
CHALMERS
UNIVERSITY OF TECHNOLOGY


UNIVERSITY OF GOTHENBURG

MASTER'S THESIS

Robust Design and Analysis of Automotive Collision Avoidance Algorithms

ANDERS SJÖBERG

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2017

Thesis for the Degree of Master of Science

Robust Design and Analysis of Automotive Collision Avoidance Algorithms

Anders Sjöberg

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE – 412 96 Gothenburg, Sweden
Gothenburg, January 2017

Abstract

Automotive collision avoidance systems help the driver to avoid or mitigate a collision. The main objective of this project is to find a methodology to improve the performance of Volvo's automotive collision avoidance system by optimizing its configurable parameters. It is important that the parameter setting is chosen in such a way that the automotive collision avoidance system is not too sensitive to uncertainties. However, finding an optimal parameter setting is an overwhelmingly complex problem. Therefore, our approach is to make the problem tractable, by choosing specific and realistic uncertainties, defining performance, and choosing a fundamental algorithm that describes and mimics Volvo's automotive collision avoidance system. This approach preserves the foundation of the problem.

The idea behind the methodology that solves this tractable problem is to find, and exclude, all the parameter values that can cause undesired assistance intervention and, out of the remaining parameter values, find the ones that prevent collision in the best way. This is done under the condition that the chosen realistic uncertainties can occur. To evaluate a parameter setting, data simulation is used. Due to the complexity of the simulation, efficient optimization tools are not available. Therefore, we have created a surrogate model that mimics the behaviour of the simulation as closely as possible by using a response surface, in this case accomplished by a radial basis function interpolation. Through this surrogate model we have found a satisfying parameter setting to the tractable problem. The methodology has laid a promising foundation of finding the optimal parameter setting to Volvo's automotive collision avoidance system.

Keywords: Simulation-based optimization, response surface methodology, radial basis functions, multi-objective optimization, Pareto optimal solutions, trigger edge, tunable parameters, false intervention, robustness, positive and negative performance scenarios.

Acknowledgments

I would like to thank my supervisors Claes Olsson and Andreas Runhäll at Volvo and my supervisor Michael Patriksson at the Department of Mathematical Sciences at Chalmers University of Technology and the University of Gothenburg for their help and support through this project. I would also like to thank my examiner Ann-Brith Strömberg at the Department of Mathematical Sciences at Chalmers University of Technology and the University of Gothenburg for the help in finding this project.

Anders Sjöberg
Gothenburg, January 2017

Contents

Chapter 1

Introduction

1.1 Background

Volvo Car Corporation is a leading company in developing collision avoidance systems for passenger cars. Each new car model is equipped with high-tech devices combined with state-of-the-art automotive collision avoidance algorithms. The car itself provides safety by continuously monitoring the surroundings and using that information to avoid dangerous situations. The car automatically triggers an avoidance maneuver if a certain threat metric exceeds some predefined threshold values. However, it is important that the car does not take action when the driver has full control over the situation, because that can lead to dire consequences. The threshold values have during a long period of time been developed and tuned by experts and through extensive field collection. The aim of this project is to investigate a more mathematical approach of finding the threshold values. Moreover, the collected information from the surroundings contains noise and therefore the threshold values need to be such that the automotive collision avoidance system is not too sensitive to this noise.

1.2 Limitations

Volvo's automotive collision avoidance system is very extensive, mainly since it has to deal with a large number of different situations with potential threat that can occur in traffic. This makes the system difficult to process and analyze. Therefore we make the problem more tractable by replacing the automotive collision avoidance algorithm with an analytical counterpart. Moreover, a number of traffic scenarios are carefully chosen to reflect the fundamental behavior of a driver. Realistic uncertainties that can occur are included as well. From the tractable problem we are able to gain analytical results and a deeper understanding of the real problem. In this thesis we process and solve only the tractable problem. However, we develop some generalizations of the solution methodology to make it more applicable to Volvo's automotive collision avoidance system.

1.3 Outline

In Chapter 2 we describe how to make the problem more tractable; this is done in three parts. In the first part we describe a fundamental algorithm including all its tunable parameters, developed by Volvo, that mimics well Volvo's automotive collision avoidance system. In the second part we define a performance measure for the evaluation of parameter settings. In the third part we define all dominant uncertainties that are assumed to occur. In Chapter 3 we present the essential optimization theory that is needed to understand the methodology developed. Three areas are concerned: global, multi-objective and simulation-based optimization. Global optimization is about finding an optimal solution, multi-objective optimization is used if more than one optimization goal is considered, and simulation-based optimization is about optimizing results modeled by a simulation. In Chapter 4 we introduce the deeper but necessary theory of radial basis functions. In Chapter 5 we present our robust methodology, discuss our definition of a robust solution, and present the algorithms developed. In Chapter 6 we discuss the methodology and present conclusions about what new doors this work has opened and what future work may involve.

Chapter 2

The problem description

2.1 The original problem description

For about ten years Volvo Car Corporation have developed cars that actively help to reduce and prevent collisions by utilizing an automotive collision avoidance system. The concept is that the car constantly monitors various factors, such as distance and speed, of the objects in its surroundings and by using that information the car can help in averting potential threats if the driver does not seem to handle the situation appropriately. Figure 2.1 illustrates how the car collects information through the use of a sensor. There are several options for the car to avoid the danger, such as full braking or steering in the appropriate direction and, of course, a combination of these. In this thesis we only focus on full braking. The decision on whether the car should take action or not depends on whether certain threshold values, determined by tunable parameters, are exceeded. Moreover, these thresholds can be exceeded also when braking assistance is not desirable, then called *false intervention*. There is always some degree of noise from the sensors; since Volvo Car Corporation want reliable performance of their cars it is important that the automotive collision avoidance system is not too sensitive to these uncertainties. We can hence state the problem to tackle:

Definition 2.1.1 (Original problem)**.** Find a parameter setting in the automotive collision avoidance system which results in low collision speed and, at the same time, minimizes the risk for false intervention. The parameter setting has to be chosen in such a way that the automotive collision avoidance system is not too sensitive to uncertainties

The stated problem is a so-called robust optimization problem ([1]).

2.2 The approach of making the problem tractable

It is hard to get a grip on the problem described in Definition 2.1.1, since it seems overwhelmingly complex, due to endless variations of scenarios and countless numbers of rows of data code in the automotive collision avoidance system. Our approach is to make the problem tractable by concretizing it into a smaller problem but still preserve its foundation. If we can find a satisfying approach to solve the smaller problem the general idea of that approach is likely to be applicable to the original problem as well.

Figure 2.1: An illustration of how the car is collecting information such as distance and speed. Source [2].

To get a graspable overview of the problem we will use a fundamental algorithm of Volvo's automotive collision avoidance system. The fundamental algorithm, which is developed by Volvo, is a lot less complex—it includes fewer parameters and fewer expressions. However, the concept is still the same and it mimics well Volvo's automotive collision avoidance system, which means that it constitutes a good foundation. The fundamental algorithm is presented in Section 2.3.

We define performance (to be detailed in Section 2.4) through the selection of representative scenarios. Both *positive* and *negative performance scenarios* are selected. In the positive ones the car should prevent collision as well as possible, and in the negative ones the risk for false intervention should be as low as possible. To evaluate different parameter settings, the fundamental algorithm will be used to simulate the car's reaction for each parameter setting and scenario.

The final step to make the problem tractable is choosing the uncertainties considered to be the most dominant, as well as their range (detailed in Section 2.5). These uncertainties are the only ones that can arise in the performance scenarios.

We can now state the tractable problem:

Definition 2.2.1 (Tractable problem)**.** Find a parameter setting in the fundamental algorithm which results in low collision speed in the positive performance scenarios, and at the same time, minimizes the risk for false intervention in the negative performance scenarios. The parameter setting has to be chosen in such a way that the fundamental algorithm is not sensitive to uncertainties within the defined range.

2.3 The fundamental algorithm

Volvo provided a MATLAB-script where the fundamental algorithm is compiled. From this script we have derived all relevant equations in the fundamental algorithm; this gave an insight into their meaning as well as possible modifications of them. Our findings are presented below.

The sensors of the car that collect information from the surroundings can filter information with a frequency of 0.02 seconds. Therefore, it is convenient to use this frequency in the simulations.

The car that is our reference point, i.e., the car equipped with an automotive collision avoidance system, we call the *host car* and the car that is in view of the sensor of the host car we call the *target car*.

In each time step in a simulation the amount of longitudinal acceleration, denoted $a_{\text{req}}^{\text{long}}$, required by the host car in order to avoid a crash is computed. Since there are unpredictable factors in real life, such as the road condition and the temperature of the tires, it is necessary to have some longitudinal precaution margins to compensate for the uncertainties. Moreover, it takes some time to build up the pressure in the break system of the car to enable full braking. This time depends on the longitudinal acceleration of the host car. Together with the relative longitudinal velocity of the host and the target cars, this needed time determines the required increase of the longitudinal margins. This modified longitudinal distance, denoted x_{mod}, between the host car and the target car is defined as

$$x_{\text{mod}} := x_{\text{rel}} - x_{\text{margin}} + t_{\text{pressure}}(a_{\text{h}}^{\text{long}}) \cdot v_{\text{rel}}^{\text{long}}, \tag{2.1}$$

where $x_{\text{rel}} := x_{\text{tar}} - x_{\text{h}}$ is the relative longitudinal distance between the target and the host car, x_{margin} is the precaution margin, t_{pressure} is the time needed to build up the pressure to enable breaking, $a_{\text{h}}^{\text{long}}$ is the acceleration of the host car, and $v_{\text{rel}}^{\text{long}} := v_{\text{tar}}^{\text{long}} - v_{\text{h}}^{\text{long}}$ is the relative longitudinal velocity between the target and the host car. If the following inequality holds for all times $t \geq 0$, then no crash will occur:

$$\frac{a_{\text{tar}}^{\text{long}} \cdot t^2}{2} + v_{\text{tar}}^{\text{long}} \cdot t + x_{\text{mod}} \geq \frac{a_{\text{req}}^{\text{long}} \cdot t^2}{2} + v_{\text{h}}^{\text{long}} \cdot t, \quad \forall t \geq 0, \tag{2.2}$$

where $a_{\text{tar}}^{\text{long}}$ is the longitudinal acceleration of the target car, $v_{\text{tar}}^{\text{long}}$ is the longitudinal velocity of the target car, and $v_{\text{h}}^{\text{long}}$ is the longitudinal velocity of the host car. If $a_{\text{req}}^{\text{long}} = a_{\text{tar}}^{\text{long}}$ and the inequality (2.2) is satisfied, then it implies that $v_{\text{tar}}^{\text{long}} \geq v_{\text{h}}^{\text{long}}$. However, if we assume that $a_{\text{req}}^{\text{long}} \neq a_{\text{tar}}^{\text{long}}$, then we can conclude that $a_{\text{tar}}^{\text{long}} > a_{\text{req}}^{\text{long}}$ whenever the inequality (2.2) is fulfilled. In that case inequality (2.2) can be rewritten as

$$t^2 + 2\left(\frac{v_{\text{tar}}^{\text{long}} - v_{\text{h}}^{\text{long}}}{a_{\text{tar}}^{\text{long}} - a_{\text{req}}^{\text{long}}}\right) \cdot t + \frac{2 \cdot x_{\text{mod}}}{a_{\text{tar}}^{\text{long}} - a_{\text{req}}^{\text{long}}} \geq 0, \quad \forall t \geq 0. \tag{2.3}$$

Now we search for the roots for the polynomial of the second degree in the left-hand side of the inequality (2.3) by completing the square, which yields the equation

$$\left(t + \frac{v_{\text{tar}}^{\text{long}} - v_{\text{h}}^{\text{long}}}{a_{\text{tar}}^{\text{long}} - a_{\text{req}}^{\text{long}}}\right)^2 = \left(\frac{v_{\text{tar}}^{\text{long}} - v_{\text{h}}^{\text{long}}}{a_{\text{tar}}^{\text{long}} - a_{\text{req}}^{\text{long}}}\right)^2 - \frac{2 \cdot x_{\text{mod}}}{a_{\text{tar}}^{\text{long}} - a_{\text{req}}^{\text{long}}}. \tag{2.4}$$

If the right-hand side of equation (2.4) is less than or equal to zero we have that inequality (2.2) is fulfilled, i.e.,

$$\left(\frac{v_{\text{tar}}^{\text{long}} - v_{\text{h}}^{\text{long}}}{a_{\text{tar}}^{\text{long}} - a_{\text{req}}^{\text{long}}}\right)^2 - \frac{2 \cdot x_{\text{mod}}}{a_{\text{tar}}^{\text{long}} - a_{\text{req}}^{\text{long}}} \leq 0 \Leftrightarrow$$
$$\underbrace{\frac{1}{a_{\text{tar}}^{\text{long}} - a_{\text{req}}^{\text{long}}}}_{>0} \left(\frac{(v_{\text{tar}}^{\text{long}} - v_{\text{h}}^{\text{long}})^2}{a_{\text{tar}}^{\text{long}} - a_{\text{req}}^{\text{long}}} - 2 \cdot x_{\text{mod}}\right) \leq 0.$$

Assuming that $x_{mod} > 0$, which reflects the relevant situations, it follows that

$$a_{\text{req}}^{\text{long}} \leq a_{\text{tar}}^{\text{long}} - \frac{\left(v_{\text{tar}}^{\text{long}} - v_{\text{h}}^{\text{long}}\right)^2}{2 \cdot x_{\text{mod}}}.$$

We want $a_{\text{req}}^{\text{long}}$ to be as high as possible, which reflects how $a_{\text{req}}^{\text{long}}$ is computed in the fundamental algorithm, and which is a sufficiently good way for all the scenarios in this thesis, i.e.,

$$a_{\text{req}}^{\text{long}} := a_{\text{tar}}^{\text{long}} - \frac{\left(v_{\text{rel}}^{\text{long}}\right)^2}{2 \cdot x_{\text{mod}}}. \tag{2.5}$$

The relation (2.5) is used for any real values on $a_{\text{tar}}^{\text{long}}$, $v_{\text{rel}}^{\text{long}}$ and x_{mod}. However, for the case when $x_{\text{mod}} = 0$ we use the following natural limits:

1. If $v_{\text{rel}}^{\text{long}} \neq 0$ and $x_{\text{mod}} = 0$, we set $a_{\text{req}} := -\infty$. We handle $-\infty$ as in the extended real number system; see [3].
2. If $v_{\text{rel}}^{\text{long}} = x_{\text{mod}} = 0$, we set $a_{\text{req}}^{\text{long}} := a_{\text{tar}}^{\text{long}}$.

Now we present the first tunable parameter, the *maximum available longitudinal acceleration*, denoted $a_{\text{avail}}^{\text{long}}(x_{\text{rel}})$, which is dependent on the relative longitudinal distance, i.e, x_{rel}, between the target car and the host car. The boundaries on this tunable parameter, derived from realistic usage, are $-10 \leq a_{\text{avail}}^{\text{long}} \leq -1$. Before stating the first threshold we need to define the *braking threat number*, denoted T_{BTN}, as

$$T_{\text{BTN}} := \frac{a_{\text{req}}^{\text{long}}}{a_{\text{avail}}^{\text{long}}(x_{\text{rel}})}.$$

In each time step, T_{BTN} is computed and if $T_{\text{BTN}} > 1$ a threshold value is exceeded and we say that the *BTN-condition* is true. Once the BTN-condition is true it remains true until the host car has passed the target car.

In each time step the required amount of lateral acceleration, denoted $a_{\text{req}}^{\text{lat}}$, to avoid a crash, i.e. to steer aside, is computed. To be able to compute $a_{\text{req}}^{\text{lat}}$ in each time step we need a prediction of the time until the relative longitudinal distance is equal to zero. We call it *time-to-collision*, denoted t_{ttc}. The computation of the time-to-collision depends on the relative longitudinal acceleration, $a_{\text{rel}}^{\text{long}} := a_{\text{tar}}^{\text{long}} - a_{\text{h}}^{\text{long}}$, and the relative longitudinal velocity, $v_{\text{rel}}^{\text{long}}$; we distinguish this between three different cases:

1. If $|a_{\text{rel}}^{\text{long}}| > 0$, proceed from (2.2) with some minor changes: change x_{mod} to x_{rel} and change the inequality to an equality. Once again we want to find the roots and we find them to be

$$t = -\frac{v_{\text{rel}}^{\text{long}}}{a_{\text{rel}}^{\text{long}}} \pm \sqrt{\left(\frac{v_{\text{rel}}^{\text{long}}}{a_{\text{rel}}^{\text{long}}}\right)^2 - \frac{2 \cdot x_{\text{rel}}}{a_{\text{rel}}^{\text{long}}}}. \tag{2.6}$$

   The time-to-collision is defined as the smallest positive root. However, if none of the roots are positive then no collision will occur and we have that $t_{\text{ttc}} := +\infty$.

2. If $a_{\text{rel}}^{\text{long}} = 0$ and $v_{\text{rel}}^{\text{long}} < 0$, the host car has a higher longitudinal velocity than the target car, in which it holds that

$$t_{\text{ttc}} := -\frac{x_{\text{rel}}}{v_{\text{rel}}^{\text{long}}}. \tag{2.7}$$

3. Otherwise $t_{\text{ttc}} := +\infty$.

We can use t_{ttc} to make a prediction on the relative lateral position of the cars when the relative longitudinal distance is zero. This is denoted $y_{\text{rel}}^{\text{pred}}$ and is calculated as:

$$y_{\text{rel}}^{\text{pred}} = y_{\text{rel}} + v_{\text{rel}}^{\text{lat}} \cdot t_{\text{ttc}} + \frac{a_{\text{tar}}^{\text{lat}} \cdot t_{\text{ttc}}^2}{2}, \tag{2.8}$$

where $y_{\text{rel}} := y_{\text{tar}} - y_{\text{h}}$ is the relative lateral distance, $v_{\text{rel}}^{\text{lat}} := v_{\text{tar}}^{\text{lat}} - v_{\text{h}}^{\text{lat}}$ is the relative lateral velocity and $a_{\text{tar}}^{\text{lat}}$ is the lateral acceleration of the target car. Note that y_{tar} and y_{h} are the lateral positions of the target car and host car, respectively, and $v_{\text{tar}}^{\text{lat}}$ and $v_{\text{h}}^{\text{lat}}$ are the corresponding respective lateral velocities. Note that the lateral acceleration of the host car is not included in (2.8), since we want to compute the total required lateral acceleration of the host, regardless of the current lateral acceleration.

Now we introduce the second tunable parameter, called *safety zone*, denoted $y_{\text{safe}}(v_{\text{h}}^{\text{long}})$, which is the lateral margin depending on the velocity of the host car. However, safety zone may be a misleading name, since the boundaries of y_{safe} is given by $-1 \leq y_{\text{safe}} \leq 0$. The reason why the safety zone can take negative values is that it may be favorable to "shrink" the width of the target car in order to avoid false intervention while compensating for sensor noise. Figure 2.2 shows an overview of the orientation of the coordinate system and the safety zone representation.

Now we present the last components, the accelerations of steering right or left, needed to compute $a_{\text{req}}^{\text{lat}}$. We have the following relations:

$$a_{\text{req}}^{\text{left}} = \frac{2y_{\text{rel}}^{\text{pred}} - w_{\text{tar}} - w_{\text{h}} - 2y_{\text{safe}}}{t_{\text{ttc}}^2}, \tag{2.9}$$

$$a_{\text{req}}^{\text{right}} = \frac{2y_{\text{rel}}^{\text{pred}} + w_{\text{tar}} + w_{\text{h}} + 2y_{\text{safe}}}{t_{\text{ttc}}^2}, \tag{2.10}$$

where w_{tar} and w_{h} are the widths of the target car and the host car, respectively. There are two possible outcomes:

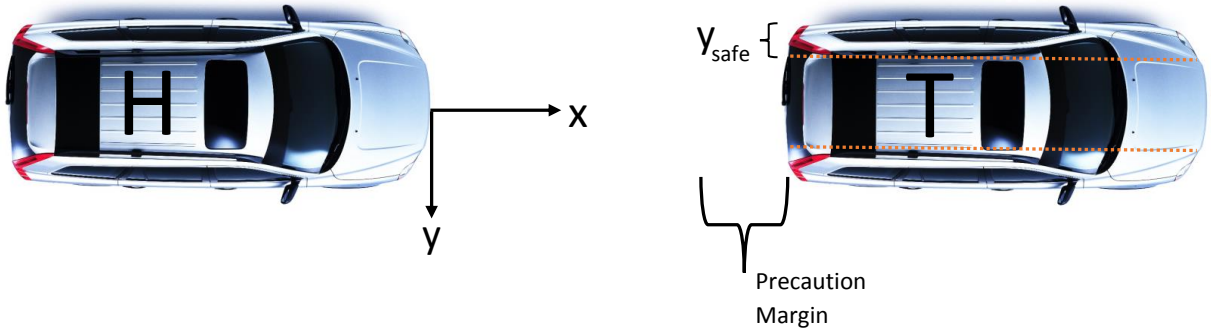

Figure 2.2: An overview of the coordinate orientation and the tunable parameter safety zone.

1. If $\text{sign}(a_{\text{req}}^{\text{left}}) = \text{sign}(a_{\text{req}}^{\text{right}})$, then the target car is not in the path of the host car, so it holds that $a_{\text{req}}^{\text{lat}} := 0$.
2. Otherwise

$$a_{\text{req}}^{\text{lat}} = \min\left\{\left|a_{\text{req}}^{\text{left}}\right|, \left|a_{\text{req}}^{\text{right}}\right|\right\}. \tag{2.11}$$

Now we present the last tunable parameter called *maximum available lateral acceleration*, denoted $a_{\text{avail}}^{\text{lat}}(v_{\text{h}}^{\text{long}})$, which depends on the velocity of the host car. The boundaries on this tunable parameter, derived from realistic usage are given by the inequalities $1 \leq a_{\text{avail}}^{\text{lat}} \leq 10$. The *steering threat number*, denoted T_{STN}, is then defined as

$$T_{\text{STN}} := \frac{a_{\text{req}}^{\text{lat}}}{a_{\text{avail}}^{\text{lat}}(v_{\text{h}}^{\text{long}})}.$$

In each time step T_{STN} is computed, and if $T_{\text{STN}} > 1$ then a threshold value is exceeded and we say that the *STN-condition* is true. Automatic full braking is applied whenever both the STN- and the BTN-conditions are true. Table 2.1 summarizes the tunable parameters.

Table 2.1: A compilation of the tunable parameters.

Tunable parameters	Notation
Maximum available longitudinal acceleration	$a_{\text{avail}}^{\text{long}}(x_{\text{rel}})$
Maximum available lateral acceleration	$a_{\text{avail}}^{\text{lat}}(v_{\text{h}}^{\text{long}})$
Safety zone	$y_{\text{safe}}(v_{\text{h}}^{\text{long}})$

The fundamental algorithm also includes computations, such as filtering of sensor information. Since none of these computations concern any of the tunable parameters, there is no need to describe them in detail.

2.4 Defining performance scenarios

Volvo has a number of cars around the world that constantly collect data from situations on the road. The variation of scenarios that can occur is almost endless, but we have identified five fundamental scenarios—see Figure 2.3—which capture the trade-off between minimizing the risk for false intervention and avoiding collisions as well as possible. We categorize these five scenarios into two groups, *positive performance scenarios*, and *negative performance scenarios*. In the negative performance scenarios no breaking is desired. In the positive performance scenarios the host car is going to collide unless it breaks sufficiently, so an as high velocity reduction as possible is desired. We define the term *offset* which describes how the target car is positioned relative to the host car when the relative longitudinal distance is zero. If the offset is 0% then the target car is not in the path of the host car; if the offset is 100% the target car is completely in the front of the host car.

1- The first negative performance scenario is defined by the host car driving straight with a certain velocity with 0% offset and the target car being stationary.

2- The second negative performance scenario is defined by the host car driving with a certain velocity and turning tightly past the stationary target car. In this scenario the car steers with a certain lateral acceleration, which is dependent on the velocity of the host car.

1+ The first positive performance scenario is defined by the host car driving straight with a certain velocity with 100% offset and the target car being stationary.

2+ The second positive performance scenario is defined by the host car driving straight with a certain velocity with 50% offset and the target car being stationary.

3+ The third positive performance scenario is defined by the host car and the target are driving straight with same velocity, with 100% offset, and the target car immediately starts to fully break.

In all the positive scenarios the host car is able to turn either right or left in order to avoid a collision.

2.5 Uncertainties

The dominant uncertainties assumed to occur are uncertainties from the sensor, since there is almost always some degree of noise. Table 2.2 lists the errors considered. Furthermore, we assume that the range of each error is given and all errors are independent. We make this assumption because the distribution of sensor errors are out of the scope of this thesis. However, Volvo have good knowledge of the spread of the errors.

We collect all the errors in Table 2.2 in a vector $\boldsymbol{\xi} = (\xi_{x_{rel}}, \xi_{v_{rel}^{long}}, \xi_{a_{tar}^{long}}, \xi_{y_{rel}}, \xi_{v_{rel}^{lat}}, \xi_{a_{tar}^{lat}}, \xi_{w_{tar}})^T$. We let b_i be the assumed range for error ξ_i for $i = 1, \ldots, 7$, so $-b_i \leq \xi_i \leq b_i$.

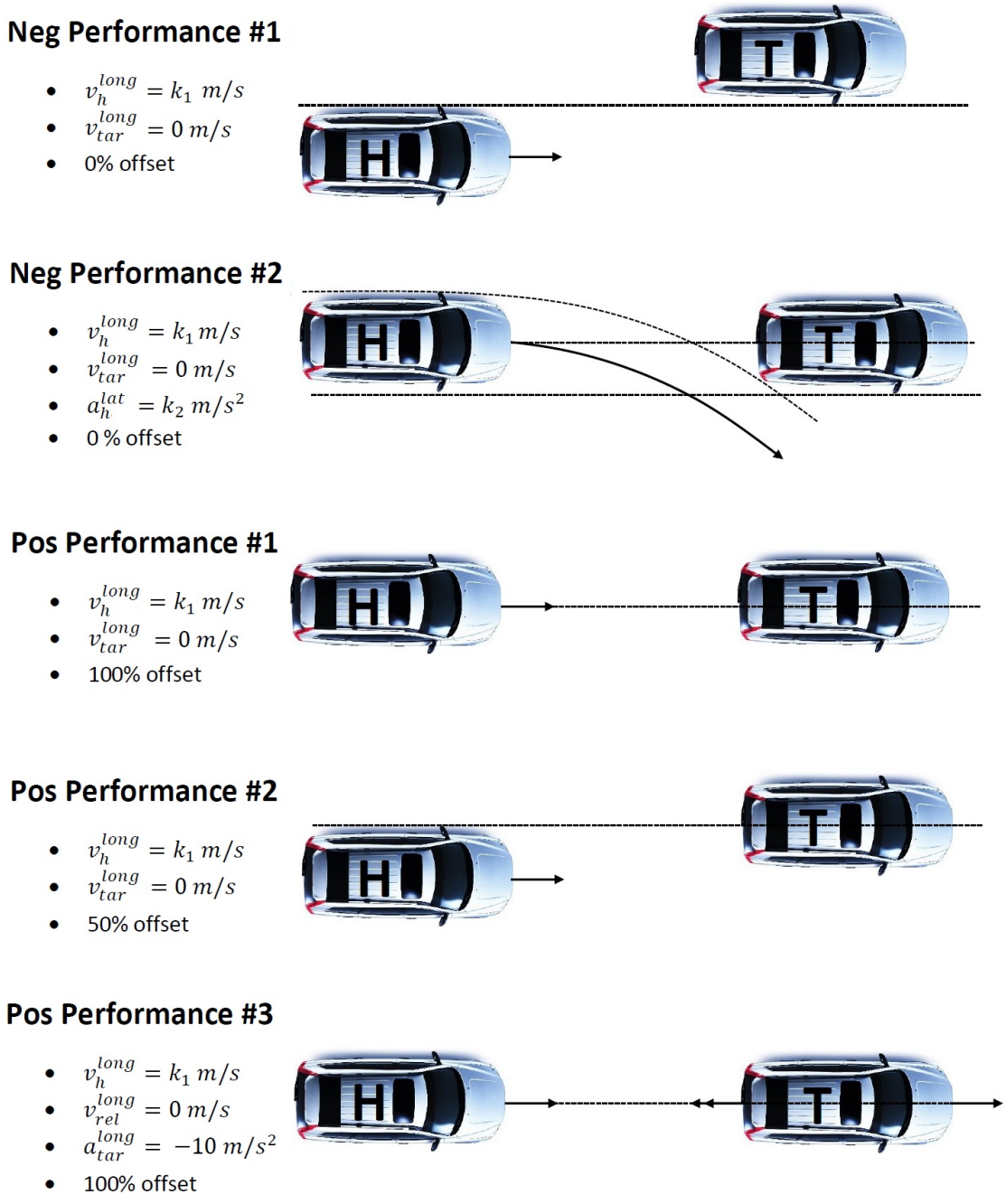

Figure 2.3: An overview of the negative and the positive performance scenarios.

Table 2.2: All assumed errors from the sensor.

Sensor uncertainties	Notation of Error
Relative longitudinal distance	$\xi_{x_{\text{rel}}}$
Relative longitudinal velocity	$\xi_{v_{\text{rel}}^{\text{long}}}$
Longitudinal acceleration of the target car	$\xi_{a_{\text{tar}}^{\text{long}}}$
Relative lateral distance	$\xi_{y_{\text{rel}}}$
Relative lateral velocity	$\xi_{v_{\text{rel}}^{\text{lat}}}$
Lateral acceleration of the target car	$\xi_{a_{\text{tar}}^{\text{lat}}}$
Width of the target car	$\xi_{w_{\text{tar}}}$

Chapter 3

Optimization background

In this chapter we present an introduction of the scientific areas concerned in this thesis. We start with essential theory and terminology in global optimization—see Section 3.1—which lays the theoretical foundation for this thesis. After that we present multi-objective optimization, the theory concerning more than one objective function—see Section 3.2. We conclude this chapter with a description of simulation-based optimization—see Section 3.3.

3.1 Global optimization

All the definitions and theorems in this section are taken from [4].

Consider the problem to

$$\begin{aligned} &\text{minimize } f(\boldsymbol{x}), \\ &\text{subject to } \boldsymbol{x} \in \Omega, \end{aligned} \tag{3.1}$$

where $\boldsymbol{x}$ is the decision variable, $\Omega \subseteq \mathbb{R}^d$ is a nonempty set and $f : \mathbb{R}^d \to \mathbb{R}$ is a given function.

Definition 3.1.1 (Global minimum)**.** Consider the optimization problem (3.1) and let $\boldsymbol{x}^* \in \Omega$. We say that $\boldsymbol{x}^*$ is a *global minimum* of f over Ω if f attains its lowest value over Ω at $\boldsymbol{x}^*$.

In other words $\boldsymbol{x}^* \in \Omega$ is a global minimum of f over Ω if

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega,$$

holds. □

The goal of the optimization problem (3.1) is to find an optimal solution, i.e., a global minimum, $\boldsymbol{x}^* \in \Omega$, of the objective function f over the feasible set Ω. The field regarding the search for a global minimum is called *global optimization*; see [5] for a more comprehensive introduction. Note that if the function is to be maximized it is equivalent to minimize $-f$.

However, there is another type of minimum that we also present, namely the *local minimum*. Let $B_\varepsilon(\boldsymbol{x}^*) := \{\boldsymbol{y} \in \mathbb{R}^d : \|\boldsymbol{y} - \boldsymbol{x}^*\| < \varepsilon\}$ be the Euclidean ball centered at $\boldsymbol{x}^*$ with radius ε.

Definition 3.1.2 (Local minimum)**.** Consider the problem (3.1) and let $\boldsymbol{x}^* \in \Omega$.

a) We say that $\boldsymbol{x}^*$ is a *local minimum* of f over Ω if there exists a small enough Euclidean ball intersected with Ω around $\boldsymbol{x}^*$ such that $\boldsymbol{x}^*$ is a global optimal solution in that smaller set.

In other words, $\boldsymbol{x}^* \in \Omega$ is a local minimum of f over Ω if

$$\exists \varepsilon > 0 \text{ such that } f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega \cap B_\varepsilon(\boldsymbol{x}^*), \tag{3.2}$$

b) We say that $\boldsymbol{x}^* \in \Omega$ is a *strict local minimum* of f over Ω if the inequality in (3.2) holds strictly for $\boldsymbol{x} \neq \boldsymbol{x}^*$. □

Now we define some desirable properties of the feasible set Ω and the objective function f.

Definition 3.1.3 (Convex set)**.** Let $\Omega \subseteq \mathbb{R}^d$. The set Ω is convex if

$$\left.\begin{array}{r} \boldsymbol{x}_1, \boldsymbol{x}_2 \in \Omega \\ \lambda \in (0,1) \end{array}\right\} \implies \lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \in \Omega$$

holds. □

Definition 3.1.4 (Convex function)**.** Assume that $\Omega \subseteq \mathbb{R}^d$. A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex at $\hat{\boldsymbol{x}} \in \Omega$ if

$$\left.\begin{array}{r} \boldsymbol{x} \in \Omega \\ \lambda \in (0,1) \\ \lambda\hat{\boldsymbol{x}} + (1-\lambda)\boldsymbol{x} \in \Omega \end{array}\right\} \implies f(\lambda\hat{\boldsymbol{x}} + (1-\lambda)\boldsymbol{x}) \leq \lambda f(\hat{\boldsymbol{x}}) + (1-\lambda) f(\boldsymbol{x}).$$

The function f is convex on Ω if it is convex at every $\hat{\boldsymbol{x}} \in \Omega$. □

Assuming that the objective function f and the set Ω are both convex, the following property regarding local and global minimum can be established.

Theorem 3.1.5 (Fundamental Theorem of global optimality)**.** *Consider the problem (3.1), where Ω is a convex set and f is convex on Ω. Then every local minimum of f over Ω is also a global minimum.*

Proof. Assume that $\boldsymbol{x}^*$ is a local minimum but not a global one. Then consider a point $\bar{\boldsymbol{x}} \in \Omega$ with property that $f(\bar{\boldsymbol{x}}) < f(\boldsymbol{x}^*)$. Let $\lambda \in (0,1)$. By the convexity of the set Ω and the function f, $\lambda\bar{\boldsymbol{x}} + (1-\lambda)\boldsymbol{x}^* \in \Omega$, and $f(\lambda\bar{\boldsymbol{x}} + (1-\lambda)\boldsymbol{x}^*) \leq \lambda f(\bar{\boldsymbol{x}}) + (1-\lambda) f(\boldsymbol{x}^*)$. By choosing $\lambda > 0$ small enough it leads to contradiction to the local optimality of $\boldsymbol{x}^*$. □

This means that if an optimization problem fulfills the convexity conditions it is sufficient to apply a local optimization algorithm to find the global minimum. This is desirable since, in general, local optimization algorithms have a low computational complexity.

If the objective function f and the set Ω are both convex, then the following theorem provides a tool to verify if a point $\boldsymbol{x} \in \Omega$ is a global minimum of f over Ω.

Theorem 3.1.6. *Assume that $\Omega \subseteq \mathbb{R}^d$ is nonempty and convex. Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and C^1 on Ω. Then,*

$$\nabla f(\boldsymbol{x}^*)^T(\boldsymbol{x} - \boldsymbol{x}^*) \geq 0 \implies \boldsymbol{x}^* \text{ is a global minimum of } f \text{ over } \Omega. \tag{3.3}$$

Proof. Take $\boldsymbol{x}^*, \boldsymbol{x} \in \Omega$ and $\lambda \in (0,1)$. Then,

$$\begin{aligned} &\lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{x}^*) \geq f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{x}^*) \iff \\ &f(\boldsymbol{x}) - f(\boldsymbol{x}^*) \geq (1/\lambda)[f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{x}^*) - f(\boldsymbol{x}^*)]. \end{aligned} \tag{3.4}$$

Let $\lambda \to 0$. Then, the right-hand side of the inequality (3.4) tends to the directional derivative of f at $\boldsymbol{x}^*$ in the direction of $(\boldsymbol{x} - \boldsymbol{x}^*)$, so that in the limit it becomes

$$\begin{aligned} &f(\boldsymbol{x}) - f(\boldsymbol{x}^*) \geq \nabla f(\boldsymbol{x}^*)^T (\boldsymbol{x} - \boldsymbol{x}^*) \implies \\ &f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*) + \nabla f(\boldsymbol{x}^*)^T (\boldsymbol{x} - \boldsymbol{x}^*) \geq f(\boldsymbol{x}^*). \end{aligned}$$

□

Typically the feasible set Ω is determined by inequality and/or equality constraints. If that is the case, the optimization problem (3.1) can be expressed as

$$\begin{aligned} &\text{minimize } f(\boldsymbol{x}), \\ &\text{subject to } g_i(\boldsymbol{x}) \leq 0,\ i \in \mathcal{I}, \quad \text{(inequality constraints)} \\ &\qquad\quad g_i(\boldsymbol{x}) = 0,\ i \in \mathcal{E}, \quad \text{(equality constraints)} \\ &\qquad\quad \boldsymbol{x} \in \mathbb{R}^d, \end{aligned} \tag{3.5}$$

where $g_i(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ define the constraint functions, and $\mathcal{I}$ and $\mathcal{E}$ are finite index sets. If, in addition, the functions f and g_i are continuous the problem (3.5) is called a continuous optimization problem.

Proposition 3.1.7 (Convex intersection)**.** *Assume that Ω_k, $k \in \mathcal{K}$, is any collection of convex sets. Then the intersection*

$$\Omega := \bigcap_{k \in \mathcal{K}} \Omega_k$$

is a convex set.

Proof. Let both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ belong to Ω. (If two such points cannot be found then the results holds vacuously). Then, $\boldsymbol{x}_1 \in \Omega_k$ and $\boldsymbol{x}_2 \in \Omega_k$ for all $k \in \mathcal{K}$. Take $\lambda \in (0,1)$. Then, $\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \in \Omega_k, k \in \mathcal{K}$, by the convexity of the sets Ω_k. So, $\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \in \bigcap_{k \in \mathcal{K}} \Omega_k = \Omega$. □

If the objective function f is convex, the functions g_i, $i \in \mathcal{I}$, are convex and g_i, $i \in \mathcal{E}$, are affine, then the problem (3.5) is called a *convex problem*. From Proposition 3.1.7 it follows that the constraints in a convex problem form a convex set and thereby Theorem 3.1.5 can be applied.

In nonconvex optimization problems we have to expect multiple local minima, and the objective function value in some local minima can be far from the minimum value. These problems can be extremely difficult to solve. For a general nonconvex global optimization problem, where the evaluation of the objective function is sufficiently time efficient, we can apply algorithms that vary between a local and a global phase. During the global phase the idea of the algorithm is to explore roughly the whole feasible set while during the local phase it is restricted to explore in a local portion of the feasible set. The intention with the local phase is to refine the currently best solution found.

3.2 Multi-objective optimization

Often more than one optimization goal is desired, for instance in product and process design. A good design usually involves multiple criteria such as capital investment, profit, quality and/or lifespan, efficiency, process safety, operation time, and so on. We want to optimize all these multiple criteria and this section describes the mathematics needed to analyze such problem settings. All the theory in this section is taken from [6].

Consider the problem to

$$\begin{aligned} &\text{minimize} \ \ \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})\}, \\ &\text{subject to} \ \ \boldsymbol{x} \in \Omega, \end{aligned} \tag{3.6}$$

with n (≥ 2) objective functions $f_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, \ldots, n$, and $\Omega \subseteq \mathbb{R}^d$ denoting the feasible set. The problem (3.6) is a so-called *multi-objective optimization problem*. The objective functions f_i are likely to be in conflict, which means that there does not exist a single solution $\boldsymbol{x} \in \Omega$ that is optimal for all n objective functions. However, if the objective functions f_i are in conflict in the problem (3.6) then the problem is actually not well-defined, because there is no hierarchy between the functions. We have to present a definition of optimality for multi-objective optimization problems. Let $Z := \{\boldsymbol{z} \in \mathbb{R}^n : z_i = f_i(\boldsymbol{x}), \text{ for all } i = 1, \ldots, n \text{ and } \boldsymbol{x} \in \Omega\}$. We say that a point $\boldsymbol{x} \in \Omega$ is a *decision point* and that a point $\boldsymbol{z} \in Z$ is an *objective point*.

Definition 3.2.1 (Pareto optimality)**.** A decision point $\boldsymbol{x}^* \in \Omega$ is *Pareto optimal* if there does not exist another decision point $\boldsymbol{x} \in \Omega$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, n$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one index j.

An objective point $\boldsymbol{z}^* \in Z$ is Pareto optimal if there does not exist another objective point $\boldsymbol{z} \in Z$ such that $z_i \leq z_i^*$ for all $i = 1, \ldots, n$ and $z_j < z_j^*$ for at least one index j; or equivalently, $\boldsymbol{z}^*$ is Pareto optimal if the decision point corresponding to it is Pareto optimal. □

Note that there may be an infinite number of Pareto optimal points. The set of Pareto optimal objective points $Z^* \subseteq Z$ is called the *Pareto optimal set*. Figure 3.1 illustrates the variable space and the objective space.

From a mathematical point of view every Pareto optimal point is an equally acceptable solution to the multi-objective optimization problem (3.6). However, in general only one point is desired as a solution. Therefore, we need a so-called *decision maker* to select one solution out of the set of Pareto optimal solutions, since the information that is needed to make the selection is not contained in the objective functions. The decision maker is a person, or a group of persons, who has better insight into the problem and who can formulate preferences among the Pareto optimal points.

Similarly to the case with one objective function, we can define some desirable properties.

Definition 3.2.2 (Convex problem)**.** The multi-objective optimization problem (3.6) is *convex* if all the objective functions f_i and the feasible set Ω are convex. □

The methods used in multi-objective optimization are typically divided into four classes based on whether a decision maker is available or not (e.g., [6, 7]).

- If no decision maker is available then *no-preference methods* are used, where a neutral compromise Pareto optimal solution has to be selected.

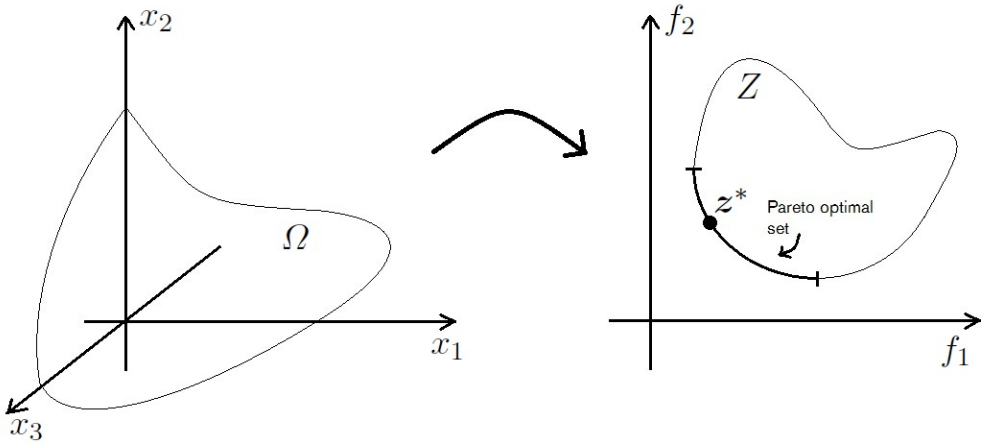

Figure 3.1: An illustration of the Pareto optimal set for a bi-objective optimization problem with the feasible set $\Omega \subset \mathbb{R}^3$. Left: the variable space. Right: the objective space.

- Another class of methods is used if the decision maker formulates hopes. Then the closest solution to those hopes is found. Those methods are denoted *a priori methods*. However, it may be difficult to express preferences without deep knowledge about the problem.
- In *a posteriori methods* a representation of the Pareto optimal set is found before the decision maker chooses one solution.
- The final class of methods are the *interactive methods*, which iteratively search through the Pareto optimal set in guidance of the decision maker.

To exemplify one no-preference method we first have to define the ideal objective point.

Definition 3.2.3 (Ideal objective point)**.** The components z_i^* of the *ideal objective point* $\boldsymbol{z}^* \in \mathbb{R}^n$ are obtained by minimizing each of the objective functions individually subject to the constraints, that is, by solving the problem to

$$\begin{aligned} &\text{minimize } f_i(\boldsymbol{x}), \\ &\text{subject to } \boldsymbol{x} \in \Omega, \end{aligned}$$

for $i = 1, \ldots, n$. □

Now we consider a so-called L_p*-problem*, which is the optimization problem to

$$\begin{aligned} &\text{minimize } \left(\sum_{i=1}^{n} |f_i(\mathbf{x}) - z_i^*|^p \right)^{1/p}, \\ &\text{subject to } \boldsymbol{x} \in \Omega, \end{aligned} \tag{3.7}$$

where z_i^* are the components of the ideal objective point $\boldsymbol{z}^*$ and $1 \leq p < \infty$.

Theorem 3.2.4. *The solution to the L_p-problem (3.7) is Pareto optimal in (3.6).*

Proof. Let $\boldsymbol{x}^* \in \Omega$ be a solution to problem (3.7) with $1 \leq p < \infty$. Assume that $\boldsymbol{x}^*$ is not Pareto optimal to (3.6). Then, there exists a point $\boldsymbol{x} \in \Omega$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, n$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one j. Now, the inequality $(f_i(\boldsymbol{x}) - z_i^*)^p \leq (f_i(\boldsymbol{x}^*) - z_i^*)^p$ holds for all $i = 1, \ldots, n$, and the strict inequality $(f_j(\boldsymbol{x}) - z_j^*)^p < (f_j(\boldsymbol{x}^*) - z_j^*)^p$ holds. We have

$$\sum_{i=1}^{n} (f_i(\boldsymbol{x}) - z_i^*)^p < \sum_{i=1}^{n} (f_i(\boldsymbol{x}^*) - z_i^*)^p.$$

Raising both sides of the inequality to the power $1/p$ yields reach a contradiction to the assumption that $\boldsymbol{x}^*$ is optimal in (3.7). □

Another possible approach to solving multi-objective optimization problems is to weigh all the objective functions into one objective function and then apply suitable single objective global optimization methods. However, there may not always exist information to base the weight decision on.

We conclude this section with an a posteriori method, called the weighting method, which is based on the weighting idea. Consider the *weighted problem*, which is the optimization problem to

$$\begin{aligned} &\text{minimize } \sum_{i=1}^{n} w_i f_i(\boldsymbol{x}), \\ &\text{subject to } \boldsymbol{x} \in \Omega, \end{aligned} \tag{3.8}$$

where it holds that $w_i \geq 0$ for all $i = 1, \ldots, n$ and $\sum_{i=1}^{n} w_i = 1$.

Theorem 3.2.5. *The solution to the weighted problem (3.8) is Pareto optimal if all the weighting coefficients are positive, that is $w_i > 0$ for all $i = 1, \ldots, n$.*

Proof. Let $\boldsymbol{x}^* \in \Omega$ be an optimal solution to (3.8) with positive weighting coefficients. Assume that x^* is not Pareto optimal. This means that there exists a solution $\boldsymbol{x} \in \Omega$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, n$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one j. Since $w_i > 0$ for all $i = 1, \ldots, n$ we have that the inequality $\sum_{i=1}^{n} w_i f_i(\boldsymbol{x}) < \sum_{i=1}^{n} w_i f_i(\boldsymbol{x}^*)$ holds. This contradicts the assumption that $\boldsymbol{x}^*$ is an optimal solution to the weighted problem (3.8). □

3.3 Simulation-based optimization

A frequently used tool to evaluate outputs from models of real systems is computer simulation. Its applications appear in many different areas, such as portfolio selection ([8]), manufacturing ([9]), engineering design ([10]), and bio medicine ([11]). By choosing optimal parameter settings for the simulation an extensively improved results can be achieved. However, finding the optimal parameter values is a challenging problem and this is where the field of *simulation-based optimization* has emerged. In simulation-based optimization the assumption is that the objective function, the simulation-based function, is not directly available due to the complexity of the simulation. Thereby, many mathematical tools, e.g., derivatives, are not available.

To avoid confusion we need to clarify that we want to find the optimal set of parameters for a computer simulation (see Section 2.3) which means that the parameters will act as variables in the optimization problem. Precisely as [12], we are treating the computer simulations as black-box functions.

There exist many continuous simulation-based optimization methods, but none of them can guarantee finding an optimal solution in finitely many steps. This is due to the fact that the objective function is not directly available, and thereby no strong convergence analysis can be made. The methods can be categorized into various groups, for instance *gradient based search methods*, *metaheuristics* and *response surface methodology* ([13, 14]).

Gradient based search methods estimate the gradient of the black-box function and employ deterministic mathematical programming techniques.

Metaheuristics are methods that interact between local improvement procedures and effective strategies of escaping from local optima and performing an efficient search in the solution space. Three of the most popular are *tabu search*, *simulated annealing* and *genetic algorithms* ([15, 16]). Usually the metaheruistics include strategies to handle multiple objective functions; see [17] for a tutorial of multi-objective optimization using genetic algorithms.

The idea of the response surface methodology is to construct a *surrogate model*, also known as a *response surface*, that mimics the behavior of the black-box function as closely as possible. Then global optimization algorithms are applied to the surrogate model. The advantage is that more efficient algorithms can be applied to the surrogate problem, as it is typically explicitly stated. Typically, response surface methods are used when the evaluation of the simulation-based function is very time consuming. The general procedure (see [7]) for a response surface method is detailed in Algorithm 1.

In this thesis we present one response surface method, namely *radial basis function interpolation* (RBF); see [18]. The RBF interpolation is independent of the dimension of the variable space, which is a desirable property for the problem studied in this project.

Algorithm 1 General response surface method

Step 0:
Create an initial set of sample points and evaluate them through a simulation.

Step 1:
Construct a surrogate model of the simulation-based function by using the evaluated points and their corresponding function values.

Step 2:
Select and evaluate a new sample point, and balance local and global search, to refine the surrogate model.

Step 3:
Return to **Step 1** unless a termination criterion is fulfilled.

Step 4:
Solve the simulation-based optimization problem where the objective function is replaced by the constructed surrogate model.

Chapter 4

Radial basis functions

The theory presented in Sections 4.1 and 4.2 is mainly taken from [18] and complemented with theory from [7] and [12]. In Section 4.1 we present the inspiration and background of radial basis functions. In Section 4.2 we present theory of radial basis functions, and we conclude in Section 4.3 with error estimation, which presents convergence properties of radial basis functions. We state and prove Theorem 4.2.5, inspired from theorems and proofs in [18].

4.1 Background of radial basis functions

Interpolation is the task of determining a continuous function $s : \mathbb{R}^d \to \mathbb{R}$ such that each point in a given finite set $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$ as well as the unknown function $f : \mathbb{R}^d \to \mathbb{R}$ satisfy

$$s(\boldsymbol{x}_i) = f(\boldsymbol{x}_i), \qquad i = 1, \ldots, n. \tag{4.1}$$

If the dimension d is equal to 1 and $s \in C^0$, i.e., s is from the space of continuous functions, then the problem (4.1) has multiple solutions. However, if we consider a specific finite dimensional linear subspace then the problem (4.1) has a unique solution. An intuitive choice of space for n points in one dimension is the space of polynomials of degree at most $n-1$, denoted $\pi_{n-1}(\mathbb{R})$. Then, the existence of a unique solution to the system of equations (4.1) is guaranteed, but the utility of polynomials is limited, because the required degree of the polynomials increases with the number of evaluated points. The result of using higher degree polynomials is often strongly oscillating interpolating functions (see Figure 4.1), which is an undesired effect. However, this can be avoided by partitioning the one-dimensional space into intervals between the data points, and then utilizing a polynomial interpolation of lower degree m, such as cubic, where $m = 3$, in each interval. The function values and the values of the first $m-1$ derivatives of these polynomials have to agree at the points where they join. These piecewise polynomials are called *splines*. We summarize this problem in the following way: Let the data points be ordered according to

$$a < x_1 < \ldots < x_n < b.$$

Define $x_0 := a$, $x_{n+1} := b$, and the function space of cubic splines by

$$S_3(X) := \left\{ s \in C^2([a,b]) : s|_{[x_i, x_{i+1}]} \in \pi_3(\mathbb{R}),\ i = 0, \ldots, n \right\}.$$

The task is to find $s \in S_3$ such that the equations (4.1) are fulfilled. Figure 4.1 shows a comparison between spline and polynomial interpolation.

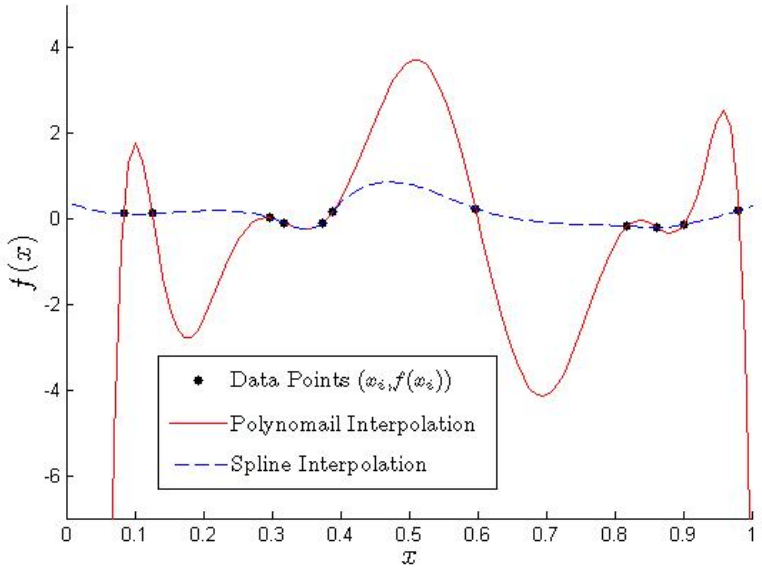

Figure 4.1: An illustration of the differences between the spline and polynomial interpolations.

There is no guarantee that there is a unique interpolation $s \in S_3$ that fulfills (4.1), but it is possible to enforce uniqueness through the concept of natural cubic splines according to

$$\mathcal{N}S_3(X) := \left\{ s \in S_3(X) : s|_{[a,x_1]}, s|_{[x_n,b]} \in \pi_1(\mathbb{R}) \right\}.$$

Unfortunately, interpolating multivariate functions is much more complicated. Therefore, we introduce Haar spaces to understand the complication.

Definition 4.1.1 (Haar space)**.** Assume that $\Omega \subseteq \mathbb{R}^d$ contains at least n points. Let $V \subseteq C(\Omega)$ be an n-dimensional linear space. Then V is called a *Haar space* of dimension n on Ω if for arbitrary distinct points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \Omega$ and arbitrary $f_1, \ldots, f_n \in \mathbb{R}$ there exists exactly one function $v \in V$ with $v(\boldsymbol{x}_i) = f_i$, $1 \leq i \leq n$. □

For instance, $V = \pi_{n-1}(\mathbb{R})$ is a n-dimensional Haar space for any set $\Omega \subseteq \mathbb{R}$ that contains at least n points, as we noted above. We now present a theorem which provides the insight into the problem of interpolation when the dimension of the domain is higher than 1. Its proof is found in [18, Thm. 2.3].

Theorem 4.1.2 (Mairhuber–Curtis)**.** *Assume that $\Omega \subseteq \mathbb{R}^d$, $d \geq 2$, contains an interior point. Then there exists no Haar space on Ω of dimension $n \geq 2$.* □

Fortunately, there exists a field with multi-variable settings which takes its inspiration from the one-dimensional natural cubic splines. We introduce radial basis functions.

4.2 Radial basis functions

If we still want to interpolate some data values $f_1, \ldots, f_n \in \mathbb{R}$ at some given data points $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$, for any positive integer d, despite Theorem 4.1.2 one simple way is to choose a fixed function $\Phi : \mathbb{R}^d \to \mathbb{R}$ and form the interpolant as

$$s_{f,X}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \Phi(\boldsymbol{x} - \boldsymbol{x}_i), \tag{4.2}$$

where the values of the coefficients α_j are determined by the interpolation conditions

$$s_{f,X}(\boldsymbol{x}_i) = f_i, \quad i \in \{1, \ldots, n\}. \tag{4.3}$$

The desirable property would be that the function Φ could be chosen for all kinds of data point sets, i.e., for any number $n \in \mathbb{N}$ and any possible combination $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$. An equivalent formulation of the interpolant (4.2) and interpolation conditions (4.3) is asking for an invertible interpolation matrix

$$A_{\Phi,X} := (\Phi(\boldsymbol{x}_i - \boldsymbol{x}_j))_{1 \le i,j \le n},$$

We know that any real symmetric matrix that is positive definite is also invertible; see Appendix B. This makes it natural to introduce the following definition.

Definition 4.2.1 (Positive definite function)**.** A continuous function $\Phi : \mathbb{R}^d \to \mathbb{C}$ is called *positive definite* if, for all $n \in \mathbb{N}$, all sets of pairwise distinct points $X = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\} \subseteq \mathbb{R}^d$, and all $\boldsymbol{\alpha} \in \mathbb{C}^n \backslash \{\boldsymbol{0}^n\}$ it holds that

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \overline{\alpha}_j \Phi(\boldsymbol{x}_i - \boldsymbol{x}_j) > 0, \tag{4.4}$$

where $\overline{\boldsymbol{\alpha}}$ is complex conjugation of $\boldsymbol{\alpha}$. The function Φ is called *positive semi-definite* if the left-hand-side of (4.4) is nonnegative for all $\boldsymbol{\alpha} \in \mathbb{C}^n$. □

As can be seen in Definition 4.2.1 a more general definition for complex-valued functions have been used; the reason is that it allows more natural for techniques such as Fourier transforms. Next we introduce the term *radial basis function* which is the foundation of the interpolation theory to be presented.

Definition 4.2.2 (Radial basis function, RBF)**.** A function $\Phi : \mathbb{R}^d \to \mathbb{R}$ is called a *radial basis function* if there exists a univariate function $\phi : [0, \infty) \to \mathbb{R}$ such that

$$\Phi(\boldsymbol{x}) = \phi(\|\boldsymbol{x}\|), \quad \boldsymbol{x} \in \mathbb{R}^d,$$

where $\|\cdot\|$ denotes the Euclidean norm. □

We link radial basis functions and positive definite functions by the following definition.

Definition 4.2.3. A univariate function $\phi : [0, \infty) \to \mathbb{R}$ is said to be *positive definite* on $\mathbb{R}^d$ if the corresponding multivariate function $\Phi(\boldsymbol{x}) := \phi(\|\boldsymbol{x}\|), \boldsymbol{x} \in \mathbb{R}^d$, is positive definite. □

However, using an interpolant of the form described in (4.2) is not the only approach possible. A more general is to start with a function $\Phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{C}$ and form the interpolant as

$$s_{f,X}(\boldsymbol{x}) = \sum_{j=1}^{n} \alpha_j \Phi(\boldsymbol{x}, \boldsymbol{x}_j).$$

Furthermore, if we are only interested in some points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ that belong to a certain subset $\Omega \subseteq \mathbb{R}^d$ then we only need a function $\Phi : \Omega \times \Omega \to \mathbb{C}$. This kind of function Φ will be called a *kernel*, to mark the difference from functions defined on $\mathbb{R}^d \times \mathbb{R}^d$.

Definition 4.2.4 (Positive definite kernel)**.** A continuous kernel $\Phi : \Omega \times \Omega \to \mathbb{C}$ is called *positive definite* on a non-empty set $\Omega \subseteq \mathbb{R}^d$, if for all $n \in \mathbb{N}$, all sets of pairwise distinct points $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \Omega$, and all $\boldsymbol{\alpha} \in \mathbb{C}^n \backslash \{\boldsymbol{0}^n\}$ it holds that

$$\sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_j \overline{\alpha}_k \Phi(\boldsymbol{x}_j, \boldsymbol{x}_k) > 0.$$

□

This definition is not precise due to the fact that the set Ω is not specified, so the set might be finite. If this is the case it would be impossible to find for all $n \in \mathbb{N}$ pairwise distinct points in Ω. However, if the set is finite the only values of $n \in \mathbb{N}$ that would have to be considered are those that allow the choice of n pairwise distinct points.

The radial basis function $\phi : [0, \infty) \to \mathbb{R}$ fits into this generalization of introducing the kernel by defining $\Phi(\boldsymbol{x}, \boldsymbol{y}) := \phi(\|\boldsymbol{x} - \boldsymbol{y}\|)$. The univariate function ϕ is called *positive definite* on $\Omega \subseteq \mathbb{R}^d$ if the kernel $\Phi(\boldsymbol{x}, \boldsymbol{y})$ is positive definite on Ω.

The restriction to real coefficients for a positive definite kernel, i.e., $\boldsymbol{\alpha} \in \mathbb{R}^n$ instead of $\boldsymbol{\alpha} \in \mathbb{C}^n$, in Definition 4.2.4, is explained in the following theorem.

Theorem 4.2.5. *Assume that $\Phi : \Omega \times \Omega \to \mathbb{R}$ is continuous. Then Φ is positive definite on $\Omega \subseteq \mathbb{R}^d$ if and only if Φ is symmetric and, for all $n \in \mathbb{N}$, all sets of pairwise distinct points $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \Omega$, for all $\boldsymbol{\alpha} \in \mathbb{R}^n \backslash \{\boldsymbol{0}^n\}$ it holds that*

$$\sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_j \alpha_k \Phi(\boldsymbol{x}_j, \boldsymbol{x}_k) > 0. \tag{4.5}$$

Proof. [$\Longrightarrow$]: Assume that Φ is positive definite on Ω. First we prove that $\Phi(\boldsymbol{x}, \boldsymbol{x}) > 0$, for all $\boldsymbol{x} \in \Omega$. Choose $n = 1$ and $\alpha_1 = 1$ in Definition 4.2.4 and let $\boldsymbol{x}_1 = \boldsymbol{x}$, where $\boldsymbol{x} \in \Omega$ and the desired result is obtained. Further we prove that Φ is symmetric, i.e., $\Phi(\boldsymbol{x}, \boldsymbol{y}) = \Phi(\boldsymbol{y}, \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$. Assume there are at least two distinct points in Ω (otherwise the result is trivial). Choose $n = 2, \alpha_1 = 1, \alpha_2 = c, \boldsymbol{x}_1 = \boldsymbol{x}$, and $\boldsymbol{x}_2 = \boldsymbol{y}$, where $\boldsymbol{x}, \boldsymbol{y} \in \Omega$. If we let $c = 1$ and $c = i$, respectively, we have that the inequalities

$$\Phi(\boldsymbol{x}, \boldsymbol{x}) + \Phi(\boldsymbol{y}, \boldsymbol{y}) + \Phi(\boldsymbol{x}, \boldsymbol{y}) + \Phi(\boldsymbol{y}, \boldsymbol{x}) > 0,$$

and

$$\Phi(\boldsymbol{x},\boldsymbol{x}) + \Phi(\boldsymbol{y},\boldsymbol{y}) + i\big(\Phi(\boldsymbol{y},\boldsymbol{x}) - \Phi(\boldsymbol{x},\boldsymbol{y})\big) > 0$$

holds. Since $\Phi(\boldsymbol{x},\boldsymbol{x}) > 0$ and $\Phi(\boldsymbol{y},\boldsymbol{y}) > 0$, both $\Phi(\boldsymbol{x},\boldsymbol{y}) + \Phi(\boldsymbol{y},\boldsymbol{x})$ and $i\big(\Phi(\boldsymbol{y},\boldsymbol{x}) - \Phi(\boldsymbol{x},\boldsymbol{y})\big)$ must be real. This is only possible if $\Phi(\boldsymbol{x},\boldsymbol{y}) = \Phi(\boldsymbol{y},\boldsymbol{x})$. Since Φ is positive definite on Ω the inequality (4.5) is obviously satisfied.
[$\Longleftarrow$]: Now assume that Φ satisfies the given conditions. Let $\alpha_j = a_j + ib_j$, where $a_j, b_j \in \mathbb{R}$. Then it holds that

$$\sum_{j=1}^{n}\sum_{k=1}^{n} \alpha_j \overline{\alpha}_k \Phi(\boldsymbol{x}_j,\boldsymbol{x}_k) = \sum_{j=1}^{n}\sum_{k=1}^{n} (a_j a_k + b_j b_k)\Phi(\boldsymbol{x}_j,\boldsymbol{x}_k) + i\sum_{j=1}^{n}\sum_{k=1}^{n} \Big(a_k b_j \Phi(\boldsymbol{x}_j,\boldsymbol{x}_k) - a_j b_k \Phi(\boldsymbol{x}_k,\boldsymbol{x}_j)\Big).$$

The second sum on the right-hand side, resulting from the symmetry of Φ, is equal to zero. The first sum on the right-hand side is nonnegative because of the assumption and vanishes only if $a_j = b_j = 0$, $j = 1,\dots,n$. □

Next we present a theorem to verify when a function is positive definite, but first we need to introduce the following definition.

Definition 4.2.6 (Completely monotone)**.** A function ϕ is called *completely monotone* on $(0,\infty)$ if $\phi \in C^\infty(0,\infty)$ and

$$(-1)^l \phi^{(l)}(r) \geq 0,$$

for all $l \in \mathbb{N} \cup \{0\}$ and all $r > 0$. The function ϕ is called completely monotone on $[0,\infty)$ if it is in addition in $C[0,\infty)$. □

The proof of Theorem 4.2.7 is found in [18, Thm. 7.14].

Theorem 4.2.7. *The function $\phi : [0,\infty) \to \mathbb{R}$ is positive definite on every $\mathbb{R}^d$ if and only if $\phi(\sqrt{\cdot})$ is completely monotone on $[0,\infty)$ and not constant.* □

We now present two radial basis functions, and verify that they are positive definite. The first radial basis function that we present is the *Guassian* radial basis function defined by

$$\Phi(\boldsymbol{x}) := \phi(r) := e^{-\alpha r^2},$$

where $r = \|\boldsymbol{x}\|$ and $\alpha > 0$. By Theorem 4.2.7 the Gaussian function is positive definite on every $\mathbb{R}^d$ due to the following: Set $f(r) := \phi(\sqrt{r})$; then f is completely monotone, since

$$(-1)^l f^{(l)}(r) = (-1)^{2l}\alpha^l e^{-\alpha r} \geq 0.$$

Since f is not constant, ϕ must be positive definite.

The second radial basis function that we present is the *inverse multiquadrics* radial basis function, defined as

$$\Phi(\boldsymbol{x}) := \phi(r) := (c^2 + r^2)^{-\beta},$$

where $r = \|\boldsymbol{x}\|$, $\beta > 0$ and $c > 0$. Again by Theorem 4.2.7 the inverse multiquadrics function is positive definite on every $\mathbb{R}^d$ according to the following: Set $f(r) := \phi(\sqrt{r})$; then f is completely monotone since

$$(-1)^l f^{(l)}(r) = (-1)^{2l}\beta(\beta+1)\cdots(\beta+l-1)(r+c^2)^{-\beta-l} \geq 0.$$

Since f is not constant, ϕ must be positive definite.

Next we will relax the condition of positive definiteness to allow a wider range of radial basis functions.

Definition 4.2.8 (Conditionally positive definite function)**.** A continuous function $\Phi : \mathbb{R}^d \to \mathbb{C}$ is said to be *conditionally positive definite of order* m if, for all $n \in \mathbb{N}$, all sets of pairwise distinct points $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \mathbb{R}^d$, and all $\boldsymbol{\alpha} \in V_m \backslash \{\boldsymbol{0}^n\}$ it holds that

$$\sum_{j=1}^{n} \alpha_j \overline{\alpha}_k \Phi(\boldsymbol{x}_j - \boldsymbol{x}_k) > 0,$$

where

$$V_m = \left\{ \boldsymbol{\alpha} \in \mathbb{C}^n : \sum_{j=1}^{n} \alpha_j p(\boldsymbol{x}_j) = 0, \quad p \in \pi_{m-1}(\mathbb{R}^d) \right\}.$$

If instead $\sum_{j=1}^{n} \alpha_j \overline{\alpha}_k \Phi(\boldsymbol{x}_j - \boldsymbol{x}_k) \geq 0$ the function is said to be *conditionally positive semidefinite* of order m in $\mathbb{R}^d$. □

Note that if $m > l$, then a conditionally positive definite function of order l is also conditionally positive definite of order m since $V_m \subseteq V_l$. Furthermore, note that if the order $m = 0$ the function is positive definite.

Definition 4.2.9. A univariate function $\phi : [0, \infty) \to \mathbb{R}$ is called *conditionally positive definite of order* m on $\mathbb{R}^d$, if $\Phi(\boldsymbol{x}) := \phi(\|\boldsymbol{x}\|)$ is conditionally positive definite of order m. □

We now present a theorem that verifies when a function is conditionally positive definite. The proof can be found in [18, Thm. 8.19].

Theorem 4.2.10. *Suppose that $\phi \in C[0, \infty) \cap C^\infty(0, \infty)$ is given. Then the function $\Phi = \phi(\|\cdot\|^2)$ is conditionally positive semi-definite of order $m \in \mathbb{N} \cup \{0\}$ on every $\mathbb{R}^d$ if and only if $(-1)^m \phi^{(m)}$ is completely monotone on $(0, \infty)$.* □

Corollary 4.2.11. *Assume that $\phi \in C[0, \infty) \cap C^\infty(0, \infty)$, and that it is not a polynomial of degree at most m. Then $\phi(\|\cdot\|)$ is conditionally positive definite of order m on every $\mathbb{R}^d$ if $(-1)^m \phi^{(m)}$ is completely monotone on $(0, \infty)$.* □

Now we present three additional radial basis functions, and verify that they are conditionally positive definite. The third radial basis function that we present in this thesis is the *multiquadrics* radial basis function defined by

$$\Phi(\boldsymbol{x}) := \phi(r) := (-1)^{\lceil \beta \rceil}(c^2 + r^2)^\beta,$$

where $\|\boldsymbol{x}\| = r$, $c, \beta > 0$, $\beta \notin \mathbb{N}$, and $\lceil \cdot \rceil$ denotes the ceiling function. The multiquadrics is conditionally positive definite of order $m = \lceil \beta \rceil$ on every $\mathbb{R}^d$, which can be verified by using Corollary 4.2.11: If we let $f_\beta(r) := (-1)^{\lceil \beta \rceil}(c^2 + r)^\beta$ then we have that

$$(-1)^{\lceil \beta \rceil} f_\beta^{(\lceil \beta \rceil)}(r) = (-1)^{2\lceil \beta \rceil}\beta(\beta - 1)\cdots(\beta - \lceil \beta \rceil + 1)(c^2 + r)^{\beta - \lceil \beta \rceil},$$

which is completely monotone, and $m = \lceil \beta \rceil$ is the smallest possible choice that makes $(-1)^m f_\beta^{(m)}$ completely monotone. Thus, ϕ is conditionally positive definite of order $m = \lceil \beta \rceil$ on every $\mathbb{R}^d$, since $\phi \notin \pi_m(\mathbb{R})$.

The fourth radial basis function that we present is the *powers* radial basis function, defined as

$$\boldsymbol{\Phi}(\boldsymbol{x}) := \phi(r) := (-1)^{\lceil \beta/2 \rceil} r^\beta,$$

where $\|\boldsymbol{x}\| = r$, and $\beta > 0, \beta \notin 2\mathbb{N}$. The powers is conditionally positive definite of order $m = \lceil \beta/2 \rceil$ on every $\mathbb{R}^d$ which can be verified by using Corollary 4.2.11: If we define $f_\beta(r) := (-1)^{\lceil \beta/2 \rceil} r^{\beta/2}$ then we get

$$(-1)^{\lceil \beta/2 \rceil} f_\beta^{(\lceil \beta/2 \rceil)}(r) = (-1)^{2\lceil \beta/2 \rceil}\beta/2(\beta/2 - 1)\cdots(\beta/2 - \lceil \beta/2 \rceil + 1) r^{\beta/2 - \lceil \beta/2 \rceil},$$

which is completely monotone, and $m = \lceil \beta/2 \rceil$ is the smallest possible choice that makes $(-1)^m f_\beta^{(m)}$ completely monotone. Hence ϕ is conditionally positive definite of order $m = \lceil \beta/2 \rceil$ on every $\mathbb{R}^d$, since $\phi \notin \pi_m(\mathbb{R})$.

The fifth and final radial basis function that we present is the *thin-plate* or *surface splines* radial basis function defined by

$$\boldsymbol{\Phi}(\boldsymbol{x}) := \phi(r) := (-1)^{k+1} r^{2k} \log(r),$$

where $\|\boldsymbol{x}\| = r$, and $k \in \mathbb{N}$. The thin-plate spline is positive definite of order $m = k + 1$ on every $\mathbb{R}^d$, which can be verified by Corollary 4.2.11. Since $2\phi(r) = (-1)^{k+1} r^{2k} \log(r^2)$, we define $f_k(r) := (-1)^{k+1} r^k \log(r)$, and achieve

$$f_k^{(l)}(r) = (-1)^{k+1} k(k-1)\cdots(k - l + 1) r^{k-l} \log(r) + p_l(r), \quad 1 \le l \le k,$$

where p_l is a polynomial of degree $k - l$. This means that $f_k^{(k)}(r) = (-1)^{k+1} k! \log(r) + c$, where c is a constant. Finally we have that $(-1)^{k+1} f_k^{(k+1)}(r) = k! r^{-1}$, so f is clearly completely monotone on $(0, \infty)$. Hence, ϕ is conditionally positive definite of order $m = k + 1$ on every $\mathbb{R}^d$, since $\phi \notin \pi_m(\mathbb{R})$.

Note that the thin-plate spline is not defined at $r = 0$ since $\log(r)$ is not defined there. A radial basis should be defined on $[0, \infty)$ so we need to extend it. Since it holds that $\lim_{r \to 0^+} (-1)^k r^{2k} \log(r) = 0$, for all $k \in \mathbb{N}$ it follows naturally that

$$\phi(r) := \begin{cases} (-1)^k r^{2k} \log(r), & \text{if } r > 0, \\ 0, & \text{if } r = 0. \end{cases}$$

All the presented radial basis functions, for this thesis, are compiled in Table 4.1.

Table 4.1: A compilation of the radial basis functions presented in this thesis with their corresponding order of conditionally positive definiteness.

Name	$\Phi(\boldsymbol{x}) = \phi(r), \quad r = \|\boldsymbol{x}\|$	Conditionally positive definite of order
Guassians	$e^{-\alpha r^2}, \quad \alpha > 0$	0
Multiquadrics	$(-1)^{\lceil\beta\rceil}(c^2 + r^2)^\beta, \quad \beta > 0, \beta \notin \mathbb{N}$	0
Inverse multiquadrics	$(c^2 + r^2)^\beta, \quad \beta < 0$	$\lceil\beta\rceil$
Thin-plate splines	$(-1)^{k+1} r^{2k} \log r, \quad k \in \mathbb{N}$	$\lceil\beta/2\rceil$
Powers	$(-1)^{\lceil\beta/2\rceil} r^\beta, \quad \beta > 0, \beta \notin 2\mathbb{N}$	$k+1$

Now we have all the necessary tools to define the interpolation problem and how to solve it. Let $\pi_m(\mathbb{R}^d)$ denote the space of all polynomials of degree at most m in $\mathbb{R}^d$. The points $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ with corresponding function values $f_1, \ldots, f_n$, the task is to find $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\boldsymbol{\beta} \in \mathbb{R}^Q$ such that:

$$s_{f,X}(\boldsymbol{x}) := \sum_{j=1}^{n} \alpha_j \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) + \sum_{k=1}^{Q} \beta_k p_k(\boldsymbol{x}), \tag{4.6}$$

$$s_{f,X}(\boldsymbol{x}_i) = f_i, \quad i = 1, \ldots, n, \tag{4.7}$$

$$\sum_{j=1}^{n} \alpha_j p_k(\boldsymbol{x}_j) = 0, \quad k = 1, \ldots, Q, \tag{4.8}$$

where $Q = \dim(\pi_{m-1}(\mathbb{R}^d))$ and $\{p_k\}_{k=1}^Q$ is a basis of $\pi_{m-1}(\mathbb{R}^d)$. Therefore $\boldsymbol{\alpha}$ satisfying the equations (4.8) is equivalent to $\boldsymbol{\alpha} \in V_m$ (from Definition 4.2.8). Solving this interpolation problem is equivalent to solving the linear system

$$\begin{pmatrix} A & P \\ P^T & 0^{Q\times Q} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0}^Q \end{pmatrix}, \tag{4.9}$$

where

$$\begin{aligned} A_{ij} &:= \phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \quad && i, j = 1, \ldots, n, \\ P_{ik} &:= p_k(\boldsymbol{x}_i), && i = 1, \ldots, n,\ k = 1, \ldots, Q. \end{aligned}$$

For convenience we use the notation

$$\tilde{A} := \begin{pmatrix} A & P \\ P^T & 0^{Q\times Q} \end{pmatrix}. \tag{4.10}$$

Assuming suitable condition on the points in the set X, we can establish the existence and uniqueness of a solution to the system (4.9).

Definition 4.2.12. The points $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \mathbb{R}^d$, with $n \geq \dim(\pi_m(\mathbb{R}^d))$, are called $\pi_m(\mathbb{R}^d)$-*unisolvent* if the zero polynomial is the only polynomial from $\pi_m(\mathbb{R}^d)$ that vanishes at all of the points in X. □

To provide an example from Definition 4.2.12, we assume that we are given some points X that are $\pi_1(\mathbb{R}^d)$-unisolvent. This means that not all points in X belong to a common hyperplane. We can now present following theorem.

Theorem 4.2.13. *Assume that the radial function ϕ is conditionally positive definite of order m and the points $X := \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ are $\pi_{m-1}(\mathbb{R}^d)$-unisolvent. Then the linear system (4.9) is uniquely solvable.*

Proof. Assume that $(\boldsymbol{\alpha}^T, \boldsymbol{\beta}^T)^T$ lies in the null space of the matrix $\tilde{A}$. Then we have that

$$A\boldsymbol{\alpha} + P\boldsymbol{\beta} = \mathbf{0}^n, \tag{4.11}$$

$$P^T\boldsymbol{\alpha} = \mathbf{0}^Q. \tag{4.12}$$

The equation (4.12) means that $\boldsymbol{\alpha} \in V_m$ (see Definition 4.2.8). By multiplying equation (4.11) by $\boldsymbol{\alpha}^T$ from the left, it follows that $0 = \boldsymbol{\alpha}^T A\boldsymbol{\alpha} + (P^T\boldsymbol{\alpha})^T\boldsymbol{\beta} = \boldsymbol{\alpha}^T A\boldsymbol{\alpha}$. Since ϕ is conditionally positive definite of order m, we have that $\boldsymbol{\alpha} = 0$ and hence $P\boldsymbol{\beta} = \mathbf{0}^n$ holds. Since X is $\pi_{m-1}(\mathbb{R}^d)$-unisolvent and the vectors$\{p_k\}_{k=1}^Q$ are linearly independent we can conclude that $\boldsymbol{\beta} = \mathbf{0}^Q$. Thus, the only vector in the null space of $\tilde{A}$ is the null vector, which implies that $\tilde{A}$ is invertible. □

4.3 Error estimation for radial basis functions

In this section we discuss error estimation of radial basis functions. We begin by introducing *reproducing kernels* and the spaces generated by them, and in time we will see that a positive definite kernel can be identified as a reproducing kernel. Relevant definitions and theorems needed for the theory in this section can be found in Appendix B.

Definition 4.3.1 (Reproducing kernel)**.** Let $\mathcal{F}$ be a real Hilbert space of functions $f : \Omega \to \mathbb{R}$. A function $\Phi : \Omega \times \Omega \to \mathbb{R}$ is called a *reproducing kernel* for $\mathcal{F}$ if it fulfills the following two properties:

(1) $\Phi(\cdot, \boldsymbol{y}) \in \mathcal{F}$ for all $\boldsymbol{y} \in \Omega$, *and*
(2) $f(\boldsymbol{y}) = (f, \Phi(\cdot, \boldsymbol{y}))_{\mathcal{F}}$ for all $f \in \mathcal{F}$ and all $\boldsymbol{y} \in \Omega$,

where $(\cdot, \cdot)_{\mathcal{F}}$ denotes the inner product of the Hilbert space $\mathcal{F}$. □

Note that the reproducing kernel of a Hilbert space is uniquely determined. Assume that there exist two reproducing kernels Φ_1 and Φ_2. From property (2) in Definition 4.3.1 we have that $(f, \Phi_1(\cdot, \boldsymbol{y}))_{\mathcal{F}} - (f, \Phi_2(\cdot, \boldsymbol{y}))_{\mathcal{F}} = (f, \Phi_1(\cdot, \boldsymbol{y}) - \Phi_2(\cdot, \boldsymbol{y}))_{\mathcal{F}} = f(\boldsymbol{y}) - f(\boldsymbol{y}) = 0$ for all $f \in \mathcal{F}$ and all $y \in \Omega$. By letting $f = \Phi_1(\cdot, \boldsymbol{y}) - \Phi_2(\cdot, \boldsymbol{y})$ for a fixed $\boldsymbol{y}$, it follows from the definition of a norm that Φ_1 and Φ_2 are identical.

Definition 4.3.2 (Point evaluation functional)**.** Let $\mathcal{F}$ be Hilbert space of functions $f : \Omega \to \mathbb{R}$. A linear functional $\delta_y : \mathcal{F} \to \mathbb{R}$ is called the *point evaluation functional* for a fixed $\boldsymbol{y} \in \Omega$ on $\mathcal{F}$ if $\delta_{\boldsymbol{y}}(f) = f(\boldsymbol{y})$ for all $f \in \mathcal{F}$. □

Now we present a connection between the reproducing kernel and the point evaluation functional.

Theorem 4.3.3. *Assume that $\mathcal{F}$ is a Hilbert space of functions $f : \Omega \to \mathbb{R}$. Then the following statements are equivalent:*

(1) all the point evaluation functionals are continuous, i.e., $\delta_{\boldsymbol{y}} \in \mathcal{F}^*$ for all $\boldsymbol{y} \in \Omega$,

(2) $\mathcal{F}$ has a reproducing kernel,

where $\mathcal{F}^*$ denotes the dual space of $\mathcal{F}$.

Proof. (1) $\Rightarrow$ (2): Assume that the point evaluation functionals are continuous. By the Riesz Representation Theorem, see Theorem B.17 in Appendix B, we have that for every $\boldsymbol{y} \in \Omega$ there exists a unique element $\Phi_{\boldsymbol{y}} \in \mathcal{F}$ such that $\delta_{\boldsymbol{y}}(f) = (f, \Phi_{\boldsymbol{y}})$ for all $\boldsymbol{y} \in \Omega$. Hence, $\Phi(\boldsymbol{x}, \boldsymbol{y}) := \Phi_{\boldsymbol{y}}(\boldsymbol{x})$ is the reproducing kernel of $\mathcal{F}$.
(2) $\Rightarrow$ (1): Assume that $\mathcal{F}$ has a reproducing kernel Φ. This means that $\delta_{\boldsymbol{y}}(f) = f(\boldsymbol{y}) = (f, \Phi(\cdot, \boldsymbol{y}))_{\mathcal{F}}$ for $\boldsymbol{y} \in \Omega$ and for all $f \in \mathcal{F}$. Since the inner product is continuous, so is $\delta_{\boldsymbol{y}}$. □

A reproducing-kernel Hilbert space posses several special properties; a few of them are presented in Theorem 4.3.4.

Theorem 4.3.4. *Assume that $\mathcal{F}$ is a Hilbert space of functions $f : \Omega \to \mathbb{R}$ with reproducing kernel Φ. Then, the following hold:*

(1) $\Phi(\boldsymbol{x}, \boldsymbol{y}) = (\Phi(\cdot, \boldsymbol{x}), \Phi(\cdot, \boldsymbol{y}))_{\mathcal{F}} = (\delta_{\boldsymbol{x}}, \delta_{\boldsymbol{y}})_{\mathcal{F}^*}$ for $\boldsymbol{x}, \boldsymbol{y} \in \Omega$.

(2) $\Phi(\boldsymbol{x}, \boldsymbol{y}) = \Phi(\boldsymbol{y}, \boldsymbol{x})$ for $\boldsymbol{x}, \boldsymbol{y} \in \Omega$.

Proof. From the Riesz Representation Theorem we have that $F : \mathcal{F}^* \to \mathcal{F}$, which reduces for point evaluations to $F(\delta_{\boldsymbol{y}}) = \Phi(\cdot, \boldsymbol{y})$ due to the definition of a reproducing kernel. This means that

$$(\delta_{\boldsymbol{x}}, \delta_{\boldsymbol{y}})_{\mathcal{F}^*} = (F(\delta_{\boldsymbol{x}}), F(\delta_{\boldsymbol{y}}))_{\mathcal{F}} = (\Phi(\cdot, \boldsymbol{x}), \Phi(\cdot, \boldsymbol{y}))_{\mathcal{F}}$$

hold. Furthermore, it holds that

$$\Phi(\boldsymbol{x}, \boldsymbol{y}) = \delta_{\boldsymbol{x}}(\Phi(\cdot, \boldsymbol{y})) = (\Phi(\cdot, \boldsymbol{y}), \Phi(\cdot, \boldsymbol{x}))_{\mathcal{F}} = (\Phi(\cdot, \boldsymbol{x}), \Phi(\cdot, \boldsymbol{y}))_{\mathcal{F}}.$$

Hence, property (1) is proven; property (2) follows immediately from property (1), since the inner product is symmetric. □

We can now disclose the connection between positive definite kernels and reproducing-kernel Hilbert spaces.

Theorem 4.3.5. *Assume that $\mathcal{F}$ is a reproducing-kernel Hilbert function space with reproducing kernel $\Phi : \Omega \times \Omega \to \mathbb{R}$. Then Φ is positive semi-definite. Moreover, Φ is positive definite if and only if the point evaluation functionals are linearly independent in $\mathcal{F}^*$.*

Proof. Since the kernel Φ is symmetric and real-valued it follows from Theorem 4.2.5 that we can restrict ourselves to real coefficients in the quadratic form. For pairwise distinct points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \Omega \subseteq \mathbb{R}^d$ and $\boldsymbol{c} \in \mathbb{R}^n \backslash \{\boldsymbol{0}\}$ we have that

$$\sum_{j=1}^{n} \sum_{k=1}^{n} c_j c_k \Phi(\boldsymbol{x}_j, \boldsymbol{x}_k) = \left(\sum_{j=1}^{n} c_j \delta_{\boldsymbol{x}_j}, \sum_{k=1}^{n} c_k \delta_{\boldsymbol{x}_k} \right)_{\mathcal{F}^*} = \left\| \sum_{j=1}^{n} c_j \delta_{\boldsymbol{x}_j} \right\|_{\mathcal{F}^*}^2 \geq 0.$$

The last expression can and will only be zero if the point evaluation functionals are linearly dependent. □

We have now established that a positive definite kernel appears naturally as the reproducing kernel of a Hilbert space. However, normally we don't start with a function space but rather with a positive definite kernel. In other words, we are facing the problem of finding the associated function space having a a positive definite kernel as its reproducing kernel. Suppose that $\Phi : \Omega \times \Omega \to \mathbb{R}$ is a symmetric positive definite kernel. We define the following space:

$$F_\Phi(\Omega) := \operatorname{span}\{\Phi(\cdot, \boldsymbol{y}) : \boldsymbol{y} \in \Omega\}$$

and equip it with the bilinear form

$$\left(\sum_{j=1}^{n} \alpha_j \Phi(\cdot, \boldsymbol{x}_j), \sum_{k=1}^{m} \beta_k \Phi(\cdot, \boldsymbol{y}_k) \right)_\Phi := \sum_{j=1}^{n} \sum_{k=1}^{m} \alpha_j \beta_k \Phi(\boldsymbol{x}_j, \boldsymbol{y}_k). \tag{4.13}$$

Theorem 4.3.6. *If $\Phi : \Omega \times \Omega \to \mathbb{R}$ is a symmetric positive definite kernel, then the bilinear form $(\cdot, \cdot)_\Phi$ defines an inner product on $F_\Phi(\Omega)$. Moreover, $F_\Phi(\Omega)$ is a pre-Hilbert space with reproducing kernel Φ.*

Proof. Obviously $(\cdot, \cdot)_\Phi$ is symmetric since Φ is symmetric. Furthermore, if we choose an arbitrary function $f = \sum_{j=1}^{N} \alpha_j \Phi(\cdot, \boldsymbol{x}_j) \not\equiv 0$ from $F_\Phi(\Omega)$ we find that

$$(f, f)_\Phi = \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_j \alpha_k \Phi(\boldsymbol{x}_j, \boldsymbol{x}_k) > 0,$$

because Φ is positive definite. At last, for this f we obtain

$$(f, \Phi(\cdot, \boldsymbol{y}))_\Phi = \sum_{j=1}^{n} \alpha_j \Phi(\boldsymbol{x}_j, \boldsymbol{y}) = f(\boldsymbol{y}),$$

which establishes the reproducing kernel. □

The completion, $\mathcal{F}_\Phi(\Omega)$, of the pre-Hilbert space $F_\phi(\Omega)$ with respect to the norm $\|\cdot\|_\Phi$ is a candidate for a Hilbert function space with reproducing kernel Φ. However, the elements of $\mathcal{F}_\Phi(\Omega)$ are abstract elements which need to be interpreted as functions. As a result of the point-evaluation functionals being continuous on $F_\Phi(\Omega)$, their extension to the completion remain continuous, and this idea can is used for defining function values for the elements in $\mathcal{F}_\Phi(\Omega)$. Hence we define the linear mapping

$$R : \mathcal{F}_\Phi(\Omega) \to C(\Omega), \quad R(f)(x) := (f, \Phi(\cdot, \boldsymbol{x}))_\Phi, \quad \forall f \in \mathcal{F}_\Phi(\Omega), \forall \boldsymbol{x} \in \Omega.$$

The resulting functions are continuous since it holds that

$$|Rf(\boldsymbol{x}) - Rf(\boldsymbol{y})| = |(f, \Phi(\cdot, \boldsymbol{x}) - \Phi(\cdot, \boldsymbol{y}))_\Phi| \leq \|f\|_\Phi \|\Phi(\cdot, \boldsymbol{x}) - \Phi(\cdot, \boldsymbol{y})\|_\Phi$$

and

$$\|\Phi(\cdot,\boldsymbol{x}) - \Phi(\cdot,\boldsymbol{y})\|_\Phi^2 = \Phi(\boldsymbol{x},\boldsymbol{x}) + \Phi(\boldsymbol{y},\boldsymbol{y}) - 2\Phi(\boldsymbol{x},\boldsymbol{y}).$$

The desired result is obtained since Φ is continuous.

We need to conclude with a minor result, Lemma 4.3.7, to be able to define the native Hilbert space of the positive definite kernel Φ. Its proof can be found in [18, Lemma 10.8].

Lemma 4.3.7. *The linear mapping $R : \mathcal{F}_\Phi(\Omega) \to C(\Omega)$ is injective.*

Definition 4.3.8. The *native Hilbert function space* corresponding to the symmetric positive definite kernel $\Phi : \Omega \times \Omega \to \mathbb{R}$ is defined by

$$\mathcal{N}_\Phi(\Omega) := R(\mathcal{F}_\Phi(\Omega)).$$

The inner product is defined as

$$(f,g)_{\mathcal{N}_\Phi(\Omega)} := (R^{-1}f, R^{-1}g)_\Phi.$$

□

The space is a Hilbert space of continuous functions on Ω with reproducing kernel Φ. Since $\Phi(\cdot,\boldsymbol{x})$ is an element of $F_\Phi(\Omega)$ for $x \in \Omega$ it is unchanged under R, and therefore it holds that

$$f(\boldsymbol{x}) = (R^{-1}f, \Phi(\cdot,\boldsymbol{x}))_\Phi = (f, \Phi(\cdot,\boldsymbol{x}))_{\mathcal{N}_\Phi(\Omega)}, \text{ for all } f \in \mathcal{N}_\phi(\Omega) \text{ and all } \boldsymbol{x} \in \Omega.$$

Hence, positive (semi-)definite kernels and reproducing kernels of Hilbert function spaces are the same. The following theorem shows the uniqueness of the native space. Its proof can be found in [18, Thm. 10.11].

Theorem 4.3.9. *Assume that Φ is a symmetric positive definite kernel and assume further that $\mathcal{G}$ is a Hilbert space of functions $f : \Omega \to \mathbb{R}$ with reproducing kernel Φ. Then $\mathcal{G}$ is the native space and the inner products are the same.* □

We generalize the notion of conditionally positive definite function to a conditionally positive definite kernel.

Definition 4.3.10 (Conditionally positive definite kernel)**.** Assume that $\mathcal{P}$ is a finite-dimensional subspace of $C(\Omega)$, $\Omega \subseteq \mathbb{R}^d$. A continuous symmetric kernel $\Phi : \Omega \times \Omega \to \mathbb{R}$ is said to be *conditionally positive definite* on Ω with respect to $\mathcal{P}$ if, for any $n \in \mathbb{N}$, all sets of pairwise distinct points $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \in \Omega$ and all $\boldsymbol{\alpha} \in V_\mathcal{P} \backslash \{\boldsymbol{0}^n\}$ it holds that

$$\sum_{j=1}^{n}\sum_{k=1}^{n} \alpha_j \alpha_k \Phi(\boldsymbol{x}_j, \boldsymbol{x}_k) > 0,$$

where

$$V_\mathcal{P} = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^n : \sum_{j=1}^{n} \alpha_j p(\boldsymbol{x}_j) = 0, \quad p \in \mathcal{P} \right\}.$$

The domain Ω can be quite arbitrary. It should, however, contain at least one $\mathcal{P}$-unisolvent subset. Note that we use a more general space, $\mathcal{P}$. To establish the native space of a conditionally positive kernel, in the same manner as in the case of a positive definite kernel, we start by defining the linear space

$$F_\Phi(\Omega) := \left\{ \sum_{i=1}^{n} \alpha_i \Phi(\cdot, \boldsymbol{x}_j) : n \in \mathbb{N}, \boldsymbol{\alpha} \in \mathbb{R}^n, \boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \Omega, \text{ with } \sum_{i=1}^{n} \alpha_i p(\boldsymbol{x}_j) = 0,\ p \in \mathcal{P} \right\}. \quad (4.14)$$

The bilienear form presented in equation (4.13) can be used as an inner product. Note that the additional constraint on the coefficients in (4.14) ensures the definiteness of the inner product.

Again we can form the Hilbert-space completion $\mathcal{F}_\Phi(\Omega)$ of $F_\Phi(\Omega)$ with respect to $(\cdot,\cdot)_\Phi$. Unfortunately, we cannot construct an operator $R : \mathcal{F}_\Phi(\Omega) \to C(\Omega)$ in the same manner as before, because $\Phi(\cdot, \boldsymbol{x})$ is in general not included in $F_\Phi(\Omega)$. However, the construction of the operator R is still possible, while rather technical (see [18, Chapter 10.3]). The operator R is defined as

$$R : \mathcal{F}_\Phi(\Omega) \to C(\Omega), \quad R(f)(x) := (f, G(\cdot, \boldsymbol{x}))_\Phi, \quad \forall f \in \mathcal{F}_\Phi(\Omega), \forall \boldsymbol{x} \in \Omega,$$

where the function G is defined as

$$G(\cdot, \boldsymbol{x}) := \Phi(\cdot, \boldsymbol{x}) - \sum_{j=1}^{Q} p_j(\boldsymbol{x}) \Phi(\cdot, \boldsymbol{\xi}_j), \quad \boldsymbol{x} \in \Omega,$$

and where the points $\Xi = \{\xi_1, \dots, \xi_Q\}$ defines a $\mathcal{P}$-unisolvent subset of Ω with $Q = \dim(\mathcal{P})$ and $p_j,\ 1 \le j \le Q$, define a Lagrange basis of $\mathcal{P}$ with respect to Ξ. At last, a native space can be defined.

Definition 4.3.11. The native space corresponding to a symmetric kernel Φ that is conditionally positive definite on Ω with respect to $\mathcal{P}$ is defined by

$$\mathcal{N}_\Phi(\Omega) := R(\mathcal{F}_\Phi(\Omega)) + \mathcal{P}.$$

The space is equipped with a semi-inner product

$$(f, g)_{\mathcal{N}_\Phi(\Omega)} = (R^{-1}(f - \Pi_\mathcal{P} f), R^{-1}(g - \Pi_\mathcal{P} g))_\Phi, \quad (4.15)$$

where $\Pi_\mathcal{P}$ is a projection operator, defined as

$$\Pi_\mathcal{P} : C(\Omega) \to \mathcal{P}, \quad \Pi_\mathcal{P}(f) := \sum_{j=1}^{Q} f(\boldsymbol{\xi}_j) p_j.$$

□

Note that the conditionally positive definite kernel Φ is not a reproducing kernel for the native space $\mathcal{N}_\Phi(\Omega)$. Now, we are to find an error estimation of interpolating a function f, given the set of discrete points X. We need the following two definitions.

Definition 4.3.12 (Power function)**.** Assume that Φ is a conditionally positive definite kernel on an open set Ω with respect to $\mathcal{P} \subseteq C(\Omega)$. If $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \Omega$ is $\mathcal{P}$-unisolvent, then for every $\boldsymbol{x} \in \Omega$ the *power function* is defined by

$$P_{\Phi,X}(\boldsymbol{x})^2 := \Phi(\boldsymbol{x},\boldsymbol{x}) - 2\sum_{j=1}^{n} u_j^*(\boldsymbol{x})\Phi(\boldsymbol{x},\boldsymbol{x}_j) + \sum_{i,j=1}^{n} u_i^*(\boldsymbol{x})u_j^*(\boldsymbol{x})\Phi(\boldsymbol{x}_i,\boldsymbol{x}_j),$$

where the function $u^*(\boldsymbol{x})$ is a part of the solution to the system

$$\begin{pmatrix} A & P \\ P^T & 0^{Q\times Q} \end{pmatrix} \begin{pmatrix} u^*(\boldsymbol{x}) \\ v^*(\boldsymbol{x}) \end{pmatrix} = \begin{pmatrix} R(\boldsymbol{x}) \\ S(\boldsymbol{x}) \end{pmatrix},$$

where $A = (\Phi(\boldsymbol{x}_i,\boldsymbol{x}_j))$, $P = (p_j(\boldsymbol{x}_i))$, and $p_1, \ldots, p_Q$ form a basis of $\mathcal{P}$. Furthermore $R(\boldsymbol{x}) = (\Phi(\boldsymbol{x},\boldsymbol{x}_1), \ldots, \Phi(\boldsymbol{x},\boldsymbol{x}_n))^T$ and $S(\boldsymbol{x}) = (p_1(\boldsymbol{x}), \ldots, p_Q(\boldsymbol{x}))^T$. □

Definition 4.3.13 (Fill distance)**.** The *fill distance* of a set of points $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \Omega$ for a bounded domain $\Omega \subset \mathbb{R}^d$ is defined as

$$h := \sup_{\boldsymbol{x}\in\Omega} \min_{1\le j\le n} \|\boldsymbol{x} - \boldsymbol{x}_j\|.$$

□

An intuitive picture of Definition 4.3.13 is that for any point $\boldsymbol{x} \in \Omega$ there exists a data point $\boldsymbol{x}_j$ within a distance at most h. Another picture is that the fill distance h denotes the radius of the largest ball which is completely contained in Ω and which does not contain any data points $\boldsymbol{x}_j$, so in some sense h describes the largest data points-free "hole" in Ω. Now we can state an error estimation, which gives a bound for the interpolation error; the proof can be found in [18, Thm. 11.4].

Theorem 4.3.14 (Error estimation)**.** *Let $\Omega \subseteq \mathbb{R}^d$ be open. Assume that Φ is a conditionally positive definite kernel on Ω with respect to $\mathcal{P} \subseteq C(\Omega)$. Assume further that $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \Omega$ is $\mathcal{P}$-unisolvent. Denote the interpolant of $f \in \mathcal{N}_\phi(\Omega)$ by $s_{f,X}$. Then, for every $\boldsymbol{x} \in \Omega$ it holds that*

$$|f(\boldsymbol{x}) - s_{f,X}(\boldsymbol{x})| \le P_{\Phi,X}(\boldsymbol{x})|f|_{\mathcal{N}_\Phi(\Omega)}.$$

□

It is possible to find a bound for the power function. For every radial basis function there is a function B dependent on the fill distance h such that

$$P_{\Phi,X}^2(\boldsymbol{x}) \le CB(h), \quad \boldsymbol{x} \in \Omega$$

where $C > 0$ is a constant independent of $X := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. The radial basis functions presented in Section 4.2 and their corresponding functions B are listed in Table 4.2.

Table 4.2 shows that the Gaussian RBFs have the best asymptotic convergence while the linear RBF show a rather poor asymptotic behavior. This would be a motive for using a Guassian RBF but the error estimates are asymptotic and do not necessarily imply a better interpolation with a finite number of points.

Table 4.2: A bound for the power function, where $c > 0$ and $\tilde{c} > 0$ are constants.

Name	$\Phi(\boldsymbol{x}) = \phi(r), \quad r = \|\boldsymbol{x}\|$		B(h)
Guassians	$e^{-\alpha r^2},$	$\alpha > 0$	$e^{-c\lvert \log h\rvert /h}$
Multiquadrics	$(-1)^{\lceil \beta \rceil}(c^2 + r^2)^\beta,$	$\beta > 0, \beta \notin \mathbb{N}$	$e^{-\tilde{c}/h}$
Inverse multiquadrics	$(c^2 + r^2)^\beta,$	$\beta < 0$	$e^{-\tilde{c}/h}$
Thin-plate splines	$(-1)^{k+1} r^{2k} \log r,$	$k \in \mathbb{N}$	h^{2k}
Powers	$(-1)^{\lceil \beta/2 \rceil} r^\beta,$	$\beta > 0, \beta \notin 2\mathbb{N}$	h^β
Linear (special case of powers RBF)	$-r$		h
Cubic (special case of powers RBF)	r^3		h^3

Chapter 5

Robust design methodology

In Chapter 2 the tractable problem was stated—find a parameter setting in the fundamental algorithm which results in low collision speed in the positive performance scenarios, and at the same time, minimizes the risk for false intervention in the negative performance scenarios. The parameter setting has to be chosen in such a way that the fundamental algorithm is not sensitive to uncertainties within the defined range. In this chapter we present a robust design methodology which describes the procedure for finding a parameter setting described in the tractable problem.

5.1 Introduction

Whenever there are uncertainties in a problem, robust solutions are generally desired, since they are by definition not very sensitive to uncertainties. When a company produces a product it is important to make that product as insensitive to uncertainties as possible, because then the company can guarantee a certain level of performance of its product. In the tractable problem, see Definition 2.2.1, we want to find a parameter setting such that the fundamental algorithm, see Section 2.3, is not sensitive to uncertainties in the defined range, i.e., a robust parameter setting. However, we need to explicitly define what a robust parameter setting is for this problem. First, we say a robust parameter setting has to fulfill two types of robustness—robustness of negative performance and robustness of positive performance. This separation is necessary, because the two types of robustness serve different purposes and it is also convenient in the solution process. When a parameter setting fulfills both types of robustness it will be a sensible solution to the tractable problem.

Definition 5.1.1 (Robustness of negative performance)**.** We say that a parameter setting fulfills *robustness of negative performance* if it will not cause false intervention, even with the worst combination of errors, in any of the negative performance scenarios. □

A car that might fully break even though the driver has full control over the situation is not an attractive car on the market and for this reason we have zero tolerance for false intervention. Note that if a parameter setting fulfills robustness of negative performance then no false intervention will occur as long as the errors are within the assumed range, as defined in Section 2.5. In other words: the boundaries of the errors are crucial for the degree of robustness. It is important that

the boundaries capture the errors effectively. Too pessimistic boundaries result in bad positive performance and too optimistic boundaries result in higher risk for false intervention.

Definition 5.1.2 (Robustness of positive performance)**.** We say that a parameter setting fulfills *robustness of positive performance* if it fulfills robustness of negative performance and also guarantees as low collision speed as possible, in all the positive performance scenarios with the worst combination of errors, respectively. Moreover, the parameter setting should guarantee as small spread as possible of the collision speed in all the positive performance scenarios. □

5.2 Robustness of negative performance

In this section we will describe how to find the parameter settings that fulfill robustness of negative performance. First we introduce all the defined errors from Table 2.2 into all the relevant relations in the fundamental algorithm; see Section 2.3. After that, Section 5.2.1 describes the method for finding the worst combination of errors. We conclude with Section 5.2.2, which describes how to find the parameter settings that will not cause false intervention, even with the worst combination of errors in either of the two negative performance scenarios.

The tunable parameter $a_{\text{avail}}^{\text{long}}$, see Section 2.3, cannot affect the avoidance of false intervention in the negative performance scenarios. This is due to the fact that the relative longitudinal distance decreases to zero in both scenarios, so the required longitudinal acceleration to avoid collision goes to infinity; see (2.5). Since the highest value of $a_{\text{avail}}^{\text{long}}$ is 10, the BTN-condition will always be true regardless of $a_{\text{avail}}^{\text{long}}$. This implies that the only tunable parameters that can affect avoidance of false intervention are $a_{\text{avail}}^{\text{lat}}$ and y_{safe}. Moreover, the tunable parameter $a_{\text{avail}}^{\text{long}}$ has a different dependency than the others; see Table 2.1. As a result of all these factors we will analyze $a_{\text{avail}}^{\text{long}}$ in a unique way and develop a concept which coincides well with our view of robustness; see Section 5.3.

Both the tunable parameters $a_{\text{avail}}^{\text{lat}}$ and y_{safe} depend on the velocity of the host car, see Table 2.1. We choose different values on the velocity of the host car and then construct an optimization problem for each fixed velocity. When we have found the optimal values on $a_{\text{avail}}^{\text{lat}}$ for each fixed velocity we will interpolate these to construct a function that describes the value on $a_{\text{avail}}^{\text{lat}}$ for a certain velocity. Similarly, we will interpolate the optimal values of y_{safe}. We will present the result only for the fixed velocity of 60 km/h, since it is rather the methodology to achieve an optimal solution that is important.

We introduce the errors defined in Table 2.2 into all the relations from Section 2.3. Combine (2.1) and (2.5), together with the corresponding errors, yields the equivalence

$$a_{\text{req}}^{\text{long}} = a_{\text{tar}}^{\text{long}} + \xi_{a_{\text{tar}}^{\text{long}}} - \frac{\left(v_{\text{rel}}^{\text{long}} + \xi_{v_{\text{rel}}^{\text{long}}}\right)^2}{2\left(x_{\text{rel}} + \xi_{x_{\text{rel}}} - x_{\text{margin}} + t_{\text{pressure}}(a_{\text{h}}^{\text{long}}) \cdot (v_{\text{rel}}^{\text{long}} + \xi_{v_{\text{rel}}^{\text{long}}})\right)}. \tag{5.1}$$

Note that we assume that we measure the longitudinal acceleration of the host with insignificant measurement errors. The longitudinal errors also affect the relations (2.6) and (2.7), which yields

the equivalences

$$t = -\frac{v_{\text{rel}}^{\text{long}} + \xi_{v_{\text{rel}}^{\text{long}}}}{a_{\text{rel}}^{\text{long}} + \xi_{a_{\text{tar}}^{\text{long}}}} \pm \sqrt{\left(\frac{v_{\text{rel}}^{\text{long}} + \xi_{v_{\text{rel}}^{\text{long}}}}{a_{\text{rel}}^{\text{long}} + \xi_{a_{\text{tar}}^{\text{long}}}}\right)^2 - \frac{2(x_{\text{rel}} + \xi_{x_{\text{rel}}})}{a_{\text{rel}}^{\text{long}} + \xi_{a_{\text{tar}}^{\text{long}}}}} \tag{5.2}$$

and

$$t_{\text{ttc}} = -\frac{x_{\text{rel}} + \xi_{x_{\text{rel}}}}{v_{\text{rel}}^{\text{long}} + \xi_{v_{\text{rel}}^{\text{long}}}}, \tag{5.3}$$

respectively.

The lateral errors affect (2.8), so including the errors yields the equivalence

$$y_{\text{rel}}^{\text{pred}} = y_{\text{rel}} + \xi_{y_{\text{rel}}} + (v_{\text{rel}}^{\text{lat}} + \xi_{v_{\text{rel}}^{\text{lat}}}) \cdot t_{\text{ttc}} + \frac{(a_{\text{tar}}^{\text{lat}} + \xi_{a_{\text{tar}}^{\text{lat}}}) \cdot t_{\text{ttc}}^2}{2}. \tag{5.4}$$

Finally, (2.9) and (2.10) result in the equivalences

$$a_{\text{req}}^{\text{left}} = \frac{2y_{\text{rel}}^{\text{pred}} - (w_{\text{tar}} + \xi_{w_{\text{tar}}}) - w_{\text{h}} - 2y_{\text{safe}}}{t_{\text{ttc}}^2} \tag{5.5}$$

and

$$a_{\text{req}}^{\text{right}} = \frac{2y_{\text{rel}}^{\text{pred}} + (w_{\text{tar}} + \xi_{w_{\text{tar}}}) + w_{\text{h}} + 2y_{\text{safe}}}{t_{\text{ttc}}^2}. \tag{5.6}$$

5.2.1 Finding the worst combination of errors

We start by analyzing the errors in the first negative performance scenario, because in that scenario it is straight-forward what the worst combination of errors are in each time step. In the longitudinal direction all the errors, $\xi_{x_{\text{rel}}}$, $\xi_{v_{\text{rel}}^{long}}$ and $\xi_{a_{\text{tar}}^{\text{long}}}$, that cause the target car to appear closer to the host car are the errors that cause a higher risk to trigger a full break, because the required longitudinal acceleration appears to be higher. This means that the worst-case values of the longitudinal errors are $\xi_{x_{\text{rel}}} = -b_1$, $\xi_{v_{\text{rel}}^{\text{long}}} = -b_2$, and $\xi_{a_{\text{tar}}^{\text{long}}} = -b_3$. The host car drives past the target car on the right side, i.e., in the positive y-direction. We conclude that the worst lateral errors, $\xi_{y_{\text{rel}}}, \xi_{v_{\text{rel}}^{\text{lat}}}$, and $\xi_{a_{\text{tar}}^{\text{lat}}}$, are those that make the target car appear to the right of its true position, i.e., in the path of the host car. This means that the worst-case values are $\xi_{y_{\text{rel}}} = b_4$, $\xi_{v_{\text{rel}}^{\text{lat}}} = b_5$, and $\xi_{a_{\text{tar}}^{\text{lat}}} = b_6$. Finally, the wider the target car appears to be the more likely a false intervention will be, so we have $\xi_{w_{\text{tar}}} = b_7$.

In the second negative performance scenario we can make a similar analysis of the errors in the longitudinal direction and regarding the width of the target car. In the lateral direction it is not equally straight-forward. However, we know that the higher values on the required lateral acceleration the more likely the car is to trigger a full break. The lateral errors can only affect (5.4), which in turn affects the required lateral acceleration for steering either left or right. From (2.11) follows that high magnitudes of $a_{\text{req}}^{\text{left}}$ and $a_{\text{req}}^{\text{right}}$ result in high required lateral acceleration, unless

$a_{\text{req}}^{\text{left}}$ and $a_{\text{req}}^{\text{right}}$ have equal sign, as discussed in Section 2.3. This implies that the most problematic value the lateral prediction $y_{\text{rel}}^{\text{pred}}$ can take is zero, i.e., when the target car is completely in the way of the host car.

The closer $y_{\text{rel}}^{\text{pred}}$ is to zero, the higher the risk is to trigger a full break. Therefore, we want to find the lateral errors that cause $y_{\text{rel}}^{\text{pred}}$ to be as close to zero as possible. In each time step of the simulation of negative performance scenario 2 we are given the values of the parameters y_{rel}, $v_{\text{rel}}^{\text{lat}}$, $a_{\text{tar}}^{\text{lat}}$ and t_{ttc} and we want to find the values on $\xi_{y_{\text{rel}}}$, $\xi_{v_{\text{rel}}^{\text{lat}}}$ and $\xi_{a_{\text{tar}}^{\text{lat}}}$ such that $y_{\text{rel}}^{\text{pred}}$ is as close to zero as possible. This results in the optimization problem is to

$$\begin{aligned} \text{minimize } \left|y_{\text{rel}}^{\text{pred}}\right| = & \left| y_{\text{rel}} + \xi_{y_{\text{rel}}} + (v_{\text{rel}}^{\text{lat}} + \xi_{v_{\text{rel}}^{\text{lat}}}) \cdot t_{\text{ttc}} + \frac{(a_{\text{tar}}^{\text{lat}} + \xi_{a_{\text{tar}}^{\text{lat}}}) \cdot t_{\text{ttc}}^2}{2} \right| \\ s.t. \quad & -b_4 \leq \xi_{y_{\text{rel}}} \leq b_4, \\ & -b_5 \leq \xi_{v_{\text{rel}}^{\text{lat}}} \leq b_5, \\ & -b_6 \leq \xi_{a_{\text{tar}}^{\text{lat}}} \leq b_6. \end{aligned} \tag{5.7}$$

We can formulate the problem (5.7) in a more general setting, which is beneficial if more errors affecting y_{rel}^{pred} would be introduced. The general optimization problem is to

$$\text{minimize } f(\boldsymbol{\zeta}) := |c_0 + c_1\zeta_1 + \ldots + c_n\zeta_n| = |c_0 + \boldsymbol{c}^T\boldsymbol{\zeta}| \tag{5.8a}$$

$$s.t. \quad -\beta_i \leq \zeta_i \leq \beta_i, \quad \text{for } i = 1, \ldots, n, \tag{5.8b}$$

$$\boldsymbol{\zeta} \in \mathbb{R}^n, \tag{5.8c}$$

where $c_0, \ldots, c_n$ and $\beta_1, \ldots, \beta_n$ are given constants. With the notation in (5.8), (5.7) can be reformulated with $c_0 = y_{\text{rel}} + v_{\text{rel}}^{\text{lat}} \cdot t_{\text{ttc}} + a_{\text{tar}}^{\text{lat}} \cdot t_{\text{ttc}}^2/2$, $c_1 = 1$, $c_2 = t_{\text{ttc}}$, $c_3 = t_{\text{ttc}}^2/2$, $\beta_1 = b_4$, $\beta_2 = b_5$, and $\beta_3 = b_6$. The function $f(\boldsymbol{\zeta})$ is convex on $\mathbb{R}^n$, due to the triangle inequality, and the constraints (5.8b) are affine. Hence, (5.8) defines convex optimization problem. From Theorem 3.1.5 we know that every local minimum of a convex function f over a convex feasible set is also a global minimum. In other words, local search methods are applicable to (5.8). In fact, it can even be solved to optimum by a greedy algorithm[1] that we have developed; see Algorithm 2.

If $c_0 < 0$ make the replacement specified in Algorithm 2. We explain the convergence of Algorithm 2 if $c_0 \geq 0$ (similar analysis for $c_0 < 0$). The idea is to first set $\boldsymbol{\zeta} := \mathbf{0}^n$ and then check if c_0 is equal to zero. If this is the case, $\boldsymbol{\zeta} = \mathbf{0}^n$ is an optimal solution, since $f(\mathbf{0}) = 0$ and f is a nonnegative function. If $c_0 > 0$ we construct a vector $\boldsymbol{p}$ which is axis parallel to the ζ_i-axis where i corresponds to the index of the c_i with the highest magnitude. We let $\boldsymbol{p} := -\text{sign}(c_i) \cdot \text{sign}(c_0) \cdot \beta_i \cdot \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is i-th standard vector in the standard basis for $\mathbb{R}^n$. In other words, if we add $\boldsymbol{p}$ to $\boldsymbol{\zeta}$ we move as far as we can in the ζ_i-axis direction such that we still stay feasible. Now we check whether $c_0 + \boldsymbol{c}^T(\boldsymbol{\zeta} + \boldsymbol{p}) \leq 0$ or not. If it is not less than or equal to 0 we update $\boldsymbol{\zeta}$ to be $\boldsymbol{\zeta} + \boldsymbol{p}$ and go back to reconstruct $\boldsymbol{p}$, but in another axis parallel direction that not have been chosen earlier, if there are any, otherwise the algorithm will terminate. However, if $c_0 + \boldsymbol{c}^T(\boldsymbol{\zeta} + \boldsymbol{p}) \leq 0$ we update $\boldsymbol{\zeta}$ to be $\boldsymbol{\zeta} + \frac{-c_0 - \boldsymbol{c}^T\boldsymbol{\zeta}}{\boldsymbol{c}^T\boldsymbol{p}}\boldsymbol{p}$ and the algorithm will terminate. This means that there are two possible outcomes

[1] An algorithm that is generally very simple in its structure. The general idea is to find what is "locally" best in each iteration ([19]).

Algorithm 2 Find worst errors, i.e, to solve the optimization problem (5.8)

Step 0:
Let $\mathcal{I} := \{1, \ldots, n\}$ be the index set, where $n > 0$
Let $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ be the standard basis for $\mathbb{R}^n$
Let $c_0, \ldots, c_n$ and $\beta_1, \ldots, \beta_n$ be given constants
Let $\boldsymbol{\zeta} := \mathbf{0}^n$

Step 1:
if $c_0 = 0$
Proceed to **Step 5**
else
Proceed to **Step 2**
end if

Step 2:
Choose one $i \in \underset{k \in \mathcal{I}}{\operatorname{argmax}} |c_k|$
Let $\boldsymbol{p} := -\operatorname{sign}(c_i) \cdot \operatorname{sign}(c_0) \cdot \beta_i \cdot \boldsymbol{e}_i$

Step 3:
if $c_0 + \boldsymbol{c}^T(\boldsymbol{\zeta} + \boldsymbol{p}) \leq 0$ (replace "$\leq$" with "$\geq$" if $c_0 < 0$)
Proceed to **Step 4**
else
Update $\boldsymbol{\zeta} \leftarrow \boldsymbol{\zeta} + \boldsymbol{p}$
Update $\mathcal{I} \leftarrow \mathcal{I} \backslash \{i\}$
if $|\mathcal{I}| > 0$
Return to **Step 2**
else
Proceed to **Step 5**
end if
end if

Step 4:
Update $\boldsymbol{\zeta} \leftarrow \boldsymbol{\zeta} + \dfrac{-c_0 - \boldsymbol{c}^T\boldsymbol{\zeta}}{\boldsymbol{c}^T\boldsymbol{p}}\boldsymbol{p}$

Step 5:
Let $\boldsymbol{\zeta}^* := \boldsymbol{\zeta}$. The vector $\boldsymbol{\zeta}^*$ is an optimal solution to problem (5.8)

from Algorithm 2. The first one is that $c_0 + \boldsymbol{c}^T(\boldsymbol{\zeta} + \boldsymbol{p})$ never gets less than or equal to zero in all the n iterations, i.e., we update $\boldsymbol{\zeta}$ in all the axis parallel directions. This means that the outcome vector is

$$\boldsymbol{\zeta}^* = (-\operatorname{sign}(c_1) \cdot \operatorname{sign}(c_0) \cdot \beta_1), \ldots, -\operatorname{sign}(c_n) \cdot \operatorname{sign}(c_0) \cdot \beta_n)^T. \tag{5.9}$$

From Theorem 3.1.6 it follows that $\boldsymbol{\zeta}^*$ described in (5.9) is an optimal solution to the optimization

problem (5.8), since

$$\nabla f(\boldsymbol{\zeta}) = \begin{cases} \boldsymbol{c}, & \text{if } c_0 + \boldsymbol{c}^T\boldsymbol{\zeta} \geq 0, \\ -\boldsymbol{c}, & \text{otherwise,} \end{cases}$$

and $c_0 + \boldsymbol{c}^T\boldsymbol{\zeta}^* > 0$, so $\nabla f(\boldsymbol{\zeta}^*) = \boldsymbol{c}$. Furthermore,

$$\nabla f(\boldsymbol{\zeta}^*)^T(\boldsymbol{\zeta} - \boldsymbol{\zeta}^*) = \sum_{k=1}^{n} c_k(\zeta_k + sign(c_k)\cdot\beta_k) = \sum_{k=1}^{n} |c_k| \underbrace{(sign(c_k)\cdot\zeta_k + \beta_k)}_{\geq 0} \geq 0,$$
$$\forall \boldsymbol{\zeta} \in \Omega = \{\boldsymbol{\zeta} \in \mathbb{R}^n : -\beta_i \leq \zeta_i \leq \beta_i,\ i = 1, \ldots, n\}.$$

The second possible outcome is that $c_0 + \boldsymbol{c}^T(\boldsymbol{\zeta}+\boldsymbol{p}) \leq 0$ at a certain iteration. This means $c_0 + \boldsymbol{c}^T\boldsymbol{\zeta} > 0$, otherwise $c_0 + \boldsymbol{c}^T(\boldsymbol{\zeta}+\boldsymbol{p}) \leq 0$ at the previous iteration. Therefore, by choosing an appropriate factor, namely $\frac{-c_0 - \boldsymbol{c}^T\boldsymbol{\zeta}}{\boldsymbol{c}^T\boldsymbol{p}}$, to $\boldsymbol{p}$ it follows that $f(\boldsymbol{\zeta} + \frac{-c_0-\boldsymbol{c}^T\boldsymbol{\zeta}}{\boldsymbol{c}^T\boldsymbol{p}}\boldsymbol{p}) = |c_0 + \boldsymbol{c}^T(\boldsymbol{\zeta} + \frac{-c_0-\boldsymbol{c}^T\boldsymbol{\zeta}}{\boldsymbol{c}^T\boldsymbol{p}}\boldsymbol{p})| = 0$, which is an optimal solution to the optimization problem (5.8) since f is nonnegative.

5.2.2 Finding the trigger edge

We have developed a procedure to find the worst combination of errors in each time step of the negative performance scenario simulations. Now we wish to find all combinations of y_{safe} and $a^{\text{lat}}_{\text{avail}}$ that don not cause a false intervention given these errors. The approach is to find a relation between y_{safe} and $a^{\text{lat}}_{\text{avail}}$ such that, for a given value of y_{safe} we compute what value $a^{\text{lat}}_{\text{avail}}$ needs to have to avoid false intervention.

If no trigger should occur in the negative performance scenarios then the maximum available lateral acceleration, $a^{\text{lat}}_{\text{avail}}$, has to be higher than or equal to the required lateral acceleration, $a^{\text{lat}}_{\text{req}}$, at each time steps. This implies that the lowest value that $a^{\text{lat}}_{\text{avail}}$ can take is $a^{\text{lat}}_{\text{avail}} = a^{\text{lat}}_{\text{req}}$. From (2.11) follows that the equivalence

$$\begin{aligned} a^{\text{lat}}_{\text{avail}} = a^{\text{lat}}_{\text{req}} &= \min\left(\left|a^{\text{lat}}_{\text{left}}\right|, \left|a^{\text{lat}}_{\text{right}}\right|\right) \\ &= \min\left(\underbrace{\left|\frac{2y^{\text{pred}}_{\text{rel}} - (w_{\text{tar}} + \xi_{w_{\text{tar}}}) - w_{\text{h}} - 2y_{\text{safe}}}{t^2_{\text{ttc}}}\right|}_{=(1)}, \underbrace{\left|\frac{2y^{\text{pred}}_{\text{rel}} + (w_{\text{tar}} + \xi_{w_{\text{tar}}}) + w_{\text{h}} + 2y_{\text{safe}}}{t^2_{\text{ttc}}}\right|}_{=(2)}\right) \end{aligned} \tag{5.10}$$

hold as long as $\text{sign}(a^{\text{lat}}_{\text{left}}) \neq \text{sign}(a^{\text{lat}}_{\text{right}})$. So, in turn,

$$(1) = \left|\underbrace{\frac{2y^{\text{pred}}_{\text{rel}} - (w_{\text{tar}} + \xi_{w_{\text{tar}}}) - w_{\text{h}}}{t^2_{\text{ttc}}}}_{=m_1} + \underbrace{\frac{-2}{t^2_{\text{ttc}}}}_{=k_1} \cdot y_{\text{safe}}\right| =: |m_1 + k_1 \cdot y_{\text{safe}}|, \tag{5.11}$$

and

$$(2) = \left| \underbrace{\frac{2y_{\text{rel}}^{\text{pred}} + (w_{\text{tar}} + \xi_{w_{\text{tar}}}) + w_{\text{h}}}{t_{\text{ttc}}^2}}_{=m_2} + \underbrace{\frac{2}{t_{\text{ttc}}^2}}_{=k_2} \cdot y_{\text{safe}} \right| =: |m_2 + k_2 \cdot y_{\text{safe}}|. \tag{5.12}$$

We are clearly only interested in situations when $t_{\text{ttc}} < \infty$, since otherwise there is no threat. Note that $k_1 = -k_2$. We can conclude that the inequalities $k_1 < 0$ and $k_2 > 0$ hold. It must hold that

$$|m_1 + k_1 \cdot y_{\text{safe}}| = \begin{cases} -m_1 - k_1 \cdot y_{\text{safe}}, & \text{if } y_{\text{safe}} \geq \dfrac{-m_1}{k_1} =: t_1, \\ m_1 + k_1 \cdot y_{\text{safe}}, & \text{if } y_{\text{safe}} < t_1, \end{cases}$$

and

$$|m_2 + k_2 \cdot y_{\text{safe}}| = \begin{cases} m_2 + k_2 \cdot y_{\text{safe}}, & \text{if } y_{\text{safe}} \geq \dfrac{-m_2}{k_2} =: t_2, \\ -m_2 - k_2 \cdot y_{\text{safe}}, & \text{if } y_{\text{safe}} < t_2. \end{cases}$$

In each time step we are given m_1, m_2, k_1 and k_2. So depending on the values of t_1 and t_2 we get different affine functions. However, we need to remember that if the inequalities

$$\begin{cases} \dfrac{2y_{\text{rel}}^{\text{pred}} - (w_{\text{tar}} + \xi_{w_{\text{tar}}}) - w_{\text{h}} - 2y_{\text{safe}}}{t_{\text{ttc}}^2} < 0 & (5.13) \\ \dfrac{2y_{\text{rel}}^{\text{pred}} + (w_{\text{tar}} + \xi_{w_{\text{tar}}}) + w_{\text{h}} + 2y_{\text{safe}}}{t_{\text{ttc}}^2} < 0 & (5.14) \end{cases}$$

hold or the inequalities

$$\begin{cases} \dfrac{2y_{\text{rel}}^{\text{pred}} - (w_{\text{tar}} + \xi_{w_{\text{tar}}}) - w_{\text{h}} - 2y_{\text{safe}}}{t_{\text{ttc}}^2} > 0 & (5.15) \\ \dfrac{2y_{\text{rel}}^{\text{pred}} + (w_{\text{tar}} + \xi_{w_{\text{tar}}}) + w_{\text{h}} + 2y_{\text{safe}}}{t_{\text{ttc}}^2} > 0 & (5.16) \end{cases}$$

hold, then $a_{\text{req}}^{\text{lat}}$ is not computed as in (5.10), and it will be equal to zero by definition and therefore any value on $a_{\text{avail}}^{\text{lat}}$ is considered valid to avoid false intervention. As long as

$$y_{\text{safe}} \in \mathcal{B}_1 := \left(y_{\text{rel}}^{\text{pred}} - \frac{w_{\text{tar}} + \xi_{w_{\text{tar}}} + w_{\text{h}}}{2}, -y_{\text{rel}}^{\text{pred}} - \frac{w_{\text{tar}} + \xi_{w_{\text{tar}}} + w_{\text{h}}}{2} \right)$$

the inequalities (5.13) and (5.14) hold. Note that if $y_{\text{rel}}^{\text{pred}} \geq 0$, then $\mathcal{B}_1 = \emptyset$. Furthermore, as long as

$$y_{\text{safe}} \in \mathcal{B}_2 := \left(-y_{\text{rel}}^{\text{pred}} - \frac{w_{\text{tar}} + \xi_{w_{\text{tar}}} + w_{\text{h}}}{2}, y_{\text{rel}}^{\text{pred}} - \frac{w_{\text{tar}} + \xi_{w_{\text{tar}}} + w_{\text{h}}}{2} \right)$$

the inequalities (5.15) and (5.16). Note that if $y_{\text{rel}}^{pred} \leq 0$, then $\mathcal{B}_2 = \emptyset$. Thus, we can conclude that (5.10) is valid as long as

$$y_{\text{safe}} \in \mathcal{B} := [-1, 0] \backslash (\mathcal{B}_1 \cup \mathcal{B}_2). \tag{5.17}$$

In Algorithm 3, see Appendix A, we generate all the affine constraints that describe the relation between $a_{\text{avail}}^{\text{lat}}$ and y_{safe} which do not cause a false intervention for a certain time step. Note that we only use Algorithm 3 when the BTN-condition is true. If the BTN-condition is not true, the host car will not fully break since not both conditions can be true, wherefore the algorithm is not needed. Algorithm 3 is based on case analysis; the different cases that can occur from (5.11) and (5.12) for different values on y_{safe}.

Figures 5.1 and 5.2 show the generated constraints from all time steps in the negative performance scenarios 1 and 2 having the worst combination of errors. All redundant constraints from each scenario have been removed. By collecting all the constraints from both negative performance scenarios we can conclude that all the constraints in the first negative performance scenario are redundant, because the constraints from the second negative performance scenario possess a higher restriction on the tunable parameters y_{safe} and $a_{\text{avail}}^{\text{lat}}$. Figure 5.2 show that the constraints form an edge, such that every parameter setting that lies on or above the edge will not cause a false intervention in any of the negative performance scenarios. We call this the *trigger edge*. Furthermore, all constraints that are not defined on the whole interval [-1,0] intersect with the y_{safe}-axis, i.e., $a_{\text{avail}}^{\text{lat}} = 0$, on the interval [-1,0]. This means that we can extend these constraints to the whole interval [-1,0], since they are affine and will not cause any more restrictions. Actually, this will always be the case, since in each time step—regardless of the values on y_{safe}—it is always better or equally good to steer either in the right or the left direction.

We know that the lower the value on $a_{\text{avail}}^{\text{lat}}$ is, and the higher the value on y_{safe} is, the earlier the car breaks. We can therefore conclude that the best solutions to the positive performance scenarios will lie on the trigger edge, since we want the car to break as early as possible. The trigger edge can easily be described as a function, dependent on y_{safe}, of the constraints that contribute to the edge; we denote this function by f_{edge}. In Algorithm 4 (see Appendix A) we locate all the constraints that contribute to the trigger edge and then construct the function $f_{\text{edge}}(y_{\text{safe}})$. The idea behind this algorithm is to start in $y_{\text{safe}} = -1$ and find the constraint that restricts $a_{\text{avail}}^{\text{lat}}$ the most. Then we follow this affine constraint until we reach an intersection point with another constraint, whence we follow that constraint and so on. Eventually, it holds that $a_{\text{avail}}^{\text{lat}} > 10$ or $y_{\text{safe}} > 0$, which means that all the relevant constraints are found. Note that all the constraints from the negative performance scenario 1 and 2 are used to create the trigger edge. Figure 5.3 illustrates the function f_{edge} created from Algorithm 4.

By creating the function f_{edge} the variable space is reduced by one dimension, since there is a one-to-one correspondence between $a_{\text{avail}}^{\text{lat}}$ and y_{safe}. For a certain value on y_{safe} it is known what value $a_{\text{avail}}^{\text{lat}}$ should take to be as good as possible. Figure 5.3 shows that y_{safe} needs to be lower than a certain value in order to avoid a false intervention. That certain value can be obtained by solving the equation $f_{\text{edge}}(y_{\text{safe}}) = 10$.

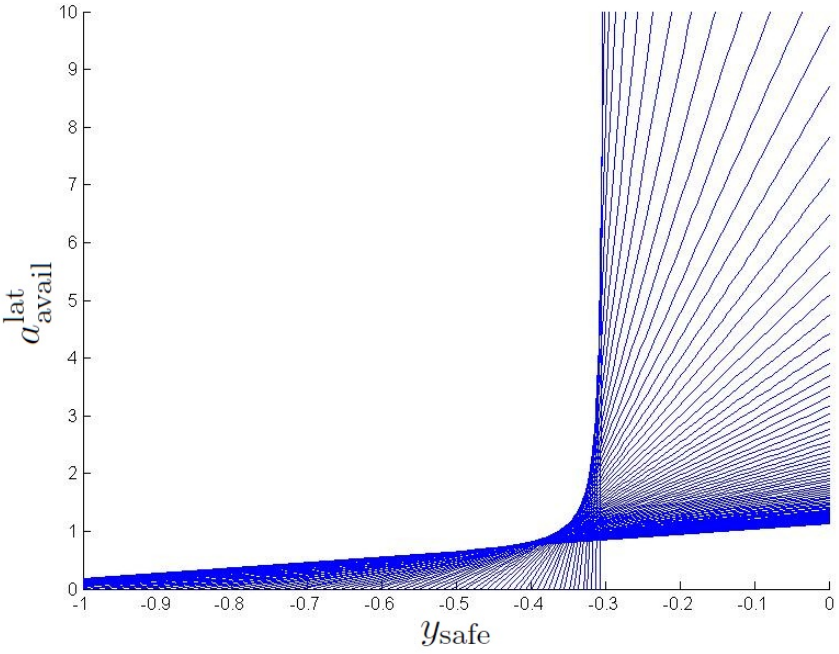

Figure 5.1: The constraints for negative performance scenario 1 generated from Algorithm 3.

5.3 Analysis of maximum available longitudinal acceleration

In Section 5.2 we found that $a^{\text{long}}_{\text{avail}}$ could not affect whether false intervention occurred or not. Therefore it is more interesting to view it from a positive performance perspective. In other words we will now work with positive performance scenarios 1, 2, and 3.

Under ideal conditions, when there are no uncertainties from the sensors, we would set $a^{\text{long}}_{\text{avail}} := -10 + \varepsilon$ for all x_{rel}, where ε is a small positive number. This is because the acceleration when the car fully breaks is $-10\ m/s^2$. However, we cannot set $a^{\text{long}}_{\text{avail}} := -10$, because then the T_{BTN} would not be strictly higher than 1 if $a^{\text{long}}_{\text{req}} = -10$, and therefore the car would not fully break, see Section 2.3. The small positive number ε compensate that problem and makes sure that the car fully breaks if $a^{\text{long}}_{\text{req}} = -10$. So, the host car avoids collision by breaking. Unfortunately, in reality conditions are rarely ideal. Therefore we introduce all the longitudinal errors from Table 2.2. Our definition of robustness in this case is that collision should be avoided when the measurements from the sensor are uncertain.

We let $a^{\text{long}}_{\text{req}} = -10$. From (2.5) and (2.1) we have that

$$-10 = a^{\text{long}}_{\text{tar}} - \frac{(v^{\text{long}}_{\text{rel}})^2}{2\left(x_{\text{rel}} - x_{\text{margin}} + t_{\text{pressure}}(a^{\text{long}}_{\text{h}}) \cdot v^{\text{long}}_{\text{rel}}\right)},$$

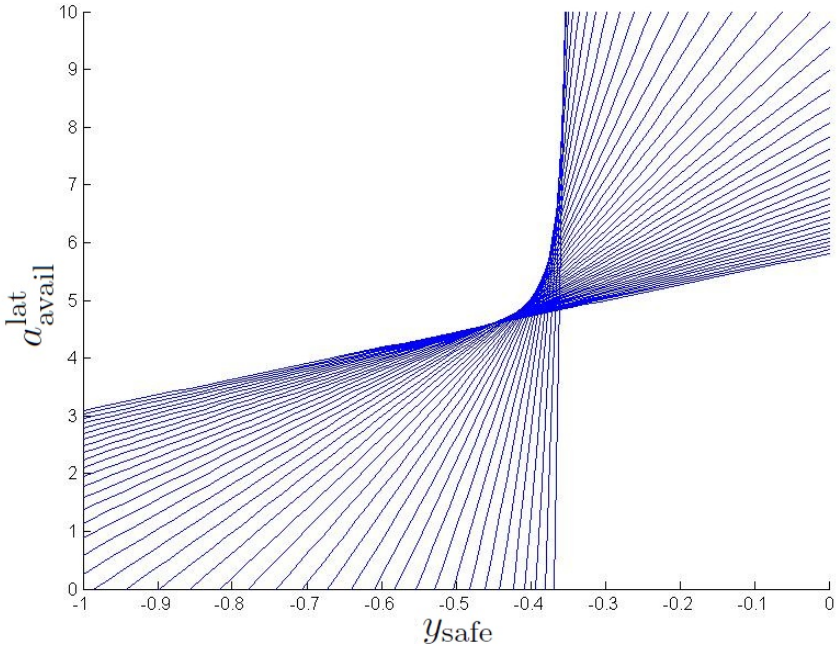

Figure 5.2: The constraints for negative performance scenario 2 generated from Algorithm 3.

which can be rewritten as

$$v_{\text{rel}}^{\text{long}} = (a_{\text{tar}}^{\text{long}} + 10)t_{\text{pressure}}(a_{\text{h}}^{\text{long}}) \pm \sqrt{(a_{\text{tar}}^{\text{long}} + 10)^2 t_{\text{pressure}}(a_{\text{h}}^{\text{long}})^2 + 2(a_{\text{tar}}^{\text{long}} + 10)(x_{\text{rel}} - x_{\text{margin}})}.$$

The expression under the square root will always be nonnegative because we are only interested in finding optimal values for $a_{\text{avail}}^{\text{long}}$ for $x_{\text{rel}} \geq 1$. The reason for not considering cases where $x_{\text{rel}} < 1$ is that it only concerns low velocity cases, which include factors that are not to be analyzed in this thesis. We are interested in the case when $v_{\text{rel}}^{\text{long}}$ is the most negative because that means that the host car drives faster than the target car (or that the target car is reversing towards the host car), as in the positive performance scenarios. Thus, we can conclude that

$$v_{\text{rel}}^{\text{long}} = (a_{\text{tar}}^{\text{long}} + 10)t_{\text{pressure}}(a_{\text{h}}^{\text{long}}) - \sqrt{(a_{\text{tar}}^{\text{long}} + 10)^2 t_{\text{pressure}}(a_{\text{h}}^{\text{long}})^2 + 2(a_{\text{tar}}^{\text{long}} + 10)(x_{\text{rel}} - x_{\text{margin}})}. \tag{5.18}$$

Replacing $v_{\text{rel}}^{\text{long}}$ in (5.1) by $v_{\text{rel}}^{\text{long}}$ from (5.18) it yields that

$$a_{\text{req}}^{\text{long}} = a_{\text{tar}}^{\text{long}} + \xi_{a_{\text{tar}}^{\text{long}}} - \frac{\left(B \cdot t(a_{\text{h}}^{\text{long}}) - \sqrt{B^2 t(a_{\text{h}}^{\text{long}})^2 + 2BC} + \xi_{v_{\text{rel}}^{\text{long}}}\right)^2}{2\left(C + \xi_{x_{\text{rel}}} + t(a_{\text{h}}^{\text{long}}) \cdot (B \cdot t(a_{\text{h}}^{\text{long}}) - \sqrt{B^2 t(a_{\text{h}}^{\text{long}})^2 + 2BC} + \xi_{v_{\text{rel}}^{\text{long}}})\right)}, \tag{5.19}$$

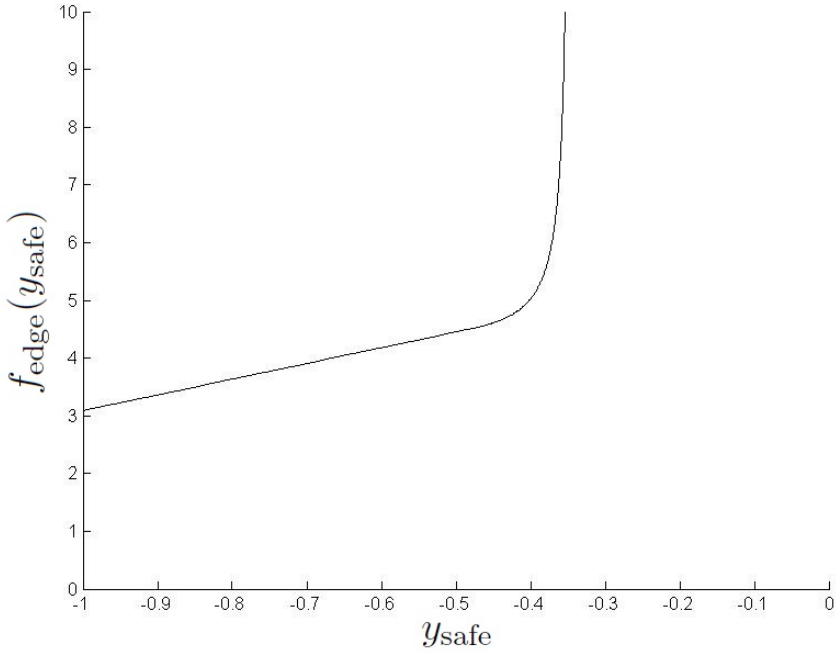

Figure 5.3: The function f_{edge}, the trigger edge, established by Algorithm 4.

where t denotes t_{pressure}, $B = a_{\text{tar}}^{\text{long}} + 10$ and $C = x_{\text{rel}} - x_{\text{margin}}$. We have now found an expression to describe the required longitudinal acceleration dependent on $a_{\text{tar}}^{\text{long}}$ and x_{rel}. In the positive performance scenarios the longitudinal acceleration of the target car, $a_{\text{tar}}^{\text{long}}$, is either -10 or 0. Let $a_{\text{tar}}^{\text{long}} \in [-10, 0]$. Then the value on $a_{\text{tar}}^{\text{long}}$ that causes $a_{\text{req}}^{\text{long}}$ to be as high as possible is when $a_{\text{tar}}^{\text{long}} = 0$ for each fixed value on $x_{rel} \geq 1$, which can be verified by the derivative of (5.19). The most unfavorable errors are those that make the target car appear further away from the host car, which correspond to the parameter values $\xi_{x_{\text{rel}}} = b_1$, $\xi_{v_{\text{rel}}^{\text{long}}} = b_2$, and $\xi_{a_{\text{tar}}^{\text{long}}} = b_3$. We can therefore define $a_{\text{avail}}^{\text{long}}(x_{\text{rel}})$ as

$$a_{\text{avail}}^{\text{long}}(x_{\text{rel}}) := b_3 - \frac{\left(10 \cdot t(a_{\text{h}}^{\text{long}}) - \sqrt{10^2 t(a_{\text{h}}^{\text{long}})^2 + 20C} + b_2\right)^2}{2\left(C + b_1 + t(a_{\text{h}}^{\text{long}}) \cdot (10t(a_{\text{h}}^{\text{long}}) - \sqrt{10^2 t(a_{\text{h}}^{\text{long}})^2 + 20C} + b_2)\right)} + \varepsilon, \qquad (5.20)$$

where t denotes t_{pressure}, $C = x_{\text{rel}} - x_{\text{margin}}$ and ε is a small positive number. The expression (5.20) describes the required value of $a_{\text{avail}}^{\text{long}}$ to avoid collision, given a certain relative distance, x_{rel}. These values are illustrated in Figure 5.4. From a robustness point of view this solution is considered to be the optimal solution, since it allows for handling unfavorable errors such that the host car avoids collision.

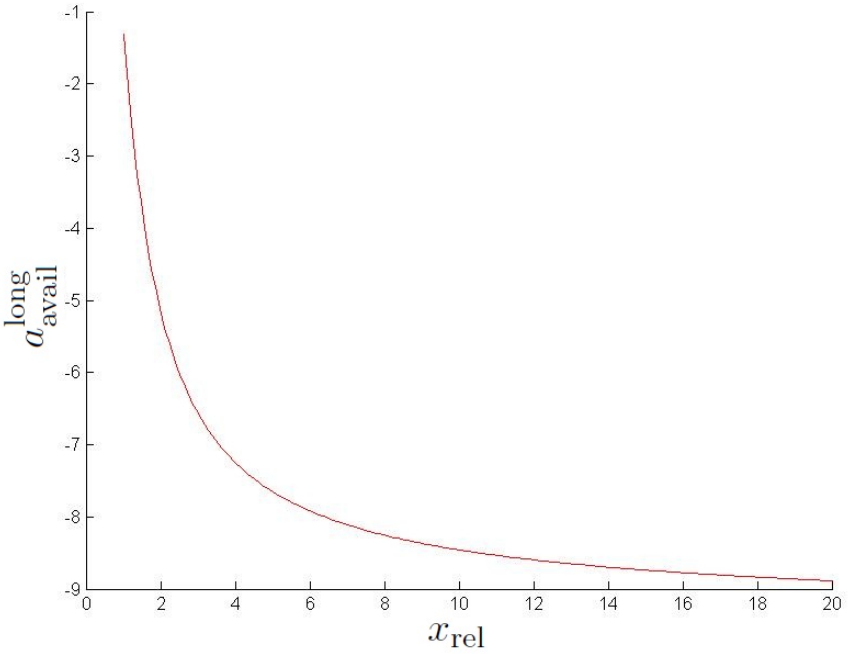

Figure 5.4: The optimal values on the tunable parameter $a^{\text{long}}_{\text{avail}}$ as a function of x_{rel}.

5.4 Robustness of positive performance

In Section 5.2 we computed the trigger edge where no false intervention occur in either of negative performance scenario one or two. In this section we describe how to find the solutions that fulfill robustness of positive performance.

The objective functions are created through response surfaces. This means that for each of the three positive performance scenarios we simulate them a certain number of times n_j, $j \in \{1, 2, 3\}$, with a different parameter settings from the trigger edge. From each simulation is then collected the collision speed with the most unfavorable errors and the spread of the collision speed. With these data points six linear systems of equations are solved, where a linear system is described in (4.9), and six response surfaces are constructed. In other words, choose n_j distinct points $X_j := \{(y_{\text{safe}})_1, \ldots, (y_{\text{safe}})_{n_j}\}$ and the other parameter $a^{\text{lat}}_{\text{avail}}$ is automatically given for each point, since the parameter settings should lie on the trigger edge, i.e, $(a^{\text{lat}}_{\text{avail}})_i = f_{\text{edge}}((y_{\text{safe}})_i)$. Then a scenario of the positive performance scenarios is simulated, for instance positive performance scenario one, with all chosen parameter settings, i.e., $(y_{\text{safe}})_i \in X_1,\ i \in \{1, \ldots, n_1\}$. For each parameter setting the collision speed is achieved, f^1_i, with the worst combination of errors and the spread of the collision speed, f^2_i. From this two linear systems of equations can be solved, in (4.9), and $s^1_{f^1,X_1}$ and $s^2_{f^2,X_1}$

are obtained, where the exponents 1 and 2 are indices. So, we have the following response surfaces:

$s^1_{f^1,X_1}$, the collision speed for positive performance scenario 1 with the most unfavorable errors.
$s^2_{f^2,X_1}$, the spread of the collision speed for positive performance scenario 1 due to errors.
$s^3_{f^3,X_2}$, the collision speed for positive performance scenario 2 with the most unfavorable errors.
$s^4_{f^4,X_2}$, the spread of the collision speed for positive performance scenario 2 due to errors.
$s^5_{f^5,X_3}$, the collision speed for positive performance scenario 3 with the most unfavorable errors.
$s^6_{f^6,X_3}$, the spread of the collision speed for positive performance scenario 3 due to errors.

We can now present the multi-objective surrogate optimization problem:

$$\text{minimize} \quad \{s^1_{f^1,X_1}(y_{\text{safe}}), s^2_{f^2,X_1}(y_{\text{safe}}), s^3_{f^3,X_2}(y_{\text{safe}}), s^4_{f^4,X_2}(y_{\text{safe}}), s^5_{f^5,X_3}(y_{\text{safe}}), s^6_{f^6,X_3}(y_{\text{safe}})\} \tag{5.21a}$$

$$\text{subject to} \quad -1 \leq y_{\text{safe}} \leq u, \tag{5.21b}$$

where $u \in \{a \in [-1, 0] : f_{\text{edge}}(a) = 10\}$. Note that the problem (5.21) has been reduced to a 1-dimensional six-objective optimization problem by utilizing the function f_{edge}.

In Section 4.3 we established that given only finite number of points it cannot be determined which radial basis function that is the most efficient. Therefore, we experimented with obtaining response surfaces from the RBFs listed in Table 5.1. In particular the linear RBF resulted in good results, in the sense that the graphs describing the response surfaces look as to the expected behavior of the real problem. All results presented in this thesis are obtained by the linear RBF. One important note is that some choices of RBFs can cause the matrix $\tilde{A}$ in (4.10) to become ill-conditioned. This may, in turn cause small changes in the input data to yield very large changes in the output data; see [20]. This is undesirable property should be kept in mind when choosing the basis as well. For the interested reader, a stability analysis can be found in [18].

Table 5.1: The RBFs tested.

RBF	$\phi(r)$
Linear	$-r$
Thin-plate splines	$r^2 \log r$
Multiquadrics	$(-1)\sqrt{r^2+1}$
Guassians	e^{-r^2}
Inverse multiquadrics	$1/\sqrt{1+r^2}$
Cubic	r^3

The choice of sample points is an interesting problem. We chose to use a fine uniform mesh of points which is of great importance when aiming for finding well distributed Pareto optimal solutions. This approach is possible since a simulation with the fundamental algorithm is not time consuming and therefore there is no problem in having a fine mesh of points. However, when considering problem instances with time consuming function evaluations then the choice of points are important. A typical approach is to create an initial set of points for the surrogate model and

then use methods to find new points for refinement of the surrogate model, as stated in Algorithm 1. Example of initializations of this type of approach are the so called Latin Hypercube Design ([21]), and the strategy for selecting new points to evaluate presented by Gutmann ([22]). These two methods might be relevant and appropriate for Volvo's automotive collision avoidance system.

It is rather easy to realize what the most favorable and the most unfavorable errors are for the positive performance scenarios, because in all the positive performance scenarios the host car drives straight ahead with a 100% offset. The most favorable longitudinal errors occur when the target car appears closer to the host car. This means that the values of the longitudinal errors are $\xi_{x_{\text{rel}}} = -b_1$, $\xi_{v_{\text{rel}}^{\text{long}}} = -b_2$, and $\xi_{a_{\text{tar}}^{\text{long}}} = -b_3$, respectively. The most favorable lateral errors occur when the target car appears to be directly in line with the host car. This means that the values of the lateral errors are $\xi_{y_{\text{rel}}} = 0$, $\xi_{v_{\text{rel}}^{\text{lat}}} = 0$, and $\xi_{a_{\text{tar}}^{\text{lat}}} = 0$, respectively. Finally, if the target car appears to be wider than it is, then the host car breaks earlier, so $\xi_{w_{\text{tar}}} = b_7$. The most unfavorable longitudinal errors occur when the target car appears to be further away from the host car than it really is. This means that the values of the longitudinal errors are $\xi_{x_{\text{rel}}} = b_1$, $\xi_{v_{\text{rel}}^{\text{long}}} = b_2$, and $\xi_{a_{\text{tar}}^{\text{long}}} = b_3$, respectively. The most unfavorable lateral errors are those that make the target car appears on the right or left side of the host car. This means that the values of the lateral errors are, for instance, $\xi_{y_{\text{rel}}} = b_4$, $\xi_{v_{\text{rel}}^{\text{lat}}} = b_5$, and $\xi_{a_{\text{tar}}^{\text{lat}}} = b_6$. Finally, if the target car appears to be narrower than it is, then the host car will break later, so $\xi_{w_{\text{tar}}} = -b_7$.

Figure 5.5 shows the response surface of the collision speed dependent on y_{safe}, with the most favorable and the most unfavorable errors for positive performance scenario one.

A posteriori methods, for example by solving (3.8), yields a representation of the Pareto optimal set which ease the choice of the Pareto optimal solution, since the trade-off between the objective functions is then visible.

One optimization solver used by Volvo is modeFRONTIER (see [23]), and which is used in this thesis. The software modeFRONTIER includes a variety of algorithms, mainly metaheuristics, that utilize a posteriori methods. Figure 5.6 shows a visualization of some of the Pareto optimal solutions, where the response surfaces was constructed by 1000 points, respectively. The leftmost axis shows the values of the variable y_{safe}. The other six axes illustrate the values of the six objective functions. The panel meter can be altered to specify requirements on the values of each of the objective functions. It also shows the resulting trade-off between the objective functions.

5.5 Generalization

We have found an excellent way to solve the tractable problem, see Definition (2.2.1). However, some of the steps in our methodology requires good knowledge of the fundamental algorithm. Especially when finding the trigger edge, because the development of Algorithm 3 required some manipulation of the relations in the fundamental algorithm. In Volvo's automotive collision avoidance system the corresponding relations might be more difficult to identify. For this reason it would be convenient to present a more general approach to finding the trigger edge or describing the function f_{edge}.

We start by defining a new term, namely *lateral acceleration margin*, defined as $a_{\text{margin}} := a_{\text{avail}}^{\text{lat}} - a_{\text{req}}^{\text{lat}}$. The idea is to choose a set of points $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, where $\boldsymbol{x}_i = ((y_{safe})_i, (a_{avail}^{lat})_i)^T$, $i = 1, \ldots, n$, as a uniform mesh over the feasible set given by $1 \leq a_{\text{avail}}^{\text{lat}} \leq 10$ and $-1 \leq y_{\text{safe}} \leq 0$. For

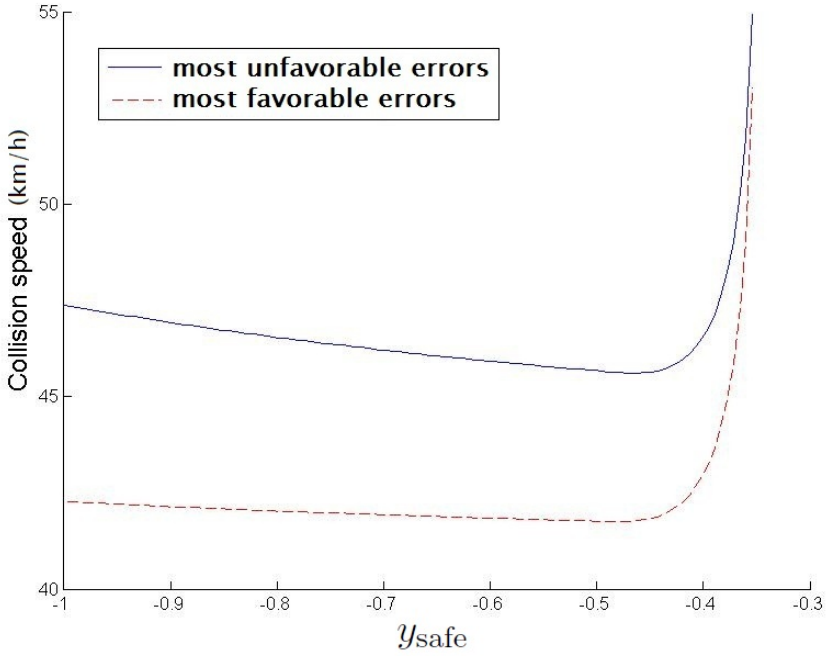

Figure 5.5: The relation between the collision speed, with most unfavorable errors, and y_{safe}. For this example we used a 30 point uniform mesh.

each point $\boldsymbol{x}_i \in X$ both the negative performance scenarios are simulated and in each time step in each simulation the value of a_{margin} is computed, if the BTN-condition is true. The minimum value of a_{margin} computed, over all time steps in both scenarios, is stored in the variable $(a_{\text{margin}}^{\text{final}})_i$. If $(a_{\text{margin}}^{\text{final}})_i < -10$ then it is set to $(a_{\text{margin}}^{\text{final}})_i := -10$. A response surface, $s_{a_{\text{margin}}^{\text{final}},X}(\boldsymbol{x})$, is constructed with input points $\boldsymbol{x}_i$ and output $(a_{\text{margin}}^{\text{final}})_i$ for all $i = 1, \ldots, n$. Figure 5.7 shows an example of such a response surface.

We are interested in those parameter settings, $\boldsymbol{x}$, that yield $s_{a_{\text{margin}}^{\text{final}},X}(\boldsymbol{x}) = 0$. The following simple method is used to find the roots for $s_{a_{\text{margin}}^{\text{final}},X}(\boldsymbol{x})$: Select a number of points $Y = \{y_1, \ldots, y_m\} \subset [-1, 0]$, which is the interval for y_{safe}, as a fine uniform mesh. For each point $y_j \in Y$ we search for a value $a_j \in [1, 10]$ which satisfies $|s_{a_{\text{margin}}^{\text{final}},X}((y_j, a_j)^T)| < \varepsilon$, where ε is a small positive chosen number. The search for a_j is done by bisecting the interval successively until the termination criterion is met or a maximum number of iterations is reached. If a maximum number of iterations has been reached we say that there is no $a_j \in [1, 10]$ such that $|s_{a_{\text{margin}}^{\text{final}},X}((y_j, a_j)^T)| < \varepsilon$. When all a_j, $j \in \{1, \ldots, m\}$ have been found, we construct a spline interpolation of the existing points a_i and the corresponding y_i; see Section 4.1 for spline interpolation. We construct a function $f_{\text{edge}}^{\text{approx}}(y_{\text{safe}})$ such that $f_{\text{edge}}^{\text{approx}}(y_j) = a_j$ for all $j \in \{1, \ldots, m\}$.

Figure 5.8 shows the differences between f_{edge} and $f_{\text{edge}}^{\text{approx}}$, and which depend on the choice of the radial basis function and on the resolution of points. In general, the finer the mesh the smaller

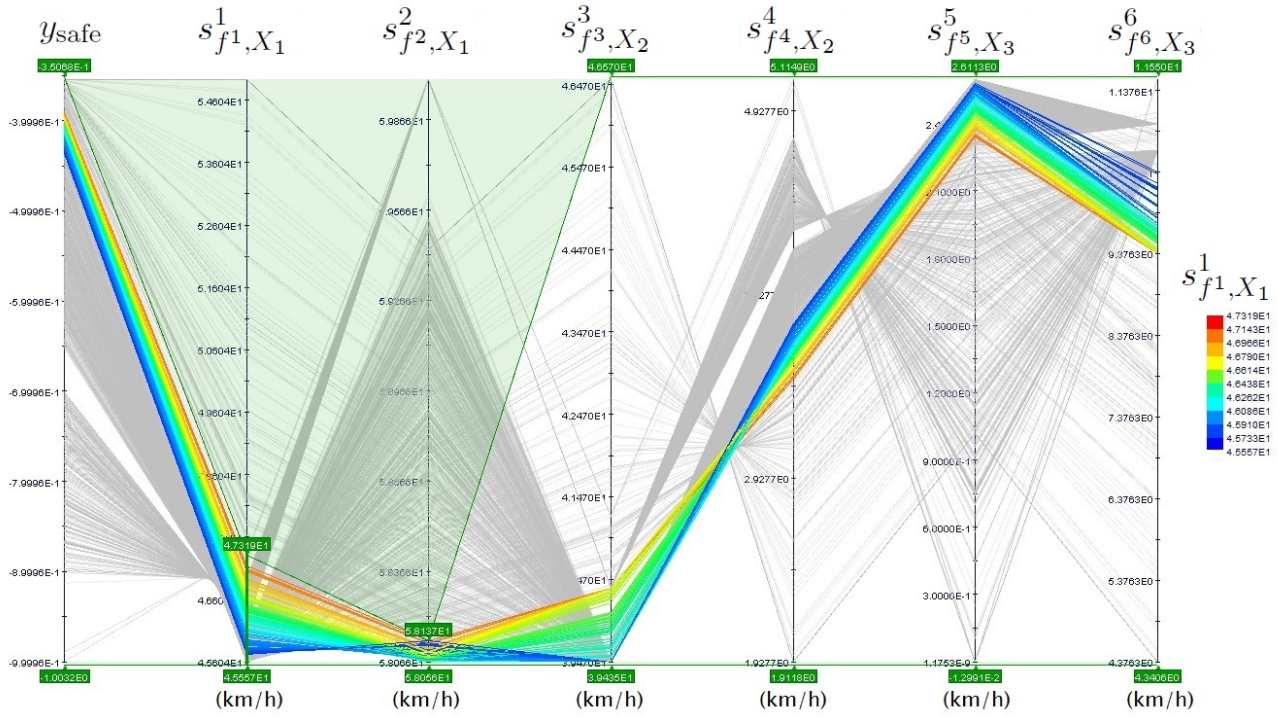

Figure 5.6: A representation of some of the Pareto optimal solutions in modeFRONTIER.

the differences.

5.6 Summary of the methodology of finding robust solutions

Figure 5.9 shows an overview of the methodology for finding a parameter setting that fulfills robustness of both negative and positive performance. Remember that first the tunable parameter $a^{\text{long}}_{\text{avail}}$ is optimized, then the steps in the flowchart are taken for certain different velocities. There are two possible ways to find the edge, either by the analytical approach or by the approximate one. A description of each block in the flowchart is presented bellow.

1a. Compute the worst combination of errors and generate the constraints

Simulate negative scenarios one and two. In each time step of each simulation, find the worst combination of errors by using Algorithm 2. In each time step, generate also the constraints, which describe the lowest bound on the two tunable parameters y_{safe} and $a^{\text{lat}}_{\text{avail}}$ which do not cause false intervention. This is done by using Algorithm 3.

2a. Find the trigger edge from the generated constraints

Combine all the constraints from step 1a to form the trigger edge. No point on or above the trigger edge will cause any false intervention. Knowing that the optimal parameter setting for the positive performance scenarios will lie on the edge, we compute a function f_{edge}, by using Algorithm 4, which describes the combination of y_{safe} and $a^{\text{lat}}_{\text{avail}}$ that lie on the edge.

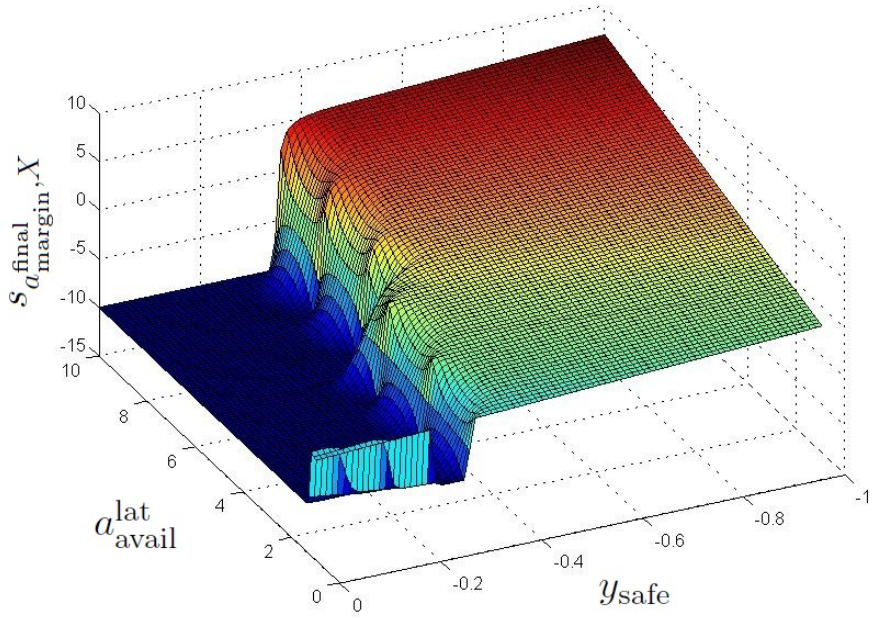

Figure 5.7: The response surface $s_{a^{\text{final}}_{\text{margin}},X}$. For this example we used an 85×85 points uniform mesh.

1b. Compute the worst combination of errors and the lateral acceleration margin

For each parameter setting, simulate negative scenarios one and two; in each time step of each simulation, find the worst combination of errors using Algorithm 2 and compute a_{margin}; see Section 5.5. Then $a^{\text{final}}_{\text{margin}}$ is computed using the minimum value of a_{margin} from each simulation.

2b. Find the trigger edge from the response surface of the lowest lateral acceleration margin by finding its roots

Construct a response surface $s_{a^{\text{final}}_{\text{margin}},X}(\boldsymbol{x})$ of the $a^{\text{final}}_{\text{margin}}$ and the parameter settings. Then search for parameter settings, $\boldsymbol{x}$, such that $s_{a^{\text{final}}_{\text{margin}},X}(\boldsymbol{x}) = 0$, which forms the trigger edge.

3. Construct response surfaces that reflect robust positive performance

Construct six response surfaces by selecting sets of points which lie on the trigger edge, and simulate the collision speed and its spread from the three positive performance scenarios.

4. Search for Pareto optimal solutions

To find the parameter settings that fulfill robustness of positive performance, we use the software modeFRONTIER to search for Pareto optimal solutions to the optimization problem (5.21).

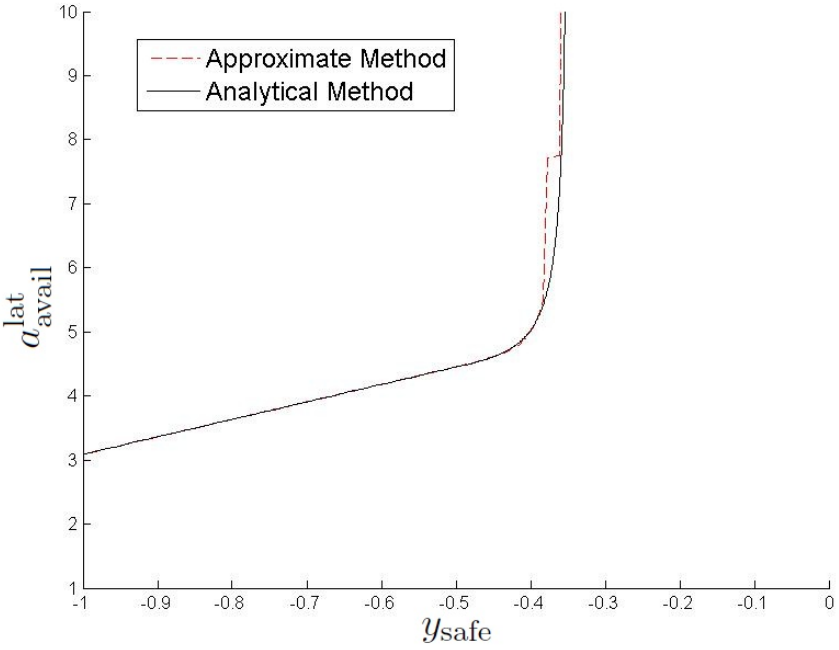

Figure 5.8: The different representation of the trigger edge from approximate and the analytical approach. We used the same settings as in Figure 5.7.

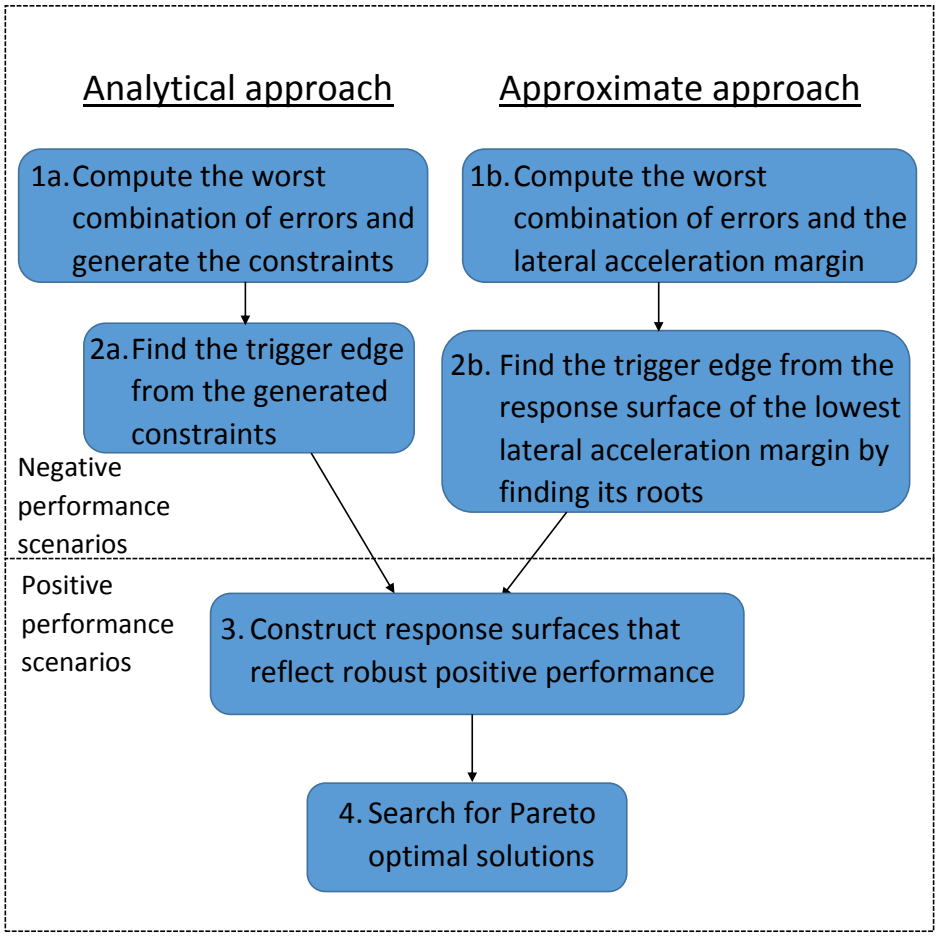

Figure 5.9: A flowchart of the methodology to find parameter settings that fulfill robustness of both negative and positive performance.

Chapter 6

Discussion and conclusions

In this chapter we discuss and analyze the robust design methodology. We start, in Section 6.1, to evaluate the robust design methodology in terms of result, what general tools it has provided, and what weakness it has. Next it is discussed if the approach of handling the worst combination of errors is a valid approach; see Section 6.2. In Section 6.3 it is analyzed how the choice of the objective functions affect the positive performance. Thereafter, we analyze the tunable parameter safety zone and its impact on the fundamental algorithm; see Section 6.4. In Section 6.5 pros and cons of the two different approaches of finding the trigger edge is discussed. At last, a discussion of what the future work may involve concludes this chapter; see Section 6.6.

6.1 Evaluation of the robust design methodology

The methodology finds parameter settings that fulfill robustness of negative and positive performance. The fulfillment of robustness of negative performance scenarios reflects the high priority of avoiding false intervention in case of uncertain measurements. False intervention is highly undesirable since automotive collision avoidance systems are meant to assist and help the driver. The fulfillment of robustness of positive performance entails that the car assists the driver in preventing a collision as well as possible, and guarantees a certain level of performance even though errors from the sensor may occur. A parameter setting found by the robust design methodology solves the tractable problem; see Definition 2.2.1. The tractable problem reflects the real problem of finding a robust parameter setting for Volvo's automotive collision avoidance system.

The methodology describes several general concepts that are likely to be applicable within Volvo's automotive collision avoidance system; examples are the general idea of robustness, methods for finding the trigger edge, and the methodology of using response surfaces. As mentioned in Section 4.2 radial basis functions are independent of the dimension of the variable space. This property has been of great use in this thesis: it is used to create the six objective univariate functions that reflect positive performance (see Section 5.4), as well as the bivariate $s_{a^{\text{final}}_{\text{margin}},X}$ surface (see Section 5.5). Furthermore, this property also make it possible for response surfaces to be applicable for Volvo's automotive collision avoidance system which contains even more tunable parameters. Another important property of response surfaces is that they are computational efficient tools for handling time consuming simulations, as Volvo's automotive collision avoidance system. However,

we always need to have in mind that there is no guarantee that the response surface mimics the simulation in a satisfying way. If that is the case, the optimal parameter setting found, can be far from optimum. This is the problem with simulation-based optimization in general, as mentioned in Section 3.3. However, the risk of finding a parameter setting far from optimum is extremely small when solving the tractable problem as we did, since a really fine mesh of points is used (see Section 5.4) to a low dimensional problem; see problem (5.21). Furthermore, linear RBFs together with a very fine mesh of points probably mimics the simulation in a satisfying way, since small changes of the parameter setting results in small changes of when the host car starts to fully brake, which has a linear relation with the collision speed. Lastly, Volvo have verified that the results are good.

6.2 Analysis of worst case scenario

The approach of ascertaining, that even the worst combination of errors should not cause a false intervention, was made to guarantee a certain level of performance. Moreover, this was a natural approach when considering bounds on the errors but not on their distributions. However, the worst combination of errors might be very unlikely to occur, making this approach very pessimistic. A more comprehensive analysis could be made by introducing statistical data for the error distributions. With that said—since false intervention is very undesirable—if the boundaries of the errors capture the spread of the errors satisfactory, a zero tolerance for false intervention within the bounds is probably the way to go anyway.

We have disregarded the possibility of errors depending on, for example, distance. The measurements might be more accurate at closer range. If such data are introduced, less pessimistic bounds could be used, which would result in a better performance in the positive performance.

6.3 Analysis of positive performance

Defining what the objective functions should represent was an interesting problem. It is important that the objective functions are chosen such that the performance is kept above a certain level. When Volvo Car Corporation wants to launch a new car model it has to undergo multiple performance tests and therefore it is important that the system can guarantee a high lowest performance, so the cars always pass the tests. This means that we want to minimize the collision speed, subject to the most unfavorable errors. By minimizing the spread of the collision speed we strengthen the approach of consistency. However, this requires that the boundaries of the errors capture the spread of the errors satisfactory.

By using statistical data, that enables insights into the importance of the different scenarios, it is possible to weight the objective functions into one function. This eliminates the necessity to search for Pareto optimal solutions.

6.4 Analysis of the safety zone

In Section 5.2 we established that a certain size of the safety zone is needed to avoid false intervention. We conclude from Figures 5.1 and 5.2 that the only negative performance scenario that generates

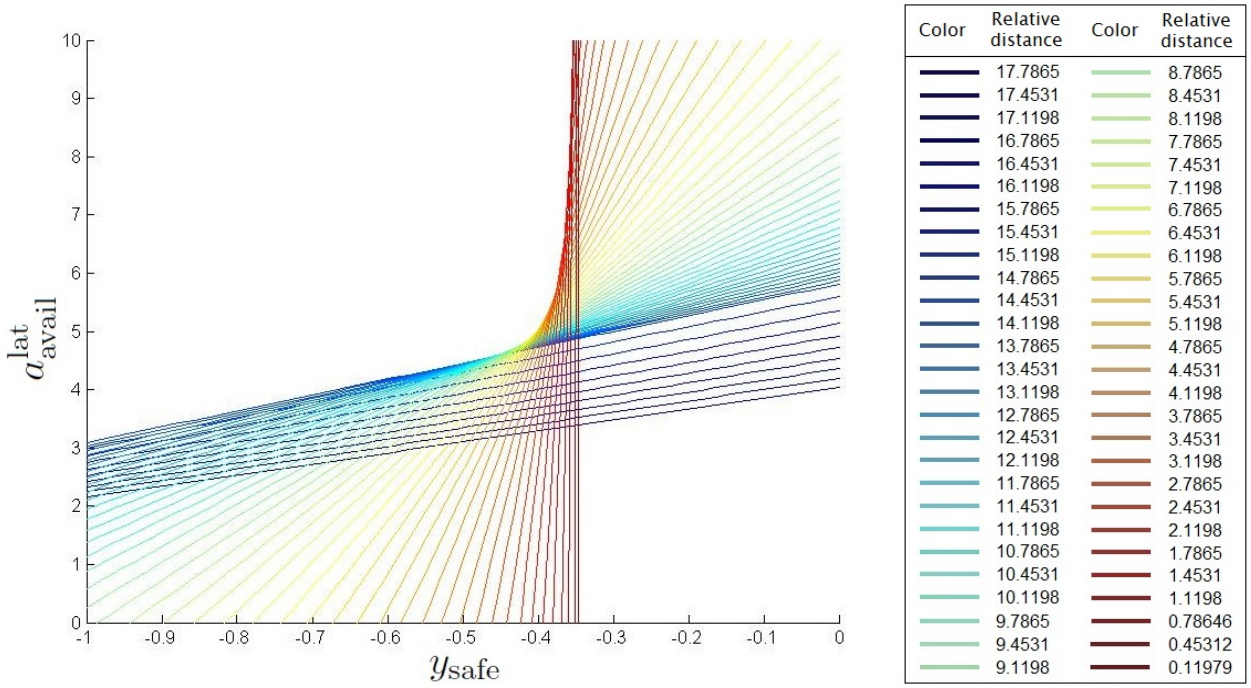

Figure 6.1: The relative distance where constraints are generated.

relevant, i.e., not redundant, constraints is the second scenario. It is interesting to investigate in which time step a certain constraint is generated in the second negative performance scenario, to get an understanding of the relation between the two tunable parameters y_{safe} and $a^{\text{lat}}_{\text{avail}}$. In Figure 6.1 each constraint is represented by a specific color for each specific relative distance to the target car. The figure reveals that a certain size of the safety zone is needed to avoid a false intervention when the host car is close to the target car. We conclude that the only way to avoid a false intervention is to compensate for the errors that cause the target car to appear as being in the way of the host car, by making the target car appear to be narrower.

6.5 Pros and cons of the analytical and approximate approach for finding the trigger edge

As mentioned in Section 5.5, the reason for an approximate approach is that good knowledge of the fundamental algorithm is needed to develop an analytical approach, as Algorithm 3. In Volvo's automotive collision avoidance system the corresponding relations might be more difficult to find and identify, whence it might be impossible to develop an analogous algorithm to Algorithm 3. However, an analytical algorithm would of course be more desirable, since the analytical approach yields an exact representation of the trigger edge and requires only one simulation. Figure 5.8 reveals the differences between the results from the analytical and the approximate approaches; since these are

rather small, the approximate method may be useful. Note also, in general, the more points used for constructing the response surface of $a_{\text{margin}}^{\text{final}}$ (see Section 5.5), the more accurate the approximate approach will be.

6.6 Future work

In Section 5.5 we generalized the approach of finding the trigger edge by utilizing a response surface. Perhaps even more steps in the methodology need to be generalized. For instance, introducing more performance scenarios would ensure an even better quality, but in some of those scenarios it could be hard to identify the worst possible combination of errors. An interesting approach would be to search for the trigger edge and the worst combination of errors simultaneously.

Another interesting problem is the choice of the radial basis function. It was out of the scope of this project to test more than the most popular RBFs. There are strategies of measuring the errors of radial basis functions such as cross-validation; see [24]. This is especially important if more variables are introduced, which is the case of Volvo's automotive collision avoidance system; in such cases, the problem and its solutions will no longer be as easy to visualize.

The main objective of the future work is, of course, trying to optimize the tunable parameters in Volvo's automotive collision avoidance system.

Appendix A

Technical description of algorithms developed in the thesis

In this appendix we present two algorithms that we have developed to solve the tractable problem, see Definition (2.2.1).

Algorithm 3 Find all constraints for the tunable parameters y_{safe} and $a_{\text{avail}}^{\text{lat}}$

Step 0:

Let $\mathcal{I} := \{1, 2\}$ be the index set
Let $\mathcal{N} := \emptyset$ be the set of constraints
Let $\mathcal{B}$ defined as in (5.17)
Let m_1, m_2, k_1, and k_2 defined as in (5.11) and (5.12)
Let $t_1 := \frac{-m_1}{k_1}$ and $t_2 := \frac{-m_2}{k_2}$

Step 1:

if $t_1 > 0$ & $t_2 > 0$
Let $f_1(y_{\text{safe}}) := m_1 + k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := -m_2 - k_2 \cdot y_{\text{safe}}$
Let $y_{\text{mid}} := -\frac{1}{2}$
Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, 0] \cap \mathcal{B}\}$
Proceed to **Step 2**

else if $t_1 < -1$ & $t_2 < -1$
Let $f_1(y_{\text{safe}}) := -m_1 - k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := m_2 + k_2 \cdot y_{safe}$
Let $y_{\text{mid}} := -\frac{1}{2}$
Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, 0] \cap \mathcal{B}\}$
Proceed to **Step 2**

Continued on next page

else if $t_1 \geq t_2$

if $t_1 \geq 0$ & $-1 \leq t_2 \leq 0$

Let $f_1(y_{\text{safe}}) := m_1 + k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := m_2 + k_2 \cdot y_{\text{safe}}$

Intersection point $y_{\text{inter}} = \frac{m_1 - m_2}{k_2 - k_1}$

if $t_2 < y_{\text{inter}} < 0$

Let $y_{\text{mid}} := \frac{y_{\text{inter}}}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [y_{\text{inter}}, 0] \cap \mathcal{B}\}$

Let $y_{\text{mid}} := \frac{t_2 + y_{\text{inter}}}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_2, y_{\text{inter}}] \cap \mathcal{B}\}$

else

Let $y_{\text{mid}} := \frac{t_2}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_2, 0] \cap \mathcal{B}\}$

end if

Let $f_1(y_{\text{safe}}) := m_1 + k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := -m_2 - k_2 \cdot y_{\text{safe}}$

Let $y_{\text{mid}} := \frac{t_2 - 1}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, t_2] \cap \mathcal{B}\}$

Proceed to **Step 2**

else if $t_1 \geq 0$ & $t_2 \leq -1$

Let $f_1(y_{\text{safe}}) := m_1 + k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := m_2 + k_2 \cdot y_{\text{safe}}$

Intersection point $y_{\text{inter}} = \frac{m_1 - m_2}{k_2 - k_1}$

if $-1 < y_{\text{inter}} < 0$

Let $y_{\text{mid}} := \frac{y_{\text{inter}}}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [y_{\text{inter}}, 0] \cap \mathcal{B}\}$

Let $y_{\text{mid}} := \frac{y_{inter} - 1}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, y_{inter}] \cap \mathcal{B}\}$

else

Let $y_{\text{mid}} = -\frac{1}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, 0] \cap \mathcal{B}\}$

end if

Proceed to **Step 2**

else if $-1 \leq t_1 \leq 0$ & $t_2 \leq -1$

Let $f_1(y_{\text{safe}}) := -m_1 - k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := m_2 + k_2 \cdot y_{\text{safe}}$

Let $y_{\text{mid}} = \frac{t_1}{2}$

Continued on next page

Let $i \in \underset{j \in \mathcal{I}}{\operatorname{argmin}} f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_1, 0] \cap \mathcal{B}\}$
Let $f_1(y_{\text{safe}}) := m_1 + k_1 \cdot y_{safe}$ and $f_2(y_{\text{safe}}) := m_2 + k_2 \cdot y_{\text{safe}}$
Intersection point $y_{\text{inter}} = \frac{m_1 - m_2}{k_2 - k_1}$
if $-1 < y_{inter} < t_1$
Let $y_{\text{mid}} := \frac{y_{\text{inter}} + t_1}{2}$
Let $i \in \underset{j \in \mathcal{I}}{\operatorname{argmin}} f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [y_{\text{inter}}, t_1] \cap \mathcal{B}\}$
Let $y_{\text{mid}} = \frac{y_{\text{inter}} - 1}{2}$
Let $i \in \underset{j \in \mathcal{I}}{\operatorname{argmin}} f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, y_{\text{inter}}] \cap \mathcal{B}\}$
else
Let $y_{\text{mid}} = \frac{t_1 - 1}{2}$
Let $i \in \underset{j \in \mathcal{I}}{\operatorname{argmin}} f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, t_1] \cap \mathcal{B}\}$
end if
Proceed to **Step 2**
else
Let $f_1(y_{\text{safe}}) := -m_1 - k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := m_2 + k_2 \cdot y_{\text{safe}}$
Let $y_{\text{mid}} := \frac{t_1}{2}$
Let $i \in \underset{j \in \mathcal{I}}{\operatorname{argmin}} f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_1, 0] \cap \mathcal{B}\}$
Let $f_1(y_{\text{safe}}) := m_1 + k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := m_2 + k_2 \cdot y_{safe}$
Intersection point $y_{\text{inter}} = \frac{m_1 - m_2}{k_2 - k_1}$
if $t_2 < y_{\text{inter}} < t_1$
Let $y_{\text{mid}} := \frac{y_{\text{inter}} + t_1}{2}$
Let $i \in \underset{j \in \mathcal{I}}{\operatorname{argmin}} f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [y_{\text{inter}}, t_1] \cap \mathcal{B}\}$
Let $y_{\text{mid}} := \frac{y_{\text{inter}} + t_2}{2}$
Let $i \in \underset{j \in \mathcal{I}}{\operatorname{argmin}} f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_2, y_{\text{inter}}] \cap \mathcal{B}\}$
else
Let $y_{\text{mid}} := \frac{t_1 + t_2}{2}$
Let $i \in \underset{j \in \mathcal{I}}{\operatorname{argmin}} f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_2, t_1] \cap \mathcal{B}\}$
end if
Let $f_1(y_{\text{safe}}) := m_1 + k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := -m_2 - k_2 \cdot y_{\text{safe}}$
Let $y_{\text{mid}} = \frac{-1 + t_2}{2}$
Let $i \in \underset{j \in \mathcal{I}}{\operatorname{argmin}} f_j(y_{\text{mid}})$
Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, t_2] \cap \mathcal{B}\}$
Proceed to **Step 2**
end if

Continued on next page

else

if $t_2 \geq 0$ & $-1 \leq t_1 \leq 0$

Let $f_1(y_{\text{safe}}) := -m_1 - k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := -m_2 - k_2 \cdot y_{\text{safe}}$

Intersection point $y_{\text{inter}} = \frac{m_1 - m_2}{k_2 - k_1}$

if $t_1 < y_{inter} < 0$

Let $y_{\text{mid}} = \frac{y_{\text{inter}}}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [y_{\text{inter}}, 0] \cap \mathcal{B}\}$

Let $y_{\text{mid}} = \frac{t_1 + y_{\text{inter}}}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_1, y_{\text{inter}}] \cap \mathcal{B}\}$

else

Let $y_{\text{mid}} = \frac{t_1}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_1, 0] \cap \mathcal{B}\}$

end if

Let $f_1(y_{\text{safe}}) := m_1 + k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := -m_2 - k_2 \cdot y_{\text{safe}}$

Let $y_{\text{mid}} = \frac{-1 + t_1}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, t_1] \cap \mathcal{B}\}$

Proceed to **Step 2**

else if $t_2 \geq 0$ & $t_1 \leq -1$

Let $f_1(y_{\text{safe}}) := -m_1 - k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := -m_2 - k_2 \cdot y_{\text{safe}}$

Intersection point $y_{\text{inter}} = \frac{m_1 - m_2}{k_2 - k_1}$

if $-1 < y_{\text{inter}} < 0$

Let $y_{\text{mid}} = \frac{y_{\text{inter}}}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{safe} \in [y_{\text{inter}}, 0] \cap \mathcal{B}\}$

Let $y_{\text{mid}} := \frac{-1 + y_{\text{inter}}}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, y_{\text{inter}}] \cap \mathcal{B}\}$

else

Let $y_{\text{mid}} = -\frac{1}{2}$

Choose one $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, 0] \cap \mathcal{B}\}$

end if

Proceed to **Step 2**

else if $-1 \leq t_2 \leq 0$ & $t_1 \leq -1$

Let $f_1(y_{\text{safe}}) := -m_1 - k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := m_2 + k_2 \cdot y_{\text{safe}}$

Let $y_{\text{mid}} = \frac{t_2}{2}$

Let $i \in \underset{j \in \mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_2, 0] \cap \mathcal{B}\}$

Let $f_1(y_{\text{safe}}) := -m_1 - k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := -m_2 - k_2 \cdot y_{\text{safe}}$

Intersection point $y_{\text{inter}} = \frac{m_1 - m_2}{k_2 - k_1}$

Continued on next page

if $-1 < y_{\text{inter}} < t_2$

Let $y_{\text{mid}} = \frac{y_{\text{inter}}+t_2}{2}$

Let $i \in \underset{j\in\mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [y_{\text{inter}}, t_2] \cap \mathcal{B}\}$

Let $y_{\text{mid}} = \frac{y_{\text{inter}}-1}{2}$

Let $i \in \underset{j\in\mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, y_{\text{inter}}] \cap \mathcal{B}\}$

else

Let $y_{\text{mid}} = \frac{t_2-1}{2}$

Let $i \in \underset{j\in\mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, t_2] \cap \mathcal{B}\}$

end if

Proceed to **Step 2**

else

Let $f_1(y_{\text{safe}}) := -m_1 - k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := m_2 + k_2 \cdot y_{\text{safe}}$

Let $y_{\text{mid}} = \frac{t_2}{2}$

Let $i \in \underset{j\in\mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_2, 0] \cap \mathcal{B}\}$

Let $f_1(y_{\text{safe}}) := -m_1 - k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := -m_2 - k_2 \cdot y_{\text{safe}}$

Intersection point $y_{\text{inter}} = \frac{m_1-m_2}{k_2-k_1}$

if $t_1 < y_{\text{inter}} < t_2$

Let $y_{\text{mid}} := \frac{y_{\text{inter}}+t_2}{2}$

Let $i \in \underset{j\in\mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [y_{\text{inter}}, t_2] \cap \mathcal{B}\}$

Let $y_{\text{mid}} = \frac{y_{\text{inter}}+t_1}{2}$

Let $i \in \underset{i\in\mathcal{I}}{\text{argmin}}\ f_i(y_{mid})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{safe})$, where $y_{safe} \in [t_1, y_{inter}] \cap \mathcal{B}\}$

else

Let $y_{\text{mid}} := \frac{t_1+t_2}{2}$

Let $i \in \underset{j\in\mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [t_1, t_2] \cap \mathcal{B}\}$

end if

Let $f_1(y_{\text{safe}}) := m_1 + k_1 \cdot y_{\text{safe}}$ and $f_2(y_{\text{safe}}) := -m_2 - k_2 \cdot y_{\text{safe}}$

Let $y_{\text{mid}} := \frac{-1+t_1}{2}$

Let $i \in \underset{j\in\mathcal{I}}{\text{argmin}}\ f_j(y_{\text{mid}})$

Update $\mathcal{N} \leftarrow \mathcal{N} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [-1, t_1] \cap \mathcal{B}\}$

Proceed to **Step 2**

end if

end if

Step 2: The set $\mathcal{N}$ contains all constraints.

Algorithm 4 Construct the function f_{edge} from the constraints generated of Algorithm 3

Step 0:

Let $\mathcal{N}$ be the set of constraints from Algorithm 3, where all definition sets are extended to the interval [-1,0]

Let $\widetilde{\mathcal{N}} := \emptyset$ be the set of relevant constraints

Let $\mathcal{I} := \{1, \ldots, |\mathcal{N}|\}$ be the index set

Let $y_{\text{left}} := -1$

Step 1:

Let $S := \underset{i \in \mathcal{I}}{\text{argmax}}\ f_i(y_{\text{left}})$, where $f_i \in \mathcal{N}$

if $|S| > 1$

Choose $i \in S$ such that $k_i \geq k_l$ for all $l \in S$, where k_j is the slope of function $f_j \in \mathcal{N}$

else

Choose $i \in S$

end if

Step 2:

Update $\mathcal{I} \leftarrow \mathcal{I} \backslash \{i\}$

Let $\widetilde{S} := \underset{j \in \left\{ l \in \mathcal{I} : \frac{f_i(0) - f_l(0)}{k_l - k_i} > y_{left} \right\}}{\text{argmin}} \frac{f_i(0) - f_j(0)}{k_j - k_i}$, where k_j is the slope of function $f_j \in \mathcal{N}$

if $|\widetilde{S}| > 1$

Choose $j \in S$ such that $k_j \geq k_l$ for all $l \in \widetilde{S}$ where k_l is the slope of function $f_l \in \mathcal{N}$

else

Choose $j \in \widetilde{S}$

end if

Let $y_{\text{right}} := \frac{f_i(0) - f_j(0)}{k_j - k_i}$

Step 3:

if $f_i(y_{\text{right}}) \geq 10$

Update $\widetilde{\mathcal{N}} \leftarrow \widetilde{\mathcal{N}} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [y_{\text{left}}, 0]\}$

Proceed to **Step 4**

else

Update $\widetilde{\mathcal{N}} \leftarrow \widetilde{\mathcal{N}} \cup \{f_i(y_{\text{safe}}) : y_{\text{safe}} \in [y_{\text{left}}, y_{\text{right}})\}$

Let $i := j$

Return to **Step 2**

Step 4:

The set $\widetilde{\mathcal{N}}$ contains all constraints that defines the edge

Appendix B

Supplementary theory for Chapter 4

All theory in this appendix is taken from [25], except for the first definition and theorem which are taken from [26].

Definition B.1. *A symmetric $n \times n$ real matrix A is said to be positive definite if $x^T Ax > 0$ for all $x \in \mathbb{R}^n \neq \mathbf{0}^n$.*

Theorem B.2. *A positive definite matrix A is invertible.*

Proof. Assume that the matrix A is not invertible. Then there exists a non-zero vector $\boldsymbol{x}$ such that $A\boldsymbol{x} = \mathbf{0}$. This implies that $\boldsymbol{x}^T A\boldsymbol{x} = \mathbf{0}$ which contradicts the assumptions. □

Definition B.3 (Vector space)**.** *By a vector space we mean a non-empty set E with two operations:*

a mapping $(x, y) \mapsto x + y$ from $E \times E$ onto E, called addition,
a mapping $(\lambda, x) \mapsto \lambda x$ from $\mathbb{F} \times E$ onto E, called multiplication by scalars,

such that the following conditions are satisfied:

(a) $x + y = y + x$.
(b) $(x + y) + z = x + (y + z)$.
(c) For every $x, y \in E$ there exists $z \in E$ such that $x + z = y$.
(d) $\alpha(\beta x) = (\alpha\beta)x$.
(e) $(\alpha + \beta)x = \alpha x + \beta x$.
(f) $\alpha(x + y) = \alpha x + \alpha y$.
(g) $1x = x$.

Elements of E are called vectors. If $\mathbb{F} = \mathbb{R}$, then E is called a real vector space, and if $\mathbb{F} = \mathbb{C}$, E is called a complex vector space.

Example 1. The scalar fields $\mathbb{R}$ and $\mathbb{C}$ are the simplest non-trivial vector spaces. Further, $\mathbb{R}^n$ and $\mathbb{C}^n$ are vector spaces. □

Example 2 (Function spaces)**.** Let X be an arbitrary non-empty set and let E be a vector space. Denote by F the space of all functions from X into E. Then F becomes a vector space if the addition and multiplication by scalars are defined in the following way:

$$(f+g)(x) = f(x) + g(x),$$
$$(\lambda f)(x) = \lambda f(x).$$

The zero vector in F is the function which assigns the zero vector of E to every element of X. □

Definition B.4 (Norm)**.** *A function $x \mapsto \|x\|_E$ from a vector space E into $\mathbb{R}$ is called a norm if it satisfies the following conditions:*

(a) $\|x\|_E = 0$ if and only if $x = 0$.
(b) $\|\lambda x\|_E = |\lambda|\|x\|_E$ for every $x \in E$ and $\lambda \in \mathbb{F}$.
(c) $\|x + y\|_E \leq \|x\|_E + \|y\|_E$ for every $x, y \in E$.

It is called a semi-norm if all conditions are satisfied except (a).

Definition B.5 (Normed space)**.** *A vector space with a norm is called a normed space.*

Definition B.6 (Cauchy sequence)**.** *A sequence of vectors (x_n) in a normed space is called a Cauchy sequence if for every $\varepsilon > 0$ there exists a number M such that $\|x_m - x_n\| < \varepsilon$ for all $m, n > M$.*

Definition B.7 (Banach space)**.** *A normed space E is called complete if every Cauchy sequence in E converges to an element of E. A complete normed space is called a Banach space.*

Definition B.8 (Linear mappings)**.** *A mapping $L : E_1 \to E_2$ is called a linear mapping if $L(\alpha x + \beta y) = \alpha L(x) + \beta L(y)$ for all $x, y \in E_1$ and all scalars α, β.*

Definition B.9 (Continuous mappings)**.** *Let E_1 and E_2 be normed spaces. A mapping F from E_1 into E_2 is called continuous at $x_0 \in E_1$ if for any sequence (x_n) of elements of E_1 convergent to x_0, the sequence $(F(x_n))$ converges to $F(x_0)$, i.e., F is continuous at x_0 if $\|x_n - x_0\| \to 0$ implies $\|F(x_n) - F(x_0)\| \to 0$. If F is continuous at every $x \in E_1$, then we simply sat that F is continuous.*

Definition B.10 (Bounded linear mappings)**.** *Let E_1 and E_2 be normed spaces. A linear mapping $L : E_1 \to E_2$ is called bounded if there exists a number K such that $\|L(x)\| \leq K\|x\|$ for all $x \in E_1$.*

Theorem B.11. *A linear mapping is continuous if and only if it is bounded.*

Example 3. The space of all linear mappings from a vector space E_1 into a vector space E_2 becomes a vector space if the addition and multiplication by scalars are defined as follows:

$$(L_1 + L_2)(x) = L_1(x) + L_2(x) \text{ and } (\lambda L)(x) = \lambda(L(x)).$$

If E_1 and E_2 are normed spaces, then the set of all bounded linear mappings from E_1 into E_2, denoted by $\mathscr{B}(E_1, E_2)$ is a vector subspace of the space defined above. □

Elements of spaces, $\mathscr{B}(E, \mathbb{F})$, of bounded linear mappings from a normed space E into a scalar field $\mathbb{F}$ are called *functionals*. The space $\mathscr{B}(E, \mathbb{F})$ is sometimes denoted E^* and called the *dual space* of E.

Theorem B.12. *If E_1 is a normed space and E_2 is a Banach space, then $\mathscr{B}(E_1, E_2)$ is a Banach space.*

Definition B.13 (Bilinear form)**.** *By a bilinear form ϕ on a real vector space E, we mean a mapping $\phi : E \times E \to \mathbb{C}$ satisfying the following two conditions:*

$$\begin{aligned} &(a)\ \phi(\alpha x_1 + \beta x_2, y) = \alpha\phi(x_1, y) + \beta\phi(x_2, y), \\ &(b)\ \phi(x, \alpha y_1 + \beta y_2) = \overline{\alpha}\phi(x, y_1) + \overline{\beta}\phi(x, y_2), \end{aligned}$$

for any $\alpha, \beta \in \mathbb{C}$ and any $x, x_1, x_2, y, y_1, y_2 \in E$. Note that the bar over α and β denotes complex conjugation.

Definition B.14 (Inner product space)**.** *Let E be a complex vector space. A mapping $\langle \cdot, \cdot \rangle : E \times E \to \mathbb{C}$ is called an inner product in E if for any $x, y, z \in E$ and $\alpha, \beta \in \mathbb{C}$ the following conditions are satisfied:*

(a) $\langle x, y \rangle = \overline{\langle y, x \rangle}$.
(b) $\langle \alpha x + \beta y, z \rangle = \alpha\langle x, z \rangle + \beta\langle y, z \rangle$.
(c) $\langle x, x \rangle \geq 0$, *and* $\langle x, x \rangle = 0 \Leftrightarrow x = 0$.

A vector space with an inner product is called an inner product space or a pre-Hilbert space.
If all conditions are satisfied except the last part in (c) namely $\langle x, x \rangle = 0 \Leftrightarrow x = 0$ it is called a semi-inner product.

Proposition B.15. *Every inner product space is also a normed space with the norm defined by $\|x\| = \sqrt{\langle x, x \rangle}$.*

Definition B.16 (Hilbert space)**.** *A complete inner product space is called a Hilbert Space.*

Example 4. The spaces $\mathbb{R}, \mathbb{R}^n, \mathbb{C}$ and $\mathbb{C}^n$ are Hilbert spaces. □

Theorem B.17 (Riesz representation Theorem)**.** *Let f be a bounded linear functional on a Hilbert space H. There exists exactly one $x_0 \in H$ such that $f(x) = \langle x, x_0 \rangle$ for all $x \in H$. Moreover, we have $\|f\|_{H^*} = \|x_0\|_H$.* □

The space H^* of all bounded linear functionals on a Hilbert space H is a Banach space, see Theorem B.12. The Riesz representation Theorem states that $H^* = H$, or more precisely, H^* and H are isomorphic. Thereby the dual space H^* has an inner product.

Bibliography

[1] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, Princeton, USA (2009).

[2] urlhttps://www.media.volvocars.com/global/en-gb/media/pressreleases/163733/volvo-cars-standard-safety-technology-cuts-accident-claims-by-28-per-cent

[3] W. Rudin. *Principles of Mathematical Analysis Third Edition*. McGraw-Hill Book Co, Singapore, Republic of Singapore (1976).

[4] N. Andréasson, A. Evgrafov, M. Patriksson, E. Gustavsson and M. Önnheim. *An Introduction to Continuous Optimization 2nd Edition*. Studentlitteratur, Lund, Sweden (2013).

[5] R. Horst, P. M. Pardalos and N. V. Thoai. *Introduction to Global Optimization 2nd Edition*. Kluwer Academic Publishers, Dordrecht, Netherlands (2000).

[6] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Dordrecht, Netherlands (1998).

[7] Z. Šabartová. *Mathematical modelling for optimization of truck tyres selection*, Licentiate thesis, Department of Mathematical Sciences, Chalmers University of Technology, Gothenburg, Sweden (2015).

[8] R. Tyrrell Rockafellar and S. Uryasev. *Optimization of Conditional Value-at-Risk*. Journal of risk 2, pp. 21–42 (2000).

[9] T. H. Truong and F. Azadivar. *Simulation optimization in manufacturing analysis: simulation based optimization for supply chain configuration design*. Proceedings of the 35th Conference on Winter Simulation: Driving Innovation. Winter Simulation Conference, pp. 1268—1275 (2003).

[10] P. Kim and Y. Ding. *Optimal engineering system design guided by data-mining methods*. Technometrics, Volume 47, Issue 3, pp. 336–348 (2005).

[11] M. F. Iskander and A. M. Tumeh. *Design Optimization of Intersitial Antennas*. IEEE Transactions on Biomedical Engineering, Volume 36, Issue 2, pp. 238–246 (1989).

[12] S. Jakobsson, M. Patriksson, J. Rudholm and A. Wojciechowski. *A method for simulation based optimization using radial basis functions.* Optimization and Engineering, Volume 11, Issue 4, pp. 501–532 (2010).

[13] C. Yolanda and M. Anu. *Simulation Optimization: Methods and Applications*. Proceedings of the 29th conference on Winter simulation, IEEE Computer Society, pp. 118–126 (1997).

[14] G. Deng. *Simulation-based optimization*. PhD thesis, University of Wisconsin, Madison (2007).

[15] M. Wahde. *Biologically Inspired Optimization Methods an Introduction*. WIT Press, Southampton, United Kingdom (2008).

[16] F. W. Glover and G. A. Kochenberger. *Handbook of Metaheuristics*. Kluwer Academic Publishers, Dordrecht, Netherlands (2003).

[17] A. Konak, D. W. Coit and A. E. Smith. *Multi-objective optimization using genetic algorithms: A tutorial*. Reliability Engineering & System Safety, Volume 91, Issue 9, pp. 992–1007 (2006).

[18] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, Cambridge, United Kingdom (2004).

[19] J. Lundgren, M. Rönnqvist and P. Värbrand. *Optimization*. Studentlitteratur, Lund, Sweden (2010).

[20] J. W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, USA (1997).

[21] Q. Y. Kenny, W. Li and A. Sudjianto. *Algorithmic construction of optimal symmetric latin hypercube designs*. Journal of Statistical Planning and Inference, Volume 90, Issue 1, pp. 145–159 (2000).

[22] H-M. Gutmann. *A radial basis function method for global optimization*. Journal of Global Optimization, Volume 19, Issue 3, pp. 201–227 (2001).

[23] modeFRONTIER's homepage, http://www.esteco.com/modefrontier.

[24] R. Kohavi. *A study of cross-validation and bootstrap for accuracy estimation and model selection*. International Joint Conference on Articial Intelligence, Volume 14, Issue 2, pp. 1137–1145 (1995).

[25] L. Debnath and P. Mikusiński. *Introduction to Hilbert Spaces with Applications 2nd Edition*. Academic Press, London, United Kingdom (1999).

[26] Numerical Analysis for Engineering. University of Waterloo, https://ece.uwaterloo.ca/~dwharder/NumericalAnalysis/04LinearAlgebra/posdef/.