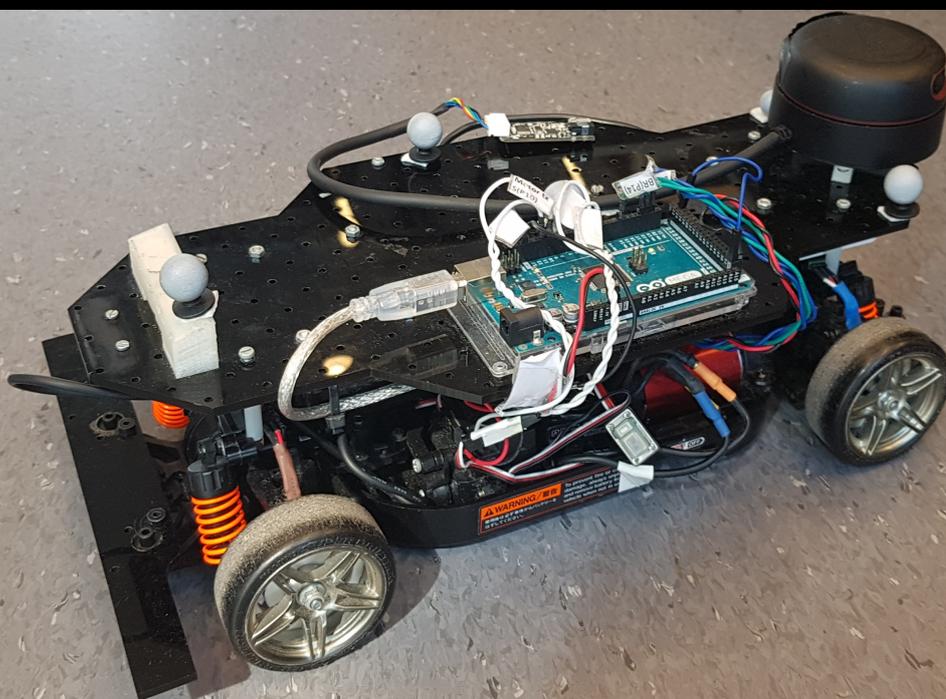


# Lyapunov Stable Path Planning and Control for Autonomous Vehicles

K.J. Haak

Master of Science Thesis





# Lyapunov Stable Path Planning and Control for Autonomous Vehicles

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft  
University of Technology

K.J. Haak

July 3, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of  
Technology



Copyright © Delft Center for Systems and Control (DCSC)  
All rights reserved.



DELFT UNIVERSITY OF TECHNOLOGY  
DEPARTMENT OF  
DELFT CENTER FOR SYSTEMS AND CONTROL (DCSC)

The undersigned hereby certify that they have read and recommend to the Faculty of  
Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis  
entitled

LYAPUNOV STABLE PATH PLANNING AND CONTROL FOR AUTONOMOUS  
VEHICLES

by

K.J. HAAK

in partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE SYSTEMS AND CONTROL

Dated: July 3, 2019

Supervisor(s):

\_\_\_\_\_  
Prof. dr. ir. J. Hellendoorn

\_\_\_\_\_  
Dr. M. Alirezai

Reader(s):

\_\_\_\_\_  
Dr. ir. K. Batselier



---

# Abstract

In modern society cars are one of the most important means of transportation. Unfortunately, many people die in car accidents around the world. Research shows that the number of fatal casualties in car accidents has been increasing for the past decade and that the largest cause of these accidents is the human driver. For this reason, research on fully autonomous vehicles has gained a lot of attention. However, currently autonomous driving is only implemented to reduce the errors of human drivers. More research is necessary in order for fully autonomous vehicles to be implemented and to remove the human driver completely.

A robust navigation algorithm which is able to run in real time is one of the challenges in development of fully autonomous vehicles. Important topics in navigation of autonomous vehicles include the path planner and the motion controller. The path planner finds a path for the vehicle from its current location to the target location. At the same time the path planner avoids obstacles and fulfills the non-holonomic constraints of the autonomous vehicle. The motion controller tries to follow the path the path planner made as close as possible by controlling the vehicle. These two topics influence each other and are therefore dependent. In literature little research is done on integrated algorithms that combine path planning and motion control. Therefore, this thesis will research navigation of autonomous vehicles by using an integrated algorithm that includes both path planning and motion control.

The objective of this thesis is to develop a Lyapunov stable control algorithm that is capable of planning a path for all possible vehicle maneuvers. Besides path planning the proposed algorithm must be capable of controlling the vehicle along this path. Furthermore, the algorithm needs to include obstacles and the non-holonomic dynamics of an autonomous vehicle. The main contribution of this thesis is an integrated path planner and motion controller for navigation of autonomous vehicles. The stability of the proposed algorithm is proven by using the Lyapunov method. Simulation results prove that the algorithm is capable of planning the path and the motion of the autonomous vehicle with non-holonomic constraints and with the presence of obstacles.



---

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1-1	Lyapunov Stable Path Planning . . . . .	1
1-2	Vehicle Model . . . . .	2
1-2-1	Assumptions . . . . .	3
1-3	Research objectives . . . . .	4
1-4	Outline of the thesis . . . . .	4
<b>2</b>	<b>Theoretical Framework</b>	<b>5</b>
2-1	Lyapunov stability . . . . .	5
2-2	Lyapunov theory extended with obstacles . . . . .	7
2-2-1	Mapping vehicle & obstacle to a point world . . . . .	8
2-2-2	Extension of multiple obstacles to a point world . . . . .	9
2-3	Obstacle distance function . . . . .	12
<b>3</b>	<b>Control Strategy</b>	<b>15</b>
3-1	Lyapunov candidate function . . . . .	15
3-2	Controller structure . . . . .	16
3-3	Proposed algorithm for vehicle control . . . . .	17
3-3-1	Potential fields . . . . .	18
3-3-2	Derivative of $\beta$ . . . . .	19
3-4	Proof of Lyapunov stability . . . . .	21
<b>4</b>	<b>Simulation</b>	<b>25</b>
4-1	Vehicle convergence . . . . .	25
4-2	Vehicle convergence from different initial locations . . . . .	27
4-3	Vehicle convergence with obstacles . . . . .	28
4-4	Tuning of the algorithm . . . . .	30

4-4-1	Tuning parameter for velocity . . . . .	30
4-4-2	Tuning parameter for steering . . . . .	31
4-4-3	Tuning parameter for obstacle avoidance . . . . .	33
4-5	Switching dynamics . . . . .	34
<b>5</b>	<b>Conclusion and Recommendations</b>	<b>37</b>
5-1	Conclusion . . . . .	37
5-1-1	Develop an integrated Lyapunov stable algorithm . . . . .	37
5-1-2	Proof the stability of the algorithm . . . . .	38
5-1-3	Evaluate the proposed approach by simulation . . . . .	38
5-2	Recommendations . . . . .	38
5-2-1	Model . . . . .	38
5-2-2	Switching behaviour . . . . .	39
5-2-3	Tuning limits and optimization . . . . .	39
5-2-4	Obstacle avoidance . . . . .	39
5-2-5	Experimental testing . . . . .	39
<b>A</b>	<b>Control Strategy</b>	<b>41</b>
A-1	The control algorithm . . . . .	41
A-1-1	Derivative of $\beta$ calculation . . . . .	41
<b>B</b>	<b>Simulation</b>	<b>45</b>
B-1	Vehicle convergence from different initial positions . . . . .	45
B-2	Vehicle convergence with obstacles . . . . .	49
B-3	Tuning of the controller . . . . .	49
B-3-1	Tuning parameter for velocity . . . . .	49
B-3-2	Tuning parameter for steering . . . . .	50
B-3-3	Tuning parameter for obstacle avoidance . . . . .	50
	<b>Bibliography</b>	<b>51</b>

---

# List of Figures

1-1	Schematic drawing of the kinematic bicycle model . . . . .	3
2-1	Lyapunov and inverse Lyapunov function height map. Left: Lyapunov function going to zero at the origin. Right: Inverse Lyapunov function going to infinity at the origin. . . . .	7
2-2	$h_1$ mapping of the vehicle represented as a circle to a point. Left: Vehicle (blue) and obstacle (black) represented as circles (Mind you that a circle is a special case of an ellipsoid). Right: Vehicle (blue) mapped to a point, obstacle (black) boundary is deformed by the $h_1$ mapping. . . . .	9
2-3	$h_2$ mapping from vehicle point and deformed obstacle to two points in a grid. Left: Vehicle (blue) represented as point at the origin, obstacle (black) deformed under $h_1$ mapping. Right: Vehicle (blue) and obstacle (black) represented as points, vehicle is translated by the $h_2$ mapping. . . . .	10
2-4	$h_1$ mapping with multiple obstacles present. Left: The vehicle (blue) and two obstacles (red and black) are represented as circles. Right: The vehicle (blue) is mapped to a point and the obstacles (red and black) are deformed by the $h_1$ mapping. . . . .	10
2-5	$h_2$ mapping of the first obstacle. The vehicle (blue) and obstacle 1 (black) are represented as points. Obstacle 2 (red) is represented as a deformed boundary. $D_1$ is the distance between the vehicle and the first obstacle. . . . .	11
2-6	$h_2$ mapping of the second obstacle. The vehicle (blue) and obstacle 2 (red) are represented as points. Obstacle 1 (black) is represented as a deformed boundary. $D_2$ is the distance between the vehicle and the second obstacle. . . . .	11
3-1	Controller structure of the Lyapunov algorithm and the autonomous vehicle. Red dotted line: Controller part. Blue dotted line: Vehicle part. . . . .	17
4-1	Convergence of the autonomous vehicle in simulation with starting location $x = 0$ , $y = 5$ , $\theta = 0$ . Blue solid line: path taken by the autonomous vehicle. Green cross: target location. Yellow car: initial location Green car: end location . . . . .	26
4-2	Lyapunov and inverse Lyapunov functions of the simulation with starting location $x = 0$ , $y = 5$ , $\theta = 0$ . Blue solid line: Lyapunov function. Red solid line: Inverse Lyapunov function . . . . .	26

4-3	Derivative of Lyapunov and inverse Lyapunov function of the simulation with starting location $x = 0, y = 5, \theta = 0$ . Blue solid line: derivative of Lyapunov function. Red solid line: derivative of inverse Lyapunov function. . . . .	27
4-4	Different paths from several initial locations reaching the target location. Green cross: target location. Yellow cars: Initial locations. Other colours: Different paths from several initial locations to the target location. . . . .	28
4-5	Convergence of the autonomous vehicle with one obstacle present. Solid blue line: path taken with obstacle avoidance. Dotted black line: path taken without obstacle avoidance. Red circle: obstacle. Green cross: target location. Yellow car: initial location. Green car: end configuration of the vehicle. . . . .	29
4-6	Convergence of the autonomous vehicle in simulation. Left: initial conditions, $x = 0, y = 5, \theta = 0, k_1 = 1e - 5$ . Right: initial conditions, $x = 0, y = 5, \theta = 0, k_1 = 1e - 4$ . Blue solid line: path taken. Green cross: target location. Yellow car: initial location. Green car: end configuration of the vehicle. . . . .	30
4-7	Velocity of the autonomous vehicle in simulation. Left: initial conditions, $x = 0, y = 5, \theta = 0, k_1 = 1e - 5$ . Right: initial conditions, $x = 0, y = 5, \theta = 0, k_1 = 1e - 4$ . . . . .	31
4-8	Convergence of the autonomous vehicle in simulation. Left: initial conditions, $x = 0, y = 5, \theta = 0, k_2 = 0.1$ . Right: initial conditions, $x = 0, y = 5, \theta = 0, k_2 = 0.01$ . Blue solid line: path taken. Green cross: target location. Yellow car: initial location. Green car: end configuration of the vehicle. . . . .	32
4-9	Steering angle of the autonomous vehicle in simulation. Left: initial conditions, $x = 0, y = 5, \theta = 0, k_2 = 0.1$ . Right: initial conditions, $x = 0, y = 5, \theta = 0, k_2 = 0.01$ . . . . .	32
4-10	Convergence of the autonomous vehicle in simulation with obstacle. Left: initial conditions, $x = 0, y = 5, \theta = 0, k = 10$ . Right: initial conditions, $x = 0, y = 5, \theta = 0, k = 1$ . Black dotted line: original path, without obstacle avoidance. Solid blue line: path with obstacle avoidance. Green cross: target location. Yellow car: initial location. Green car: end configuration of the vehicle . . . . .	33
4-11	Velocity of the autonomous vehicle in simulation with initial conditions, $x = 0, y = 5, \theta = 0, k = 10$ and switching behaviour. . . . .	35
4-12	Steering angle of the autonomous vehicle in simulation with initial conditions, $x = 0, y = 5, \theta = 0, k = 10$ and switching behaviour. . . . .	35
B-1	Stability proof of the multiple initial conditions plot. Initial condition: $x = 0, y = 5, \theta = 0$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	45
B-2	Stability proof of the multiple initial conditions plot. Initial condition: $x = -5, y = 0, \theta = 0$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	46
B-3	Stability proof of the multiple initial conditions plot. Initial condition: $x = 5, y = 0, \theta = 0$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	46
B-4	Stability proof of the multiple initial conditions plot. Initial condition: $x = 0, y = -5, \theta = 0$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	46
B-5	Stability proof of the multiple initial conditions plot. Initial condition: $x = -5, y = -5, \theta = 0$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	47
B-6	Stability proof of the multiple initial conditions plot. Initial condition: $x = -5, y = 5, \theta = 0$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	47

B-7	Stability proof of the multiple initial conditions plot. Initial condition: $x = 5$ , $y = 5$ , $\theta = 0$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	47
B-8	Stability proof of the multiple initial conditions plot. Initial condition: $x = 5$ , $y = -5$ , $\theta = 0$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	48
B-9	Stability proof of the example with obstacle avoidance. Initial conditions: $x = 0$ , $y = 5$ , $\theta = 0$ , $k = 10$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	49
B-10	Stability proof of the example with increased velocity tuning. Initial conditions: $x = 0$ , $y = 5$ , $\theta = 0$ , $k1 = 1e - 4$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	49
B-11	Stability proof of the example with decreased steering angle tuning. Initial conditions: $x = 0$ , $y = 5$ , $\theta = 0$ , $k2 = 0.01$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	50
B-12	Stability proof of the example with increased obstacle avoidance. Initial conditions: $x = 0$ , $y = 5$ , $\theta = 0$ , $k = 1$ Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions . . . . .	50



---

# List of Tables

A-1	Parameters appearing in the partial derivatives of $\beta$ . . . . .	43
-----	--	----



---

# Chapter 1

---

## Introduction

In modern society cars are one of the most important means of transportation. Unfortunately, 1.35 million people died last year due to car accidents [1]. This number of fatal casualties in car accidents has been increasing for the past decade. The leading cause of accidents in the United States is the human driver by 94% [2]. Due to this high number of accidents caused by human driver and the large number of accidents itself, research on autonomous driving in the past decade has grown exponentially.

Many companies have started developing either autonomous vehicles or products that can be used in autonomous vehicles. Tesla uses adaptive cruise control, lane detection and automatic lane changing to drive their vehicle in "autopilot" on highways. Waymo, tries to launch a fully autonomous car. This car needs to be able to drive in every road situation possible. One of the important topics in autonomous vehicles is the ability to map a viable path. This path, which extends from the location of the vehicle to the goal location, must be reached without colliding with obstacles or other vehicles and should be able to handle the non-holonomic constraints of the vehicle. Another important topic in navigation is motion control. A motion controller should control the vehicle to follow the generated path of the path planner. In this research an integrated path planner and motion controller will be investigated. First, an overview of Lyapunov based path planning and motion control in literature will be given. Following, the vehicle model that is used in this research will be introduced. Finally, the research objective will be defined and an outline of the research will be given.

### 1-1 Lyapunov Stable Path Planning

In literature, navigation of autonomous vehicles consists of two main parts:

- Path Planning
- Motion control

Lyapunov stable path planning and motion control is a technique where the path and the control inputs to the vehicle are derived using Lyapunov stability. In literature the navigation problem has been approached from several angles. First of all, local algorithms have been researched which handle a single aspect of navigation. These aspects, e.g., parking [3] [4], motion control [5] [6], obstacle avoidance [7] [8] and path planning [9] [10], are limited to their respective task. Local algorithms behave well in their respective field, but since they are dependant they interfere with each other. This means that the vehicle will not converge to the target location, or collide with obstacles. Take for example an algorithm derived with a path planner that doesn't account for non-holonomic constraints of the vehicle. The motion controller for an autonomous vehicle that tries to follow this path can not do this. The non-holonomic constraints interfere with the path following and the vehicle will therefore not reach the target location.

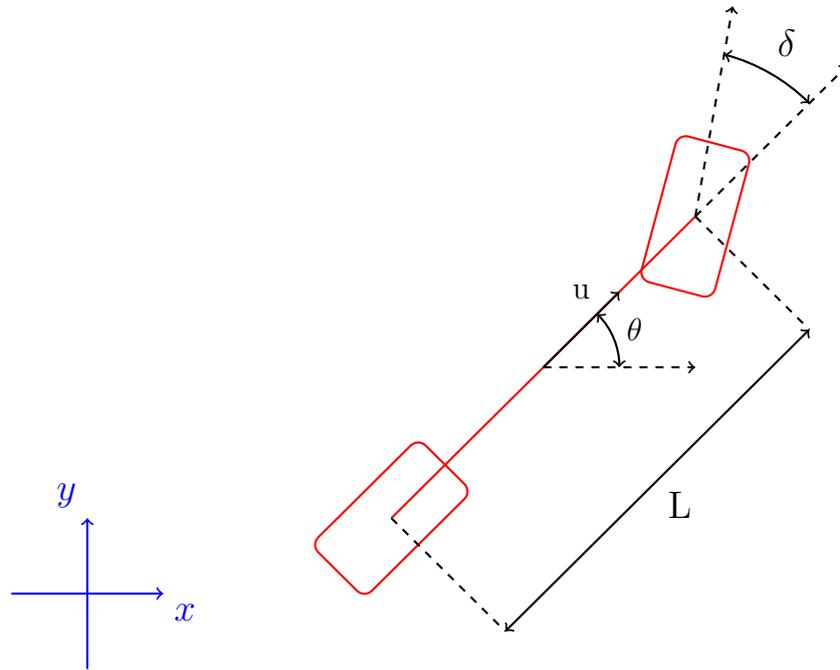
Global algorithms [11] [12] [13] combine local algorithms. The integrated algorithms are capable of path planning and motion control. For full control of the autonomous vehicle the path planning part of the global algorithm must include non-holonomic dynamics and obstacle avoidance. The main advantage of these algorithms is that stability can be tested for the full algorithm instead of just the local parts.

The main focus of this thesis follows the research of Tanner, Loizou and Kyriakopoulos [14]. The focus of this research lies on controlling mobile robots using Lyapunov stable control while avoiding obstacles. This research is chosen since it provides an integrated method for path planning and motion control. The research uses a discontinuous feedback algorithm to achieve the best possible convergence. However, the method is applied for a holonomic robot and in this thesis the work of Tanner et. al. is extended to a car-like vehicle and stability of the proposed algorithm is proven with Lyapunov theory. The next section will explain the vehicle model that is used.

## 1-2 Vehicle Model

Different vehicle models can be used for control of autonomous vehicles. For instance, cartesian coordinate models [15] [16], polar coordinate models [11] [12] [17] [18] and models that calculate the steering angle using the velocity of the vehicle [3] [6] [19]. The vehicle model that is used in this thesis, is based on the kinematic bicycle model with a steering angle that depends on the velocity. The kinematic bicycle model represents a vehicle where two wheels in each axle are lumped into one wheel (Figure 1-1). The inputs to the bicycle model are the velocity  $u$  and the steering angle  $\delta$ . The kinematic bicycle model provides a simple model which models the non-holonomic constraints of a car-like vehicle. The kinematic bicycle model is based on Ackermann geometry and constrained to low velocity and steady state cornering. Furthermore, the vehicle is front wheel steered.

The equations of motion describing this model are given in Equation 1-1.



**Figure 1-1:** Schematic drawing of the kinematic bicycle model

$$\begin{aligned} \dot{x} &= u \cos \theta \\ \dot{y} &= u \sin \theta \\ \dot{\theta} &= \frac{\delta u}{u^2 \frac{K_{us}}{g} + L} \end{aligned} \quad (1-1)$$

Where  $\theta$  is the heading of the vehicle,  $K_{us}$  is the understeer coefficient,  $g$  is the gravitational constant and  $L$  is the wheel base of the vehicle.

### 1-2-1 Assumptions

In this research the following assumptions are considered for the vehicle model:

- Low speed
- Steady state cornering
- No slip
- Steering angles of front wheels are equal
- There is a linear relationship between the lateral forces and the slip angles
- The mass of the model is represented as a point mass in the center of gravity of the vehicle

### 1-3 Research objectives

The aim of this research is to develop an integrated path planner and motion controller for an autonomous vehicle based on Lyapunov stability. The proposed algorithm should be able to determine a viable path and control the vehicle towards the target location. Furthermore, the proposed algorithm should be able to avoid obstacles. The objectives of this thesis are formulated as follows:

- Develop an integrated Lyapunov stable algorithm that:
  - Converges the vehicle to the target location from any initial point.
  - Can avoid obstacles.
  - Can be tuned for different velocity and steering.
- Proof the stability of the algorithm.
- Evaluate the proposed approach by simulation.

### 1-4 Outline of the thesis

The thesis is structured as follows: Chapter 2, describes the theoretical framework for path planning and motion control. Next, the proposed algorithm and its stability proof will be discussed in chapter 3. The evaluation of the proposed method in a simulation environment will be discussed in chapter 4. Finally, conclusions and recommendations for future work will be drawn in chapter 5.

# Theoretical Framework

In this chapter, the theoretical framework of the proposed algorithm will be explained. The first section contains information about the Lyapunov stability and how it is used in this research. The second section, explains how the theory is extended to include obstacles. In the third section, an explanation is given about the quantification of the distance between the vehicle and the different obstacles and how this quantification can be applied to the proposed algorithm.

### 2-1 Lyapunov stability

The objective of the proposed algorithm is to derive the inputs to the model described in chapter 1 using a Lyapunov stable method. In theory this means finding a valid Lyapunov function that accounts for the dynamics of the model and with that function finding a velocity and steering angle for the vehicle. This velocity and steering angle need to be constrained to the limits of the vehicle. Furthermore, the Lyapunov stability must be maintained while using the control inputs. Lyapunov stability begins with the definition of a Lyapunov function  $V(x)$ . Lyapunov theory says that if three conditions are met, the Lyapunov function will guarantee global convergence for the proposed control inputs. The three conditions that need to be met are defined in Equation 2-1.

$$\begin{aligned} V(x) &= 0 \quad \text{if and only if } x = 0 \\ V(x) &> 0 \quad \text{if and only if } x \neq 0 \\ \frac{d}{dt}V(x) &\leq 0 \quad \text{for all values of } x \neq 0 \end{aligned} \tag{2-1}$$

Mind you that for asymptotic stability (and thus global convergence), the third condition must be  $\frac{d}{dt}V(x) < 0$  for all values of  $x \neq 0$ . In this thesis a variant of the Lyapunov function

will be used that is the inverse of the Lyapunov function and it is called  $W(x)$ . This function can be described by Equation 2-2.

$$W(x) = \frac{1}{V(x)} \quad (2-2)$$

The stability of this inverse Lyapunov function is derived using the requirements of the Lyapunov function. The first requirement of  $V(x)$ , states that whenever  $x$  goes to zero  $V(x)$  will also go to zero. For the inverse this means that whenever  $x$  goes to zero  $W(x)$  will go to infinity. The second requirement of  $V(x)$  is the same for  $W(x)$ : whenever  $x$  is not zero  $V(x)$  and thus the inverse should be positive. Mind you that if  $x$  goes to zero  $W(x)$  is also positive and the requirement is slightly different from the original requirement. For the third requirement, the derivative of the Lyapunov function can be taken to investigate what is happening to the derivative of the inverse. The derivative of  $V(x)$  is given in Equation 2-3.

$$\begin{aligned} V(x) &= \frac{1}{W(x)} \\ \dot{V}(x) &= -\frac{1}{W(x)^2} \dot{W}(x) \end{aligned} \quad (2-3)$$

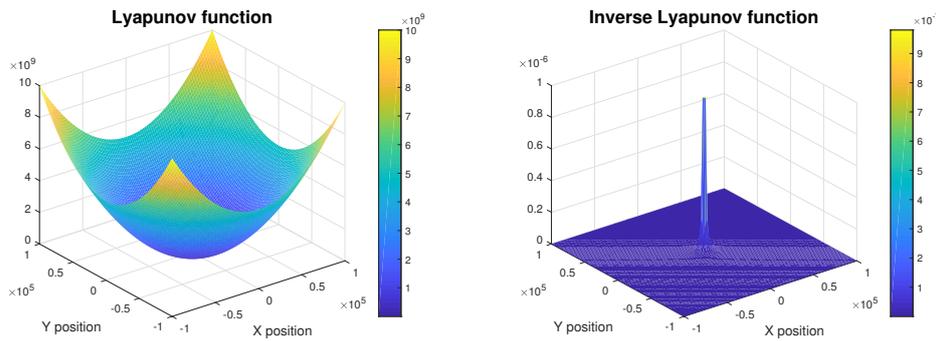
Adding the third requirement of Equation 2-1 leads to Equation 2-4.

$$\begin{aligned} \dot{V}(x) &\leq 0 \\ \dot{V}(x) &= -\frac{1}{W(x)^2} \dot{W}(x) \leq 0 \\ -\dot{V}(x) &= \frac{1}{W(x)^2} \dot{W}(x) \geq 0 \end{aligned} \quad (2-4)$$

Where  $\frac{1}{W(x)^2}$  stays positive due to the squared function. Mind you that division by a negative parameter has changed the direction of the requirement. The three new requirements for inverse Lyapunov stability are defined in Equation 2-5.

$$\begin{aligned} \lim_{x \rightarrow 0} W(x) &= +\infty \\ W(x) &\geq 0 \quad \text{for all } x \\ \frac{d}{dt} W(x) &\geq 0 \quad \text{for all values of } x \neq 0 \end{aligned} \quad (2-5)$$

Since the conditions of inverse Lyapunov stability are derived from Lyapunov stability, it can be stated that there is Lyapunov stability if there is inverse Lyapunov stability. Mind you that for asymptotic stability the last condition needs to be  $\frac{d}{dt} W(x) > 0$  for all values of  $x \neq 0$ . In order to better comprehend the inverse Lyapunov function, an example Lyapunov function is defined in Equation 2-6.



**Figure 2-1:** Lyapunov and inverse Lyapunov function height map. Left: Lyapunov function going to zero at the origin. Right: Inverse Lyapunov function going to infinity at the origin.

$$V(x) = \frac{1}{2}x^2 + \frac{1}{2}y^2 \quad (2-6)$$

The inverse Lyapunov function that can be derived from this function is depicted in Equation 2-7.

$$W(x) = \frac{1}{V(x)} = \frac{2}{x^2 + y^2} \quad (2-7)$$

To depict how these functions behave, they are plotted for a grid of x and y coordinates (Figure 2-1).

It can be examined that the inverse Lyapunov function creates an infinite peak at the origin and that the Lyapunov function goes to zero at the origin. Meaning that the Lyapunov and inverse Lyapunov conditions from Equation 2-1 and Equation 2-5 are met. As previously mentioned, the basis for a Lyapunov stable algorithm is the Lyapunov function  $V(x)$ . Using the change in the Lyapunov function the algorithm can determine a velocity and steering angle that minimizes the Lyapunov function. The real question in controlling an autonomous vehicle using Lyapunov stability is: How can one find a Lyapunov function that can be used to determine  $u$  and  $\delta$  and that decreases to zero, while keeping in mind the obstacles and non-holonomic dynamics of the autonomous vehicle. This question will be answered in the next chapter. However, in this chapter obstacles will be added to the Lyapunov theory.

## 2-2 Lyapunov theory extended with obstacles

Before finding the Lyapunov function, first the obstacles that the autonomous vehicle must avoid need to be discussed. The difficulty in using Lyapunov stability for autonomous vehicles is that the algorithm will always choose the path to the lowest value. This lowest value will not always lie in a clear path from a certain starting location. Think of a parking spot where an autonomous vehicle must park. Other vehicles can be parked close to the parking spot that the autonomous vehicle must reach. Using a regular Lyapunov function the vehicle will just take the path and not account for the other vehicles. But since there can't be any collisions,

there is need for a constraint Lyapunov function that uses only certain regions of the total grid. In order to keep the path planning simple the grid is first mapped to a grid where the vehicle and the obstacles are represented as points. This topology where vehicle and obstacles are reduced to points will be called a "point world". This topology makes stability analysis and thus control easier to derive. The section begins with explaining the point world and how the vehicle and a single obstacle are mapped to it. The mapping will be extended to multiple obstacles in the second subsection. The section ends with an explanation on how to use this mapping concerning control of the autonomous vehicle.

### 2-2-1 Mapping vehicle & obstacle to a point world

The point world mapping is based on the fact that globally converging potential functions can be constructed on sphere worlds [20]. It works by introducing a series of transformations that transform the vehicle and obstacles to single points, creating a point world. The method can be extended to mobile robots [14] in order to create a point world for autonomous vehicles. The method begins with representing the vehicle and obstacle as ellipsoids. In this research a single ellipsoid is used to represent the vehicle and the obstacle. However, multiple ellipsoids that fill the volume of the vehicle and obstacle can also be used in order to better represent the geometry. In this research one ellipsoid is chosen, since it reduces computational time. The boundary of the vehicle can be represented by Equation 2-8.

$$b_r(x, y) = \frac{(x - x_{rc})^2}{a_r^2} + \frac{(y - y_{rc})^2}{b_r^2} - 1 \quad (2-8)$$

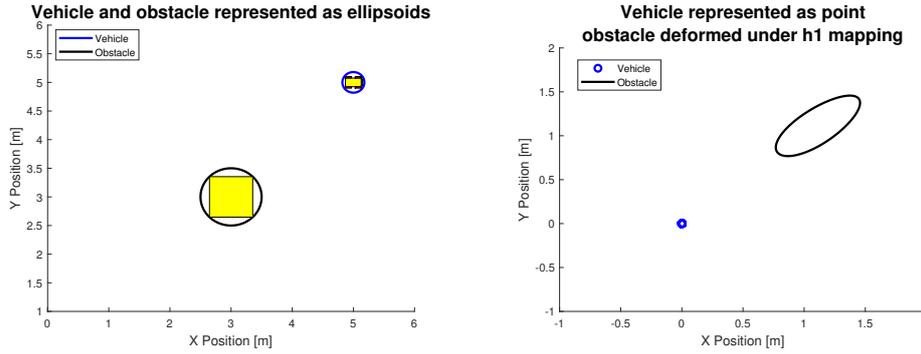
Where  $x_{rc}$  and  $y_{rc}$  are the center coordinates of the vehicle and  $a_r$  and  $b_r$  describe the length and width of the ellipsoid covering the vehicle. The same technique can be used for the obstacles, describing the obstacle by an ellipsoid covering it. The obstacle equation is given in Equation 2-9.

$$b_o(x, y) = \frac{(x - x_{oc})^2}{a_o^2} + \frac{(y - y_{oc})^2}{b_o^2} - 1 \quad (2-9)$$

Where  $x_{oc}$  and  $y_{oc}$  are the center coordinates of the obstacle and  $a_o$  and  $b_o$  describe the length and width of the ellipsoid covering the obstacle. Now using these representations the ellipsoids of the vehicle and the obstacle are mapped to a mapping  $h_1$ , where the vehicle is represented as a point and the obstacle boundary is deformed under this mapping. The equation describing the mapping for the vehicle is depicted in Equation 2-10.

$$h_{1r} = \sqrt{\frac{b_r(x_{rc}, y_{rc})}{b_r(x_{rc}, y_{rc}) + 1}} \begin{bmatrix} x_{rc} \\ y_{rc} \end{bmatrix} \quad (2-10)$$

Mind you that  $h_{1r}$  becomes zero, making the vehicle a point at the origin. The same mapping is used on the boundary of the obstacle ellipse. This creates a deformed boundary for the obstacle. The equation for this deformed boundary is given in Equation 2-11.



**Figure 2-2:**  $h_1$  mapping of the vehicle represented as a circle to a point. Left: Vehicle (blue) and obstacle (black) represented as circles (Mind you that a circle is a special case of an ellipsoid). Right: Vehicle (blue) mapped to a point, obstacle (black) boundary is deformed by the  $h_1$  mapping.

$$h_{1oi} = \sqrt{\frac{b_r(x_{ob}, y_{ob})}{b_r(x_{ob}, y_{ob}) + 1}} \begin{bmatrix} x_{ob} \\ y_{ob} \end{bmatrix} \quad (2-11)$$

Where  $x_{ob}$  and  $y_{ob}$  are the coordinates of the obstacle boundary. The  $i$  parameter denotes the  $i^{th}$  obstacle, multiple obstacles will be discussed in the next subsection. The  $h_1$  mapping is visualized in Figure 2-2 for a single obstacle and the vehicle.

The second mapping from  $h_1$  to  $h_2$  maps the deformed obstacle boundary to a point. The location of the vehicle in the  $h_1$  mapping is translated due to the second mapping. The mapping of the deformed obstacle boundary is done according to Equation 2-12.

$$h_{2oi} = \left( \frac{b_{oi}(x_{oc}, y_{oc})}{b_{oi}(x_{oc}, y_{oc}) + 1} \right) (h_{1oi} - h_{coi}) + h_{coi} \quad (2-12)$$

Where  $h_{coi}$  is a reference point in the deformed obstacle interior, now representing the obstacle point. In this research the reference point is the deformed center of the obstacle since that will always lie in the deformed obstacle interior. The translation of the vehicle point due to the second mapping can be expressed by Equation 2-13.

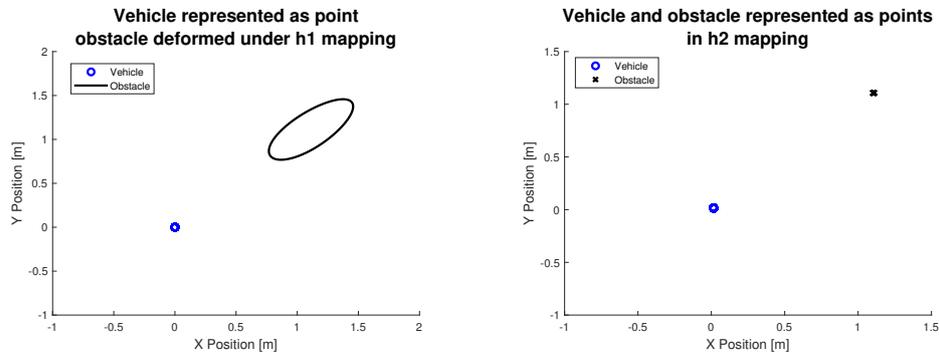
$$h_{2ri} = \left( \frac{b_{oi}(x_{rc}, y_{rc})}{b_{oi}(x_{rc}, y_{rc}) + 1} \right) (h_{1r} - h_{coi}) + h_{coi} \quad (2-13)$$

The  $h_2$  mapping is visualized in Figure 2-3.

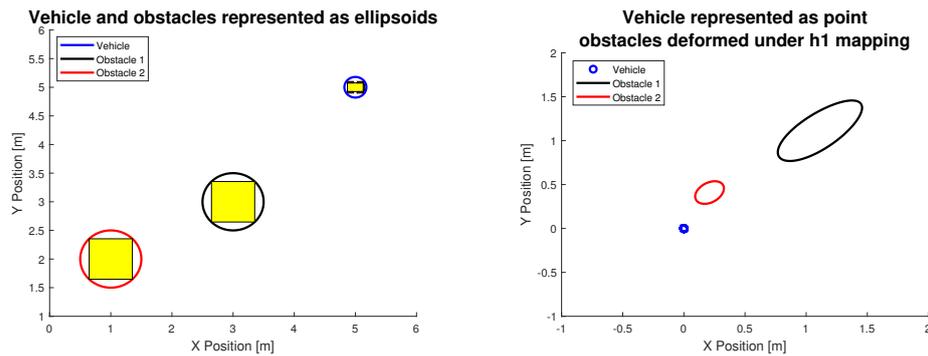
It can be observed that both vehicle and obstacle are reduced to points. In the next section the theory is extended to multiple obstacles.

## 2-2-2 Extension of multiple obstacles to a point world

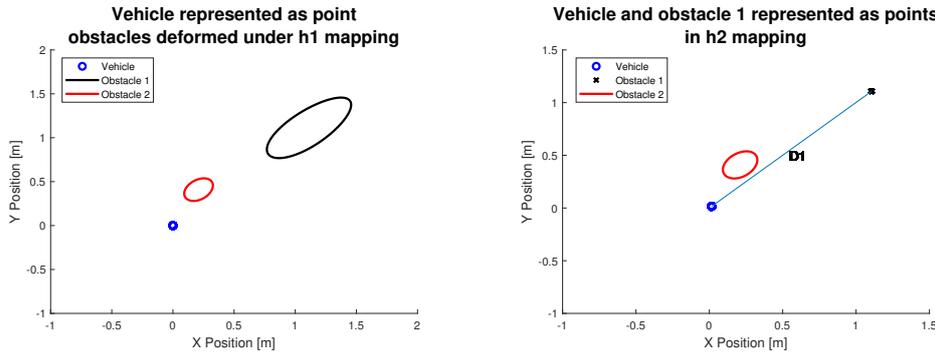
Now let's switch from one obstacle to multiple obstacles. The  $h_1$  mapping is done in the same way as with a single obstacle. The deformation of the vehicle to a point creates a number



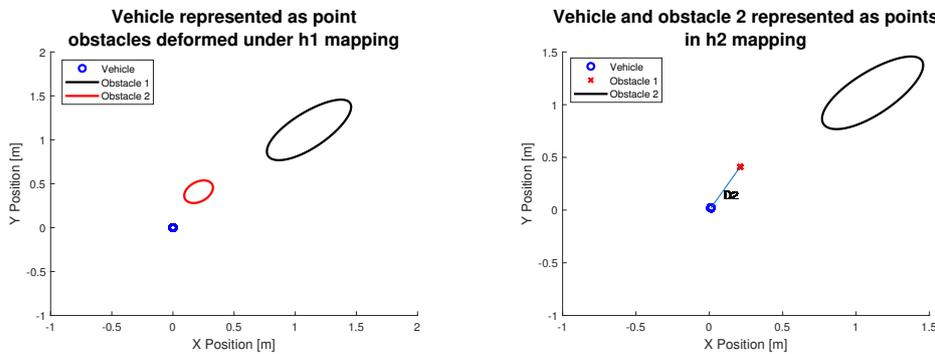
**Figure 2-3:**  $h_2$  mapping from vehicle point and deformed obstacle to two points in a grid. Left: Vehicle (blue) represented as point at the origin, obstacle (black) deformed under  $h_1$  mapping. Right: Vehicle (blue) and obstacle (black) represented as points, vehicle is translated by the  $h_2$  mapping.



**Figure 2-4:**  $h_1$  mapping with multiple obstacles present. Left: The vehicle (blue) and two obstacles (red and black) are represented as circles. Right: The vehicle (blue) is mapped to a point and the obstacles (red and black) are deformed by the  $h_1$  mapping.



**Figure 2-5:**  $h_2$  mapping of the first obstacle. The vehicle (blue) and obstacle 1 (black) are represented as points. Obstacle 2 (red) is represented as a deformed boundary.  $D_1$  is the distance between the vehicle and the first obstacle.



**Figure 2-6:**  $h_2$  mapping of the second obstacle. The vehicle (blue) and obstacle 2 (red) are represented as points. Obstacle 1 (black) is represented as a deformed boundary.  $D_2$  is the distance between the vehicle and the second obstacle.

of deformed obstacle boundaries equal to the number of obstacles. The deformation for two obstacles is visualized in Figure 2-4.

These two boundary deformations create two translations for the vehicle point in the  $h_2$  mapping. But instead of using two transformations in series, the transformations are done in parallel. This creates two different vehicle representations in  $h_2$ , one for each obstacle. This behavior is sketched in Figure 2-5 for obstacle one and Figure 2-6 for obstacle 2.

It can be observed that there are two distances  $D_1$  and  $D_2$ .  $D_1$  represents the distance between the vehicle point and obstacle 1. Mind you that the vehicle point is translated due to the  $h_2$  transformation of obstacle 1.  $D_2$  represents the distance between the vehicle point and the second obstacle transformation. The second  $h_2$  transformation done by obstacle 2 translates the vehicle point to a different location than the first  $h_2$  transformation. The next section describes how these distances are used in the proposed algorithm.

### 2-3 Obstacle distance function

In order to use the mapping described in the previous section a quantification is needed that links the distances from the vehicle point representations to the obstacle points. This quantification is best described by the distance from the vehicle to a certain obstacle but weighed on how far an obstacle is. This makes that the measure takes into account multiple obstacles while weighing closer obstacles more than obstacles far away. This means that the car will divert its path more when obstacles are closer, in opposite of diverting the path less when obstacles are far away. The measure used for this quantification is the normalized euclidean distance. The normalized euclidean distance is calculated by Equation 2-14.

$$\hat{d} = \sqrt{\sum_{i=1}^v \left( \frac{(P_{1i} - P_{2i})^2}{v} \right)} \quad (2-14)$$

Where  $P_{1i}$  and  $P_{2i}$  are two different items to be compared. Furthermore  $v$  is the total number of items present. First, the euclidean distance is used to obtain a final location of the obstacles for the different vehicle representations in the  $h_2$  point world. The equation representing this calculation is given in Equation 2-15.

$$z_{oj} = \prod_{i \in R} \hat{d}_i(h_{2ri}, h_{2oj})(z_r - h_{2oj}) + z_r \quad (2-15)$$

Where  $z_r$  is the representation of the whole vehicle, thus the location of the vehicle in the  $h_2$  mapping.  $\hat{d}_i(h_{2ri}, h_{2oj})$  is the normalized euclidean distance of the  $i^{th}$  vehicle representation. Mind you that this normalized euclidean distance can be calculated using the distances  $D_1$  and  $D_2$ . The equation can then be expressed as Equation 2-16

$$\hat{d}_i(h_{2ri}, h_{2oj}) = \sqrt{\sum_{i=1}^v \left( \frac{D_i^2}{v} \right)} \quad (2-16)$$

With the final location of each obstacle a final measure can be calculated that can be used in the inverse Lyapunov function. The equation is given in Equation 2-17.

$$\beta = \prod_{j \in O} \hat{d}_j(z_{oj}, z_r) \quad (2-17)$$

One other question arises when using the mapping from a grid with ellipsoids to a grid with points, why is this done? The answer lies within the different mapping equations. The mappings make each coordinate on the boundary of the obstacle zero, while the coordinates within the obstacle are infinity. This means that if used in an inverse Lyapunov function, the obstacle location will also be zero at that location. If using the Lyapunov function for control these points will become infinity. Due to this phenomenon the value of the Lyapunov function will become bigger when going to these points. If the Lyapunov function is used for control this means that the control inputs will steer the vehicle away from this point. Thus the vehicle will always divert from these points meaning that in theory the autonomous

vehicle will avoid the obstacles due to the point world mapping. It can be concluded that the final measure  $\beta$  minimizes the inverse Lyapunov function and thus maximizes the Lyapunov function in the vicinity of the obstacle.

In this chapter, the Lyapunov theory that can be used for path planning is discussed. Secondly, a mapping is determined to map the autonomous vehicle and obstacles to a point world. Furthermore, a measure for quantifying obstacle distances in this mapping has been determined. The next chapter will explain how the mapping and especially  $\beta$  will be used to control the vehicle towards the target location.



---

## Chapter 3

---

# Control Strategy

In this chapter the control strategy for the autonomous vehicle will be explained. The mapping in chapter 2 will be used to model the obstacles in the variable  $\beta$ . With this parameter the inputs to the model described in chapter 1 will be calculated. The goal of the algorithm is to use the vehicle location  $x$ ,  $y$  and heading  $\theta$  and the parameter for the location of the obstacles  $\beta$  to determine the control inputs for the autonomous vehicle. The two control inputs that need to be calculated are the vehicles longitudinal velocity  $u$  and the steering wheel input  $\delta$ . The autonomous vehicle will converge to the target location whenever the inverse Lyapunov stability conditions from the previous chapter are met. The first section of this chapter introduces a candidate inverse Lyapunov function and rewrites the input model from the introduction. The second section defines the controller structure. The third section will define the proposed algorithm and show how it is derived. The last section will handle the proof of the algorithm and show that the candidate inverse Lyapunov function meets the stability requirements.

### 3-1 Lyapunov candidate function

As mentioned in the previous chapter there are three requirements for the control of the autonomous vehicle.

- An inverse Lyapunov function is needed that goes to infinity whenever the car reaches the origin.
- The inverse Lyapunov function should go to zero whenever an obstacle is present at that location.
- The derivative of the inverse Lyapunov function should decrease whenever the car moves towards an obstacle.

The proposed candidate inverse Lyapunov function that meets these requirements is given in Equation 3-1.

$$W(z) = \frac{|x|\beta^{\frac{1}{k}}}{\|z\|^2} \quad (3-1)$$

Where  $k$  is an optimization parameter determining the weight of  $\beta$ . Furthermore  $z$  is the model parameter describing the configuration of the vehicle ( $z = x, y, \theta$ ). The 2 norm of  $z$  is given in Equation 3-2.

$$\|z\|^2 = x^2 + y^2 + \lambda\theta^2 \quad (3-2)$$

Where  $\lambda$  is a positive weighting parameter.  $\lambda$  makes sure that the location is more important than the heading. This makes sure that the vehicle can move freely and only goes to  $z = 0$  in the origin. The kinematic bicycle model from chapter 1 can be rewritten as Equation 3-3 using  $z$ .

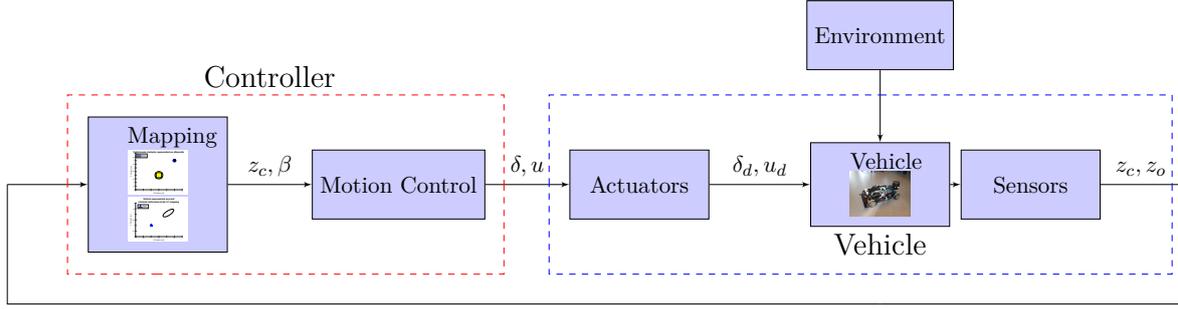
$$\begin{aligned} \dot{z} &= f(z)u + g(z)\delta \\ \dot{z} &= \begin{bmatrix} \cos\theta & \sin\theta & 0 \end{bmatrix}^T u + \begin{bmatrix} 0 & 0 & \frac{u}{u^2 \frac{K_{us}}{g} + L} \end{bmatrix}^T \delta \end{aligned} \quad (3-3)$$

Mind you that the inputs are not independent because the third state is influenced by both the velocity and the steering angle. This means that the calculation of the steer angle will have to account for the influence of the velocity. This is the non-holonomic constraint problem that is present in autonomous vehicles and needs to be accounted for by the algorithm.

## 3-2 Controller structure

This section will explain the control structure of the autonomous vehicle. The full model consists of two parts. The first is the controller and the second is the vehicle. The controller consists of a mapping part and a motion control part. The vehicle consists of actuators, the vehicle itself and the sensors. The vehicle will be simulated with the model from chapter 1 thus the actuators and sensors are not taken into account. The controller structure is graphically depicted in Figure 3-1.

The inputs to the controller are the configuration vectors  $z_c$  and  $z_o$  of the vehicle and the obstacle. These inputs are led to the mapping. In the mapping the point world transformations are done and the distance measure  $\beta$  is calculated. The distance measure and the configuration vector of the vehicle are then send to the motion controller. The motion controller determines the inputs  $u$  and  $\delta$  and sends them to the actuators of the vehicle. The actuators move and determine delayed inputs  $\delta_d$  and  $u_d$ . These are the real inputs that the vehicle is moved with. The inputs are send to the wheels and the vehicle moves. After moving, sensors will return the new position of the vehicle and the obstacles and send them to the controller. An external disturbance can be modelled as an environment block. This block can for instance model the influences of the road or the weather. The previous chapter defined the mapping part of the controller. The next section describes the motion control part.



**Figure 3-1:** Controller structure of the Lyapunov algorithm and the autonomous vehicle. Red dotted line: Controller part. Blue dotted line: Vehicle part.

### 3-3 Proposed algorithm for vehicle control

To control the vehicle the technique mentioned in chapter 2 will be used. The candidate inverse Lyapunov function will be divided into two sections. The first will be  $x$  is greater or equal to zero, the second will be  $x$  is smaller than zero. This is depicted in Equation 3-4. This division makes sure that the inverse Lyapunov function is always positive for all values of  $x$ .

$$W(z) = \begin{cases} \frac{x\beta^{\frac{1}{k}}}{\|z\|^2} & \text{if } x \geq 0 \\ \frac{-x\beta^{\frac{1}{k}}}{\|z\|^2} & \text{if } x < 0 \end{cases} \quad (3-4)$$

The control inputs that can be determined with this candidate inverse Lyapunov function, are given in Equation 3-5.

$$u = k_1 \text{sign}(f_x \cos \theta + f_y \sin \theta) (f_x^2 + f_y^2 + f_\theta^2) \quad (3-5)$$

$$\delta = \begin{cases} k_2 (\theta_d - \theta) \left( \frac{L + \frac{K_{us} u^2}{g}}{u} \right) & \text{if } p \geq 0 \\ -u \frac{1}{f_\theta} \left( f_x \cos(\theta) + f_y \sin(\theta) \right) \left( \frac{L + \frac{K_{us} u^2}{g}}{u} \right) & \text{if } p < 0 \end{cases} \quad (3-6)$$

The first unknown parameter  $\theta_d$  is defined in Equation 3-7.

$$\theta_d = \arctan2(-\text{sign}(x)f_y, -\text{sign}(x)f_x) \quad (3-7)$$

Where  $\arctan2$  is the second argument arc tangent. This function is used to account for the change in heading between positive and negative values. The sign function is positive one if the input is greater or equal to zero and negative one for negative inputs. This is depicted in Equation 3-8.

$$\text{sign}(h) = \begin{cases} 1 & \text{if } h \geq 0 \\ -1 & \text{if } h < 0 \end{cases} \quad (3-8)$$

Furthermore, the switching parameter  $p$  is defined in Equation 3-9.

$$p = u(f_x \cos(\theta) + f_y \sin(\theta)) + k_2(\theta_d - \theta)f_\theta \quad (3-9)$$

Mind you that  $k_1$  and  $k_2$  are optimization parameters used for limiting the velocity and the steering input respectively. The remaining parameters will be described in the following two subsections. In the first subsection the parameters  $f_x$ ,  $f_y$  and  $f_\theta$  which are called potential fields will be described. In the second subsection the derivative of  $\beta$  which are used in the potential fields will be discussed.

### 3-3-1 Potential fields

The potential fields that are used in the control inputs can be defined as Equation 3-10.

$$\begin{aligned} f_x &= \frac{\partial W(z)}{\partial x} \|z\|^4 \\ f_y &= \frac{\partial W(z)}{\partial y} \|z\|^4 \\ f_\theta &= \frac{\partial W(z)}{\partial \theta} \|z\|^4 \end{aligned} \quad (3-10)$$

First, the derivatives of  $W(z)\|z\|^4$  will be calculated. This is done since it cancels out the denominator, but keeps the dynamics of the inverse Lyapunov stability intact. The equations describing the potential fields are expressed in Equation 3-11.

$$\begin{aligned}
f_x = \|z\|^4 \frac{\partial W(z)}{\partial x} &= \begin{cases} \frac{\|z\|^2 \left( \beta^{\frac{1}{k}} + x \frac{1}{k} \frac{\partial \beta}{\partial x} \left( \frac{1}{k} - 1 \right) \right) - 2x^2 \beta^{\frac{1}{k}}}{\|z\|^4} = & \\ \|z\|^2 \left( \beta^{\frac{1}{k}} + x \frac{1}{k} \frac{\partial \beta}{\partial x} \left( \frac{1}{k} - 1 \right) \right) - 2x^2 \beta^{\frac{1}{k}} & \text{if } x \geq 0 \\ \frac{\|z\|^2 \left( -\beta^{\frac{1}{k}} - x \frac{1}{k} \frac{\partial \beta}{\partial x} \left( \frac{1}{k} - 1 \right) \right) + 2x^2 \beta^{\frac{1}{k}}}{\|z\|^4} = & \\ \|z\|^2 \left( -\beta^{\frac{1}{k}} - x \frac{1}{k} \frac{\partial \beta}{\partial x} \left( \frac{1}{k} - 1 \right) \right) + 2x^2 \beta^{\frac{1}{k}} & \text{if } x < 0 \end{cases} \quad (3-11) \\
f_y = \|z\|^4 \frac{\partial W(z)}{\partial y} &= \begin{cases} \frac{\|z\|^2 \left( x \frac{1}{k} \frac{\partial \beta}{\partial y} \left( \frac{1}{k} - 1 \right) \right) - 2xy \beta^{\frac{1}{k}}}{\|z\|^4} = & \\ \|z\|^2 \left( x \frac{1}{k} \frac{\partial \beta}{\partial y} \left( \frac{1}{k} - 1 \right) \right) - 2xy \beta^{\frac{1}{k}} & \text{if } x \geq 0 \\ \frac{\|z\|^2 \left( -x \frac{1}{k} \frac{\partial \beta}{\partial y} \left( \frac{1}{k} - 1 \right) \right) + 2xy \beta^{\frac{1}{k}}}{\|z\|^4} = & \\ \|z\|^2 \left( -x \frac{1}{k} \frac{\partial \beta}{\partial y} \left( \frac{1}{k} - 1 \right) \right) + 2xy \beta^{\frac{1}{k}} & \text{if } x < 0 \end{cases} \\
f_\theta = \|z\|^4 \frac{\partial W(z)}{\partial \theta} &= \begin{cases} \|z\|^4 \frac{-2\lambda\theta\beta^{\frac{1}{k}}}{\|z\|^4} = -2\lambda\theta\beta^{\frac{1}{k}} & \text{if } x \geq 0 \\ \frac{2x\lambda\theta\beta^{\frac{1}{k}}}{\|z\|^4} = 2x\lambda\theta\beta^{\frac{1}{k}} & \text{if } x < 0 \end{cases}
\end{aligned}$$

The term  $\|z\|^4$  drops out providing an easier method for control. The next subsection will handle the calculation of the derivative of  $\beta$ .

### 3-3-2 Derivative of $\beta$

The potential fields of  $W(z)$  contain the partial derivative of  $\beta$  to  $x$ ,  $y$  and  $\theta$ . However, in the last chapter  $\beta$  was defined to be in the  $h_2$  mapping. In order to get the partial derivatives of  $\beta$  it needs to be expressed in the  $z$  mapping. To do this  $\beta$  is rewritten with the inverse mapping from the last chapter. Remember that  $\beta$  expressed in the  $h_2$  mapping is given as Equation 3-12.

$$\beta_{h2} = \sqrt{\sum_j \frac{(z_{oj} - z_r)^2}{n_{oj}}} \quad (3-12)$$

And that  $z_o(j)$  is defined as in Equation 3-13.

$$z_{oj} = \sqrt{\sum_i \frac{(h_{2ri} - h_{2oj})^2}{n_{ri}}} (z_{rj} - h_{2oj}) + z_{rj} \quad (3-13)$$

If Equation 3-13 is substituted in Equation 3-12 it can be derive to Equation 3-14.

$$\begin{aligned}
\beta_{h2} &= \sqrt{\sum_j \frac{\left(\sqrt{\sum_i \frac{(h_{2ri} - h_{2oj})^2}{n_{ri}}}(z_{rj} - h_{2oj}) + z_{rj} - z_{rj}\right)^2}{n_{oj}}} \\
&= \sqrt{\sum_j \frac{\left(\sqrt{\sum_i \frac{(h_{2ri} - h_{2oj})^2}{n_{ri}}}(z_{rj} - h_{2oj})\right)^2}{n_{oj}}} \tag{3-14}
\end{aligned}$$

$h_{2o}$  and  $h_{2r}$  from the previous chapter can be substituted in the previous equation. The two equations are repeated in Equation 3-15 and Equation 3-16.

$$h_{2o} = \frac{b_o(x_{oc}, y_{oc})}{b_o(x_{oc}, y_{oc}) + 1}(h_{1o} - h_{co}) + h_{co} \tag{3-15}$$

$$h_{2r} = \frac{b_o(x_{rc}, y_{rc})}{b_o(x_{rc}, y_{rc}) + 1}(h_{1r} - h_{co}) + h_{co} \tag{3-16}$$

With these equations  $\beta_{h2}$  can be rewritten into  $\beta_{h1}$  which is given in Equation 3-17.

$$\begin{aligned}
\beta_{h1} &= \sqrt{\sum_j \frac{(F_1 F_2)^2}{n_{oj}}} \tag{3-17} \\
F_1 &= \sqrt{\sum_i \frac{\left(\frac{b_{oi}(x_{rc}, y_{rc})}{b_{oi}(x_{rc}, y_{rc}) + 1}(h_{1ri} - h_{coj}) - \frac{b_{oj}(x_{oc}, y_{oc})}{b_{oj}(x_{oc}, y_{oc}) + 1}(h_{coj} - h_{1oj})\right)^2}{n_{ri}}} \\
F_2 &= \left(z_{rj} - \left(\frac{b_{oj}(x_{oc}, y_{oc})}{b_{oj}(x_{oc}, y_{oc}) + 1}(h_{1oj} - h_{coj}) + h_{coj}\right)\right)
\end{aligned}$$

$h_{1ri}$  and  $h_{1oj}$  can be substituted for their respective formula's defined in Equation 3-18 and Equation 3-19. This results in the respective formula for  $\beta_{h1}$  in the z coordinate system.  $\beta_z$  is defined in Equation 3-20.

$$h_{1r} = \sqrt{\frac{b_r(x_{rc}, y_{rc})}{b_r(x_{rc}, y_{rc}) + 1}} z_r \tag{3-18}$$

$$h_{1o} = \sqrt{\frac{b_r(x_{ob}, y_{ob})}{b_r(x_{ob}, y_{ob}) + 1}} z_o \tag{3-19}$$

$$\beta_z = \sqrt{\sum_j \frac{(F_1 F_2)^2}{n_{oj}}} \tag{3-20}$$

Where  $F_1$  and  $F_2$  are given in Equation 3-21 and Equation 3-22.

$$F_1 = \sqrt{\sum_i \frac{\left( \frac{b_{oi}(x_{rc}, y_{rc})}{b_{oi}(x_{rc}, y_{rc})+1} \left( \sqrt{\frac{b_{ri}(x_{rc}, y_{rc})}{b_{ri}(x_{rc}, y_{rc})+1}} z_{ri} - h_{coj} \right) - \frac{b_{oj}(x_{oc}, y_{oc})}{b_{oj}(x_{oc}, y_{oc})+1} \left( h_{coj} - \sqrt{\frac{b_{rj}(x_{ob}, y_{ob})}{b_{rj}(x_{ob}, y_{ob})+1}} z_{oj} \right) \right)^2}{n_{ri}}} \quad (3-21)$$

$$F_2 = \left( z_{rj} - \left( \frac{b_{oj}(x_{ob}, y_{ob})}{b_{oj}(x_{ob}, y_{ob})+1} \left( \sqrt{\frac{b_{rj}(x_{ob}, y_{ob})}{b_{rj}(x_{ob}, y_{ob})+1}} z_o - h_{coj} \right) + h_{coj} \right) \right) \quad (3-22)$$

With the formula's for  $b_o$  and  $b_r$  from the previous chapter,  $\beta_z$  is fully known. The derivative can now be calculated and used in Equation 3-10. The worked out version of  $\beta$  with the equations from the previous chapter and the partial derivatives of  $\beta$  to the respective function input for one obstacle, can be found in the appendix in subsection A-1-1. Now that the control inputs to the vehicle that are derived using the inverse Lyapunov function are known, the vehicle can be controlled. The next section will prove that the proposed algorithm is stable according to Lyapunov theory.

### 3-4 Proof of Lyapunov stability

In order to proof that the proposed algorithm in the previous section is stable the candidate inverse Lyapunov function is tested against the conditions proposed in chapter 2. In order to test this the inverse Lyapunov function and the derivative of the inverse Lyapunov function need to be known. Since the function itself is already known only the derivative needs to be calculated. Recall the rewritten model in Equation 3-3 from the first section of this chapter. Taking the Lie derivatives with respect to the vector fields  $f(z)$  and  $g(z)$  results in the derivative of the inverse candidate Lyapunov function. The derivation is depicted in Equation 3-23.

$$\begin{aligned} \dot{W} &= uL_{f(x)}W_j(z) + \delta L_{g(x)}W_j(z) \\ &= u \left( f(x)^T \frac{dW}{dz} \right) + \delta \left( g(x)^T \frac{dW}{dz} \right) \end{aligned} \quad (3-23)$$

Mind you that there will be multiple derivatives due to the two control parts for the sign of  $x$ . Recall that the derivatives of  $W(z)$  are the potential fields divided by  $\|z\|^4$ . Using this the terms in Equation 3-23 can be substituted to Equation 3-24.

$$\begin{aligned} \dot{W} &= u \left( \begin{bmatrix} \cos \theta & \sin \theta & 0 \end{bmatrix} \begin{bmatrix} \frac{f_x}{\|z\|^4} \\ \frac{f_y}{\|z\|^4} \\ \frac{f_\theta}{\|z\|^4} \end{bmatrix} \right) + \delta \left( \begin{bmatrix} 0 & 0 & \frac{u}{u^2 \frac{K_{us}}{g} + L} \end{bmatrix} \begin{bmatrix} \frac{f_x}{\|z\|^4} \\ \frac{f_y}{\|z\|^4} \\ \frac{f_\theta}{\|z\|^4} \end{bmatrix} \right) \\ &= u \left( \frac{f_x \cos \theta + f_y \sin \theta}{\|z\|^4} \right) + \delta \left( \frac{u f_\theta}{(u^2 \frac{K_{us}}{g} + L) \|z\|^4} \right) \end{aligned} \quad (3-24)$$

Using the control inputs from the last section the derivatives can be rewritten to Equation 3-25.

$$\dot{W} = \begin{cases} k_1 \text{sign}(f_x \cos \theta + f_y \sin \theta) (f_x^2 + f_y^2 + f_\theta^2) \frac{f_x \cos \theta + f_y \sin \theta}{\|z\|^4} \\ + k_2 (\theta_d - \theta) \left( \frac{L + \frac{K_{us}}{g} u^2}{u} \right) \frac{u f_\theta}{(u^2 \frac{K_{us}}{g} + L) \|z\|^4} & \text{if } p \geq 0 \\ k_1 \text{sign}(f_x \cos \theta + f_y \sin \theta) (f_x^2 + f_y^2 + f_\theta^2) \frac{f_x \cos \theta + f_y \sin \theta}{\|z\|^4} \\ - u \left( \frac{1}{f_\theta} \right) (f_x \cos(\theta) + f_y \sin(\theta)) \left( \frac{L + \frac{K_{us}}{g} u^2}{u} \right) \frac{u f_\theta}{(u^2 \frac{K_{us}}{g} + L) \|z\|^4} & \text{if } p < 0 \end{cases} \quad (3-25)$$

These equations can be rewritten minding that a sign function multiplied with the same function makes it the absolute representation. Furthermore, due to the form of the proposed control input  $\delta$  the non-holonomic dynamics  $\frac{u}{u^2 \frac{K_{us}}{g} + L}$  drop out of the equation. The rewritten equation is depicted in Equation 3-26.

$$\dot{W} = \begin{cases} k_1 (f_x^2 + f_y^2 + f_\theta^2) \frac{|f_x \cos \theta + f_y \sin \theta|}{\|z\|^4} + k_2 (\theta_d - \theta) \frac{f_\theta}{\|z\|^4} & \text{if } p \geq 0 \\ k_1 (f_x^2 + f_y^2 + f_\theta^2) \frac{|f_x \cos \theta + f_y \sin \theta|}{\|z\|^4} - u (f_x \cos(\theta) + f_y \sin(\theta)) \frac{1}{\|z\|^4} & \text{if } p < 0 \end{cases} \quad (3-26)$$

Since the proposed control input  $\delta$  is dependant on  $u$ ,  $\dot{W}$  can be rewritten again. This substitution is depicted in Equation 3-27.

$$\dot{W} = \begin{cases} k_1 (f_x^2 + f_y^2 + f_\theta^2) \frac{|f_x \cos \theta + f_y \sin \theta|}{\|z\|^4} + k_2 (\theta_d - \theta) \frac{f_\theta}{\|z\|^4} & \text{if } p \geq 0 \\ k_1 (f_x^2 + f_y^2 + f_\theta^2) \frac{|f_x \cos \theta + f_y \sin \theta|}{\|z\|^4} \\ - k_1 \text{sign}(f_x \cos \theta + f_y \sin \theta) (f_x^2 + f_y^2 + f_\theta^2) (f_x \cos(\theta) + f_y \sin(\theta)) \frac{1}{\|z\|^4} & \text{if } p < 0 \end{cases} \quad (3-27)$$

Mind you that the sign function multiplied with the same function is again the absolute representation of this term. This makes the second part of the function equal to it's first part but with opposite sign, thus eliminating both terms and resulting in zero. This is depicted in Equation 3-28.

$$\dot{W} = \begin{cases} k_1 (f_x^2 + f_y^2 + f_\theta^2) \frac{|f_x \cos \theta + f_y \sin \theta|}{\|z\|^4} + k_2 (\theta_d - \theta) \frac{f_\theta}{\|z\|^4} & \text{if } p \geq 0 \\ 0 & \text{if } p < 0 \end{cases} \quad (3-28)$$

The first equation of  $\dot{W}$  can be rewritten using that the top part is the same as the switching parameter  $p$ . This is depicted in Equation 3-29.

$$\dot{W} = \begin{cases} \frac{p}{\|z\|^4} & \text{if } p \geq 0 \\ 0 & \text{if } p < 0 \end{cases} \quad (3-29)$$

With the derivatives known the Lyapunov stability can be tested according to the requirements in Equation 2-5 in the last chapter. The first requirement is fulfilled by the fact that

whenever  $z$  goes to zero the norm of  $\|z\|^4$  will also go to zero. This means division by this parameter makes the candidate inverse Lyapunov function infinite. Again by using the first part of  $W(z)$  if  $x$  is positive or zero and the second part of  $W(z)$  if  $x$  is negative, the top part will remain positive and thus  $W(z)$  will go to positive infinity for  $z$  going to zero.

The second requirement is satisfied, because when  $x$  is positive the first part of  $W(z)$  will be used and when  $x$  is negative the second part of  $W(z)$  will be used. The other term  $\beta$  is a quantification of the distance between the vehicle and the obstacle. Distance will always be a positive measure and therefore doesn't influence the stability. Furthermore, the 4-norm,  $\|z\|^4$  will always be positive, due to the fact that even norms are always positive.

The derivatives can be used to check the third requirement. The derivative should remain positive for all values of  $z \neq 0$  in order to have asymptotic stability which in term is needed for full convergence. First of all mind you that the first part of  $\dot{W}(z)$  is used whenever  $p$  is greater or equal to zero. This means that the first part of  $\dot{W}(z)$  will also be positive or zero. Also mind you that the norm  $\|z\|^4$  will remain positive. Furthermore, whenever  $p$  is negative, the second part of  $\dot{W}(z)$  will be used and the derivative will be zero. Thus the derivative of  $W(z)$  will be greater or equal to zero. Recall from chapter 2 that this means that there is Lyapunov stability but not asymptotic stability. However, mind you that due to the division of  $W(z)$  and also the division of  $\delta$  that this will not happen unless  $u = \delta = 0$ . This will only happen whenever the vehicle reaches the target location. This means that the control algorithm is asymptotically stable for every point but the target location. In practice this means that the vehicle will converge infinitely close to the target location but will never fully reach it.

This chapter has provided the proposed algorithm that can be used to control an autonomous vehicle to its target location from any initial location. This target location can be reached while avoiding obstacles in the vicinity of the vehicle. Furthermore, this chapter proved that the algorithm is Lyapunov stable and asymptotically stable for all locations except the target location. In the next chapter the algorithm will be evaluated in simulation.



---

# Chapter 4

---

## Simulation

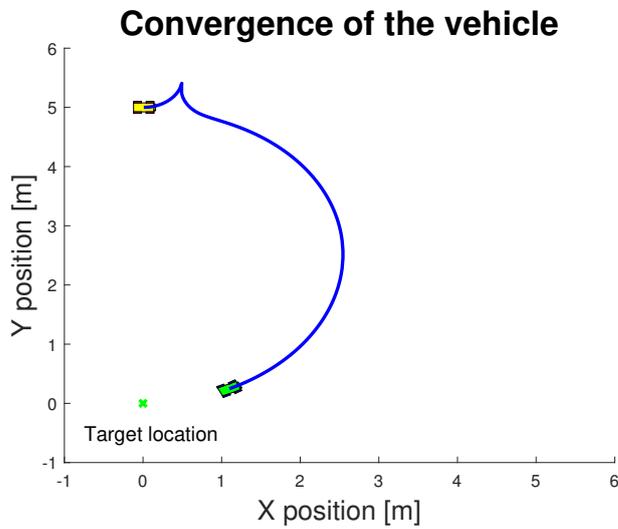
In this chapter the implementation of the algorithm for the autonomous vehicle in simulation will be described. First the vehicle will be simulated without obstacles to depict the convergence of the algorithm. In the second section different initial conditions will be used to proof that the convergence is valid for each location. Following, in the third section obstacles will be added and again convergence will be tested. The fourth chapter contains information about the tuning parameters of the algorithm. This chapter closes with a section on switching dynamics that are present in the simulation.

### 4-1 Vehicle convergence

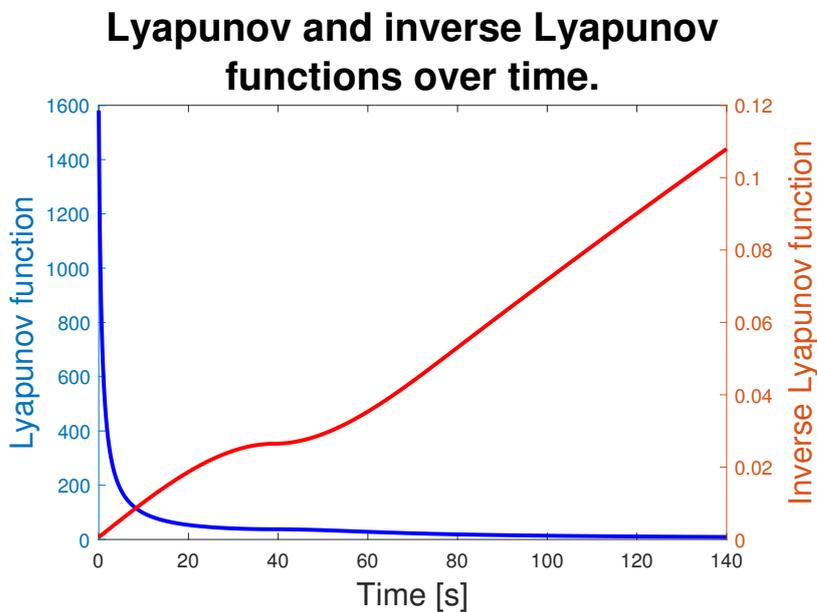
Full convergence of the vehicle will be validated if the target location is reached from any initial condition. This also means that the proposed algorithm is asymptotically stable in simulation. The Lyapunov and inverse Lyapunov functions and its derivatives can be investigated to make sure the conditions set in chapter 2 are validated throughout the simulation. This subsection will handle a single example and show its result. This convergence plot, that is used throughout the chapter, can be found in Figure 4-1. Mind you that the heading of the car is zero for a configuration to the east.

From the figure it can be observed that the vehicle starts heading forward to the right. In order to reach the correct heading in the target location, the vehicle switches from forward driving to backward driving. In reverse the vehicle will approach the target location. The target location, in this case the origin, is never fully reached. This is due to the dynamics of the discontinuous Lyapunov function described in chapter 3. The closer the autonomous vehicle will be to the origin, the lower the velocity will be. Thus reaching the target location takes more time. In order to check the Lyapunov stability the Lyapunov and inverse Lyapunov functions can be examined in Figure 4-2.

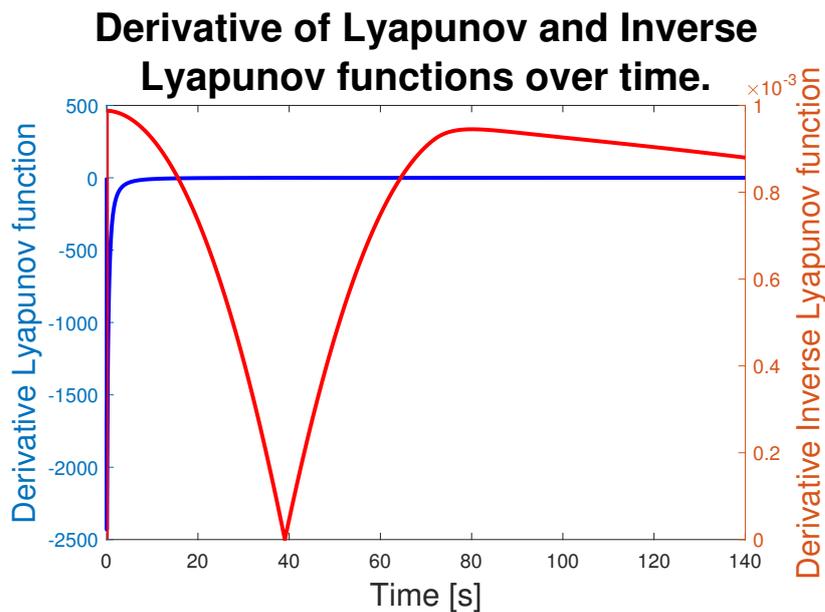
From the figure it can be derived that the Lyapunov function (blue) starts high and goes to zero as time goes to infinity. This means that the two first requirements of Lyapunov stability from chapter 2 are met. From the inverse Lyapunov function it can be concluded that it is



**Figure 4-1:** Convergence of the autonomous vehicle in simulation with starting location  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ . Blue solid line: path taken by the autonomous vehicle. Green cross: target location. Yellow car: initial location Green car: end location



**Figure 4-2:** Lyapunov and inverse Lyapunov functions of the simulation with starting location  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ . Blue solid line: Lyapunov function. Red solid line: Inverse Lyapunov function



**Figure 4-3:** Derivative of Lyapunov and inverse Lyapunov function of the simulation with starting location  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ . Blue solid line: derivative of Lyapunov function. Red solid line: derivative of inverse Lyapunov function.

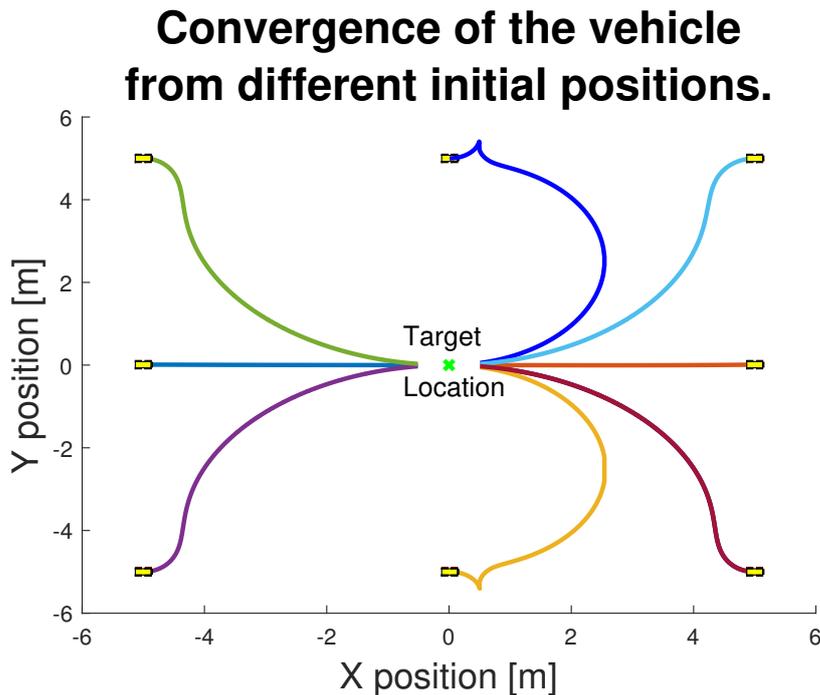
indeed positive. This means that the first requirement of the inverse Lyapunov stability is met. However the second requirement is not met. This is explained by the fact that the target location isn't reached yet and the fastest increase to infinity will happen very close to the origin. However, since the velocity will decrease closer to the origin it will take some time for this to happen. For the third requirement, the derivatives of the Lyapunov and inverse Lyapunov functions can be checked. The derivatives are depicted in Figure 4-3.

From the derivative of the Lyapunov function it can be concluded that it stays negative and goes to zero when it is closer to the target location. This means that the third requirement for Lyapunov stability is met. Of course, this should also mean that the third requirement for the inverse Lyapunov stability is met. Indeed this can be concluded from the figure. The derivative stays positive and goes to zero for the final target location. The next section will describe convergence from different initial locations.

## 4-2 Vehicle convergence from different initial locations

In order to further validate the convergence of the Lyapunov algorithm another convergence plot is made. This convergence plot contains several initial locations. This is done to show that each initial location converges to the target location. The convergence plot is depicted in Figure 4-4.

From the figure it can be concluded, that each initial location converges to the final target location. Mind you that the simulation time for this plot is higher than the simulation time of Figure 4-1. This is why the end configuration of the vehicle is closer to the target location. It also shows that the convergence towards the target location is valid. The Lyapunov functions,



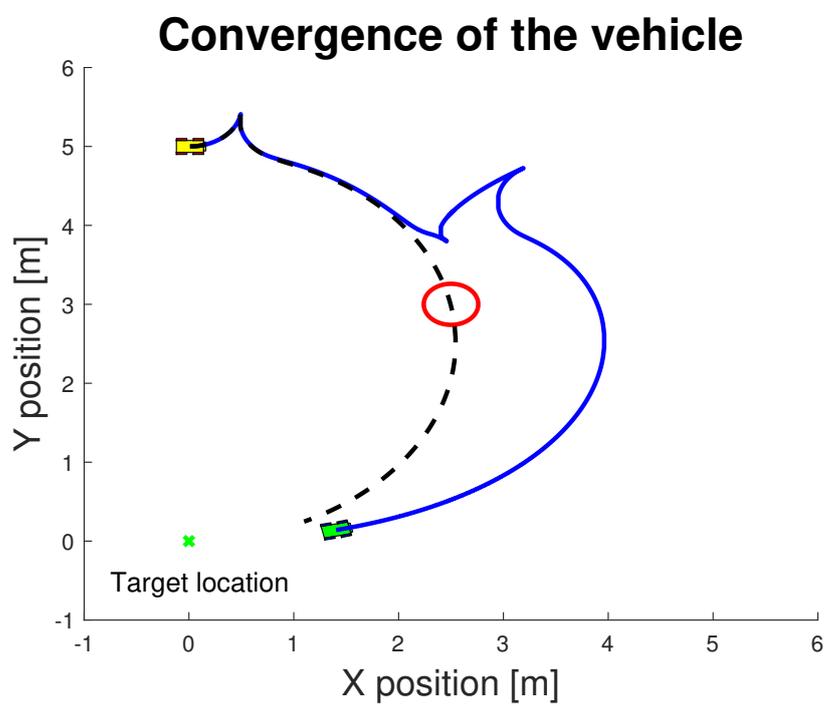
**Figure 4-4:** Different paths from several initial locations reaching the target location. Green cross: target location. Yellow cars: Initial locations. Other colours: Different paths from several initial locations to the target location.

inverse Lyapunov functions and the derivatives that prove the convergence of these paths can be found in the appendix in Figure B-1 to Figure B-8. The next section will discuss the addition of obstacles to the Lyapunov algorithm.

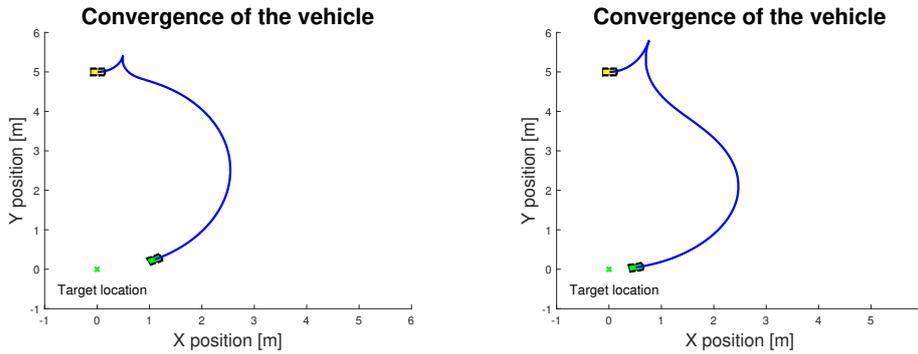
### 4-3 Vehicle convergence with obstacles

In this section an obstacle will be added. The path in the convergence plot from the first section in Figure 4-1 is used to illustrate what happens when an obstacle is added. The obstacle is added at a location where the vehicle should have passed through when there was no obstacle present. The convergence of the vehicle with obstacle is depicted in Figure 4-5. The initial path without obstacles is also depicted as a black dotted line in the figure.

From the figure it can be concluded that the vehicle steers away from the obstacle and returns to the same location as the original path with a detour. It can also be observed that the original path collided with the obstacle. From the path taken it can be concluded that the obstacle avoidance of the algorithm is working as it should. It also means that inverse Lyapunov and Lyapunov stability are met. The Lyapunov function, inverse Lyapunov function and the derivatives that prove the convergence of this path can be found in the appendix in Figure B-9. The next section will handle the tuning of the Lyapunov algorithm.



**Figure 4-5:** Convergence of the autonomous vehicle with one obstacle present. Solid blue line: path taken with obstacle avoidance. Dotted black line: path taken without obstacle avoidance. Red circle: obstacle. Green cross: target location. Yellow car: initial location. Green car: end configuration of the vehicle.



**Figure 4-6:** Convergence of the autonomous vehicle in simulation. Left: initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k_1 = 1e - 5$ . Right: initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k_1 = 1e - 4$ . Blue solid line: path taken. Green cross: target location. Yellow car: initial location. Green car: end configuration of the vehicle.

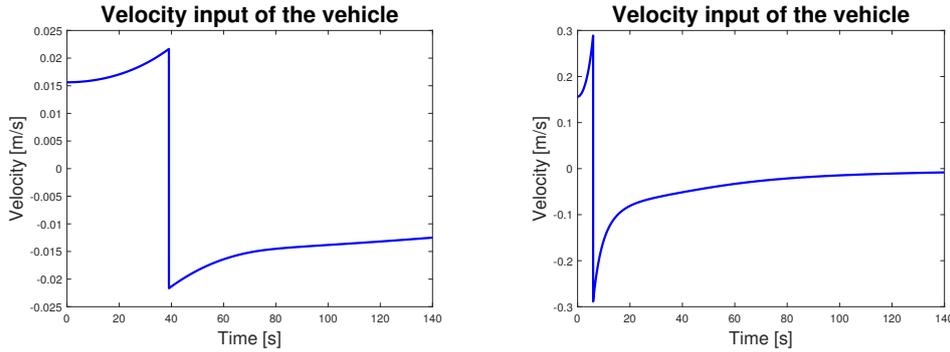
## 4-4 Tuning of the algorithm

The algorithm contains three tuning parameters that can be used to determine the desired path. The tuning parameters  $k$ ,  $k_1$  and  $k_2$  can be used to set the desired safe obstacle distance, velocity and steering respectively. Tuning for velocity and steering is quite straightforward. If a higher velocity input is required,  $k_1$  can be set higher. If a higher steering input is required,  $k_2$  can be set higher. These higher velocity or steering inputs do not alter the convergence properties. This means that the vehicle will still converge to the target location. However, the path taken will most likely be different from the original path. The model is assumed to be under low velocity and steady state cornering. For the tuning this means that  $k_1$  and  $k_2$  need to stay within bounds in order to keep the desired model dynamics. The tuning parameter for the safe obstacle distance is somewhat harder to imagine. The tuning section will be divided into three subsections, one for each tuning parameter. The first subsection, will describe the influence of  $k_1$  on the velocity and the path taken. The second subsection, will explain the influence of  $k_2$  on the steering angle and the path taken. The last subsection, will discuss the influence of  $k$  on the distance between the vehicle and the obstacle and the influence on the path.

### 4-4-1 Tuning parameter for velocity

In this subsection the influence of the tuning parameter  $k_1$  is investigated. The initial location from the first section depicted in Figure 4-1 is again used. The value of  $k_1$  for this path was set to be  $1e - 5$ .  $k_1$  will be increased by a factor ten in order to depict the influence of the velocity tuning parameter. The convergence for both situations is sketched in Figure 4-6. The velocity of the two situations is sketched in Figure 4-7.

From the first figure it can be concluded that the path taken for the second case is more steep. This can also be concluded from the second figure. It can be observed that the velocity of the vehicle is higher in the second case. A higher velocity with the same steering tuning results in a steeper path to the origin. This behaviour has no influence on the convergence as long as  $k_1$  keeps the velocity within the low speed assumption range of the model. In theory



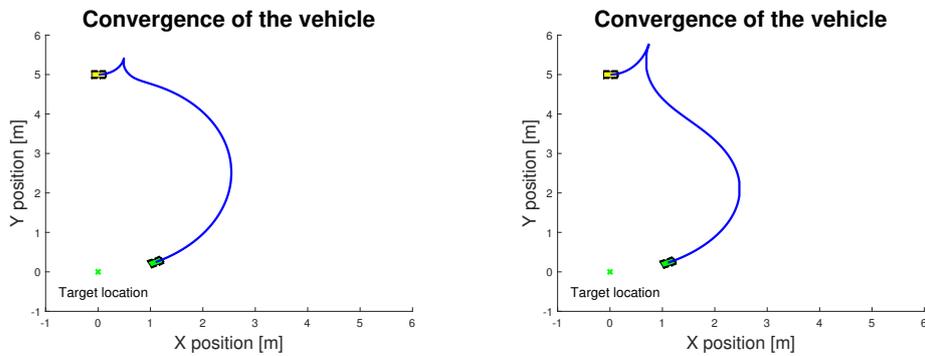
**Figure 4-7:** Velocity of the autonomous vehicle in simulation. Left: initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k_1 = 1e - 5$ . Right: initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k_1 = 1e - 4$ .

this means that the convergence to the target location will not deteriorate. However, due to the higher velocity this does mean that the vehicle will take a different path to the target location. Mind you that the velocity goes to zero the closer the vehicle gets to the target location. This is why it takes some time for the vehicle to converge to the target location. The Lyapunov function, inverse Lyapunov function and their derivatives can be found in the appendix in Figure B-10. The next subsection will discuss the influence of  $k_2$  on the steering angle and the path.

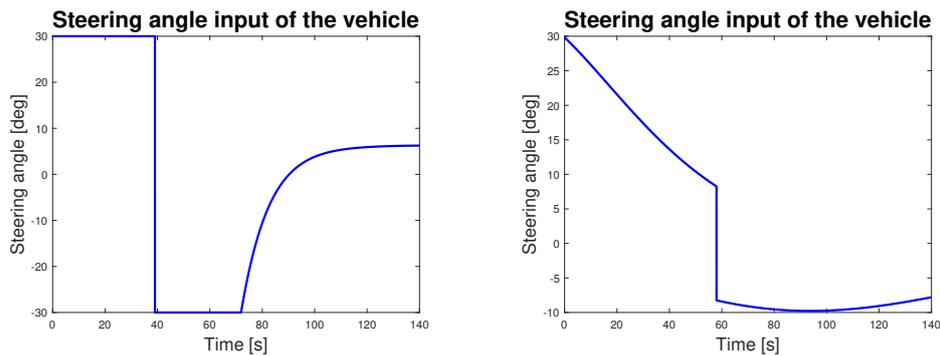
#### 4-4-2 Tuning parameter for steering

The tuning parameter  $k_2$  will be compared using the same technique of the previous subsection. The tuning parameter for steering will make sure that the vehicle can make the correct steering maneuver and reach the target location. If the parameter is too low the car does not have enough steering capacity to reach the desired configuration. If the parameter is set too high the steering reacts too fast for the actuators to follow. Mind you that in simulation without constraints the steering can be as high as desired. However, in real life situations the vehicles actuator constrains the tuning parameter. In order to simulate a limit for the steering in simulation, the maximum steering angle is set to 30 degrees. The tuning parameter in the original path of Figure 4-1 was set to be  $k_2 = 0.1$ . In the second case the tuning parameter is reduced by a factor of ten. The convergence of the two cases is depicted in Figure 4-8. The steering angle for the two cases is sketched in Figure 4-9.

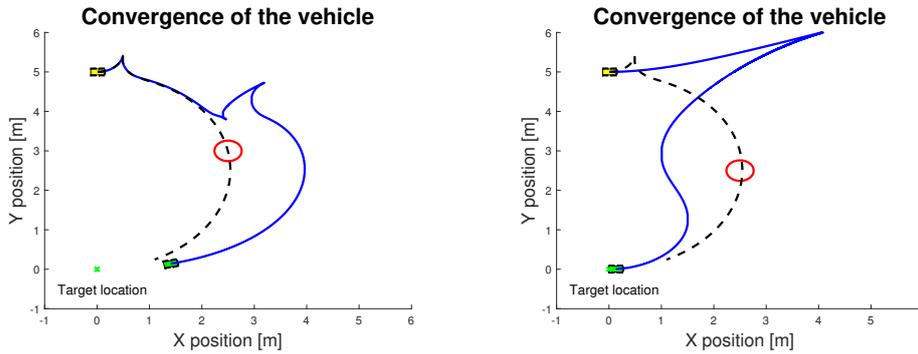
The value for  $k_2$  in the initial convergence is quite high. This causes the sharp turn at the point where the velocity changes signs for the first case in the path figure. Mind you that the steering angle for the first case in the steering angle figure is at the limit of 30 degrees. The steering angle in the second case doesn't fully reach the maximum steering angle of 30 degrees. This can also be observed in the path for the second case in the path figure. The path is steeper meaning that the car uses less steering capacity. In conclusion the convergence doesn't change, the car still goes to the target location. However, again the path taken is different from the original path. In conclusion to the tuning of  $k_1$  and  $k_2$ , the parameters can be used to set the input to the model and determine the path to the target location. However, the tuning parameters must be used with care. If set too high or too low they may cause unwanted behaviour. The figures depicting the stability of this example can be found



**Figure 4-8:** Convergence of the autonomous vehicle in simulation. Left: initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k_2 = 0.1$ . Right: initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k_2 = 0.01$ . Blue solid line: path taken. Green cross: target location. Yellow car: initial location. Green car: end configuration of the vehicle.



**Figure 4-9:** Steering angle of the autonomous vehicle in simulation. Left: initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k_2 = 0.1$ . Right: initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k_2 = 0.01$ .



**Figure 4-10:** Convergence of the autonomous vehicle in simulation with obstacle. Left: initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k = 10$ . Right: initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k = 1$ . Black dotted line: original path, without obstacle avoidance. Solid blue line: path with obstacle avoidance. Green cross: target location. Yellow car: initial location. Green car: end configuration of the vehicle

in the appendix in Figure B-11. The next section will explain the influence of  $k$  on the safe obstacle distance and the path.

### 4-4-3 Tuning parameter for obstacle avoidance

In the two previous subsections the two tuning parameters for the input of the model have been discussed. In this subsection an obstacle will be added and the tuning parameter  $k$  that influences the safe distance to an obstacle will be discussed. The same obstacle will be used as in the third section of this chapter. The tuning parameter of this case was set to be  $k = 10$ . This case is compared with a second case where  $k$  is set to be 1. The convergence of the two cases is depicted in Figure 4-10.

In the first case of the figure the value for  $k$  is higher than in the second case. Therefore, the safety distance in the first case will be lower. This means that the influence of the obstacle will only start acting when the vehicle is close to the obstacle. This creates a situation where the vehicle will react late to the influence of the obstacle. This explains the behaviour that when the vehicle gets close to the obstacle, it turns around and takes another path. In the second case  $k$  is set lower. Therefore, the safety distance will be higher and the vehicle will react earlier on the presence of an obstacle. From the first figure it can be observed that the path taken in the second case is completely different from the path in the first case. The influence of the obstacle is higher in the second case. The result is that the vehicle takes another path in order to avoid the obstacle. Depending on the safety situation  $k$  can be tuned to avoid obstacles in the way desired. Lastly, from the figure it can be concluded that the velocity is higher in the second case. The end configuration lies closer to the target location. This means that  $k$  and  $k_1$  are not independent. The stability of this case can be tested by looking at Figure B-12 in the appendix.

Since  $k$  appears in the inverse Lyapunov function it means that the value of  $k$  will also alter the convergence properties of the algorithm. If  $k$  is decreased,  $\frac{1}{k}$  is increased. This in turn makes that  $B^{\frac{1}{k}}$  increases. Due to  $B^{\frac{1}{k}}$  increasing, the inverse Lyapunov function will increase. This results in a smaller Lyapunov function. Since the vehicle is controlled to the lowest value

of the Lyapunov function it means that decreasing the Lyapunov function with  $k$  can create a problem for the convergence. In order to keep the desired convergence of the algorithm  $k$  needs a lower limit. Mind you that the smaller  $k$  is the smaller also the Lyapunov function will be. This means that if  $k$  is made smaller the vehicle will react to the obstacle earlier. But it also means that the vehicle is allowed to come closer to the obstacle. Eventually, if  $k$  is made too small the convergence will be depreciated. This means that the vehicle will not converge to the target location but it will converge to the obstacle and collide with it. This provides both advantages as disadvantages. Picture for instance a case with one obstacle. If  $k$  is made large enough the autonomous vehicle will get close to an obstacle but will never collide with it. However, if the vehicle maneuvers a narrow corridor and  $k$  is made too large the autonomous vehicle will be avoiding both walls. This can create a non feasible path through the corridor. This situation requires  $k$  to be smaller while keeping in mind the lower limit of  $k$ . There is a trade-off between obstacle avoidance and convergence and care should be taken in tuning  $k$ .

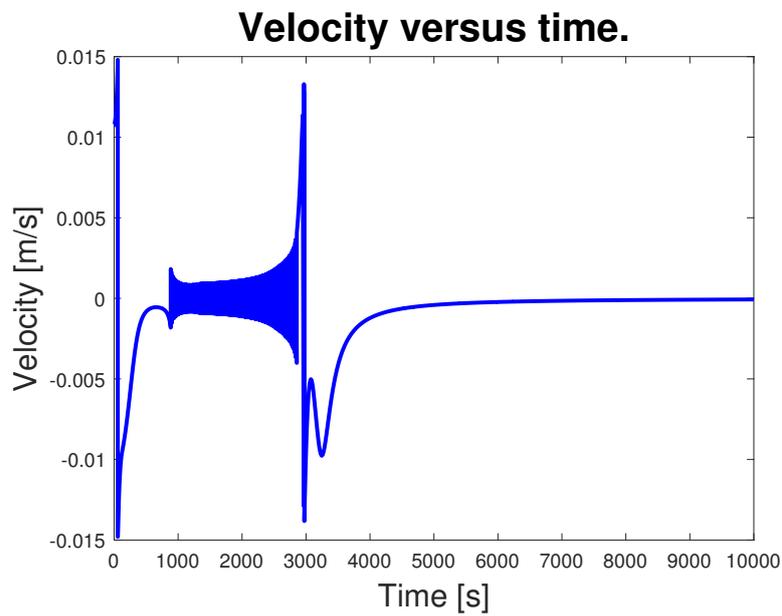
Finalizing this subsection, the tuning parameter  $k$  can be used to set the desired path for obstacle avoidance. There is a trade-off between obstacle avoidance and convergence and care should be taken with tuning  $k$ . The next section will handle switching dynamics that occur in the algorithm due to the influence of the discontinuous Lyapunov function.

## 4-5 Switching dynamics

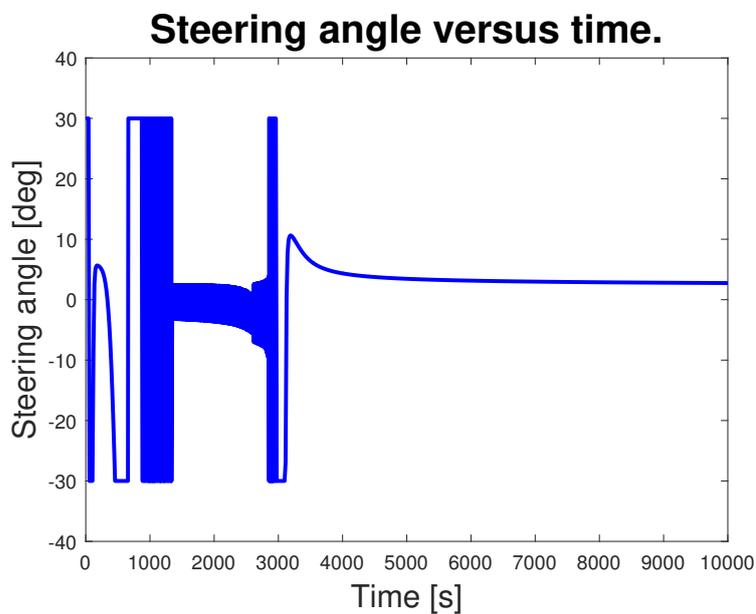
The previous section described the influence of the tuning parameters on the dynamics and the convergence. The last section will focus on the influence of the discontinuous Lyapunov function. Switching behaviour is happening whenever the steering angle and velocity are both almost zero. This behaviour is for instance happening in the path depicted in Figure 4-5 close to the obstacle. In order to sketch this behaviour the velocity and steering of this path are sketched in Figure 4-11 and Figure 4-12.

From the figure it can be concluded that the switching behaviour (the region where the plot is almost solid) influences the steering and the velocity. The behaviour is happening whenever the velocity and the steering angle are almost zero. This is undesired behaviour since the control input can't be followed by the actuators. Either the actuators stop and the car isn't moving or the actuator switches from one state to the other. This false trigger of the discontinuous Lyapunov function makes the algorithm believe that the vehicle is at its target location while in fact it is not. The behaviour can be prevented by taking a lower value of  $k$ . The upper limit of  $k$  is beyond the topic of this thesis. However, the behaviour needs to be expressed while discussing the algorithm.

In conclusion this chapter discussed the simulation of the autonomous vehicle and the algorithm. The tuning parameters of the algorithm can be altered to get the desired dynamics. However, the tuning parameters must be used with caution especially at lower values for  $k_1$  and  $k_2$  and higher values of  $k$ . The next chapter will conclude the thesis and give recommendations for future research.



**Figure 4-11:** Velocity of the autonomous vehicle in simulation with initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k = 10$  and switching behaviour.



**Figure 4-12:** Steering angle of the autonomous vehicle in simulation with initial conditions,  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k = 10$  and switching behaviour.



# Conclusion and Recommendations

This chapter will conclude the thesis and recapitulate on the research objectives defined in the introduction. Furthermore, recommendations for future work will be given and discussed. The first section, handles the conclusion. The second section, elaborates on the recommendations.

## 5-1 Conclusion

Current algorithms for autonomous vehicles are divided into separate local algorithms. Algorithms like path planning and motion control are used to control the autonomous vehicle. However, these controllers are not integrated and therefore path following is not achieved. The main objective of this research is to develop an integrated Lyapunov stable algorithm that includes the two topics of navigation: path planning and motion control. The conclusion will be discussed according to the different sections of the research objectives.

### 5-1-1 Develop an integrated Lyapunov stable algorithm

As basis for the integrated Lyapunov stable algorithm an existing controller is used that drives holonomic robots to their target location. The algorithm is extended with non-holonomic dynamics of a car-like vehicle. A measure for the distance between the vehicle and the obstacles is used to determine how close the vehicle is to a certain object. The main contribution of this thesis is an integrated Lyapunov stable algorithm which can be used to drive autonomous vehicles.

#### **Convergence of the vehicle to the target location from any initial point**

The proposed algorithm can be used to drive the car to the target location from any initial location. The discontinuous properties of the algorithm make sure that Lyapunov stability is guaranteed for all vehicle locations.

## Obstacle avoidance

The algorithm includes obstacle avoidance. Obstacles can be added to the algorithm and are then avoided by the vehicle. The derivative of the obstacle distance function is used to determine the type of action needed for collision avoidance. Furthermore, the path taken to avoid obstacles can be tuned by the parameters of the algorithm.

## Tuning for different velocity and steering

Tuning for different velocity and steering is done by the parameters  $k_1$  and  $k_2$ . The tuning is simple and can be used to make the desired inputs to the autonomous vehicle. These parameters depend to the vehicle parameters and the limitation of the actuators.

### 5-1-2 Proof the stability of the algorithm

The stability of the proposed algorithm is proved using Lyapunov theory. The Lyapunov theory is also used to derive the control inputs to the vehicle, keeping the algorithm stable at all times.

### 5-1-3 Evaluate the proposed approach by simulation

The algorithm is evaluated in a simulation environment. The convergence of the algorithm is validated and also the obstacle avoidance is tested in different scenarios. Besides the validation of the algorithm, the simulation also brings forth a problem where switching dynamics occur. The discontinuous Lyapunov function falsely triggers whenever  $u$  and  $\delta$  lie close to zero. This behaviour can be avoided by using a smaller gain in the algorithm.

## 5-2 Recommendations

The algorithm proposed in this thesis is a simple and effective way of navigating autonomous vehicles. However, in order to apply the algorithm in practice more research is required. The recommendations are listed in different subsections in order to give structure to future research.

### 5-2-1 Model

The kinematic bicycle model is a simple but effective model to use. This means that it works for the assumptions given. In practice these assumptions are not always present. A more extensive model can be used to provide a smaller difference between the control for experimental implementation and simulation. The addition of a model that can handle higher velocities is one that is vital in autonomous vehicle research. Furthermore, a model that includes slip and lateral forces is an extension that will provide more accurate results.

### 5-2-2 Switching behaviour

The switching behaviour occurring in the algorithm is undesired. This behaviour makes control harder and actuators might not be able to handle these dynamics. In order to avoid these dynamics the limits of the tuning parameter  $k$  can be investigated. Furthermore, switching logic can be added to avoid false triggering of the discontinuous Lyapunov function.

### 5-2-3 Tuning limits and optimization

Tuning of the algorithm determines the path the vehicle takes towards the target location. Due to these parameters the optimal path to the target location is not always chosen. Furthermore, the limits of the tuning parameters interfere with desired behaviour of the vehicle. Take as an example the velocity that goes to zero the closer the vehicle is to the target location. The tuning parameter can be altered to neglect this behaviour and thus optimize the motion control of the autonomous vehicle. Furthermore, the limits of the tuning parameters  $k_1$  and  $k_2$  can be researched. This helps in gaining more insight in the efficiency of the algorithm.

### 5-2-4 Obstacle avoidance

The scalability of the obstacle avoidance is another topic that can be researched. Due to the nature of the obstacle distance measure, the derivative of the measure will grow with increasing obstacles. This means that  $\beta$  and the partial derivatives of  $\beta$  need to be determined for each appended obstacle. By selecting the appropriate measure for the obstacle distance,  $\beta$  and the partial derivatives of  $\beta$  can be reduced. This will not only improve the scalability of the algorithm, but will also reduce the computational time.

### 5-2-5 Experimental testing

Lastly, experimental testing can be considered as a next step to evaluate the algorithm in real time applications. Experimental testing introduces new challenges that simulation does not include. Therefore, testing the algorithm on a scaled car or on a real car is vital before the algorithm can be implemented on autonomous vehicles.



---

# Appendix A

---

## Control Strategy

### A-1 The control algorithm

#### A-1-1 Derivative of $\beta$ calculation

The partial derivative of  $\beta$  to  $x$  can be written as in Equation A-1

$$\frac{\partial \beta}{\partial x} = \frac{a_x}{\sqrt{\frac{b_x}{c_x} d_x}} \quad (\text{A-1})$$

Then the different terms in the equation can be expressed as in Equation A-2 to Equation A-5.

$$\begin{aligned} a_x = & 2ar^2br^4 \left( (2bo^4xo^2 - 2xo(y_r - yo)^2(xo + yo)bo^2 + (xo^2 + yo^2)(y_r - yo)^4)ao^4 \right. \\ & + 2(-xo(xo + yo)bo^2 + (xo^2 + yo^2)(y_r - yo)^2)bo^2(x_r - xo)^2ao^2 \\ & \left. + bo^4(xo^2 + yo^2)(x_r - xo)^4 \right) (x_r - xo) \left( (y_o - br - yr)(br - yr + y_o)ar^2 + br^2(xr - xo)^2 \right) \end{aligned} \quad (\text{A-2})$$

$$\begin{aligned} b_x = & \left( (2bo^4xo^2 - 2xo(y_r - yo)^2(xo + yo)bo^2 + (xo^2 + yo^2)(y_r - yo)^4)ao^4 \right. \\ & + 2(-xo(xo - yo)bo^2 + (xo^2 + yo^2)(y_r - yo)^2)bo^2(x_r - xo)^2ao^2 \\ & \left. + bo^4(xo^2 + yo^2)(x_r - xo)^4 \right) \left( (y_o - br - yr)(br - yr + y_o)ar^2 + br^2(xr - xo)^2 \right)^2 \end{aligned} \quad (\text{A-3})$$

$$c_x = \left( (x_r - xo)^2bo^2 + (y_r - yo)^2ao^2 \right)^2 \left( (y_o - yr)^2ar^2 + br^2(xr - xo)^2 \right)^2 \quad (\text{A-4})$$

$$d_x = \left( (x_r - x_o)^2 b o^2 + (y_r - y_o)^2 a o^2 \right)^2 \left( (y_o - y_r)^2 a r^2 + b r^2 (x_r - x_o)^2 \right)^3 \quad (\text{A-5})$$

The same thing can be done for the partial derivative of  $\beta$  to  $y$ . The simplification is written in Equation A-6.

$$\frac{\partial \beta}{\partial y} = \frac{a_y}{\sqrt{\frac{b_y}{c_y} d_y}} \quad (\text{A-6})$$

The different terms are expressed in Equation A-7 to Equation A-10.

$$\begin{aligned} a_x = & -2ar^4br^2 \left( (2bo^4xo^2 - 2xo(y_r - yo)^2(xo + yo)bo.^2 + (xo^2 + yo^2)(y_r - yo)^4)ao^4 \right. \\ & + 2(-xo(xo + yo)bo^2 + (xo^2 + yo^2)(y_r - yo)^2)bo^2(x_r - xo)^2ao^2 \\ & \left. + bo.^4(xo^2 + yo^2)(x_r - xo)^4 \right) (y_o - y_r) \left( (y_o - br - yr)(br - yr + y_o)ar^2 + br^2(xr - x_o)^2 \right) \end{aligned} \quad (\text{A-7})$$

$$\begin{aligned} b_x = & \left( (2bo^4xo^2 - 2xo(y_r - yo)^2(xo + yo)bo^2 + (xo^2 + yo^2)(y_r - yo)^4)ao^4 \right. \\ & + 2(-xo(xo + yo)bo^2 + (xo^2 + yo^2)(y_r - yo)^2)bo^2(x_r - xo)^2ao^2 \\ & \left. + bo.^4(xo^2 + yo^2)(x_r - xo)^4 \right) \left( (y_o - br - yr)(br - yr + y_o)ar^2 + br^2(xr - x_o)^2 \right)^2 \end{aligned} \quad (\text{A-8})$$

$$c_y = ((x_r - x_o)^2 b o^2 + (y_r - y_o)^2 a o^2)^2 ((y_o - y_r)^2 a r^2 + b r^2 (x_r - x_o)^2)^2 \quad (\text{A-9})$$

$$d_y = \left( (x_r - x_o)^2 b o^2 + (y_r - y_o)^2 a o^2 \right)^2 \left( (y_o - y_r)^2 a r^2 + b r^2 (x_r - x_o)^2 \right)^3 \quad (\text{A-10})$$

The different terms in Equation A-2, Equation A-3, Equation A-4 and Equation A-5 and the terms in Equation A-7, Equation A-8, Equation A-9 and Equation A-10 are listed in Table A-1.

<b>Parameter</b>	<b>Definition</b>	<b>Unit</b>
$ar$	Ellipse x-axis definition of the vehicle	[-]
$br$	Ellipse y-axis definition of the vehicle	[-]
$ao$	Ellipse x-axis definition of the object	[-]
$bo$	Ellipse y-axis definition of the object	[-]
$xr$	X-axis location of the vehicle	[ $m$ ]
$yr$	Y-axis location of the vehicle	[ $m$ ]
$x_r$	Boundary x-axis coordinates of the vehicle	[ $m$ ]
$y_r$	Boundary y-axis coordinates of the vehicle	[ $m$ ]
$xo$	X-axis location of the object	[ $m$ ]
$yo$	Y-axis location of the object	[ $m$ ]
$x_o$	Boundary x-axis coordinates of the object	[ $m$ ]
$y_o$	Boundary y-axis coordinates of the object	[ $m$ ]

**Table A-1:** Parameters appearing in the partial derivatives of  $\beta$ .



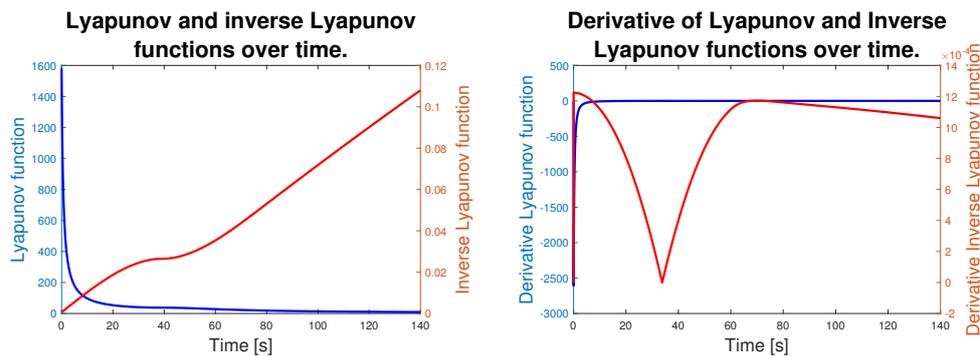
---

# Appendix B

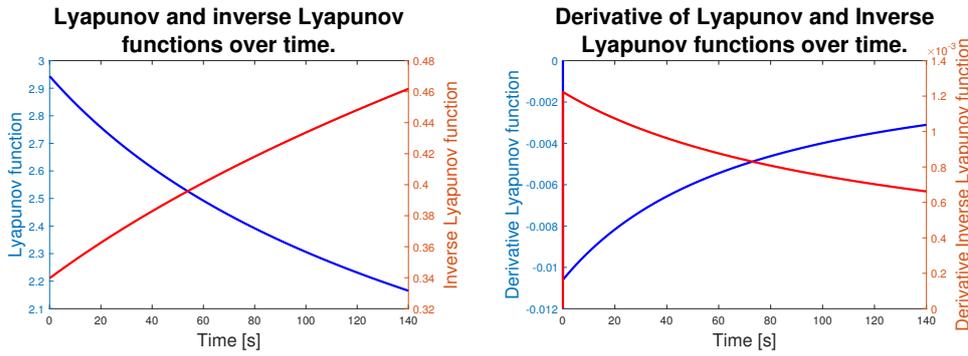
---

## Simulation

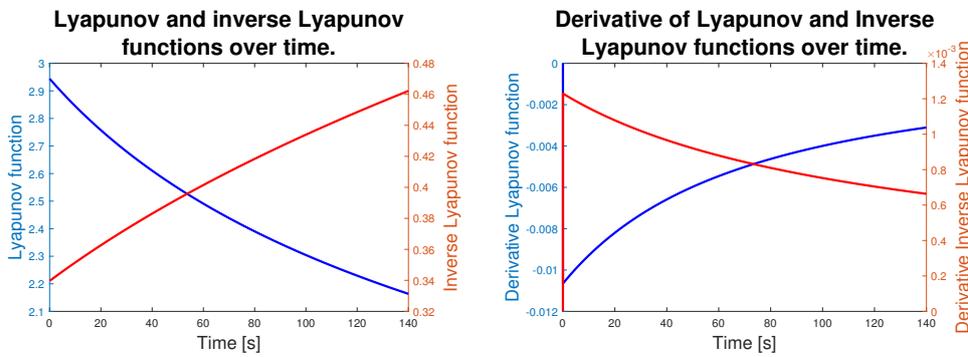
### B-1 Vehicle convergence from different initial positions



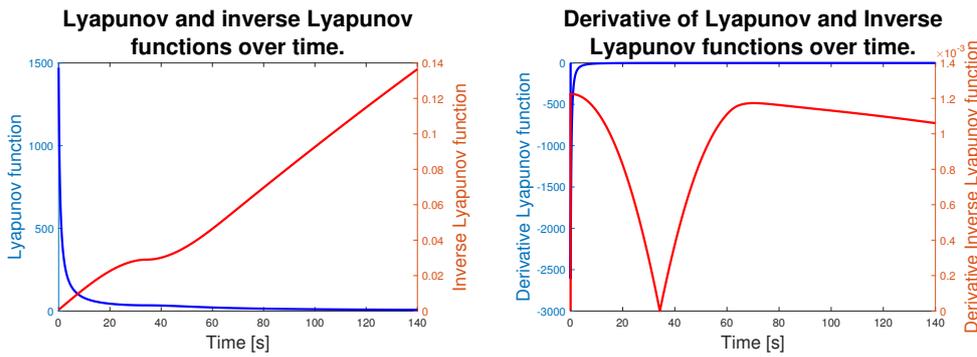
**Figure B-1:** Stability proof of the multiple initial conditions plot. Initial condition:  $x = 0$ ,  $y = 5$ ,  $\theta = 0$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions



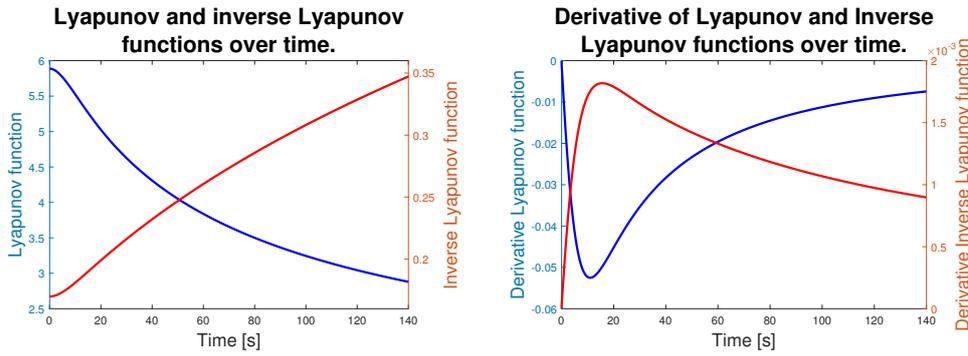
**Figure B-2:** Stability proof of the multiple initial conditions plot. Initial condition:  $x = -5$ ,  $y = 0$ ,  $\theta = 0$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions



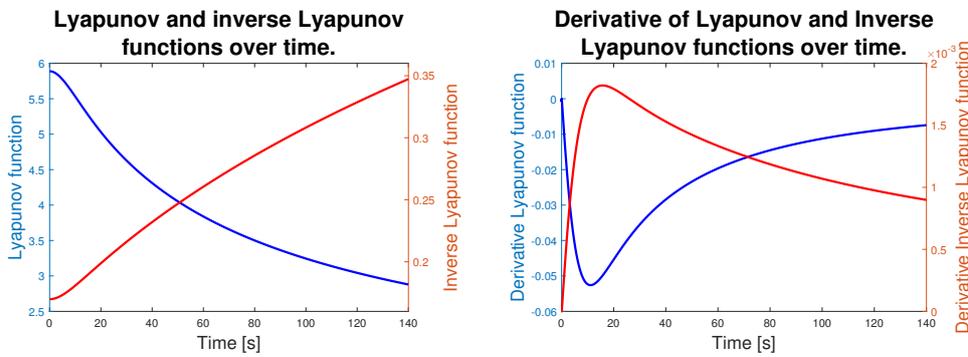
**Figure B-3:** Stability proof of the multiple initial conditions plot. Initial condition:  $x = 5$ ,  $y = 0$ ,  $\theta = 0$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions



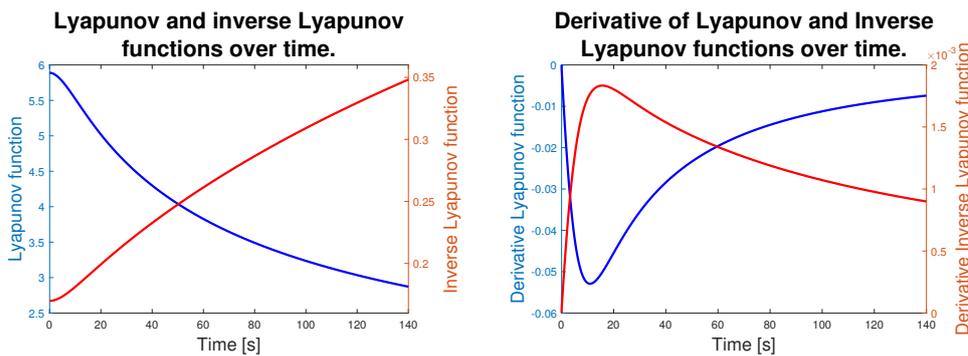
**Figure B-4:** Stability proof of the multiple initial conditions plot. Initial condition:  $x = 0$ ,  $y = -5$ ,  $\theta = 0$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions



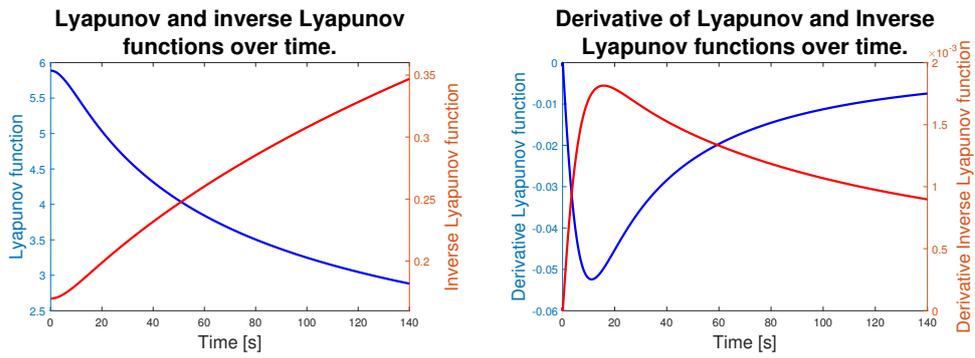
**Figure B-5:** Stability proof of the multiple initial conditions plot. Initial condition:  $x = -5$ ,  $y = -5$ ,  $\theta = 0$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions



**Figure B-6:** Stability proof of the multiple initial conditions plot. Initial condition:  $x = -5$ ,  $y = 5$ ,  $\theta = 0$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions

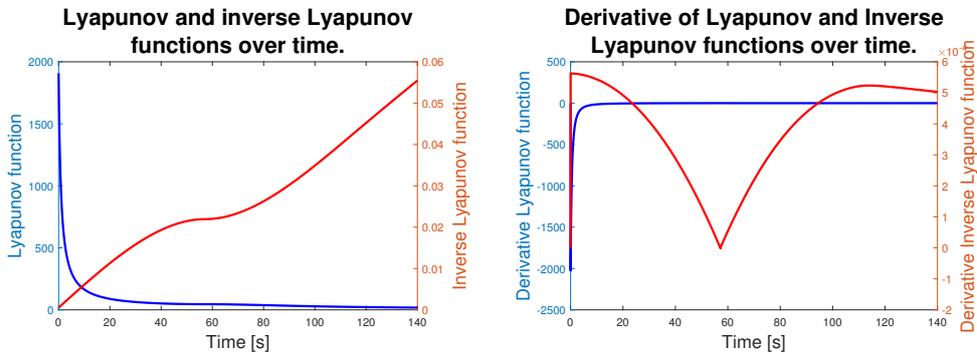


**Figure B-7:** Stability proof of the multiple initial conditions plot. Initial condition:  $x = 5$ ,  $y = 5$ ,  $\theta = 0$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions



**Figure B-8:** Stability proof of the multiple initial conditions plot. Initial condition:  $x = 5$ ,  $y = -5$ ,  $\theta = 0$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions

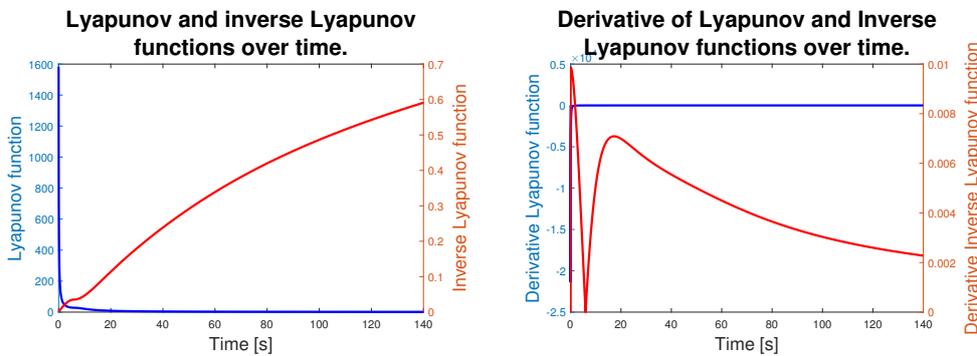
## B-2 Vehicle convergence with obstacles



**Figure B-9:** Stability proof of the example with obstacle avoidance. Initial conditions:  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k = 10$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions

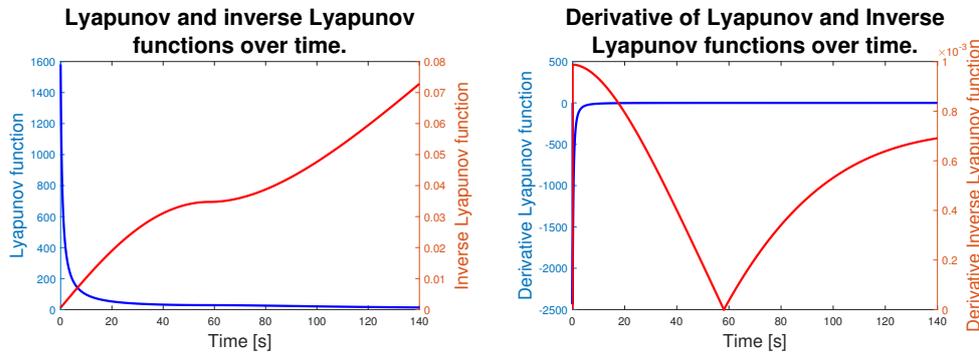
## B-3 Tuning of the controller

### B-3-1 Tuning parameter for velocity



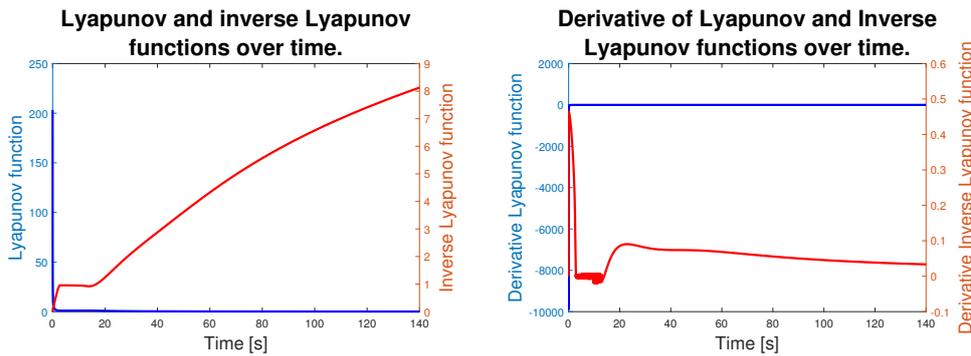
**Figure B-10:** Stability proof of the example with increased velocity tuning. Initial conditions:  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k_1 = 1e - 4$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions

### B-3-2 Tuning parameter for steering



**Figure B-11:** Stability proof of the example with decreased steering angle tuning. Initial conditions:  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k_2 = 0.01$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions

### B-3-3 Tuning parameter for obstacle avoidance



**Figure B-12:** Stability proof of the example with increased obstacle avoidance. Initial conditions:  $x = 0$ ,  $y = 5$ ,  $\theta = 0$ ,  $k = 1$  Left: Lyapunov and inverse Lyapunov functions Right: derivative of Lyapunov and inverse Lyapunov functions

---

# Bibliography

- [1] WHO, *Global status report on road safety*. World Health Organization, 2018.
- [2] NHTSA, *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey*. National Motor Vehicle Crash Causation Survey, 2018.
- [3] Sungon Lee, Minchul Kim, Youngil Youm, and Wankyun Chung, “Control of a car-like mobile robot for parking problem,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, pp. 1–6 vol.1, May 1999.
- [4] M. Q. Dao and K. Liu, “Development of a practical automatic parking technology for automobiles,” in *2006 Chinese Control Conference*, pp. 2159–2164, Aug 2006.
- [5] J. Desai and V. Kumar, “Nonholonomic motion planning for multiple mobile manipulators,” in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 3409–3414, 1997.
- [6] S. Lee, Y. Youm, and W. Chung, “Control of car-like mobile robots for posture stabilization,” in *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*, vol. 3, pp. 1745–1750 vol.3, 1999.
- [7] D. Castro, U. Nunes, and A. Ruano, “Obstacle avoidance in local navigation,” in *IEEE Mediterranean Conference on Control and Automation MED 2002*, 2002.
- [8] K. Sekiguchi, M. Deng, and A. Inoue, “Obstacle avoidance and two wheeled mobile robot control using potential function,” in *2006 IEEE International Conference on Industrial Technology*, pp. 2314–2319, IEEE, 2006.
- [9] T.-C. Liang, J.-S. Liu, G.-T. Hung, and Y.-Z. Chang, “Practical and flexible path planning for car-like mobile robot using maximal-curvature cubic spiral,” *Robotics and Autonomous Systems*, vol. 52, no. 4, pp. 312 – 335, 2005.
- [10] A. Scheuer and T. Fraichard, “Continuous-curvature path planning for car-like vehicles,” in *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and*

- Systems. Innovative Robotics for Real-World Applications. IROS '97*, vol. 2, pp. 997–1003 vol.2, Sep. 1997.
- [11] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, “Closed loop steering of unicycle like vehicles via lyapunov techniques,” *IEEE Robot. Automat. Mag.*, vol. 2, pp. 27–35, 1995.
- [12] J. J. Park and B. Kuipers, “A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 4896–4902, IEEE, 2011.
- [13] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, “Closed loop steering of unicycle like vehicles via lyapunov techniques,” *IEEE Robotics Automation Magazine*, vol. 2, pp. 27–35, Mar 1995.
- [14] H. G. Tanner, S. Loizou, and K. J. Kyriakopoulos, “Nonholonomic stabilization with collision avoidance for mobile robots,” in *In Proceedings of the 2001 International Conference on Intelligent Robots and Systems*, pp. 1220–1225, 2001.
- [15] H. G. Tanner and K. J. Kyriakopoulos, “Nonholonomic motion planning for mobile manipulators,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2, pp. 1233–1238, IEEE, 2000.
- [16] E. Alcalá, V. Puig, J. Quevedo, T. Escobet, and R. Comasolivas, “Autonomous vehicle control using a kinematic lyapunov-based technique with lqr-lmi tuning,” *Control Engineering Practice*, vol. 73, pp. 1–12, 04 2018.
- [17] D. Dobriborsci, A. Kapitonov, and N. Nikolaev, “The basics of the identification, localization and navigation for mobile robots,” in *2017 International Conference on Information and Digital Technologies (IDT)*, pp. 100–105, July 2017.
- [18] H. Secchi, R. Carelli, and V. Mut, “Design of stable algorithms for mobile robot control with obstacle avoidance,” in *14th IFAC World Congress on Automatic Control*, pp. 185–190, 1999.
- [19] M. Q. Dao and K.-Z. Liu, “Development of a practical automatic parking technology for automobiles,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 1727–1732, IEEE, 2006.
- [20] E. Rimon and D. Koditschek, “Exact robot navigation using artificial potential fields,” *Robotics and Automation, IEEE Transactions on*, vol. 8, pp. 501 – 518, 11 1992.