NTNU
Norwegian University of
Science and Technology

Guidance Algorithms for Planar Path-based Motion Control Scenarios

Joakim Haugen

Master of Science in Engineering Cybernetics
Submission date: June 2010
Supervisor: Thor Inge Fossen, ITK
Co-supervisor: Morten Breivik, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

The candidate will consider problems associated with feasibly controlling a maneuverability-constrained vehicle for motion control scenarios such as planar path maneuvering and path tracking. The following elements must thus be considered:

1. Develop guidance laws for:
   a) Feasible convergence of a maneuverability-constrained vehicle to a planar path (steering and speed problems).
   b) Feasible movement of a maneuverability-constrained vehicle along a planar path (speed problem).
2. Investigate the stability properties of the suggested guidance laws.
3. Suggest how a physical path (like a road) can be converted into an analytic form which is suitable for use with the proposed guidance laws.
4. Develop guidance algorithms which solve the path-tracking problem when the path must be generated online based on the movements of a leader vehicle.
5. Investigate the problem of creating feasible paths associated with lawn-mower survey operations.
6. Explore the behavior of the suggested guidance algorithms through numerical simulations.

Assignment given: 11. January 2010
Supervisor: Thor Inge Fossen, ITK

Summary

The problem of performing accurate path maneuvering tasks in planar space is investigated in thesis. The purpose is to utilize limited knowledge about the vehicle's maneuverability constraints to output feasible reference signals.

Acceleration limitations of the vehicle have been used in an algorithm that determines forward speeds in such way that a predefined path can be followed at high speeds. The algorithm ensures that the speed is reduced before acute turns. Furthermore, an existing steering law has been modified to dynamically take the limitations of the vehicle into consideration when determining the desired course. This modified steering law exhibits desirable convergence characteristics toward the desired path.

A complete guidance system, which combines the path convergence algorithm with the path speed algorithm, has been proposed. This system is able to rapidly converge to the desired path and follow this path, even for paths where the curvature is large.

The modified steering law has been combined with a path-tracking speed controller. The path-tracking speed controller makes sure the vehicle can track a target on a predefined path. The resulting path-tracking system is able to follow a leader vehicle's path by creating accurate paths online from periodically sampled positions.

A method for creating feasible U-turns in a lawn-mower pattern has been proposed. For a given vehicle speed, the resulting path obeys angular speed and angular acceleration constraints.

Finally, the proposed algorithms are tested in simulations to illustrate their behavior and usefulness.

Preface

The results presented in this thesis are the product of my work during the last year as a student for a Master's degree in Engineering Cybernetics. Although some of this work is not up the alley of control theory, it reflects the diversity of a Master's degree in Engineering Cybernetics. The last five years have been both challenging and interesting, and it is with mixed feeling I perform the last finishing touches on my thesis.

I would like to thank my supervisor Morten Breivik for several profitable discussions throughout the semester. Furthermore, I would like to thank my fellow students at the office, Jon Erik Loberg, Øivind Kjerstad, Per Nord and in particular Vetle Vintervold, which has been my classmate and friend for over a decade.

In the last six months, lots of work have been put in to this thesis. Many of the working hours have been executed at untraditional times of the day, some of them in frustration, but all in all, the work has been rewarding. Even though my road ahead is only partially determined, the direction at the next intersection is already chosen, hopefully being the right one.

Joakim Haugen

Trondheim, June 21, 2010

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

With the exception of station keeping, the purpose of a vehicle is to move around. To be able to do this, appropriate reference signals must be produced by a guidance system. For several hundred years vehicles have been controlled by steersmen, whose task were to determine forward speed and course. With the increased need for automatic control of unmanned vehicles (Breivik 2010), several propositions for path following algorithms have emerged.

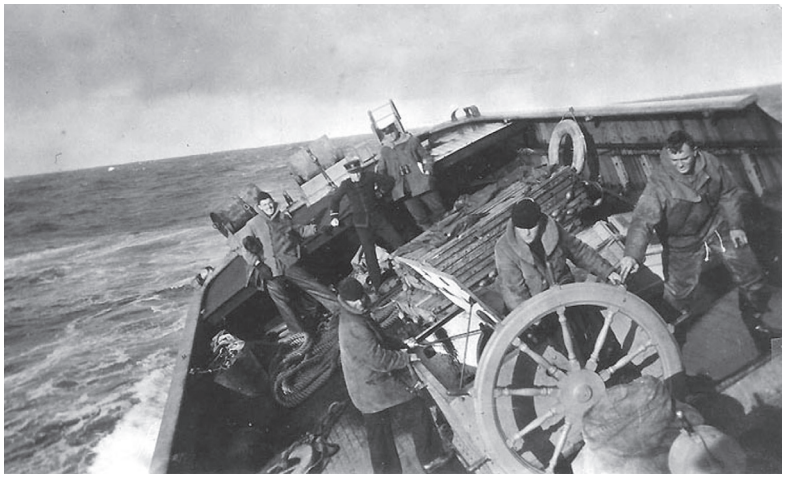


Figure 1.1: A steersman at work aboard USS *Noma* during World War I. Courtesy of Captain Lamar R. Leahy, www.history.navy.mil.

The oil and gas sector is very important to the Norwegian economy. Hence, a crucial part of succeeding in this competitive sector is having cutting edge technology which is cost effective and efficient. According to (Breivik et al. 2008), it is expected that unmanned vehicle technology will play a significant role for hydrocarbon exploration and exploitation in the future. Since these future vehicles do not need to accommodate people, they can employ new kinds of vehicle design, that is, high power-to-weight ratio and different hull designs (Cooper et al. 2002). Unfortunately, the traditional autopilots are designed for slow maneuvers like course changing, and hence cannot take advantage of rapid maneuvers at high speed, which might characterize an unmanned vehicle.

Figure 1.2: There exist a *Viknes 830* which has been retrofitted by Maritime Robotics. This modified vehicle operates as a USV with accommodation for people, such that it is convenient to do experimental trials. Courtesy of Viknes, www.viknes.no.

Unmanned surface vehicles (USVs) have existed for a long time, but in the last couple of decades, a great deal of research effort has given attention to the control of USVs in order to develop more agile and maneuverable vehicles (Bertram 2008). For instance, a USV must often push the limits of the vehicle in order to satisfy a given task. To push the envelope, the maneuverability constraints of the USV must be taken into consideration when the reference signals are determined. Furthermore, USVs often work in speed regimes were underactuation is inevitable, that is, there are fewer actuators than degrees of freedom, and hence control laws developed for fully actuated systems cannot be employed.

Some Guidance, Navigation and Control (GNC) systems are designed using the principle of modularity. These systems distinguish between the guidance system and the control system, that is, they are not that interwoven, meaning that an underlying controller is able to converge to the reference signal produced by a guidance system (Skjetne 2005). Thus, an upgrade of the guid-

ance system may yield powerful upgrades with serious value for the money. This include potential benefits such as optimization of path traversing and the ability to perform diverse tasks in different environments.

1.2 Contributions

The contributions of this thesis are miscellaneous, but can mainly be summarized in four categories as follows:

1) Speed assignment system:

- **Lookahead system**: The lookahead system is able to gather curvature information from a predefined path which is not arc-length parameterized. The information is gathered a given along-path distance ahead of the vehicle position. Specifically, the curvature information gathered from the path is stored for use in a speed assignment algorithm.
- **Speed assignment algorithm**: The speed assignment algorithm employs curvature information and vehicle maneuverability constraints to plan purposeful reference speeds along the path. The commanded speeds are reachable and makes it easier for the vehicle to follow the path.

2) Path convergence algorithms:

- **Steering algorithm**: An existing steering law for regularly parameterized paths is modified. Specifically, the lookahead distance is manipulated using different techniques to incorporate vehicle maneuverability constraints in the steering assignments.
- **Semi-constant lookahead**: An investigation of the steering law's equations is done to yield a constant lookahead distance which utilizes the vehicle's angular speed constraint. The lookahead is constant for a given speed.
- **Variable lookahead**: Employment of clothoid properties to construct a variable lookahead distance, where the both the angular speed and angular acceleration constraint of the vehicle are taken into consideration.

- **Convergence heuristics**: Additional modifications to the look-ahead distance is provided to make it more robust for course errors.

3) Path-tracking system:

- **Online path generation**: The online path generator creates a continuous path from sampled positions during the course of execution. Two different approaches for path generation are reviewed, with considerations toward AIS-retrieved data or only position data. The generated paths are applicable for path scenarios such as path following or path tracking.
- **Path-tracking speed assignment**: The path-tracking speed assignment algorithm employs along-path distances between the target and the follower to output a bounded speed assignment for the following vehicle.

4) Path-planning scenarios:

- **Path planner for lawn-mower patterns**: A path planner for lawn-mower patterns using clothoids is proposed. This method creates feasible parameterizations for a given speed under vehicle maneuverability constraints on angular speed and angular acceleration.
- **Comparison to η-splines** The clothoid approach is compared to an η-spline approach, emphasizing the curvature rate of the two approaches.

1.3 Thesis outline

The thesis starts with mathematical definitions and common abbreviations in Chapter 2. This chapter is followed by a introduction of relevant path parameterizations and their properties. Chapter 4 covers background information on the vehicle's characteristics and motion control fundamentals.

Then, each topic considered in this thesis is treated in its own section of Chapter 5. Here, specific motivation, previous work and additional background information are presented.

After that, each method is verified through simulations in Chapter 6. In Chapter 7, conclusions are drawn, and future work are proposed. Finally, an Appendix follows with computational details and further background theory.

Chapter 2

Preliminaries

In this chapter, frequently used mathematical definitions and notations are presented. Furthermore, uncommon abbreviations are also summarized.

2.1 Mathematical definitions and notations

In this thesis, vectors and matrices are written in boldface, while scalars are not. Time derivatives of $\boldsymbol{x}(t)$ are denoted $\dot{\boldsymbol{x}}$, $\ddot{\boldsymbol{x}}$, $\boldsymbol{x}^{(3)}, \ldots, \boldsymbol{x}^{(i)}$. Derivatives with respect to some other variable is denoted $\boldsymbol{x}'$, $\boldsymbol{x}''$, $\boldsymbol{x}^{(3')}, \ldots \boldsymbol{x}^{(i')}$. It is clear from the context which variable it has been differentiated with respect to. The norm $|\boldsymbol{x}|$ is the Euclidean norm $\sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$, while $|x|$ is the absolute value of $x \in \mathbb{R}$. The dot product of two vectors of the same dimension is defined as $\boldsymbol{a} \cdot \boldsymbol{b} \triangleq \boldsymbol{a}^\top \boldsymbol{b}$. The index set $\mathcal{I}^n \subset \mathbb{N}_0$ is all natural number from 0 to n including n: $\{0, 1, 2, \ldots, n\}$. The column vector is stated as $\operatorname{col}[\boldsymbol{x}, \boldsymbol{y}] \triangleq \left[\boldsymbol{x}^\top, \boldsymbol{y}^\top\right]^\top$.

2.2 Abbreviations

CLF	Control Lyapunov Function
ECEF	Earth-Centered, Earth-Fixed
ECI	Earth-Centered Intertial
DOF	Degree Of Freedom
DP	Dynamic Positioning
FIFO	First-In, First-Out
GES	Globally Expontentially Stable
GNC	Guidance, Navigation and Control
LES	Locally Exponentially Stable
LOS	Line-Of-Sight
MPC	Model Predictive Control
MPG	Model Predictive Guidance
NED	North East Down
SNAME	Society of Naval Architects and Marine Engineers
UAV	Unmanned Aerial Vehicle
UGAS	Uniformly Globally Asymptotically Stable
ULES	Uniformly Locally Exponentially Stable
USV	Unmanned Surface Vehicle

Chapter 3

Path parameterizations and their characteristics

3.1 The path parameterization

3.1.1 Parametric curve

A parametric curve is a univariate parametric representation of a set of equations, where each state is parameterized by a common scalar variable $\varpi \in \mathbb{R}$. The position $\boldsymbol{p}_p(\varpi) = [x_p(\varpi), y_p(\varpi)]^\top \in \mathbb{R}^2$ belongs to the curve, which is a one-dimensional manifold that can be expressed by the set (Breivik & Fossen 2009)

$$\mathcal{P} \triangleq \{\boldsymbol{p} \in \mathbb{R}^2 \,|\, \boldsymbol{p} = \boldsymbol{p}_p(\varpi) \,\forall\, \varpi \in \mathbb{R}\}. \tag{3.1}$$

Both *curve* and *path* have the same meaning in this thesis, and are used interchangeably.

3.1.2 Piecewise parametric curve

A single curve cannot represent complex shapes without high complexity. Hence, by dividing a shape into smaller curve segments with limited validity interval, the shape can be accurately approximated without the potential of ill-conditioned parameters, inaccuracy and Runge's phenomenon (Runge 1901), which a single curve may suffer from. Define a curve segment,

enumerated i, which similarily to a single curve, also is a one-dimensional manifold expressed by the set

$$\mathcal{P}_i \triangleq \{\boldsymbol{p}_i \in \mathbb{D}_i \subset \mathbb{R}^2 \,|\, \boldsymbol{p}_i = \boldsymbol{p}_{i,p}(\varpi) \,\forall\, \varpi \in \mathcal{I}_i = [\varpi_{i,0},\, \varpi_{i,1}] \subset \mathbb{R}\}. \quad (3.2)$$

We assume that each curve segment has a predecessor and a successor, with the exception of the first and last curve segment. Adjacent curve segments have successive indexing, that is, $\mathcal{P}_k$ has a predecessor $\mathcal{P}_{k-1}$ and a successor $\mathcal{P}_{k+1}$. Furthermore, we assume that the curve segments are connected, meaning that the end point of one curve segment has the same planar position as the start point of the successor. This notation allows each curve segment to have a validity interval $\mathcal{I}_i$ which best serves a proper representation of the particular parameterization.

Formally, a piecewise curve can written as a superset of n curve segments:

$$\mathcal{P}_s = \bigcup_{i=1}^{n} \mathcal{P}_i. \quad (3.3)$$

This notation implies that the piecewise curve $\boldsymbol{p}$ is defined in

$$\boldsymbol{p} \in \mathbb{D}_s = \bigcup_{i=1}^{n} \mathbb{D}_i \subset \mathbb{R}^2. \quad (3.4)$$

The validity region $\mathbb{D}_s$ need not be defined in the whole planar space, only in a reasonable vicinity of the vehicle, making it well defined for all practical purposes. In Section 3.1.6 additional restrictions to the resulting curve are presented.

3.1.3 Continuity

In a Cartesian coordinate system it is interesting to have a precise meaning of what continuity of a path really means. In the following, we will define *parametric* and *geometric* continuity for planar parameterizations and elaborate on the differences between these two types of continuities.

Parametric continuity

Defintion 3.1 ($\mathcal{C}^n$ and regularity (Barsky & DeRose 1984))**.**

a) A scalar function $f(\varpi)$ belongs to the class $\mathcal{C}^n$ on $[\varpi_0, \varpi_1]$ if it is n-times continuously differentiable on $[\varpi_0, \varpi_1]$. It is regular if

$$\frac{df}{d\varpi} \neq 0 \quad \forall \quad \varpi \in [\varpi_0, \varpi_1]. \tag{3.5}$$

b) A parameterization $\boldsymbol{p}_p(\varpi) = [x_p(\varpi), y_p(\varpi)]^\top$, $\varpi \in [\varpi_0, \varpi_1]$ is $\mathcal{C}^n$ if each of the coordinate functions $x_p(\varpi)$ and $y_p(\varpi)$ is $\mathcal{C}^n$ on $[\varpi_0, \varpi_1]$. It is regular if

$$\frac{d\boldsymbol{p}_p}{d\varpi} \neq 0 \quad \forall \quad \varpi \in [\varpi_0, \varpi_1]. \tag{3.6}$$

A regular parameterization means that the path never degenerate into a point. A parameterization that satisfies Definition 3.1 of order n is said to be $\mathcal{C}^n$ parametric continuous.

Example 3.1. Define

$$\boldsymbol{p}_1(\varpi) = \begin{bmatrix} \varpi \\ -\varpi \end{bmatrix}, \varpi \in [0, \pi/4], \tag{3.7}$$

$$\boldsymbol{p}_2(\varpi) = \begin{bmatrix} \sin(\varpi + \pi/4) + \pi/4 - \sin(\pi/4) \\ \cos(\varpi + \pi/4) - \pi/4 - \cos(\pi/4) \end{bmatrix}, \varpi \in [0, \pi]. \tag{3.8}$$

Let the parameterizations just defined be stitched together into a single curve, see Figure 3.1.

From Figure 3.1 it is clear that this curve is $\mathcal{C}^0$ continuous since they join at a join point $\mathcal{J}$. For the curve to be $\mathcal{C}^1$ continuous, however, the first derivative must be equal at $\mathcal{J}$, that is

$$\left.\frac{d\boldsymbol{p}_1}{d\varpi}\right|_{\varpi=\pi/4} \stackrel{!}{=} \left.\frac{d\boldsymbol{p}_2}{d\varpi}\right|_{\varpi=0}. \tag{3.9}$$

Evaluating the derivatives at $\mathcal{J}$ reveal that the curve is not $\mathcal{C}^1$ continuous:

$$\left.\frac{d\boldsymbol{p}_1}{d\varpi}\right|_{\varpi=\pi/4} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \neq \left.\frac{d\boldsymbol{p}_2}{d\varpi}\right|_{\varpi=0} = \begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix}. \tag{3.10}$$

The conclusion of Example 3.1 is that a curve is not necessarily parametric continuous, even though it may appear smooth geometrically. In some geometric problems, only the path needs to be smooth, not the parameterization itself. This motivates a different type of continuity that only requires that the resulting curve is sufficiently smooth *geometrically*.

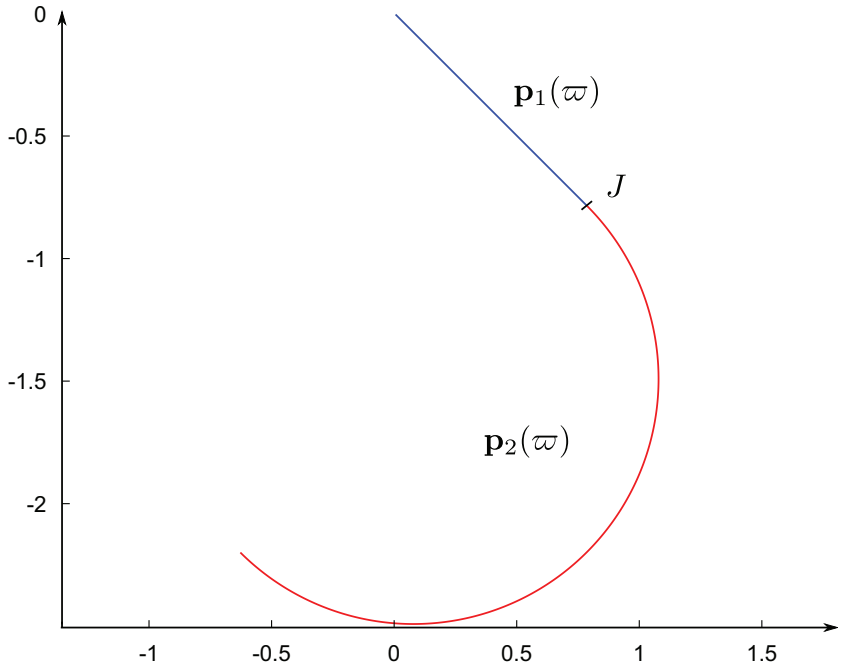


Figure 3.1: The curve is only $\mathcal{C}^0$ continuous.

Geometric continuity

Defintion 3.2 (Geometric continuity (Barsky & DeRose 1984))**.** Let $\boldsymbol{p}_i(\varpi)$ and $\boldsymbol{p}_{i+1}(\varpi)$ be regular $\mathcal{C}^n$ parameterizations such that $\boldsymbol{p}_i(\varpi_1) = \boldsymbol{p}_{i+1}(\varpi_0) = \mathcal{J}$, where $\mathcal{J}$ is a simple point. They meet with n^{th} order geometric $\mathcal{G}^n$ continuity at $\mathcal{J}$ if $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ can be reparameterized to have $\mathcal{C}^n$ parametric continuity at $\mathcal{J}$.

Geometric continuity up to $n = 2$ can be interpreted as:

$\mathcal{G}^0$: The curve is connected.

$\mathcal{G}^1$: The unit tangent vector is continuous.

$\mathcal{G}^2$: The curvature is continuous.

The different degrees of geometric continuity is illustrated in Figure 3.2.

Example 3.2. Consider the curve presented in Example 3.1. It is obvious that the curve is $\mathcal{G}^0$ continuous, and by normalizing (3.10), we see that the

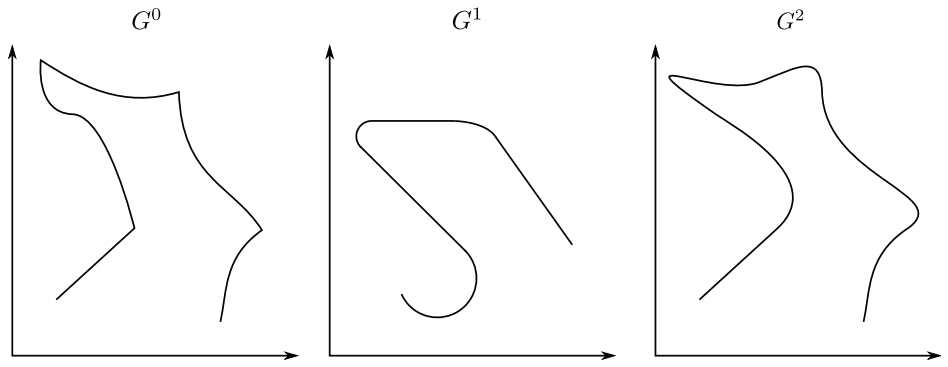


Figure 3.2: Left: $\mathcal{G}^0$ continuous curve. Middle: $\mathcal{G}^1$ continuous curve. Right: $\mathcal{G}^2$ continuous curve.

unit tangent vectors are equal. Hence, the curve is $\mathcal{G}^1$ continuous, as expected after inspecting Figure 3.1.

3.1.4 Arc length of a path

The arc length between two points is the distance a particle needs to travel along the path when moving from one point to the other. This positive scalar number, which is deducted from Pythagoras' rule, is defined by

$$s(\varpi) = \int_{\alpha}^{\varpi} |\boldsymbol{p}_p'(\tau)| d\tau, \tag{3.11}$$

where the integration variable is denoted by τ to distinguish it from the upper limit ϖ.

Some paths are arc-length parameterized which means that $\varpi = s$. The paths belonging to $\mathcal{P}_s$ however, are generally not arc-length parameterized, which complicates the determination of the arc length between arbitrary path points. Specifically, (3.11) cannot be solved explicitly in the general case.

Arc-length speed

The time derivative of the arc length is

$$\begin{aligned} \dot{s}(\varpi) &= \frac{d}{dt}\left(\int_{\alpha}^{\varpi} |\boldsymbol{p}_p'(\tilde{\varpi})| d\tilde{\varpi}\right) \\ &= |\boldsymbol{p}_p'(\varpi)| \dot{\varpi}. \end{aligned} \tag{3.12}$$

Let the along-path speed of a particle be defined as

$$U_p \triangleq \sqrt{\dot{x}_p^2 + \dot{y}_p^2}. \tag{3.13}$$

Furthermore, by the chain rule we have

$$\dot{x}_p(\varpi) = \frac{dx_p}{d\varpi}\frac{d\varpi}{dt} = x_p'\dot{\varpi}, \tag{3.14}$$

and similarly for $y_p(\varpi)$, such that we can rewrite (3.13) to

$$U_p(t) = \sqrt{(x_p'\dot{\varpi})^2 + (y_p'\dot{\varpi})^2} = \sqrt{x_p'^2 + y_p'^2}\cdot\dot{\varpi},$$

where we define

$$\sqrt{x_p'^2 + y_p'^2} \triangleq |\boldsymbol{p}_p'(\varpi)|, \tag{3.15}$$

resulting in

$$\dot{\varpi}(t) = \frac{U_p(t)}{\sqrt{x_p'^2 + y_p'^2}} = \frac{U_p(t)}{|\boldsymbol{p}_p'(\varpi)|}, \tag{3.16}$$

which describes the relationship between the path variable's speed and the along-path speed.

Substituting (3.16) into (3.12) we get

$$\dot{s}(\varpi) = U_p(t), \tag{3.17}$$

which confirms that the time rate of change of the arc length is equal to the along-path speed.

3.1.5 Path curvature

Every point of a curved path has a tangent circle, called the osculating circle, see Figure 3.3. This circle has the same curvature as the path at the given point. For a straight line, the radius of this circle is infinity, but in general, it varies with the path. By knowing the curvature of a path point, the corresponding osculating circle can be calculated by

$$R(\varpi) = \frac{1}{\kappa(\varpi)}, \tag{3.18}$$

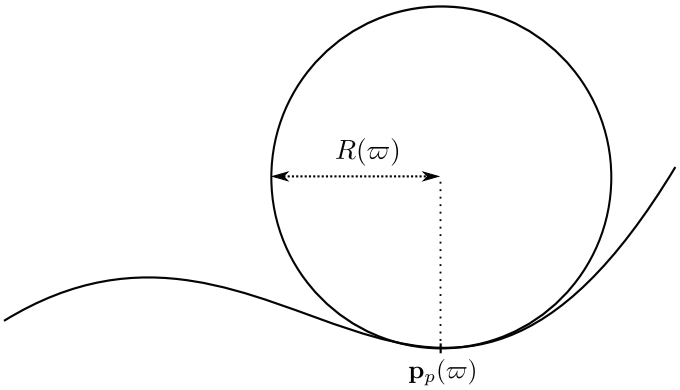


Figure 3.3: Osculating circle at $\boldsymbol{p}_p(\varpi)$.

where $\kappa(\varpi)$ is the curvature at the path point. The curvature is defined as the magnitude of rate of change of the angle of the tangent vector with respect to arc length (SpringerLink 2010)

$$\kappa(\varpi) = \left| \frac{d\boldsymbol{T}(\varpi)}{ds} \right|, \tag{3.19}$$

where $\boldsymbol{T}(\varpi)$ is the unit tangent vector of the path parameterization. Since most parameterizations are not arc length parameterized, this calculation is not as trivial as it may seem from the definition. The curvature can in the general case be shown to be

$$\kappa(\varpi) \;=\; \frac{|\boldsymbol{p}'_p(\varpi) \times \boldsymbol{p}''_p(\varpi)|}{|\boldsymbol{p}'_p(\varpi)|^3} = \frac{|x'_p y''_p - y'_p x''_p|}{|\boldsymbol{p}'|^3} \tag{3.20}$$

where we have used that $\boldsymbol{p}'_p(\varpi) \triangleq d\boldsymbol{p}_p(\varpi)/d\varpi$. See Appendix C.1 for computational details.

Signed curvature

In some cases it is convenient to know in what direction the path curves. The signed version of the path curvature is

$$\varkappa(\varpi) \;=\; \frac{\boldsymbol{p}'_p(\varpi) \times \boldsymbol{p}''_p(\varpi)}{|\boldsymbol{p}'_p(\varpi)|^3} = \frac{x'_p y''_p - y'_p x''_p}{|\boldsymbol{p}'_p|^3}. \tag{3.21}$$

The convention is that the curvature is signed in accordance with the right-hand rule. That is, for $x'_p > 0$, a path with positive curvature curves in counter-clockwise direction, while a path with negative curvature curves in clockwise direction, see Figure 3.4.

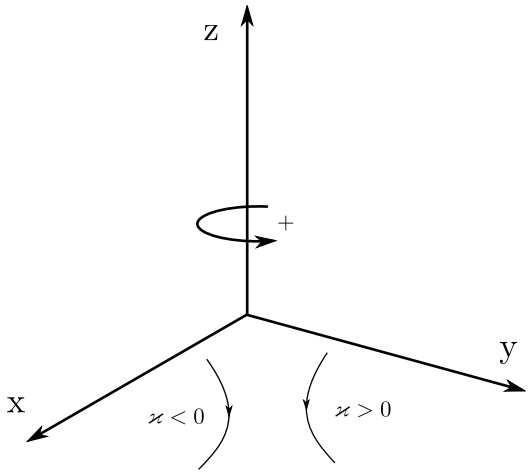


Figure 3.4: The convention of signed curvature.

3.1.6 Path parameterizations in this thesis

The path parameterizations used in this thesis are $\mathcal{P}_r \subset \mathcal{P}_s$. The subset $\mathcal{P}_r$ has additional restrictions to the curves, making it applicable to path maneuvering scenarios. The subset contains curves with $\mathcal{G}^2$-continuous paths and bounded curvature, which means that the path tangent $|\boldsymbol{p}'_p(\varpi)| \triangleq |d\boldsymbol{p}_p(\varpi)/d\varpi|$ is non-zero and finite for all points on the curve. In practice, this means that such curves never degenerate into a point nor have corners.

3.2 Splines

In a mathematical context, a spline is a piecewise curve where each curve segment is a polynomial parameterization of a finite degree n. A planar spline segment can be written as:

$$x = a_n\varpi^n + a_{n-1}\varpi^{n-1}\cdots a_1\varpi + a_0, \tag{3.22a}$$
$$y = b_n\varpi^n + b_{n-1}\varpi^{n-1}\cdots b_1\varpi + b_0, \tag{3.22b}$$
$$\varpi \in [\varpi_0,\, \varpi_1]\,, \tag{3.22c}$$

where the polynomial coefficients are real numbers. Compactly, we can write

$$\boldsymbol{p}(\varpi)^\top = \begin{bmatrix} x(\varpi) \\ y(\varpi) \end{bmatrix}^\top = \begin{bmatrix} 1 & \varpi & \cdots & \varpi^n \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{bmatrix}, \quad (3.23a)$$

$$\varpi \in [\varpi_0, \varpi_1], \quad (3.23b)$$

$$(a_i, b_i) \in \mathbb{R} \forall\, i \in \mathcal{I}^n \subset \mathbb{N}_0. \quad (3.23c)$$

Splines have been used in many different applications, including vehicle body design, computer-aided graphical design, curve-fitting, shape generation and path planning. A common feature of all these problems is that the objective is to somehow represent a finite number of discrete data points, denoted *control polygon*, in a continuous manner. The research community is replete with different types of splines and ways to obtain the polynomial coefficients, given a specific control polygon. Basis splines, Catmull-Rom splines, Quintic splines, $\boldsymbol{\eta}$-splines and β-splines are all different kinds of approaches for representing a control polygon parametrically. Generally, we can distinguish between two types of approaches:

Interpolating splines are designed such that the curve pass through each discrete data point.

Approximating splines do not necessarily pass through every data point, but approximate the trend of the data points.

Common for all types of splines are the basis polynomials, or *blending functions*. The blending functions blend the points of the control polygon together, yielding a continuous curve. Given a control polygon $\boldsymbol{V}_0, \boldsymbol{V}_1, \cdots, \boldsymbol{V}_m$, where $\boldsymbol{V}_i \in \mathbb{R}^r$, a spline can be written as a function of the control polygons:

$$\boldsymbol{p}(\varpi) = \sum_{i=0}^{m} W_i(\varpi) \boldsymbol{V}_i, \quad \forall\, \varpi \in [\varpi_0, \varpi_1], \quad (3.24)$$

where $W_i(\varpi)$ is the i-th blending function. There are many different ways of obtaining the blending functions, and they greatly depend on the desired properties of the curve. In the remaining part of this thesis only planar positions in the control polygon will be considered, that is $\boldsymbol{V}_i \in \mathbb{R}^2$. It

should, however, be emphasized that splines treat each degree of freedom separately, making it easily scalable.

With the vast possibilities for creating a spline curve, it is required by the implementor to choose the approach with the most desirable properties for the specific problem at hand. To be able to do this, important properties of each spline should be compared with each other, choosing the spline which is most applicable to the intended application. In (Blanc & Schlick 1995), one such check-list is provided, presenting important properties splines can possess. These properties are also presented in (Weston 2002) and some of them are restated here for convenience:

Affine invariance: Geometric properties remain unchanged after linear transformations, such as rotations, and translations.

Convex hull: The spline is contained within the polygon created by the control polygon.

Shape parameters: Add additional degrees of freedom to control the shape of the spline.

Local/Global control: Manipulation of shape parameters or control vertices may affect the spline locally or globally. A spline has L^p locality if the manipulation affects at most p curve segments.

Continuity where segments meet: A specific degree of either parametric or geometric continuity characterize each spline method.

In the next section, the most sentral spline used in this thesis is presented.

3.2.1 The Bézier curve

A Bézier curve is a polynomial parametric curve with origin from France. More specifically, algorithms for presenting Bézier curves were invented by Paul de Casteljau in 1959, who worked for Citroën (Casteljau 1959). This type of curve was also used by a French engineer named Pierre Bézier in the early 1960's to create smooth automobile bodies for Renault. Since the name de Castelajau already is used on (one of) the algorithms for calculating the Bézier curves; the *de Castelajau's algorithm*, the curve has been given the name *Bézier curves*.

Matematically, the general form for a planar Bézier curve of degree n is

$$\boldsymbol{p}(\varpi) = \sum_{i=0}^{n} B_{i,n}(\varpi)\boldsymbol{V}_i, \quad \varpi \in [0,\, 1], \tag{3.25}$$

where $B_{i,n}(\varpi)$ are the Bernstein basis polynomials of degree n and $\boldsymbol{V}_i$ are planar positions in the control polygon. By carefully stitching Bézier curves together, a *Bézier spline* is obtained.

In (Joy 2000) the Bernstein basis polynomials of degree n are defined by

$$B_{i,n}(\varpi) = \binom{n}{i}\varpi^i(1-\varpi)^{n-i}, \quad \forall\, i \in \mathcal{I}^n. \tag{3.26}$$

The binomial coefficients are defined as

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}. \tag{3.27}$$

Example 3.3. The Bernstein basis polynomials of degree 3 are

$$B_{0,3}(\varpi) = (1-\varpi)^3 \tag{3.28a}$$
$$B_{1,3}(\varpi) = 3\varpi(1-\varpi)^2 \tag{3.28b}$$
$$B_{2,3}(\varpi) = 3\varpi^2(1-\varpi) \tag{3.28c}$$
$$B_{3,3}(\varpi) = \varpi^3. \tag{3.28d}$$

The basis functions are displayed in Figure 3.5.

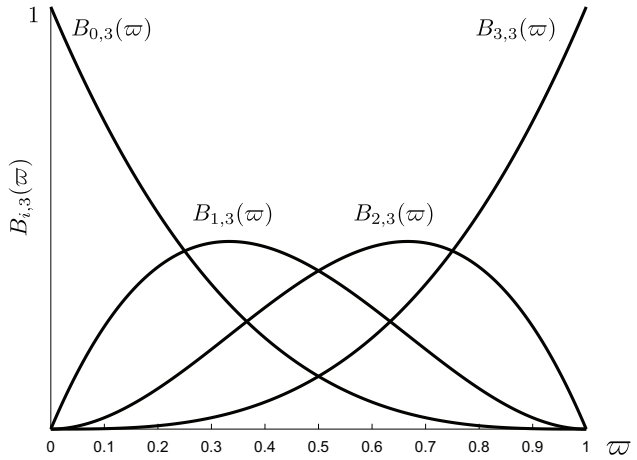


Figure 3.5: The Bernstein basis polynomials.

Define the four planar points

$$\begin{aligned} \boldsymbol{V}_0 &= [1 \quad 1]^\top, & \boldsymbol{V}_1 &= [2 \quad 5]^\top \\ \boldsymbol{V}_2 &= [6 \quad 7]^\top, & \boldsymbol{V}_3 &= [7 \quad 2]^\top, \end{aligned}$$

to be the control polygon of a simple example. Using (3.25) we get a Bézier curve of degree 3. The curve can be seen in Figure 3.6.

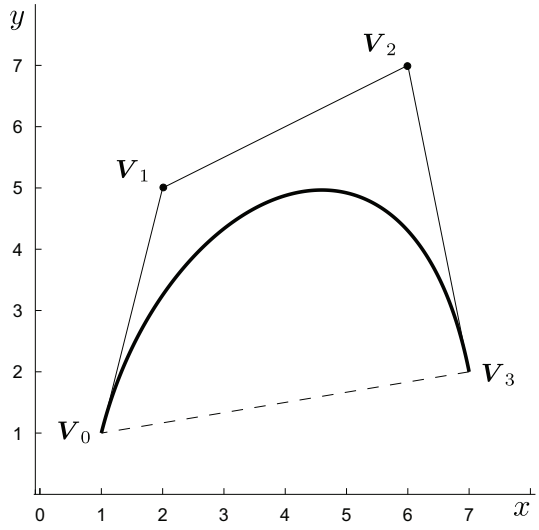


Figure 3.6: A Bézier curve.

Figure 3.6 actually illustrates several important properties of the Bézier curve:

Interpolating: The end points are interpolated.

Approximating: Intermediate points are only approximated.

Convex hull: The curve is within the convex hull of the polygon.

Furthermore, it can be proven by induction that the sum of the Bernstein basis polynomials always is equal to unity:

$$\sum_{i=0}^{n} B_{i,n}(\varpi) \quad = \quad 1, \quad \forall\, \varpi \in [0,\, 1]\,, \tag{3.29}$$

making the Bézier curve affine invariant (Joy 2000).

Matrix form

In some cases it is convenient to organize the Bézier curve in a matrix from where the parameters are collected in an own parameter vector (Joy 2000):

$$\boldsymbol{p}(\varpi)^\top = \begin{bmatrix} 1 & \varpi & \cdots & \varpi^n \end{bmatrix} \begin{bmatrix} b_{0,0} & 0 & \cdots & 0 \\ b_{1,0} & b_{1,1} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ b_{n,0} & b_{n,1} & \cdots & b_{n,n} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_0^\top \\ \boldsymbol{V}_1^\top \\ \vdots \\ \boldsymbol{V}_n^\top \end{bmatrix}, \quad (3.30)$$

where $b_{i,j}$ are coefficients defined by:

$$b_{i,j} = (-1)^{i-j} \binom{n}{i} \binom{i}{j}. \quad (3.31)$$

This arrangement makes it easier to find derivatives of arbitrary order, since only the parameter vector needs to be differentiated, not each Bernstein basis polynomial of the parameterization.

By introducing additional control points in the control polygon, the desired degree of geometric continuity of a Bézier spline can be achieved by carefully placing the control points of each Bézier curve (DeRose & Barsky 1988).

3.3 Clothoids

Euler discovered a special spiral when he considered the problem of "an elastic spring freely coiled in the form of a spiral" (Archibald 1918). The spiral, called the Euler spiral, was given the name *clothoid* by Cesaro when he studied the properties of the spiral. Clothoid comes from Greek meaning "to twist by spinning" (Cesaro 1886). The spiral is defined parametrically as

$$\boldsymbol{p}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = a^\star \begin{bmatrix} C(t) \\ S(t) \end{bmatrix}, a^\star > 0, \quad (3.32)$$

where the Fresnel integrals $C(t)$ and $S(t)$ are

$$C(t) = \int_0^t \cos(\frac{1}{2}\pi u^2) du, \quad (3.33a)$$

$$S(t) = \int_0^t \sin(\frac{1}{2}\pi u^2) du. \quad (3.33b)$$

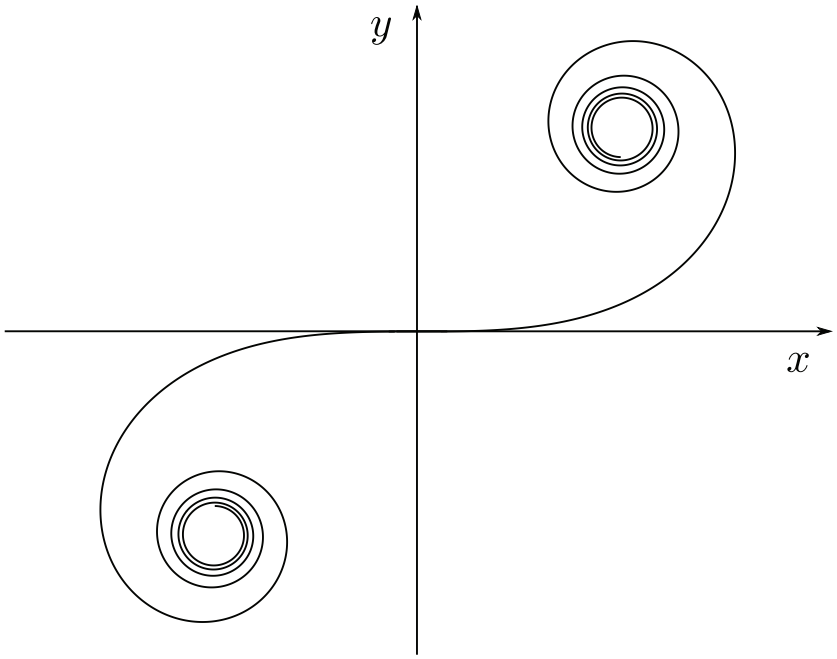


Figure 3.7: The Cornu Spiral.

The Euler spiral was first plotted accurately by Cornu and hence also carries the name Cornu spiral (Cornu 1874).

It is possible to write the expressions for the Fresnel integrals so that they depend on the angle between the x-axis and the tangent vector (Abramowitz & Stegun 1972)

$$C(\vartheta) = \frac{1}{\sqrt{2\pi}} \int_0^{\vartheta} \frac{\cos(u^2)}{\sqrt{u}} du, \tag{3.34a}$$

$$S(\vartheta) = \frac{1}{\sqrt{2\pi}} \int_0^{\vartheta} \frac{\sin(u^2)}{\sqrt{u}} du. \tag{3.34b}$$

This arrangement requires however that the parameter is $\vartheta \geq 0$. In the following we will use the scaled versions of (3.34) as in (Meek & Walton 2004)

$$C(\vartheta) = \int_0^{\vartheta} \frac{\cos(u^2)}{\sqrt{u}} du, \tag{3.35a}$$

$$S(\vartheta) = \int_0^{\vartheta} \frac{\sin(u^2)}{\sqrt{u}} du, \tag{3.35b}$$

$$\vartheta \geq 0. \tag{3.35c}$$

A clothoid with scaling factor $a = \frac{a^\star}{\sqrt{2\pi}}$ can then be written as

$$\boldsymbol{p}(\vartheta) = a \begin{bmatrix} C(\vartheta) \\ S(\vartheta) \end{bmatrix}, \vartheta \geq 0. \tag{3.36}$$

3.3.1 Properties of a clothoid

In the following we will investigate the properties of a clothoid defined by (3.36).

The tangent vector is

$$\boldsymbol{p}'(\vartheta) = \frac{a}{\sqrt{\vartheta}} \begin{bmatrix} \cos(\vartheta) \\ \sin(\vartheta) \end{bmatrix}, \vartheta \geq 0, \tag{3.37}$$

and from this it is evident that ϑ is the angle of the tangent vector with respect to the x-axis.

By calculating the curvature of a clothoid we get

$$\kappa(\vartheta) = \frac{\boldsymbol{p}'(\vartheta) \times \boldsymbol{p}''(\vartheta)}{|\boldsymbol{p}'(\varpi)|^3} = \frac{\sqrt{\vartheta}}{a}, \tag{3.38}$$

see Appendix C.1.1 for computational details.

The center of the osculating circle at a path point ϑ is found by the normal and the curvature at the particular point:

$$\boldsymbol{m}(\vartheta) = \boldsymbol{p}(\vartheta) + \frac{1}{\kappa(\vartheta)} \begin{bmatrix} -\sin(\vartheta) \\ \cos(\vartheta) \end{bmatrix}. \tag{3.39}$$

The arc length is (3.11)

$$\begin{aligned} s(\vartheta) &= \int_0^\vartheta |\boldsymbol{p}'(u)| du \\ &= \int_0^\vartheta \frac{a}{\sqrt{u}} du = 2a\sqrt{\vartheta}, \end{aligned} \tag{3.40}$$

and by expressing the curvature of the clothoid (3.38) as a function of the arc length we get

$$\kappa(\vartheta) = \frac{s(\vartheta)}{2a^2}. \tag{3.41}$$

This equation shows that the curvature is proportional to the arc length of the clothoid. This property suggest that if a particle follows the clothoid at a constant speed, the angular speed changes linearly. The relationship between the speed and the angular speed can by the use of (3.18) be rewritten to

$$U = \omega R = \frac{\omega}{\kappa(\vartheta)},$$

which together with (3.41) gives

$$\omega(\vartheta) = U\frac{s(\vartheta)}{2a^2}. \tag{3.42}$$

This confirms that the angular speed is proportional to the arc length. For a constant speed U, the angular acceleration is

$$\dot{\omega}(\vartheta) = \dot{s}(\vartheta)\frac{U}{2a^2}, \tag{3.43}$$

and since the time derivative of the arc length is the speed of the particle itself (3.17), we get

$$\dot{\omega} = \frac{U^2}{2a^2}, \tag{3.44}$$

which concludes that the angular acceleration in constant when following a clothoid at constant speed.

3.3.2 Applications of clothoids

The properties of clothoids are beneficial in many different applications. Specifically, in highway design, clothoids have been used as transition curves to acquire smooth changes in the angular velocity, (Higgins 1921, Talbot 1927, Baass 1984). Furthermore, in robot path planning, the use of clothoids has resulted in smooth and feasible trajectories for car-like vehicles (Shin & Singh 1990, Fleury et al. 1995, Scheuer & Fraichard 1997). In the design of roller coaster loops, the properties of clothoids have also been utilized (Pendrill 2005). When designing cars, the body surface needs be $\mathcal{G}^2$ continuous to have good aesthetics. Similarly, in computer vision, completing shapes with clothoids may result in more appealing graphical shapes (Kimia et al. 2003).

Figure 3.8: Left: The roller coaster *Loopen* produced by Vekoma in 1988 is found at Tusenfryd in Norway. The teardrop-shaped loop is designed by the use of clothoids. Courtesy of www.physics.gu.se.
Right: *The Troll Ladder*, a mountain road in Rauma, Norway. Courtesy of Ronnie Haug.

Chapter 4

Vehicle motion control

The subject of vehicle motion control can be divided into four scenarios, each entailing specific characteristics (Breivik 2010):

Target tracking: Track the motion of a target. There exist no future information about the target movement, only instantaneous motion. Tracking of a stationary target is called *point stabilization*.

Path following: The objective is to follow a predefined path. There is only a spatial constraint, the path itself being it.

Path tracking: Track a target that is moving on a predefined path. Now there are both spatial and temporal constraints. This scenario is also often called *trajectory tracking*.

Path maneuvering: Utilize knowledge about vehicle maneuverability to negotiate a predefined path. This often means to somehow optimize this negotiation.

In this thesis the focus is on path maneuvering and path tracking. As can be seen from the definitions above, path maneuvering is a subset of path following. Hence, path-maneuvering methods can also be used to handle path-following scenarios, but in a more optimal manner. In fact, path-maneuvering methods can also be used to solve path-tracking objectives by employing specific speed constraints. This sheds light on the applicability of path-maneuvering methods. In (Skjetne et al. 2004), the maneuvering problem is divided into two tasks:

Figure 4.1: A Lockheed Martin F-22 Raptor being maneuvered in a 3-dimensional work space. Courtesy of www.wikivisual.com.

1. **Geometric task:** Force the vehicle to converge to and follow the desired, predefined path.

2. **Dynamic task:** Fulfill desired speed or perhaps acceleration constraints along the path.

The desired speed can either be a predefined speed profile or inputs from the pilot. In most articles about path following however, the main concern has indeed been the geometric task. The dynamic task has usually been solved by employing a constant forward speed. In general, this is a non-optimal solution seen from a path-maneuvering point of view. If the speed is so low that every curvature can be negotiated, the speed often is too conservative. On the other hand, choosing a too optimistic constant speed will lead to failure of keeping on the path when the curvature becomes too large.

4.1 Motion control fundamentals

To be able to discuss specific approaches of the motion control scenarios, some terminology and definitions must be presented.

4.1.1 Operating spaces

The most relevant operating spaces of a vehicle are (Breivik & Fossen 2009)

Work space: A physical space where the vehicle moves. For a car, this is the planar space:

$$\mathcal{M}_W \in \mathbb{R}^2. \tag{4.1}$$

Configuration space: Defines a set of variables which is sufficient to specify all points of a rigid-body vehicle in the work space. For a car, this is the planar position and the orientation of the car:

$$\mathcal{M}_C \in \mathbb{R}^2 \times \mathbb{S}, \tag{4.2}$$

where $\mathbb{S} \in [0,\, 2\pi]$. A configuration variable is often denoted as a *degree of freedom* (DOF).

4.1.2 Actuation properties

When controlling a vehicle, it is often distinguished between different degrees of actuation (Fossen 2002)

Full actuation: Independent control of each DOF can be obtained simultaneously.

Underactuation: The converse of full actuation, that is, independent control of each DOF is not possible.

The path scenarios considered in this thesis is a subset of all possible tasks in the configuration space. Specifically, when following a path, there are no explicit restrictions on the orientation of the vehicle. This is an important relaxation, since underactuated vehicle cannot independently control each degree of freedom. Hence, the considered path scenarios can be achieved with underactuated vehicles, provided that the paths are feasible with respect to the vehicle maneuverability constraints.

4.1.3 Kinematics

Kinematics only includes the geometrical aspects of motion. The guidance and control laws need a mathematical description of the vehicle relative some desired position and velocity. Hence, to accurately describe the vehicle in the configuration space, each degree of freedom must be described relative a reference frame.

Reference frames

We distinguish between global and local reference frames. The origin of global reference frames are the Earth's center, with the z-axis parallel to the axis of rotation, pointing toward the North pole. These reference frames are used to describe vehicle motion globally. Local reference frames have its origin placed on a geographically stationary point, or on the vehicle itself. A summary of the reference-frame definitions defined in (Fossen 2002) are:

Global reference frames

ECI: Earth-centered inertial frame $\{i\}$. This frame follows Earth's elliptic motion around the sun, but does not rotate around its own axis in the manner the Earth in fact does. A geographical position on Earth is not a constant coordinate in this frame.

ECEF: The Earth-centered Earth-fixed frame $\{e\}$ is similar to {i}, but also rotates with an angular speed ω_e relative the z_i-axis, making each geographical position on Earth uniquely determined in this reference frame.

Local reference frames

NED: The North-East-Down frame $\{n\}$. The origin of this frame is defined relative to Earth's reference ellipsoid. As the name suggests, the x-axis points North, the y-axis East, and the z-axis downward toward Earth's center. By using the angles denoted *longitude* and *latitude*, the location of the frame's origin can be determined.

BODY: The body-frame $\{b\}$ is fixed to a point on the vehicle. Thus, by considering the position and orientation of this frame relative a stationary reference frame, the specific configuration of the vehicle can be determined, see Figure 4.3. Furthermore, the time rate of change of the this frame's positions are used to determine the vehicle's linear and angular velocity.

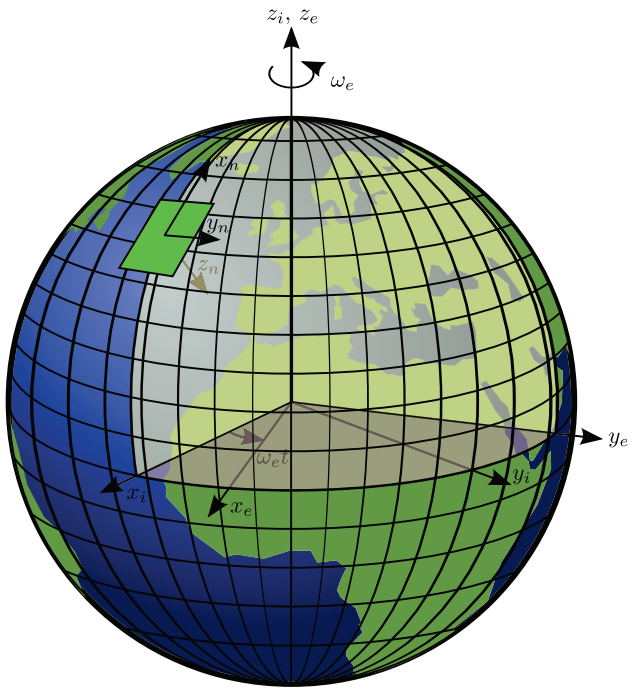


Figure 4.2: The origin of the ECEF-frame and the ECI-frame are identical. The z_n-axis of the NED-frame points toward the origin of the global reference frames. Courtesy of (Hildrestrand 2010).

There exist a common nomenclature for marine motion variables (SNAME 1950). Necessary quantities to describe the vehicle in the configuration space, and the vehicle's linear and angular velocities will now be defined. The configuration of a vehicle is defined relative a stationary reference frame. When the stationary reference frame is the {n}-frame, the following definitions can be made.

Position and orientation Define the configuration of a vehicle in 3 DOF as

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{p}^n \\ \Theta \end{bmatrix} = \begin{bmatrix} N \\ E \\ \psi \end{bmatrix} \in \mathcal{M}_C, \tag{4.3}$$

where $\mathcal{M}_C$ is the configuration space (4.1). N and E is the North and East position of the vehicle in [m], while ψ [rad] is the orientation of the vehicle relative the {n}-frame. Unfortunately, $\boldsymbol{\eta}$ conflicts with the notation of the $\boldsymbol{\eta}$-*spline*, but since it is clear from the contex which vector is discussed, alternative notation will not be introduced.

Linear and angular velocity The body-fixed velocity is

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{v}^b \\ \omega^b \end{bmatrix} = \begin{bmatrix} u \\ v \\ r \end{bmatrix} \in \mathbb{R}^3, \tag{4.4}$$

where u and v [m/s] are time rate of change of the x-axis and y-axis of the body-frame relative a stationary reference frame. Furthermore, r [rad/s] is the angular speed of the vehicle.

Rotation matrices

To determine a position given in an arbitrarty reference frame {p} in another reference frame {q}, a transformation is needed. This transformation manipulates the original coordinates in such way that the same position now is represented in the secondary reference frame. To achieve this, a rotation between the reference frames is needed. A rotation matrix $\boldsymbol{R}$ is an element in the special orthogonal group of order 3 (Egeland & Gravdahl 2002):

$$\mathcal{SO}(3) = \left\{ \boldsymbol{R} | \boldsymbol{R} \in \mathbb{R}^{3\times 3},\, \boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I},\, \det \boldsymbol{R} = 1 \right\}. \tag{4.5}$$

Define $\boldsymbol{R}_p^q$ as the rotation matrix which transforms a position $\boldsymbol{\nu}^p$ *from* the {p}-frame *to* the {q}-frame:

$$\boldsymbol{\nu}^q = \boldsymbol{R}_p^q \boldsymbol{\nu}^p. \tag{4.6}$$

The principal Euler rotation matrix about the z-axis is:

$$\boldsymbol{R}_{z,\psi} \triangleq \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.7}$$

where ψ is the rotation angle. For instance, transforming a velocity vector $\boldsymbol{\nu}^b$ in {b} to {n} is performed by

$$\dot{\boldsymbol{\eta}} = \boldsymbol{\nu}^n = \boldsymbol{R}_{z,\psi}(\psi)\boldsymbol{\nu}^b. \tag{4.8}$$

As remarked in (Breivik 2010), the velocity vector of a vehicle is not necessarily pointed in the same direction as the x-axis of the body-frame. To successfully follow a path with an underactuated marine vehicle, it is not sufficient to only steer the heading of the vehicle; the orientation of the velocity vector must be controlled. We define the course angle as the orientation of the velocity vector, while the heading is the orientation of the vehicle. The difference between these orientations is called the sideslip angle β. We get the relationship (Fossen 2002)

$$\chi = \psi + \beta, \tag{4.9}$$

see Figure 4.3.

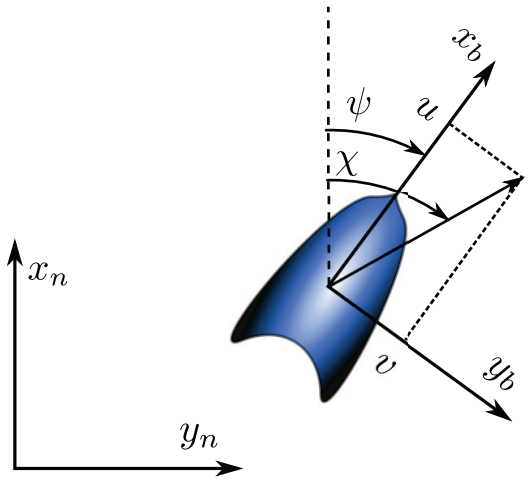


Figure 4.3: The heading ψ is the orientation of the {b}-frame relative the {n}-frame. χ is the orientation of the velocity vector of the vehicle relative the {n}-frame. Adapted from (Breivik 2010).

4.1.4 Characterization of marine surface vessels

According to (Faltinsen 2005), one often divide vessels into three main categories depending on which forces dominate the interaction on the hull at maximum speed. These categories are

Displacement vessels: Buoyancy forces dominate.

Semi-displacement vessels: Buoyancy forces no longer dominates over hydrodynamic forces.

Planing vessels: Hydrodynamic forces dominates.

For instance, planing vessels have different dominating forces depending on the actual speed of the vessel. Thus, if the intended tasks of a vessel are defined in such way that the vessel must operate in many different displacement regions, unified control of the vessel is difficult (Breivik 2010).

4.2 Vehicle characteristics

4.2.1 Vehicle dynamics

The task of following a predefined path introduces the challenge of controlling the velocity vector of the vehicle. Since the vehicle has dynamics, neither the magnitude, nor the orientation of the velocity vector can change abitrarily fast.

The magnitude of the velocity vector

For most vehicles, the acceleration of speed is dependent on the speed itself, hence a differential system is needed to predict the magnitude of the velocity vector.

Let the vector $\boldsymbol{p} \triangleq [x,\, y]^\top$ represent the planar position of a vehicle relative to the origin of a stationary reference frame. Further, let the magnitude of the velocity vector be defined as $U \triangleq \sqrt{\dot{x}^2 + \dot{y}^2} \geq 0$. Then denote $f(U)$ as the acceleration function. Strictly speaking, $f(U)$ depends on the actual

control input, but that is omitted here. The following system represents the simplified dynamics for the vehicle speed response in the general case:

$$\dot{U} = a \tag{4.10a}$$
$$a = f(U). \tag{4.10b}$$

Most ships have different characteristics when increasing or decreasing their speed. A ship can be made to accelerate by using its propellers, while hydrodynamic damping forces can make it decelerate. Hence, $f(U)$ is vehicle-dependent and changes depending on acceleration or deceleration.

Figure 4.4: A *Viknes 830* experiencing hydrodynamic damping. Courtesy of www.maritimerobotics.no.

Angular speed

Recall that χ is the angle between the x-axis of a stationary reference frame and the velocity vector of the vehicle. The velocity vector can then be written as

$$\frac{d\boldsymbol{p}}{dt} = U \begin{bmatrix} \cos(\chi) \\ \sin(\chi) \end{bmatrix}. \tag{4.11}$$

The angular speed of the velocity vector is given by

$$\dot{\chi} = \omega, \tag{4.12}$$

where the relationship between the speed and angular speed is

$$U = \omega R. \tag{4.13}$$

For a given speed, there is a maximum angular speed of a vehicle. The maximum angular speed can be estimated as a function of the vehicle speed: $\omega_{max}(U) > 0$. Although this function sometimes is assumed to be constant in the area of interest (Yoshimoto et al. 2000, Bibuli, Bruzzone & Caccia 2009), it is here assumed that it does in fact depend on the vehicle speed. The inequality

$$\dot{\chi} \leq \omega_{max}(U) \tag{4.14}$$

must be satisfied in order not to violate the angular speed constraint. Hence, there is an implicit relationship between the speed and the maximum angular speed. Thus, the following inequality must also be satisfied:

$$U \leq \frac{\omega_{max}(U)}{\kappa}, \tag{4.15}$$

where $U \in [U_{min}, U_{max}]$, and U_{min} and U_{max} is the smallest and largest vehicle speed. $\kappa = \frac{1}{R}$ (3.18) is the curvature.

Angular acceleration

A vehicle cannot obtain angular speeds instantaneously, there is a bounded angular acceleration that must be obeyed. For a car, this corresponds to a bounded turning rate on the wheels, while for a boat this can for instance correspond to bounded angular speed on the rudder angle. The maximum angular acceleration must obey

$$|\ddot{\chi}| \leq \dot{\omega}_{max}(U), \tag{4.16}$$

where $\dot{\omega}_{max}(U) > 0$ is a given constraint.

4.2.2 Vehicle model

The most common way of modelling marine vessels are the vectorial model representation of Fossen (Fossen 2002). This representation compactly describes each degree of freedom using matrices similar to those found in robot modelling. From (Fossen 2002) we have

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}, \tag{4.17}$$

where

$$\boldsymbol{\nu} = \begin{bmatrix} u, & v, & w, & p, & q, & r \end{bmatrix}^\top \tag{4.18}$$

$$\boldsymbol{\eta} = \begin{bmatrix} x, & y, & z, & \phi, & \theta, & \psi \end{bmatrix}^\top \tag{4.19}$$

are vectors of velocity and position/Euler angles, respectively. Furthermore, $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are inertia, Coriolis and damping matrices, respectively. $\boldsymbol{g}$ is a vector of gravitational/buoyancy forces and moments.

Unfortunately, this model is constructed to be valid for low speed regimes in the displacement region, and hence is not very accurate for higher speeds. Thus, constructing a control law which satisfies this model may not be fruitful when the model itself is inaccurate. The scope of this thesis is not construction of control laws, but rather desired state assignments constructed by a guidance system. Hence, only a closed-loop model of the vehicle behavior is presented.

The proposed model is a simplified version of a small vehicle. It simulates the closed-loop behavior of the course χ [rad] and speed U [m/s] when the reference signals χ_d and U_d are given.

Course model

Similar to the yaw-rate controller in (Breivik et al. 2008), define the course rate as:

$$\dot{\chi} = \dot{\chi}_{\max}(U) \frac{\tilde{\chi}}{\sqrt{\tilde{\chi}^2 + \Delta_{\dot{\chi}}^2}}, \tag{4.20}$$

where $\Delta_{\dot{\chi}}$ is a shaping variable which determines the rendezvous behavior of the course rate when the course error becomes small. A small rendezvous variable forces the course rate to faster attain the maximal course rate. This model acts as a bounded low-pass filter, so there are no limitations on the course acceleration. The angular speed is bounded by $\dot{\chi}_{\max}(U) \triangleq \omega_{\max}(U)$.

Define the course error as

$$\tilde{\chi} \triangleq \chi_d - \chi, \quad \in \langle -\pi, \pi]\,. \tag{4.21}$$

To avoid possible wrap-around problems, care must be taken when calculating $\tilde{\chi}$. Specifically, it cannot be calculated simply by taking the difference

between the desired and actual course. Motivated by (Breivik et al. 2008), employment of the four quadrant version of arctan yields the desired behavior:

$$\tilde{\chi} \;=\; \text{atan2}(\sin(\tilde{\chi}), \cos(\tilde{\chi})), \tag{4.22}$$

where $\sin(\tilde{\chi})$ and $\cos(\tilde{\chi})$ can be calculated by

$$\sin(\tilde{\chi}) = \sin(\chi_d - \chi) \;=\; \sin\chi_d \cos\chi - \cos\chi_d \sin\chi, \tag{4.23a}$$
$$\cos(\tilde{\chi}) = \cos(\chi_d - \chi) \;=\; \cos\chi_d \cos\chi + \sin\chi_d \sin\chi. \tag{4.23b}$$

Forward speed model

Small seaborne vehicles, such as USVs, start to plane at intermediate speeds. In the semi-displacement/planing regions, the characteristics of the acceleration is different than for the low speed displacement region. In (Breivik et al. 2008) the step response of the surge speed revealed a rapid increase of speed up to about 5 [m/s] and a more linear increase after that. Motivated by this result, the response for low speeds is nonlinear, while the model has constant acceleration for higher speeds. Furthermore, acceleration is slower than deceleration, hence distinguishing between acceleration and deceleration is important.

Define the speed error as

$$\tilde{U} \;\triangleq\; U_d - U, \tag{4.24}$$

such that when $\tilde{U} \geq 0$, an acceleration finds place:

$$\dot{U} \;=\; \dot{U}_{\text{max,acc,nl}}(U)\frac{\tilde{U}}{\sqrt{\tilde{U}^2 + \Delta^2_{\text{acc,nl}}}} + \dot{U}_{\text{max,l}}\frac{\tilde{U}}{\sqrt{\tilde{U}^2 + \Delta^2_{\text{l}}}}, \tag{4.25}$$

where $\dot{U}_{\text{max,l}}$ is the maximal linear acceleration, and the maximal nonlinear acceleration vanishes exponentially as

$$\dot{U}_{\text{max,acc,nl}}(U) = Ke^{-a_{\text{acc}}U}, \tag{4.26}$$

where $K > 0$ and $a_{\text{acc}} > 0$. $\Delta_{\text{acc,nl}} > 0$ and $\Delta_{\text{l}} > 0$ are shaping parameters which determine what $\tilde{U}$ must be to give maximal nonlinear acceleration and linear acceleration, respectively.

When $\tilde{U} < 0$, a deceleration finds place. The decelerating response is modelled by

$$\dot{U} = \dot{U}_{\max,\mathrm{dec,nl}}(U)\frac{\tilde{U}}{\sqrt{\tilde{U}^2 + \Delta^2_{\mathrm{dec,nl}}}} + \dot{U}_{\max,\mathrm{l}}\frac{\tilde{U}}{\sqrt{\tilde{U}^2 + \Delta^2_{\mathrm{l}}}}, \tag{4.27}$$

where the nonlinear term dominates at higher speed according to

$$\dot{U}_{\max,\mathrm{dec,nl}}(U) = a_{\mathrm{dec}}U, \tag{4.28}$$

where $a_{\mathrm{dec}} > 0$. Again, $\Delta_{\mathrm{dec,nl}} > 0$ and $\Delta_{\mathrm{l}} > 0$ are shaping parameters, where the latter is identical with the one defined for linear acceleration.

Polar coordinate representation

The two closed-loop models just presented is a decomposition of the velocity vector. This decomposition, which has been proposed in (Breivik 2010) and references therein, is motivated by control laws that are able to control the surge speed and the yaw rate. When the desired forward speed and course are known, meaningful feedback action is possible from the desired values. Figure 4.5 illustrates the decomposition of the velocity vectors.

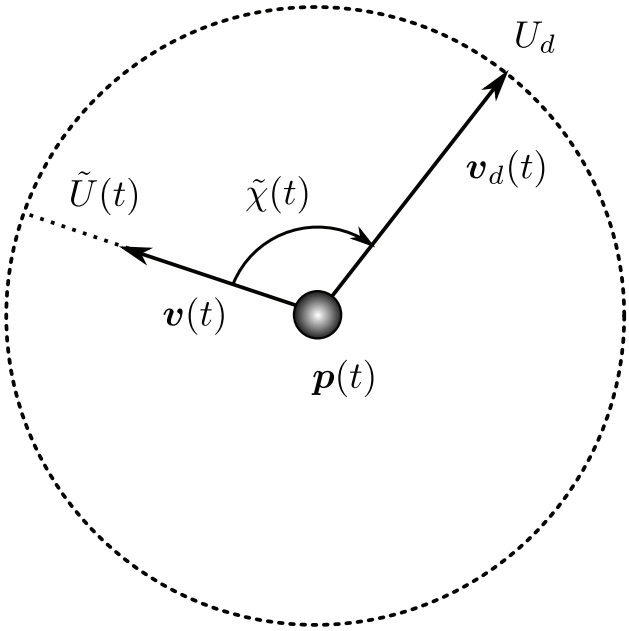


Figure 4.5: The decomposition of the velocity vectors into magnitudes and orientations. Adopted from (Breivik 2010).

Chapter 5

Motion control for path scenarios

There are many different ways of constructing a vehicle motion control system. As a whole, this system is complex and consists of many different subproblems. A convenient way of structuring these problems is the principle of modularity. Each module solves its own problems, and by combining all the modules, a fully functional motion control system is made possible. This makes it easier to modify a specific part of the system without redesigning the whole system. Furthermore, considering less comprehensive problems individually are often easier than trying to solve the whole problem at once.

One possible way of dividing the motion control problem is shown in Figure 5.1. Each block takes care of their own problems, and output results when the problems are solved. By changing the guidance system, different tasks can be achieved with the same measurement and control system.

5.1 The guidance system

A guidance system is responsible for prescribing the commands needed to achieve a desired motion in the physical environment in which a vehicle moves. Typically, for underactuated vehicles the orientation and magnitude of the velocity vector need to be determined. In general, a path maneuvering problem can be divided into two objectives:

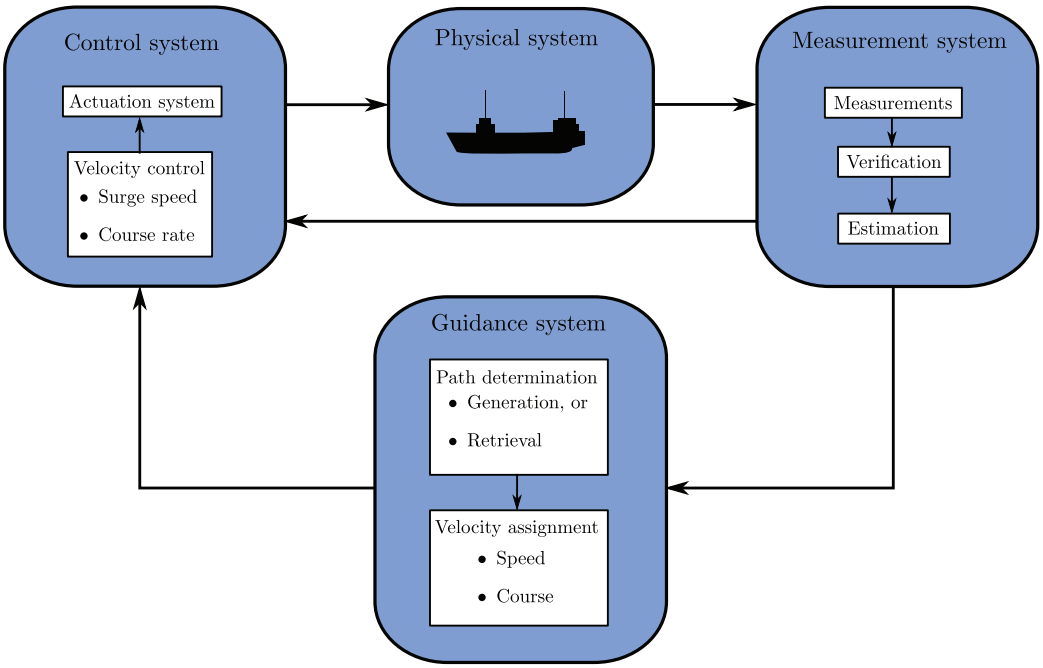


Figure 5.1: Dividing a vehicle control system into subproblems.

Path Traversion: Assume that the vehicle is confined to move along a path, that is, the velocity vector always points along the tangent of the path, like trains. The vehicle has maneuverability constraints which must be obeyed in order to stay on the path. In this case only appropriate speed assignments are needed to fulfill the objective, constituting a speed problem where U must be decided.

Path Convergence: The vehicle is not on the path and the orientation of the velocity vector must be determined for the vehicle to converge to the path. This is a steering problem where the appropriate orientation of the velocity vector χ must be determined.

Figure 5.2 shows the idea of path traversion and path convergence. By combining the two objectives, the total solution gives a complete guidance system applicable to path maneuvering of underactuated vehicles. In accordance with the path maneuvering definition, this guidance system does not only consider the geometric aspects of a motion control scenario, but also the vehicle maneuverability constraints.

In the next sections, methods for both these objectives are presented. In Section 5.2 the path traversion problem is considered. The main challenge

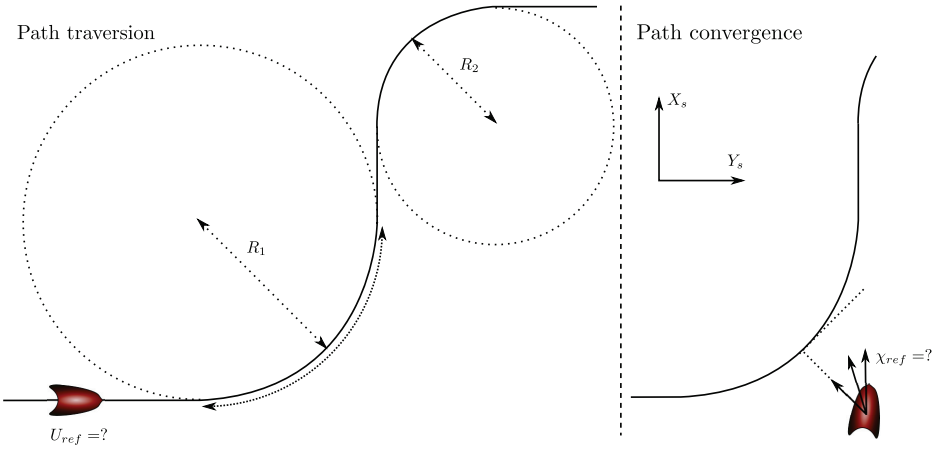


Figure 5.2: The concepts of path traversion and path convergence.

explored in this section is to determine the largest speed for which the vehicle can have at a path point, and satisfy both current and future maneuverability constraints imposed by the path curvature. Next, in Sections 5.3 to 5.6, algorithms for path convergence are presented. The main obstacle in these sections is to create guidance algorithms that try avoid too aggressive convergence, or somehow optimalize the convergence. In Section 5.7 the path maneuvering algorithm is combined with a path-tracking method, resulting in a scheme for tracking a target on a path which is only defined a limited distance ahead.

5.2 Path speed algorithm

5.2.1 Motivation

An important quality for many marine vehicles is the ability to follow a given geometrical path accurately. This path can either be fully defined beforehand or given such that only a limited distance ahead of the vehicle is known. Since vehicles have maneuverability constraints, these limitations cannot be ignored when following a path. Hence, the speed assignments must be determined with care, employing information about the path. The topic of this chapter is development of such a speed control algorithm.

5.2.2 Previous work

Previous work on the topic of speed control for path-maneuvering purposes include (Yoshimoto et al. 2000), where a speed control algorithm is employed. The concept is developed for ground vehicles where the path is discovered as the vehicle moves forward. The speed is adjusted such that the lateral acceleration does not exceed a permissible range and the curvature is used in the determination of the commanded speed. The principles are based on simple, yet beneficial physical considerations and are proven to work in experimental trials with a model car. Nonetheless, there are some limitations worth mentioning. First, the lateral acceleration limit is chosen to be constant for all speeds, which yields a suboptimal solution. Second, the commanded speed is chosen to comply with the curvature at a given lookahead distance. The lookahead distance is chosen to be sufficient for the whole speed range, so in some cases the lookahead distance may be too conservative.

In (Fox et al. 1997), the motion dynamics of a mobile robot is used to determine admissible translational and rotational velocities. The paper deals with constraints imposed by limited velocities and accelerations. Since it covers an approach to collision avoidance, only velocites considered safe are chosen. Although the task is not to follow a predefined path, interesting concepts are presented, and in particular the dynamic window approach; "The dynamic window restricts the admissible velocites to those that can be reached within a short time interval given the limited accelerations of the robot" (Fox et al. 1997). This scheme results in a set of reachable velocites. A speed algorithm which yields feasible command signals may result in a smooth and desirable system response.

Recently, planar path-following algorithms employing heuristic speed adaptation has been proposed (Bibuli, Bruzzone & Caccia 2009). The advance speed is based on path curvature measurements and steering action predictions. Specifically, when the orientation is far from the desired value, the speed is reduced to render faster convergence. The maximum curvature is measured inside a prediction horizon ahead of the vehicle, and the speed is determined such that it never exceeds the constraint imposed by the maximum curvature within the prediction horizon. However, this approach is more conservative than a scheme that explicitly employs maneuverability constraints in the speed decision.

Figure 5.3: A Ferrari 458 Italia moving on a curved track. Courtesy of www.collegecars.wordpress.com.

5.2.3 Along-path lookahead distance

The path parameterization provides information that can be used to plan the vehicle speed ahead of its current location. Specifically, information about the curvature $\kappa(\varpi)$ is particularly interesting. When the curvature is known at a given path point, it is possible to calculate the highest speed that the vehicle can have and still remain on the path given the curvature-associated restrictions on its angular speed.

The full path information is not required, only a predefined along-path lookahead distance is necessary. If the lookahead distance is large enough, the path information provided by this interval is sufficient to yield feasible speeds in order to fulfill future curvature constraints. A reasonable along-path lookahead distance can be chosen as the stopping distance from maximum speed, denoted Δ_d. We now proceed to develop the methods to be able to retrieve the path information needed.

The lookahead system

The combination of (3.16) and (3.17) yields

$$\dot{s}_c = U_p(t) \tag{5.1a}$$

$$\dot{\varpi}_c(t) = \frac{\dot{s}_c}{|\boldsymbol{p}_p'(\varpi_c)|}, \tag{5.1b}$$

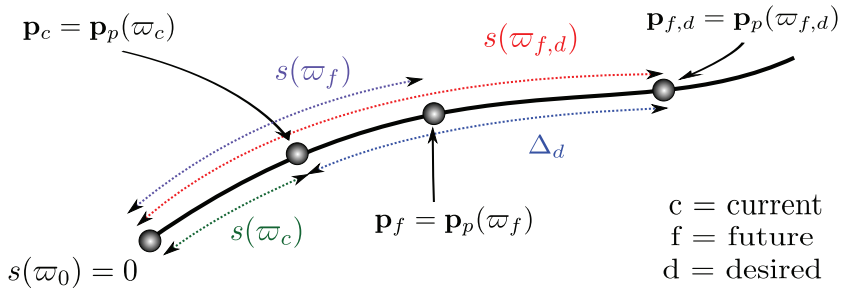


Figure 5.4: Along-path lookahead distance Δ_d between the current position and a desired future position. Here, $s(\varpi)$ is the arc length while $\boldsymbol{p}_p(\varpi)$ is a point on the path.

which gives information about the vehicle's *current* position, using the subscript c to indicate that the current position is considered. A similar scout system can be used to obtain information about what is ahead. We call the scout system state the *future*, with subscript f to indicate that this is something the vehicle will experience in the future. The lookahead distance is then the arc length between the current and future path point, see Figure 5.4. For the sake of simplicity, we let the current and future arc length $s(\varpi_c)$ and $s(\varpi_f)$ be denoted s_c and s_f, respectively. As can be seen from Figure 5.4, the desired value for the future arc length is

$$s_{f,d} \;=\; s_c + \Delta_d. \tag{5.2}$$

A feedback driving the scout system to the desired future path point is needed. The difference between the actual and desired future arc length is denoted

$$\tilde{s} \triangleq s_f - s_{f,d}, \tag{5.3}$$

and the control objective of the scout system becomes

$$\lim_{t\to\infty} \tilde{s}(t) = \;0, \tag{5.4}$$

achievable by controlling the along-path speed of the scout system. Specifically, by introducing a simple P-controller for the scout system, the objective is achieved:

$$\dot{s}_f \;=\; U_p(t) - k_p \cdot \tilde{s}(t), \tag{5.5}$$

$$\dot{\varpi}_f(t) \;=\; \frac{\dot{s}_f}{|\boldsymbol{p}_p'(\varpi_f)|}, \tag{5.6}$$

where $k_p > 0$ is the proportional gain. Fortunately, the scout system is not a physical system and hence the convergence can be arbitrarily fast.

Lyapunov analysis of the lookahead system

Introduce the radially unbounded control Lyapunov function (CLF) and its time derivative

$$V(\tilde{s}) = \frac{1}{2}\tilde{s}^2 > 0 \,\forall\, \tilde{s} \neq 0 \tag{5.7}$$

$$\dot{V}(\tilde{s}) = \dot{\tilde{s}}\tilde{s}, \tag{5.8}$$

where the time derivative of $\tilde{s}$ is

$$\dot{\tilde{s}} = \dot{s}_f - \dot{s}_{f,d} \tag{5.9}$$

$$\Downarrow \quad (5.1a), (5.2), (5.5), \dot{\Delta}_d = 0$$

$$\dot{\tilde{s}} = -k_p\tilde{s}. \tag{5.10}$$

Combining (5.8) and (5.10) gives

$$\dot{V} = -k_p\tilde{s}^2 < 0 \,\forall\, \tilde{s} \neq 0, \tag{5.11}$$

which renders the lookahead error globally exponentially stable (GES), see e.g. (Khalil 2002).

5.2.4 Speed assignment algorithm

The lookahead system provides information about the path ahead of the current position. In order to utilize this information, it needs to be stored, updated and retrieved at appropriate times. A database containing information about the path segment, path parameter, arc length and curvature can be used to assess appropriate speed limitations at different path points. These speed limitations are stored in the database and updated as new information is received.

The database is outlined in Table 5.1. The three first columns of the database represent information received from the lookahead system (5.1)-(5.6). The contents of the remaining two columns will be explained in the subsequent sections. Before elaborating the contents of these columns, it is worth pointing out that the database is constructed for a curve represented by a single

path parameterization. In the case of piecewise path parameterizations, the lookahead particle does not necessarily reside on the same curve segment as the vehicle particle. Thus, modifications to the database must be applied to make it compatible with piecewise curves.

$\boldsymbol{\varpi}$	$\boldsymbol{s(\varpi)}$	$\boldsymbol{\kappa(\varpi)}$	$\boldsymbol{U_c}$	$\boldsymbol{U_f}$
ϖ_1	$s(\varpi_1)$	$\kappa(\varpi_1)$	$U_c(\varpi_1)$	$U_f(\varpi_1 \vert \varpi_f)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
ϖ_f	$s(\varpi_f)$	$\kappa(\varpi_f)$	$U_c(\varpi_f)$	$U_f(\varpi_f \vert \varpi_f)$

Table 5.1: Database of relevant path information.

The database in Table 5.2 is a modified version of Table 5.1 which is applicable for piecewise curves. The notation is this database is as follows: $\varpi_{k,\ell}$ indicates that this parameter value belongs to the path parameterization defined by $\mathcal{P}_k$, and that it is the ℓ-th entry of the database. In the following, the explanation of the two last columns will be done by referring to Table 5.1, with the understanding that these considerations also applies for Table 5.2.

$\mathcal{P}_k$	$\boldsymbol{\varpi_k}$	$\boldsymbol{s(\varpi_k)}$	$\boldsymbol{\kappa(\varpi_k)}$	$\boldsymbol{U_c}$	$\boldsymbol{U_f}$
$\mathcal{P}_i$	$\varpi_{i,1}$	$s(\varpi_{i,1})$	$\kappa(\varpi_{i,1})$	$U_c(\varpi_{i,1})$	$U_f(\varpi_{i,1} \vert \varpi_{j,f})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathcal{P}_j$	$\varpi_{j,f}$	$s(\varpi_{j,f})$	$\kappa(\varpi_{j,f})$	$U_c(\varpi_{j,f})$	$U_f(\varpi_{j,f} \vert \varpi_{j,f})$

Table 5.2: Database of relevant path information, extended to piecewise curves.

Current curvature-constrained speed restriction

When the curvature on a path point is known, it is possible to calculate the maximum speed at which the vehicle is able to follow the curvature at this point. This speed restriction is denoted U_c. Recall from (4.13) that $U = \frac{\omega}{\kappa}$, where the curvature $\kappa(\varpi_c)$ is calculated with (3.20).

The maximum angular speed of a vehicle can be estimated as a function of vehicle speed: $\omega_{max}(U) > 0$. Hence, there is an implicit relationship between

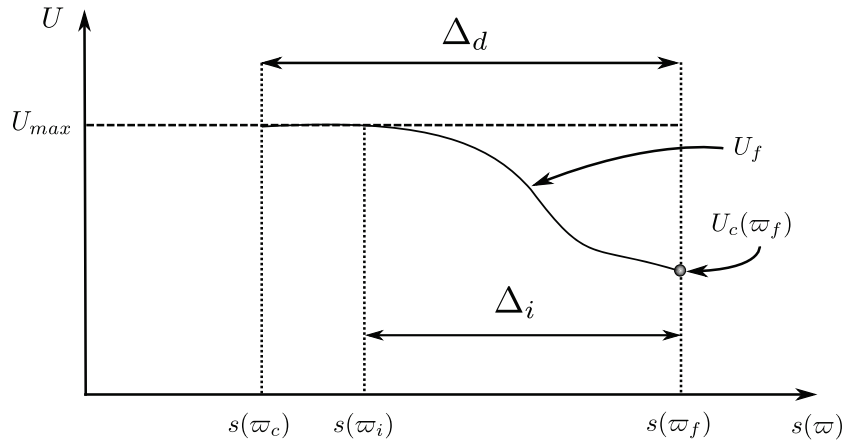


Figure 5.5: The maximum speed as a function of the arc-length distance, when a critical curvature is discovered at ϖ_f. U_f is updated in the interval Δ_i, between $s(\varpi_i)$ and $s(\varpi_f)$.

the maximum speed and the maximum angular speed. Thus, the following objective function must be solved:

$$\min_{U_c} \quad \left| U_c - \frac{\omega_{max}(U_c)}{\kappa(\omega_c)} \right| \tag{5.12a}$$

$$s.t. \quad 0 < U_{min} \leq U_c \leq U_{max}. \tag{5.12b}$$

In general, it is difficult to draw any conclusions from this optimization problem, but it should be noted that an explicit solution can be obtained in most practical cases. An explicit solution is preferred due to the computational demand of solving the objective function.

Future curvature-constrained speed restriction

The whole point of having a lookahead system is to utilize information about future curvature information. Since the vehicle cannot change speed arbitrarily fast, the speed restrictions close to a sharp turn should decrease in an ordinary fashion. Specifically, the system (4.10), with $f(U)$ equal to the decelerating response of the vehicle should be used to figure out when the vehicle should start to decelerate in time to satisfy (5.12b) and thus stay on the path.

The rightmost column of Table 5.1 contains the preliminary values for U_f. The notation $U_f(\varpi_i|\varpi_f)$ indicates that $U_f(\varpi_i|\varpi_f)$ is estimated using the

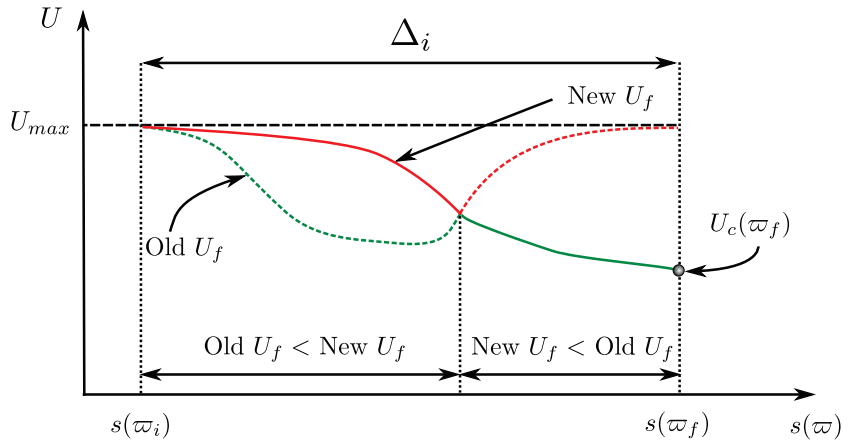


Figure 5.6: When U_f is updated, the smallest of the new and old U_f is chosen.

path curvature information provided by the scout system all the way up to ϖ_f. As long as the arc length between these two path points is smaller than the along-path lookahead distance Δ_d, the speed restriction at ϖ_i is not fully determined. In other words, only $U_f(\varpi_i|\varpi_f)$ where $s(\varpi_f) - s(\varpi_i) \geq \Delta_d$ constitutes a valid U_f.

The database contains so much information about the path that going from maximum speed to a full stop is possible within the lookahead distance. Nonetheless, for critical speed restrictions greater than zero at the lookahead distance Δ_d an update of the whole U_f-column is unnecessary. The database is therefore only updated a given distance interval $\Delta_i > 0$ from which the speed decelerates from U_{max} to the critical speed at the newly discovered path point $U_c(\varpi_f)$, see Figure 5.5. Thus, U_f is only updated for $\varpi \in \langle \varpi_i, \varpi_f \rangle$. Another thing to be kept in mind is that the update interval may contain several critical curvatures, meaning that the new U_f at a given point may be larger than an already existing speed restriction. Hence, a check between the new and old U_f must be performed, choosing the smaller one. Figure 5.6 shows the problem at hand. In the first part of the update interval, the old speed restriction is smaller than the new one, so no update is performed. When the new U_f becomes smaller, it is chosen in preference to the old U_f.

Admissible speeds during short time intervals

The database now contains two speed restrictions that should not be violated in order to keep the vehicle on the path. However, since the vehicle has dynamics, the speed cannot change arbitrarily fast. Thus, upper and lower admissible speeds should be determined according to acceleration limitations at each time step. By using the Bogacki-Shampine method for numerical solution for ordinary differential equations (Bogacki & Shampine 1989), estimates of the admissible speeds for the next time step are possible. This method is applied with following computations:

$$
\begin{aligned}
h &= t_{i+1} - t_i && \text{(5.13a)}\\
k_1 &= f\left(U(t_i)\right) && \text{(5.13b)}\\
k_2 &= f\left(U(t_i) + \frac{1}{2}hk_1\right) && \text{(5.13c)}\\
k_3 &= f\left(U(t_i) + \frac{3}{4}hk_2\right) && \text{(5.13d)}\\
U(t_i+1) &= U(t_i) + \frac{2}{9}hk_1 + \frac{1}{3}hk_2 + \frac{4}{9}hk_3. && \text{(5.13e)}
\end{aligned}
$$

Let $f_{acc}(U_c(t_i))$ and $f_{dec}(U_c(t_i))$ represent the maximum acceleration and deceleration at the speed $U_c(t_i)$, respectively. By substituting $f(U)$ with $f_{acc}(U_c)$ and $f_{dec}(U_c)$ in (5.13), estimates of upper and lower admissible speeds can be calculated. Denote $U_{acc}(t_{i+1})$ as the upper limit, while the lower limit is denoted $U_{dec}(t_{i+1})$.

Even though a vehicle has a more complex acceleration response than presented here, the interval $[U_{dec}, U_{acc}]$ represents more conservative speed changes than U_c and U_f generally can do, and should thus be included in the final decision of the reference speed.

Dynamic information window

As the vehicle moves along the path, the arc-lengths $s(\varpi_c)$ and $s(\varpi_f)$ increase, which means that new data is retrieved. Naturally, at some point data gets old and hence irrelevant for the future speed decisions. The threshold parameter is $s(\varpi_c)$, since database entries with $s(\varpi) < s(\varpi_c)$ contain information about path points that already have been traversed. Discarding useless database entries is performed continuously, always letting the upper-

most entry of the database be the best choice when determining the reference speed.

Decision of the reference speed

After calculating U_c, U_{acc}, U_{dec} and U_f, it is now possible to decide the reference speed for the next time step. Picking U_c and U_f from the uppermost entry of Table 5.1, one must decide the highest speed possible not violating the restrictions.

At a given path point it is not allowed to have a greater speed than U_c, due to the curvature constraint at this point. Hence, this is the highest possible speed no matter what, or else it is not possible to stay on the path. Furthermore, U_f indicates the highest speed at the same point in order to fulfill the curvature restrictions further ahead. For this reason, the chosen speed must be equal or lower than U_f in order to not derail further ahead. We get the following minimization:

$$U_{ref} = \min(U_c, U_f). \tag{5.14}$$

Given an initial speed U, the vehicle must accelerate or decelerate to arrive at the reference speed. If, for example, U_c and U_f both are quite far from the initial speed, the reference speed given by (5.14) is not admissible. We want to generate reference speeds that are admissible, so we have to incorporate U_{dec} and U_{acc} in the minimizing expression. For whatever reason, the result of the minimizing expression should always be within $[U_{dec}, U_{acc}]$, thus not violating the acceleration limitations. To successfully satisfy these conditions, the following minimizing expressions must be applied to ensure appropriate behavior for the algorithm:

Algorithm 1 U_{ref}

if $U_c \geq U$ **then**
  $U_{ref} \leftarrow \min(\max(U_{dec}, U_f), \min(U_{acc}, U_c))$
else if $U_c < U$ **then**
  $U_{ref} \leftarrow \min(\max(U_{dec}, U_c), \max(U_{dec}, U_f))$
end if

It should be noted that when $U_c < U$ the reference speed does not necessarily yield a feasible speed obeying the angular speed constraint. That is, when $U_{ref} > U_c$ it is not possible to follow the path. This situation can occur when

the deceleration function is too optimistic, or during bad initial conditions on U.

5.3 Path convergence

5.3.1 Motivation

The task of following a predefined path has many application areas. Transit, surveillance and systematic information gathering are all specific tasks where realization of path following algorithms are of interest (Breivik 2010). A measure of how well these tasks are executed depends on the eye of the beholder, but one plausible measure of performance is the rate of convergence. In other words, if there is an error from the desired state, how can the motion control system feasible converge to the desired states in an beautiful manner, without overshoot.

As discussed in Chapter 5, a guidance system, which is responsible for prescribing desired velocities, can affect the rate of convergence. Specifically, if the assignments are chosen with the vehicle maneuverability constraints taken into account, desirable behavior may be achieved. If an already established guidance law is used as a starting point, manipulation of the law's parameters alters the convergence behavior of the closed-loop system. If these manipulations are done purposefully, a better convergence is made possible.

In the next section, previous contributions on the topic of path maneuvering are presented.

5.3.2 Previous work

Since the turn of century, there have been several publications on path following for underactuated vehicles. In (Pettersen & Lefeber 2001, Fossen et al. 2003, Fredriksen & Pettersen 2006) the line-of-sight (LOS) approach have been investigated for straight-line path following. This approach mimics the behavior of a helmsman aiming at a moving point on the straight line, see Figure 5.7.

In (Breivik & Fossen 2009) this approach has been further upgraded, where guidance laws allow the vehicle to follow circles and even curved paths. A

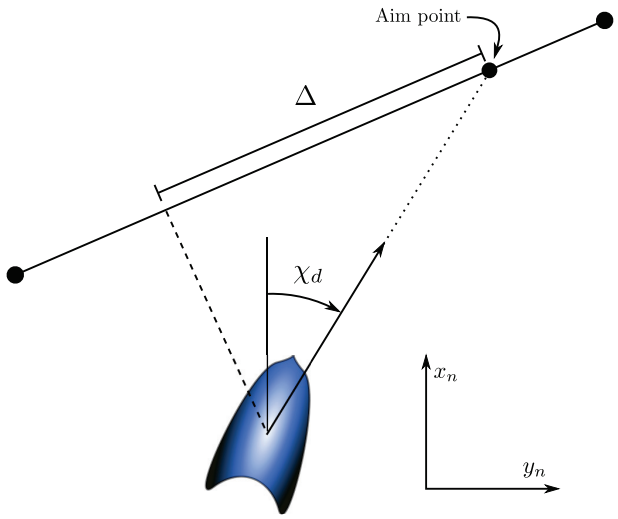


Figure 5.7: The principle of line-of-sight steering. The desired course is directed toward a moving point on the desired line, which is determined by the lookahead distance Δ.

drawback of these contributions is the performance of the methods. That is, for a constant speed, a constant lookahead may yield acceptable behavior, but if the forward speed changes, a different lookahead must be used. Furthermore, when the cross-track error is large, a more aggressive lookahead is desirable, while for intermediate cross-track errors, a conservative lookahead is appreciated, due to the vehicle maneuverability constraints.

To accommodate the vehicle maneuverability constraints, several authors have addressed this problem by employing methods which take the constraints into account. An optimal guidance scheme for cross-track control of underactuated underwater vehicles is considered in (Børhaug et al. 2006). The authors propose a model predictive guidance (MPG) scheme to find optimal commanded signals on the course rate. They compare the scheme with conventional LOS-guidance with constant lookahead to illustrate the improved convergence rate. Later, in (Pavlov et al. 2009, Oh & Sun 2010) a model predictive control (MPC) algorithm is used on a straight-line path following algorithm to update the lookahead distance in such way that a fast convergence with minimal overshoot is achieved. For curved paths, the curvature of the path influences the convergence of the vehicle. Thus, the path information has be exploited in (Subbotin et al. 2006, Gomes et al. 2007) to yield faster convergence to the path.

In the next section a steering law for regularly parameterized paths is pre-

Figure 5.8: Statoil's process plant in Kårstø, Norway. MPC is a design method which is widely used in the process industry. The dynamics of vehicles are relatively fast compared to process plant dynamics. Due to better opportunities for increased computational speed, MPC has also found its use in other control problems. In the last decade, several articles on MPC-based vehicle motion control have emerged. Courtesy of Anders J. Steensen / Teknisk Ukeblad, www.tu.no.

sented. This steering law's lookahead distance is manipulated in the subsequent chapters to achieve faster convergence.

Steering for regularly parameterized paths

Consider a particle with position $\boldsymbol{p}(t)$. This particle is not situated on the path. We want the particle to converge to the path and move along the path. To achieve this objective, the velocity vector of the particle must be decided. The material in the next paragraphs is taken from (Breivik & Fossen 2009). The orientation of the velocity vector is given by the steering law

$$\chi(e) \quad = \quad \chi_p(\varpi) + \chi_r(e), \tag{5.15}$$

where χ_p is the path-tangential angle given by

$$\chi_p(\varpi) \quad = \quad \text{atan2}(y_p'(\varpi), x_p'(\varpi)), \tag{5.16}$$

where atan2(y, x) is the four-quadrant version of arctan, that is, atan2$(y, x) \in \langle -\pi, \pi]$. The second term $\chi_r(e)$ is the path-relative steering angle:

$$\chi_r(t) \triangleq \arctan\left(\frac{-e(t)}{\Delta}\right) \in \langle -\pi/2,\, \pi/2\rangle\,. \tag{5.17}$$

Furthermore, the deviation from the path is divided into two variables $\sigma(t)$ and e(t):

$$\begin{bmatrix} \sigma(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} \cos\chi_p & -\sin\chi_p \\ \sin\chi_p & \cos\chi_p \end{bmatrix}^T (\boldsymbol{p}(t) - \boldsymbol{p}_p(\varpi)). \tag{5.18}$$

$\sigma(t)$ is called the *along-track distance*, while e(t) is the *cross-track error*.

The path parameter is given by

$$\dot{\varpi} = \frac{U(t)\cos\chi_r(e) + \gamma\sigma(t)}{|\boldsymbol{p}_p'(\varpi)|}, \tag{5.19}$$

where $\gamma > 0$. The control objective is

$$\lim_{t\to\infty} \begin{bmatrix} \sigma(t) \\ e(t) \end{bmatrix} = \mathbf{0}. \tag{5.20}$$

The idea of the algorithm is to point the velocity vector toward a point on the tangent of the direct projection of $\boldsymbol{p}(t)$ onto the path. To fulfill the objective, $\boldsymbol{p}_p(\varpi)$ can collaborate with $\boldsymbol{p}(t)$ by reducing $\sigma(t)$. This is achieved with (5.19). Figure 5.9 illustrates the principle of the steering algorithm. The path-fixed reference frame {p} has been rotated a positive angle χ_p relative to the stationary reference frame {s}. The x-axis of {p} now points along the tangent of the path at the path point $\boldsymbol{p}_p(\varpi)$. For a more thorough explanation, the reader is encouraged to consult (Breivik & Fossen 2009).

Extensions due to piecewise paths The steering law just presented is constructed for continuous paths, which is parameterized with a continuously increasing parameter. To make the law applicable for piecewise paths, where the parameter variable not necessarily is strictly increasing, care must be taken when transferring from one parameterization to the next. By re-parameterizing each curve segment, such that the parameter variable be-

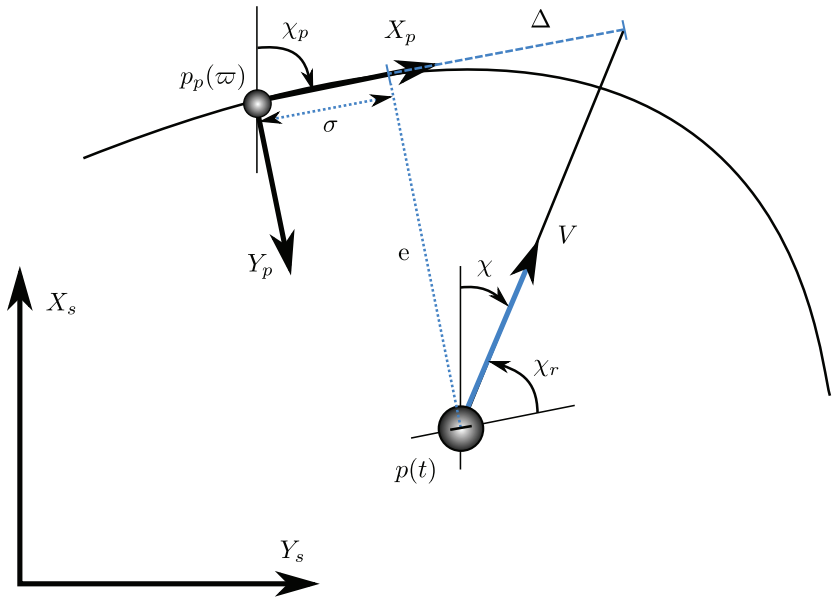


Figure 5.9: The principle of steering for regularly parameterized paths. Adapted from (Breivik & Fossen 2009).

comes continuously increasing is one possible solution:

$$\mathcal{P}_k \quad : \quad \varpi \in [\varpi_{k,0},\, \varpi_{k,1}] \tag{5.21}$$
$$\mathcal{P}_{k+1} \quad : \quad \varpi \in [\varpi_{k,1},\, \varpi_{k+1,1}] \tag{5.22}$$
$$\mathcal{P}_{k+2} \quad : \quad \varpi \in [\varpi_{k+1,1},\, \varpi_{k+2,1}] \tag{5.23}$$
$$\vdots \tag{5.24}$$

Another solution is to let the parameter variable be untouched, and solve the problem by other means. Specifically, when the numerical integral (5.19) exceeds the upper limit of the currently active curve segment $\mathcal{P}_k$, it must be reset to the corresponding parameter value $\varpi_{k+1,t+1}$ of the next curve segment $\mathcal{P}_{k+1}$. Figure 5.10 displays the relevant quantities needed to solve the problem. To find this parameter value, some calculations must be done. First, the arc-length distance between the previous time step's parameter value $\varpi_{k,t}$ and the currently desired parameter value $\varpi_{k,t+1}$ must be calculated. From (3.11) we have

$$s = \int_{\varpi_{k,t}}^{\varpi_{k,t+1}} \left| \boldsymbol{p}_p'(u; \mathcal{P}_k) \right| du. \tag{5.25}$$

Next, the arc-length from $\varpi_{k,t}$ to $\varpi_{k,1}$ is found:

$$s_1 = \int_{\varpi_{k,t}}^{\varpi_{k,1}} \left| \boldsymbol{p}_p'(u; \mathcal{P}_k) \right| du. \tag{5.26}$$

Now, the arc-length to be travelled on the new curve segment is known as

$$s_2 = s - s_1. \tag{5.27}$$

Hence, the desired parameter value can be found by solving the arc-length equation for the new curve segment with respect to the upper limit of the integral

$$s_2 = \int_{\varpi_{k+1,0}}^{\varpi_{k+1,t+1}} \left| \boldsymbol{p}_p'(u; \mathcal{P}_{k+1}) \right| du, \tag{5.28}$$

which unfortunately cannot be solved explicitly in the general case. Thus, a numerical method must be employed. For instance, Newton's method can be used. See Appendix C.4.

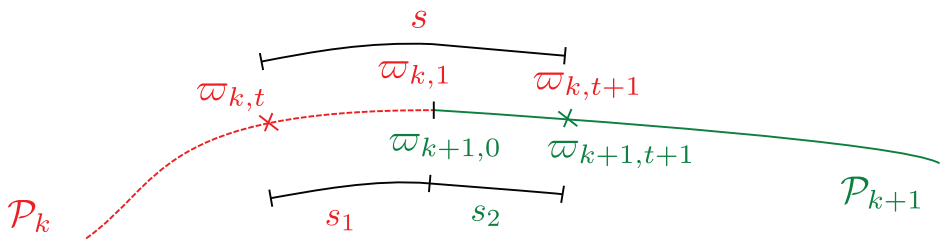


Figure 5.10: Quantities involved when transferring from one curve segment to the next.

5.4 Path convergence for straight lines

If the steering law presented in Section 5.3.2 is employed, it is possible to investigate the course rate during the convergence phase. The time derivative of the course angle (5.15) is

$$\dot{\chi}(e) = \dot{\chi}_p(\varpi) + \dot{\chi}_r(e), \tag{5.29}$$

and since the path-tangential angle χ_p is constant for a straight line, we get

$$\dot{\chi}(e) = \dot{\chi}_r(e) \tag{5.30}$$

$$\dot{\chi}_r(e) = \frac{d}{dt}\left(\arctan\left(\frac{-e(t)}{\Delta}\right)\right) \tag{5.31}$$

$$= \frac{e\dot{\Delta} - \dot{e}\Delta}{\Delta^2 + e^2}. \tag{5.32}$$

By also using Pythagoras' theorem on quantities in Figure 5.11, $\dot{e}$ can be rewritten to

$$\dot{e} = U \sin \chi_r = \frac{-Ue}{\sqrt{\Delta^2 + e^2}}, \tag{5.33}$$

which yields the following expression for the course rate

$$\dot{\chi}_r(e) = \frac{e\dot{\Delta} - \frac{-Ue\Delta}{\sqrt{\Delta^2+e^2}}}{\Delta^2 + e^2}. \tag{5.34}$$

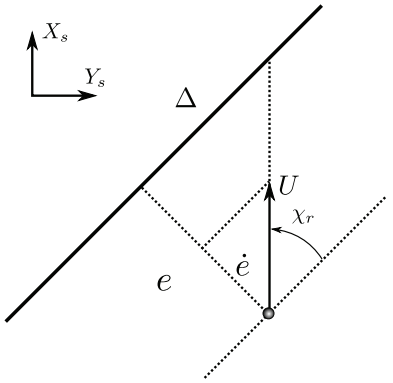


Figure 5.11: Geometric relationship between different quantities.

The course rate is the same as the angular speed of the vehicle, and hence we get the following constraint from (4.14) when steering the vehicle

$$|\dot{\chi}_r(e)| \leq \omega_{max}(U). \tag{5.35}$$

This motivates a choice of Δ in such way that this constraint is never violated.

5.4.1 Semi-constant lookahead

First, let us consider the case where the time derivative of the lookahead distance is constant for constant speed. Equation (5.34) reduces to

$$\dot{\chi}_r(e) = \frac{Ue\Delta}{(\Delta^2 + e^2)^{\frac{3}{2}}}. \tag{5.36}$$

Differentiate (5.36) with respect to the cross-track error:

$$\frac{d\dot{\chi}_r}{de} = \frac{U\Delta}{(\Delta^2 + e^2)^{\frac{5}{2}}}(\Delta^2 - 2e^2). \tag{5.37}$$

Next, set this derivative to zero, and find the corresponding cross-track error:

$$\frac{d\dot{\chi}_r}{de} \stackrel{!}{=} 0 \rightarrow \Delta^2 - 2e^2 = 0 \rightarrow e = \pm\frac{\Delta}{\sqrt{2}}. \tag{5.38}$$

By using this cross-track error, we can find an expression for the lookahead distance needed to respect the angular speed constraint (5.35). The maximum angular speed becomes

$$\dot{\chi}_r|_{e=\frac{\Delta}{\sqrt{2}}} = \dot{\chi}_{\mathrm{r,max}} = \frac{2}{3\sqrt{3}}\frac{U}{\Delta}, \tag{5.39}$$

and by substituting $\dot{\chi}_{\mathrm{r,max}} = \omega_{\max}(U)$ and solving for Δ, we get

$$\Delta(U) = \frac{2}{3\sqrt{3}}R_{\min}(U), \tag{5.40}$$

$$R_{\min}(U) = \frac{U}{\omega_{\max}(U)}. \tag{5.41}$$

This lookahead is constant for a constant speed U, and hence is called a semi-constant lookahead. Equation (5.40) reveals that the lookahead distance is proportional to the smallest circle the vehicle can follow at a given speed. Figure 5.12 shows that with the constant lookahead proposed here, the angular speed remains within feasible bounds. This choice is conservative, since more aggressive lookahead distances may still obey the angular speed constraint for cross-track errors other than $e = \pm\Delta/\sqrt{2}$. Next, we will consider the case where we let the lookahead distance vary as function of the cross-track error.

5.4.2 The principle of circular convergence

When following a circle, the angular speed is constant. If we manipulate the lookahead distance in such way that the vehicle follows a circle during the convergence toward the straight line, the angular speed is bounded. We will now consider the geometric aspects needed to achieve this objective.

From Figure 5.13 it can be seen that $(AB \perp DE) \wedge (BC \perp EF) \wedge (AC \perp DF)$. This implies that the triangles $\triangle ABC$ and $\triangle DEF$ are similar triangles. Similar triangles have the same shape, but one triangle is a scaled version of the other. Corresponding sides of the triangles are proportional to each other, hence we have

$$\frac{AB}{DE} = \frac{BC}{EF} = \frac{AC}{DF}. \tag{5.42}$$

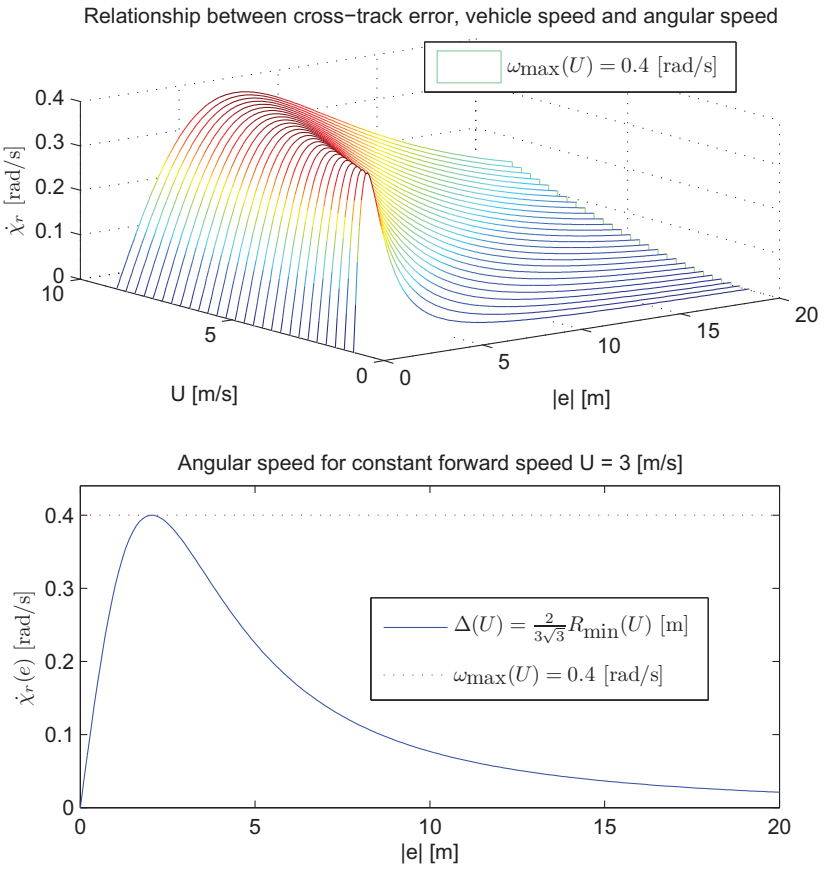


Figure 5.12: The angular speed for different speeds and cross-track errors, when using semi-constant lookahead.

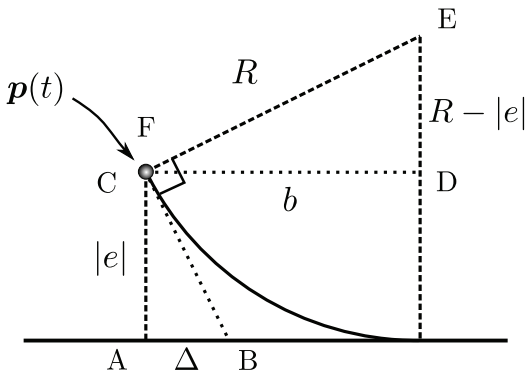


Figure 5.13: Circular convergence to straight line.

Since most of the sides of both triangles are known, it is possible to calculate Δ through the proportional relationship in (5.42). We get

$$\frac{\Delta}{R-|e|} = \frac{|e|}{b}, \tag{5.43}$$

where b is calculated using Pythagoras' theorem:

$$\begin{aligned} b^2 &= R^2 - (R-|e|)^2 && (5.44)\\ &\Downarrow \\ b &= \sqrt{2R|e| - e^2}. && (5.45) \end{aligned}$$

Substitute the expression for b into (5.43) and do some algebraic simplifications, we get the following expression for the lookahead distance

$$\Delta(e) = (R-|e|)\sqrt{\frac{|e|}{2R-|e|}}, \quad 0 < |e| < R. \tag{5.46}$$

The choice of $\Delta(e)$ yields a circular convergence toward the straight line. The radius of convergence equals R, and if the choice of U satisfies (4.15), the course rate is within the constraints imposed by (5.35). To make sure the angular speed is within feasible bounds, the expression for $\dot{\chi}_r(e)$ must be investigated.

Desired course characteristics The course rate can be shown to be

$$\dot{\chi}_r = \text{sgn}(e)\frac{U}{R}, \tag{5.47}$$

see Appendix C.2 for computational details. By combining (5.35) and (5.47) we get

$$|\dot{\chi}_r(e)| = \left|\text{sgn}(e)\frac{U}{R}\right| \leq \omega_{max}(U), \tag{5.48}$$

where R is the radius of convergence chosen in the expression of Δ. Recall from (5.41) that $R_{\min}(U)$ is the smallest radius for which (4.15) is satisfied for a given forward speed U. Equation (5.48) is then satisfied whenever $R \geq R_{\min}(U)$. This indicates that the choice of Δ respects the angular speed constraint. When R is close to $R_{\min}(U)$, the steering law is working close to the angular speed constraint and hence slack is introduced when R is increased. In Figure 5.14 the lookahead distance is shown for different forward speeds and cross-track errors when $\omega_{\max}(U) = 0.4$ [rad/s] and $R = R_{\min}(U)$.

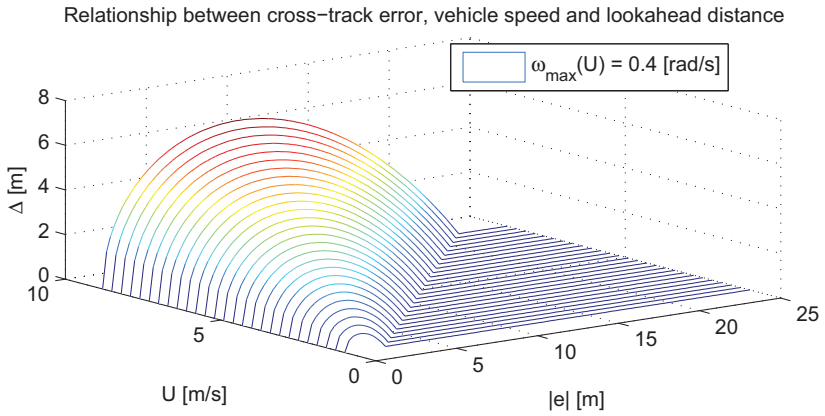


Figure 5.14: The lookahead distance when using the circular convergence for different speeds and cross-track errors.

The circular convergence only yields a $\mathcal{G}^1$-continuous path, that is, the angular speed has steps whenever there is a transition from a circular path to a straight line. Steps in the angular speed is impossible for physical vehicles, thus, the proposed approach must be improved to also obey an angular acceleration constraint.

5.4.3 Path convergence with angular acceleration constraint

To achieve a convergence toward a straight line when there is a angular acceleration constraint, transition phases with bounded angular acceleration must be developed. A possible solution is to combine the circular convergence with transition phases to obey both angular speed and acceleration constraints. We know from Section 3.3 that clothoids possess desirable properties such as bounded angular acceleration for particles following its curve. Hence, clothoids can be used to develop feasible transition phases.

Figure 5.15 illustrates the principle of the proposed approach. The convergence consists of three phases, two transistion phases made of clothoids, I and III, and a circular phase in between, denoted II. There are some aspects that need to be enlightened in order to be able to determine the appropriate course angle.

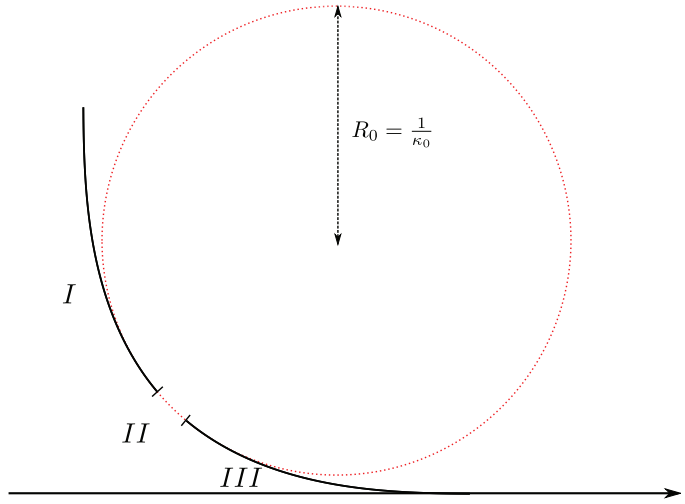


Figure 5.15: Desired path with transition phases.

Scaling the clothoids

Recall that for a given speed U, there is a upper curvature for which the vehicle is able to follow the path. Thus, the corresponding radius $R_{\min}$ is the smallest circle the vehicle can follow. Let $R_0 \geq R_{\min}$ be the desired radius on the circular path. We need to find the scaling of the clothoids so that both the angular speed constraint (4.14) and the angular acceleration constraint (4.16) are obeyed.

Let ϑ_0 be the parameter value of the clothoid for which it touches the circular path. We know that the curvature of the clothoid equals the one of the circular path where they meet, hence from (3.38) we find that the scaling of the clothoid is

$$a \;=\; R_0\sqrt{\vartheta_0}. \tag{5.49}$$

Contact angle between the clothoid and the circle

Next, we need to determine ϑ_0. Since we have assumed that the maximum angular acceleration constraint has the same magnitude as the deceleration constraint, both clothoids meet the circle with the same parameter value, that is $\vartheta_{I,0} = \vartheta_{III,0} = \vartheta_0$. For this reason, the circular arc must be equally shared between the clothoids. Recall that ϑ is the angle of the tangent of the clothoid with respect to the x-axis of the clothoid's reference frame, and thus

ϑ_0 must be in the interval $\langle 0, \pi/4]$, see Figure 5.16. The angular acceleration of a clothoid is constant and depends on the scaling a. Consequently, we can determine ϑ_0 by maximizing the angular acceleration. From (3.44) and (4.16) we have

$$\dot{\omega} = \frac{U^2}{2a^2} \leq \dot{\omega}_{max} \tag{5.50}$$

$$a^2 = \frac{U^2}{2\dot{\omega}_{max}}, \tag{5.51}$$

and by substituting the expression for the scaling (5.49) and solving for ϑ_0 we get

$$\vartheta_0 = \left(\frac{U}{R_0}\right)^2 \frac{1}{2\dot{\omega}_{max}} \in \left\langle 0, \frac{\pi}{4}\right], \tag{5.52}$$

where $R_0 = R_{\min}$. We have required that ϑ_0 falls within a validity interval. In some cases the calculation yields a value outside this interval. This means that the angular acceleration is too small to be able to reach the desired curvature within the interval $\langle 0, \pi/4]$. In this case we must increase the radius of the circular path so that ϑ_0 enters the required interval. We minimize R_0 by solving (5.52) with $\vartheta_0 = \frac{\pi}{4}$:

$$R_0 = U\sqrt{\frac{2}{\pi\dot{\omega}_{max}}}. \tag{5.53}$$

To summarize, if the contact angle $\vartheta_0 \in \left\langle 0, \frac{\pi}{4}\right]$, $R_0 = R_{\min}$, otherwise $\vartheta_0 = \frac{\pi}{4}$ and R_0 is chosen according to (5.53).

Placement of the circular path

The shortest distance from the desired straight line to the circular path is denoted l_0, as depicted in Figure 5.16. The center of the circle is given by (3.39) with $\vartheta = \vartheta_0$:

$$\boldsymbol{m}(\vartheta_0) = \begin{bmatrix} m_x \\ m_y \end{bmatrix} = \boldsymbol{p}(\vartheta_0) + \frac{1}{\kappa(\vartheta_0)} \begin{bmatrix} -\sin(\vartheta_0) \\ \cos(\vartheta_0) \end{bmatrix}, \tag{5.54}$$

and by consulting Figure 5.16 once more, we can see that

$$l_0 = m_y - R_0. \tag{5.55}$$

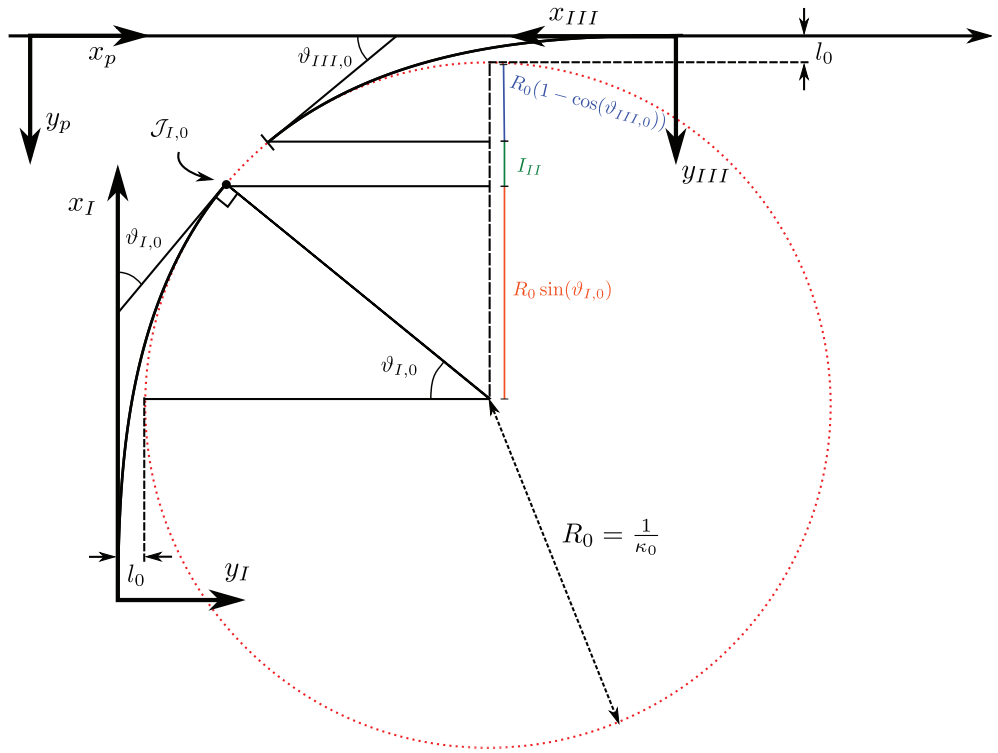


Figure 5.16: Relevant quantities with respect to placement of the clothoid reference frames.

With these quantities in place, we focus our attention toward placing the curves in relation to each other. The resulting trajectory constitute a desired path which respects the vehicle maneuverability constraints during the convergence. We want to extract the desired course angle from this trajectory only by knowing the cross-track error. The following sections describe the method for each of the phases.

The first transition phase

Transforming the cross-track-error to the clothoid parameter value
Denote the reference frame for phase I as {I}. For positive cross-track errors, the path-fixed reference frame {p} has its y-axis parallel with the x-axis of this frame, but in opposite direction, see Figure 5.16. The translational displacement of {I} with respect to the y_p-axis of {p} can be found by considering quantities in Figure 5.16. By adding: *i)* the shortest distance l_0 from the path to the circle; *ii)* the distance from this point to the contact point $J_{I,0}$ along the y_p-axis; and *iii)* the distance from origo of {I} to $J_{I,0}$ along the x_I-axis, we get

$$e_0 = l_0 + R_0\left(1 - \sin(\vartheta_0)\right) + aC(\vartheta_0), \tag{5.56}$$

where a is the clothoid scaling and $C(\vartheta_0)$ is calculated by (3.35a). Hence, for a given cross-track error e in {p}, the corresponding x-value in {I} is

$$x_I(e) \;=\; e_0 - |e|, \quad |e| \in \left[l_0 + R_0(1 - \sin(\vartheta_0)),\, e_0\right]. \tag{5.57}$$

Equation (5.57) together with the clothoid scaling and the contact angle gives sufficient information to determine the desired course $\chi_r(e)$.

To find the course angle, we first equate the x-coordinate of the clothoid (3.36) with $x_I(e)$ (5.57):

$$aC(\vartheta_I) = a\int_0^{\vartheta_I} \frac{\cos(u)}{\sqrt{u}}du \;=\; x_I(e), \tag{5.58}$$

and solve this equation with respect to ϑ_I. This is an implicit function, so a numerical method must be used. Define

$$g_I(\vartheta) \;=\; aC(\vartheta) - x_I(e) = 0, \quad \vartheta \in \langle 0, \vartheta_0\rangle, \tag{5.59}$$

$$g_I'(\vartheta) \;=\; a\frac{\cos(\vartheta)}{\sqrt{\vartheta}}, \tag{5.60}$$

and by iteratively computing the approximation of Newton's root-finding method (C.32):

$$\vartheta_{n+1} = \vartheta_n - \frac{g_I(\vartheta_n)}{g_I'(\vartheta_n)}, \tag{5.61}$$

it should converge to the correct value for $\vartheta_I(e)$, provided that the initial guess is sufficiently close. From (C.46) in Appendix C.5 we have that

$$\vartheta_{n=0} = \frac{d_I^2}{8}\left(1 + \frac{1}{\alpha_I^2}\right), \tag{5.62}$$

where $d_I = \frac{x_I(e)}{a}$ and $\alpha_I = \left(1 - \frac{\pi^2}{160}\right)$ is a reasonable initial guess.

The lookahead distance $\Delta_I(e)$ Let $\vartheta_I(e)$ be the solution of (5.58). By rotating $\vartheta_I(e)$ to the {p}-frame, we find the desired course angle for this phase to be

$$\chi_r(e) = -\text{sgn}(e)\left(\frac{\pi}{2} - \vartheta_I(e)\right) \in \pm\left[\frac{\pi}{2} - \vartheta_0,\, \frac{\pi}{2}\right], \tag{5.63}$$

see Figure 5.17.

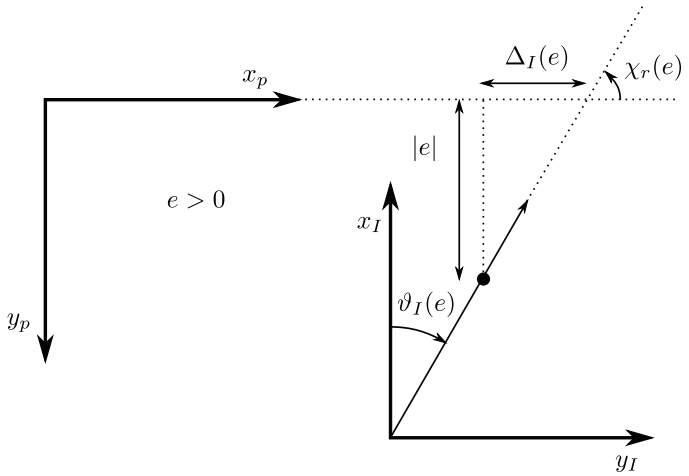


Figure 5.17: Trigonometric relations between {p} and {I}.

As described in Section (5.3.2), we can also express the desired course as a function of the lookahead distance $\Delta_I(e)$. By inspecting Figure 5.17, we can see that the lookahead distance can be expressed with the cross-track error and $\vartheta_I(e)$

$$\Delta_I(e) = |e| \tan\left(\vartheta_I(e)\right). \tag{5.64}$$

Desired course characteristics Because of the linearly increasing curvature along the clothoid path, choosing the lookahead distance according to (5.64) will yield a linearly increasing course rate

$$\dot{\chi}_r(e) \;=\; \mathrm{sgn}(e)U\kappa\left(\vartheta_I(e)\right) = \mathrm{sgn}(e)\frac{U}{R\left(\vartheta_I(e)\right)}, \tag{5.65}$$

see Appendix C.3.1 for computational details. The curvature increases from $\kappa(0) = 0$ to $\kappa(\vartheta_0) = \frac{1}{R_0}$ as it meets the circular path.

Furthermore, the course acceleration is constant since the course rate changes linearly. From (3.44) we have

$$\dot{\omega} = \ddot{\chi}_r(e) \;=\; \mathrm{sgn}(e)\frac{U^2}{2a^2}, \tag{5.66}$$

and by inserting the clothoid scaling from (5.49), we can write

$$\ddot{\chi}_r(e) \;=\; \mathrm{sgn}(e)\left(\frac{U}{R_0}\right)^2\frac{1}{2\vartheta_0}. \tag{5.67}$$

If we also substitute ϑ_0 from (5.52) we confirm that the angular acceleration is constant in this phase, obeying the angular acceleration constraint:

$$\ddot{\chi}_r(e) \;=\; \mathrm{sgn}(e)\left(\frac{U}{R_0}\right)^2\frac{1}{2\left(\frac{U}{R_0}\right)^2\frac{1}{2\dot{\omega}_{max}}} \tag{5.68}$$

$$\;=\; \mathrm{sgn}(e)\dot{\omega}_{max}. \tag{5.69}$$

Circular convergence phase, revisited

When the cross-track error becomes smaller than a given limit, phase I is no longer valid and the vehicle enters the circular phase, denoted phase II. In Section 5.4.2 the circular phase was discussed when the desired straight line was tangent to the circle. Now, the circle is placed above the straight line, so a slight modification of the lookahead distance formula (5.46) must take place. Consider Figure 5.18 with the circle displaced a distance l_0 above the desired straight line. By using the lookahead distance formula for the circular convergence as a starting point, we can lengthen this lookahead so that it aims at the desired straight line and not just the tangent of the circle. Once more, we can use the fact that $\triangle ABC$ and $\triangle DEF$ are similar triangles, and corresponding sides are scaled versions of each other. The figure shows that

$$\frac{\Delta_{II}}{\Delta} \;=\; \frac{|e|}{|e^\star|}, \tag{5.70}$$

and using $|e| = l_0 + |e^\star|$, the lookahead distance of phase II is

$$\Delta_{II}(e) \;=\; \Delta(|e| - l_0)\frac{|e|}{|e| - l_0}, \quad |e| \in I_{II} \tag{5.71}$$

where $\Delta(|e| - l_0)$ is the original formula for circular convergence (5.46) and I_{II} is the validity interval for phase II. By studying Figure 5.16, we conclude that

$$I_{II} \;=\; \langle l_0 + R_0(1 - \cos(\vartheta_0)),\, l_0 + R_0(1 - \sin(\vartheta_0))\rangle\,. \tag{5.72}$$

As expected, this interval diminishes for increasing ϑ_0 and vanish for $\vartheta_0 = \frac{\pi}{4}$.

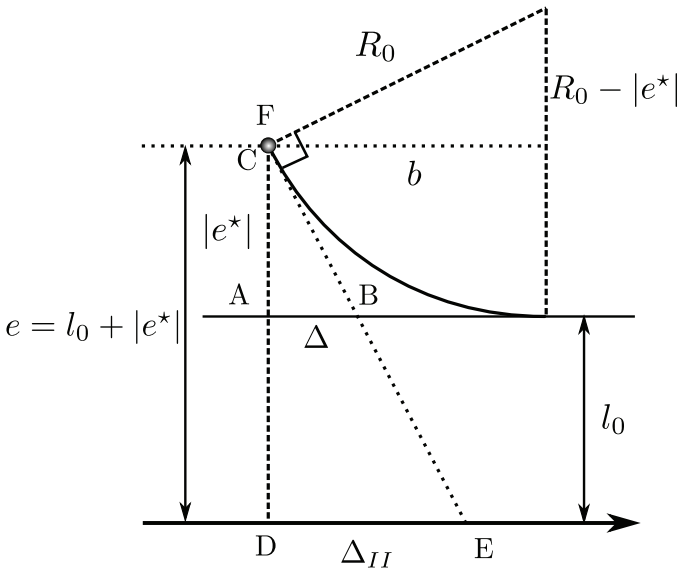


Figure 5.18: Δ_{II} is a scaled version of Δ when $l_0 > 0$.

The second transition phase

Transforming the cross-track-error to the clothoid parameter value
The transition phase with decreasing curvature is entered when the cross-track error

$$|e| \;\leq\; l_0 + R_0(1 - \cos(\vartheta_0)). \tag{5.73}$$

This phase, denoted phase III, has similar properties as phase I. Luckily, it is simpler to figure out the parameter value $\vartheta_{III}(e)$ compared to $\vartheta_I(e)$. Figure

5.16 reveals that the y-axis of {p} and {III} are parallel in the same direction for $e > 0$. Hence,

$$y_{III}(e) \;=\; |e|, \quad |e| \in [0,\, l_0 + R_0(1 - \cos(\vartheta_0))]\,. \tag{5.74}$$

Furthermore, equating the y-coordinate of the clothoid (3.35b) with $y_{III}(e)$

$$aS(\vartheta_{III}) = a\int_0^{\vartheta_{III}} \frac{sin(u)}{\sqrt{u}}du \;=\; y_{III}(e), \tag{5.75}$$

and solving with respect to ϑ_{III}, it is possible to determine the desired course angle. Again, this is an implicit function that has to be solved with Newton's root-finding method. Define

$$g_{III}(\vartheta) \;=\; aS(\vartheta) - y_{III}(e) = 0, \quad \vartheta \in \langle 0, \vartheta_{III,0}]\,, \tag{5.76}$$

$$g'_{III}(\vartheta) \;=\; a\frac{\sin(\vartheta)}{\sqrt{\vartheta}}, \tag{5.77}$$

and by iteratively computing the approximation of Newton's root-finding method (C.32):

$$\vartheta_{n+1} \;=\; \vartheta_n - \frac{g_{III}(\vartheta_n)}{g'_{III}(\vartheta)}, \tag{5.78}$$

the approximation converges to $\vartheta_{III}(e)$ for a proper choice of the initial ϑ_{III}. Equation (C.55) in Appendix C.5 provides a feasible initial choice:

$$\vartheta_{n=0} \;=\; \frac{1}{2}\left(\frac{3d_{III}}{2}\right)^{2/3}\left(1 + \frac{1}{\alpha_{III}^{2/3}}\right), \tag{5.79}$$

where $d_{III} = \frac{y_I(e)}{a}$ and $\alpha_{III} = 1 - \frac{\pi^2}{224}$.

The lookahead distance $\Delta_{III}(e)$ The desired course angle for phase III is simply

$$\chi_r(e) \;=\; -\mathrm{sgn}(e)\vartheta_{III}(e) \in \pm\,[0,\, \vartheta_0]\,, \tag{5.80}$$

and the lookahead distance is

$$\Delta_{III}(e) \;=\; \frac{|e|}{\tan(\vartheta_{III}(e)) + \Delta_\vartheta}, \tag{5.81}$$

see Figure 5.19. $\Delta_\vartheta > 0$ is a small variable to avoid singularity when $\vartheta_{III} = 0$.

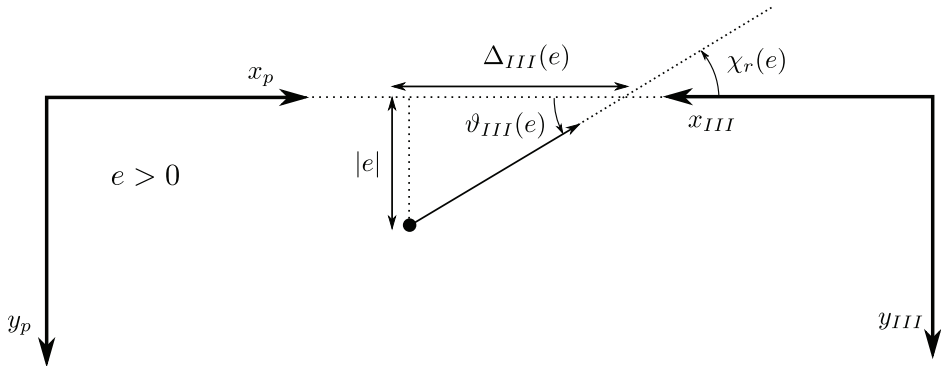


Figure 5.19: Trigonometric relations between {p} and {III}.

Desired course characteristics Similar to phase I, the curvature changes linearly along the clothoid path. However, since the path parameter $\vartheta_{III}(e)$ decreases in this phase, the course rate approaches zero linearly

$$\dot{\chi}_r(e) \;=\; \mathrm{sgn}(e) U \kappa\left(\vartheta_{III}(e)\right) = \mathrm{sgn}(e) \frac{U}{R\left(\vartheta_{III}(e)\right)}, \tag{5.82}$$

see Appendix C.3.2 for computational details. The curvature decreases from $\kappa(\vartheta_0) = \frac{1}{R_0}$ to $\kappa(0) = 0$ as it meets the desired straight line.

By following the same procedure as for phase I, we can show that the angular acceleration is constant in this phase. The difference in the deduction is that the along-path speed has changed sign to negative, which yields a negative acceleration. Specifically, (3.44) becomes

$$\dot{\omega}(\vartheta) = \ddot{\chi}_r(e) \;=\; -\mathrm{sgn}(e) \frac{U^2}{2a^2}, \tag{5.83}$$

and the course acceleration, or more precisely, the course deceleration is

$$\begin{aligned} \ddot{\chi}_r(e) &= -\mathrm{sgn}(e) \left(\frac{U}{R_0}\right)^2 \frac{1}{2\vartheta_0} && (5.84)\\ &= -\mathrm{sgn}(e) \left(\frac{U}{R_0}\right)^2 \frac{1}{2\left(\frac{U}{R_0}\right)^2 \frac{1}{2\dot{\omega}_{max}}} && (5.85)\\ &= -\mathrm{sgn}(e) \dot{\omega}_{max}, && (5.86) \end{aligned}$$

which satisfies the angular acceleration constraint (4.16).

5.4.4 Concluding remarks

The proposed clothoid approach yields lookahead distances which take the vehicle maneuverability constraints into account. Specifically, the angular speed and angular acceleration remains within feasible bounds when following the desired trajectory. Figure 5.20 shows qualitatively how different quantities vary when converging toward a straight line. Table 5.3 provides a convenient overview of the results developed in this chapter.

Property	Expression	Phase
Cross-track error	$\pm\left[l_0 + R_0(1-\sin(\vartheta_0)),\, e_0\right]$	I
	$\pm\left\langle l_0 + R_0(1-\cos(\vartheta_0)),\, l_0 + R_0(1-\sin(\vartheta_0))\right\rangle$	II
	$\pm\left[0,\, l_0 + R_0(1-\cos(\vartheta_0))\right]$	III
Lookahead distance	$\Delta_I(e) = \lvert e\rvert \tan(\vartheta_I(e))$	I
	$\Delta_{II}(e) = \Delta(\lvert e\rvert - l_0)\frac{\lvert e\rvert}{\lvert e\rvert - l_0}$	II
	$\Delta_{III}(e) = \frac{\lvert e\rvert}{\tan(\vartheta_{III}(e))}$	III
Course angle	$\chi_r(e) \in \pm\left[\frac{\pi}{2} - \vartheta_0,\, \frac{\pi}{2}\right]$	I
	$\chi_r(e) \in \pm\left\langle \vartheta_0,\, \frac{\pi}{2} - \vartheta_0\right\rangle$	II
	$\chi_r(e) \in \pm\left[0,\, \vartheta_0\right]$	III
Course rate	$\dot{\chi}_r(e) = \mathrm{sgn}(e)\frac{U}{R(\vartheta_I(e))}$	I
	$\dot{\chi}_r(e) = \mathrm{sgn}(e)\frac{U}{R(\vartheta_0)}$	II
	$\dot{\chi}_r(e) = \mathrm{sgn}(e)\frac{U}{R(\vartheta_{III}(e))}$	III
Course acceleration	$\ddot{\chi}_r(e) = \mathrm{sgn}(e)\dot{\omega}_{max}$	I
	$\ddot{\chi}_r(e) = 0$	II
	$\ddot{\chi}_r(e) = -\mathrm{sgn}(e)\dot{\omega}_{max}$	III

Table 5.3: Validity regions and equations for the different phases.

To further illustrate the lookahead distance for different speeds and cross-track errors, a 3D-plot is provided in Figure 5.21 for a vehicle with the following maneuverability constraints:

$$\omega_{\max}(U) = 0.4\,[\mathrm{rad/s}], \tag{5.87}$$
$$\dot{\omega}_{\max}(U) = 0.1\,[\mathrm{rad/s^2}]. \tag{5.88}$$

To get a preliminary comparison of the three proposed algorithms, plots of the lookahead distances as a function of cross-track error and the corresponding theoretical angular speeds are provided in Figure 5.22. The forward speed

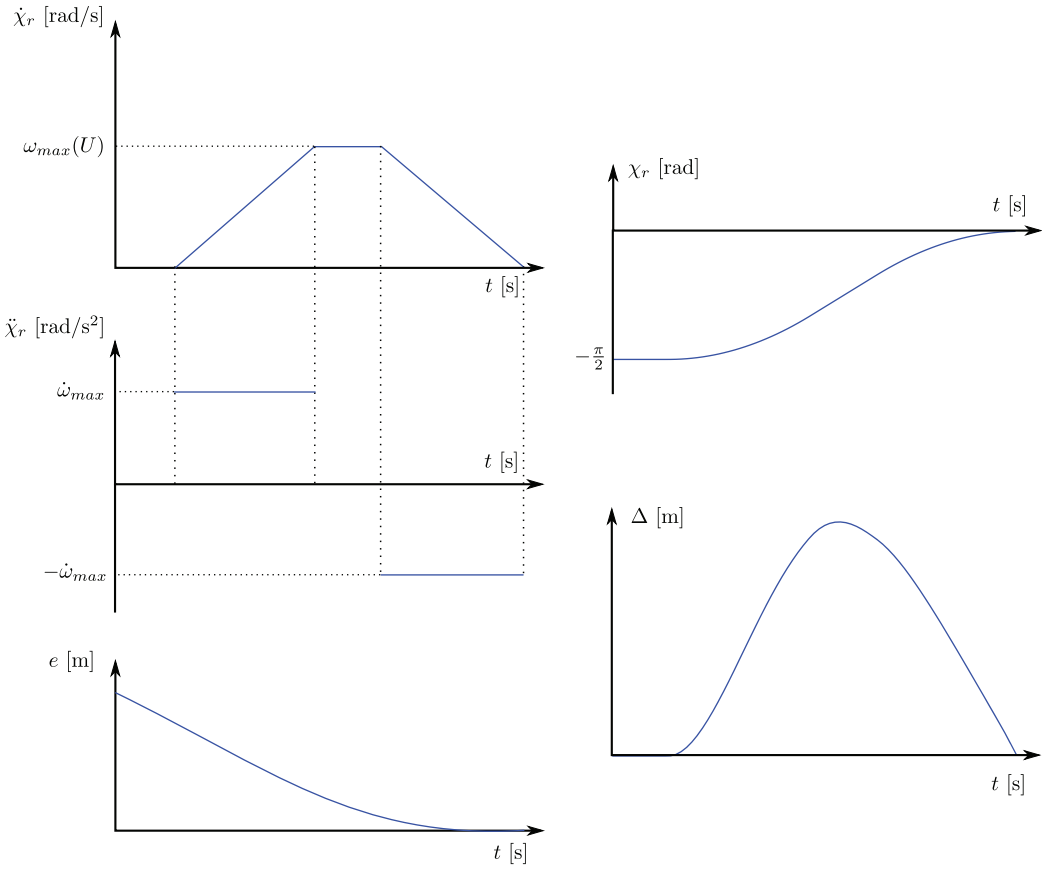


Figure 5.20: Idealized behavior of different quantities when using the proposed algorithm.

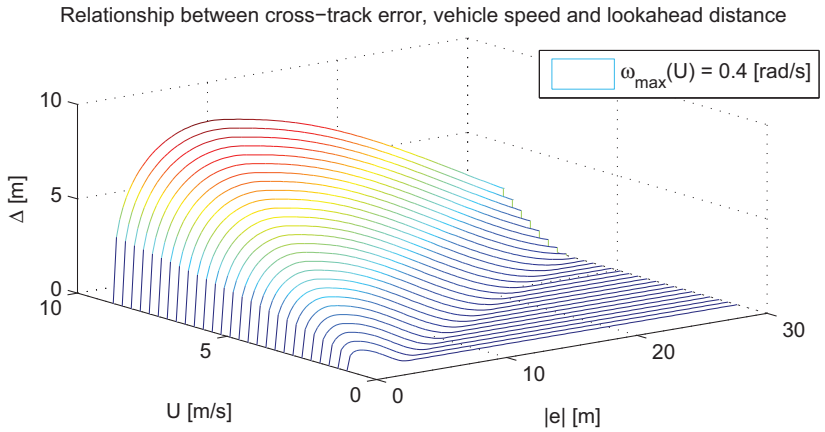


Figure 5.21: The variable lookahead distance for different speeds and cross-track errors.

is set to be $U = 3$ [m/s]. It can be seen that all the algorithms obey the angular speed constraint in this idealized setting. What cannot be seen however, is the angular acceleration of the approaches. Since the clothoid approach respect this constraint, we can deduce from the figure that neither the semi-constant, nor the circular convergence will obey this constraint.

Until now, we have only considered straight lines, where the path tangent is constant. In the next chapter we will investigate the effect a varying path tangential will have on the path convergence.

5.5 Path convergence for curved paths

The steering law in Section 5.3.2 uses knowledge about the tangent of the path's closest point to the vehicle's position. Hence, the steering law depends on the path tangent. This path tangent varies as the vehicle moves along the path, and thus becomes a part of the course dynamics. In the next section we examine how a varible path tangent influences the angular speed.

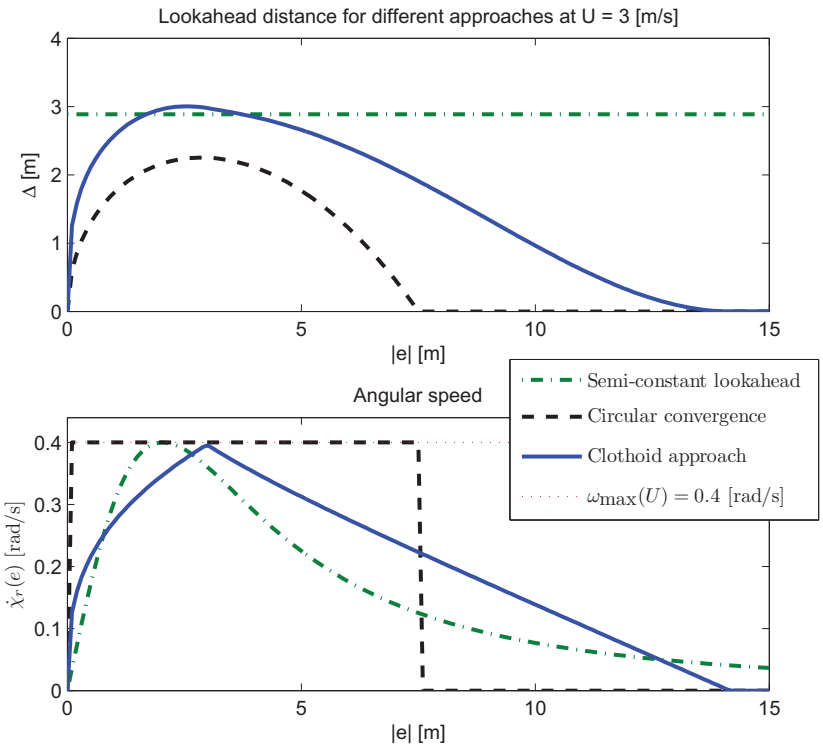


Figure 5.22: The lookahead distance and angular speed for different lookahead approaches.

5.5.1 The angular speed

When a vehicle follows a curved path, the path tangent changes with respect to time. The time derivative of the course is (5.29)

$$\dot{\chi}(e) \;=\; \dot{\chi}_r(e) + \dot{\chi}_p(\varpi), \tag{5.89}$$

but unlike the straight line scenario, we have to find an expression for $\dot{\chi}_p$. By using the result of $\dot{\chi}_r(e)$ in Chapter 5.4 and differentiate (5.16) with respect to time, we get

$$\dot{\chi}(e) \;=\; \mathrm{sgn}(e)\frac{U}{R(e)} + \frac{d}{dt}\left(\mathrm{atan2}(y'_p, x'_p)\right) \tag{5.90}$$

$$\;=\; \mathrm{sgn}(e)\frac{U}{R(e)} + \frac{1}{1+\left(\frac{y'_p}{x'_p}\right)^2}\frac{\frac{d}{dt}(y'_p)x'_p - \frac{d}{dt}(x'_p)y'_p}{x'^2_p}, \tag{5.91}$$

using the chain rule we can write

$$\frac{d}{dt}(x'_p) \;=\; x''_p\dot{\varpi}, \tag{5.92}$$

$$\frac{d}{dt}(y'_p) \;=\; y''_p\dot{\varpi}, \tag{5.93}$$

and the expression becomes

$$\dot{\chi}(e) \;=\; \mathrm{sgn}(e)\frac{U}{R(e)} + \frac{(y''_p x'_p - x''_p y'_p)\dot{\varpi}}{x'^2_p + y'^2_p}. \tag{5.94}$$

Furthermore, employing (3.15) and (3.16) yields

$$\dot{\chi}(e) \;=\; \mathrm{sgn}(e)\frac{U}{R(e)} + \underbrace{\frac{(y''_p x'_p - x''_p y'_p)}{|\boldsymbol{p}'_p(\varpi)|^3}}_{\varkappa(\varpi)} U_p, \tag{5.95}$$

where we have recognized the expression for signed curvature (3.21). Lastly, the along-path speed can be written as $U_p = U\cos(\chi_r)$, and thus the final expression for the course rate is

$$\dot{\chi}(e) \;=\; U\left(\frac{\mathrm{sgn}(e)}{R(e)} + \varkappa(\varpi)\cos(\chi_r(e))\right). \tag{5.96}$$

The angular speed should always stay within (4.14):

$$|\dot{\chi}| \;\leq\; \omega_{max}(U). \tag{5.97}$$

Since we cannot control $\dot{\chi}_p$, this must be achieved by controlling the path-relative course rate $\dot{\chi}_r$. Combine (5.96) and (5.97):

$$\left| U \left(\frac{\text{sgn}(e)}{R(e)} + \varkappa(\varpi) \cos(\chi_r(e)) \right) \right| \leq \omega_{max}(U), \tag{5.98}$$

which means that when e and $\varkappa$ have opposite signs, $\dot{\chi}_r$ can be increased and the inequality still holds. On the other hand, if they have same signs, $\dot{\chi}_r$ may have to be more conservative in order to fulfill the constraint. The new constraint for $\dot{\chi}_r$ is

$$|\dot{\chi}_r| \leq \omega_{max}(U) - \text{sgn}(e)\dot{\chi}_p, \tag{5.99}$$
$$|\dot{\chi}_r| \leq \omega_{max}(U) - \text{sgn}(e)U\varkappa(\varpi) \cos(\chi_r(e)), \tag{5.100}$$

where we assume that the angular speed of the path never exceeds the maximum angular speed of the vehicle. By using measured values for U and $\chi_r(e)$, this constraint can be used to make the convergence more conservative or aggressive depending on the cross-track error and its signum. Specifically, one can use the newly proposed angular speed constraint as the angular constraint in the already proposed lookahead dynamics from Chapter 5.4. However, with this choice of convergence, overshoot is not guaranteed to be avoided. The curvature of the path ahead of the vehicle combined with speed control must be employed to ensure proper convergence. These results will not be employed in the path convergence algorithms, and must thus be considered as future work.

5.6 Path-convergence heuristics

In this chapter, some additional heuristics are considered. The propositions are formed to make the lookahead steering more robust and pratically applicable.

5.6.1 Numerical issues

When the cross-track error is small, the lookahead distance becomes very small. Since the sampling frequency of an implementation is limited, numerical issues will arise and oscillations around the desired course is imminent. Hence, slack must be introduced. This can be done by introducing a constant

lookahead in addition to the variable lookahead. The constant lookahead may be chosen to be proportional to the semi-constant lookahead proposed in Section 5.4.1:

$$\Delta(U, e) \quad = \quad \Delta_{\text{var}}(e, U) + k_c \Delta_{\text{const}}(U), \tag{5.101}$$

where $k_c > 0$, $\Delta_{\text{var}}(e, U)$ the lookahead based on clothoids, and $\Delta_{\text{const}}(U)$ the semi-constant lookahead. This choice of lookahead makes the convergence more conservative, but avoids the numerical issues if k_c is chosen sufficiently high.

5.6.2 Course error $\tilde{\chi}$

The lookahead distances considered so far assume that the desired and actual course are identical, and will remain so, provided that the vehicle maneuverability constraints are obeyed. This is not a fair assumption, and is rarely true in practice. The course error $\tilde{\chi}$ is nonzero most of the time and affects the course response. To compensate for this, slack should be introduced to avoid a too aggressive response when the course error is large. Propose the following heuristic lookahead distance taking $\tilde{\chi}$ into account:

$$\Delta(U, e, \tilde{\chi}) \quad = \quad \Delta_{\text{var}}(e, U) + \Delta_{\text{const}}(U)(k_c + k_{\tilde{\chi}}|\tilde{\chi}|), \tag{5.102}$$

$k_{\tilde{\chi}} > 0$. This choice makes the convergence more conservative as the course error increases.

5.6.3 Speed control and $\tilde{\chi}$

If both the course error and the forward speed is large, the vehicle needs to move farther before it is able to converge to the path. Furthermore, the vehicle maneuverability constraints are more restrictive at higher speeds, motivating a speed reduction if the course error is large. A similar approach has been proposed in (Bibuli, Bruzzone & Caccia 2009). Let $U_d(t)$ be the desired forward speed. We wish to reduce this speed if the course error increases. Define the reference speed

$$U_{\text{ref}}(t, \tilde{\chi}) \quad = \quad U_{\min} + (U_d(t) - U_{\min}) S(\tilde{\chi}) \in [U_{\min}, U_d(t)], \tag{5.103}$$

where $U_{\min}$ is the smallest obtainable speed, and $S(\tilde{\chi})$ is a mapping function. The mapping function maps the course error into a scaling, such that a small

course error yields a large scaling, and vice versa. Thus, (5.103) will provide a reference speed which depends on the course error. Mathematically, the mapping function can be written as

$$S(\tilde{\chi}) : [-\pi, \pi] \mapsto [0, 1]\,. \tag{5.104}$$

A possible choice of the mapping function is

$$S(\tilde{\chi}) \;=\; \frac{1 + \cos(\tilde{\chi})}{2}, \tag{5.105}$$

but this choice does not lend any shaping capabilities for the implementor. Hence, we seek a bounded function which has shaping capabilities. Propose the more flexible mapping function

$$S(\tilde{\chi}) \;=\; \tanh\left(\frac{1}{k_U|\tilde{\chi}| + \Delta_{\tilde{\chi}}}\right), \tag{5.106}$$

where $k_U > 0$ and $\Delta_{\tilde{\chi}} > 0$. Increasing k_U will result in a more aggressive reduction in speed when a course error appears. $\Delta_{\tilde{\chi}} > 0$ is a small variable to prevent singularity when $\tilde{\chi} = 0$.

5.7 Path tracking

5.7.1 Motivation

Recall from Section 4 that *path tracking* is a motion control scenario where the objective is to track a target on a predefined path. This definition outlines a scenario for which the motion of the target is fully defined prior to executing the scenario. A practical application of the path tracking scenario may be the problem of following the path of a leader vehicle at a specified distance. When the surrounding environment is restricted, it is crucial to accurately track the path of the leader vehicle. In many cases, the leader vehicle's maneuvers may not be fully determined beforehand, making the general definition for path tracking inaccurate. If the path tracking definition is relaxed by letting the leader motion be revealed during the course of the scenario execution, several problems arise:

- How can the leader vehicle's motion be purposefully translated into a well-defined curve, making it applicable to a path tracking steering regime?

Figure 5.23: Yellow-billed ducklings tracking the motion of a mother duck. Courtesy of Melvin Gray / The Wildfowl & Wetlands Trust, www.wwt.org.uk.

- How should the path-tracking guidance law be constructed to guarantee convergence to the desired target?

Next, some previous propositions will be presented.

5.7.2 Previous work

As already stated in Section 4, path maneuvering can be divided into two tasks; a dynamic task and a geometric task (Skjetne et al. 2004). More specifically, a path-following steering law can be used to converge the vehicle to the path, while a speed adaptation regime makes sure the vehicle adheres to the desired target point along the path.

In (Breivik & Fossen 2009) this approach was proposed with the path-following steering law presented in Section 5.3.2, and a speed adaptation law. The speed adaptation uses the error between the actual and wanted parameter value of the path parameterization to create a bounded approach speed toward the moving target point. However, different parameterizations may require different tuning parameters, making them non-intuitive in the general case. Furthermore, this method cannot be used if the path consists of piecewise curve segments where the parameter value is not a continuously increasing parameter.

These shortcomings have been overcome in (Bibuli, Parodi, Lapierre & Caccia 2009). A similar method has been proposed, dividing the problem into two

separate tasks. Now, the speed adaptation depends on the arc-length deviation between the artificial vehicle particle and the target particle, making the deviation physically intuitive.

In the following sections, a similar method is developed. First, the objective is formed. Then, two methods for online generation of $\mathcal{G}^2$-continuous paths are presented. After that, the speed adaptation law is constructed together with proofs from cascaded stability theory.

5.7.3 The path-tracking objective

As pointed out in Section 4, the path-tracking scenario can be divided into two objectives

1. Path convergence
2. Path traversion

Mathematically, these two objectives can be formulated as

$$\lim_{t\to\infty} \left| \boldsymbol{p}(t) - \boldsymbol{p}_p(\varpi(t)) \right| = \boldsymbol{0}, \tag{5.107a}$$

$$\lim_{t\to\infty} \left| \boldsymbol{p}_p(\varpi(t)) - \boldsymbol{p}_p(\varpi_{tp}(t)) \right| = \boldsymbol{0}, \tag{5.107b}$$

where $\boldsymbol{p}(t)$ is the vehicle position, $\boldsymbol{p}_p(\varpi(t))$ is the projection of the vehicle onto the path, and finally; $\boldsymbol{p}_{pt}(\varpi_{pt})$ is the artificial target particle.

The first objective is solved using the path-following algorithm outlined in 5.3.2. The latter objective is a speed adaptation objective which will be elaborated on shortly. Figure 5.24 illustrates the problem at hand.

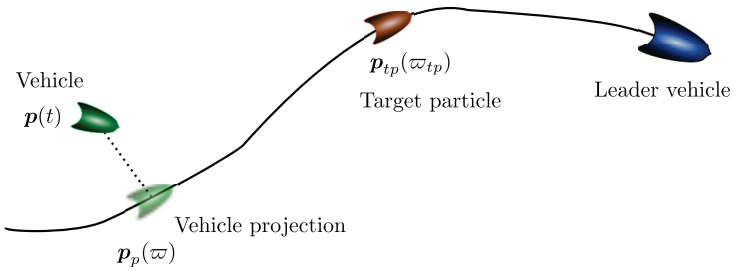


Figure 5.24: The path-tracking objective.

5.7.4 Online path generation

In marine environments, many vessels are equipped with a tracking device called *Automatic Identification System* (AIS). This system broadcasts information about the vessel's current state such as speed, position, course and rate of turn (USCG 2010). Receivers of this radio-transmitted data gain knowledge about the vessel's movement. Unfortunately, this instantaneous information is only broadcasted at certain variable time intervals, forcing continuous information to be estimated from these discrete data packages.

By employing a Kalman filter or some other appropriate estimation technique, a continuous expression for the leader vehicle's speed is obtained. In the following, it is assumed that the leader vehicle's speed is known with satisfactory accuracy. To estimate the leader vehicle's path, however, additional attention must be directed toward the continuity of the created curve segments. Specifically, care must be taken when stitching curve segments together, forcing $\mathcal{G}^2$ continuity where the segments meet. By doing so, the created curve satisfies the restrictions outlined in Section 3.1.6, and hence the curve is applicable for the already presented steering algorithm. We now proceed to present different path generation algorithms.

Path generation algorithms

The problem of path generation is a well-studied topic with many application areas. Thus, an extensive library of articles and approaches exists, each approach with its own angle of attack, using methods from the respective communities. In robot path planning, the problem is often to create a collision free trajectory, taking vehicle maneuverability constraints into account. Early work include (Delingette et al. 1991, Fleury et al. 1995, Scheuer & Fraichard 1997), where paths with continuous curvature were proposed. In the last decade, numerous authors have proposed different approaches to path planning, each optimal with respect to different objectives: (Nagy & Kelly 2001, Guarino Lo Bianco & Piazzi 2000, Lepetic et al. 2003, Yang et al. 2009), to mention only a few. Many of these approaches are complex and often specialized for one particular setting. In the two next sections, applicable methods with sufficient properties are reviewed. It is assumed that the leader vehicle is less maneuverable than the following vehicle, making the leader vehicle's path feasible for the following vehicle.

The $\boldsymbol{\eta}-$spline The objective of interpolating two configurations in planar space is solved in (Piazzi & Guarino Lo Bianco 2000). A quintic (5^{th} order) spline is proposed:

$$\boldsymbol{p}(\varpi)^\top = \begin{bmatrix} 1 & \varpi & \cdots & \varpi^5 \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ \vdots & \vdots \\ a_5 & b_5 \end{bmatrix}, \tag{5.108}$$

$$\varpi \in [0, 1]\,. \tag{5.109}$$

Rather than using blending functions, the approach is based on explicitly solving the polynomial coefficients for the given configurations. This approach guarantees that $\mathcal{G}^2$ continuity between curve segments are maintained. The equations for the polynomial coefficients can be found using the given conditions, such as positions, specified tangent directions and signed curvatures at these positions:

$$\begin{aligned} \boldsymbol{p}(0) = \boldsymbol{p}_A = \begin{bmatrix} x_A \\ y_A \end{bmatrix}, \quad & \boldsymbol{p}(1) = \boldsymbol{p}_B = \begin{bmatrix} x_B \\ y_B \end{bmatrix}, \\ \frac{\boldsymbol{p}'(0)}{|\boldsymbol{p}'(0)|} = \begin{bmatrix} \cos(\chi_A) \\ \sin(\chi_A) \end{bmatrix}, \quad & \frac{\boldsymbol{p}'(1)}{|\boldsymbol{p}'(1)|} = \begin{bmatrix} \cos(\chi_B) \\ \sin(\chi_B) \end{bmatrix}, \\ \varkappa(0) = \varkappa_A, \quad & \varkappa(1) = \varkappa_B, \end{aligned} \tag{5.110}$$

where χ_A and χ_B is given relative a stationary reference frame $\{\mathrm{s}\}$, see Figure 5.25. Further, the signed curvatures are given according to the right-hand rule.

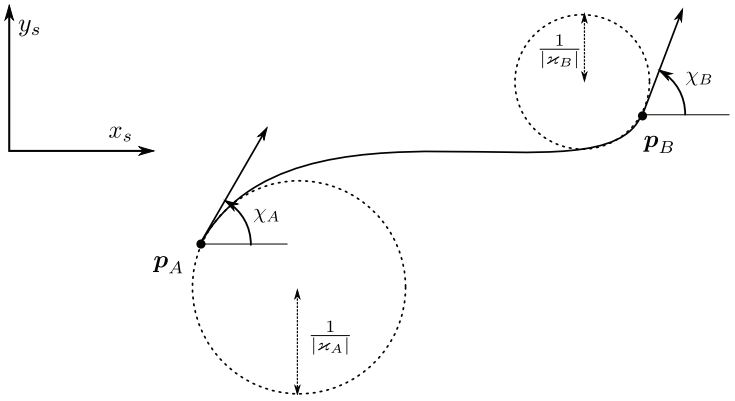


Figure 5.25: The configurations needed to find the $\boldsymbol{\eta}-$spline.

The coefficients that solve the configuration problem are found in Appendix C.6. As this solution reveals, the coefficients depend on additional param-

eters, denoted $\boldsymbol{\eta} = \begin{bmatrix} \eta_1 & \eta_2 & \eta_3 & \eta_4 \end{bmatrix}^\top$, hence the name $\boldsymbol{\eta}-$*splines*. These parameters are the shape parameters of the $\boldsymbol{\eta}-$spline, but are not intuitive to determine. In (Piazzi & Guarino Lo Bianco 2000), an optimization regime is outlined, minimizing the rate of change of curvature. However, the optimization problem is difficult to solve, so the same authors proposed a sub-optimal solution based on heuristics. The following choice of $\boldsymbol{\eta}$ gives satisfactory results in most practical cases (Piazzi et al. 2003):

$$\eta_1 = \eta_2 = |\boldsymbol{p}_A - \boldsymbol{p}_B|, \tag{5.111}$$

$$\eta_3 = \eta_4 = 0. \tag{5.112}$$

Usability with AIS Since AIS provides information about speed, position, course and rate of turn, the piecewise curve between arbitrary sampled positions can be constructed. Specifically, both position and course conditions can trivially be formed from the AIS-data. As for the curvature, the use of different mathematical relationships is needed. First, the unsigned curvature is found from (4.13) and (3.18):

$$\kappa = \frac{|\dot{\chi}|}{U}, \tag{5.113}$$

where $|\dot{\chi}|$ is the absolute value of the rate of turn, and U is the speed of the vehicle. Next, by also utilizing the signum of the provided rate of turn, the signed curvature is

$$\varkappa = \frac{\dot{\chi}}{U}, \tag{5.114}$$

making it possible to create a path similar to the leader vehicle's path.

If a different method for gathering the leader vehicle's state information is used, $\boldsymbol{\eta}-$splines cannot necessarily be applied. For instance, if only position data is sampled, a different approach must be considered. In the next section, one such method is reviewed.

A $\mathcal{G}^2-$continuous Catmull-Rom spline There exist many algorithms for creating $\mathcal{C}^2-$continuous splines. However, since the curves needed to employ the proposed path-following algorithm must be regular, that is, having non-vanishing path tangents, some of these algorithms cannot be used. In the work of (DeRose & Barsky 1988) a $\mathcal{G}^2-$continuous interpolating spline is

proposed. This approach is beneficial in online path generation due to its regularity, and interpolating properties. Now, a brief summary of this algorithm is given, emphasizing its possible application for online path generation.

Given a control polygon $\boldsymbol{V}_0, \cdots, \boldsymbol{V}_m$, the idea is to construct additional control points such that the resulting curve connects the original control points with $\mathcal{G}^2$ continuity. This is done by using Catmull-Rom splines, β-splines and Bézier curves.

A Catmull-Rom spline is a spline where the blending functions blend together functions rather than control points (Catmull & Rom 1974). These functions are called *interpolating functions*. In (DeRose & Barsky 1988) it is shown that by constructing geometrically continuous interpolating functions, the resulting curve can be made $\mathcal{G}^n$ continuous, with the proper choice of blending functions. The blending functions are taken from β-splines, which have shaping capabilities through shape parameters.

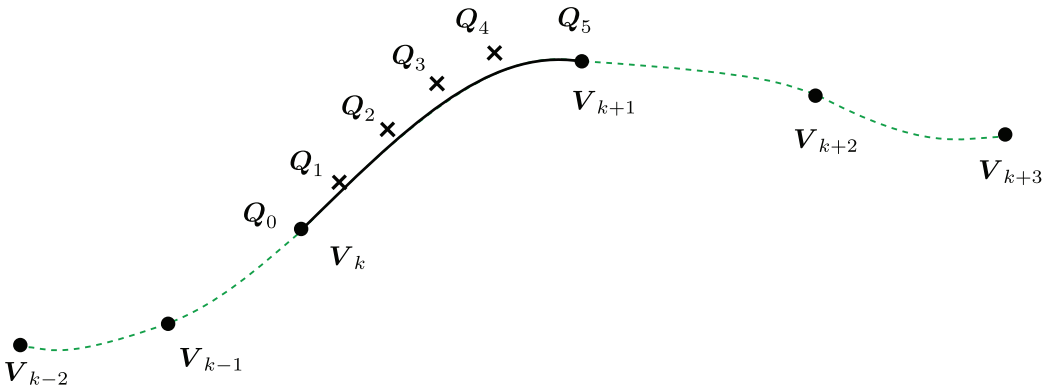


Figure 5.26: Given the original control points $\boldsymbol{V}_i$, additional control points $\boldsymbol{Q}_q$ are created between the original control points. A Bézier curve is created from the $\boldsymbol{Q}$-polygon.

The construction algorithm is complex, so a thorough explanation will not be given here, but a pseudocode is provided in Appendix C.7. For a detailed explanation of the algorithm, the reader is recommended to consult (DeRose & Barsky 1988). The output of the algorithm is a control polygon consisting of six control points between *each* original pair of control points, see Figure 5.26. Recall that a Bézier curve interpolates the end points and approximates the interior points. Hence, by finding the Bézier curve of the constructed control polygon, an interpolating curve segment between two point is created. This is simply done by using $\boldsymbol{Q}_i, \boldsymbol{Q}_{i+1}, \cdots \boldsymbol{Q}_{i+5}$ as the control polygon in (3.30), where $\boldsymbol{Q}_q$ is a control point in the constructed control polygon. When

stitching several Bézier curves together, a $\mathcal{G}^2-$continuous curve is achieved.

The created piecewise curve has L^6 locality with respect to the control points, and L^4 locality with respect to the shape parameters. On other words, pertubation of a control point affects six surrounding curve segments, while changing a shape parameter only affects four. In the implementation considered here, the shape parameters are constant, with $\boldsymbol{\beta}_1 = \mathbf{1}$ and $\boldsymbol{\beta}_2 = \mathbf{0}$, giving satisfactory behavior in our particular application. In fact, with this choice, a regular $\mathcal{C}^2$-continuous piecewise curve is obtained.

Considerations for online path generation Since the curve has L^6 locality, a minimum of six control points are needed to create a single curve segment, that is, to create a curve segment between the points $\boldsymbol{V}_k$ and $\boldsymbol{V}_{k+1}$, two points before $\boldsymbol{V}_k$ and two points after $\boldsymbol{V}_{k+1}$ are needed, as Figure 5.26 shows. Whenever a new point $\boldsymbol{V}_{k+4}$ becomes available, a new segment $\mathcal{P}_{k+1}$ can be created. The creation of this segment uses the points from $\boldsymbol{V}_{k-1}$ to $\boldsymbol{V}_{k+4}$. Hence this constitutes a First-in, first-out (FIFO) buffer with length equal to six control points. This FIFO-buffer is outlined in Table 5.4. When the buffer is populated, arrival of a new point results in creation of a new curve segment. In an initial phase, the buffer may not populated. A segment cannot be created before six points exist, so extrapolation of some kind might be necessary to yield a curve in the vicinity of the following vehicle.

Curve segment	Point	
$\mathcal{P}_{k+1}$	$\boldsymbol{V}_{k-2}$	$\leftarrow$ Old
	$\boldsymbol{V}_{k-1}$	
$\mathcal{P}_k$	$\boldsymbol{V}_k$	
	$\boldsymbol{V}_{k+1}$	$\Uparrow$ Direction
	$\boldsymbol{V}_{k+2}$	
	$\boldsymbol{V}_{k+3}$	
	$\boldsymbol{V}_{k+4}$	$\leftarrow$ New

Table 5.4: FIFO-buffer for curve segment creation.

The Catmull-Rom approach puts restricting conditions on the location of the target point with respect to the leader vehicle's position. Specifically, if the target particle is on a segment $\mathcal{P}_k$ between the positions $\boldsymbol{V}_k$ and $\boldsymbol{V}_{k+1}$, the position denoted $\boldsymbol{V}_{k+4}$ must be available before the target particle is finished with the traversion of curve segment $\mathcal{P}_k$. Hence, the target particle cannot be closer to the leader vehicle than the arc length of three curve segments.

Furthermore, some slack should also be added, so that the vehicle projection can move past the target particle in a possible overshoot. Figure 5.27 shows the problem at hand. We get

$$\Delta_{pt} > \text{slack} + \sum_{i=1}^{3} \Delta_{k+i}. \tag{5.115}$$

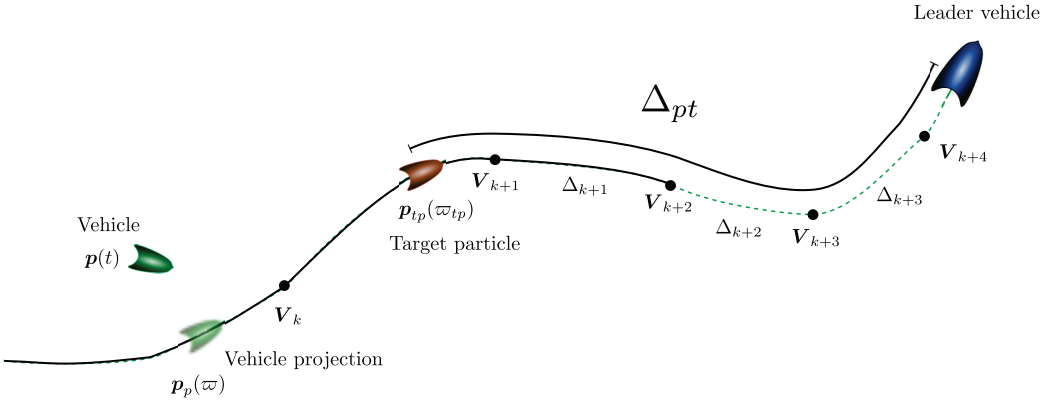


Figure 5.27: Limitation on the desired along-path, path-tracking distance Δ_{pt}.

As a remark it should be mentioned that the $\boldsymbol{\eta}-$spline puts less restrictions on the closeness of the target particle. Only a single curve segment between the target particle and the leader vehicle is undetermined. It should however be emphasized that if an AIS transponder is used, the sampling frequency can vary greatly, adding another degree of uncertainty.

If (5.115) is not satisfied, two things can be done. Either extrapolation of the leader vehicle's position is needed, or increasing the sampling frequency of the positions, if possible, making the arc-length between consecutive points smaller.

5.7.5 The path-tracking system

Artificial target particle with offset correction

None of the proposed methods produce a 100% exact reconstruction of the original path. Thus an offset in arc length between the real and generated path is inevitable. This offset must be compensated for, if not, the target

particle will eventually drift away from the desired arc length behind the leader vehicle. If we assume that the real speed of the leader vehicle is known, the numerical integral of this speed equals the distance covered by the leader vehicle. Define the real distance covered by the leader vehicle between the points $\boldsymbol{V}_k$ and $\boldsymbol{V}_{k+1}$ as

$$\Delta_k = \int_{t_k}^{t_{k+1}} U_\ell(t)dt, \tag{5.116}$$

where $U_\ell(t)$ is the real leader vehicle speed, t_k and t_{k+1} is the sampling times of the points, respectively. Further, let the corresponding arc length of the generated curve segment $\mathcal{P}_k$ be (3.11)

$$\hat{\Delta}_k \triangleq \int_{\varpi_0}^{\varpi_1} |\boldsymbol{p}'_{p,k}(\varpi)|d\varpi. \tag{5.117}$$

The error between these arc lengths should be compensated by the target particle which follows the generated path. Define the offset

$$\tilde{s}_{\text{offset},k} \triangleq \hat{\Delta}_k - \Delta_k, \tag{5.118}$$

which is the offset error for curve segment $\mathcal{P}_k$. A positive offset means that the generated curve segment is longer than the real segment, hence the target particle must speed up to avoid drifting. Over time, the distance the target particle must travel compared to the leader vehicle depends on the cumulative offset each curve segment produce:

$$\tilde{s}_{\text{offset}} = \sum_k \tilde{s}_{\text{offset},k}. \tag{5.119}$$

Let $U_{tp}(t)$ be the speed of the target particle. Propose a simple proportional regulator with feedforward from the speed of the leader vehicle:

$$U_{tp}(t) = U_\ell(t) - k_{p,pt} \underbrace{(\tilde{s}_{pt} - \tilde{s}_{\text{offset}})}_{\tilde{s}_d}, \tag{5.120}$$

where $k_{p,pt} > 0$ is the proportional path-tracking gain. Furthermore,

$$\tilde{s}_{pt}(t) = s_{tp}(t) + \Delta_{pt} - s_\ell(t) \tag{5.121}$$

is the difference in distance covered by the target particle $s_{tp}(t)$ and leader vehicle $s_\ell(t)$. Δ_{pt} is the along-path distance between the leader vehicle and target particle, see Figure 5.27.

If the real vehicle speed cannot be determined with sufficient degree of accuracy, some other means for compensating the accumulating offset must be constructed. For instance, when a position is sampled, it can be handed a temporal condition, demanding the target particle to be close to this position a specified time delay after the leader vehicle has been there. This will in practice create similar steps in the desired target particle position, thus, the same controller apply, with some modifications. These modifications will not be considered here.

Now, an artificial target particle follows the created path at the desired along-path distance behind leader vehicle. Next a path-tracking speed controller is developed.

Path-tracking speed controller

The path-tracking speed controller must force the vehicle projection to the desired target position represented by the target particle (5.107b):

$$\lim_{t\to\infty} \left| \boldsymbol{p}_p(\varpi(t)) - \boldsymbol{p}_{tp}(\varpi_{tp}(t)) \right| \;=\; \mathbf{0}. \tag{5.122}$$

In a similar manner as in (Skejic et al. 2010), propose the following bounded reference speed for the vehicle projection with feedforward from the leader vehicle's speed:

$$U_p(t) \;=\; U_\ell(t) + U_{\mathrm{a,max}} \frac{\tilde{s}_p(t)}{\sqrt{\tilde{s}_p(t)^2 + \Delta_s^2}}, \tag{5.123}$$

where $\tilde{s}_p(t) = s_{tp}(t) - s_p(t)$ and $s_p(t)$ is the arc length of the vehicle projection. Furthermore, $U_{\mathrm{a,max}}$ is the largest approach speed toward the target particle, and Δ_s is a shaping parameter determining the rendezvous behavior of the approach speed.

The leader vehicle's speed is chosen as the feedforward rather than the target particle's speed. This is due to the non-physical speeds the target particle obtain when a new offset arrives. Physical speeds could been obtained with a more conservative $k_{p,pt}$, but it is better to compensate the offset-induced error with the *bounded* feedback term. However, this choice complicates the stability analysis somewhat, as will be obvious in the next section.

Stability analysis

We can write the error states of the path-tracking system, which consists of the target-particle correction and the path-tracking speed controller, as

$$\tilde{s}_p(t) = s_{tp}(t) - s_p(t), \tag{5.124a}$$
$$\tilde{s}_d(t) = s_{tp}(t) + \Delta_{pt} - s_\ell(t). \tag{5.124b}$$

Taking the time derivatives of each term:

$$\dot{\tilde{s}}_p = \dot{s}_{tp} - \dot{s}_p, \tag{5.125a}$$
$$\dot{\tilde{s}}_d = \dot{s}_{tp} + \dot{\Delta}_{tp} - \dot{s}_\ell - \dot{\tilde{s}}_{\text{offset}}, \tag{5.125b}$$

and using the fact that the time derivative of the arc length is the speed, $\dot{\Delta}_{pt} = 0$, and with slight abuse of notation: $\dot{\tilde{s}}_{\text{offset}} = 0$, which is true most of the time, we can write

$$\dot{\tilde{s}}_p = (U_\ell - k_{\text{p,pt}}\tilde{s}_d) - \left(U_\ell + U_{\text{a,max}}\frac{\tilde{s}_p}{\sqrt{\tilde{s}_p^2 + \Delta_s^2}}\right), \tag{5.126a}$$
$$\dot{\tilde{s}}_d = (U_\ell - k_{\text{p,pt}}\tilde{s}_d) - U_\ell. \tag{5.126b}$$

The error dynamics becomes the following cascaded system:

$$\Sigma_1 \;:\; \dot{\tilde{s}}_p = -U_{\text{a,max}}\frac{\tilde{s}_p}{\sqrt{\tilde{s}_p^2 + \Delta_s^2}} - k_{\text{p,pt}}\tilde{s}_d, \tag{5.127a}$$
$$\Sigma_2 \;:\; \dot{\tilde{s}}_d = -k_{\text{p,pt}}\tilde{s}_d. \tag{5.127b}$$

This system has similar form as the cascade system in (C.58). We now apply Theorem C.3 to investigate the stability properties of the system. For this theorem to be valid, each of the Assumptions C.1-C.4 must hold. Thus, we must investigate the validity of these assumptions.

First, we define the system needed in Assumption C.1:

$$\dot{\tilde{s}}_p = -U_{\text{a,max}}\frac{\tilde{s}_p}{\sqrt{\tilde{s}_p^2 + \Delta_s^2}}. \tag{5.128}$$

Furthermore, let $\boldsymbol{x} = \text{col}[\tilde{s}_p, \tilde{s}_d]$.

Validity of Assumption C.1 Define the CLF and its derivative

$$V_1(\tilde{s}_p) = \frac{1}{2}\tilde{s}_p^2 > 0 \,\forall s_p \neq 0 \tag{5.129}$$

$$\dot{V}_1 = \tilde{s}_p\dot{\tilde{s}}_p, \tag{5.130}$$

and substitute (5.128) we get

$$\dot{V}_1 = -U_{\mathrm{a,max}}\frac{\tilde{s}_p^2}{\sqrt{\tilde{s}_p^2+\Delta_s^2}} < 0 \,\forall s_p \neq 0, \tag{5.131}$$

which renders the system UGAS. In fact, linearizing the system around the origin, it is obvious that the system also is ULES, see Appendix C.9. Thus, the system (5.128) is UGAS/ULES.

From (5.129) we find α_1, $\alpha_2 \in \mathcal{K}_\infty$ to be

$$\alpha_1(s) = \alpha_2(s) = \frac{1}{2}s^2, \tag{5.132}$$

and from (5.130), the positive semidefinite function

$$W(s) = \frac{s^2}{\sqrt{s^2+\Delta_s^2}}, \tag{5.133}$$

and finally $\alpha_4(s)$:

$$\left|\frac{\delta V}{\delta \tilde{s}_p}\right| \leq \alpha_4(s) = s. \tag{5.134}$$

Equations (5.132)-(5.134) ensure that Assumption C.1 holds.

Validity of Assumption C.2 Define the CLF and its derivative

$$V_2(\tilde{s}_d) = \frac{1}{2}\tilde{s}_d^2 > 0 \,\forall s_d \neq 0 \tag{5.135}$$

$$\dot{V}_2 = \tilde{s}_d\dot{\tilde{s}}_d, \tag{5.136}$$

and substitute (5.127b) we can conclude from

$$\dot{V}_2 = -k_{\mathrm{p,pt}}\tilde{s}_d^2 < 0 \,\forall s_d \neq 0 \tag{5.137}$$

that Σ_2 is UGES and Assumption C.2 holds.

Validity of Assumption C.3 By consulting (C.58) we see that in our system we have

$$g(t, \boldsymbol{x}(t; t_\circ, \boldsymbol{x}_\circ)) = k_{\mathrm{p,pt}}, \tag{5.138}$$

so $\alpha_5(s)$ can be defined as

$$\alpha_5(s) = \frac{k_{\mathrm{p,pt}}}{c_g(s)} = \frac{k_{\mathrm{p,pt}}}{s}, \tag{5.139}$$

where we have chosen $c_g(s) = s \in \mathcal{K}$. Next, $\alpha_6(s)$ must satisfy (C.64):

$$\alpha_6(s) \geq \alpha_4(\alpha_1^{-1}(s))\alpha_5(\alpha_1^{-1}(s)). \tag{5.140}$$

We substitute the obtained functions for α_4 and α_5 with the inverse function $\alpha_1^{-1}(s) = \sqrt{2s}$ as parameter and get

$$\alpha_6(s) \geq \sqrt{2s}\frac{k_{\mathrm{t,pt}}}{\sqrt{2s}} = k_{\mathrm{p,pt}} \tag{5.141}$$

$$\alpha_6(s) \triangleq k_{\mathrm{p,pt}}, \tag{5.142}$$

so that

$$\int_a^\infty \frac{1}{\alpha_6(s)} ds = \infty, \quad a = 0, \tag{5.143}$$

which makes Assumption C.3 valid.

Validity of Assumption C.4 The inequality (C.68)

$$\left| \frac{\delta V_1}{\delta \tilde{s}_p} g(t, \boldsymbol{x}) \right| \leq \lambda W(\tilde{s}_p), \quad |\tilde{s}_p| > \eta > 0, \tag{5.144}$$

must hold for $r > 0$, $|\tilde{s}_d| < r$ and $\lambda > 0$. Substitute expressions previously calculated and get

$$|\tilde{s}_p k_{\mathrm{p,pt}}| \leq \lambda U_{\mathrm{a,max}} \frac{\tilde{s}_p^2}{\sqrt{\tilde{s}_p^2 + \Delta_s^2}} \tag{5.145}$$

$$k_{\mathrm{p,pt}} \leq \lambda U_{\mathrm{a,max}} \frac{|\tilde{s}_p|}{\sqrt{\tilde{s}_p^2 + \Delta_s^2}}, \tag{5.146}$$

which can be made valid with the proper choice of $|\tilde{s}_p| \geq \eta > 0$ and $\lambda > 0$. Hence, Assumption C.4 is also valid.

Conclusion Since all the Assumptions C.1-C.4 hold, we can conclude from Theorem C.3 that the cascaded tracking system 5.127 is UGAS/ULES because $-U_{\mathrm{a,max}}\frac{\tilde{s}_p}{\sqrt{\tilde{s}_p^2+\Delta_s^2}}$ majorizes $k_{\mathrm{p,pt}}$.

5.8 Path planning scenarios

5.8.1 Motivation and previous work

An autonomous vehicle that performs a path maneuvering task needs a path to follow. Depending on the task at hand, the planning of the particular path varies significantly. Often, a set of waypoints is provided where the straight lines between consecutive waypoints are the ultimate goal. This is common in shipping, where the objective is to move from a starting point to a destination. One task of the path planning problem is to purposefully determine the location of each waypoint in such way that the resulting path is somehow optimal and collison free. Recent work on this account can be found in (Nord 2010).

An area of application for unmanned aerial vehicles (UAV)s and unmanned surface vehicles (USV)s is to systematically cover a geographic area. This is done to gather information about the region, for instance, monitoring a coast line or hydrocarbon exploration. A future area of application is monitoring ice conditions to support the hydrocarbon exploitation in arctic environments.

There exist several approaches for traversing a geographic region. A commonly used pattern is the *lawn-mower pattern* (Frost 1999). This is chosen due to its simplicity. As pointed out in (Ousingsawat & Earl 2007), vehicle maneuverability constraints affects performance significantly. There may be strict demands on cross-track error when following the lines. For instance, when scanning a grid, the gathered data will perhaps be post-processed and combined with earlier collected data. Thus, it is not desirable that the harvested data does not cover the intended area.

In (Ousingsawat & Earl 2007) the lawn-mower pattern is modified to include weighting of subsections of the search grid at hand. The weighting signifies the uncertainty associated with the specific subsection. The authors proposed a method to determine the optimal order the parallel lines is traversed. This is done to minimize coverage time and maximize area of coverage.

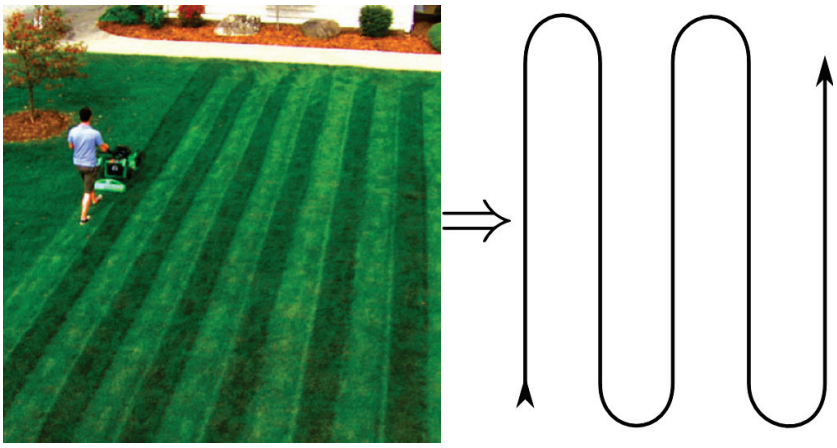


Figure 5.28: The lawn-mower pattern consists of straight lines and U-turns. Courtesy of www.uniquedaily.com.

5.8.2 Lawn-mower pattern

The lawn-mower pattern is basically a set of parallel lines. Let the pattern be defined by the planar positions $\boldsymbol{P}_k$ at the edges of each line, often called waypoints. The waypoint matrix becomes

$$\boldsymbol{P} = \left[\begin{array}{cccc} \boldsymbol{P}_0, & \boldsymbol{P}_1, & \cdots, & \boldsymbol{P}_{n-1} \end{array} \right] \in \mathbb{R}^{2\times n},\ n \geq 3, \tag{5.147}$$

where the vectors $\overrightarrow{\boldsymbol{P_i P_j}}$ and $\overrightarrow{\boldsymbol{P_{i+2} P_{j+2}}}$ are parallel and in opposite directions, and $\overrightarrow{\boldsymbol{P_i P_j}} \perp \overrightarrow{\boldsymbol{P_j P_{i+2}}}$. The index $i \in \{2\ell;\, \forall \ell \in \mathbb{N}_0\}$ is an even number and $j = i + 1$ is an odd number.

The distance between the parallel lines $\overrightarrow{\boldsymbol{P_i P_j}}$ and $\overrightarrow{\boldsymbol{P_{i+2} P_{j+2}}}$ can be written as

$$d_{\frac{j+1}{2}} \triangleq \left| \boldsymbol{P}_{j+1} - \boldsymbol{P}_j \right|, \tag{5.148}$$

where $j = i + 1$ is an odd number, and $i \leq n - 2$. In most lawn mower patterns, the distance between each parallel line is the same, but can in general vary.

When going from one parallel line to the next, the vehicle cannot follow a straight line which connects the adjacent positions $\boldsymbol{P}_j$ and $\boldsymbol{P}_{j+1}$. This path has corners with infinite curvature, which is impossible to follow for physical vehicles. A path must be constructed in such way that a vehicle that follows the path, performs U-turns between the parallel lines while obeying

angular speed and angular acceleration constraints. Furthermore, a piecewise path consisting of straight lines is not regularly parameterized with bounded curvature, and thus the path following algorithm discussed in earlier chapters cannot be used. Hence, to follow the parallel lines with the path following steering law, a smooth curve is needed.

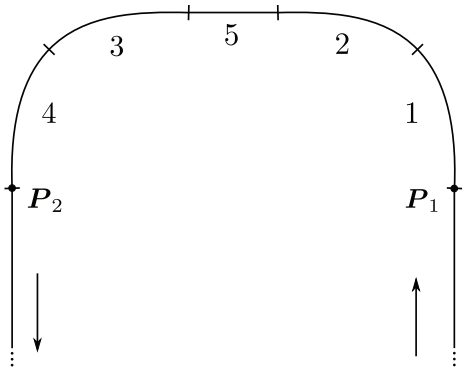
The objective of a path planning algorithm for lawn-mower patterns can be stated as:

Objective 5.1. *Lawn-mower path planning*
Given the vehicle speed U and the corresponding vehicle maneuverability constraints, $\omega_{max}(U)$ and $\dot{\omega}_{max}(U)$, explained in Chapter 4.2, construct feasible U-turns for the waypoint matrix $\boldsymbol{P}$.

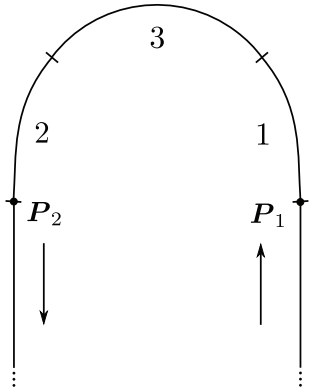
5.8.3 Lawn-mower pattern using clothoids

Clothoids can easily be scaled in such way that the angular speed and angular acceleration constraints are obeyed. The character of the U-turns depend on the distance d between the parallel lines and the vehicle maneuverability constraints stated in Objective 5.1. Thus, the turn can be divided into three different cases. Next, a qualitative description of each case is presented, followed by elaborations of each case.

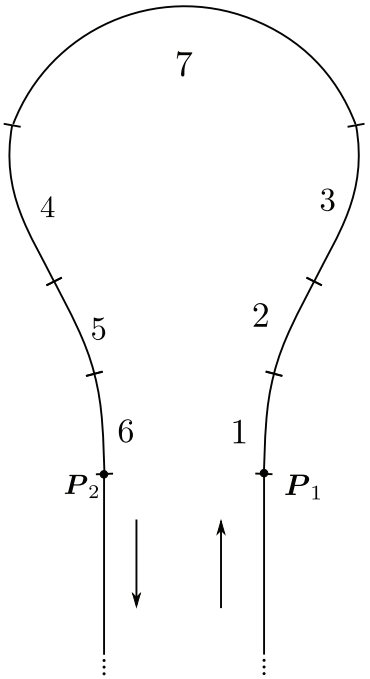
Case A: When d is sufficiently large, two 90° turns and possibly a straight line can be constructed. The figure on the right displays the resulting U-turn. The line segments enumerated *1* through *4* is clothoids, while *5* is a straight line.



Case B: For a medium d, the two clothoids *1* and *2* and possibly a circular arc *3* form the U-turn.



Case C: When d is sufficiently small, the vehicle must turn in opposite direction first to be able to feasibly converge to the next straight line. This case consists of the six clothoids enumerated *1-6* and possibly a circular arc *7*.



Preliminaries

To make the text more readable and ease the reference, central equations defined in earlier chapters will now be restated together with some additional equations.

A clothoid with positive curvature according to the right-hand rule is defined

by (3.36)

$$\boldsymbol{p}(\vartheta) \;=\; a \begin{bmatrix} C(\vartheta) \\ S(\vartheta) \end{bmatrix}, \vartheta \geq 0, \tag{5.149}$$

and by switching the signum of the y-coordinate, a clothoid that turns the other way is obtained.

The clothoid scaling a can be written in terms of the angular acceleration by rearranging (3.44)

$$a = U \frac{1}{\sqrt{2\dot{\omega}(U)}} \tag{5.150}$$

The osculating circle radius of a point on the clothoid is the inverse of (3.38):

$$R(\vartheta) \;=\; \frac{a}{\sqrt{\vartheta}}, \tag{5.151}$$

and its center is (3.39)

$$\boldsymbol{m}(\vartheta) \;=\; \boldsymbol{p}(\vartheta) + \frac{1}{\kappa(\vartheta)} \begin{bmatrix} -\sin(\vartheta) \\ \cos(\vartheta) \end{bmatrix}. \tag{5.152}$$

It should be mentioned that this equation is only valid for a clothoid on its basic form, that is, not rotated across either the x-axis or the y-axis.

The smallest feasible osculating circle for a given speed is (5.41)

$$R_{\min}(U) \;=\; \frac{U}{\omega_{\max}(U)}. \tag{5.153}$$

The derivative of the Fresnel integrals can be found from (3.37)

$$C'(\vartheta) \;=\; \frac{\cos(\vartheta)}{\sqrt{\vartheta}}, \tag{5.154a}$$

$$S'(\vartheta) \;=\; \frac{\sin(\vartheta)}{\sqrt{\vartheta}}. \tag{5.154b}$$

The angle between a vector $\boldsymbol{P} = [P_x,\, P_y]^\top \in \mathbb{R}^2$ and the x-axis is

$$\psi \triangleq \operatorname{atan2}(P_y, P_x), \in \langle -\pi,\, \pi]\,. \tag{5.155}$$

Rotating a vector $\boldsymbol{P} \in \mathbb{R}^2$ an angle ψ according to the right-hand rule in the xy-plane is achieved with a modified version of the rotation matrix (4.7):

$$\boldsymbol{R}(\psi) \triangleq \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}. \tag{5.156}$$

Basic parameterizations The parameterization of basic paths such as straight lines and circles can be found in (Breivik & Fossen 2009). They are restated here for convenience.

A straight line can be parameterized as

$$\boldsymbol{p}(\varpi) \;=\; \boldsymbol{P} + L\varpi \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix}, \varpi \in [0,\, 1]\,, \tag{5.157}$$

where $\boldsymbol{P}$ is the initial point, L is the length of the line, and α is the angle of the line with respect to the x-axis.

A circular arc can be parameterized as

$$\boldsymbol{p}(\varpi) \;=\; \boldsymbol{M}_c + R \begin{bmatrix} \cos(\varpi + \psi) \\ \sin(\varpi + \psi) \end{bmatrix}, \varpi \in [\varpi_0,\, \varpi_1]\,, \tag{5.158}$$

where $\boldsymbol{M}_c$ is the circle center and R is the radius. Further, the position vector $\boldsymbol{p}(\varpi)|_{\varpi=0}$ creates an angle ψ with respect to the x-axis.

Lawn-mower pattern, Case A

Let $\boldsymbol{P}_0, \cdots, \boldsymbol{P}_3$ be two parallel lines valid for Case A. The path starts in $\boldsymbol{P}_0$ and has an angle ψ with respect to the x-axis, see Figure 5.29.

Scaling To obtain a 90° turn using two clothoids, the common angle of the clothoids must be $\pi/4$. At this point, often defined as *apex*, the curvature is at its largest. To verify that the curve is feasible, the curvature must be checked at apex. Choosing the clothoid scaling as small as possible yields the shortest path. First, the smallest possible scaling is calculated by maximizing the angular acceleration in (5.150)

$$a_{\min} = U\frac{1}{\sqrt{2\dot{\omega}_{\max}(U)}}. \tag{5.159}$$

Next, if the inequality

$$R_{\text{apex}}(\pi/4; a_{\min}) \geq R_{\min}(U), \tag{5.160}$$

is invalid, the osculating circle is smaller than the smallest possible circle $R_{\min}$ for that particular speed. Hence, the scaling must be increased to successfully obey this constraint. Algorithm 2 summarize the determination of the scaling.

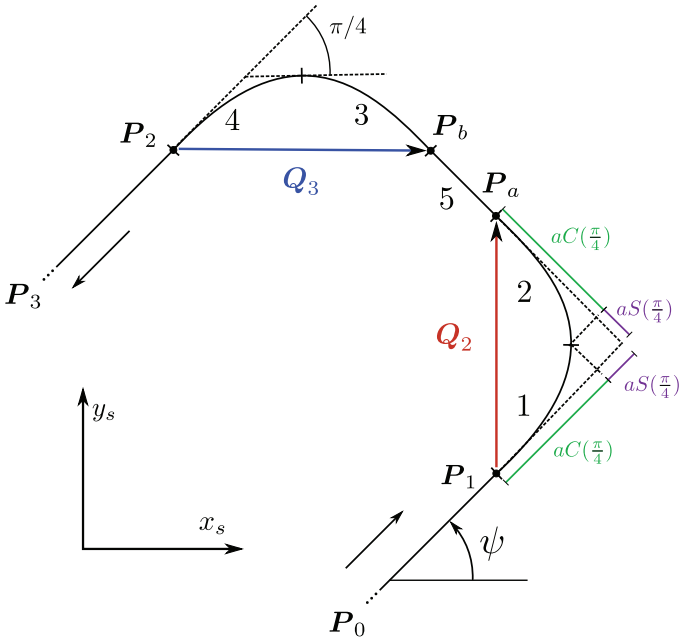


Figure 5.29: The Case A U-turn for lawn-mower patterns with different quantities.

Algorithm 2 Clothoid scaling a for lawn-mower path planning; Case A

if $R_{\text{apex}} < R_{\min}(U)$, **then**
  $a \leftarrow R_{\min}(U)\frac{\sqrt{\pi}}{2}$
else
  $a \leftarrow a_{\min}$
end if

Turning direction To determine the shape of each clothoid, one can consider the vector $\overrightarrow{\boldsymbol{P_0P_2}} \triangleq \boldsymbol{P}_{\text{dir}}$. If this vector is rotated an angle $-\psi$, the signum om the y-coordinate can be used to determine the shape of all the clothoids:

$$\begin{bmatrix} P_{\text{dir,x}} \\ P_{\text{dir,y}} \end{bmatrix} = \boldsymbol{R}^\top(\psi)\boldsymbol{P}_{\text{dir}}, \tag{5.161}$$

$$d_y \triangleq \text{sgn}(P_{\text{dir,y}}). \tag{5.162}$$

After inspecting Figure 5.29, we can conclude that the clothoids *1* and *3* always have the same shape, and likewise for *2* and *4*. Thus, to get the desired clothoid shapes, d_y can be used to determine the signum of y-coordinate of the clothoids.

Translation of a clothoid's origin To create the path, the clothoids must be translated to the correct positions. The origin of the clothoids *1* and *4* have their origin in $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, respectively. As for the two remaining clothoids, some additional considerations must be done. The vectors $\boldsymbol{Q}_2$ and $\boldsymbol{Q}_3$ in Figure 5.29 can determine the origin of clothoid *2* and *3* relative $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, respectively. We get

$$\boldsymbol{Q}_2 = a\left(S\left(\frac{\pi}{4}\right) + C\left(\frac{\pi}{4}\right)\right)\begin{bmatrix} 1 \\ d_y \end{bmatrix}, \tag{5.163a}$$

$$\boldsymbol{P}_a = \boldsymbol{P}_1 + \boldsymbol{R}(\psi)\boldsymbol{Q}_2, \tag{5.163b}$$

and

$$\boldsymbol{Q}_3 = a\left(S\left(\frac{\pi}{4}\right) + C\left(\frac{\pi}{4}\right)\right)\begin{bmatrix} 1 \\ -d_y \end{bmatrix}, \tag{5.164a}$$

$$\boldsymbol{P}_b = \boldsymbol{P}_2 + \boldsymbol{R}(\psi)\boldsymbol{Q}_3. \tag{5.164b}$$

The straight line The straight line that connects clothoid *2* and *3* is simply

$$\mathcal{P}_{III} = \left\{\boldsymbol{p}_{A,5}(\varpi) = \boldsymbol{P}_a + L\varpi\begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix},\ \varpi \in [0,\ ]\right\}, \tag{5.165}$$

where Ł $= |\boldsymbol{P}_b - \boldsymbol{P}_a|$ and $\alpha = \text{atan2}(P_{b,y} - P_{a,y}, P_{b,x} - P_{a,x})$.

The piecewise path for the U-turn We now have enough information to construct the resulting piecewise curve that connects the parallel lines. The following parameterizations form the clothoids *1* to *4*.

$$\mathcal{P}_I = \left\{ \boldsymbol{p}_{A,1}(\varpi) = \boldsymbol{P}_1 + a\boldsymbol{R}(\psi) \begin{bmatrix} C(\varpi) \\ d_y S(\varpi) \end{bmatrix}, \varpi \in \left[0, \frac{\pi}{4}\right] \right\}, \quad (5.166a)$$

$$\mathcal{P}_{II} = \left\{ \boldsymbol{p}_{A,2}(\varpi) = \boldsymbol{P}_a + a\boldsymbol{R}\left(\psi - d_y\frac{\pi}{2}\right) \begin{bmatrix} C(\varpi) \\ d_y S(\varpi) \end{bmatrix}, \varpi \in \left[\frac{\pi}{4}, 0\right] \right\}, \quad (5.166b)$$

$$\mathcal{P}_{IV} = \left\{ \boldsymbol{p}_{A,3}(\varpi) = \boldsymbol{P}_b + a\boldsymbol{R}\left(\psi + d_y\frac{\pi}{2}\right) \begin{bmatrix} C(\varpi) \\ d_y S(\varpi) \end{bmatrix}, \varpi \in \left[0, \frac{\pi}{4}\right] \right\}, \quad (5.166c)$$

$$\mathcal{P}_V = \left\{ \boldsymbol{p}_{A,4}(\varpi) = \boldsymbol{P}_2 + a\boldsymbol{R}(\psi) \begin{bmatrix} C(\varpi) \\ -d_y S(\varpi) \end{bmatrix}, \varpi \in \left[\frac{\pi}{4}, 0\right] \right\}, \quad (5.166d)$$

The resulting piecewise path for the U-turn is thus

$$\mathcal{P}_A = \bigcup_{i=I}^{V} \mathcal{P}_i. \quad (5.167)$$

Validity of Case A When $\boldsymbol{P}_a = \boldsymbol{P}_b$, the vehicle maneuverability constraints just barely allow the vehicle to manage this particular U-turn; if d is reduced only slightly, this case no longer is valid. That is, if

$$\frac{d}{2a} < S\left(\frac{\pi}{4}\right) + C\left(\frac{\pi}{4}\right), \quad (5.168)$$

some other approach must be used.

Lawn-mower pattern, Case B

When Case A no longer is valid, the possibility of a U-turn consisting of two clothoids and a circular arc arises. The circular arc meets the clothoids when the clothoid tangents create an angle ϑ_c with their initial orientation. Figure 5.30 shows the problem at hand. Furthermore, it can be deduced from the same figure that

$$d = 2aS(\vartheta_c) + 2R_c\cos(\vartheta_c), \quad (5.169)$$

where d is the distance between the parallel lines and R_c is the radius of the circular arc. Since R_c also is the osculating circle radius of the clothoids at ϑ_c, we have from (5.151) that

$$R_c = \frac{a}{\sqrt{\vartheta_c}}. \tag{5.170}$$

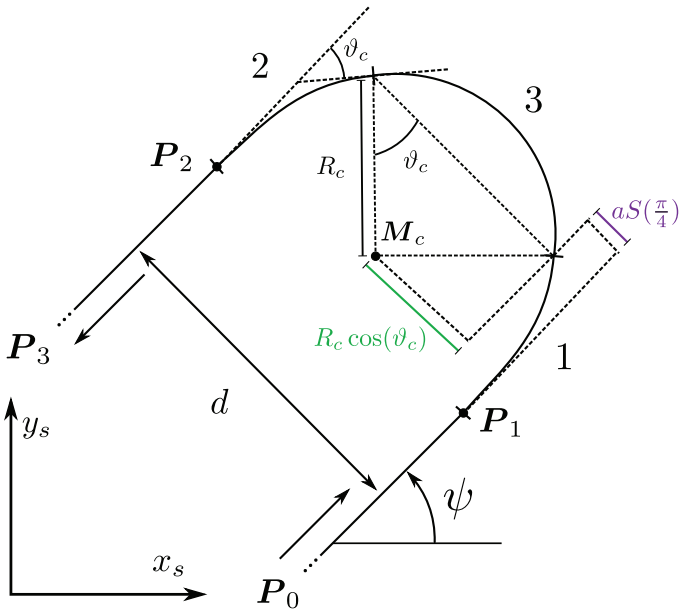


Figure 5.30: The Case B U-turn for lawn-mower patterns with different quantities.

The unknown in (5.169) is ϑ_c, hence we can write the equation as a function of ϑ

$$g_B(\vartheta) = S(\vartheta) + C'(\vartheta) - \frac{d}{2a},\ \vartheta \in \left\langle 0,\ \frac{\pi}{2} \right], \tag{5.171}$$

where $g_B(\vartheta_c) = 0$. To find ϑ_c, a numerical method must be used. For instance, Newton's method with $\vartheta_{c,0} = 10^{-6}$ finds the root with a sufficiently high number of iterations, see Appendix C.10.1.

The scaling of the clothoids is not fully determined beforehand, since the solution of (5.171) dictates the radius R_c. More specifically, for a given scaling a, the following inequality must be valid

$$R_c(\vartheta; a) \geq R_{\min}(U). \tag{5.172}$$

Once more, the optimal solution is choosing the scaling as small as possible according to (5.159). If (5.172) is invalid, a must be increased iteratively and (5.171) solved again, until the inequality is valid. However, by studying Figure 5.30, it can be seen that when

$$d < 2aS\left(\frac{\pi}{2}\right), \tag{5.173}$$

the clothoids are too large to be able to successfully create a feasible U-turn between the two parallel lines.

The circular arc To be able to parameterize the circular arc, its center must be determined. With (5.152) as a starting point, the center of the circle can be defined as

$$\boldsymbol{M}_c \;=\; \boldsymbol{p}_{B,1}(\vartheta_c) + \boldsymbol{R}(\psi)\begin{bmatrix} -\sin(\vartheta_c) \\ d_y\cos(\vartheta_c) \end{bmatrix}, \tag{5.174}$$

where $\boldsymbol{p}_{B,1}(\vartheta)$ is the parameterization of clothoid *1*, which will be defined shortly, and d_y is defined in (5.162).

Since a U-turn involves a 180° change in orientation, and the clothoids produce ϑ_c each, the circle has a orientation change of

$$\Delta\angle_c = \pi - 2\vartheta_c. \tag{5.175}$$

Thus, the circular arc can be parameterized as

$$\mathcal{P}_{II} \;=\; \left\{\boldsymbol{p}_{B,3}(\varpi) = \boldsymbol{M}_c + R_c\begin{bmatrix} \cos(\varpi+\psi) \\ \sin(\varpi+\psi) \end{bmatrix},\, \varpi \in \left[-d_y\frac{\Delta\angle_c}{2},\, d_y\frac{\Delta\angle_c}{2}\right]\right\}. \tag{5.176}$$

The piecewise path for the U-turn The parameterizations for the clothoids *1* and *2* are similar to the clothoids defined by (5.166a) and (5.166d), respectively, with the exception of the validity intervals. This can be verified by comparing the shapes of the corresponding clothoids in the Figures 5.29 and 5.30. Hence, the parameterizations for the clothoids in Case B are defined as

$$\mathcal{P}_I = \left\{\boldsymbol{p}_{B,1}(\varpi) = \boldsymbol{P}_1 + a\boldsymbol{R}(\psi)\begin{bmatrix} C(\varpi) \\ d_yS(\varpi) \end{bmatrix},\, \varpi \in [0,\, \vartheta_c]\right\}, \tag{5.177a}$$

$$\mathcal{P}_{III} = \left\{\boldsymbol{p}_{B,2}(\varpi) = \boldsymbol{P}_2 + a\boldsymbol{R}(\psi)\begin{bmatrix} C(\varpi) \\ -d_yS(\varpi) \end{bmatrix},\, \varpi \in [\vartheta_c,\, 0]\right\}. \tag{5.177b}$$

By combining the parameterizations in correct order, the resulting piecewise path for the U-turn can formally be defined as

$$\mathcal{P}_B = \bigcup_{i=I}^{III} \mathcal{P}_i. \tag{5.178}$$

Validity of Case B As stated in (5.173), this case is not valid if a scaling cannot be found such that the U-turn consists of clothoids that perform at most a $\pi/2$ [rad] change of orientation each. That is, when

$$d < 2aS\left(\frac{\pi}{2}\right), \forall a \in [a_{\min}, \rightarrow\rangle\,, \tag{5.179}$$

a different approach must be used.

Lawn-mower pattern, Case C

When the parallel lines are sufficiently narrow, Case C applies. In this case, the path first turns in the opposite direction of the next desired line. There exist heuristic maneuvers for turning a vessel called man overboard resque turns. For instance, a *Williamson turn* change course to 60° of original course in starboard or port direction before once more changing course (McPhee 2006). This achieves a U-turn converging close to the original straight line. As seen in Figure 5.31, this is not a symmetric path. The proposed solution for Case C is not similar to the Williamson turn. A symmetric solution is proposed, where the initial course change varies with the vehicle parameters and the distance between the parallel lines. This choice makes it easier to construct the path and gives sufficiently good behavior in practise.

The U-turn of Case C is, like the other cases, symmetric about the midline between parallel lines. It consists of a total of six clothoids and a circular arc. Figure 5.32 shows the components of the U-turn, which will be explained momentarily.

The clothoids *1* and *2* perform a course change of ϕ [rad] with initial and final curvature equal to zero. Each clothoid contributes equally to the course change, and hence the the course has changed $\phi/2$ [rad] at apex. ϕ is unknown and must be calculated. To simplify the discussion, set $\psi = \pi/2$ [rad] temporarily. The vector $\boldsymbol{Q}_3$ can then be determined by inspecting Figure

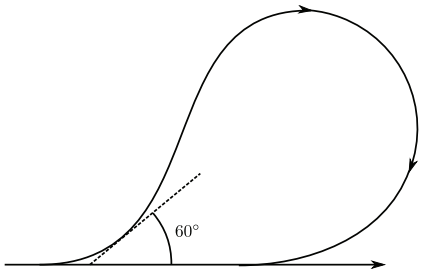


Figure 5.31: The Williamson man overboard resque turn.

5.32. After some algebraic simplifications, we get

$$\boldsymbol{Q}_3 = \underbrace{a\begin{bmatrix} S(\phi/2) \\ C(\phi/2) \end{bmatrix}}_{\text{Clothoid } \textit{1}} + \underbrace{a\boldsymbol{R}(\pi/2-\phi)\begin{bmatrix} C(\phi/2) \\ S(\phi/2) \end{bmatrix}}_{\text{Clothoid } \textit{2}}, \tag{5.180}$$

which makes us able to determine the origin of clothoid *3*:

$$\boldsymbol{P}_a = \boldsymbol{P}_1 + \boldsymbol{Q}_3. \tag{5.181}$$

Let the angle between clothoid *3*'s initial tangent vector and the meeting point with the circle be denoted ϑ_c, which we for the moment assume is known. This position can be determined relative $\boldsymbol{Q}_3$ as

$$\boldsymbol{Q}_c = a\boldsymbol{R}(\pi/2-\phi)\begin{bmatrix} C(\vartheta_c) \\ S(\vartheta_c) \end{bmatrix}, \tag{5.182}$$

see Figure 5.32. Thus, the osculating circle center with radius R_c is

$$\boldsymbol{M}_c = \boldsymbol{P}_a + \boldsymbol{Q}_c + R_c\boldsymbol{R}(\pi/2-\phi)\begin{bmatrix} -\sin(\vartheta_c) \\ \cos(\vartheta_c) \end{bmatrix}, \tag{5.183}$$

for $\psi = \pi/2$ [rad].

Since the path is symmetric about the midline between the parallel lines, we know that the x-coordinate of $\boldsymbol{M}_c$ must be

$$M_{c,x} \stackrel{!}{=} P_{1,x} - \frac{d}{2}. \tag{5.184}$$

By writing out the x-coordinate of (5.183) and simplify, we get

$$\begin{aligned} M_{c,x}(\phi) = {} & P_{1,x} + a\left[S(\phi/2) + \sin(\phi)\left(C(\phi/2) + C(\vartheta_c) - S'(\vartheta_c)\right)\right. \\ & \left. - \cos(\phi)\left(S(\phi/2) + S(\vartheta_c) + C'(\vartheta_c)\right)\right], \end{aligned} \tag{5.185}$$

where we have used that $R_c = a/\sqrt{\vartheta_c}$.

To determine ϕ, equating (5.184) and (5.185) and solving for ϕ would do the trick, except that the equation is implicit and must be solved numerically. We define the root function from the same equations:

$$g_C(\phi) = \frac{M_{c,x}(\phi) - P_{1,x}}{a} = [S(\phi/2) + \sin(\phi)\,(C(\phi/2) + C(\vartheta_c) - S'(\vartheta_c)) - \cos(\phi)\,(S(\phi/2) + S(\vartheta_c) + C'(\vartheta_c))] + \frac{d}{2a}, \tag{5.186}$$

which can be solved using Newton's method with $\phi_0 = 10^{-6}$, see Appendix C.10.2.

Until now, we have only considered the geometrical aspects of finding a feasible path. The next objective is to incorporate the vehicle maneuverability constraints such that a, ϑ_c and ϕ can be determined.

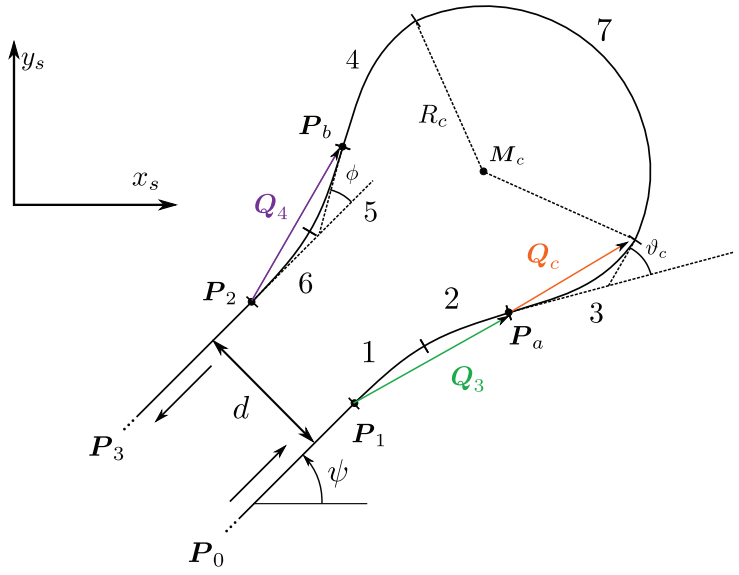


Figure 5.32: The Case C U-turn for lawn-mower patterns with different quantities.

The path parameters

Scaling parameter a Similar to the other cases, we wish to minimize the scaling because this will produce the shortest path. Hence, from (5.159) the

desired clothoid scaling is

$$a_{\min} = U\frac{1}{\sqrt{2\dot{\omega}_{\max}(U)}}. \tag{5.187}$$

A simplification of the design is to force the scaling to be equal for all the clothoids. In some cases it would be beneficial to only manipulate the scaling of the clothoid *3* and *4*, but this will not be done here, due to the increased complexity this choice implies.

The angle ϑ_c If (5.187) holds, combining (5.151), (5.153) and (5.187) and solving for ϑ gives

$$\vartheta_{\max} = \frac{\omega_{\max}(U)^2}{2\dot{\omega}_{\max}(U)}, \tag{5.188}$$

which is the largest angle the tangent vectors of the clothoids *3* and *4* can create with their initial tangent vectors. We further limit the angle such that

$$\vartheta_c = \min\left(\vartheta_{\max}, \pi\right), \tag{5.189}$$

which is the nominal choice when determining ϕ. Later, we will further restrict ϑ_c in order to avoid undesired paths.

Course change angle ϕ A course change of ϕ [rad] is equally shared between two clothoids and as mentioned earlier, the angle is $\phi/2$ at apex, implying that the maximum course change is twice $\vartheta_{\max}$. Furthermore, we restrict the course change to $\pi/2$ and thus the maximum course change is

$$\phi_{\max} = \min\left(2\vartheta_{\max}, \pi/2\right). \tag{5.190}$$

Further restriction to ϑ_c The U-turn involves a total orientation change of 180°. The circular arc accounts for

$$\Delta\angle_c = \frac{\pi}{2} - \left|-\phi + \vartheta_c\right|, \tag{5.191}$$

which implies that

$$-\phi + \vartheta_c \leq \frac{\pi}{2}. \tag{5.192}$$

When $\vartheta_c > \pi/2$, (5.192) does not necessarily apply. Consequently, when ϕ has been found, a verification of this inequality must be performed. If it is invalid, ϑ_c must be reduced. The method is summarized in Algorithm 3.

Algorithm 3 Iterative reduction of ϑ_c; Case C

```
if ϑ_c > π/2 then
  while ϑ_c − φ > π/2 do
    ϑ_c ← ϑ_c − α, α > 0
    Find φ s.t. g_C(φ; ϑ_c) = 0, using Newton's algorithm
  end while
end if
```

Final determination of the scaling a For a given scaling, the solution of (5.186) yields a feasible candidate for ϕ. As long as

$$\phi \leq \phi_{\max}, \tag{5.193}$$

the course changing clothoids do not violate the angular speed constraint. In the opposite case, several things can be done. Either augment a straight line between clothoid *2* & *3*, and *4* & *5*, or increasing the scaling a, such that (5.193) becomes valid. The latter represents the simplest choice, even though it possibly is less optimal than the former.

The x-coordinate of the circle center (5.183) lies to the left of the initial parallel line (in the nominal case used to deduce the equations) and hence

$$M^{\star}_{c,x}(\phi, \vartheta_c) \triangleq g_c(\phi) - \frac{d}{2a} \leq 0, \, \forall \, \mathrm{col}\,[\phi, \vartheta_c] \in \mathcal{I}_C. \tag{5.194}$$

The set $\mathcal{I}_C$ is the blue/green area (and the red area) of Figure (5.33). The red area is the portion which is infeasible due to (5.190), while the yellow/orange area never occurs, since the the distance d must be negative for it to apply. We identify the red region by setting $\phi = 2\vartheta_c$ and finding the zero of 5.194, which yields the critical point

$$\vartheta_{c,k} \approx 0.368. \tag{5.195}$$

To guarantee that ϕ does not violate the angular speed constraint, the scaling a must be iteratively increased such that (5.193) is satisfied. Algorithm 4 describes the necessary calculation to make ϕ feasible.

Translation of the clothoid's origins The origin of the clothoids *1* and *6* are the waypoints $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, respectively. By consulting Figure 5.32, it

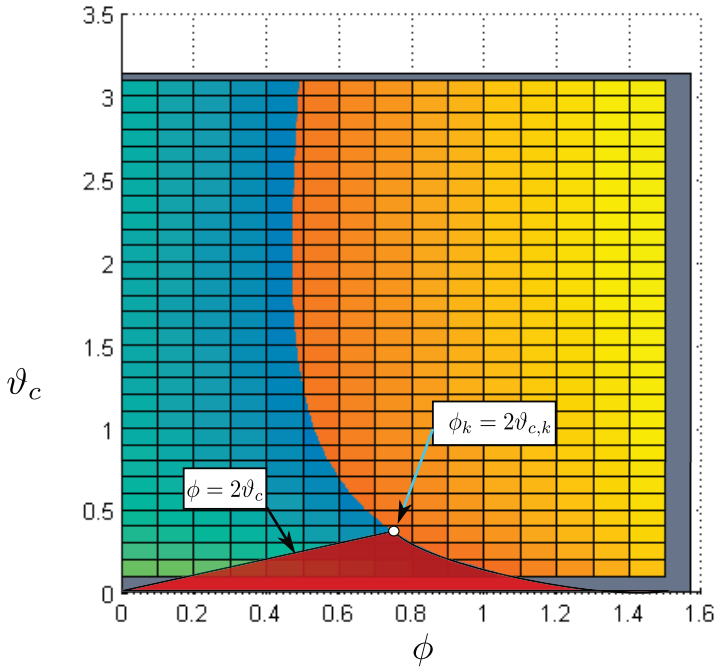


Figure 5.33: The blue/green area on the left is the feasible set for ψ and ϑ_c. The red area is infeasible due to the angular speed constraint.

Algorithm 4 Iterative increasement of a when $\phi > \phi_{\max}$; Case C

while $\vartheta_c < \vartheta_{c,k}$ **do**
  if $\phi > \phi_{\max}$ **then**
    $a \leftarrow a + \alpha$, $\alpha > 0$
    $\vartheta_c \leftarrow \frac{a^2}{R_{\min}}$
    Find ϕ *s.t.* $g_C(\phi; \vartheta_c, a) = 0$, using Newton's algorithm
  end if
end while

can be seen that the vectors $\boldsymbol{Q}_3$ and $\boldsymbol{Q}_4$ are

$$\boldsymbol{Q}_3 = a \begin{bmatrix} S(\phi/2) \\ C(\phi/2) \end{bmatrix} a\boldsymbol{R}(\pi/2 - \phi) \begin{bmatrix} C(\phi/2) \\ S(\phi/2) \end{bmatrix}, \tag{5.196}$$

$$\boldsymbol{Q}_4 = \begin{bmatrix} -Q_{3,x} \\ Q_{3,y} \end{bmatrix} \tag{5.197}$$

If the restriction $\psi = \pi/2$ is released, the vectors that determine the origin of the clothoids *2* & *3*, and *4* & *5* can be found. We get

$$\boldsymbol{P}_a = \boldsymbol{P}_1 + \boldsymbol{R}(\psi - \pi/2) \begin{bmatrix} d_y Q_{3,x} \\ Q_{3,y} \end{bmatrix}, \tag{5.198}$$

$$\boldsymbol{P}_b = \boldsymbol{P}_2 + \boldsymbol{R}(\psi - \pi/2)\boldsymbol{Q}_4, \tag{5.199}$$

where d_y is defined in (5.162).

The circular arc In a similar manner as for (5.174), the center of the circular arc can be defined as

$$\boldsymbol{M}_c = \boldsymbol{p}_{C,1}(\vartheta_c) + \boldsymbol{R}(\psi - d_y\phi) \begin{bmatrix} -\sin(\vartheta_c) \\ d_y \cos(\vartheta_c) \end{bmatrix}, \tag{5.200}$$

where $\boldsymbol{p}_{C,1}(\vartheta)$ is the parameterization of clothoid *3*, which will be defined very soon.

The parameter span for the circle is

$$\mathcal{I}_o = [-d_y\Delta\angle_c,\, d_y\Delta\angle_c], \tag{5.201}$$

where $\Delta\angle_c$ was defined in (5.191).

The parameterization for the circular arc is thus

$$\mathcal{P}_{IV} = \left\{ \boldsymbol{p}_{C,7}(\varpi) = \boldsymbol{M}_c + R_c \begin{bmatrix} \cos(\varpi + \psi) \\ \sin(\varpi + \psi) \end{bmatrix}, \varpi \in \mathcal{I}_o \right\}, \tag{5.202}$$

where R_c is the circular arc radius calculated using the final scaling a.

The piecewise path for the U-turn We now possess sufficient knowledge about the problem to construct the parameterizations. The clothoids are

parameterized in a similar manner as the clothoids in Cases A and B, so no explanation will be given. We get

$$\mathcal{P}_I = \left\{ \boldsymbol{p}_{C,1}(\varpi) = \boldsymbol{P}_1 + a\boldsymbol{R}(\psi) \begin{bmatrix} C(\varpi) \\ -d_y S(\varpi) \end{bmatrix}, \varpi \in \left[0, \frac{\phi}{2}\right] \right\}, \quad (5.203a)$$

$$\mathcal{P}_{II} = \left\{ \boldsymbol{p}_{C,2}(\varpi) = \boldsymbol{P}_a - a\boldsymbol{R}\left(\psi - d_y\phi\right) \begin{bmatrix} C(\varpi) \\ d_y S(\varpi) \end{bmatrix}, \varpi \in \left[\frac{\phi}{2}, 0\right] \right\}, \quad (5.203b)$$

$$\mathcal{P}_{III} = \left\{ \boldsymbol{p}_{C,3}(\varpi) = \boldsymbol{P}_a + a\boldsymbol{R}\left(\psi - d_y\phi\right) \begin{bmatrix} C(\varpi) \\ d_y S(\varpi) \end{bmatrix}, \varpi \in [0, \vartheta_c] \right\}, \quad (5.203c)$$

$$\mathcal{P}_V = \left\{ \boldsymbol{p}_{C,4}(\varpi) = \boldsymbol{P}_b + a\boldsymbol{R}\left(\psi + d_y\frac{\pi}{2}\right) \begin{bmatrix} C(\varpi) \\ -d_y S(\varpi) \end{bmatrix}, \varpi \in [\vartheta_c, 0] \right\}, \quad (5.203d)$$

$$\mathcal{P}_{VI} = \left\{ \boldsymbol{p}_{C,5}(\varpi) = \boldsymbol{P}_b + a\boldsymbol{R}\left(\psi + d_y\frac{\pi}{2}\right) \begin{bmatrix} -C(\varpi) \\ d_y S(\varpi) \end{bmatrix}, \varpi \in \left[0, \frac{\phi}{2}\right] \right\}, \quad (5.203e)$$

$$\mathcal{P}_{VII} = \left\{ \boldsymbol{p}_{C,6}(\varpi) = \boldsymbol{P}_2 + a\boldsymbol{R}(\psi) \begin{bmatrix} C(\varpi) \\ d_y S(\varpi) \end{bmatrix}, \varpi \in \left[\frac{\phi}{2}, 0\right] \right\}, \quad (5.203f)$$

The piecewise path for the U-turn of Case C is

$$\mathcal{P}_C = \bigcup_{i=I}^{VII} \mathcal{P}_i. \quad (5.204)$$

Whenever Case A and Case B are invalid, Case C applies.

Concluding remarks

The proposed approach connects parallel lines under vehicle maneuverability constraints. The method does not minimize the time of coverage, but opens the possibilities for doing so. The method provides feasible paths for a given speed. By augmenting this approach with algorithms that determine the order of the parallel lines, and speed assignments along the path, an optimal lawn-mower path planner is born.

Chapter 6

Simulation results

This chapter illustrates the proposed algorithms through simulations. First, preliminary information about the setup is given. Then, elaborating information and figures are given, together with discussions and conclusions.

6.1 Preliminaries

6.1.1 The simulation environment

All the simulations are done using the MathWorks™ environment MATLAB® and Simulink®. Furthermore an own package was created using object-oriented MATLAB code. This was done to simplify the construction of the system and making the simulation environment more flexible for changes. A rough UML-diagram of the package can be found in Appendix A. In Appendix B it is given a short manual on how to reproduce the simulations provided in this chapter.

6.1.2 Closed-loop model

The course model

The closed-loop model presented in Section 4.2.2 is used with the parameters shown in Table 6.1. A step response $\chi = \pi/4$ [rad] of the model with $\chi_0 = 0$

[rad] can be seen in Figure 6.1. As expected, the course changes with bounded angular speed. Furthermore, the acceleration changes rapidly and attains large values. This is also expected, since the model is a bounded low-pass filter. This limitation makes the model unrealistic, so a limited acceleration should be incorporated in the model. This extension is considered to be future work and will not be performed in this thesis. Still, the variable lookahead distance is chosen with the assumption that the maximum angular acceleration is $\dot{\omega}_{\max}(U) = 0.1$ [rad/s^2].

Parameter	Value	
$\Delta_{\dot{\chi}}$	1/1000	[rad]
$\omega_{\max}(U)$	0.4	[rad/s]

Table 6.1: Course model parameters.

The forward-speed model

The parameters chosen for the forward-speed model from Section 4.2.2 is presented in Table 6.2. The upper and lower obtainable speed is also given. Two different step responses are also provided. An accelerating step response to $U = 8$ [m/s] from $U_0 = 1$ [m/s] is found in Figure 6.2, while a decelerating step response to $U = 1$ [m/s] from $U_0 = 8$ [m/s] is shown in Figure 6.3. These responses reveal that the acceleration is much slower than the deceleration, which coincides with the expected behavior for a USV. Furthermore, the speed increases rapidly upto about 5 [m/s], and has a more linear quality after that.

6.1.3 Paths

To investigate the behavior of the guidance algorithms, we need several different paths. A total of five paths are used, ranging from basic parameterizations to challenging curved paths. Here, we merely present these paths for ease of reference.

Some of the parameterizations are defined relative the vehicle's initial position. This makes it easier to create desirable initial cross-track errors. The

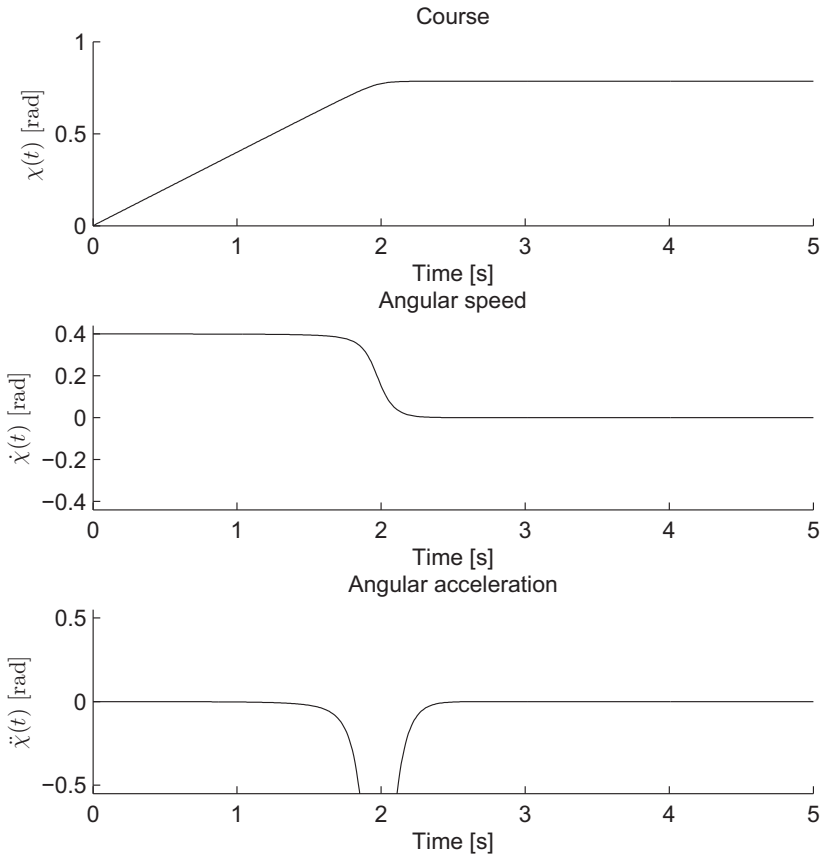


Figure 6.1: Step response of course model.

Parameter	Value	
K	1.5	$[\mathrm{m/s^2}]$
a_{acc}	0.7	[s/m]
$\Delta_{\mathrm{acc,nl}}$	0.1	[m/s]
a_{dec}	0.25	[-]
$\Delta_{\mathrm{dec,nl}}$	5	[m/s]
$\dot{U}_{\mathrm{max,l}}$	0.04	$[\mathrm{m/s^2}]$
Δ_{l}	0.005	[m/s]
U_{min}	1	[m/s]
U_{max}	8.9	[m/s]

Table 6.2: Forward-speed model parameters.

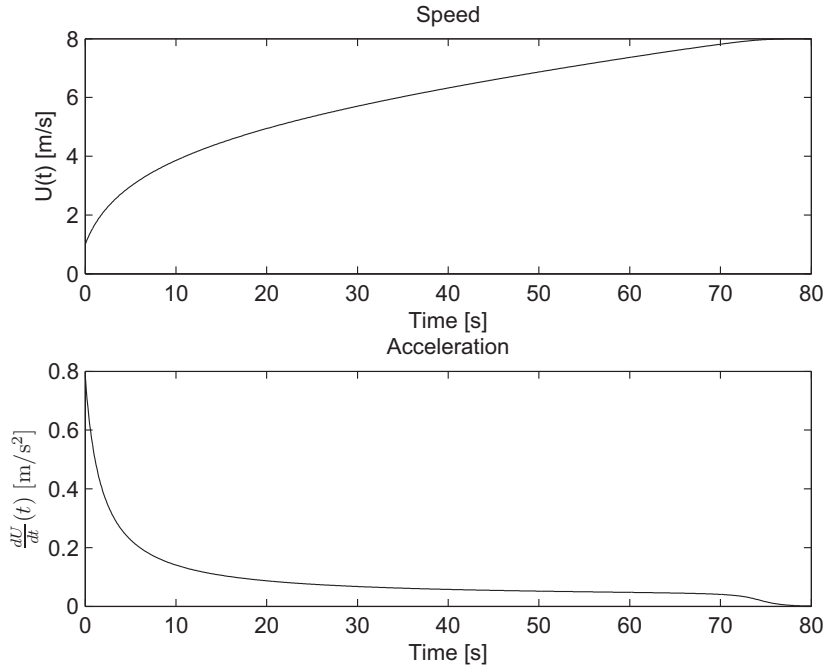


Figure 6.2: Step response of forward-speed model; acceleration.

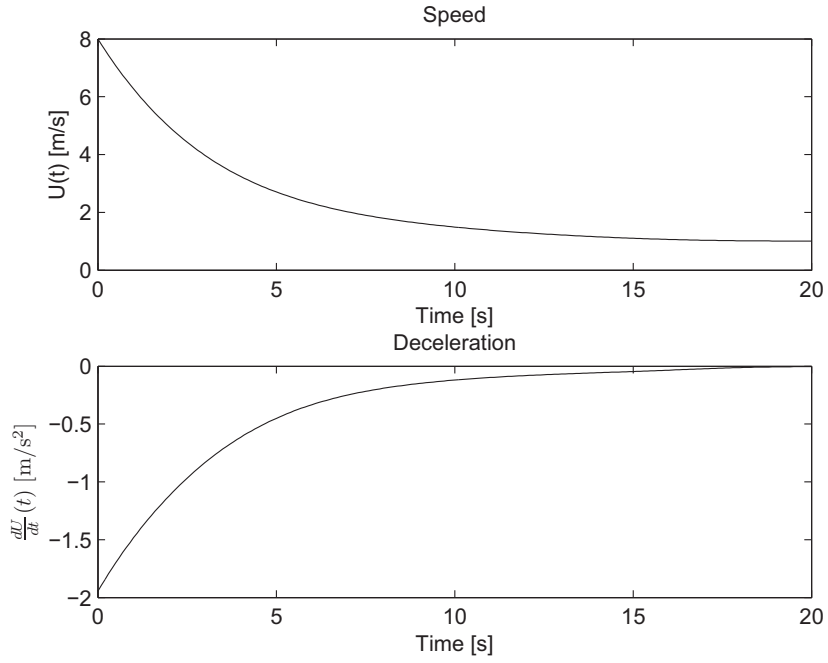


Figure 6.3: Step response of forward-speed model; deceleration.

initial vehicle position is

$$\boldsymbol{p}(0) \;=\; \begin{bmatrix} p_{x,0} \\ p_{y,0} \end{bmatrix}. \tag{6.1}$$

Straight line

The straight line is parameterized by

$$\mathcal{P}_\alpha \;=\; \left\{ \begin{bmatrix} p_{x,0} - 10 \\ p_{y,0} - 50 + 10\varpi \end{bmatrix}, \varpi \in [0,\, 40] \right\}. \tag{6.2}$$

Circular path

The circular path is parameterized by

$$\mathcal{P}_\beta \;=\; \left\{ \begin{bmatrix} p_{x,0} + R\cos(\varpi) \\ p_{y,0} - R\sin(\varpi) \end{bmatrix}, \varpi \in [-2\pi,\, 4\pi] \right\}, \tag{6.3}$$

where $R = 40$ [m]. The direction of the path is in counter-clockwise direction if plotted in a NED-frame.

Curved paths

There are used three different curved paths in this thesis. All of them are created using manually sampled positions from Figure 6.4, which is a plot of a real race track. The positions are used in an implementation of the $\mathcal{G}^2$-continuous Catmull-Rom path generation algorithm presented in Chapter 5.7.4. In order to obtain an approximated maximum curvature, the sampled positions are scaled prior to path generation. Since the paths consist of many curve segments, there is no point in restating them here. Nevertheless, the chosen parts of the track, which are denoted $\mathcal{P}_\gamma$, $\mathcal{P}_\delta$ and $\mathcal{P}_\epsilon$, can be seen in Figure 6.4.

It is worth pointing out that the chosen algorithm does not create feasible paths in general. Even though the paths are $\mathcal{G}^2$ continuous, there are no limitations on what values the curvature can attain. More specifically, if the control points create turns which are impossible in practice, the resulting path will also be impossible to follow. On the other hand, simulations will show that the piecewise path is indeed feasible if the control points are sampled from a feasible path.

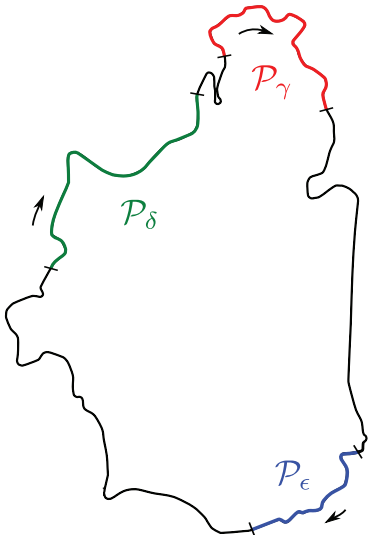


Figure 6.4: Nürburgring's Nordschleife is a motorsport race track in Nürnburg, Germany. It is an example of a predefined path that can be expressed with parameterizations. Courtesy of wikimedia.org.

Common parameters

To avoid singularities, the variables $\Delta_\vartheta = 10^{-6}$ and $\Delta_{\tilde{\chi}} = 10^{-6}$ are used in (5.81) and (5.106). The feedback term of the vehicle projection particle in the steering law (Chapter 5.3.2) is the same in all the simulations: $\gamma = 0.5$.

6.1.4 Simulation cases

The simulations are divided into three main parts. The first part simulates the path speed algorithm presented in Chapter 5.2. The initial conditions and parameters of this simulation are given in Section 6.2.1.

The second part considers path convergence scenarios. The simulation is divided into six cases, and gradually adds functionality and tests the algorithms on more challenging paths. Case I &II are straight-line scenarios, followed by a circular-path scenario in Case III. In Case IV, the response of following a curved path is investigated. Then, the path speed algorithm is combined with the path convergence algorithm and tested against a challenging curved path in simulation Case V. Finally, the path-tracking system is simulated in Case VI. Each case's initial conditions are given in Table 6.3, where the

corresponding paths also are indicated.

The last part illustrates the different paths the lawn-mower path planner creates for different speeds. Furthermore, the method is compared to a simple spline method.

Case	Initial conditions				$U_{\text{ref}}(t)$	Path
	North [m]	East [m]	U_0 [m/s]	χ_0 [rad]		
I	50	10	5	0	U_0	$\mathcal{P}_\alpha$
II	50	10	5	0	$U_{ref}(U_d, \tilde{\chi})$	$\mathcal{P}_\alpha$
III	0	0	5	0	$U_{ref}(U_d, \tilde{\chi})$	$\mathcal{P}_\beta$
IV	500	75	5	0	$U_{ref}(U_d, \tilde{\chi})$	$\mathcal{P}_\delta$
V	530	220	5	0	$U_{ref}(U_d, \tilde{\chi})$	$\mathcal{P}_\gamma$
VI	165	650	2	π	$U_{ref}(U_p(t), \tilde{\chi})$	$\mathcal{P}_\epsilon$

Table 6.3: Case overview.

6.2 Results

6.2.1 Path speed algorithm

In this section we verify that the path speed algorithm presented in Chapter 5.2 yields speed references which obey the angular speed constraint. The path denoted $\mathcal{P}_\gamma$ will be used, since this path is scaled in such way that the the maximum curvature is large. For the moment, we assume that the vehicle is constrained to follow the path. Furthermore, to be able to use the algorithm, some parameters must be determined.

Parameters

The vehicle's maximum acceleration and deceleration can be extracted from the forward speed model presented in Section 4.2.2:

$$f_{\text{acc}}(U) = Ke^{-a_{acc}U} + U_{\text{max,l}}, \tag{6.4a}$$
$$f_{\text{dec}}(U) = a_{dec}U + U_{\text{max,l}}, \tag{6.4b}$$

where the model parameters are given in Table 6.2.

The lookahead distance is chosen to be $\Delta_d = 20$ [m]. Even though this is less than the stopping distance, it is sufficient in this particular example. The initial speed is $U_0 = 5$ [m/s]. The proportinal feedback gain of the lookahead system was set to $k_p = 1$.

Result

Figure 6.5 shows the particular path at hand. The path contains several acute turns, where the speed must be reduced to obey the angular speed constraint. The reference speed along the track is shown in the uppermost plot of Figure 6.6. We can see that the reference speed is lower than, or equal to the curvature constrained speed, which also is shown in the plot. The plot in the middle shows the curvature of the path, and naturally, there is a clear correlation between the curvature and the curvature constrained speed. The lowermost plot of Figure 6.6 displays the error between the desired and the actual along-path lookahead. Since this error has been minimized offline and the lookahead database populated, there is no initial transient phase. The

error is small at all times, within ±0.15 [m], which is sufficiently accurate. The error can be reduced further by increasing the feedback gain k_p or introducing integral action, which may be necessary if the lookahead distance is small.

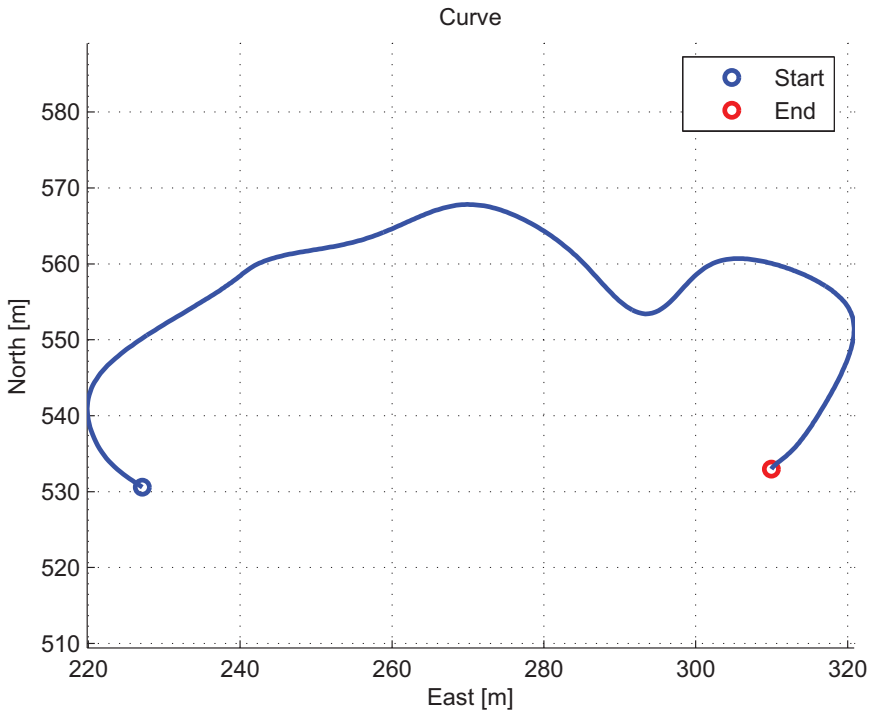


Figure 6.5: Path speed algorithm: Predefined curve $\mathcal{P}_\gamma$.

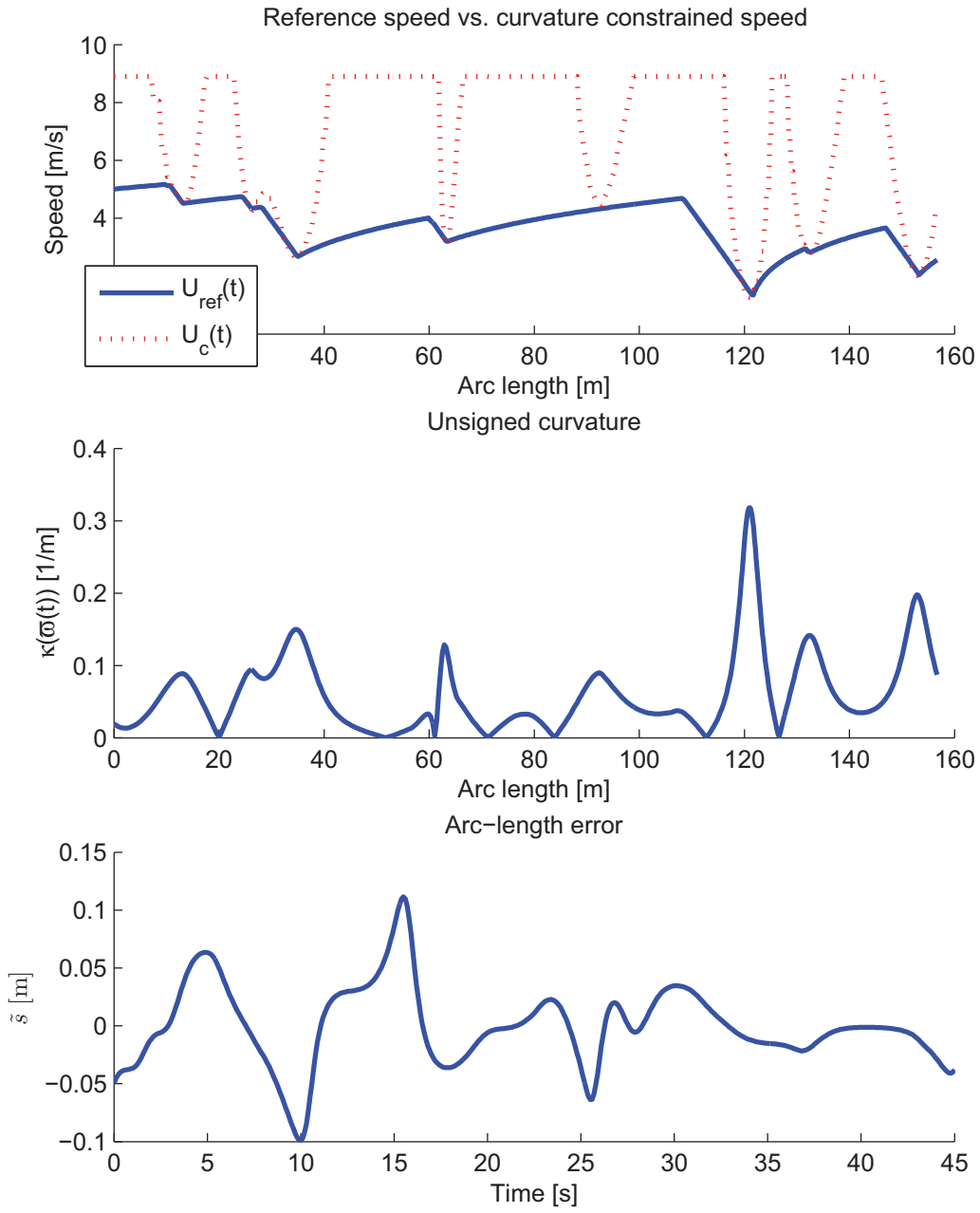


Figure 6.6: Path speed algorithm: *Top*: Reference speed vs. Curvature constrained speed. *Middle*: Unsigned curvature as a function of travelled distance. *Bottom*: Lookahead error.

6.2.2 Case I: Straight-line path convergence

Parameters

The semi-constant lookahead distance is compared to a conservative variant of the variable lookahead distance with $\tilde{\chi}$ heuristics. The expressions for the lookahead distances are given in Table 6.4.

Lookahead distance	Parameters
$\Delta(U) = k_c \Delta_{\text{const}}(U)$	$k_c = 1$
$\Delta(e, \tilde{\chi}, U, t) = \Delta_{\text{var}}(U, e) + \Delta_{\text{const}}(U)(k_c + k_{\tilde{\chi}}\tilde{\chi})$	$k_c = 0.3,\ k_{\tilde{\chi}} = 0.3$

Table 6.4: Lookahead distances for Case I.

Result

The position response can be seen in Figure 6.7. The two different methods display similar responses, where the semi-constant lookahead is slightly faster. This is due to the smaller lookahead distance during the last part of the convergence, as seen in Figure 6.8. Nevertheless, the variable lookahead distance obtains more conservative course accelerations during this phase, and thus does not carelessly ignore the angular acceleration constraint. The course responses are shown in Figure 6.9, which confirm that the course rate has less steep tangents with the variable lookahead approach.

A common drawback of these approaches is the fact that since the angular speed is bounded, a large circular arc is performed before the orientation of the vehicle points toward the straight line. The consequence of this behavior is that the vehicle increases the cross-track error for a while before it starts to decrease. A reduction of the forward speed is a possible counteraction.

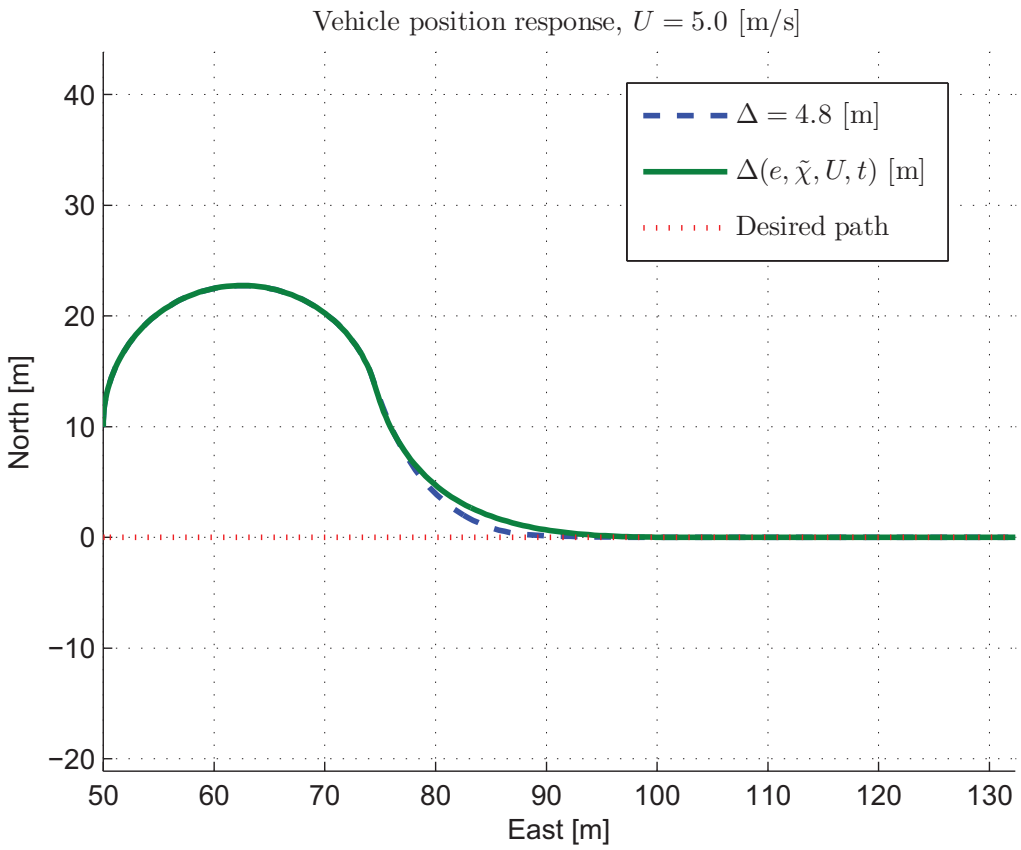


Figure 6.7: Case I: position response.

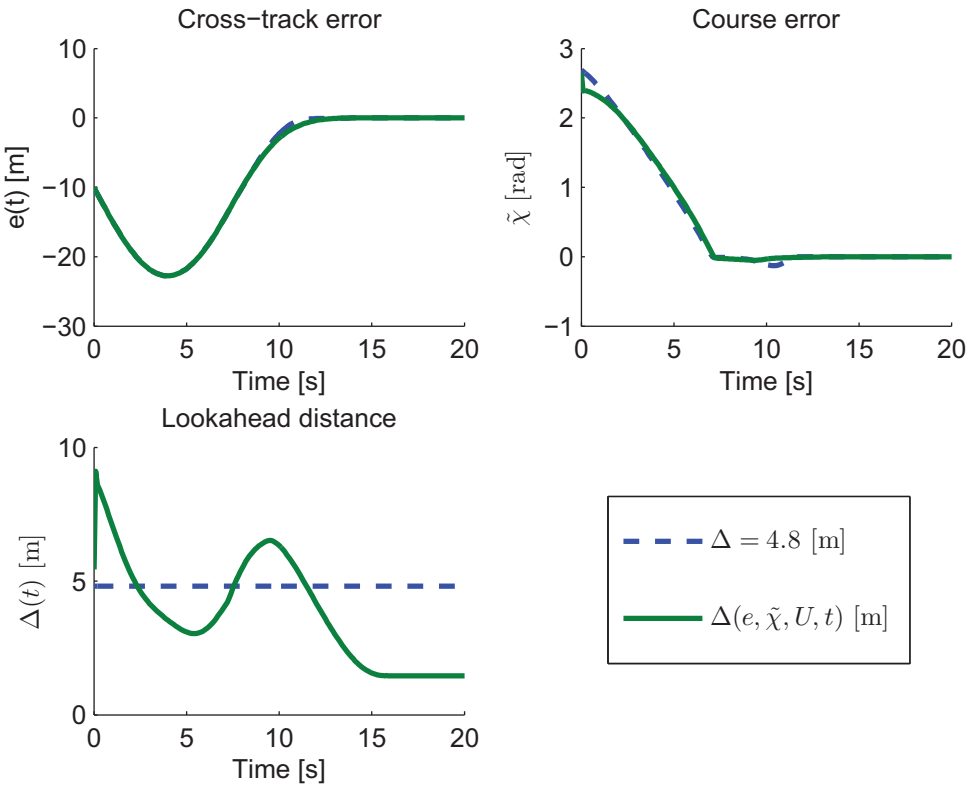


Figure 6.8: Case I: error and lookahead response.

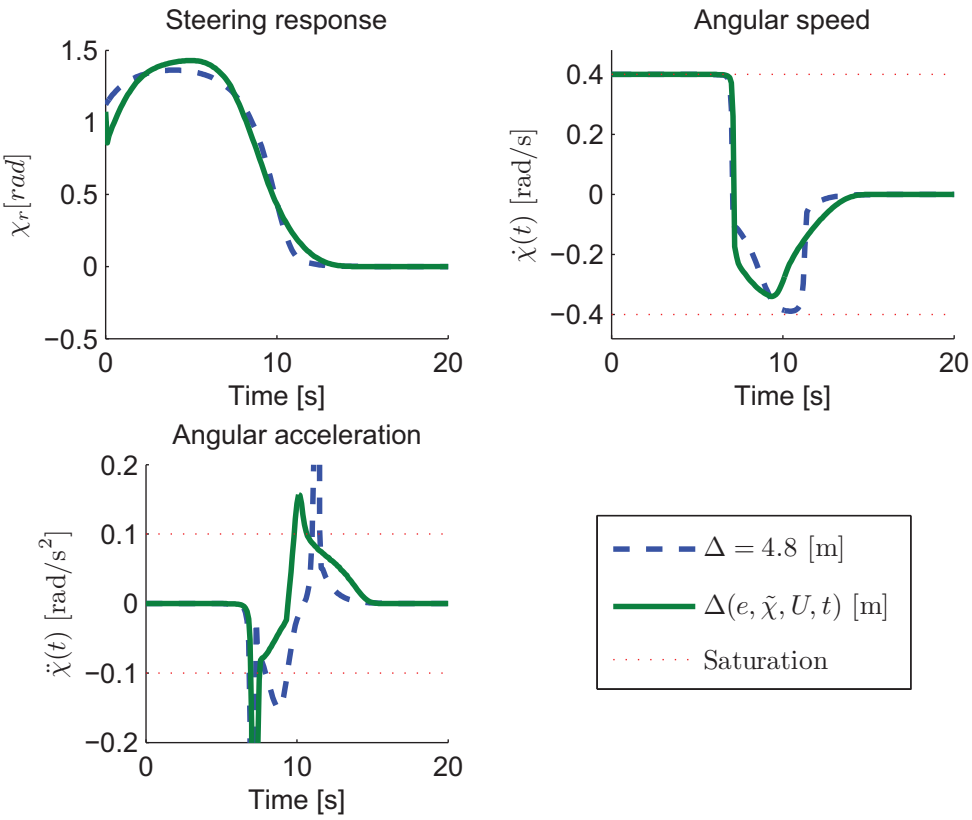


Figure 6.9: Case I: course response.

6.2.3 Case II: Straight-line path convergence with heuristic speed adaptation

This case considers the same scenario as in Case I, but with the addition that an heuristic speed adaptation scheme is employed.

Parameters

The formulas for the lookahead distance have the same parameters as in Case I. Furthermore, the speed adaptation law presented in Chapter 5.6.3 is employed with $k_U = 1$.

Result

The Figures 6.10-6.12 contain the responses for this case. Compared to the previous case, the position response is similar, but with an important distinction; the circular arc is smaller, leading to faster convergence to the straight line. This improved convergence rate is made possible at the expense of the forward speed. As can be seen in Figure 6.11, the speed is reduced to about 3.2 [m/s] before it starts to increase. This is in contrast to the previous case, where the forward speed was kept steady at 5 [m/s]. The aggressiveness of the speed reduction depends on the shaping parameter k_U. If a less aggressive speed reduction is desired, k_U must be reduced. Often, there are considerable energy costs associated with constantly changing the speed. Hence, the parameter must be carefully determined, perhaps even made speed dependent.

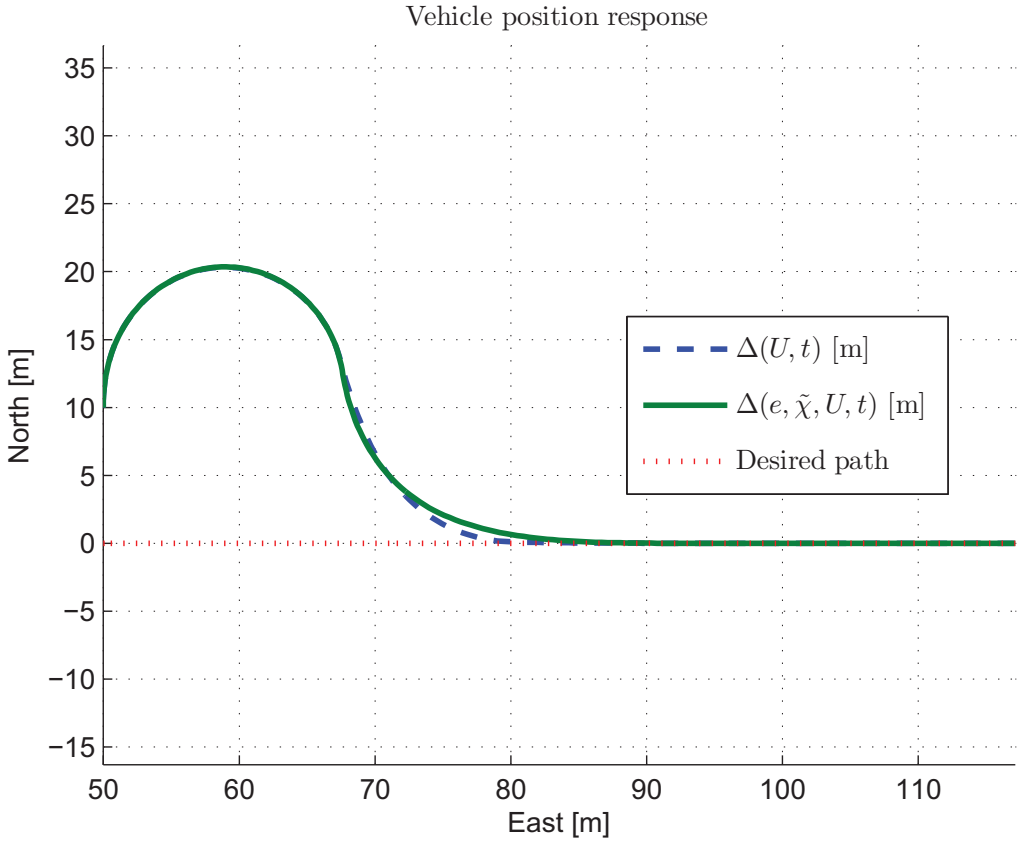


Figure 6.10: Case II: position response.

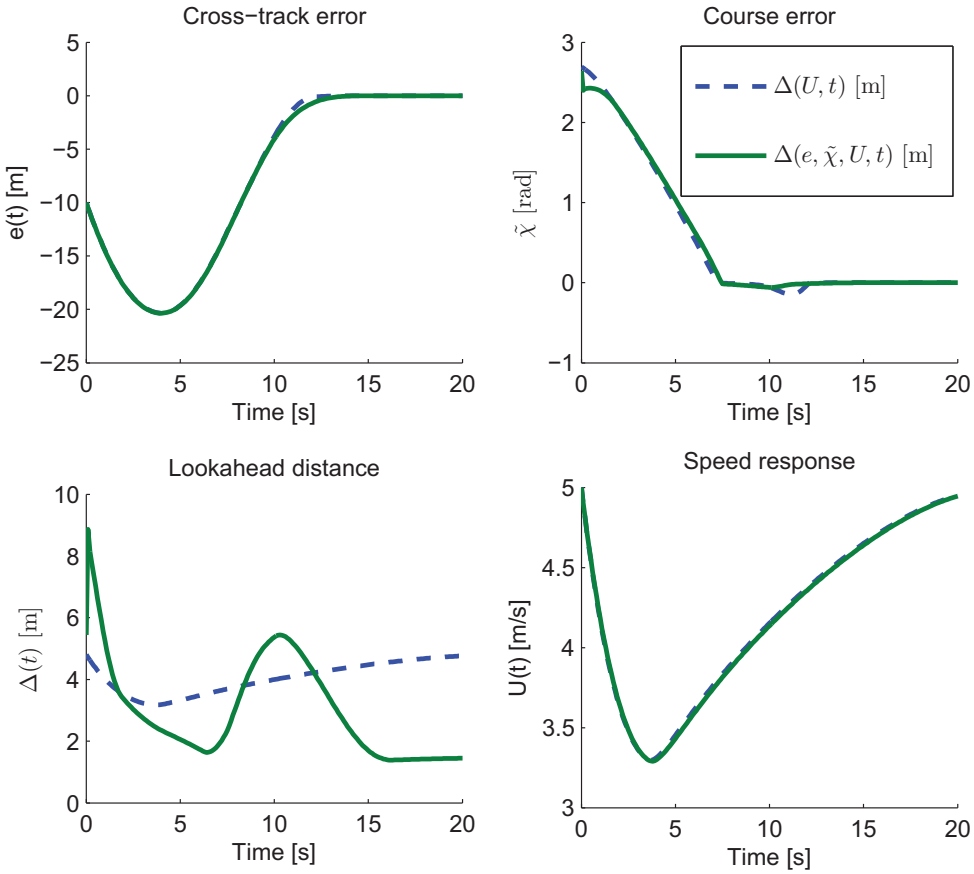


Figure 6.11: Case II: error, lookahead and speed response.

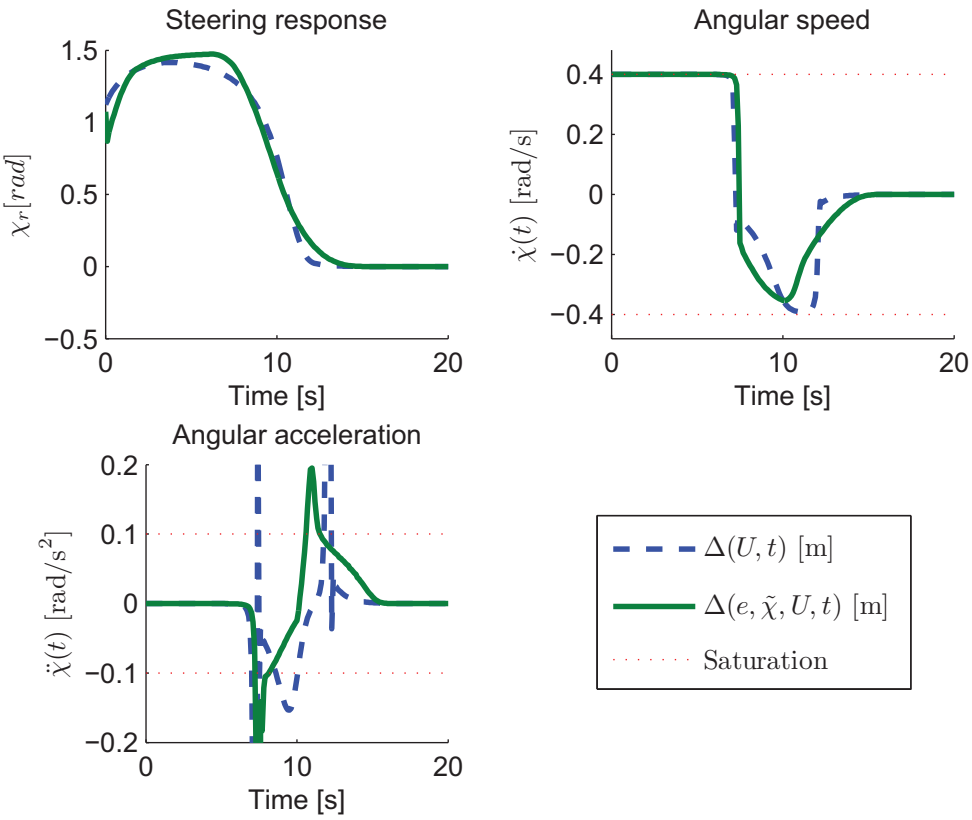


Figure 6.12: Case II: course response.

6.2.4 Case III: Circular convergence with heuristic speed adaptation

Case III considers the convergence behavior when starting in the origin of a circle for which is the desired path. The radius of the circle is 40 [m], and the speed is once more 5 [m/s], as stated in Table 6.3.

Parameters

There are no changes in the parameter values compared to Case II.

Result

Figure 6.13 shows that both methods are able to converge to the circle quite rapidly. The semi-constant lookahead method produces a small overshoot, while the other does not. Furthermore, in Figure 6.14 we can see that the speed reduction for the semi-constant lookahead is more significant than for the variable lookahead, though only 0.15 [m/s]. The difference in the speed reductions stems from the fact that the course error is much larger for the former method. The course-related responses in Figure 6.15 do not exhibit any surprises; the course accelerations are once more nonphysical due to excessive acceleration values.

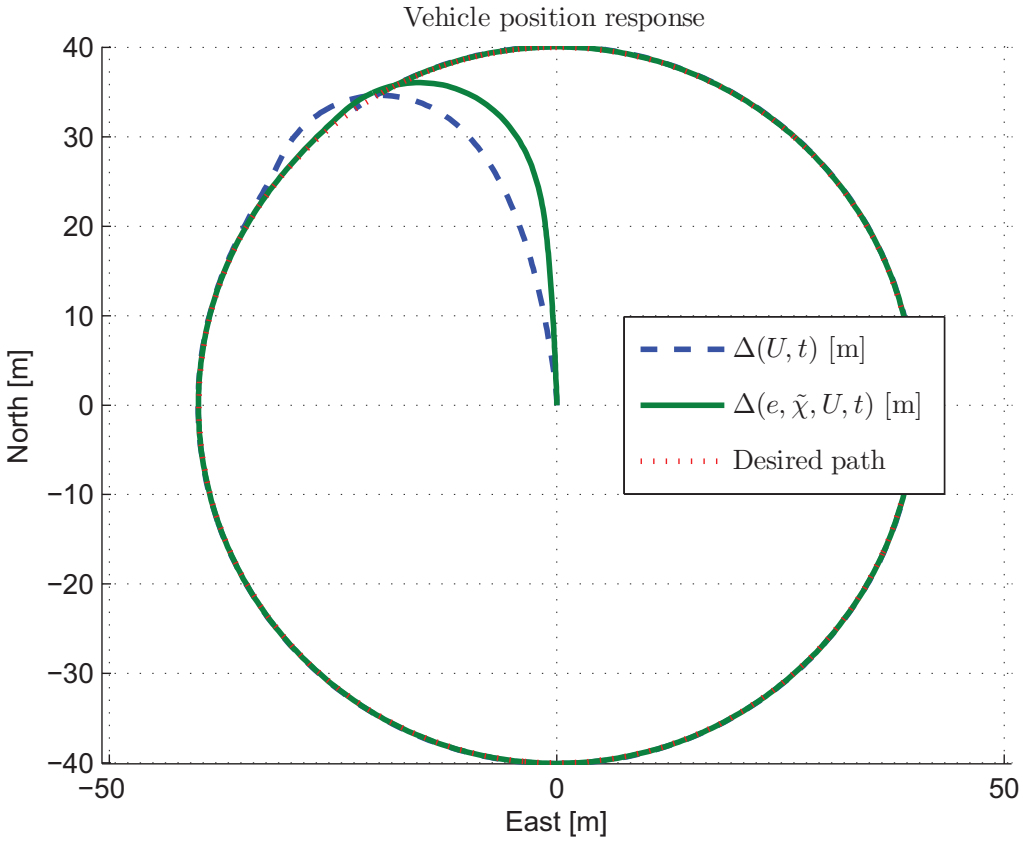


Figure 6.13: Case III: position response.

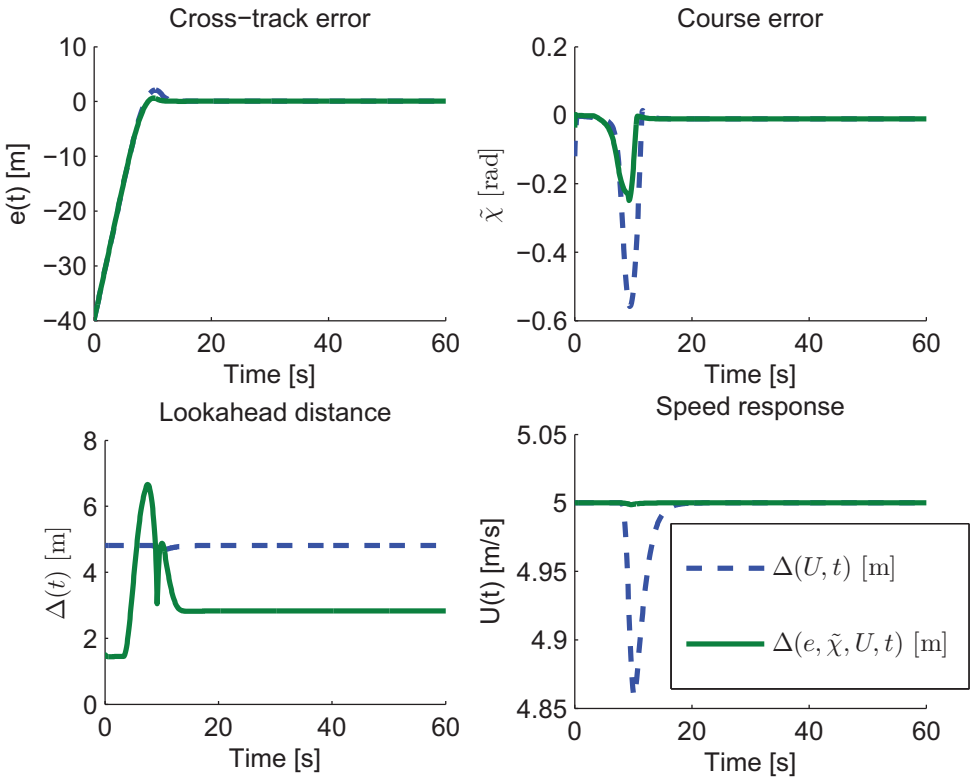


Figure 6.14: Case III: error, lookahead and speed response.

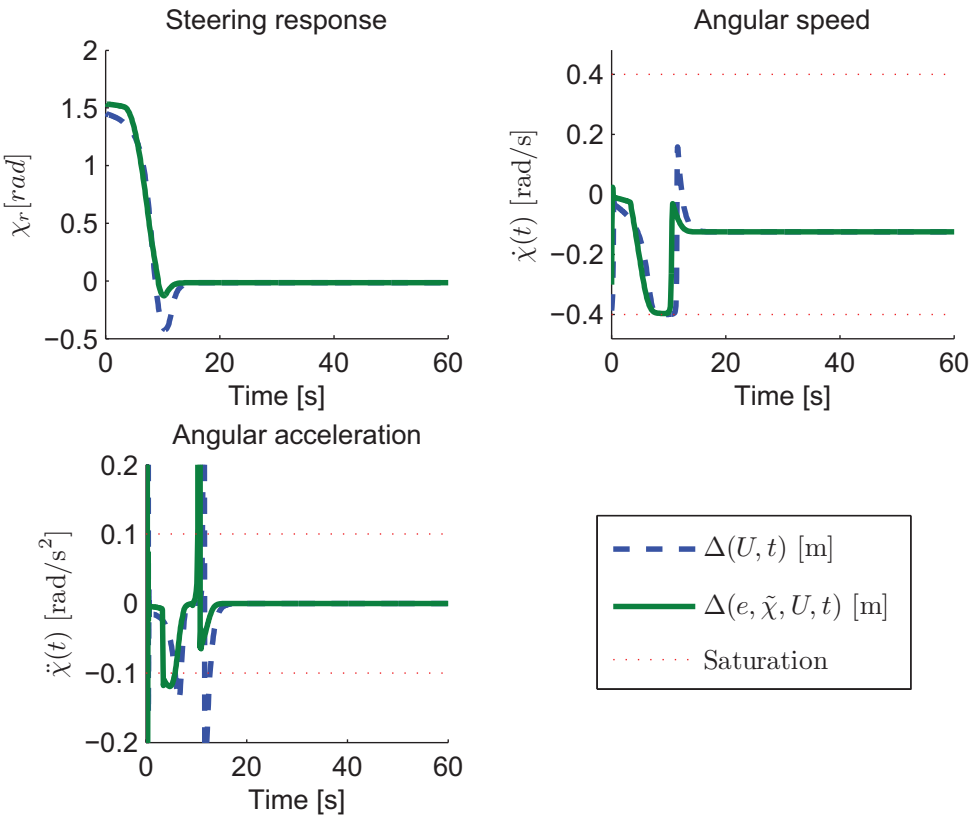


Figure 6.15: Case III: course response.

6.2.5 Case IV: Curved path with heuristic speed adaptation

This case considers the same guidance algorithms used in the previous two cases, but along a curved path. Due to the small curvature values, the piecewise path $\mathcal{P}_\delta$ is a relatively easy path to follow for a speed of 5 [m/s].

Parameters

The same parameters as in Case II is used.

Result

The results of this simulation can be seen in Figures 6.16-6.18. Both methods converge to the path and are able to follow it with high precision. A comparison of the metods reveals only small deviations between them, with the exception of the forward speed; the forward speed of the variable lookahead method is somewhat higher than the other. By inspecting the position response in Figure 6.16, it can be seen that a slight understeering is experienced during the steepest turns. This might suggest that the speed is a bit high during these turns. If the speed had been increased, the cross-track errors would most likely have increased. To feasibly follow the path at high speeds, the path speed algorithm must be incorporated.

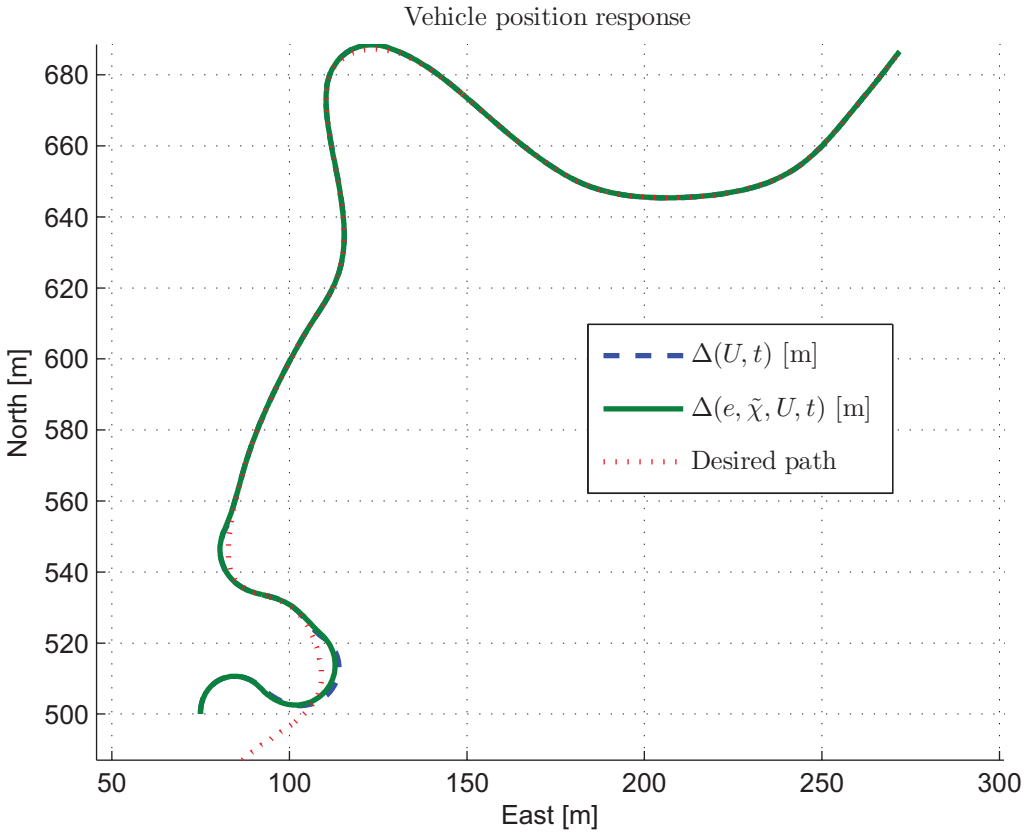


Figure 6.16: Case IV: position response.

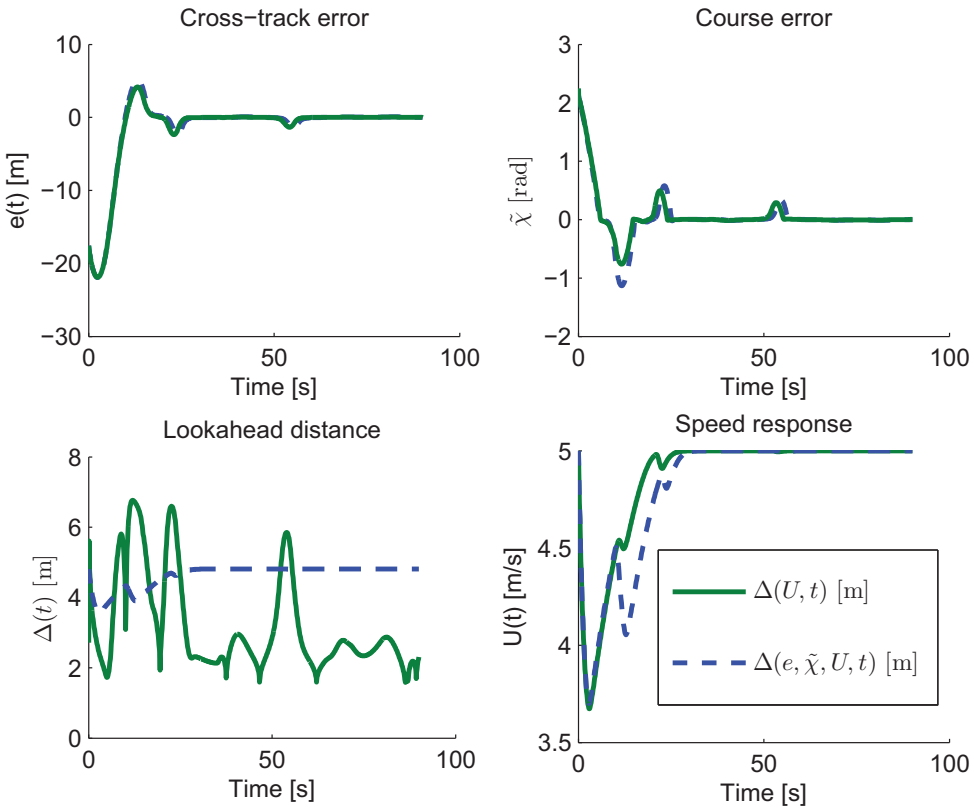


Figure 6.17: Case IV: error, lookahead and speed response.

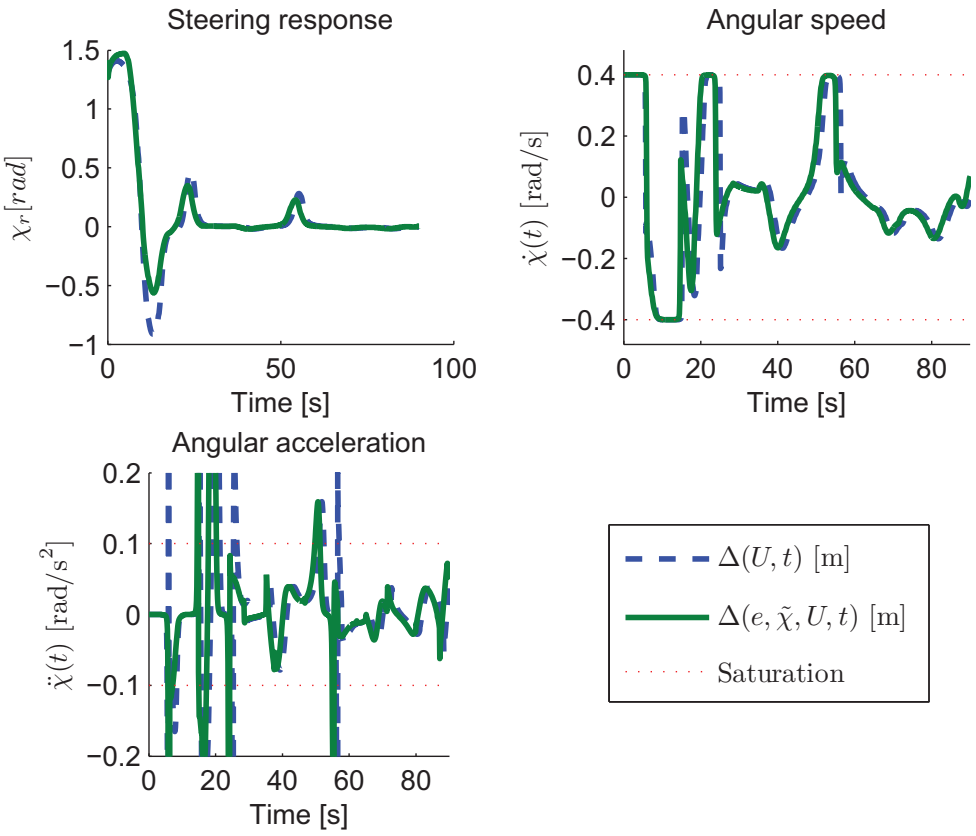


Figure 6.18: Case IV: course response.

6.2.6 Case V: Curved path with path speed algorithm

This case illustrates the enhanced path-following capabilities when the path speed algorithm is employed. This guidance algorithm is compared to a guidance algorithm where the forward speed is only determined by the heuristic speed adaptation. Both guidance algorithms use the same steering algorithm, that is, the variable lookahead distance approach. To really emphasize the benefits of the path speed algorithm, the challenging piecewise path $\mathcal{P}_\gamma$ is chosen.

Parameters

Once again, the same parameters as in Case II have been used. In addition, the lookahead system of the path speed algorithm has an along-path lookahead distance of $\Delta_d = 30$ [m] and a proportional feedback term $k_p = 1$. In contrast to the simulation in Section 6.2.1, the deceleration function is chosen to be more conservative than the one proposed in (6.4b):

$$f_{\text{dec}}(U) \quad = \quad \rho(a_{dec}U + U_{\text{max,l}}), \tag{6.5}$$

where $\rho = 0.5$. Otherwise, both the acceleration function (6.4a) and the parameters of both functions are the same as in 6.2.1. The reason for choosing a more conservative deceleration function is to allow more slack in the vehicle response. With a too aggressive deceleration function, the vehicle is not able to decelerate to the desired speed and thus the vehicle cannot follow the path.

Result

Figure 6.19 shows the vehicle response when using the two different approaches. The plot clearly shows that the vehicle has significantly better path-following capabilities when using the path speed algorithm. In this particular simulation, the guidance algorithm without path-dependent reference speed is able to traverse a bigger portion of the path. This achievement comes with the apparent drawback of higher cross-track errors, which can be seen in Figure 6.20. This figure also shows the generally higher speed the latter algorithm attains. Nevertheless, the guidance algorithm which employs the path speed algorithms by far excels the one with only heuristice speed adaptation. It is still worth noticing the cross-track error of the best

guidance algorithm during the steep turns. It is still present. This indicates that the speed has not been sufficiently reduced in time for the turns, which means that the decelerating function is to aggressive. By making it in even more conservative, this cross-track error can be reduced further.

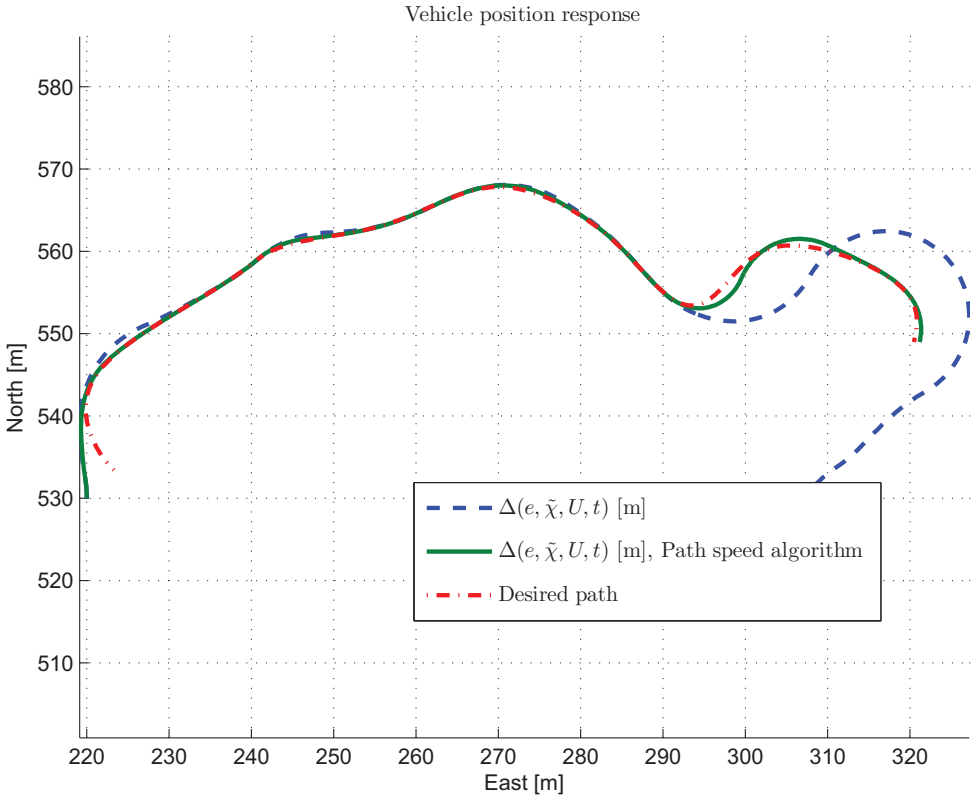


Figure 6.19: Case V: position response.

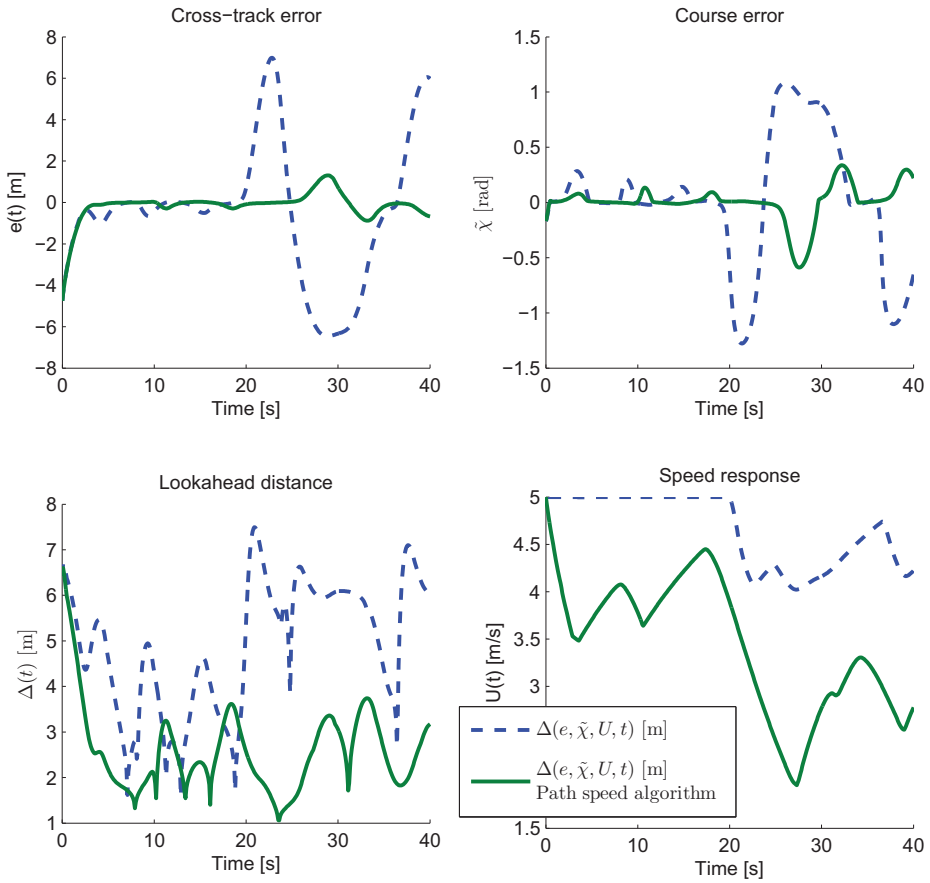


Figure 6.20: Case V: error, lookahead and speed response.

6.2.7 Case VI: Path-tracking system

Setup

The path-tracking system consists of two subsystems solving one objective each. These objectives are the path converging objective and the along-path speed objective. The former is solved using the same algorithm and parameters as in Section 6.2.5, while the latter is the path-tracking objective. Define the following parameters for the path-tracking speed controller:

$$U_{\text{a,max}} = 1\,[\text{m/s}], \tag{6.6}$$
$$\Delta_s = 4\,[\text{m}]. \tag{6.7}$$

The leader vehicle follows the "unknown" path $\mathcal{P}_\epsilon$ at constant speed $U_\ell = 2$ [m/s] and its initial position is $[146,\ 612]^\top$ [m]. The following vehicle's initial conditions are given in Table 6.3.

Let the desired along-path distance behind the leader vehicle be $\Delta_{pt} = 30$ [m]. Thus, the artificial particle's initial position on the created path can be found offline. Next, we let the online path-generation algorithm receive sampled positions every 5 [m]. Since we assume that only position data is known, we choose the $\mathcal{G}^2$-continuous Catmull-Rom path generation algorithm presented earlier.

Results

The position response The position responses are shown in Figure 6.21. After an initial transition phase, the following vehicle follows the created path, while also converging toward the artificial target. In the figure, the target vessel is invisible when the following vehicle has converged. This is because they are on top of each other. One particularly interesting observation is that the following vehicle is unable to follow the path when the leader vehicle performs acute turns. Often, the leader vehicle has slower dynamics than the following vehicle, hence this unfortunate behavior may not be a big problem in practice. Nevertheless, it affects the convergence rate due to the limited approach speed the following vehicle can have. One potential solution to overcome this drawback is to combine the path speed algorithm with the path-tracking speed controller. Even though the path is only defined a limited distance ahead, the path speed algorithm can still prevent the fol-

lowing vehicle from trying to obtain to aggressive approach speeds. Due to time constraints, this has not been verified through simulations.

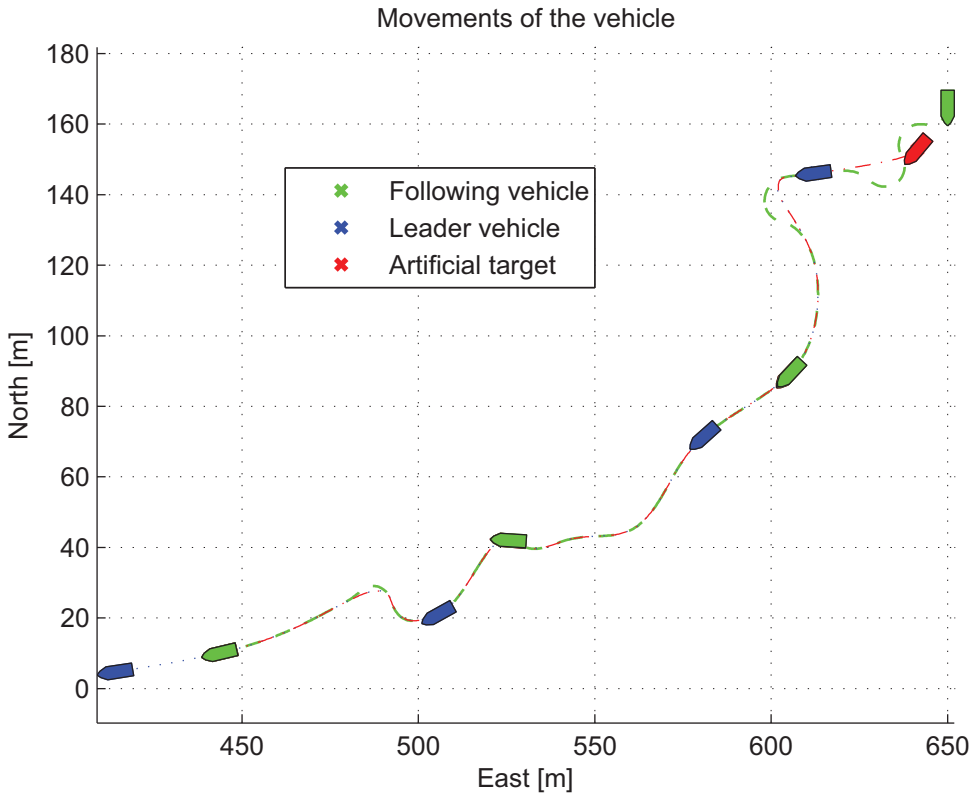


Figure 6.21: Case VI: position response.

Error responses Figure 6.22 shows the cross-track error, course error, and along-path arc-length error between the following vehicle and the artificial target vehicle. After the initial transition phase to the path, the arc-length error reduces linearly, and finally slows down as the arc-length error vanishes. This is the intended behavior, and is thus a satisfactory result. The same figure shows that the speed response varies in the beginning, due to the heuristic speed adaptation, but finally converges to speeds near the leader vehicle's speed.

The generated path versus the leader vehicle's path As can be seen in Figure 6.23, the online-generated path is quite similar to the real path. Even for sharp turns, the paths are close to each other, with a cross-track

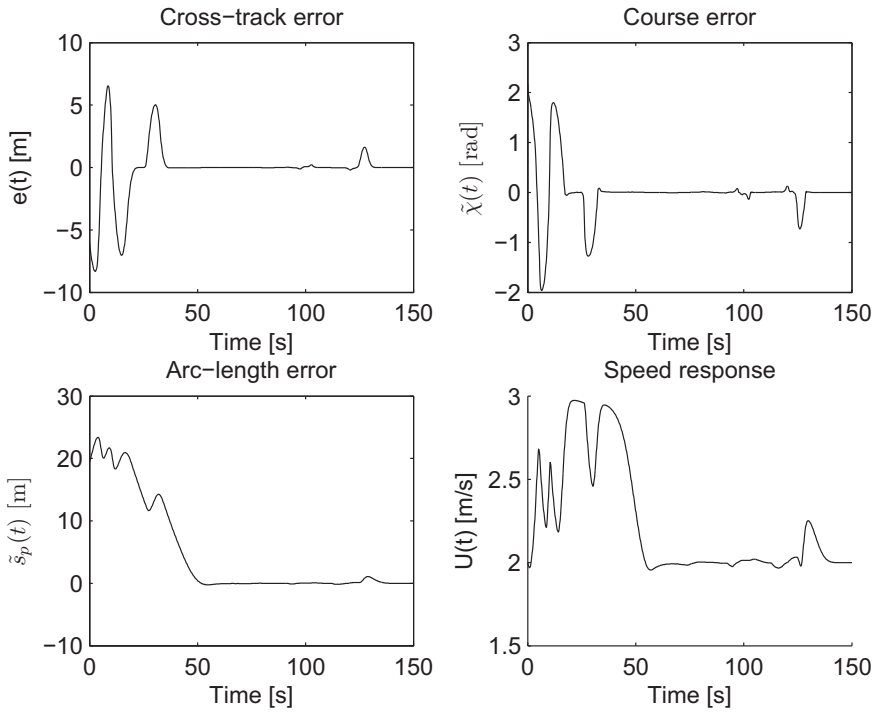


Figure 6.22: Case VI: error and speed response.

error of less than 0.5 [m]. This accuracy is sufficient in most cases, but by decreasing the sampling interval, higher accuracies can be obtained.

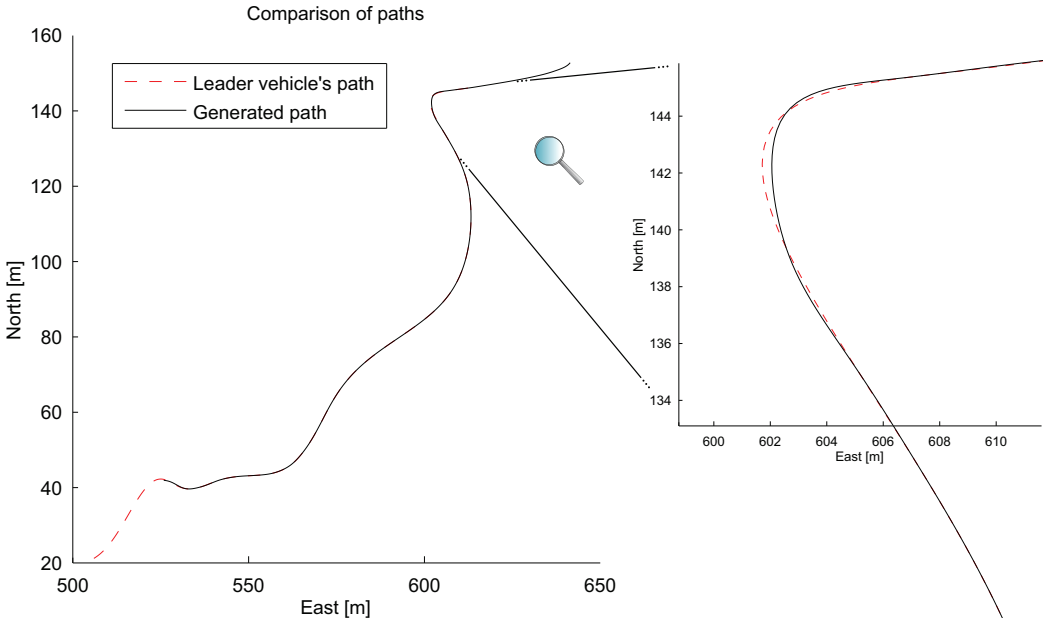


Figure 6.23: Case VI: comparison of paths.

Offset correction Each time a new segment is created by the path generator, there is a small offset between the arc length of the real and generated path. This offset creates a step in the distance the artificial target particle must travel to remain close to the real (and "unknown") artificial target particle. The lowermost plot in Figure 6.24 shows the corresponding arc-length difference between the desired position and actual position of the artificial target particle. We can see that the steps are corrected exponentially and never exceeds |0.2| [m].

In addition, the uppermost plot of the same figure shows the straight-line error between the target particle and real target particle. It remains within ±0.4 [m]. This error is a combination of the arc-length difference and the inaccuracy of the created path explained in the previous section. This result reveals that the artificial target particle is always close to the real target particle.

To summarize, the path-tracking system is able to create accurate paths online, where an artificial target particle remains close to desired along-path

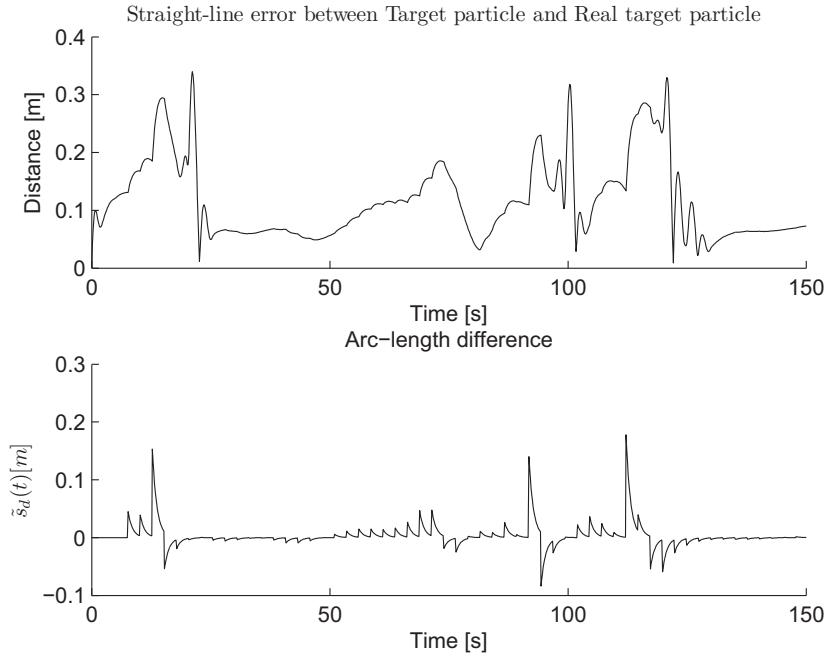


Figure 6.24: Case VI: artificial target versus real artificial target.

distance behind the leader vehicle. Furthermore, the following vehicle converges to this target with bounded speed, and is able to follow the path if the path is feasible for that particular vehicle.

6.2.8 Lawn-mower pattern

To illustrate the diverse paths the lawn-mower path planning algorithm can generate, six different speeds are given for the same set of waypoints. The desired waypoints are

$$P_0 = \boldsymbol{R}(\pi/4)\,[1,\,1]^\top, \qquad P_1 = \boldsymbol{R}(\pi/4)\,[30,\,1]^\top, \tag{6.8}$$
$$P_2 = \boldsymbol{R}(\pi/4)\,[30,\,20]^\top, \qquad P_3 = \boldsymbol{R}(\pi/4)\,[1,\,20]^\top, \tag{6.9}$$

such that the distance between the parallel lines is $d = 19$ [m]. Furthermore, the different forward speeds are

$$U \;=\; \{1,\,2,\,3,\,3.5,\,4,\,5\},\ [\mathrm{m/s}] \tag{6.10}$$

The vehicle maneuverability constraints are

$$\omega_{\max} \;=\; 0.4\,[\mathrm{rad/s}], \tag{6.11}$$
$$\dot{\omega}_{\max} \;=\; 0.1\,[\mathrm{rad/s^2}]. \tag{6.12}$$

The resulting paths are displayed in Figure 6.25. As expected, the arc lengths of the U-turns increase for higher speeds. Table 6.5 displays which case applies for the different speeds. A particularly interesting observation is that when switching from Case B to Case C, the resulting arc lengths increase significantly. Hence, to plan efficient paths, the forward speed should depend on the distance between the parallel lines. A measure of performence is the time it takes to traverse a given path. Since the arc length of the path depends on the forward speed, it is possible to construct a optimizing function such that the speed varies along the path, and the path is traversed in minimum time. Furthermore, there is a cost associate with a speed change, so the optimizing function should also penalize a change in both linear and angular speed. This topic however, is subject to future work, and will not be investigated further here.

Speed [m/s]	1	2	3	3.5	4	5
Case	A	A	B	B	C	C

Table 6.5: Relationship between speed and valid case for path planning of lawn-mower pattern.

To verify that the curvature changes in a controlled manner, we can investigate the curvature as a function of the arc length. Figure 6.26 displays the

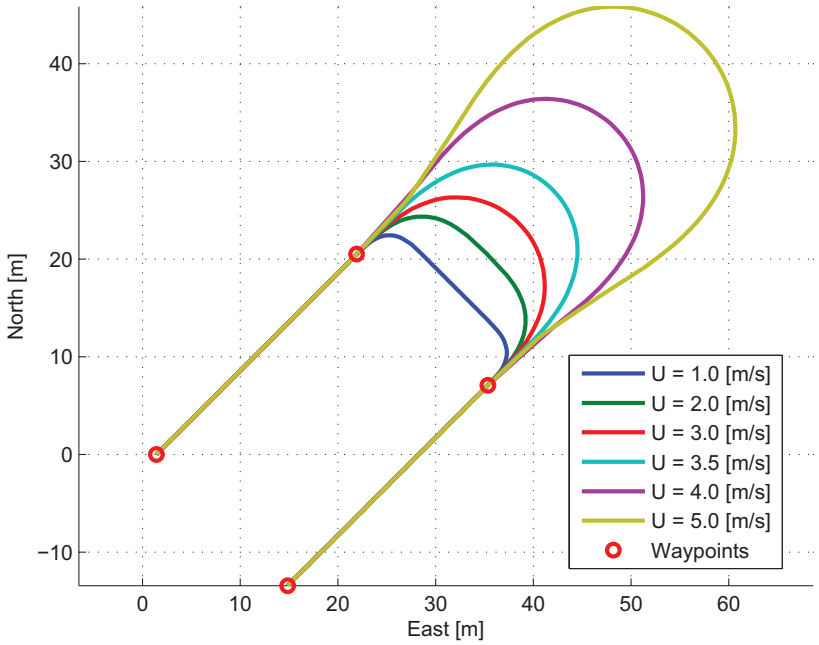


Figure 6.25: U-turn for different speeds.

result of this investigation. We can see that the curvature changes linearly for all the cases. This is expected, since the angular acceleration is constant when following clothoids. From the figure one might suspect that Case B is the most desirable, since the curvature only increase and the decrease once, unlike the other cases. If the path had been optimized with respect to the change in curvature, $d^2\varkappa/ds^2$, Case B would probably be the outcome.

Comparison to η-splines

The algorithm for constructing the lawn-mower pattern using clothoids is tedious and have many special cases which must be taken care of. On the other hand, using $\boldsymbol{\eta}$-splines to construct the U-turns seems simple. Hence, a comparison between the clothoid approach and $\boldsymbol{\eta}$-spline approach must be performed.

The $\boldsymbol{\eta}$-spline was discussed in Chapter 5.7. This method connects two configurations with a single parameterization. The drawback with this method is the difficulty of finding a feasible $\boldsymbol{\eta}$ vector. In (Piazzi et al. 2003) the

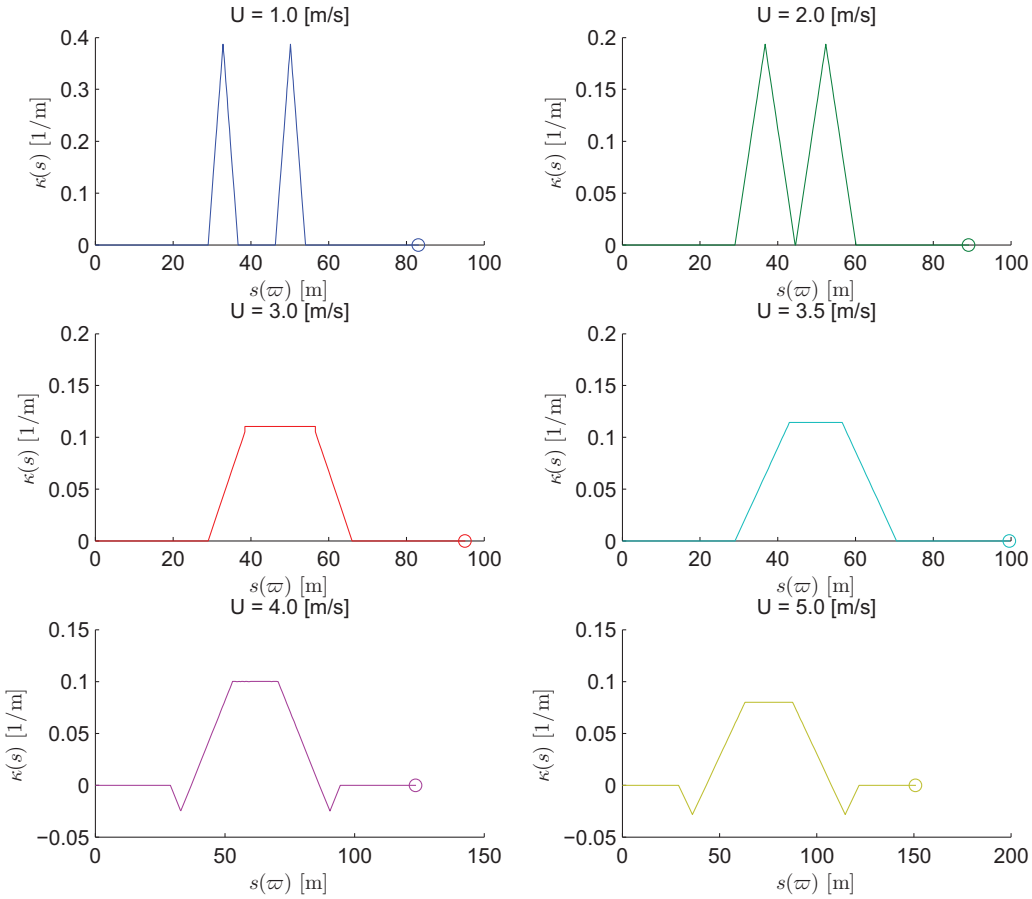


Figure 6.26: Signed curvature for the lawn-mower path.

following heuristic choice was proposed:

$$\eta_1 = \eta_2 = |\boldsymbol{p}_1 - \boldsymbol{p}_2| = d, \tag{6.13}$$

$$\eta_3 = \eta_4 = 0. \tag{6.14}$$

With this choice, an identical path will be created for a given distance d, regardless of the forward speed. To compensate for this unfortunate property, propose

$$\eta_1 = \eta_2 = b\frac{U}{\omega_{\max}(U)}d, \tag{6.15}$$

$$\eta_3 = \eta_4 = 0, \tag{6.16}$$

where $b > 0$ is a tuning parameter.

Set $b = 0.5$ and construct the paths for the same waypoint matrix as in the previous section, but only with the speeds

$$U = \{1, 3, 5\}, \text{[m/s]}. \tag{6.17}$$

The resulting paths are diplayed in Figure 6.27, and the curvatures as a function of arc length can be seen in Figure 6.28. For a given speed, the clothoid approach respects the vehicle maneuverability constraints. This means that the $\boldsymbol{\eta}$-spline path must be equal to or more conservative than the clothoid path to be feasible for a given speed. In Figure 6.27 it can be seen that for low speeds, the $\boldsymbol{\eta}$-spline is more convervative than the clothoid path, but ultimately, the $\boldsymbol{\eta}$-spline path becomes too aggressive. This is verified by Figure 6.28 where it can be seen that the slope of the curvature becomes too steep for higher speeds and is thus unfeasible, due to the angular acceleration the curvature rate represents.

The simulations show that the lawn-mower path planner using clothoids creates feasible paths in a controlled manner. When the speed and the maneuverability constraints force the use of Case C, the arc length of the U-turn becomes large. Hence, some kind of optimization of the forward speed must be created such that the resulting curve is optimal in some manner. The proposed approach creates a favorable fundament for one such extension.

Furthermore, a comparison to a $\boldsymbol{\eta}$-spline approach reveals better performance in terms of respecting the vehicle maneuverability constraints for the desired speed. It should be noted that with the proper choice of $\boldsymbol{\eta}$, the $\boldsymbol{\eta}$-spline approach can produce paths similar to those created by the clothoid approach, but with fewer parameterizations. Since no simple method for this achievement exist, the clothoid approach excels the $\boldsymbol{\eta}$-spline approach.

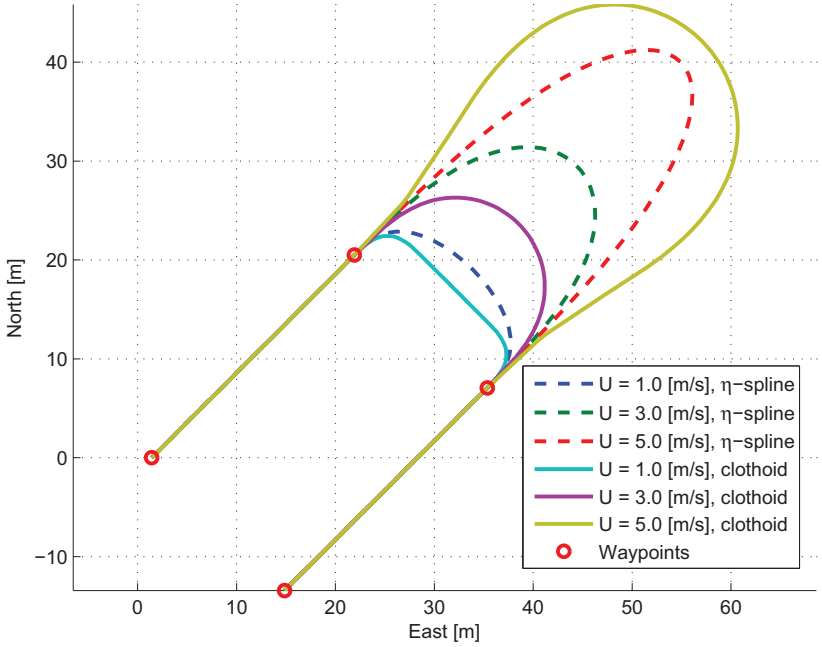


Figure 6.27: A comparison between clothoid and $\boldsymbol{\eta}$-spline U-turns.

6.3 Concluding remarks

This chapter illustrates the concepts developed in this thesis. Each simulation shows promising results which can be further improved in future work. Despite these uplifting results, there is a crucial limitation worth mentioning. The course model employed in the simulations does not have any limitations on the angular acceleration. This simplification greatly influences the simulation results, since the acceleration in reality is bounded. Thus, the results, which greatly depend on the course dynamics, are merely preliminary results which demonstrate the possible performance of the proposed guidance algorithms. Hence, an improved course model must be developed, where the angular acceleration is a part of the course dynamics. This problem is topic of future work.

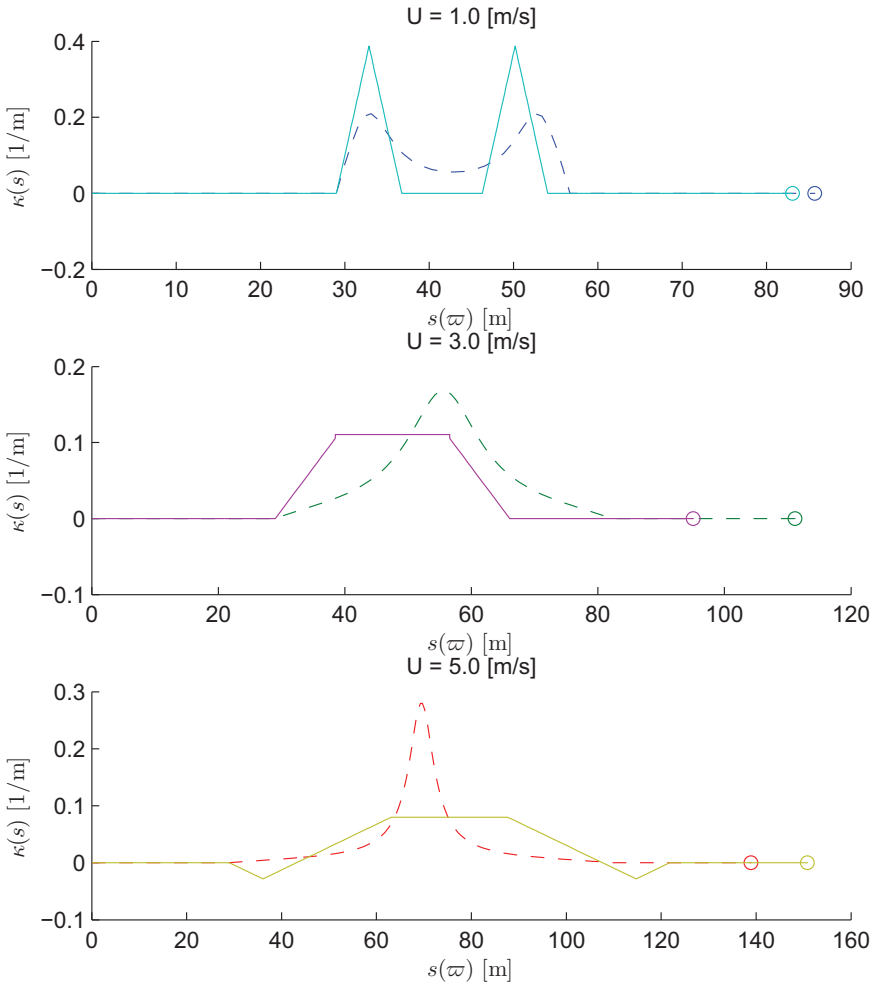


Figure 6.28: A comparison of the signed curvature for the clothoid and $\boldsymbol{\eta}$-spline approach for lawn-mower patterns.

Chapter 7

Conclusions and future work

A lookahead system, which gathers path information from piecewise paths, is proposed. The lookahead system, which is shown to be GES, is able to maintain a predefined distance ahead of the vehicle, even for curve segments which are not arc-length parameterized.

Moreover, simulations verify that the speed assignment system of the path speed algorithm is able to prescribe forward speeds which obey the vehicle maneuverability constraints. The path traversion can therefore be accomplished more accurately compared to a constant speed profile.

The two path convergence steering algorithms that have been proposed demonstrate good convergence behavior on the employed vehicle model. The more advanced steering algorithm displays more desirable behavior with respect to vehicle maneuverability constraints. Furthermore, the simple speed adaptation heuristics further improve the convergence characteristics of the vehicle.

The combination of the speed assignment system with the path convergence algorithm yields a path maneuvering algorithm which shows better path convergence and path traversion performance than a system that lacks these components.

The path-tracking system, which is proven to be UGAS/ULES, is able to accurately mimic a leader vehicle's path through the use of leader vehicle movements and splines. Furthermore, the following vehicle converges to the desired target along this path. The convergence has bounded approach speed, which makes it physically plausible.

The path planner for lawn-mower patterns is able to create U-turns which obey the vehicle maneuverability constraints for a given speed. This result lays the foundation for optimal path planning of lawn-mower patterns.

7.1 Future work

During the investigation of the topics in this thesis, new challenges have arisen. They can mainly be summarized as follows:

- Full-scale tests of the proposed guidance system.
- Employment of path characteristics to control the convergence rate such that it depends on curvature ahead of the vehicle.
- Development of a more precise course model in order to get responses which are closer to real course responses.
- Combination of the path-tracking speed controller and the speed assignment algorithm for feasible path traversion. This will able the vehicle to obtain higher approach speeds when the path allows it.
- Optimization of forward speed along the path of a lawn-mower pattern to obtain U-turn which is optimal in some sense, being time of traversion, length of total path, or some other criteria.

Bibliography

Abramowitz, M. & Stegun, I. A. (eds) (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 8th printing*, Dover Puplications, Inc, New York.

Archibald, R. C. (1918). Undergraduate mathematics clubs, Euler integrals and Euler's spiral, *American Mathematical Monthly* **25**(6): 276–282.

Baass, K. G. (1984). The use of clothoid templates in highway design, *Transportation Forum* **1**: 47–52.

Barsky, B. A. & DeRose, T. D. (1984). Geometric continuity of parametric curves, *Technical Report No. UCB/CSD 84/205*, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California.

Bertram, V. (2008). Unmanned Surface Vehicles - A survey, *Skibsteknisk Selskab, Copenhagen, Denmark* .

Bibuli, M., Bruzzone, G. & Caccia, M. (2009). Path-following algorithms and experiments for an unmanned surface vehicle, *Journal of Field Robotics* **26**(8): 669–688.

Bibuli, M., Parodi, O., Lapierre, L. & Caccia, M. (2009). Vehicle-following guidance for unmanned marine vehicles, *8th IFAC International Conference on Manoeuvring and Control of Marine Craft*, Guarujá, Brazil.

Blanc, C. & Schlick, C. (1995). X-splines : A spline model designed for the end-user, *Proc. SIGGRAPH'95*, pp. 377–386.

Bogacki, P. & Shampine, L. F. (1989). A 3(2) pair of Runge - Kutta formulas, *Applied Mathematics Letters* **2**(4): 321 – 325.

Børhaug, E., Pettersen, K. & Pavlov, A. (2006). An optimal guidance scheme for cross-track control of underactuated underwater vehicles, *Proceedings of the MED'06*, Ancona, Italy.

Breivik, M. (2010). *Topics in Guided Motion Control of Marine Vehicles*, PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Breivik, M. & Fossen, T. I. (2009). Guidance laws for autonomous underwater vehicles, *A. V. Inzartsev (Ed.), Underwater Vehicles*, IN-TECH Education and Publishing, pp. 51–76.

Breivik, M., Hovstein, V. E. & Fossen, T. I. (2008). Straight-line target tracking for unmanned surface vehicles, *Modeling, Identification and Control* **29**(4): 131–149.

Buchanan, J. L. & Turner, P. R. (1992). *Numerical Methods and Analysis*, McGraw-Hill, Inc., New York.

Casteljau, P. D. (1959). Courbes à pôles, *INPI* .

Catmull, E. & Rom, R. (1974). *Computer Aided Geometric Design*, Academic Press, New York, chapter A Class of Local Interpolating Splines.

Cesaro, E. (1886). Les lignes barycentriques, *Nouvelles annales de mathematiques* **3**(5): 511–520.

Cooper, S. L., Newborn, D. A. & Norton, M. R. (2002). New Paradigms in Boat Design: An Exploration into Unmanned Surface Vehicles, *Proceedings of the AUVSI Unmanned Systems*, Lake Buena Vista, Florida, USA.

Cornu, A. (1874). Methode nouvelle pour la discussion des problèmes de diffraction, *Journal physique théorique et appliquée* .

Delingette, H., Hebert, M. & Ikeuchi, K. (1991). Trajectory generation with curvature constraint based on energy minimization, *IROS '91*, Osaka, Japan.

DeRose, T. D. & Barsky, B. A. (1988). Geometric continuity, shape parameters and geometric constructions for Catmull-Rom splines, *ACM Transactions on Graphics* **7**(1): 1–41.

Egeland, O. & Gravdahl, J. T. (2002). *Modeling and Simulation for Automatic Control*, Marine Cybernetics, Trondheim, Norway.

Faltinsen, O. M. (2005). *Hydrodynamics of High-Speed Marine Vehicles*, Cambridge University Press.

Fleury, S., Souères, P., Laumond, J. P. & Chatila, R. (1995). Primitives for smoothing mobile robot trajectories, *IEEE Transactions on robotics and automation* **11**(3): 441–448.

Fossen, T. I. (2002). *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics, Trondheim, Norway.

Fossen, T. I., Breivik, M. & Skjetne, R. (2003). Line-of-sight path following of underactuated marine craft, *Proceedings of the 6th IFAC MCMC Conference*, Girona, Spain, pp. 244–249.

Fox, D., Burgard, W. & Thrun, S. (1997). The dynamic window approach to collision avoidance, *IEEE Robotics & Automation Magazine* **4**(1): 23–33.

Fredriksen, E. & Pettersen, K. (2006). Global κ-exponential way-point maneuvering of ships: Theory and experiments, *Automatica* **42**(4): 677 – 687.

Frost, J. R. (1999). *Principles of Search Theory*, Soza Company Ltd., Virginia, USA.

Gomes, P., Silvestre, C., Pascoal, A. & Cunha, R. (2007). A coastline following preview controller for the DELFIMx vehicle, *Proceedings of the Seventeenth International Offshore and Polar Engineering Conference*, Lisbon, Portugal.

Guarino Lo Bianco, C. & Piazzi, A. (2000). Optimal trajectory planning with quintic G^2-splines, *Proceedings of the IEEE Intelligent Vehicles Symposium*, Dearborn, USA, pp. 620–625.

Higgins, A. (1921). *The Transition Spiral and its Introduction to Railway Curves*, Constable, London.

Hildrestrand, J. R. S. (2010). *Modelling and control of anti-roll tanks*, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Joy, K. I. (2000). Bernstein polynomials, *Technical report*, Department of Computer Science, University of California, Davis, USA.

Khalil, H. K. (2002). *Nonlinear Systems*, 3 edn, Prentice Hall.

Kimia, B. B., Frankel, I. & Popescu, A.-M. (2003). Euler spiral for shape completion, *International Journal of Computer Vision* **54**(1-3): 159–182.

Lamnabhi-Lagarrigue, F., Loría, A. & Panteley, E. (eds) (2005). *Advanced Topics in Control Systems Theory*, Springer, London, chapter Cascaded nonlinear time-varying systems: Analysis and design, pp. 23–61.

Lepetic, M., Klancar, G., Skrjanc, I., Matko, D. & Potocnik, B. (2003). Time optimal path planning considering acceleration limits, *Robotics and Autonomous Systems* **45**(3-4): 199–210.

McPhee, J. (2006). *Uncommon Carriers*, New York: Farrar, Straus, and Giroux, New York, pp. 47–48.

Meek, D. S. & Walton, D. J. (2004). A note on finding clothoids, *Journal of Computational and Applied Mathematics* **170**(2): 433 – 453.

Nagy, B. & Kelly, A. (2001). Trajectory generation for car-like robots using cubic curvature polynomials, *in Field and Service Robots*, Helsinki, Finland.

Newton, I. (1711). *De analysi per aequationes numero terminorum infinitas.*

Nord, P. (2010). *Collision-free path planning for unmanned surface vehicles*, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Oh, S.-R. & Sun, J. (2010). Path following of underactuated marine surface vessels using line-of-sight based model predictive control, *Ocean Engineering* **37**(2-3): 289 – 295.

Ousingsawat, J. & Earl, M. G. (2007). Modified lawn-mower search pattern for areas comprised of weighted regions, *Proceeding of the 2007 American Control Conference*, New York City, USA, pp. 918–923.

Pavlov, A., Nordahl, H. & Breivik, M. (2009). MPC-based optimal path following for underactuated vessels, *8th IFAC International Conference on Manoeuvring and Control of Marine Craft*, Guarajá, Brazil, pp. 340–345.

Pendrill, A. M. (2005). Rollercoaster loop shapes, *Physics Education* **40**(6): 517–521.

Pettersen, K. Y. & Lefeber, E. (2001). Way-point tracking control of ships, *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, Florida, USA, pp. 940–945.

Piazzi, A. & Guarino Lo Bianco, C. (2000). Quintic G^2-splines for trajectory planning of autonomous vehicles, *Proceedings of the IEEE Intelligent Vehicles Symposium*, Dearborn, USA, pp. 199–203.

Piazzi, A., Romano, M. & Guarino Lo Bianco, C. (2003). G^3-splines for the path planning of wheeled mobile robots, *European Control Conference, ECC'03*, Cambridge - United Kingdom.

Raphson, J. (1690). *Analysis aequationum universalis*, London.

Runge, C. (1901). Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten, *Zeitschrift für Mathematik und Physik* **46**: 224–243.

Scheuer, A. & Fraichard, T. (1997). Continuous-curvature path planning for car-like vehicles, *In Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, pp. 997–1003.

Shin, D. H. & Singh, S. (1990). Path generation for robot vehicles using composite clothoid segments, *Technical report*, The Robotics Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

Skejic, R., Breivik, M., Fossen, T. I. & Faltinsen, O. M. (2010). Modeling and control of underway replenishment operations in calm water, *Technical report*, Norwegian University of Science and Technology, Trondheim, Norway.

Skjetne, R. (2005). *The Maneuvering Problem*, PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Skjetne, R., Fossen, T. I. & Kokotović, P. V. (2004). Robust output maneuvering for a class of nonlinear systems, *Automatica* **40**(3): 373–383.

SNAME (1950). The society of naval architects and marine engineers. Nomenclature for treating the motion of a submerged body through a fluid, *Technical and research bulletin No. 1-5* .

SpringerLink (2010). Curvature, `http://eom.springer.de/C/c027320.htm`. Online; accessed March 23. 2010.

Subbotin, M., Daçić, D. & Smith, R. (2006). Preview based path-following in the presence of input constraints, *Proceedings of the American Control Conference*, Minneapolis, Minnesota, USA.

Talbot, A. N. (1927). *The Railway Transition Spiral*, 6th edn, McGraw-Hill Book Company, Inc., New York and London.

USCG (2010). Ais overview, `http://navcen.uscg.gov/enav/ais/`. Online; accessed May 25. 2010.

Weston, S. (2002). An introduction to the mathematics and construction of splines, *Technical report*, Addix Software Consultancy Limited.

Yang, K., Gan, S. K. & Sukkarieh, S. (2009). An efficient path planning and control algorithm for RUAV's in unknown and cluttered environments, *Journal of Intelligent and Robotic Systems* **57**(1-4): 101–122.

Yoshimoto, K., Katoh, M. & Inoue, K. (2000). A vision-based speed control algorithm for autonomous driving, *Proceedings of the 5th International Symposium on Advanced Vehicle Control*, Ann Arbor, Michigan, USA.

Appendix A

Path maneuvering simulation environment

To be able to simulate the proposed system in a structurally neat manner, a custom simulation environment has been constructed. The tools were programmed using the MathWorks™ environment MATLAB®. More specifically, a package consisting of support classes designed using the object-oriented techniques has been created. This makes it easier to grasp for other people than the designer. Furthermore, the design can easily be extended or modified in the future to add addition functionality.

A rough UML-diagram of the package can be found in Figure A.1, where the most important components are included. An explanation beyond what the diagram itself provides is not given, since this is considered outside the scope of this thesis.

This design was intended for use in Simulink® to aid the simulations of the vehicle responses. Although it works, the current implementation is slow. Hence, optimization of the code, and maybe converting it to a so-called *S-function* is topic of future work. This opens the possibilities for real-time simulations and experimental trials.

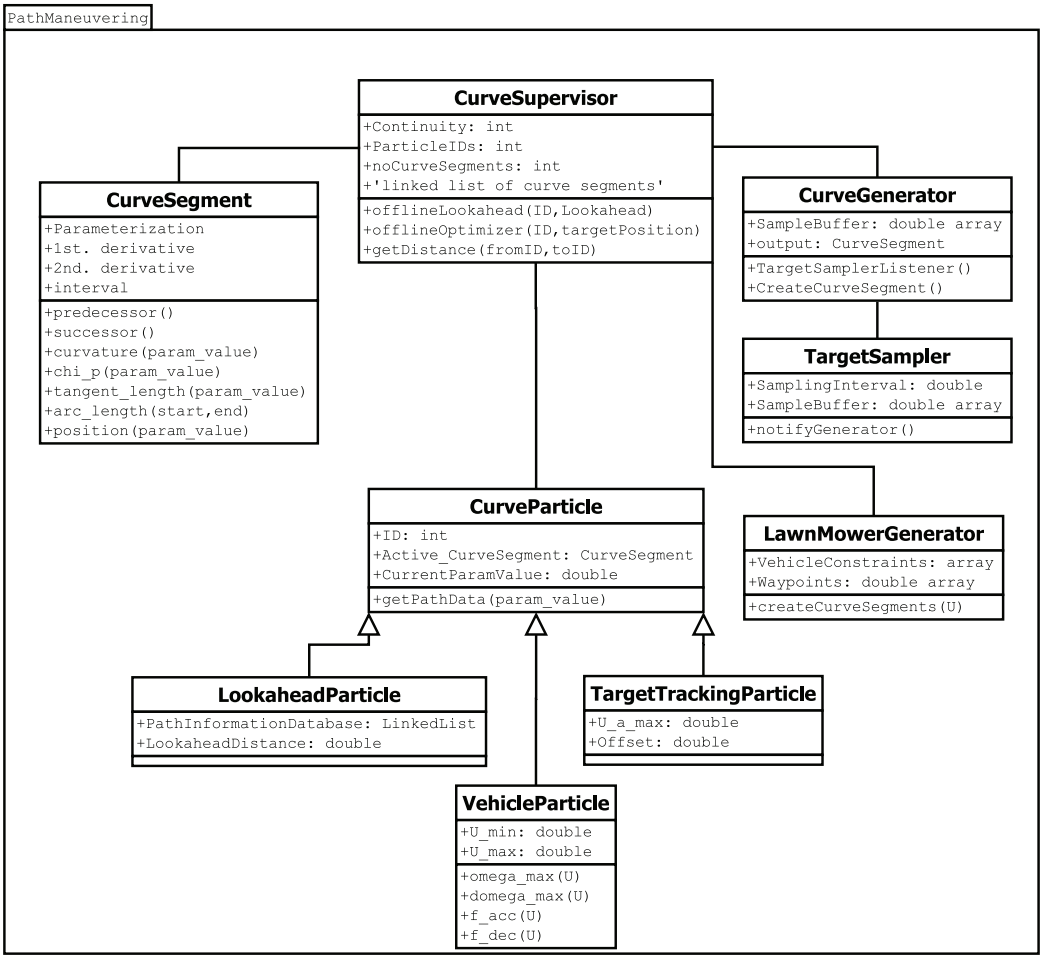


Figure A.1: Rough UML-diagram of the PathManeuvering package.

Appendix B

CD Contents

A CD is bundled with this thesis. The main contents of the CD are:

- This thesis in portable document format (PDF) named *Haugen2010.pdf*
- Referenced articles in PDF in the folder *bibliography*
- Simulation files including the package *PathManeuvering* in *simulation*

The folder *simulation* contains many different scripts and functions which are used in the simulations. There exist several simulation scripts which make it easy to reproduce the simulation scenarios. For these simulations to run correctly, *InitFunction.m* must be run first. This script adds the root folder and the subfolders to the *MATLAB* path. The simulations files can be found in the folder *simulationFiles* and its subfolders. Table B.1 lists the different cases and the corresponding simulation files.

Simulation	File in *simulationFiles/*
Course step	*course_step.m*
Speed step	*speed_step.m*
Path speed	*PathSpeed/sim_path_speed.m*
Case I	*Straight/straightI_sim.m*
Case II	*Straight/straightII_sim.m*
Case III	*Circle/circle_sim.m*
Case IV	*Curved/curved_sim.m*
Case V	*Curved/curvedV_sim.m*
Case VI	*Tracker/CaseVI_sim.m*
Lawn Mower	*LawnMower/simulation.m*

Table B.1: Simulation files.

Appendix C

Complementary material

C.1 Curvature

From (3.19) we have

$$\kappa(\varpi) = \left| \frac{d\boldsymbol{T}(\varpi)}{ds} \right|, \tag{C.1}$$

and by utilizing the chain rule we get

$$\kappa(\varpi) = \left| \frac{d\boldsymbol{T}(\varpi)}{ds} \right| = \left| \frac{\frac{d\boldsymbol{T}}{d\varpi}}{\frac{ds}{d\varpi}} \right| = \frac{|\boldsymbol{T}'(\varpi)|}{|p'(\varpi)|}, \tag{C.2}$$

where $\boldsymbol{T}'(\varpi) = \frac{\boldsymbol{p}'(\varpi)}{|\boldsymbol{p}'(\varpi)|}$.

In the following we will use $\frac{d\boldsymbol{p}}{d\varpi}(\varpi) \triangleq \boldsymbol{p}'$ and $\boldsymbol{p} = \begin{bmatrix} x \\ y \end{bmatrix}$.

$$\boldsymbol{T}'(\varpi) = \frac{\boldsymbol{p}'' \cdot |\boldsymbol{p}'| - \frac{\boldsymbol{p}' \overbrace{(x' \cdot x'' + y' \cdot y'')}^{(\boldsymbol{p}' \cdot \boldsymbol{p}'')}}{|\boldsymbol{p}'|}}{|\boldsymbol{p}'|^2}. \tag{C.3}$$

$$|\boldsymbol{T}'(\varpi)| = \frac{1}{|\boldsymbol{p}'|^2} \sqrt{\left(x'' \cdot |\boldsymbol{p}'| - \frac{1}{|\boldsymbol{p}'|} x' \cdot (\boldsymbol{p}' \cdot \boldsymbol{p}'') \right)^2 + \left(y'' \cdot |\boldsymbol{p}'| - \frac{1}{|\boldsymbol{p}'|} y' \cdot (\boldsymbol{p}' \cdot \boldsymbol{p}'') \right)^2}$$
$$= \frac{1}{|\boldsymbol{p}'|^2} \sqrt{\underbrace{(x''^2 + y''^2)}_{|\boldsymbol{p}''|^2} |\boldsymbol{p}'|^2 - 2 \underbrace{(x' \cdot x'' + y' \cdot y'')}_{(\boldsymbol{p}' \cdot \boldsymbol{p}'')} (\boldsymbol{p}' \cdot \boldsymbol{p}'') + \underbrace{(x'^2 + y'^2)}_{|\boldsymbol{p}'|^2} \frac{1}{|\boldsymbol{p}'|^2} (\boldsymbol{p}' \cdot \boldsymbol{p}'')^2}$$
$$= \frac{1}{|\boldsymbol{p}'|^2} \sqrt{|\boldsymbol{p}''|^2 \cdot |\boldsymbol{p}'|^2 - (\boldsymbol{p}' \cdot \boldsymbol{p}'')^2}$$

$$
\begin{aligned}
&= \frac{1}{|\boldsymbol{p}'|^2}\sqrt{(x'\cdot y'')^2 - 2x'\cdot x''\cdot y'\cdot y'' + (y'\cdot x'')^2} \\
&= \frac{1}{|\boldsymbol{p}'|^2}\sqrt{(x'\cdot y'' - y'\cdot x'')^2} \\
|\boldsymbol{T}'(\varpi)| &= \frac{|(x'\cdot y'' - y'\cdot x'')|}{|\boldsymbol{p}'|^2}.
\end{aligned} \tag{C.4}
$$

Insert (C.4) into (C.2) and we get the following equation for path curvature

$$
\begin{aligned}
\kappa(\varpi) &= \frac{|(x'\cdot y'' - y'\cdot x'')|}{|\boldsymbol{p}'|^3} \\
&\Downarrow \quad \boldsymbol{p}'\times\boldsymbol{p}'' \triangleq (x'\cdot y'' - y'\cdot x'') \\
\kappa(\varpi) &= \frac{|\boldsymbol{p}'(\varpi)\times\boldsymbol{p}''(\varpi)|}{|\boldsymbol{p}'(\varpi)|^3}
\end{aligned} \tag{C.5}
$$

C.1.1 Curvature of clothoids

The curvature of a clothoid can be calculated using the formula for curvature (C.5). The 1st derivative of the clothoid is (3.37)

$$
\boldsymbol{p}'(\vartheta) = \frac{a}{\sqrt{\vartheta}}\begin{bmatrix}\cos(\vartheta)\\ \sin(\vartheta)\end{bmatrix}, \vartheta \geq 0, \tag{C.6}
$$

and the 2nd derivative is

$$
\boldsymbol{p}''(\vartheta) = \frac{a\sqrt{\vartheta}}{\vartheta}\begin{bmatrix}-\sin(\vartheta) - \frac{1}{2\vartheta}\cos(\vartheta)\\ \cos(\vartheta) - \frac{1}{2\vartheta}\sin(\vartheta)\end{bmatrix}, \vartheta \geq 0. \tag{C.7}
$$

Furthermore the denominator of (C.5) is

$$
|\boldsymbol{p}'(\vartheta)|^3 = \left(\frac{a}{\sqrt{\vartheta}}\right)^3, \tag{C.8}
$$

and by using (C.6)-(C.8) in (C.5) we get

$$
\begin{aligned}
\kappa(\vartheta) &= \frac{a^2/\vartheta}{a^3/\vartheta^{3/2}}\left(\underbrace{\cos^2(\vartheta)+\sin^2(\vartheta)}_{1} - \frac{1}{2\vartheta}\underbrace{(\sin(\vartheta)\cos(\vartheta) - \sin(\vartheta)\cos(\vartheta))}_{0}\right) \\
\kappa(\vartheta) &= \frac{\sqrt{\vartheta}}{a}.
\end{aligned} \tag{C.9}
$$

C.2 Calculation of $\dot{\chi}_r$ in the circular convergence phase

Given (5.33) and (5.46):

$$\dot{\chi}_r(e) = \frac{e\dot{\Delta} - \frac{-Ue\Delta}{\sqrt{\Delta^2+e^2}}}{\Delta^2+e^2} \tag{C.10}$$

$$\Delta(e) = (R-|e|)\sqrt{\frac{|e|}{2R-|e|}}, \quad e < R. \tag{C.11}$$

First, eliminate Δ from the following expression by employing (C.11):

$$\begin{aligned}
\Delta^2+e^2 &= \frac{e^2(R-|e|)^2}{2R|e|-e^2} \\
&= \frac{e^2}{2R|e|-e^2}\left(R^2-2R|e|+e^2+2R|e|-e^2\right) \\
&\Downarrow \\
\Delta^2+e^2 &= \frac{R^2|e|}{2R-|e|}.
\end{aligned} \tag{C.12}$$

Second, eliminate Δ from (C.10) by using (C.11) and (C.12)

$$\begin{aligned}
\dot{\chi}_r &= \frac{e\dot{\Delta} + Ue(R-|e|)\sqrt{\frac{|e|}{2R-|e|}}\sqrt{\frac{2R-|e|}{R^2|e|}}}{\frac{R^2|e|}{2R-|e|}} \\
&= \mathrm{sgn}(e)\frac{2R-|e|}{R^2}\left(\dot{\Delta}+\frac{U}{R}(R-|e|)\right).
\end{aligned} \tag{C.13}$$

The expression for $\dot{e}$ (5.33) can be written as

$$\begin{aligned}
\dot{e} &= \frac{-Ue}{\sqrt{\Delta^2+e^2}} \\
&\Downarrow \ (C.12) \\
&= -\frac{Ue}{R}\sqrt{\frac{2R-|e|}{|e|}} \\
\dot{e} &= -\mathrm{sgn}(e)\frac{U}{R}\sqrt{|e|(2R-|e|)}.
\end{aligned} \tag{C.14}$$

Furthermore, $\dot{\Delta}$ is given by

$$\begin{aligned}
\dot{\Delta} &= \frac{d}{dt}\left((R-|e|)\sqrt{\frac{|e|}{2R-|e|}}\right) \\
&= -\mathrm{sgn}(e)\dot{e}\sqrt{\frac{|e|}{2R-|e|}} + \mathrm{sgn}(e)\dot{e}\frac{(R-|e|)(2R-|e|-(-|e|))}{2\sqrt{\frac{|e|}{2R-|e|}}(2R-|e|)^2} \\
&= \mathrm{sgn}(e)\dot{e}\left(-\sqrt{\frac{|e|}{2R-|e|}} + R(R-|e|)\sqrt{\frac{2R-|e|}{|e|}}\frac{1}{(2R-|e|)^2}\right) \\
&= \mathrm{sgn}(e)\dot{e}\frac{1}{\sqrt{2R-|e|}}\left(-\sqrt{|e|} + \frac{R(R-|e|)}{2R-|e|}\frac{1}{\sqrt{|e|}}\right), \qquad \text{(C.15)}
\end{aligned}$$

where $\dot{e}$ can be eliminated with (C.14):

$$\begin{aligned}
&= -\mathrm{sgn}(e)\mathrm{sgn}(e)\frac{U}{R}\sqrt{\frac{|e|(2R-|e|)}{2R-|e|}}\left(-\sqrt{|e|} + \frac{R(R-|e|)}{2R-|e|}\frac{1}{\sqrt{|e|}}\right) \\
&= -\frac{U}{R}\left(-|e| + \frac{R(R-|e|)}{2R-|e|}\right) \\
\dot{\Delta} &= -\frac{U}{R}\left(\frac{R^2-3R|e|+e^2}{(2R-|e|)}\right). \qquad \text{(C.16)}
\end{aligned}$$

Now, $\dot{\Delta}$ must be eliminated from the expression of $\dot{\chi}_r$ (C.13)

$$\begin{aligned}
\dot{\chi}_r &= \mathrm{sgn}(e)\frac{2R-|e|}{R^2}\left(\dot{\Delta} + \frac{U}{R}(R-|e|)\right) \\
&= \mathrm{sgn}(e)\frac{2R-|e|}{R^2}\left(-\frac{U}{R}(\frac{R^2-3R|e|+e^2}{(2R-|e|)}) + \frac{U}{R}(R-|e|)\right) \\
&= \mathrm{sgn}(e)\frac{2R-|e|}{R^2}\frac{U}{R}\left(\frac{-R^2+3R|e|-e^2+(R-|e|)(2R-|e|)}{2R-|e|}\right) \\
&= \mathrm{sgn}(e)\frac{U}{R^3}\left(-R^2+3R|e|-e^2+2R^2-R|e|-2R|e|+e^2\right) \\
\dot{\chi}_r &= \mathrm{sgn}(e)\frac{U}{R}. \qquad \text{(C.17)}
\end{aligned}$$

C.3 Calculation of $\dot{\chi}_r$ in the clothoid transition phases

The path-relative course rate $\dot{\chi}_r(e)$ when steering for straight lines is given by (5.32):

$$\dot{\chi}_r(e) \;=\; \frac{e\dot{\Delta} - \dot{e}\Delta}{\Delta^2 + e^2}. \tag{C.18}$$

In the two next sections we show how this expression can simplified when following clothoid paths.

C.3.1 $\dot{\chi}_r$ in phase I

In the first transition phase, the lookahead distance is given by (5.64)

$$\Delta_I(e) \;=\; |e| \tan\left(\vartheta_I(e)\right), \tag{C.19}$$

and its time derivative is

$$\dot{\Delta}_I \;=\; \mathrm{sgn}(e)\dot{e}\tan(\vartheta_I) + |e|(1+\tan^2(\vartheta_I))\dot{\vartheta}_I. \tag{C.20}$$

Next, we eliminate $\dot{\vartheta}_I$ by employing (3.16)

$$\dot{\vartheta}_I \;=\; \frac{U}{|\boldsymbol{p}'(\vartheta_I)|}, \tag{C.21}$$

and inserting the length of the clothoid tangent vector (3.37)

$$\begin{aligned} |\boldsymbol{p}'(\vartheta_I)| \;&=\; \left| \frac{a}{\sqrt{\vartheta_I}} \begin{bmatrix} \cos(\vartheta_I) \\ \sin(\vartheta_I) \end{bmatrix} \right| \\ &=\; \frac{a}{\sqrt{\vartheta_I}}, \end{aligned} \tag{C.22}$$

we get

$$\dot{\vartheta}_I \;=\; U\frac{\sqrt{\vartheta_I}}{a}. \tag{C.23}$$

Furthermore, by noticing that the curvature of a clothoid is (3.38)

$$\kappa(\vartheta_I) \;=\; \frac{\sqrt{\vartheta_I}}{a}, \tag{C.24}$$

we can finally write the time derivative of the path parameter as

$$\dot{\vartheta}_I = U\kappa(\vartheta_I). \tag{C.25}$$

By inserting this result in (C.20) and eliminating $\dot{\Delta}$ and Δ from (C.18) we get

$$\begin{aligned}\dot{\chi}_r(e) &= \frac{e(\operatorname{sgn}(e)\dot{e}\tan(\vartheta_I) + |e|(1+\tan^2(\vartheta_I))U\kappa(\vartheta_I)) - \dot{e}|e|\tan(\vartheta_I(e))}{(|e|\tan(\vartheta_I(e)))^2 + e^2} \\ &= \frac{e|e|(1+\tan^2(\vartheta_I))U\kappa(\vartheta_I)}{e^2(1+\tan^2(\vartheta_I))} \\ &= \operatorname{sgn}(e)\frac{U}{R(\vartheta_I(e))}.\end{aligned} \tag{C.26}$$

C.3.2 $\dot{\chi}_r$ in phase III

In the second transition phase, the lookahead distance is (5.81)

$$\Delta_{III}(e) = \frac{|e|}{\tan(\vartheta_{III}(e))}. \tag{C.27}$$

The time derivative of the lookahead distance is

$$\dot{\Delta}_{III} = \frac{\operatorname{sgn}(e)\dot{e}\tan(\vartheta_{III}) - |e|\sec^2(\vartheta_{III})\dot{\vartheta}_{III}}{\tan^2(\vartheta_{III})}. \tag{C.28}$$

Similar to (C.25), it can be shown that

$$\dot{\vartheta}_{III} = -U\kappa(\vartheta_{III}). \tag{C.29}$$

The negative sign comes due to decreasing path parameter in this phase.

Again, we can arrive at the simplified expression for the course rate by eliminating $\dot{\Delta}$ and Δ from (C.18)

$$\begin{aligned}\dot{\chi}_r(e) &= \frac{e(\operatorname{sgn}(e)\dot{e}\frac{\tan(\vartheta_{III})}{\tan^2(\vartheta_{III})} + |e|\frac{1+\tan^2(\vartheta_{III})}{\tan^2(\vartheta_{III})}U\kappa(\vartheta_{III})) - \frac{\dot{e}|e|}{\tan(\vartheta_{III}(e))}}{\left(\frac{|e|}{\tan(\vartheta_{III}}\right)^2 + e^2} \\ &= \frac{e|e|\frac{1+\tan^2(\vartheta_{III})}{\tan^2(\vartheta_{III})}U\kappa(\vartheta_I)}{e^2\left(\frac{1+\tan^2(\vartheta_{III})}{\tan^2(\vartheta)}\right)} \\ &= \operatorname{sgn}(e)\frac{U}{R(\vartheta_{III}(e))}.\end{aligned} \tag{C.30}$$

C.4 Newton-Raphson method

The Newton-Raphson method is a root-finding algorithm which use tangent lines to successively converge to the root of a function. The method was first described by Newton in 1669, but published much later (Newton 1711). In the mean time, Raphson also described same principle, but in a more elegant manner (Raphson 1690). In textbooks today, the algorithm, which is also known as *Newton's root-finding method*, is presented with Raphson's approach.

The idea is to start with an initial guess of $x = x_k$ and approximate the function in a neighborhood of x_k with a first order Taylor series. Next, we find the point $x_{k+1} = x_k + \epsilon$ where this approximation intersects the x-axis.

$$\begin{aligned} f(x) &\approx f(x_k) + f'(x_k)\epsilon \\ 0 &= f(x_k) + f'(x_k)\epsilon \\ &\Downarrow \\ \epsilon &= -\frac{f(x_k)}{f'(x_k)}. \end{aligned} \tag{C.31}$$

The point

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{C.32}$$

is a new approximation of the location of the root. Figure C.1 illustrates the principle of the algorithm. By repeating (C.32) in an iterative loop, x may or may not converge to the root. Certain condition must however be satisfied to guarantee convergence.

Theorem C.1 (From (Buchanan & Turner 1992))**.** *If the function $f(x)$ is twice differentiable and satisfies the following conditions on the interval $[a, b]$:*

a) $f(a)$ and $f(b)$ are opposite signs,

b) $f'(x)$ is nonzero in $[a, b]$,

c) $f''(x)$ does not change sign in $[a, b]$,

d) $\left[\frac{f(a)}{f'(a)}\right] < b - a$ and $\left[\frac{f(b)}{f'(b)}\right] > b - a$,

the Newton's root-finding method converges to the unique root of the equation $f(x) = 0$ in $[a, b]$ for any starting value in $[a, b]$.

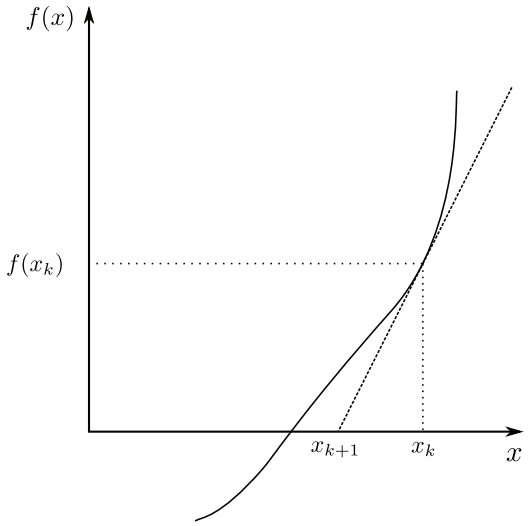


Figure C.1: The principle of the Newton-Raphson method.

In some cases a precise value of $f'(x_k)$ is not available; the interval $[a, b]$ has to be widened to meet the conditions of Theorem C.1. When $f(x)$ contains one of the Fresnels integrals (3.34a)-(3.34b), a widened interval must be developed (Meek & Walton 2004).

Lemma C.1 (From (Meek & Walton 2004))**.** *A function $f(x)$ is on standard form if:*

a) $f(r_L) < 0 < f(r_H)$,

b) $f'(r) > 0$,

c) $f''(r) > 0$,

d) $f(r) = 0$.

Theorem C.2 (From (Meek & Walton 2004))**.** *Given a function $f(x)$ on standard form. If $f(x)$ changes sign over the interval $[r_L, r_H]$, define the widened interval $[a, b]$, where $a = r_L - h$, $b = r_H - h$, and $h = r_H - r_L$. If $f'(x) > 0$ and $f''(x) > 0$ over the interval $[a, b]$, and if $2h(3f''_{max} - f''_{min}) < f'_{min}$, where f'_{min}, f''_{min}, and f''_{max} are smallest and largest values on $[a, b]$, then Newton's root-finding method converges to the unique root of $f(x) = 0$ in $[a, b]$ for any starting value in $[a, b]$.*

Proof. See (Meek & Walton 2004). □

C.5 Conditions for convergence of Newton's root-finding method

To guarantee that Newton's root-finding method converges to the root, the conditions presented in Theorem C.1 must be satisfied. When dealing with the Fresnels integrals, the last condition of this theorem may not be satisfied (Meek & Walton 2004). Thus, a widened interval must be used to guarantee the last condition. Now, we restate the Fresnels integrals and find upper and lower limits for $\sin(\vartheta)$ and $\cos(\vartheta)$ for later use.

From (3.35) we have

$$C(\vartheta) = \int_0^{\vartheta} \frac{\cos(u^2)}{\sqrt{u}} du, \tag{C.33a}$$

$$S(\vartheta) = \int_0^{\vartheta} \frac{\sin(u^2)}{\sqrt{u}} du, , \tag{C.33b}$$

and the reduced feasible interval for the transition phases

$$\vartheta \in \left\langle 0, \frac{\pi}{4} \right] . \tag{C.34}$$

The derivatives with respect to ϑ are

$$C'(\vartheta) = \frac{\cos(\vartheta)}{\sqrt{\vartheta}} \tag{C.35a}$$

$$S'(\vartheta) = \frac{\sin(\vartheta)}{\sqrt{\vartheta}} \tag{C.35b}$$

Taylor series gives

$$\vartheta \left(1 - \frac{\vartheta^2}{6}\right) < \sin(\vartheta) < \vartheta, \tag{C.36a}$$

$$1 - \frac{\vartheta^2}{2} < \cos(\vartheta) < 1 - \frac{\vartheta^2}{2} + \frac{\vartheta^4}{24} < 1. \tag{C.36b}$$

Phase I

We have the root-function (5.59) and its first derivative (5.60)

$$g_I(\vartheta) = C(\vartheta) - \frac{x_I(e)}{a} = 0, \quad \vartheta \in \langle 0, \vartheta_0] , \tag{C.37a}$$

$$g_I'(\vartheta) = \frac{\cos(\vartheta)}{\sqrt{\vartheta}}, \tag{C.37b}$$

and the second derivative

$$g_I''(\vartheta) \;=\; -\frac{\sin(\vartheta)}{\sqrt{\vartheta}} - \frac{1}{2\vartheta}\frac{\cos(\vartheta)}{\sqrt{\vartheta}}. \tag{C.37c}$$

To be able to use Theorem C.2, the root-function must be on standard form according to Lemma C.1. Hence, $g_I(\vartheta)$ must be reflected across both axes to get $g_I' > 0$ and $g_I'' > 0$. The second derivative becomes:

$$g_I''(\vartheta) \;=\; \frac{\sin(\vartheta)}{\sqrt{\vartheta}} + \frac{1}{2\vartheta}\frac{\cos(\vartheta)}{\sqrt{\vartheta}}. \tag{C.38}$$

We need to show that both the first and second derivatives in fact satisfy $g_I' > 0$ and $g_I'' > 0$.

Estimates of $g_I'(\vartheta)$ and $g_I''(\vartheta)$ Substituting the lower limit of (C.36b) into (C.37b) we get

$$\frac{1-\frac{\vartheta^2}{2}}{\sqrt{\vartheta}} < \;\; g_I'(\vartheta) \;\; , \vartheta \in \left\langle 0, \frac{\pi}{4}\right] \tag{C.39}$$

and can conclude that the first derivative satisfies the positivity condition of Lemma C.1.

Next, we use (C.36) in (C.38)

$$\begin{aligned}
\frac{\vartheta\left(1-\frac{\vartheta^2}{6}\right)}{\sqrt{\vartheta}} + \frac{1}{2\vartheta}\frac{1-\frac{\vartheta^2}{2}}{\sqrt{\vartheta}} &< \;\; g_I''(\vartheta) \;\; < \frac{\vartheta}{\sqrt{\vartheta}} + \frac{1}{2\vartheta\sqrt{\vartheta}} \\
\frac{2\vartheta^2\left(1-\frac{\vartheta^2}{6}\right)+1-\frac{\vartheta^2}{2}}{2\vartheta\sqrt{\vartheta}} &< \;\; g_I''(\vartheta) \;\; < \frac{2\vartheta^2+1}{2\vartheta\sqrt{\vartheta}} \\
0 < \frac{1+\vartheta^2\left(\frac{3}{2}-\frac{\vartheta^2}{3}\right)}{2\vartheta\sqrt{\vartheta}} &< \;\; g_I''(\vartheta) \;\; < \frac{1+2\vartheta^2}{2\vartheta\sqrt{\vartheta}},
\end{aligned} \tag{C.40}$$

which is lower bounded for $\vartheta \in \left\langle 0, \frac{\pi}{4}\right]$.

We have shown that the conditions in Lemma C.1 are satisfied and the widening technique of Theorem C.2 can be used.

The widened interval The upper and lower bound of (C.33a) can be found by using (C.36b) in the integral

$$1 - \frac{\vartheta^2}{10} < \frac{C(\vartheta)}{2\sqrt{\vartheta}} < 1, \tag{C.41}$$

further, using that (C.37a) can be rewritten to

$$C(\vartheta) \;=\; \frac{x_I(e)}{a} \triangleq d_I, \tag{C.42}$$

we get

$$\begin{aligned} 2\sqrt{\vartheta}\overbrace{\left(1 - \frac{\vartheta^2}{10}\right)}^{\alpha_I|_{\vartheta=\frac{\pi}{4}}} < \;& d_I \;< 2\sqrt{\vartheta} \\ 2\sqrt{\vartheta}\alpha_I < \;& d_I \;< 2\sqrt{\vartheta}, \end{aligned} \tag{C.43}$$

so the upper and lower limit of ϑ becomes

$$\begin{aligned} \frac{d_I}{2} < \;& \sqrt{\vartheta} \;< \frac{d_I}{2\alpha_I} \\ r_L = \frac{d_I^2}{4} < \;& \vartheta \;< \frac{d_I^2}{4\alpha_I^2} = r_H. \end{aligned} \tag{C.44}$$

The widened interval $[a,\, b]$ which guarantees convergence can thus be found from (C.2)

$$h \;=\; r_H - r_L \tag{C.45a}$$
$$a \;=\; r_L - h = 2r_L - r_H \tag{C.45b}$$
$$b \;=\; r_H + h = 2r_H - r_L. \tag{C.45c}$$

A feasible initial choice within the interval is

$$\begin{aligned} \vartheta_{n=0} \;&=\; \frac{a+b}{2} = \frac{r_L + r_H}{2} \\ &=\; \frac{1}{2}\left(\frac{d_I^2}{4} + \frac{d_I^2}{4\alpha_I^2}\right) \\ &=\; \frac{d_I^2}{8}\left(1 + \frac{1}{\alpha_I^2}\right), \end{aligned} \tag{C.46}$$

where $\alpha_I = 1 - \frac{\pi^2}{10\cdot 16}$.

Phase III

Finding a feasible interval for ϑ_{III} can be found using the same procedure as in Section C.5.

We have the root-function (5.76) and its first derivative (5.77)

$$g_{III}(\vartheta) = S(\vartheta) - \frac{y_{III}(e)}{a} = 0, \quad \vartheta \in \langle 0, \vartheta_{III,0}], \tag{C.47a}$$

$$g'_{III}(\vartheta) = \frac{\sin(\vartheta)}{\sqrt{\vartheta}}, \tag{C.47b}$$

and the second derivative

$$g''_{III}(\vartheta) = \frac{\cos(\vartheta)}{\sqrt{\vartheta}} - \frac{1}{2\vartheta}\frac{\sin(\vartheta)}{\sqrt{\vartheta}}. \tag{C.47c}$$

Estimates of $g'_{III}(\vartheta)$ and $g''_{III}(\vartheta)$ Substituting the lower limit of (C.36a) into (C.47b) we get

$$\begin{aligned} \frac{\vartheta\left(1-\frac{\vartheta^2}{6}\right)}{\sqrt{\vartheta}} &< g'_{III}(\vartheta) \quad , \vartheta \in \left\langle 0, \frac{\pi}{4}\right] \\ \sqrt{\vartheta}\left(1-\frac{\vartheta^2}{6}\right) &< g'_{III}(\vartheta) \end{aligned} \tag{C.48}$$

and can conclude that the first derivative satisfies the positivity condition of Lemma C.1.

Furthermore, the second derivative (C.38) can be upper and lower bounded with the Taylor series (C.36)

$$\begin{aligned} \frac{1}{\sqrt{\vartheta}}\left(1-\frac{\vartheta^2}{2}-\frac{1}{2\vartheta}\vartheta\right) < \; & g''_{III}(\vartheta) \; < \frac{1}{\sqrt{\vartheta}}\left(1-\frac{1}{2\vartheta}\vartheta\left(1-\frac{\vartheta^2}{6}\right)\right) \\ \frac{1}{2\sqrt{\vartheta}}(1-\vartheta^2) < \; & g''_{III}(\vartheta) \; < \frac{1}{2\sqrt{\vartheta}}\left(1+\frac{\vartheta^2}{6}\right) \end{aligned} \tag{C.49}$$

which is lower bounded for $\vartheta \in \left\langle 0, \frac{\pi}{4}\right]$.

Once more, we have shown that the conditions in Lemma C.1 are satisfied and the widening technique of Theorem C.2 can be used.

The widened interval The upper and lower bound of (C.33b) can by found by using (C.36a) in the integral

$$\frac{\vartheta}{3}\left(1-\frac{\vartheta^2}{14}\right) < \frac{S(\vartheta)}{2\sqrt{\vartheta}} < \frac{\vartheta}{3}. \tag{C.50}$$

For simplicity, we rewrite (C.47a) to

$$S(\vartheta) \;=\; \frac{y_{III}(e)}{a} \triangleq d_{III}, \tag{C.51}$$

and can write

$$\begin{aligned} \frac{2}{3}\vartheta\sqrt{\vartheta}\overbrace{\left(1-\frac{\vartheta^2}{14}\right)}^{\alpha_{III}|_{\vartheta=\frac{\pi}{4}}} < \;& d_{III} \;< \frac{2}{3}\vartheta\sqrt{\vartheta} \\ \frac{2}{3}\vartheta^{3/2}\alpha_{III} < \;& d_{III} \;< \frac{2}{3}\vartheta^{3/2}, \end{aligned} \tag{C.52}$$

so the upper and lower limit of ϑ becomes

$$\begin{aligned} \frac{3}{2}d_{III} < \;& \vartheta^{3/2} \;< \frac{3d_{III}}{2\alpha_{III}} \\ r_L = \left(\frac{3d_{III}}{2}\right)^{2/3} < \;& \vartheta \;< \left(\frac{3d_{III}}{2\alpha_{III}}\right)^{2/3} = r_H. \end{aligned} \tag{C.53}$$

The widened interval $[a,\, b]$ which guarantees convergence can thus be found from (C.2)

$$h \;=\; r_H - r_L \tag{C.54a}$$

$$a \;=\; r_L - h = 2r_L - r_H \tag{C.54b}$$

$$b \;=\; r_H + h = 2r_H - r_L. \tag{C.54c}$$

A feasible initial choice within the interval is

$$\begin{aligned} \vartheta_{n=0} \;&=\; \frac{a+b}{2} = \frac{r_L + r_H}{2} \\ &=\; \frac{1}{2}\left(\left(\frac{3d_{III}}{2}\right)^{2/3} + \left(\frac{3d_{III}}{2\alpha_{III}}\right)^{2/3}\right) \\ &=\; \frac{1}{2}\left(\frac{3d_{III}}{2}\right)^{2/3}\left(1+\frac{1}{\alpha_{III}^{2/3}}\right), \end{aligned} \tag{C.55}$$

where $\alpha_{III} = 1 - \frac{\pi^2}{14\cdot 16}$.

C.6 The coefficients of the $\eta-$spline

The quintic $\boldsymbol{\eta}-$spline is defined by twelve polynomial coefficients. a_i and b_i, $i \in \mathcal{I}^5$ are the coefficients for the x-axis and y-axis, respectively. This result is taken from (Guarino Lo Bianco & Piazzi 2000).

$$a_0 = x_A \tag{C.56a}$$

$$a_1 = \eta_1 \cos\chi_A \tag{C.56b}$$

$$a_2 = \frac{1}{2}(\eta_3 \cos\chi_A - \eta_1^2 \varkappa_A \sin\chi_A) \tag{C.56c}$$

$$\begin{aligned} a_3 = {} & 10(x_B - x_A) - (6\eta_1 + \frac{3}{2}\eta_3)\cos\chi_A - (4\eta_2 - \frac{1}{2}\eta_4)\cos\chi_B \\ & + \frac{3}{2}\eta_1^2 \varkappa_A \sin\chi_A - \frac{1}{2}\eta_2^2 \varkappa_B \sin\chi_B \end{aligned} \tag{C.56d}$$

$$\begin{aligned} a_4 = {} & -15(x_B - x_A) + (8\eta_1 + \frac{3}{2}\eta_3)\cos\chi_A + (7\eta_2 - \eta_4)\cos\chi_B \\ & - \frac{3}{2}\eta_1^2 \varkappa_A \sin\chi_A + \eta_2^2 \varkappa_B \sin\chi_B \end{aligned} \tag{C.56e}$$

$$\begin{aligned} a_5 = {} & 6(x_B - x_A) - (3\eta_1 + \frac{1}{2}\eta_3)\cos\chi_A - (3\eta_2 - \frac{1}{2}\eta_4)\cos\chi_B \\ & + \frac{1}{2}\eta_1^2 \varkappa_A \sin\chi_A - \frac{1}{2}\eta_2^2 \varkappa_B \sin\chi_B \end{aligned} \tag{C.56f}$$

$$b_0 = y_A \tag{C.57a}$$

$$b_1 = \eta_1 \sin\chi_A \tag{C.57b}$$

$$b_2 = \frac{1}{2}(\eta_3 \sin\chi_A + \eta_1^2 \varkappa_A \cos\chi_A) \tag{C.57c}$$

$$\begin{aligned} b_3 = {} & 10(y_B - y_A) - (6\eta_1 + \frac{3}{2}\eta_3)\sin\chi_A - (4\eta_2 - \frac{1}{2}\eta_4)\sin\chi_B \\ & - \frac{3}{2}\eta_1^2 \varkappa_A \cos\chi_A + \frac{1}{2}\eta_2^2 \varkappa_B \cos\chi_B \end{aligned} \tag{C.57d}$$

$$\begin{aligned} b_4 = {} & -15(y_B - y_A) + (8\eta_1 + \frac{3}{2}\eta_3)\sin\chi_A + (7\eta_2 - \eta_4)\sin\chi_B \\ & + \frac{3}{2}\eta_1^2 \varkappa_A \cos\chi_A - \eta_2^2 \varkappa_B \cos\chi_B \end{aligned} \tag{C.57e}$$

$$\begin{aligned} b_5 = {} & 6(y_B - y_A) - (3\eta_1 + \frac{1}{2}\eta_3)\sin\chi_A - (3\eta_2 - \frac{1}{2}\eta_4)\sin\chi_B \\ & - \frac{1}{2}\eta_1^2 \varkappa_A \cos\chi_A + \frac{1}{2}\eta_2^2 \varkappa_B \cos\chi_B \end{aligned} \tag{C.57f}$$

C.7 The algorithm for a $\mathcal{G}^2$-continuous Catmull-Rom spline

First, some precomputations are needed. No explanations are given. Consult (DeRose & Barsky 1988).

Algorithm 5 Precompute($\boldsymbol{V}_0, \cdots \boldsymbol{V}_m, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2$)

$\beta_{1,0} \leftarrow \beta_{1,m} \leftarrow 1$
$\beta_{2,0} \leftarrow \beta_{2,m} \leftarrow 0$
for $i = 1$ to $m-1$ **do**
  $\gamma_{2,i} \leftarrow \frac{1+\beta_{1,i}}{\beta_{2,i}+\beta_{1,i}(1+\beta_{1,i})}$
  $\gamma_{3,i} \leftarrow \frac{2(1+\beta_{1,i})}{\beta_{2,i}+2\beta_{1,i}(1+\beta_{1,i})}$
end for
for $i = 0$ to $m-2$ **do**
  $\boldsymbol{Pb}_{i,0,2} \leftarrow \boldsymbol{Pb}_{i,1,0} \leftarrow \boldsymbol{V}_i$
  $\boldsymbol{Pb}_{i,1,2} \leftarrow \boldsymbol{Pb}_{i,2,0} \leftarrow \boldsymbol{V}_{i+1}$
  $\boldsymbol{Pb}_{i,2,2} \leftarrow \boldsymbol{Pb}_{i,3,0} \leftarrow \boldsymbol{V}_{i+2}$
  $\boldsymbol{Pb}_{i,1,1} \leftarrow \frac{\beta_{1,i+1}^2\gamma_{2,i+1}\boldsymbol{V}_i+(1+\gamma_{2,i+1})(1+\beta_{1,i+1})\boldsymbol{V}_{i+1}-\gamma_{2,i+1}\boldsymbol{V}_{i+2}}{(1+\beta_{1,i+1})(1+\beta_{1,i+1}\gamma_{2,i+1})}$
  $\boldsymbol{Pb}_{i,2,1} \leftarrow \boldsymbol{V}_{i+1} + \beta_{1,i+1}(\boldsymbol{V}_{i+1} - \boldsymbol{Pb}_{i,1,1})$
  $\boldsymbol{Pb}_{i,0,1} \leftarrow \boldsymbol{V}_i + \frac{1}{\beta_{1,i}}(\boldsymbol{V}_i - \boldsymbol{Pb}_{i,1,1})$
  $\boldsymbol{T}_1 \leftarrow \boldsymbol{Pb}_{i,1,1} + \gamma_{2,i}(\boldsymbol{Pb}_{i,1,1} - \boldsymbol{V}_{i+1})$
  $\boldsymbol{Pb}_{i,0,0} \leftarrow \boldsymbol{Pb}_{i,0,1} + \frac{1}{\beta_{1,i}^2\gamma_{2,i}}(\boldsymbol{Pb}_{i,0,1} - \boldsymbol{T}_1)$
  $\boldsymbol{Pb}_{i,3,1} \leftarrow \boldsymbol{V}_{i+2} + \beta_{1,i+2}(\boldsymbol{V}_{i+2} - \boldsymbol{Pb}_{i,2,1})$
  $\boldsymbol{T}_2 \leftarrow \boldsymbol{Pb}_{i,2,1} + \beta_{1,i+2}\gamma_{2,i+1}(\boldsymbol{Pb}_{i,2,1} - \boldsymbol{V}_{i+1})$
  $\boldsymbol{Pb}_{i,3,2} \leftarrow \boldsymbol{Pb}_{i,3,1} + \frac{1}{\gamma_{2,i+2}}(\boldsymbol{Pb}_{i,3,1} - \boldsymbol{T}_2)$
end for

After the precomputations, control polygons for the Bézier curve segments can be created. Curve segments indexed within $q = 2, \cdots, m-3$ is computed through the following algorithm:

Algorithm 6 ConstructBézierPolygon($\boldsymbol{V}_0, \cdots \boldsymbol{V}_m, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, q$)

for $r = 0$ to 3 **do**
  $\boldsymbol{R}_{r,0} \leftarrow \boldsymbol{Pb}_{q+r-2,3-r,0}$
  $\boldsymbol{R}_{r,1} \leftarrow \boldsymbol{Pb}_{q+r-2,3-r,1}$
  $\boldsymbol{R}_{r,2} \leftarrow \boldsymbol{Pb}_{q+r-2,3-r,2}$
end for
for $c = 0$ to 2 **do**
  $\boldsymbol{T}_1 \leftarrow \frac{\beta_{1,q}^2 \gamma_{3,q} \boldsymbol{R}_{0,c} + (1+\gamma_{3,q-1}) \boldsymbol{R}_{1,c}}{1+\gamma_{3,q-1}+\beta_{1,q}^2 \gamma_{3,q}}$
  $\boldsymbol{S}_{1,c} \leftarrow \frac{(1+\beta_{1,q+1}^2 \gamma_{3,q+1}) \boldsymbol{R}_{1,c} + \gamma_{3,q} \boldsymbol{R}_{2,c}}{1+\gamma_{3,q}+\beta_{1,q+1}^2 \gamma_{3,q+1}}$
  $\boldsymbol{S}_{2,c} \leftarrow \frac{\beta_{1,q+1}^2 \gamma_{3,q+1} \boldsymbol{R}_{1,c} + (1+\gamma_{3,q}) \boldsymbol{R}_{2,c}}{1+\gamma_{3,q}+\beta_{1,q+1}^2 \gamma_{3,q+1}}$
  $\boldsymbol{T}_2 \leftarrow \frac{(1+\beta_{1,q+2}^2 \gamma_{3,q+2}) \boldsymbol{R}_{2,c} + \gamma_{3,q+1} \boldsymbol{R}_{3,c}}{1+\gamma_{3,q+1}+\beta_{1,q+2}^2 \gamma_{3,q+2}}$
  $\boldsymbol{S}_{0,c} \leftarrow \frac{\beta_{1,q} \boldsymbol{T}_1 + \boldsymbol{S}_{1,c}}{1+\beta 1,q}$
  $\boldsymbol{S}_{3,c} \leftarrow \frac{\beta_{1,q+1} \boldsymbol{S}_{2,c} + \boldsymbol{T}_2}{1+\beta_{1,q+1}}$
end for
$\boldsymbol{Q}_0 \leftarrow \boldsymbol{S}_{0,0}$
$\boldsymbol{Q}_1 \leftarrow \frac{3\boldsymbol{S}_{1,0}+2\boldsymbol{S}_{0,1}}{5}$
$\boldsymbol{Q}_2 \leftarrow \frac{3\boldsymbol{S}_{2,0}+6\boldsymbol{S}_{1,1}+\boldsymbol{S}_{0,2}}{10}$
$\boldsymbol{Q}_3 \leftarrow \frac{\boldsymbol{S}_{3,0}+6\boldsymbol{S}_{2,1}+3\boldsymbol{S}_{1,2}}{10}$
$\boldsymbol{Q}_4 \leftarrow \frac{2\boldsymbol{S}_{3,1}+3\boldsymbol{S}_{2,2}}{5}$
$\boldsymbol{Q}_5 \leftarrow \boldsymbol{S}_{3,2}$
return $(\boldsymbol{Q}_0, \cdots, \boldsymbol{Q}_5)$

C.8 Cascaded nonlinear time-varying systems (NLTV)

This material is taken from (Lamnabhi-Lagarrigue et al. 2005). A cascaded NLTV can be written in the form

$$\Sigma_1 \quad : \quad \dot{\boldsymbol{x}}_1 = f_1(t, \boldsymbol{x}_1) + g(t, \boldsymbol{x})\boldsymbol{x}_2 \tag{C.58a}$$

$$\Sigma_2 \quad : \quad \dot{\boldsymbol{x}}_2 = f_2(t, \boldsymbol{x}_2), \tag{C.58b}$$

where $\boldsymbol{x}_1 \in \mathbb{R}^n$, $\boldsymbol{x}_2 \in \mathbb{R}^m$, $\boldsymbol{x} \triangleq [\boldsymbol{x}_1\, \boldsymbol{x}_2]^\top$. Further, $f_1(\cdot,\cdot)$, $f_2(\cdot,\cdot)$, $g(\cdot,\cdot)$ are continuous, locally Lipschits in $\boldsymbol{x}$, uniformly in t, and $f_1(\cdot,\cdot)$ is continuously differentiable in both arguments. There must also exist a nondecreasing function $G(\cdot)$ such that

$$|g(t, \boldsymbol{x})| \leq G(|\boldsymbol{x}|). \tag{C.59}$$

In (Lamnabhi-Lagarrigue et al. 2005) sufficient conditions is outlined for the subsystem

$$\dot{\boldsymbol{x}}_1 \quad = \quad f_1(t, \boldsymbol{x}_1) \tag{C.60}$$

to ramain UGAS (or UGES) when it is perturbed by the output of another UGAS (or UGES) system on the form Σ_2.

We now restate the relevant assumptions needed to guarantee stability for the cascaded system developed in this thesis.

Assumption C.1.

a) (C.60) *is UGAS (or UGES)*

b) *There exist a known* $\mathcal{C}^1$ *Lyapunov function* $V(t, \boldsymbol{x}_1)$, α_1, $\alpha_2 \in \mathcal{K}_\infty$, *a positive semidefinite function* $W(\boldsymbol{x}_1)$, *a continuous non-decreasing function* α_4, *such that*

$$\alpha_1(|\boldsymbol{x}_1|) \leq V(t, \boldsymbol{x}_1) \leq \alpha_2(|\boldsymbol{x}_1|) \tag{C.61}$$

$$\dot{V}_{(\text{C.60})}(t, \boldsymbol{x}_1) \leq -W(\boldsymbol{x}_1) \tag{C.62}$$

$$\left|\frac{\delta V}{\delta \boldsymbol{x}_1}\right| \leq \alpha_4(|\boldsymbol{x}_1|). \tag{C.63}$$

Assumption C.2. *The subsystem Σ_2 is UGAS (or UGES).*

Assumption C.3. *There exist a continuous non-decreasing function $\alpha_6 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and a constant $a \geq 0$, such that $\alpha_6(a) > 0$ and*

$$\alpha_6(s) \geq \alpha_4(\alpha_1^{-1}(s))\alpha_5(\alpha_1^{-1}(s)), \tag{C.64}$$

where α_5 is taken from

$$|g(t, \boldsymbol{x}(t; t_\circ, \boldsymbol{x}_\circ))| \leq c_g(r)\alpha_5(|\boldsymbol{x}_1(t; t_\circ, \boldsymbol{x}_\circ)|),\ \forall |\boldsymbol{x}_{2,\circ}| < r,\ \forall t \geq t_\circ,\ r > 0 \tag{C.65}$$

$$c_g(\cdot) \triangleq \theta_1(\beta(\cdot, 0)), \text{ a class } \mathcal{K} \text{ function}, \tag{C.66}$$

and

$$\int_a^\infty \frac{ds}{\alpha_6(s)} = \infty. \tag{C.67}$$

Assumption C.4. *The function $g(t, \boldsymbol{x})$ is majorized by the function $f_1(t, \boldsymbol{x}_1)$ in the following sense: for each $\boldsymbol{r} > \boldsymbol{0}$ there exist $\lambda > 0$, $\boldsymbol{\eta} > \boldsymbol{0}$ such that, for all $t \geq 0$ and all $|\boldsymbol{x}_2| < \boldsymbol{r}$*

$$\left| \frac{\delta V}{\delta \boldsymbol{x}} g(t, \boldsymbol{x}) \right| \leq \lambda W(\boldsymbol{x}_1), \quad |\boldsymbol{x}_1| \geq \boldsymbol{\eta}, \tag{C.68}$$

where $W(\boldsymbol{x}_1)$ is defined in Assumption C.1.

Theorem C.3. *If Assumptions C.1-C.4 hold, the cascade* (C.58) *is UGAS (or UGES) because $f_1(t, \boldsymbol{x}_1)$ majorizes $g(t, \boldsymbol{x})$.*

C.9 Local stability of the path-tracking controller

We have the system (5.128)

$$\dot{\tilde{s}}_p = -U_{\mathrm{a,max}} \frac{\tilde{s}_p}{\sqrt{\tilde{s}_p^2 + \Delta_s^2}} = f(\tilde{s}_p). \tag{C.69}$$

Let us linearize around the origin to determine the stability close to this equilibrium. First, find the Taylor series linearization around the solution:

$$\begin{aligned}\left.\frac{df}{d\tilde{s}_p}\right|_{\tilde{s}_p=0} &= \left.\frac{-U_{\mathrm{a,max}}\sqrt{\tilde{s}_p^2+\Delta_s^2}-\frac{1}{4\sqrt{\tilde{s}_p^2+\Delta_s^2}}}{\tilde{s}_p^2+\Delta_s^2}\right|_{\tilde{s}_p=0} \\ &= \frac{-U_{\mathrm{a,max}}\Delta_s-\frac{1}{4\Delta_s}}{\Delta_s^2} \\ &= -\left(\frac{\Delta_s^2 U_{\mathrm{a,max}}}{\Delta_s}+\frac{1}{4\Delta_s^3}\right). \end{aligned} \tag{C.70}$$

We get the linearized system

$$\begin{aligned}\Delta\dot{\tilde{s}}_p &= \left.\frac{df}{d\tilde{s}_p}\right|_{\tilde{s}_p=0}\Delta\tilde{s}_p \\ &= \underbrace{-\left(\frac{\Delta_s^2 U_{\mathrm{a,max}}}{\Delta_s}+\frac{1}{4\Delta_s^3}\right)}_{A}\Delta\tilde{s}_p, \end{aligned} \tag{C.71}$$

where A clearly is Hurwitz; that is $A<0$, and hence the system is LES.

C.10 Lawn-mower path planner

C.10.1 Newton's method for case B

Newton's method is explained in Appendix C.4. For Newton's method to be applicable, Theorem C.1 must be valid. We will now verify most of this conditions.

The root function is (5.171)

$$g_B(\vartheta) = S(\vartheta)+C'(\vartheta)-\frac{d}{2a}, \tag{C.72}$$

ands its first derivative is

$$g_B'(\vartheta) = S'(\vartheta)+C''(\vartheta) \tag{C.73}$$

$$= -\frac{C'(\vartheta)}{2\vartheta}, \tag{C.74}$$

which is negative for all $\vartheta \in \left\langle 0,\, \frac{\pi}{2} \right]$.

The second derivative is

$$g''_B(\vartheta) = -(C''(\vartheta)2\vartheta - 2C'(\vartheta))\frac{1}{4\vartheta^2} \tag{C.75}$$

$$= \frac{3C'(\vartheta)}{4\vartheta^2} + \frac{S'(\vartheta)}{2\vartheta}, \tag{C.76}$$

which is non-negative in $\vartheta \in \left\langle 0,\, \frac{\pi}{2} \right]$.

These results verify that the conditions a), b) and c) of Theorem C.1 are met. As for condition d), linearization of the expression and considering these bounds must be done according to (Meek & Walton 2004). This work is tedious and considered outside the scope of this thesis. It is however believed that condition d) in fact can be met. To compensate for the lack of an approximate initial guess, an increased number of iterations can be performed with $\vartheta_0 = 10^{-6}$, which is applicable in most practical cases.

C.10.2 Newton's method for case C

Newton's method is explained in Appendix C.4. For Newton's method to be applicable, Theorem C.1 must be valid. We will now verify most of this conditions.

The root function is (5.186)

$$\begin{aligned} g_C(\phi) = {} & [S(\phi/2) + \sin(\phi)\,(C(\phi/2) + C(\vartheta_c) - S'(\vartheta_c)) \\ & - \cos(\phi)\,(S(\phi/2) + S(\vartheta_c) + C'(\vartheta_c))] + \frac{d}{2a}, \end{aligned} \tag{C.77}$$

ands its first derivative is

$$\begin{aligned} g'_C(\phi) = {} & S'(\phi/2)\,(1 - \cos(\phi)) + C'(\phi/2)\sin(\phi) \\ + {} & \cos(\phi)\,(C(\phi/2) + C(\vartheta_c) - S'(\vartheta_c)) \\ + {} & \sin(\phi)\,(S(\phi/2) + S(\vartheta_c) + C'(\vartheta_c)) \end{aligned} \tag{C.78}$$

which can be shown to be positive for all $\operatorname{col}[\vartheta_c,\, \varphi] \in \langle 0,\, \pi] \times \left\langle 0,\, \frac{\pi}{2} \right]$, by inspecting each term separately.

The second derivative is

$$\begin{aligned} g_C''(\phi) &= \frac{\cos(\phi/2)}{\sqrt{\phi/2}}\left(\frac{3}{4}\cos(\phi) - \frac{1}{\phi}\sin(\phi) + \frac{1}{4}\right) \\ &+ \frac{\sin(\phi/2)}{\sqrt{\phi/2}}\left(\frac{3}{4}\sin(\phi) - \frac{1}{\phi}(1-\cos(\phi))\right) \\ &- \sin(\phi)(C(\phi/2) + C(\vartheta_c) - S'(\vartheta_c)) \\ &+ \cos(\phi)(S(\phi/2) + S(\vartheta_c) + C'(\vartheta_c)), \end{aligned} \tag{C.79}$$

which is non-trivial to examine. Fortunately, we know that the x-coordinate of the circle center always lies to the left of the initial parallel line (in the nominal case used to deduce the equations) and hence

$$M_{c,x}^{\star}(\phi, \vartheta_c) \triangleq g_c(\phi) - \frac{d}{2a} \leq 0, \, \forall \, \mathrm{col}\,[\phi, \vartheta_c] \in \mathcal{I}_C. \tag{C.80}$$

To prove that $g_C''(\phi, \vartheta_c)$ is positive in the area of interest, that is, $\mathrm{col}\,[\phi, \vartheta_c] \in \mathcal{I}_C$, several approaches apply. It can be proven that

$$M_{c,x}^{\star}(\phi, \vartheta_c) + g_C''(\phi, \vartheta_c) > 0 \forall \mathrm{col}(\phi, \vartheta_c) \in \langle 0, \, \pi/2] \times \langle 0, \, \pi] \,, \tag{C.81}$$

which is sufficient, since $\mathcal{I}_c \subset \langle 0, \, \pi/2] \times \langle 0, \, \pi]$. However, this proof is very tedious, so a graphical interpretation will be given instead. Consider Figure C.2, where the second derivative of the root function (C.79), and a scaled version of the x-coordinate of the circle center (C.80) are plotted together. It can be seen that whenever (C.80) is negative, equation (C.79) is positive, and hence condition c) of Theorem C.1 applies.

We have shown that the conditions a), b) and c) of Theorem C.1 are met. As for condition d), linearization of the expression and considering these bounds must be done according to (Meek & Walton 2004). This work is tedious and considered outside the scope of this thesis. It is however believed that condition d) in fact can be met. To compensate for the lack of an approximate initial guess, an increased number of iterations can be performed with $\vartheta_0 = 10^{-6}$, which is applicable in most practical cases. Rather than trying to find a feasible initial condition, a different root finding method should seriously be considered. This problem will not be considered here.

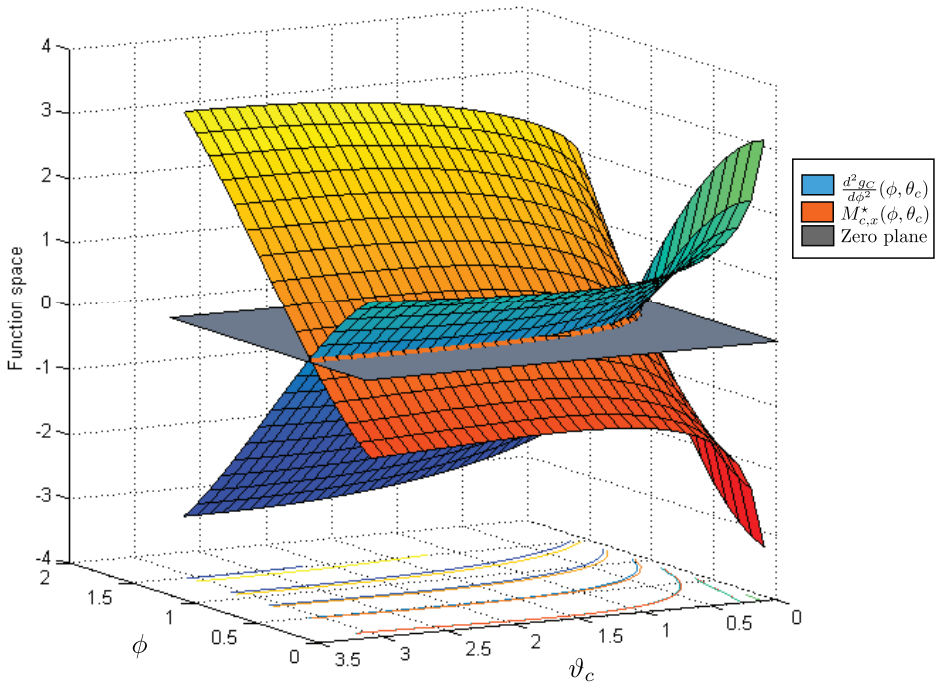


Figure C.2: The plot shows that the second derivative is positive whenever the scaled x-coordinate of the circle is negative.