z	$\in \mathbb{N}$	Length of optimization vector
D	$\in \mathbb{R}$	Delay
J	$\in \mathbb{R}$	Cost
$\mathbf{z}$	$\in \mathbb{R}^{n_z}$	Optimization column vector
$\mathbf{H}$	$\in \mathbb{R}^{n_z \times n_z}$	Weighting matrix
$\mathbf{A}_{eq}$	$\in \mathbb{R}^{nH_x n_a \times n_z}$	Equality constraint matrix
$\mathbf{B}_{eq}$	$\in \mathbb{R}^{nH_x n_a}$	Equality constraint column vector
$\mathbf{A}_{in}$	$\in \mathbb{R}^{(2nH_x+2mH_u)n_a \times (2nH_x+2mH_u)n_a}$	Inequality constraint matrix
$\mathbf{B}_{in}$	$\in \mathbb{R}^{(2nH_x+2mH_u)}$	Inequality constraint column vector

Notations associated with each vehicle i:

$\mathbf{A}^i_{eq}$	$\in \mathbb{R}^{nH_x \times nH_x}$	Equality constraint matrix
$\mathbf{B}^i_{eq}$	$\in \mathbb{R}^{nH_x}$	Equality constraint column vector
$\bar{\mathbf{A}}^i_{inX}$	$\in \mathbb{R}^{nH_x \times (nH_x+mH_u)}$	Upper bound inequality constraint
$\underline{\mathbf{A}}^i_{inX}$	$\in \mathbb{R}^{nH_x \times (nH_x+mH_u)}$	Lower bound inequality constraint
$\bar{\mathbf{B}}^i_{inX}$	$\in \mathbb{R}^{nH_x}$	Inequality constraint matrix for actor i
$\underline{\mathbf{B}}^i_{inX}$	$\in \mathbb{R}^{nH_x}$	Inequality constraint matrix for actor i
$\bar{\mathbf{A}}^i_{inU}$	$\in \mathbb{R}^{mH_u \times (nH_x+mH_u)}$	Inequality constraint matrix
$\underline{\mathbf{A}}^i_{inU}$	$\in \mathbb{R}^{mH_u \times (nH_x+mH_u)}$	Inequality constraint matrix
$\bar{\mathbf{B}}^i_{inU}$	$\in \mathbb{R}^{mH_u}$	Inequality constraint matrix
$\underline{\mathbf{B}}^i_{inU}$	$\in \mathbb{R}^{mH_u}$	Inequality constraint matrix
$\mathbf{Q}^i$	$\in \mathbb{R}^{nH_x \times nH_x}$	Weighting matrix over time of the states
$\mathbf{q}^i$	$\in \mathbb{R}^{n \times n}$	Weighting matrix of individual time instance
q^i	$\in \mathbb{R}^{n \times n}$	Weight of individual states
$\mathbf{R}^i$	$\in \mathbb{R}^{mH_u \times mH_u}$	Weighting matrix over time of the input
$\mathbf{r}^i$	$\in \mathbb{R}^{m \times m}$	Weighting matrix of individual time instance
r^i	$\in \mathbb{R}^{m \times m}$	Weight of individual input

Prediction horizon notations

H_x	$\in \mathbb{N}$	Prediction horizon
h	$\in \mathbb{N}$	Additional time steps
T_{H_x}	$\in \mathbb{R}$	Prediction time
T_h	$\in \mathbb{R}$	Additional prediction time
d_b	$\in \mathbb{R}$	Possible breaking distance in H_x steps
d_a	$\in \mathbb{R}$	Possible acceleration distance in H_x steps
d_h	$\in \mathbb{R}$	Distance between trajectory point at H_x and worst case zonotope at $H_x + h$
Δd_b	$\in \mathbb{R}$	Additional distance due to multiple potential collisions

Contents

1

Introduction

This chapter introduces and motivates the thesis. A general explanation of the project is presented in Section 1.1. The background of the thesis is given in Section 1.2 and the problem is described in Section 1.3. The confines of the project is established in Section 1.5 in order to make the project manageable. An overview of the implemented Centralized Collision Avoidance System (CCAS) is given in Section 1.4 providing the reader with an overview of the project's components and the information flow between those. In Section 1.6 similar bodies of work are presented which have supported decisions taken during the thesis. This is followed by Section 1.7 that presents scientific contributions. The chapter ends with Section 1.8 presenting the structure of the thesis.

1.1 Centralized Collision Avoidance System concept

This thesis presents the development of a CCAS with the purpose of preventing collision risks in a traffic intersection scenario between controllable and uncontrollable vehicles, the full scenario is detailed in Section 1.3. A controllable vehicle refers to the CCAS ability to convey new trajectories to it, in position over time, while an uncontrollable vehicle has an unknown and uncontrollable trajectory.

The CCAS intervene the controllable vehicles' preset course in the case the CCAS detects a collision risk. The detection is performed by utilizing over-approximated models for the future states of all vehicles in the scenario. The predictions are kept as narrow as they can be trusted given the available information at the time, ensuring that the CCAS remains minimally intrusive to the preexisting trajectories by not being over sensitive.

The reachable sets of states for the uncontrollable vehicle are determined by utilizing classic Reachability Analysis (RA) which utilizes a representative vehicle model, taking into account the uncertainty in its future behaviour. In contrast to stochastic RA, classic RA has no added probable assumptions to narrow the reachable sets and instead includes all reachable sets based on the uncertainty in states and/or inputs

of the model.

In the case of a detected collision risk, Model Predictive Control (MPC) is used to derive new trajectories for the controllable vehicles which will avoid the future positions in the intersection were a collision otherwise could have occurred. The trajectories are solved under one shared cost function in order to achieve coordinated movements between the controllable vehicles.

The added system is referred to as the Centralized Collision avoidance system because all communication is centralized to a server in the near vicinity of the intersection on which all computations are performed. The CCAS is created, integrated and evaluated in Volvo cars' existing Simulation Platform for Active Safety (SPAS).

1.2 Background

Technological development have led to vehicles and its surrounding infrastructure to become increasingly more intelligent, expanding the potential to handle more intricate and potentially dangerous traffic situations autonomously in real time. Volvo Cars are active contributors in this development of new safety features, helping to improve the safety and well being in society. During year 2017, Volvo cars' Drive Me project will introduce self-driving cars that will run in ordinary traffic situations in Gothenburg [1]. These new developments have opened up for new concepts such as the CCAS that utilizes these vehicles in a new combined way rather than having them work individually. This alteration is an unique and previously untested form of traffic safety regulation for a real traffic setting which incentivize a thorough study of its functionality from a control theory aspect.

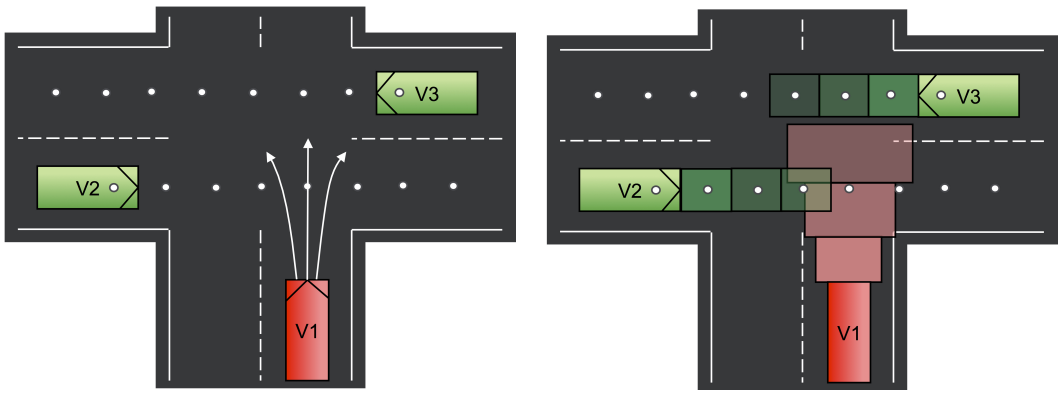
In this project, a CCAS is created and implemented as a part of the collaboration research project iTRANSIT (intelligent TRAffic maNagement System based on ITS (intelligent traffic system)), financed by Vinnova. The project involves Volvo Cars, SP, Chalmers, ÅF, and AstaZero with the main objective of aggregating different kinds of traffic data into a centralized view on a server, enabling increased traffic management and support for automated driving[2, 3].

The created CCAS utilizes RA and MPC where RA has previously been used mainly within the fields of aircrafts and MPC has its origin in the process industry, but could today be found in a lot of other applications [4, 5].

1.3 Problem description

The CCAS is specifically designed for the traffic intersection scenario illustrated in Figure 1.1a. The scenario consists of the uncontrollable vehicle V1 and the two con-

trollable vehicles V2 and V3, all providing the centralized server with their present dynamical state at fixed time intervals. RA is then used to predict the reachable sets of states of the uncontrollable vehicles as shown in Figure 1.1b. In the case of a detected collision risk, i.e. the predefined trajectories of the controllable vehicles are within the reachable sets, new safe trajectories are calculated using MPC.



(a) Investigated traffic intersection scenario.

(b) RA applied for the investigated scenario.

Figure 1.1: Traffic intersection scenario for which the CCAS is specifically designed for. The scenario consists of the uncontrollable vehicle V1 for which RA is applied and the controllable vehicles V2 and V3 having predefined trajectories which can be altered by the centralized server.

The involved vehicles operates under these specific conditions:

- V1 is uncontrollable and enters the intersection from below with unknown trajectory.
- V2 and V3 are restricted to the confines of the road.
- V2 is controllable and enters the intersection from the left with the goal of leaving the intersection to the right.
- V3 is controllable and enters the intersection from the right with the goal of leaving the intersection to the left.
- Trajectories for V2 and V3 can be updated at fixed time intervals throughout the scenario.

The conditions associated with the server are defined as:

- The state of all vehicles provided to the server are available at fixed time intervals.
- The computational performance in the centralized server is large and is not evaluated.

- Information between the server and involved vehicles is associated with a constant but adjustable transfer delay.

To keep the CCAS minimally intrusive, the smallest prediction horizon H_x is wanted, referring to how far ahead in time the comparison between the reachable sets and trajectories are made. This discrete number of steps ahead in time defines the last time step for both the RA and MPC. The reason being that the larger the prediction horizon H_x, the larger the reachable sets becomes as the uncontrollable vehicle will have time to cover a larger area. Hence, if the prediction horizon is chosen too large, the CCAS will intervene unnecessarily often.

Based on this problem formulation, the main research questions that this thesis will answer are

1. Can classic RA together with MPC be utilized in an algorithm for a CCAS?
2. How large velocities can the vehicles entering the intersection have while maintaining feasible solutions?
3. How can the prediction horizon H_x be determined in order to have minimum intrusion to the ordinary controls of the controllable vehicles?

To answer the research questions a CCAS utilizing classic RA and MPC is created and evaluated for a worst case scenario challenging the derived prediction horizon. The evaluation is performed for three different velocities of the involved vehicles.

1.4 Centralized Collision Avoidance System overview

An overview of how the the CCAS is implemented is illustrated in Figure 1.2 containing the three main components RA, collision risk detection and MPC illustrated as blocks. With this structure, the RA algorithm calculates the future sets of states up to some specified prediction horizon H_x that the uncontrollable vehicle could reach, based on its present dynamical state. These sets are then compared against the trajectories of the controllable vehicles in the collision risk detection block. If a collision risk is detected, the MPC calculates new safe trajectories based on the reachable sets as constraints, which are then sent to the controllable vehicles.

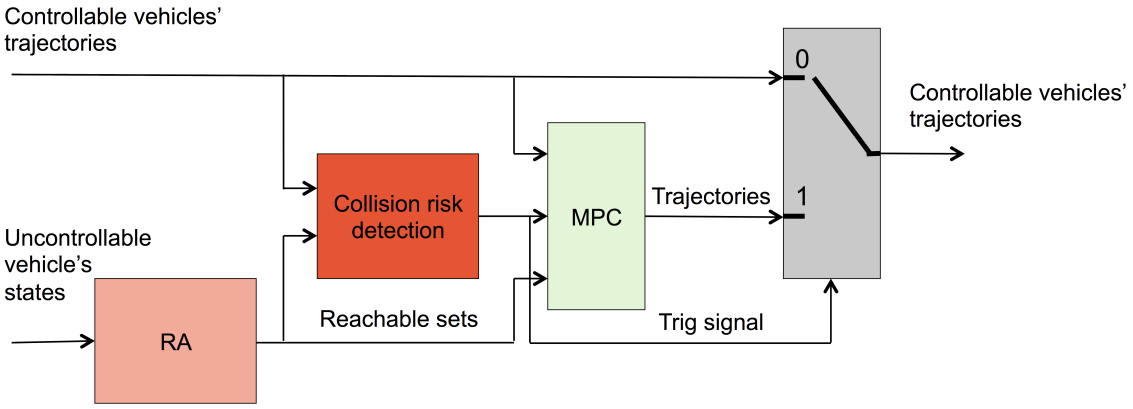


Figure 1.2: Flow chart showing the structure of the CCAS. Applying RA to the uncontrollable vehicle in order to gain its reachable sets, comparing those to the controllable vehicles trajectories, solving for new trajectories for the controllable vehicles if collision risks were detected, then passing them on instead, otherwise passing the originals.

1.5 Scope and limitations

This thesis addresses how the two main theories RA and MPC are adapted for the problem description given in Section 1.3. The development and results of the RA, MPC, collision detection algorithm as well as the determination of the minimum prediction horizon needed are presented. The following limitations are set for the thesis in order to keep the project focused:

- Only longitudinal control of the controllable vehicles through acceleration or breaking is considered as control inputs. The longitudinal axis of the controllable vehicles coincide with the global x-axis of the scenario.
- In order to generalize the problem, the uncontrollable vehicle is assumed to move perpendicularly to the controllable vehicles at the point of a detected collision risk, being equally inclined to diverge left or right towards either of the controllable vehicles.
- The measured states of the vehicle provided to the server are assumed to be ideal, i.e. without any uncertainties.
- In the scenario in Section 1.3, it is given that the uncontrollable vehicle has an unknown trajectory. Therefore, no driver model for the uncontrollable vehicle is assumed and only classical RA is investigated. Only higher velocities for all vehicles in the scenario are considered as it is more representative of a real traffic intersection.

1.6 Related work

The RA part of this thesis is mainly based on the doctoral thesis [6] that starts by presenting elementary set representations and mathematical operations commonly used in RA. Classic as well as stochastic RA are introduced and the thesis ends with presenting results from practical implementations of stochastic RA in autonomous cars proving the applicability of the theory for scenarios involving road following, vehicle following, intersection crossing and lane changing. The main difference between classic and stochastic RA is that the latter also formulates the probability of that a set is occupied. This approach becomes useful in that it shrinks the sets of occupied states by excluding sets with low probability to be true. In order to use the stochastic RA, probabilities of the sets has to be assigned. This could be done with the use of a driver model as done in [7] which argues that traffic participants tends to follow the same path for different kind of scenarios, the only thing that varies between different drivers are the speed and acceleration. The restrictions that are indirectly made by introducing a driver model might become useful for standard scenarios where vehicles follows the common paths.

One problem when applying RA is that the model of the system might be nonlinear. This implies that the superposition principle is not valid and has to be circumvented. A common approach is to linearize the system around an operating point and use that as an approximation of the system around that region. This idea could also be applied for RA which is proposed in [8] which presents an algorithm that, based on the Lagrange remainders from the linearization, decides whether the linearized system is representative enough or have to be re-linearized. By using a nonlinear model, a more representative response of the system could be achieved which might result in narrower reachable sets.

In [9], an online method for verifying whether two automated vehicles might collide by use of RA are presented. For this, a nonlinear model are used for the controllable vehicle and a simpler model is used for the uncontrollable vehicle. In this case, the uncontrollable vehicle is assumed to follow traffic regulations restricting the sets of future occupied states. For the controllable vehicle, linearizations are calculated in advance. The results of this article shows how this method could be implemented in an system executed online in the vehicles, proving the potential of implementing stochastic RA solutions.

A reoccurring choice for achieving control of scenarios that includes constraints and future predictions have been MPC [10, 11]. In [10], a simple nonlinear Bicycle Model (BM) is used to describe the dynamics of a nonholonomic wheeled mobile robot and with use of MPC trying to follow a predefined trajectory. The article proves the concept by presenting successful simulation results.

The superior functionality of MPC compared to linear control is highlighted in [12] which involves control of multiple vehicles for a platooning motion through traffic. Its functionality is further emphasized in [13] in comparison to using a Linear

Quadratic Regulator (LQR) for traffic flow optimization and in [14] compared to anti windup. The drawback with MPC is the computational burden in comparison to previously mentioned options. Precomputed strategies such as Fast Model Predictive Control (FMPC) exists [15] which performs most computations once to generate a look up table.

1.7 Scientific contributions

A method for how the combination of RA and MPC can be used for traffic safety regulation by restricting the MPC with the reachable sets achieved from the RA is presented. No single study was found that utilizes the combination of the two fields for such an application.

New insight is provided into how linear RA can be used instead of nonlinear RA. The proposition to add an over approximating transformation interval matrix into the linear RA algorithm to circumvent the issue of nonlinearity is presented.

A mathematical analysis which determines the minimum prediction horizon needed as described in Section 1.3 is presented together with an ad hoc solution used in the implementation.

The possibility of introducing a CCAS as an extension to the specific ITS environment at Volvo cars is presented.

1.8 Outline of the thesis

The introductory Chapter 1 presents the background the thesis. Then in chapter 2, the necessary theory of the RA and MPC is covered. Chapter 3 explains the approach that has been applied to the problem. Towards the end of the thesis, Chapter 4 presents the outcome of the implementation. The results are discussed in Chapter 5 followed with conclusions of the project in Chapter 5. The thesis then ends with 6, reflecting on the most relevant future work based on previous conclusions.

2

Reachability Analysis and Model Predictive Control Theory

In order to be able to create and implement the RA and MPC blocks which form the basis of the CCAS as shown in Section 1.4, underlining theory for the RA and MPC are presented in this chapter. In Section 2.1, the concept of RA is clarified and mathematical set operations used are presented. In Section 2.2, the necessary background and steps to set up the MPC is provided.

2.1 Reachability analysis

In order to predict future state of a system by use of RA computationally efficient sets are needed. In this thesis, the sets used to represent the states of the uncontrollable vehicle takes the form of zonotopes. A zonotope is a special case of polytope with the difference being that a zonotope are centrally symmetric and hence easier to represent mathematically. An introduction to zonotopes and how these are represented is given in Section 2.1.1. Mathematical operations on those sets are provided in Section 2.1.2 and can be found in [6].

2.1.1 Zonotopes

To describe the zonotopes, generator representation (G-representation) is used throughout this thesis and is a commonly used set representation for RA due to its superior computationally efficiency and scalability to higher dimensional state space representations [6, 16, 17]. However, other set representations exists such as interval representation (I-representation), half space representation (H-representation) and vertex representation (V-representation).

An example of a zonotope is presented in Figure 2.1a. This two dimensional set could intuitively be described by an interval consisting of a lower an upper limit in each dimension, which encloses the set. An other way of representing the same set would be to describe the set as a center vector along with a number of additional vectors,

called generators, that could be scaled with the factor $-1 \leq \beta_i \leq 1$. This is what is called a G-representation of a zonotope, see figure 2.1b. The G-representation of a n dimensional zonotope $\mathcal{Z}$ is defined as

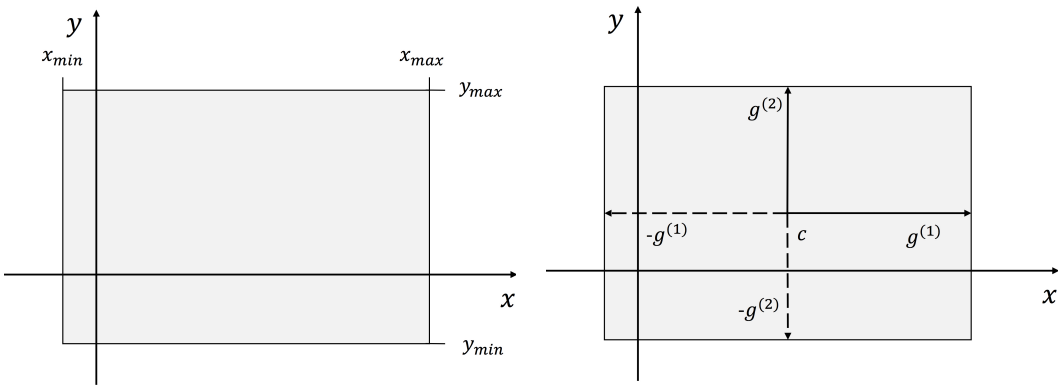
$$\mathcal{Z} = \left\{ \mathbf{X} \in \mathbb{R}^n \mid \mathbf{X} = \mathbf{c} + \sum_{i=1}^{e} \beta_i \mathbf{g}^{(i)} \quad -1 \leq \beta_i \leq 1 \right\} \tag{2.1}$$

where $\mathbf{c} \in \mathbb{R}^n$ is the centre of the set and $\mathbf{g}^{(i)} \in \mathbb{R}^n$ a generator with its associated scaling factor β_i and $e \in \mathbb{N}$. The immediate advantage of a G-representation is the possibility to describe a set without perpendicular facets, allowing for a more narrow set representation.

The H-representation of a zonotope is defined as

$$\mathcal{Z}_H = \left\{ \mathbf{X} \in \mathbb{R}^n \mid \mathbf{C}_H \mathbf{X} \leq \mathbf{d}_H \right\} \tag{2.2}$$

where $\mathbf{C}_H \in \mathbb{R}^{q \times n}$ and $\mathbf{d}_H \in \mathbb{R}^{q \times 1}$ and q are the number of half-spaces.



(a) Interval representation. **(b)** G-representation

Figure 2.1: An illustration of a zonotope in two dimensions. To the left, the set is represented with I-representation limiting the set with lower and upper bounds. To the right, the set is represented with G-representation for which the center vector and generators forms its basis.

2.1.2 Operations on zonotopes

Mathematical operations on zonotopes are used in RA. Here, the operations on the general, n dimensional, zonotopes $\mathcal{Z}_1 = (\mathbf{c}^{(1)}, \mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(e)})$ and $\mathcal{Z}_2 = (\mathbf{c}^{(2)}, \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(u)})$ are clarified. Also conversions between different set representations are presented.

2.1.2.1 Addition

The addition of the two zonotopes $\mathcal{Z}_1$ and $\mathcal{Z}_2$ are defined as

$$\mathcal{Z}_1 + \mathcal{Z}_2 = (\mathbf{c}^{(1)} + \mathbf{c}^{(2)}, \mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(e)}, \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(u)}) \tag{2.3}$$

where the resulting center of the new zonotope are defined as the addition of the two centers $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ followed by the concatenated set of generators from $\mathcal{Z}_1$ and $\mathcal{Z}_2$.

2.1.2.2 Multiplication

The linear transformation, or matrix multiplication, of a general matrix $\mathbf{M}$ and a zonotope $\mathcal{Z}_1$ is defined as

$$\mathbf{M}\,\mathcal{Z} = (\mathbf{M}\mathbf{c}^{(1)}, \mathbf{M}\mathbf{g}^{(1)}, \ldots, \mathbf{M}\mathbf{g}^{(e)}), \quad \mathbf{M} \in \mathbb{R}^{n \times n} \tag{2.4}$$

where the linear transformation operates on each component, i.e. the center and each generator, of the zonotope individually.

2.1.2.3 Convex hull

The over approximated convex hull operator acting on the two zonotopes $\mathcal{Z}_1$ and Z_2 are defined as

$$\begin{aligned} \overline{CH}(\mathcal{Z}_1, \mathcal{Z}_2) = \tfrac{1}{2}(&\mathbf{c}^{(1)} + \mathbf{c}^{(2)}, \mathbf{g}^{(1)} + \mathbf{f}^{(1)}, \ldots, \mathbf{g}^{(e)} + \mathbf{f}^{(e)}), \\ &\mathbf{c}^{(1)} - \mathbf{c}^{(2)}, \mathbf{g}^{(1)} - \mathbf{f}^{(1)}, \ldots, \mathbf{g}^{(e)} - \mathbf{f}^{(e)}) \end{aligned} \tag{2.5}$$

and results in a zonotope which encloses the two zonotopes $\mathcal{Z}_1$ and $\mathcal{Z}_2$.

2.1.2.4 G to I represenation

The box operator, defined as

$$\mathit{box}(\mathcal{Z}) = [\mathbf{c} - \Delta\mathbf{g}, \mathbf{c} + \Delta\mathbf{g}], \quad \Delta\mathbf{g} = \sum_{i=1}^{e} |\mathbf{g}^{(i)}| \tag{2.6}$$

over approximates a zonotope $\mathcal{Z}$ with an interval aligned with the unit vectors of the coordinate system.

2.1.2.5 G to H representation

A theorem that describes the conversion from G-representation to H-representation of a zonotope utilizing the n-dimensional cross product is found in [6]. The n-dimensional cross product of the $n-1$ linearly independent vectors forming a matrix $\mathbf{S} \in \mathbb{R}^{n\times(n-1)}$ results in a vector that is orthogonal to the $n-1$ vectors. By introducing the matrix $\mathbf{S}^{[i]} \in \mathbb{R}^{(n-1)\times(n-1)}$ where the superscript denotes that the i-th row is removed from $\mathbf{S}$, the n-dimensional cross product is defined as

$$\mathrm{nX}(\mathbf{S}) = \begin{bmatrix} \det \mathbf{S}^{[1]} & \dots & (-1)^{i+1} \det \mathbf{S}^{[i]} & \dots & (-1)^{n+1} \det \mathbf{S}^{[n]} \end{bmatrix}^T \tag{2.7}$$

In a similar way, the matrix $\mathbf{G}^{<\gamma,\dots,\eta>}$ can be introduced where the superscript $\gamma,\dots,\eta$ denotes that the generators with the indices $e-n+1$ are removed from the generator matrix $\mathbf{G} = [\mathbf{g}^{(1)},\dots,\mathbf{g}^{(e)}]$ whose e number of generators are assumed to be independent. The set representation conversion resulting in H-representation $\mathbf{C}_H$ and $\mathbf{d}_H$ are given by

$$\mathbf{C}_H = \begin{bmatrix} \mathbf{C}^+ \\ -\mathbf{C}^+ \end{bmatrix} \quad \text{where} \quad \mathbf{C}^+ = \begin{bmatrix} \mathbf{C}_1^+ \\ \vdots \\ \mathbf{C}_\nu^+ \end{bmatrix} \quad \text{and} \quad \mathbf{C}_i^+ = \frac{\mathrm{nX}(\mathbf{G}^{<\gamma,\dots,\eta>})^T}{||\mathrm{nX}(\mathbf{G}^{<\gamma,\dots,\eta>})||_2} \tag{2.8}$$

$$\mathbf{d}_H = \begin{bmatrix} \mathbf{d}^+ \\ \mathbf{d}^- \end{bmatrix} = \begin{bmatrix} \mathbf{C}^+\mathbf{c} + \Delta\mathbf{d} \\ -\mathbf{C}^+\mathbf{c} + \Delta\mathbf{d} \end{bmatrix} \quad \text{where} \quad \Delta\mathbf{d} = \sum_{\nu=1}^{p} |\mathbf{C}^+\mathbf{g}^{(\nu)}| \tag{2.9}$$

where the index i goes from 1 to $\nu = \binom{e}{n-1}$.

2.2 Model predictive control

If a collision is detected in CCAS, new safe trajectories for the controllable vehicle are calculated by using MPC. In Section 2.2.1 an introduction to the MPC framework is presented in order to clarify its main elements consisting of the model, cost function, equality constraints and inequality constraints. The model which the MPC utilizes are discussed in Section 2.2.2. The cost function are presented in Section 2.2.3 and the equality and inequality constraints are presented in Section 2.2.4 and 2.2.5 respectively.

2.2.1 Model predictive control framework

MPC is a quadratic optimization problem which relies on predictions based on a linear time invariant model defined as

$$\mathbf{X}(k+1|k) = \mathbf{A}(k|k)\mathbf{X}(k|k) + \mathbf{B}(k|k)\mathbf{U}(k|k) \tag{2.10}$$

where $\mathbf{X}(k|k)$ is the state vector for the system with the system and input matrices $\mathbf{A}^i$ and $\mathbf{B}^i$ respectively describing the evolution of the system. The model formulates future desired dynamics for the system up to a prediction horizon H_x by repeatedly reusing the model to define all proceeding time steps. A set of the control inputs up to a control horizon H_u is included to the optimization problem in order to achieve the desired outcome depending on the system. H_u can be defined up to H_x but control inputs defined past H_u are ordinarily considered to be constant. Solving the optimization problem results in a sequence of control inputs that can be applied for each time step. By the concept of receding horizon however, only the most current input is applied after which the process is repeated at every new time step. Ordinarily there are physical limitations that have to be considered in order to sustain feasible solutions, in that case those limitations are formulated as a sequence of lower bounded linear equations.

The MPC uses a quadratic optimization problem, meaning that all optimization variables are squared by them self. They are also individually weighted by some scalar values. The MPC then attempts to minimize the summation of these squared and scaled variables, therefore the scalar weights formulates priority between those. This can be structured in a standard form

$$\begin{aligned} \min_{\mathbf{z}} J(k|k) &= \frac{1}{2}\mathbf{z}^T(k|k)\mathbf{H}(k|k)\mathbf{z}(k|k) \\ &s.t. \\ \mathbf{A}_{eq}\mathbf{z}(k|k) &= \mathbf{B}_{eq}(k|k) \\ \mathbf{A}_{in}\mathbf{z}(k|k) &\leq \mathbf{B}_{in}(k|k) \end{aligned} \tag{2.11}$$

where $\mathbf{z}$ is a vector containing the states and inputs to be minimized, individually weighted by scalar values in the diagonal matrix $\mathbf{H}(k|k)$. The matrices $\mathbf{A}_{eq}$ and $\mathbf{B}_{eq}(k|k)$ formulates the desired dynamics of the system while the linear inequality constraints formed by the $\mathbf{A}_{in}$ and $\mathbf{B}_{in}(k|k)$ matrices set the limitations [18].

A useful aspect of the MPC is that it can solve a shared optimization problem between different sub systems, allowing control over the interaction between the different systems. Assuming that all subsystems are linear and time invariant then changes the equation 2.10 to

$$\mathbf{X}^i(k+1|k) = \mathbf{A}(k|k)^i\mathbf{X}^i(k|k) + \mathbf{B}(k|k)^i\mathbf{U}^i(k|k) \tag{2.12}$$

for $i = 1, 2, \ldots, n_a$ where n_a is the number of subsystems.

2.2.2 Error model

As the MPC solves a minimization problem, desired states need to be introduced as the offset from their desired outcome. By doing so the error from the desired outcome is instead minimized. These error states, denoted $\tilde{\mathbf{X}}(k|k)$, describes the deviation between the state $\mathbf{X}(k|k)$ of the system that is to be controlled and the reference $\mathbf{X}_r(k|k)$ describing the desired outcome. The error model then becomes

$$\tilde{\mathbf{X}}(k|k) = \mathbf{X}(k|k) - \mathbf{X}_r(k|k) \tag{2.13}$$

In an ideal case, $\tilde{\mathbf{X}}(k|k)$ becomes zero at which a desired behaviour is achieved.

2.2.3 Cost function

To set up the cost function, both the vector of controlled variables, $\mathbf{z}(k|k)$, and diagonal weighting matrix $\mathbf{H}(k|k)$ have to be defined. The vector $\mathbf{z}(k|k)$ contains the future error state vectors $\tilde{\mathbf{X}}(k)^i$ and inputs $\mathbf{U}(k)^i$ for each vehicle $i = 1, 2, \ldots, n_a$ up to the prediction and control horizon H_x and H_u respectively and is defined as

$$\begin{aligned}
\mathbf{z}(k|k) = [&\tilde{\mathbf{X}}^1(k+1|k), \tilde{\mathbf{X}}^1(k+2|k), \ldots, \tilde{\mathbf{X}}^1(k+H_x|k), \\
&\mathbf{U}^1(k|k), \mathbf{U}^1(k+1|k), \ldots, \mathbf{U}^1(k+H_u|k), \\
&\tilde{\mathbf{X}}^2(k+1|k), \tilde{\mathbf{X}}^2(k+2|k), \ldots, \tilde{\mathbf{X}}^2(k+H_x|k), \\
&\mathbf{U}^2(k|k), \mathbf{U}^2(k+1|k), \ldots, \mathbf{U}^2(k+H_u|k), \\
&\quad \cdots \\
&\tilde{\mathbf{X}}^{n_a}(k+1|k), \tilde{\mathbf{X}}^{n_a}(k+2|k), \ldots, \tilde{\mathbf{X}}^{n_a}(k+H_x|k), \\
&\mathbf{U}^{n_a}(k|k), \mathbf{U}^{n_a}(k+1|k), \ldots, \mathbf{U}^{n_a}(k+H_u|k)]^T
\end{aligned} \tag{2.14}$$

The quadratic cost of the vehicles is weighted by the diagonal matrix $\mathbf{H}$ that assigns weights on the error state vector and input vector for each vehicle accordingly

$$\mathbf{H}(k|k) = \begin{bmatrix}
\mathbf{Q}^1(k|k) & \mathbf{0} & & \cdots & & & \mathbf{0} \\
\mathbf{0} & \mathbf{R}^1(k|k) & & & & & \\
& & \mathbf{Q}^2(k|k) & & & & \\
\vdots & & & \mathbf{R}^2(k|k) & & & \vdots \\
& & & & \ddots & & \\
& & & & & \mathbf{Q}^{n_a}(k|k) & \mathbf{0} \\
\mathbf{0} & & & \cdots & & \mathbf{0} & \mathbf{R}^{n_a}(k|k)
\end{bmatrix} \tag{2.15}$$

The error state weighting matrix $\mathbf{Q}^i$ is defined for each vehicle i as

$$\mathbf{Q}^i(k|k) = \begin{bmatrix} \mathbf{q}^i(k+1|k) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{q}^i(k+2|k) & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{q}^i(k+H_x|k) \end{bmatrix} \tag{2.16}$$

and for each specific element $1, \ldots, n$ in the state vector by the matrix

$$\mathbf{q}^i(k+j|k) = \begin{bmatrix} q_1^i(k+j|k) & 0 & \cdots & 0 \\ 0 & q_2^i(k+j|k) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & q_n^i(k+j|k) \end{bmatrix} \tag{2.17}$$

where the weight could be set for each individual time step $j = 1, 2, \ldots, H_x$.

In the same way, the weighting matrix $\mathbf{R}^i$ in the H matrix is defined as

$$\mathbf{R}^i(k|k) = \begin{bmatrix} \mathbf{r}^i(k|k) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{r}^i(k+1|k) & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{r}^i(k+H_u|k) \end{bmatrix} \tag{2.18}$$

in turn consisting of the matrices

$$\mathbf{r}^i(k+l|k) = \begin{bmatrix} r_1^i(k+l|k) & 0 & \cdots & 0 \\ 0 & r_2^i(k+l|k) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & r_m^i(k+l|k) \end{bmatrix} \tag{2.19}$$

where $r_1^i(k+l|k), \ldots, r_m^i(k+l|k)$ are scalar weights on each individual element of the control inputs $1, \ldots, m$ for each time step $l = 0, 1, \ldots, H_u$.

The linear sum of quadratic variables, as a result of the diagonal matrix multiplica-

tions, is expressed as

$$\begin{aligned} J = \frac{1}{2}\sum_{j=1}^{H_x} \Big(&(\tilde{\mathbf{X}}^1(k+j|k))^T \mathbf{q}^1(k+j|k)(\tilde{\mathbf{X}}^1(k+j|k)) + \\ &(\tilde{\mathbf{X}}^2(k+j|k))^T \mathbf{q}^2(k+j|k)(\tilde{\mathbf{X}}^2(k+j|k)) + \ldots + \\ &(\tilde{\mathbf{X}}^{n_a}(k+j|k))^T \mathbf{q}^{n_a}(k+j|k)(\tilde{\mathbf{X}}^{n_a}(k+j|k))\Big) + \\ \frac{1}{2}\sum_{l=0}^{H_u} \Big(&(\mathbf{U}^1(k+l|k))^T \mathbf{r}^1(k+l|k)(\mathbf{U}^1(k+l|k)) + \\ &(\mathbf{U}^2(k+l|k))^T \mathbf{r}^2(k+l|k)(\mathbf{U}^2(k+l|k)) + \ldots + \\ &(\mathbf{U}^{n_a}(k+l|k))^T \mathbf{r}^{n_a}(k+l|k)(\mathbf{U}^{n_a}(k+l|k))\Big) \end{aligned} \tag{2.20}$$

where the derivation of this expression is found in Appendix A.

2.2.4 Equality constraints

The error state to be minimized in order to achieve the desired dynamics for each subsystem are defined as

$$\begin{aligned} &\tilde{\mathbf{X}}^i(k+H_x|k) = \\ &= \mathbf{A}\mathbf{X}^i(k+H_x-1|k) + \mathbf{B}\mathbf{U}^i(k+H_x-1|k) - \mathbf{X}_r^i(k+H_x|k) \\ &= \mathbf{A}(\mathbf{A}\mathbf{X}^i(k+H_x-2|k) + \mathbf{B}\mathbf{U}^i(k+H_x-2|k)) + \\ &\qquad\qquad \mathbf{B}\mathbf{U}^i(k+H_x-1|k) - \mathbf{X}_r^i(k+H_x|k) \\ &\quad \vdots \\ &= \mathbf{A}^{H_x}\mathbf{X}^i(k|k) + \mathbf{A}^{H_x-1}\mathbf{B}\mathbf{U}^i(k|k) + \mathbf{A}^{H_x-2}\mathbf{B}\mathbf{U}^i(k+1|k) + \cdots + \\ &\qquad\qquad \mathbf{B}\mathbf{U}^i(k+H_x|k) - \mathbf{X}_r^i(k+H_x|k) \\ &= \mathbf{A}^{H_x}\mathbf{X}^i(k|k) + \sum_{i=0}^{H_x-1} \mathbf{A}\mathbf{B}\mathbf{U}^i(k+i|k) - \mathbf{X}_r^i(k+H_x|k) \end{aligned} \tag{2.21}$$

where the index i and time step is omitted for the matrices $\mathbf{A}$ and $\mathbf{B}$ for simplicity.

This sum is then reoriented and stacked into a matrix representation forming the matrices $\mathbf{A}_{eq}^i$ and $\mathbf{B}_{eq}^i(k|k)$ as

$$\mathbf{A}_{eq}^i = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \ldots & & \mathbf{0} & -\mathbf{B} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & & & \vdots & -\mathbf{A}\mathbf{B} & -\mathbf{B} & & \mathbf{0} \\ \vdots & & \ddots & & \vdots & -\mathbf{A}^2\mathbf{B} & -\mathbf{A}\mathbf{B} & \ddots & \\ & & & & \mathbf{0} & \vdots & \vdots & & \\ \mathbf{0} & \ldots & & \ldots\ \mathbf{0} & \mathbf{I} & -\mathbf{A}^{H_x-1}\mathbf{B} & -\mathbf{A}^{H_x-2}\mathbf{B} & \ldots & -\mathbf{B} \end{bmatrix} \tag{2.22}$$

$$\mathbf{B}_{eq}^{i}(k|k) = \begin{bmatrix} \mathbf{A}\mathbf{X}^{i}(k|k) - \mathbf{X}_{r}^{i}(k+1|k) \\ \mathbf{A}^{2}\mathbf{X}^{i}(k|k) - \mathbf{X}_{r}^{i}(k+2|k) \\ \vdots \\ \mathbf{A}^{H_x}\mathbf{X}^{i}(k|k) - \mathbf{X}_{r}^{i}(k+H_x|k) \end{bmatrix} \tag{2.23}$$

The complete equality matrices $\mathbf{A}_{eq}$ and $\mathbf{B}_{eq}(k|k)$ are formed by concatenating the matrices $\mathbf{A}_{eq}^{i}$ and $\mathbf{B}_{eq}^{i}(k|k)$ for all individual systems, as

$$\mathbf{A}_{eq} = \begin{bmatrix} \mathbf{A}_{eq}^{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{eq}^{2} & & \vdots \\ \vdots & & \ddots & 0 \\ \mathbf{0} & \cdots & 0 & \mathbf{A}_{eq}^{n_a} \end{bmatrix} \qquad \mathbf{B}_{eq}(k|k) = \begin{bmatrix} \mathbf{B}_{eq}^{1}(k|k) \\ \mathbf{B}_{eq}^{2}(k|k) \\ \vdots \\ \mathbf{B}_{eq}^{n_a}(k|k) \end{bmatrix} \tag{2.24}$$

2.2.5 Inequality constraints

The inequality constraints describes physical limitations on both states and inputs to the system. Given that there is an upper bound $\mathbf{X}^{i}(k|k) \leq \bar{\mathbf{X}}^{i}(k|k)$, then the inequality constraint is formulated as

$$\begin{gathered} \tilde{\mathbf{X}}^{i}(k|k) = \mathbf{X}^{i}(k|k) - \mathbf{X}_{r}^{i}(k|k) \\ \Leftrightarrow \\ \tilde{\mathbf{X}}^{i}(k|k) + \mathbf{X}_{r}^{i}(k|k) = \mathbf{X}^{i}(k|k) \leq \bar{\mathbf{X}}^{i}(k|k) \\ \Leftrightarrow \\ \tilde{\mathbf{X}}^{i}(k|k) \leq \bar{\mathbf{X}}^{i}(k|k) - \mathbf{X}_{r}^{i}(k|k) \end{gathered} \tag{2.25}$$

which is stacked for each time step forms the inequality function for all restricted states

$$\bar{\mathbf{A}}_{inX}^{i} = \underbrace{\left[\begin{array}{cccc} \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & & \vdots \\ \vdots & & \ddots & \\ \mathbf{0} & \dots & & \mathbf{I} \end{array}\right.}_{H_x} \underbrace{\left.\begin{array}{ccc} \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \mathbf{0} & \vdots \\ & & \ddots \\ \mathbf{0} & \dots & \mathbf{0} \end{array}\right]}_{H_u} \tag{2.26}$$

$$\bar{\mathbf{B}}^i_{inX}(k|k) = \begin{bmatrix} \bar{\mathbf{X}}^i(k+1|k) - \mathbf{X}^i_r(k+1|k) \\ \bar{\mathbf{X}}^i(k+2|k) - \mathbf{X}^i_r(k+2|k) \\ \vdots \\ \bar{\mathbf{X}}^i(k+H_x|k) - \mathbf{X}^i_r(k+H_x|k) \end{bmatrix} \tag{2.27}$$

Given that the inputs are also subjected to an upper constraint $\mathbf{U}^i(k|k) \leq \bar{\mathbf{U}}^i(k|k)$, then the stacked upper bounded inequality matrix is similarly defined as

$$\bar{\mathbf{A}}^i_{inU} = \underbrace{\begin{bmatrix} \mathbf{0} & \dots & & \mathbf{0} \\ \vdots & \mathbf{0} & & \vdots \\ & & \ddots & \\ \mathbf{0} & \dots & & \mathbf{0} \end{bmatrix}}_{H_x} \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & & \vdots \\ \vdots & & \ddots & \\ \mathbf{0} & \dots & & \mathbf{I} \end{bmatrix}}_{H_u} \tag{2.28}$$

$$\bar{\mathbf{B}}^i_{inU}(k|k) = \begin{bmatrix} \bar{\mathbf{U}}^i(k|k) \\ \bar{\mathbf{U}}^i(k+1|k) \\ \vdots \\ \bar{\mathbf{U}}^i(k+H_u|k) \end{bmatrix} \tag{2.29}$$

Lower bounds will also be of interest but in order to keep the standard notation $\mathbf{A}^i_{in}\mathbf{z}(k|k) \leq \mathbf{B}^i_{in}(k|k)$, the lower bound constraints must be reoriented

$$\begin{array}{ccc} \tilde{\mathbf{X}}^i(k|k) = \mathbf{X}^i(k|k) - \mathbf{X}^i_r(k|k) & & \\ \Leftrightarrow & & \mathbf{U}^i(k|k) \geq \underline{\mathbf{U}}^i(k|k) \\ \tilde{\mathbf{X}}^i(k|k) + \mathbf{X}^i_r(k|k) = \mathbf{X}^i \geq \underline{\mathbf{X}}^i(k|k) & & \Leftrightarrow \\ \Leftrightarrow & & -\mathbf{U}^i(k|k) \leq -\underline{\mathbf{U}}^i(k|k) \\ -\tilde{\mathbf{X}}^i(k|k) \leq -(\underline{\mathbf{X}}^i(k|k) - \mathbf{X}^i_r(k|k)) & & \end{array} \tag{2.30}$$

which equates to previous upper bounded inequality constraints except for the addition of minus signs and upper bound parameters being exchanged for lower bounds. All these inequality constraints are then stacked to form the complete inequality constraint definition

$$\mathbf{A}_{in}^{i} = \begin{bmatrix} \bar{\mathbf{A}}_{inX}^{i} \\ -\underline{\mathbf{A}}_{inX}^{i} \\ \bar{\mathbf{A}}_{inU}^{i} \\ -\underline{\mathbf{A}}_{inU}^{i} \end{bmatrix} \qquad \mathbf{B}_{in}^{i}(k|k) = \begin{bmatrix} \bar{\mathbf{B}}_{inX}^{i}(k|k) \\ -\underline{\mathbf{B}}_{inX}^{i}(k|k) \\ \bar{\mathbf{B}}_{inU}^{i}(k|k) \\ -\underline{\mathbf{B}}_{inU}^{i}(k|k) \end{bmatrix} \tag{2.31}$$

The complete inequality matrices $\mathbf{A}_{in}$ and $\mathbf{B}_{in}$ are formed by structuring the matrices $\mathbf{A}_{in}^{i}$ and $\mathbf{B}_{in}^{i}$ diagonally as

$$\mathbf{A}_{in} = \begin{bmatrix} \mathbf{A}_{in}^{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{in}^{2} & & \vdots \\ \vdots & & \ddots & 0 \\ \mathbf{0} & \cdots & 0 & \mathbf{A}_{in}^{n_a} \end{bmatrix} \qquad \mathbf{B}_{in}(k|k) = \begin{bmatrix} \mathbf{B}_{in}^{1}(k|k) \\ \mathbf{B}_{in}^{2}(k|k) \\ \vdots \\ \mathbf{B}_{in}^{n_a}(k|k) \end{bmatrix} \tag{2.32}$$

3

Development of the Centralized Collision Avoidance System

This chapter describes the development of the CCAS building on the theory in Chapter 2 based on the problem described in Chapter 1. First a motivation for what vehicle models that are used representing the controllable and uncontrollable vehicles is made in Section 3.1. The created RA algorithm which is based on [6] is then presented in Section 3.2. Section 3.3 describes the workings of the collision risk detection algorithm and Section 3.4 highlights the Linear Programming (LP) problem that arises in connection with the collision risk detection. This is followed by Section 3.5 that explains the setup of the MPC. Finally, the effect of the prediction horizon length for minimum intrusion is underlined in Section 3.6, along with its necessary condition to keep it minimized and how it was implemented.

3.1 Vehicle models

In order to model a vehicle different models can be used. Usually a simple but representative model is desired in order to model the dynamics of interest. Common model assumptions and the notations used in the mathematical models are presented in Section 3.1.1. A nonlinear BM model is then presented in Section 3.1.2 followed by the linear BM presented in Section 3.2.2. Finally a simple but representative model used to model the controllable vehicles are presented in Section 3.1.4.

3.1.1 Modelling a vehicle

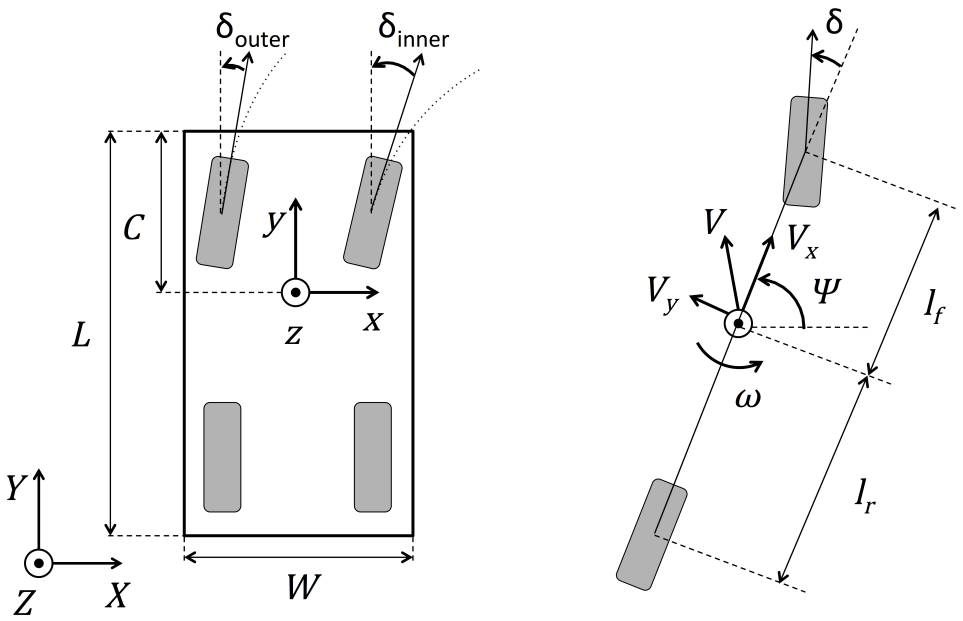
The considered vehicles in this thesis consists of four wheeled cars and two different types of models are utilized: the BM and the Constant Velocity (CV) model. One assumption for the BMs are that the steering angle δ of the front wheels are equal, i.e. $\delta_{outer} = \delta_{inner} = \delta$. This is a commonly used simplification and why it is refereed to as a bicycle model. In practice, each wheel will have different traveling radii as depicted in figure 3.1a along with the dimensions used to describe all vehicles in the scenario. As the name indicates, the BM could be associated with a bicycle

consisting of a rear wheel that is fixed in the longitudinal direction of the vehicle and a front wheel used for steering, see figure 3.1b. In contrast to the CV model, the BM takes into account that the vehicle cannot move sideways without moving forward which makes the model more representative for a turning vehicle.

The BM describes the vehicles local dynamics along its Cartesian coordinate frame with its origin fixed in its centre of gravity (CoG) and its longitudinal axis pointing towards the front of the vehicle and its lateral axis pointing towards the left of the vehicle according to Figure 3.1a. These axes are generalized as the local x- and y-coordinates respectively. The local z-axis are perpendicular to both the x- and y-axis, pointing upwards from CoG. The global coordinate frame are the standard Cartesian coordinate system aligned with the intersection scenario as previously depicted in Figure 1.1a with the global X- and Y-axes pointing right and up respectively.

The dynamics of a vehicle depends on its speed and hence, different bicycle models could be used for different speeds which provides better representations of the dynamics of the vehicle. For low speeds the assumption that the vehicle follows the direction of its wheels could be made. This is excluded for higher speeds representation as this is to be considered in this thesis, limiting the lower speeds to $5m/s$ or higher. In that case, the slip angle are nonzero which results in a force perpendicular to the direction of the wheels, implying that the corning stiffness has to be included in the model [19].

Discussion with people at Volvo cars led to assuming that all vehicles are able to apply steering angles of $\delta \in [-45, 45]$ degrees and accelerations of $a \in [-4, 2]m/s^2$.



(a) Steering angles and dimensions of the vehicle.

(b) Bicycle model with its associated parameters.

Figure 3.1: To the left, a the vehicle are depicted together with its dimensions, steering angles and the global and local coordinate frame. To the right, the bicycle model with its associated parameters are illustrated.

3.1.2 Nonlinear bicycle model

A nonlinear BM used for a path following MPC framework is found in [20] yielding a successful result and was therefor adopted to this thesis. The state of the model consists of the global positions X and Y, local velocities V_x and V_y heading angle Ψ and heading angle rate ω defined in accordance to Figure 3.1b. The dynamics of the model, taking an applied force F_x and steering angle δ as input, are described by the set of Ordinary Differential Equations (ODE)

$$f_{NL} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{V}_x \\ \dot{V}_y \\ \dot{\Psi} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} V_x \cos(\Psi) - V_y \sin(\Psi) \\ V_x \sin(\Psi) - V_y \cos(\Psi) \\ \frac{F_x}{2M}(\cos(\delta) + 1) - \frac{C_f}{M}(\delta - \frac{V_y + l_f\omega}{V_x}) \sin(\delta) + \omega V_y \\ \frac{F_x}{2M} \sin(\delta) - \frac{C_f}{M}(\delta - \frac{V_y + l_f\omega}{V_x}) \cos(\delta) + \frac{C_r}{M}(-\frac{V_y + l_f\omega}{V_x}) \cos(\delta) - \omega V_x \\ \omega \\ \frac{F_x l_f}{2I_z} \sin(\delta) + \frac{C_f l_f}{I_z}(\delta - \frac{V_y + l_f\omega}{V_x}) \cos(\delta) - \frac{C_r l_r}{I_z}(-\frac{V_y + l_f\omega}{V_x}) \cos(\delta) \end{bmatrix} \tag{3.1}$$

The annotations C_f and C_r denotes the corning stiffnesses of the front and rear wheels respectively, M is the mass and I_z is the moment of inertia around the z-axis. Finally, l_f and l_r are the lengths between the CoG and the front and rear wheel axis respectively.

3.1.3 Linear bicycle model

A simpler linear bicycle model whose state vector only consists of the local position y and velocity V_y together with the heading angle Ψ and heading angle rate ω is found in [19]. This model is linear allowing linear mathematical operations to be applied. A linear state space representation of this model are formulated as

$$\dot{\mathbf{X}}^{BM} = \mathbf{A}^{BM}\mathbf{X}^{BM} + \mathbf{B}^{BM}\mathbf{U}^{BM} \tag{3.2}$$

where

$$\mathbf{A}^{BM} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_f+2C_r}{MV_x^{const}} & 0 & -V_x - \frac{2C_f l_f - 2C_r l_r}{MV_x^{const}} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_f l_f - 2C_r l_r}{I_z V_x^{const}} & 0 & -\frac{2C_f l_f^2 + 2C_r l_r^2}{I_z V_x^{const}} \end{bmatrix} \qquad \mathbf{B}^{BM} = \begin{bmatrix} 0 \\ \frac{2C_f}{M} \\ 0 \\ \frac{2C_f l_f}{I_z} \end{bmatrix} \tag{3.3}$$

$$\mathbf{X}^{BM} = \begin{bmatrix} y & V_y & \Psi & \omega \end{bmatrix}^{uc} \qquad \mathbf{U}^{BM} = \delta \tag{3.4}$$

having the steering angle δ as input. This model assumes that the longitudinal velocity V_x^{const} is kept constant.

3.1.4 Constant velocity model

A model that does not take the heading angle of the vehicle into account is the CV model. This is a simple model that in the one dimensional case, describes the movement of the CoG of an object along a straight path. The state $\mathbf{X}^{CV}$ of the model

$$\dot{\mathbf{X}}^{CV} = \mathbf{A}^{CV}\mathbf{X}^{CV} + \mathbf{B}^{CV}\mathbf{U}^{CV} \tag{3.5}$$

where

$$\mathbf{A}^{CV} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad \mathbf{B}^{CV} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.6}$$

$$\mathbf{X}^{CV} = \begin{bmatrix} x & V_x \end{bmatrix}^{uc} \qquad \mathbf{U}^{CV} = a \tag{3.7}$$

consist of the local position x and velocity V_x along the longitudinal direction of the vehicle. The input to the model is the acceleration a.

3.2 Reachability analysis

Initially the nonlinear BM was considered to be used for the RA as this would allow for more narrow reachable sets. It was however excluded as being to computationally demanding as it requires nonlinear RA, further motivated under chapter 4. It is still included in Section 3.2.1 as the approach of using nonlinear RA is most relevant for discussing future work.

An other approach based on linear RA was investigated, focusing on a model which was based on the linear BM and the CV model and whose state consisted of the longitudinal and lateral velocities together with the heading and yaw rate which is presented in Section 3.2.2. Since the velocities in this new model is defined in the local coordinate frame of the vehicle, a state transformation interval matrix, presented in Section 3.2.3, which is based on a rotation matrix is implemented in the created RA algorithm found in Section 3.2.4. To facilitate the computational burden of the algorithm, a function described in Section 3.2.5 that reduces the number of generators representing the set are implemented and could be executed for fixed time steps throughout the execution of the algorithm.

3.2.1 Nonlinear reachability analysis

The nonlinear model presented in section 3.1.2 was implemented in a nonlinear RA algorithm available at [21] and the underlying theory of the algorithm are found in [8]. A conceptual flowchart of the algorithm illustrated in Figure 3.2. The main idea with the nonlinear RA algorithm is to linearize the system around a certain operating point, from which the linearization error L and metric $\bar{L}$ could be calculated. The linearization error is then compared to the metric specifying whether the linearization of the system around the operating point is still representative enough or not. If not, the reachable set is splitted into two sets both containing a new linearization point from which the process is repeated. When all eventually splitted sets are approved, the new reachable set is calculated. The process iterates until the desired final time is reached.

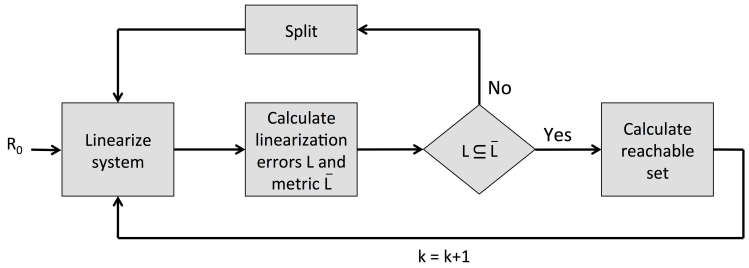


Figure 3.2: A conceptual flowchart of the preexisting nonlinear RA algorithm for with the nonlinear BM is implemented. Firstly, the model is linearized after with the linearization error and metric $\bar{L}$ are calculated. If the linearization is not representative enough the reachable set are splitted, otherwise a new reachable set are achieved.

3.2.2 Linear bicycle model

The main idea of the linear RA approach is to solve the local reachable sets of the uncontrollable vehicle and then transform these sets in an over approximating way in order to get its global coordinates. To this end there is no point in calculating the local coordinates as it was only the local changes in position that are used together with their associated orientation at that point in time, working from the initial global position. To form a linear model used for the linear RA algorithm, the position states of the BM and CV model presented in section 3.1.3 and 3.1.4 are excluded and the remaining states from the two models are concatenated. By merging the two models, the RA algorithm returns zonotopes of matching sizes. The resulting model f_{uc} yields the state space representation

$$f_{uc} = \begin{bmatrix} \dot{V}_x \\ \dot{V}_y \\ \dot{\Psi} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{2C_f+2C_r}{M\bar{V}_x^{const}} & 0 & -\bar{V}_x^{const} - \frac{2C_f l_f - 2C_r l_r}{M\bar{V}_x^{const}} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_f l_f - 2C_r l_r}{I_z \bar{V}_x^{const}} & 0 & -\frac{2C_f l_f^2 + 2C_r l_r^2}{I_z \bar{V}_x^{const}} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \Psi \\ \omega \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & \frac{2C_f}{M} \\ 0 & 0 \\ 0 & \frac{2C_f l_f}{I_z} \end{bmatrix} \begin{bmatrix} a \\ \delta \end{bmatrix} \tag{3.8}$$

with the state consisting of the local velocities V_x and V_y, the global heading angle Ψ and the heading angle rate ω. The input to the model is the longitudinal acceleration a and steering angle δ. Note that the previously defined V_x^{const} is replaced with $\bar{V}_x^{const}$ which is an over approximation of the longitudinal velocity that is restricted to be constant for the BM in section 3.1.3 to be valid.

For a given prediction horizon H_x and maximum acceleration $\bar{a}$, the maximum ve-

locity that could be achieved was given by

$$\bar{V}_x^{const} = V_x^{const} + \bar{a}tH_x \tag{3.9}$$

where t was the sampling time. This over approximation ensures the worst possible dynamic outcome of the model during the execution of the RA algorithm.

3.2.3 State transformation interval matrix

The model used in the RA algorithm describes the velocity of the vehicle in its local coordinate frame. Hence, a transformation to the global coordinate frame is needed for further computations. In this case, a state transformation interval matrix is introduced based on a rotation matrix. A priori, the heading angle of the uncontrollable vehicle Ψ is set to zero. During the execution of the RA algorithm, this angle would take different values spanning an interval depending on the applied inputs to the model and its evolution through time.

In the general case, an arbitrary point $\mathbf{p} = [x, y]$ in $\mathbb{R}^2$ rotated θ radians counter clockwise around the origin is expressed as $\mathbf{p}^{rotated} = \mathbf{M}(\theta)\mathbf{p}$ where the rotation matrix is defined as

$$\mathbf{M}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{3.10}$$

In this particular case, the rotation matrix $\mathbf{M}$ is evaluated for two cases depending of the heading angle interval $\Psi^I = [\underline{\Psi}, \bar{\Psi}]$ resulting in the state transformation interval matrix

$$M^I = \left[\underline{\mathbf{M}}, \bar{\mathbf{M}}\right] \tag{3.11}$$

The values of the interval matrix are achieved by evaluating the maximum and minimum value of each element in the rotation matrix for angles within the interval. In the first case, Ψ^I contains the angle $\Psi = 0$ and the lower limit and upper limit of the state transformation matrix are calculated to be

$$\underset{0 \in \Psi^I}{\text{Case 1}} \qquad \underline{\mathbf{M}} = \begin{bmatrix} \cos(\bar{\Phi}) & -\sin(\bar{\Psi}) \\ \sin(\underline{\Psi}) & \cos(\bar{\Phi}) \end{bmatrix} \qquad \bar{\mathbf{M}} = \begin{bmatrix} 1 & -\sin(\underline{\Psi}) \\ \sin(\bar{\Psi}) & 1 \end{bmatrix} \tag{3.12}$$

where

$$\bar{\Phi} = \begin{cases} |\bar{\Psi}|, & \text{if } |\underline{\Psi}| < |\bar{\Psi}| \\ |\underline{\Psi}|, & \text{otherwise} \end{cases} \qquad \underline{\Phi} = \begin{cases} |\underline{\Psi}|, & \text{if } |\underline{\Psi}| < |\bar{\Psi}| \\ |\bar{\Psi}|, & \text{otherwise} \end{cases}$$

An illustration for the first case can be seen in Figure 3.3.

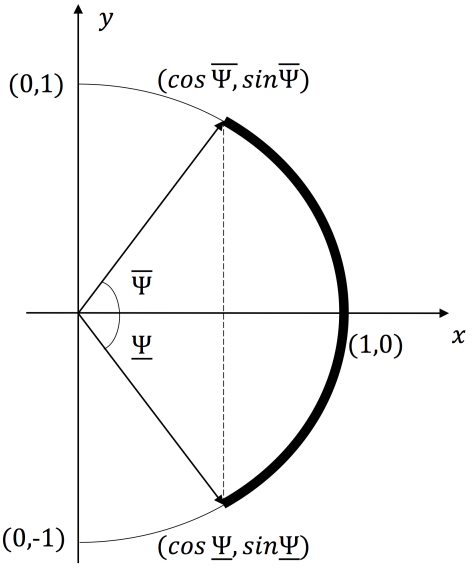


Figure 3.3: An illustration of the heading angle interval Ψ^I that includes the heading angle $\Psi = 0$. From this interval, the elements of the rotation matrix can individually be evaluated in order to form the state transformation interval matrix.

In the second case $\Psi = 0$ is not contained in Ψ^I and hence, the state transformation interval matrix was calculated to be

$$\underset{0 \notin \Psi^I}{\text{Case 2}} \qquad \underline{\mathbf{M}} = \begin{bmatrix} \cos(\bar{\Phi}) & -\sin(\bar{\Psi}) \\ \sin(\underline{\Psi}) & \cos(\bar{\Phi}) \end{bmatrix} \qquad \bar{\mathbf{M}} = \begin{bmatrix} \cos(\underline{\Phi}) & -\sin(\underline{\Psi}) \\ \sin(\bar{\Psi}) & \cos(\underline{\Phi}) \end{bmatrix} \tag{3.13}$$

Given the initial position of the uncontrollable vehicle, the now global changes in position could be added for every step forward in time, representing the global reachable set of positions the uncontrollable vehicle might occupy.

3.2.4 Reachability analysis algorithm

The implemented RA algorithm is based on Algorithm 2 in [6] where also a more detailed explanation of the functions in the algorithm and related proofs could be found. The algorithm assumes that the origin is contained in the input set $\mathcal{U}$ and before the execution of the the implemented algorithm, the accessible data from the scenario have to be prepared.

Given the measured state of the uncontrollable vehicle,

$$\mathbf{X} = [X, Y, V_x, V_y, \Psi, \omega]^{uc} \tag{3.14}$$

with its associated uncertainties

$$\boldsymbol{\Delta}_X = [\Delta_X, \Delta_Y, \Delta_{V_x}, \Delta_{V_y}, \Delta_\Psi, \Delta_\omega]^{uc} \tag{3.15}$$

the initial set $\mathcal{X}_0$ and input set $\mathcal{U}$ are defined as

$$\mathcal{X}_0 = \begin{bmatrix} [\underline{V}_x, \bar{V}_x] \\ [\underline{V}_y, \bar{V}_y] \\ [\underline{\Psi}, \bar{\Psi}] \\ [\underline{\omega}, \bar{\omega}] \end{bmatrix} \qquad \mathcal{U} = \begin{bmatrix} [\underline{\delta}, \bar{\delta}] \\ [\underline{a}, \bar{a}] \end{bmatrix} \tag{3.16}$$

where

$$\underline{V}_x = V_x - \Delta_{V_x} \qquad \bar{V}_x = V_x + \Delta_{V_x} \tag{3.17}$$
$$\underline{V}_y = V_y - \Delta_{V_y} \qquad \bar{V}_y = V_y + \Delta_{V_y} \tag{3.18}$$
$$\underline{\Psi} = -\Delta_{\Psi} \qquad \bar{\Psi} = \Delta_{\Psi} \tag{3.19}$$
$$\underline{\omega} = \omega - \Delta_{\omega} \qquad \bar{\omega} = \omega + \Delta_{\omega} \tag{3.20}$$

noting that heading angle defined in $\mathcal{X}_0$ only consists of the uncertainties of the measured heading, i.e. the vehicle is forced to be aligned along the global x-axis with its CoG in the origin. After each iteration of the algorithm, the final result is rotated back to the initial orientation of the vehicle.

From the measured state, the set p_0 describing the initial global position of the vehicle was defined as

$$p_0 = \left[\underline{\mathbf{p}}, \bar{\mathbf{p}}\right] \qquad \begin{aligned} \underline{\mathbf{p}} &= \mathbf{p} - \boldsymbol{\Delta}_p \\ \bar{\mathbf{p}} &= \mathbf{p} + \boldsymbol{\Delta}_p \end{aligned} \tag{3.21}$$

where

$$\mathbf{p} = \begin{bmatrix} X \\ Y \end{bmatrix} \qquad \boldsymbol{\Delta}_p = \begin{bmatrix} \Delta_X \\ \Delta_Y \end{bmatrix} \tag{3.22}$$

The vehicle dimensions used in the algorithm is defined as the set

$$d = \begin{bmatrix} [C - L, C] \\ \left[-\frac{W}{2}, \frac{W}{2}\right] \end{bmatrix} \tag{3.23}$$

where L, W and C were introduced in section 3.1.3.

With this data available together with the prediction horizon H_x defining the number of time instances to be predicted and the time interval t the algorithm can be executed. The sets used in the algorithm can be found in Table 3.1 and a series of figures illustrating the initialization of the algorithm can be seen in Figure 3.4

for which the velocity or position dimension are considered. The initial state of the vehicle can be seen in Figure 3.4a.

Table 3.1: Sets used in the RA algorithm

Notation	Description
$\mathcal{X}_0$	Initial set
$\mathcal{U}$	Input set
$\mathcal{H}$	Homogeneous reachable set
V	Increment of particular reachable set
$\mathcal{P}$	Particular reachable set
$\mathcal{R}$	Local reachable set
$\mathcal{R}^p$	Unbiased global reachable set of positions
$\mathcal{R}^{uc}$	Biased global reachable set
r	Subset of local velocities

The first step of Algorithm 1 is to calculate the initial sets. The initial homogeneous set $\mathcal{H}_0$ is calculated with the convex hull operator, defined in chapter 2, enclosing the initial set and the propagated initial set together with a correction term $\mathcal{F}$ defined as

$$\mathcal{F} = \sum_{i=2}^{\eta}[(i^{\frac{-i}{i-1}} - i^{\frac{-1}{i-1}})t^i, 0]\frac{\mathbf{A}^i}{i!} + \mathcal{E}(t) \tag{3.24}$$

as shown in Figure 3.4b and 3.4c. The matrix exponential is used for the state propagation and approximated with an η-order Taylor expansion

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{1}{2!}(\mathbf{A}t)^2 + \frac{1}{3!}(\mathbf{A}t)^3 + \ldots + \frac{1}{\eta!}(\mathbf{A}t)^\eta \tag{3.25}$$

The correction term is introduced to compensate for the discretization error $\mathcal{E}$ due to the omitted terms of the Taylor series and the assumption that the state propagates linearly between two consecutive time instances. This implies that the reachable set will be over approximated to ensure that all possible occupied states are enclosed by the set. The discretization error is defined as

$$\mathcal{E}(t) = [\mathbf{-1}, \mathbf{1}]\frac{(||\mathbf{A}||_\infty t)^{\eta+1}}{(\eta+1)!}\frac{1}{1-\epsilon} \tag{3.26}$$

where $[\mathbf{-1}, \mathbf{1}]$ is an interval matrix whose elements consist of the interval [-1, 1] and the condition

$$\epsilon = \frac{||\mathbf{A}||_\infty t}{\eta+2} < 1 \tag{3.27}$$

had to be fulfilled by a sufficiently large choice of η according to [6].

Based on the input set $\mathcal{U}$ and the discretization error $\mathcal{E}$, V_0 and $\mathcal{P}_0$ are calculated which in this particular case are equal due to the independence of the two concatenated models as shown in Figure 3.4d. By adding the two sets, $\mathcal{H}_0$ and $\mathcal{P}_0$, the initial reachable set $\mathcal{R}_0$ is formed, illustrated in Figure 3.4e.

The uncertainty in position and the dimensions are added as illustrated in Figure 3.4f. The convex hull operator then encloses this set and the vehicle's change in position, i.e. the time step t multiplied with the subset r_0 defined as

$$r_k = \Big\{(V_x, V_y) \mid V_x, V_y \in \mathcal{R}_k\Big\} \quad \forall k \in \mathbb{N} \tag{3.28}$$

and showed in Figure 3.4g and 3.4h. Together they are rotated back and the previously removed bias in position is reintroduced resulting in $\mathcal{R}_0^{uc}$, see Figure 3.4i.

For the succeeding time instance, $\mathcal{H}_0$ and V_0 are propagated by use of the matrix exponential, forming $\mathcal{H}_1$ and V_1. The new increment V_1 is then added to the previously calculated $\mathcal{P}_0$ and the new particular solution $\mathcal{P}_1$ is achieved. Once again, $\mathcal{H}_1$ and $\mathcal{P}_1$ are added to form the reachable set $\mathcal{R}_1^{uc}$ which is rotated and added with the bias. This process iteratively continued up to time instance $k = H_x$. Note that the index annotation of the sets refers to the set within the time interval $[kt, (k+1)t]$.

During the execution of the algorithm, the number of generators in $\mathcal{P}$ is reduced with the function *GeneratorReduction* described in section 3.2.5 in order to reduce the computational burden. The box operator used in the algorithm is presented in chapter 2 together with elementary operations such as addition and matrix transformation and the interval rotation matrix $\mathbf{M}$ presented in section 3.2.3.

Input: $\mathbf{X}_0, \mathcal{X}_0, \mathcal{U}, d, H_x, t$
Output: $\mathcal{R}^{uc}$
$\mathcal{H}_0 = CH(\mathcal{X}_0, e^{At}\mathcal{X}_0) + \mathcal{F}\mathcal{X}_0$
$V_0 = \sum_{i=0}^{\eta} \left(\frac{A^i t^{i+1}}{(i+1)!}\mathcal{U}\right) + \mathcal{E}(t)t\mathcal{U}$
$\mathcal{P}_0 = \mathit{box}(V_0)$
$\mathcal{R}_0 = H_0 + \mathcal{P}_0$
$\mathcal{R}_0^p = CH(\mathbf{\Delta}_p + d, r_0 t)$
$\mathcal{R}_0^{uc} = \mathbf{M}(\mathbf{X}_0)\mathcal{R}_0^p + \mathbf{p}$
for $k = 1, \ldots, H_x$ **do**
  $\mathcal{H}_k = e^{At}\mathcal{H}_{k-1}$
  $V_k = e^{At}V_{k-1}$
  $\mathcal{P}_k = \mathcal{P}_{k-1} + \mathit{box}(V_k)$
  $\mathcal{R}_k = \mathcal{H}_k + \mathcal{P}_k$
  $\mathcal{R}_k^p = \mathcal{R}_{k-1}^p + M^I(\mathcal{R}_k) r_k t$
  $\mathcal{R}_k^{uc} = \mathbf{M}(\mathbf{X}_0)\mathcal{R}_k^p + \mathbf{p}$
end

Algorithm 1: Reachability analysis algorithm showing the structure of the algorithm for the linear RA with the state transformation interval matrix.

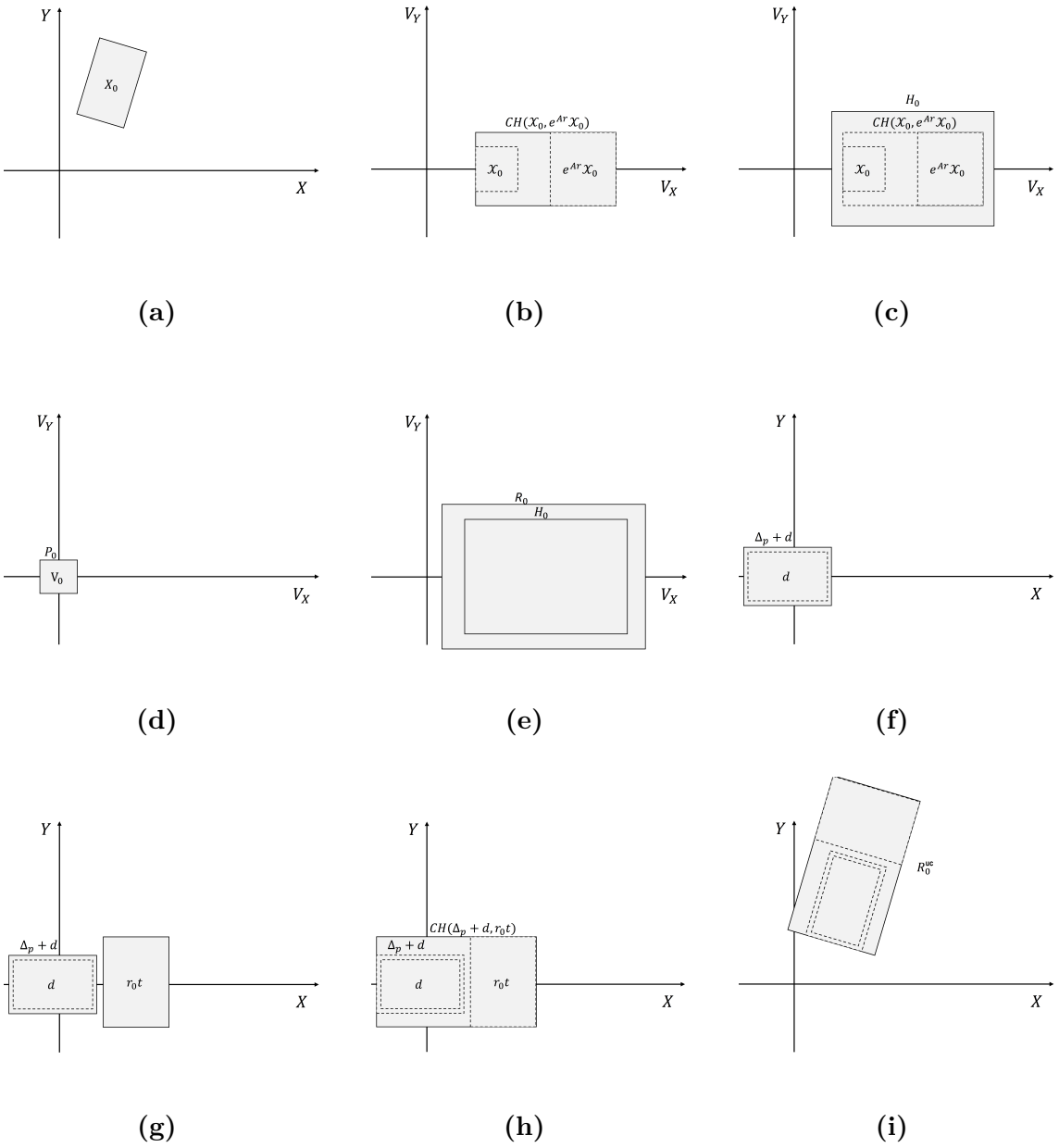


Figure 3.4: Initialization steps of RA algorithm, Algorithm 1.

3.2.5 Generator reduction

Due to the operation $\mathcal{P}_k = \mathcal{P}_{k-1} + \mathcal{b} o \chi(V_k)$ in Algorithm 1, the number of generators in the zonotope $\mathcal{P}_k$ increases linearly according to Equation (2.3). In order to limit this growth and the computational burden, the function *GeneratorReduction* is implemented which in an over-approximated fashion encloses the zonotope by over-representing the larger generators, eliminating the need for the smaller ones. The concept behind the algorithm and the used heuristic can be found in [6].

The created function assumes that the generators of the zonotope, whose number of generators is reduced, are not aligned with each other. If so, the two generators are replaced by the sum of the two. The generators are then sorted in a descending order and the zonotope is split into two, i.e. $\mathcal{Z} = \hat{\mathcal{Z}} + \check{\mathcal{Z}}$ where larger generators belongs to $\hat{\mathcal{Z}}$ and the smaller to $\check{\mathcal{Z}}$ and the centre of the original zonotope $\mathcal{Z}$ is inherited by any of the two. By then over approximate $\check{\mathcal{Z}}$ by the parallellotope $\Phi = \Pi \, \textit{box}(\Pi^{-1}\check{\mathcal{Z}})$ a resulting zonotope with a reduced number of generators is achieved as

$$\mathcal{Z}^{red} = \hat{\mathcal{Z}} + \Phi \tag{3.29}$$

To form the parallelotope, the matrix Π consisting of a selection of generators from $\check{\mathcal{Z}}$ had to be determined. The selection was based on an exhaustive search of finding the n_g generators maximizing the heuristic

$$v = |\det \left[\mathbf{g}^{(i_1)}, \ldots, \mathbf{g}^{(i_{n_g})}\right]|^{-1} \tag{3.30}$$

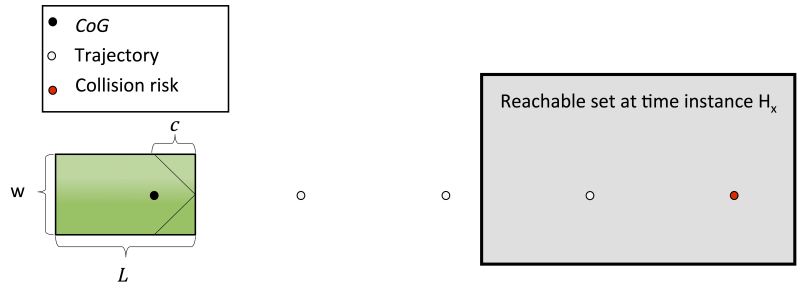
representing the inverse volume spanned by those.

3.3 Collision risk detection

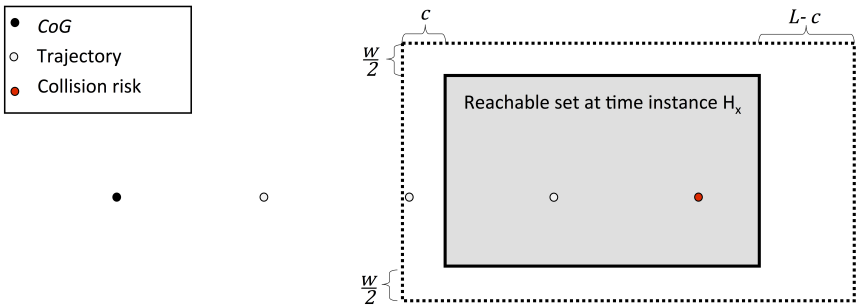
The collision detection algorithm operates at the top of the CCAS which individually checks the most current desired trajectory point from the controllable vehicles, determining if they intersect with the reachable set of the uncontrollableed vehicle. In this case the implemented CCAS takes over and alter the trajectories of the controllable vehicles to stay clear. The trajectories describes only the future positions of the controllable vehicles' CoG, therefore the physical dimensions of the vehicles must be included to ensure safe passage. This means that the algorithm must compare two sets to one another.

Instead of adding the dimensions to each controllable vehicle, by instead mirroring its dimensions in reference to it longitudinal direction, the dimensions could be directly added to the reachable set of the uncontrollable vehicle before checking for the collision risk. This allows the collision detection to be performed directly with the desired trajectory provided as CoGs. The set addition is illustrated in Figure 3.5 where the length C was added to the left, the length $L - C$ to the right and the width evenly distributed on the top and bottom of the set. This is done using zonotope arithmetics.

Before the comparison could be done, the reachable set of the uncontrollable vehicle had to be converted to a H-representation, both to match the structure used by the MPC but also because checking a G-representation directly is obscure. This conversion is done as presented in Section 2.1.2.5 such that the half plane representation



(a) Controllable vehicle and its future trajectory together with the zonotope of the uncontrollable vehicle at time instance H_x.



(b) *CoG* for the controllable vehicle and the expanded zonotope of the uncontrollable vehicle at time instance H_x.

Figure 3.5: Controllable vehicle's dimensions added to the last zonotope of the uncontrollable vehicle in order to be able to only compare the controllable vehicles CoG to the to the resulting set.

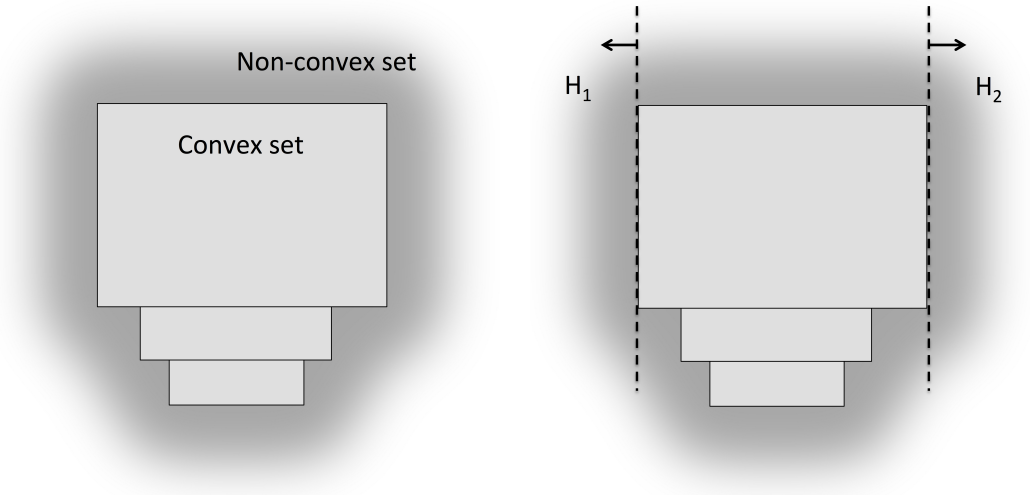
$$\mathbf{C}_H\mathbf{X}_r = \begin{bmatrix}\mathbf{C}_X & \mathbf{C}_Y\end{bmatrix}\begin{bmatrix}X_r \\ Y_r\end{bmatrix} \leq \mathbf{d}_H \tag{3.31}$$

is achieved. The matrix $\mathbf{C}_H$ row-wise scales the values of $\mathbf{X}_r$, corresponding to the reference trajectory point to be evaluated, which resulting values must remain below the values in the column vector $\mathbf{d}_H$ to be inside the set. The matrix $\mathbf{C}_H$ can be divided into the individual columns $\mathbf{C}_X$ and $\mathbf{C}_Y$ for each coordinate X_r and Y_r respectively. There is a total of two times n_g number of half planes to check where n_g is the number of generators of the zonotope. If the trajectory point is within all half planes then it must be within the set, otherwise the CCAS remains dormant.

3.4 Linear programming problem

The reachable sets of the uncontrollable vehicle is used to define the inequality constraints to the MPC, confining the altered trajectories from the MPC to safe

areas only. The MPC only handles convex inequality constraints and the resulting sets from the RA are convex but describe the forbidden area. The MPC needs the allowed set which becomes the opposite area as depicted in Figure 3.6a which is non-convex. Figure 3.6b depicts how the mirrored convex constraints H_1 and H_2 would tell the MPC to simultaneously stay to the left side of H_1 and to the right of H_2, telling it that the controllable vehicles must exist at two places at once which is physically impossible.



(a) The opposite non-convex set.

(b) Half planes H_1 and H_2 which together become non-convex.

Figure 3.6: Illustration of convex reachable sets from the RA algorithm, shoving opposite facets and how they causes contradictions to the MPC.

To remedy this issue a LP problem is formulated to select the relevant side of the set and discarding the rest, resulting in a convex set to pass to the MPC.

The LP problem sorts the shortest distances out of the set and checks in an ascending order if they are feasible given the limited amount of control input for the given direction based on the worst case outcome from section 3.6 for the given scenario. In turn this means that for strictly longitudinal control the trajectories of the controllable vehicles will be parallel to the global x-axis. Then the best option is formulated as the shortest distance out of the set along the global x-axis. The LP problem is formulated as

$$\begin{aligned} & Y_H = Y_r \\ \Rightarrow \min(|X_r - X_H|) = \min(|X_r - \frac{\mathbf{d}_H - \mathbf{C}_y Y_H}{\mathbf{C}_x}|) &= \min(|X_r - \frac{\mathbf{d}_H - \mathbf{C}_y Y_r}{\mathbf{C}_x}|) \end{aligned} \tag{3.32}$$

where X_r, Y_r,$\mathbf{C}_x$ and $\mathbf{C}_y$ are defined in Section 3.3 while X_H and Y_H corresponds to the intersecting point on the half plane and the trajectory of the controllable vehicle.

3.5 Model predictive control

In order to acquire the desired behaviour of the system, several aspects had to be evaluated and tuned for the MPC. In Section 3.5.1 all practical selections to implement the MPC are included. In Section 3.5.2 it is presented how the choice of the individual weights affect the behaviour of the system. Lastly in Section 3.5.3 it is derived how the transfer delays from the server to the controllable vehicles could be generically defined.

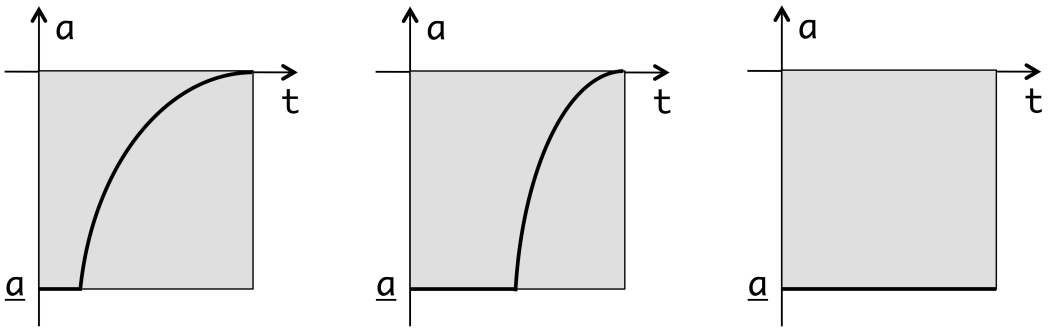
3.5.1 Practical choices

Having a generic implementation of the MPC algorithm allows for more flexibility with the SPAS environment as it during the course of this project have been in development itself. Therefore the prediction horizon and the linear model can be arbitrarily chosen. Since only longitudinal control is considered, the CV model along that dimension is used as it has a dimensionally small representation. The solver used for the MPC is taken from Matlab's *mpcqpsolver* as it is the only viable option at the time of the projects development that could achieve the projects needs, supporting C-code generation and the handling of changeable inequality constraints that would be active for time instances between $k+1$ and $k+H_x$.

3.5.2 Weighting matrices

All control inputs, being acceleration at different time steps for each controllable vehicle, up to the prediction horizon H_x are applied, meaning that the control horizon equals the prediction horizon in length minus one as the last input wont have an effect on the system.

To keep the prediction horizon as short as possible it was desirable to ensure that as much of the available control input remain available in that limited time span. The weighting matrix $\mathbf{H}$ is therefore formulated such that earlier control inputs are considered cheaper and therefore used to their fullest. By always utilizing the most of the earliest control inputs, all the energy are made available for the limited time horizon for any further observed collision risks. This is illustrated in Figure 3.7 where consecutively observed collision risks demands more energy for deceleration, allowing at most all defined control input for the given time horizon.



(a) First control input. **(b)** Intermediate control input. **(c)** Maximum control input.

Figure 3.7: Applied control inputs over one simulation for three time instances in case of maximum input demands. The left figure illustrates the applied control input sequence to be applied in order to avoid the collision. Since maximum control input are applied for the first time instance, the system is able to handle consecutive collision risks as illustrated in the middle figure. The maximum control input that can be applied are illustrated in the right figure.

3.5.3 Transfer delay

The server on which the CCAS is implemented is associated with a constant transfer delay, denoted D, between the controllable vehicles and the server. Its inclusion to the CCAS is seen in the model of the controllable vehicles, in turn shifting the equality constraints in Equation 2.10 as

$$\mathbf{X}(k+1|k) = \mathbf{A}\mathbf{X}(k|k) + \mathbf{B}\mathbf{U}(k-D|k) \tag{3.33}$$

with the appearance of the control input $\mathbf{U}(k-D|k)$ being a total of D steps behind its associated state $\mathbf{X}(k|k)$. Any control input that describe a time before the current time would have to already been applied and is therefore not included. This results in a D number of shifts down in the right half of the equality matrix $\mathbf{A}_{eq}^{i}$ corresponding to the columns that places the control inputs, but otherwise remained unchanged as seen in Equation 3.34 where a delay by one shifted the right side of the matrix down

$$\mathbf{A}_{eq}^{i} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & & \vdots & -\mathbf{B} & \mathbf{0} & & \mathbf{0} \\ \vdots & & \ddots & \vdots & -\mathbf{AB} & -\mathbf{B} & \ddots & \\ & & & \mathbf{0} & \vdots & \vdots & & \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{I} & -\mathbf{A}^{H_x-2}\mathbf{B} & -\mathbf{A}^{H_x-3}\mathbf{B} & \dots & \mathbf{0} \end{bmatrix} \tag{3.34}$$

The last D number of control inputs will not have time to effect the system which can be seen by the last D number of columns in $\mathbf{A}_{eq}^{i}$ being all zeros. These control inputs could be removed from the problem entirely. However, they are kept for a more generic implementation.

3.6 Prediction horizon

With the goal of presenting two different approaches to derive the necessary prediction horizon H_x, this section explains how a minimally intrusive prediction horizon is derived. In 3.6.1 all general and important aspects that must be clarified are introduced. In 3.6.2 an analytic problem statement and the necessary condition that has to be fulfilled in order for the prediction horizon to be sufficient are presented. Then to this extent, an ad hoc solution is presented in 3.6.3 that is implemented in SPAS and evaluated for a worst case scenario.

3.6.1 Important points

In order to come up with a value for H_x, it has to be assumed that the reachable sets of the controllable vehicles grows faster than the reachable sets of the uncontrollable vehicle. After solving for new trajectories for the controllable vehicles given an observed collision risk after H_x time steps, there is a risk of further observed collision risks after that point as the uncontrollable vehicle will take time to pass the intersection and may be moving towards one of the controllable vehicles. To handle the issue of consecutive collision risks, enough energy must have been defined to handle the largest amount of additional collision detections, denoted h. The times associated with the prediction horizon H_x and the number of additional collision detections are referred to as T_{H_x} and T_h respectively.

3.6.2 Analytic approach

The reachable sets $\mathcal{R}^c$ and $\mathcal{R}^{uc}$ of the controllable and uncontrollable vehicles respectively can be formulated as

$$\mathcal{R}^c_{H_x} = \left\{ \mathbf{X}(T_{H_x}) = \int_0^{T_{H_x}} f(\mathbf{X}(\tau), \mathbf{U}(\tau)) d\tau \mid \mathbf{X}(0) \in \mathcal{X}^c_0, \mathbf{U} \in \mathcal{U}^c \right\} \tag{3.35}$$

$$\mathcal{R}^{uc}_{H_x} = \left\{ \mathbf{X}(T_{H_x}) = \int_0^{T_{H_x}} f(\mathbf{X}(\tau), \mathbf{U}(\tau)) d\tau \mid \mathbf{X}(0) \in \mathcal{X}^{uc}_0, \mathbf{U} \in \mathcal{U}^{uc} \right\} \tag{3.36}$$

where the model f could be defined as the nonlinear BM f_{NL} presented in section 3.1.2. In that case, the initial set $\mathcal{X}^c_0, \mathcal{X}^{uc}_0 \subset \mathbb{R}^{6\times 1}$ and input set $\mathcal{U}^c$,$\mathcal{U}^{uc} \subset \mathbb{R}^{2\times 1}$. Here, the dimensions of the vehicle is included in the initial set $\mathcal{X}_0$.

From these sets, the subsets

$$r^c = \left\{ (X, Y) \mid X, Y \in \mathcal{R}^c \right\} \tag{3.37}$$

$$r^{uc} = \left\{ (X, Y) \mid X, Y \in \mathcal{R}^{uc} \right\} \tag{3.38}$$

are introduced, consisting of the global position of the sets $\mathcal{R}^c$ and $\mathcal{R}^{uc}$ respectively. Then, by assuming that r^c grows faster than r^{uc}, that is

$$||r^c_k||_2 - ||r^c_{k-1}||_2 > ||r^{uc}_k||_2 - ||r^{uc}_{k-1}||_2 \quad \forall k > 1 \tag{3.39}$$

it is possible to ensure that a collision always could be avoidable if the condition

$$r^c_{H_x+h} \setminus r^{uc}_{H_x} \notin \emptyset \tag{3.40}$$

is fulfilled, i.e. the reachable positions of the controllable vehicles at time step H_x+h must be outside the possible positions of the uncontrollable vehicle's reachable set at time step H_x.

3.6.3 Practical approach

In a practical setting, the reachable set of the uncontrollable vehicle is over approximated by use of the RA algorithm presented in section 3.2.4 producing the set $\mathcal{R}^{uc}$. This set is extended with the dimensions of the controllable vehicles as described in section 3.3. To describe the dynamics of the controllable vehicles the discrete version of the CV model is used.

From the given scenario the uncontrollable vehicle is assumed to move perpendicularly to the controllable vehicles, with their longitudinal direction aligned with the

global X-axis. This means that the width of the reachable set of the uncontrollable vehicle must be smaller than the change in the sets of the controllable vehicles. For the longitudinal control case there is no practical implication of having both vehicles in the scenario as they did not risk of intersecting with each other, as they are on completely separate lanes. The worst case is therefore focused on the controllable vehicle approaching from the left as the worst case could only come true for one of the controllable vehicles.

A worst case scenario is a placement of the controllable vehicles trajectory in reference to the reachable set of the uncontrollable vehicle that barely gives the controllable vehicle enough time to alter its future position to either side of the uncontrollable vehicles reachable set, after which the uncontrollable vehicle continues to steer towards the controllable vehicle. This results in the longest additional time T_h that the controllable vehicle must continue to move away as depicted in figure 3.8. As it is assumed that the controllable vehicle can accelerate faster than the uncontrollable vehicle, if it initially clear the reachable set of the uncontrollable vehicle in H_x steps, then it will be able to continue to do so and therefore do not need a further evaluation.

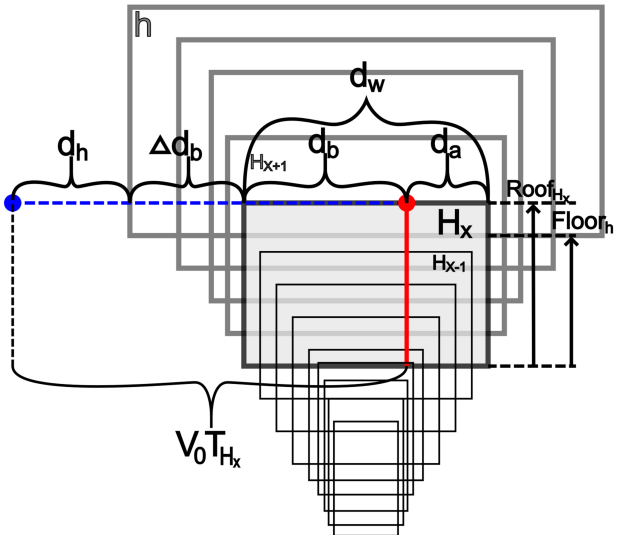


Figure 3.8: Worst case point of collision detection and needed alteration in order to place it outside the reachable set.

The distance $V_0T_{H_x}$ corresponds to the unaltered distance between the controllable vehicle and the point of collision detection given that the original trajectory have had no change in velocity for the upcoming H_x steps. The distance d_b is the needed minimum amount of change in the controllable vehicles future position to clear the reachable set of the uncontrollable vehicle in H_x steps while Δd_b is the additional distance that it in worst case must clear in the additional time T_h before the collision risk will have passed. The remaining distance that closes the gap d_h results in the longest distance it is allowed to reach from its initial position to clear the worst case. The difference in heights $Roof_{H_x}$ and $Floor_h$ indicates that a collision risk can still

be observed at proceeding time steps.

The formulation of change in distance is based on the discrete version of the CV model based on sampling time t, resulting in

$$V_0 T_{H_x} = d_h + d_b + \Delta d_h \tag{3.41}$$

$$d_h \leq V_0(T_{H_x} + T_h) + \frac{1}{2}\underline{a}(T_{H_x} + T_h)^2 \tag{3.42}$$

$$\Leftrightarrow d_b = V_0 T_{H_x} - (d_h + \Delta d_h) \geq -(V_0 T_h + \frac{1}{2}\underline{a}(T_{H_x} + T_h)^2 + \Delta d_h) \tag{3.43}$$

Inversely to the longest allowed distance d_h, d_b had a minimum distance to clear that must be at least as large as to cover the total width d_w of the uncontrollable vehicle's reachable set together with the longest reachable distance d_a through acceleration, that is

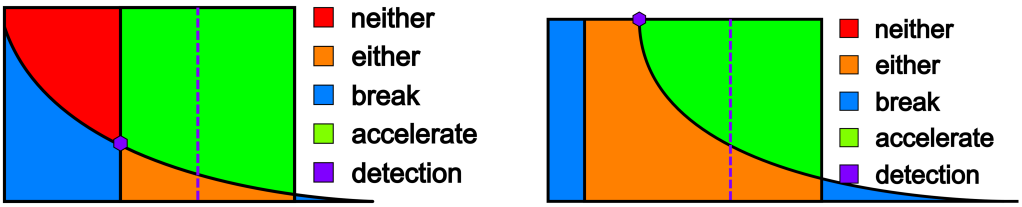
$$d_b \geq d_w - d_a \tag{3.44}$$

These equations are implemented in an algorithm which recursively solves for the needed prediction horizon H_x to clear an increasing number of waiting time steps h, stopping if the collision risk have passed or if the controllable vehicle have been able to completely break. In that case it is assumed in similar manner as with acceleration, that it will be able to continuously move away from the uncontrollable vehicle as it can decelerate more.

Output: H_x
$H_x = 1$
$h = 0$
while $Roof_{H_x} \geq Floor_h$ **and** $d_h \geq 0$ **do**
  while $d_b \leq d_w - d_a$ **do**
    $H_x = H_x + 1$
  end
  while $d_b \geq d_w - d_a$ **and** $Roof_{H_x} \geq Floor_h$ **and** $d_h \geq 0$ **do**
    $h = h + 1$
  end
end

Algorithm 2: Minimum prediction horizon algorithm showing the pseudo code for how the minimally intrusive prediction horizon H_x is derived.

In Algorithm 2, each incremented prediction horizon H_x or additional time T_h results in new distances according to equations (3.41) to (3.44). The algorithm is later penalized to achieve the same necessary prediction horizon, but with the added delay D as mentioned earlier. The practical approach is necessary as there are a multitude of variables and approximations used to formulate the reachable set of the uncontrollable vehicle. A visualization of the algorithm is shown in Figure 3.9 where the extended waiting time T_h is increased until the area "neither", which represents the position in the reachable set unable to avoid the worst case scenario, is eliminated.



(a) A few consecutive collision detections.

(b) Maximum consecutive collision detections.

Figure 3.9: Zones of a sets for feasible choices in case of a detected collision risk. Region "neither" representing unsolvable, "either" solvable by either breaking or accelerating, "break" solvable by only breaking, "accelerating" solvable by only accelerating and "detection" representing the observed point of the collision detection for a increasing number of collision detections h.

4

Results

This chapter present the main results of the implementation of the CCAS from the implementation described in Chapter 3. The results from the approach of using nonlinear and linear RA are presented in Section 4.1 and 4.2 respectively. The result from the full implementation are then presented in Section 4.3.

4.1 Nonlinear RA algorithm

The result from the nonlinear BM implementation in the nonlinear RA algorithm presented in Section 3.2 can be seen in Figure 4.1 together with the solution of solving the system of ODEs. The input set consists of the steering angles and accelerations that could be set by the driver. In this case, the steering angles used in the RA algorithm and the system solved by the ODE solver are set to $\pm 5^o$ and $\pm 45^o$ respectively and an acceleration of $-4m/s^2$ to $2m/s^2$. The result shows that the high nonlinearity of the model caused the algorithm to grind to a halt for any larger steering angles than $\pm 5^o$ degrees which is clearly just a portion of the maximum steering angle of $\pm 45^o$ that could be set by the driver.

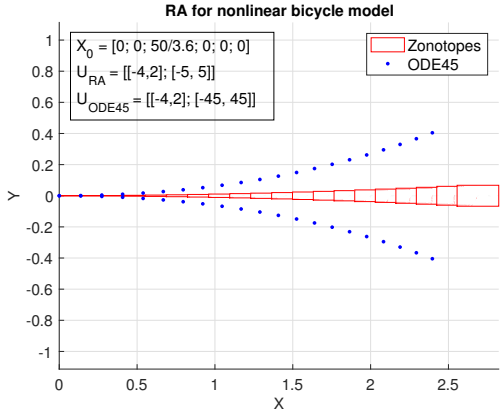


Figure 4.1: Outcome of the nonlinar RA algorithm for the critical steering angle $\delta = \pm 5^o$ compared to the solution from the ODE solver in Matlab for the desired steering angle $\delta = \pm 45^o$.

4.2 Linear RA algorithm

The result of Algorithm 1 acting on the linear BM f_t is presented in Figure 4.2. To evaluate the result of the algorithm, its outcome was compared to the solution of solving the system of ODEs, i.e. the model. Here, the steering angles and accelerations are set to $\pm 45^o$ and $-4m/s^2$ to $2m/s^2$ respectively. From the figure, it can be seen that the sets produced by the algorithm encloses the solution of the ODE solver which is to be expected due to the over approximation property of the RA.

It should be noted that the comparison between the outcome of the algorithm and the solutions achieved by the ODE solver assumes that the model is representative enough of the uncontrollable vehicle which only holds around the added assumptions for which it is made.

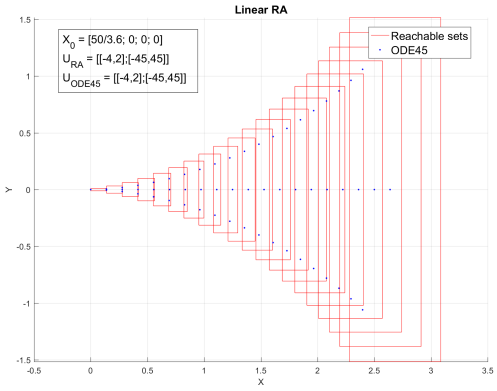


Figure 4.2: Outcome of the linear RA algorithm and ODE solver in Matlab for a steering angle of $\pm 45^o$ and allowed accelerations of $-4m/s^2$ to $2m/s^2$. It is clear that the RA over approximates the solution of solving the set of ODEs.

4.3 Full implementation

In order to evaluate the complete CCAS, the restrictions described in Section 4.3.1 was introduced. These restrictions made in possible to show the overall functionality of the implemented system as presented in Section 4.3.2. To validated the correctness of the derived length of the prediction horizon, the control inputs calculated by the MPC was investigated and are presented in Section 4.3.3.

4.3.1 Evaluation setup

The greatest challenge for the CCAS proved to be that the smallest prediction horizon H_x needed is already large given the limited control inputs of the controllable vehicles. This is because of the uncontrollable vehicle's ability to quickly cover a large area when it has a completely unknown behaviour. The needed assumption in Equation (3.39) therefore breaks if the uncontrollable vehicle is on par with the controllable vehicles in applicable control inputs.

To simply be able to move out of the minimum set consisting of both vehicles dimensions as seen in Figure 3.5, roughly 1.7 seconds would be needed given that the prediction horizon algorithm with no uncertainties of the uncontrollable vehicle, and control inputs for the controllable vehicle limited to $-4m/s^2$ to $2m/s^2$, and an initial velocity for both uncontrollable and controllable vehicles set to $50km/h$. This is an already large prediction horizon to apply RA to, as the uncontrollable vehicle would be able to cover a lot of ground in that time if it have large variation in control inputs, demanding an even larger prediction horizon to cope with it. This results in a CCAS that has to take over relatively early from the intersection.

The simulations in SPAS are therefore focused on moderate uncertainty of the uncontrollable vehicles behavior in order to acquire feasible solutions that shows the full functionality of the implementation. No uncertainty in acceleration of the uncontrollable vehicle are assumed, focusing on a minor uncertainty of 1 degree in steering angle. Though a small value, given the minimum prediction horizons length, any larger could be considered unreasonable assuming the uncontrollable vehicle can not drive of the road before reaching the intersection.

Three different commonly used speed limits are tested, 30, 50 and $70km/h$ shared by all vehicles in the scenario. The uncontrollable vehicle is set to move as close to the most challenging path it could take for the MPC to barley have enough time to move the controllable vehicles out of the way.

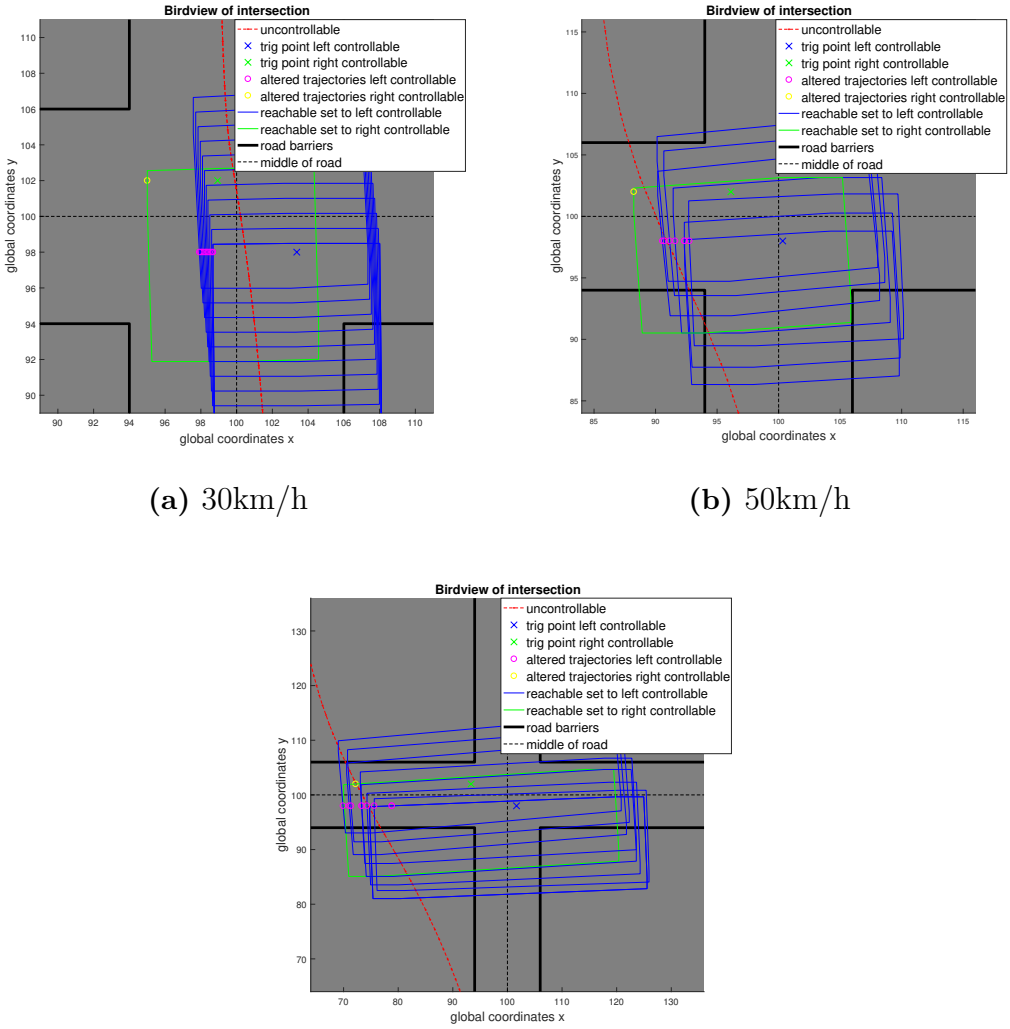
4.3.2 Evaluation of the Centralized Collision Avoidance System

The Figure 4.3 shows the point forward from when the CCAS is activated by a collision detection risk, each for the different initial velocities of 30, 50 and $70km/h$ respectively. The figures depicts a bird-eye view of the intersection, showing the reachable sets that must be averted for each controllable vehicle and their resulting altered trajectory. They show how the reachable sets rapidly expands for more challenging scenarios but also that the implementation successfully averting the collision risks. The left vehicle is facing the largest challenge as it had to continuously back up in order to stay out of each new reachable set. The vehicle coming from the right on the other hand was just close enough to accelerate past the reachable set and has thereafter safely moved past the intersection.

As the Figures 4.3a to 4.3c show, the MPC successfully calculates new trajectories that the remainder of SPAS follows correctly. After the pending collision passed, the controls are restored to the original setup in SPAS. The desired trajectories of the controllable vehicles are completely based on the predefined scenario, causing each new desired trajectory that they present to be unaffected by the alterations done by the MPC. As a result, the MPC has to hold on to the controls for an extended H_x steps in order to ensure that all its derived trajectory points are applied. After these extra sets of iterations the controls are returned.

4.3.3 Optimal control

To show that the applied input is used to its fullest, Figure 4.4 shows how it is not succeeded by a larger applied input for the remainder of the additionally observed collision risks. Therefore that input is used up to its fullest and therefore all formulated energy are applicable.

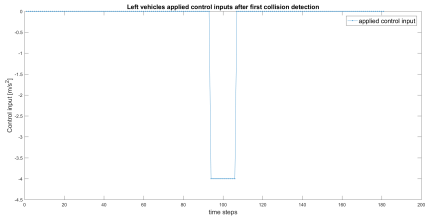


(a) 30km/h

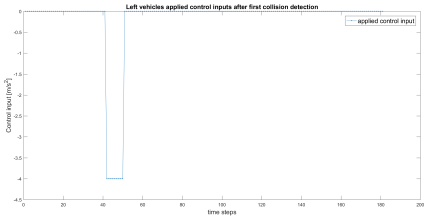
(b) 50km/h

(c) 70km/h

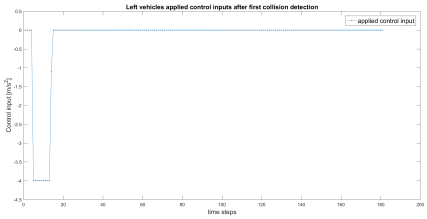
Figure 4.3: Triggered scenarios with vehicles for different initial velocities showing the propagation of the uncontrollable vehicles reachable sets with the last one being biased by the controllable vehicles dimensions. Presenting How detected collision risks are continuously kept outside the reachable sets with x marking the point of collision risk detection and rings shoving the allocation of those points.



(a) 30km/h



(b) 50km/h



(c) 70km/h

Figure 4.4: Applied control signals for the left incoming vehicle, for scenarios with different initial velocities. Showing over the time of the simulation how the applied control input is saturated during the time the CCAS takes over.

5

Conclusion

A successful CCAS was implemented into the SPAS environment. Able to formulate the non-convex sets from the RA to convex counterparts, select the most feasible way out of the predicted reachable set along the controllable vehicles longitudinal trajectory and shows enough energy to handle the formulated worst case amount of waiting time T_h. However, based on the results it was concluded that the implemented collision avoidance system only achieves feasible solutions when the uncertainty of the uncontrollable vehicle's behaviour is limited, due to the otherwise rapidly expanding sets of the RA. The attempt to use nonlinear RA in order to get narrower sets could not be utilized due to the high non-linearity of the model, completely halting the algorithm for any larger control inputs. Though a nonlinear RA would narrow the sets, they would still be expanding rapidly due to the large unknown behaviour of the uncontrollable vehicle. The described unknown behaviour of the uncontrollable vehicle should therefore be changed in order to allow for sufficiently small reachable sets.

The initial velocities of all vehicles in the defined scenario have a large effect on the CCAS as the reachable sets increases rapidly at larger velocities. Though solvable, the effect could be deemed intrusive to the ordinary controls of the controllable vehicles at larger velocities as the CCAS would be taking over increasingly often.

The minimum prediction horizon was derived successfully based on resulting simulations where the ad hoc solution determined the needed prediction horizon given the applied setup. The ad hoc solution is however specific to the defined scenario and to be used by other scenarios, a different formulation has to be made.

6

Future Work

The main result of this thesis have been that classical RA is an insufficient approach without a form of driver model for the uncontrollable vehicle. Stochastic RA would allow the prediction horizon to be shortened into a feasible time frame for the MPC to be capable of still avoiding the collision risks. This however breaks away from the thesis original problem formulation as collision avoidance would not be guarantee, but probable.

The reachable sets could be narrowed if nonlinear RA could be applied. Though to utilize nonlinear RA a less nonlinear model would likely be needed as the computation time proved very large for the used BM, alternatively these sets could possibly be computed offline and stored in memory. It is also possible that given a sufficient driver model then the reachable sets from nonlinear RA would never be that large, and therefore not as computationally demanding.

Including lateral control to the controllable vehicles would also shorten the needed prediction horizon as the MPC could move the trajectories in two dimensions, allowing to take the shortest path out of the set, not only along the controllable vehicles longitudinal trajectories. Because of the confining area of the road however, not all that much additional freedom can be added as the controllable vehicles would reasonably have to keep to the confines of the road. The computational burden of the MPC would also significantly increase as all positions of the controllable vehicles in reference to each other as well as to the borders of the intersection would have to be included as additional constraints.

A general explanation for the needed prediction horizon was presented and an ad hoc solution was implemented for the given scenario. For future work, refinement of the ad hoc solution to work for more general cases could be of interest. This could also be said about the existing general formulation as it does not include the limiting area of the intersection to limit the reachable sets.

Bibliography

[1] (2017, jan) Drive me – the self-driving car in action | volvo cars. [Online]. Available: http://www.volvocars.com/intl/about/our-innovation-brands/intellisafe/autonomous-driving/drive-me

[2] A. Effekta. (2016, dec) itransit - intelligent traffic management system based on its. [Online]. Available: http://www.vinnova.se/sv/Resultat/Projekt/Effekta/2009-02186/iTRANSIT---intelligent-TRAffic-maNagement-System-based-on-ITS/

[3] H. Karlsson, "Application within ffi - effektiva och uppkopplade transportsystem," mar 2015.

[4] Y. Yang, J. Zhang, K. q. Cai, and M. Prandini, "A stochastic reachability analysis approach to aircraft conflict detection and resolution," in *2014 IEEE Conference on Control Applications (CCA)*, Oct 2014, pp. 2089–2094.

[5] G. C. Nunes, "Design and analysis of multivariable predictive control applied to an oil-water-gas separator: A polynomial approach," Ph.D. dissertation, University of Florida, 2001.

[6] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technischen Universität München, 2010.

[7] D. Greene, J. Liu, J. Reich, Y. Hirokawa, A. Shinagawa, H. Ito, and T. Mikami, "An efficient computational architecture for a collision early-warning system for vehicles, pedestrians, and bicyclists," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 942–953, Dec 2011.

[8] O. S. Matthias Althoff and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," *47th IEEE Conference on Decision and Control*, 2008.

[9] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, Aug 2014.

[10] W. F. L. Felipe Kühne, João Manoel Gomes da Silva Jr., "Mobile robot trajectory tracking using model predictive control," *SBAI / II IEEE LARS*, 2005.

[11] Y. S. Rajashekar Chandru, "Motion planning for autonomous lane change manoeuvre with abort ability," Master's thesis, Chalmers University of Technology, SE-412 96 Gothenburg, 2016.

[12] R. Kianfar, B. Augusto, A. Ebadighajari, U. Hakeem, J. Nilsson, A. Raza, R. S. Tabar, N. V. Irukulapati, C. Englund, P. Falcone, S. Papanastasiou, L. Svensson, and H. Wymeersch, "Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 994–1007, Sept 2012.

[13] M. Kutil, "Modeling and optimization of traffic flow in urban areas," Ph.D. dissertation, Czech Technical University, jan 2010.

[14] M. M. S. José A. De Doná, Graham C. Goodwin, "Anti-windup and model predictive control: Reflections and connections," *European Journal of Control*, pp. 467–477, 2000.

[15] S. B. Yang Wang, "Fast model predictive control using online optimization," *IEEE Transaction on Control Systems Technology*, vol. VOL. 18, no. NO. 2, pp. 267–278, 2010.

[16] M. Althoff, "On computing the minkowski difference of zonotopes," *eprint arXiv:1512.02794*, 2015.

[17] G. A., "Reachability of uncertain linear systems using zonotopes," in *Morari M., Thiele L. (eds) Hybrid Systems: Computation and Control*. Springer, 2005, p. 302.

[18] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. Springer, 2009.

[19] R. Rajamani, *Vehicle Dynamics and Control*. Springer, 2006.

[20] J. R. Ida Petersson, "Automotive path following using model predictive control," Master's thesis, Chalmers University of Technology, 2014.

[21] M. Althoff. (2016). [Online]. Available: http://www6.in.tum.de/Main/SoftwareCORA

A

MPC

The full generic cost function is formulated as

$$
\begin{aligned}
J =& \frac{1}{2}\Bigg((\tilde{\mathbf{X}}^1(k+1|k))^T \mathbf{Q}^1(k+1|k)(\tilde{\mathbf{X}}^1(k+1|k)) + \\
& (\tilde{\mathbf{X}}^1(k+2|k))^T \mathbf{Q}^1(k+2|k)(\tilde{\mathbf{X}}^1(k+2|k)) + \dots \\
& + (\tilde{\mathbf{X}}^1(k+H_x|k))^T \mathbf{Q}^1(k+H_x|k)(\tilde{\mathbf{X}}^1(k+H_x|k)) + \\
& (\mathbf{U}^1(k|k))^T \mathbf{R}^1(k+1|k)(\mathbf{U}^1(k|k)) + \\
& (\mathbf{U}^1(k+1|k))^T \mathbf{R}^1(k+2|k)(\mathbf{U}^1(k+1|k)) + \dots \\
& + (\mathbf{U}^1(k+H_u|k))^T \mathbf{R}^1(k+H_u|k)(\mathbf{U}^1(k+H_u|k)) + \\
& (\tilde{\mathbf{X}}^i(k+1|k))^T \mathbf{Q}^i(k+1|k)(\tilde{\mathbf{X}}^i(k+1|k)) + \\
& (\tilde{\mathbf{X}}^i(k+2|k))^T \mathbf{Q}^i(k+2|k)(\tilde{\mathbf{X}}^i(k+2|k)) + \dots \\
& + (\tilde{\mathbf{X}}^i(k+H_x|k))^T \mathbf{Q}^i(k+H_x|k)(\tilde{\mathbf{X}}^i(k+H_x|k)) + \\
& (\mathbf{U}^i(k|k))^T \mathbf{R}^i(k+1|k)(\mathbf{U}^i(k|k)) + \\
& (\mathbf{U}^i(k+1|k))^T \mathbf{R}^i(k+2|k)(\mathbf{U}^i(k+1|k)) + \dots \\
& + (\mathbf{U}^i(k+H_u|k))^T \mathbf{R}^i(k+H_u|k)(\mathbf{U}^i(k+H_u|k)) \Bigg) \\
=& \frac{1}{2}\Bigg(\sum_{j=1}^{H_x} \Big((\tilde{\mathbf{X}}^1(k+j|k))^T \mathbf{Q}^1(k+j|k)(\tilde{\mathbf{X}}^1(k+j|k)) + \\
& (\tilde{\mathbf{X}}^2(k+j|k))^T \mathbf{Q}^2(k+j|k)(\tilde{\mathbf{X}}^2(k+j|k)) + \dots \\
& + (\tilde{\mathbf{X}}^i(k+j|k))^T \mathbf{Q}^i(k+j|k)(\tilde{\mathbf{X}}^i(k+j|k)) \Big) + \\
& \sum_{l=0}^{H_u} \Big((\mathbf{U}^1(k+l|k))^T \mathbf{R}^1(k+l|k)(\mathbf{U}^1(k+l|k)) + \\
& (\mathbf{U}^2(k+l|k))^T \mathbf{R}^2(k+l|k)(\mathbf{U}^2(k+l|k)) + \dots \\
& + (\mathbf{U}^i(k+l|k))^T \mathbf{R}^i(k+l|k)(\mathbf{U}^i(k+l|k)) \Big) \Bigg)
\end{aligned}
\tag{A.1}
$$

with its variables previously defined in chapter 2.