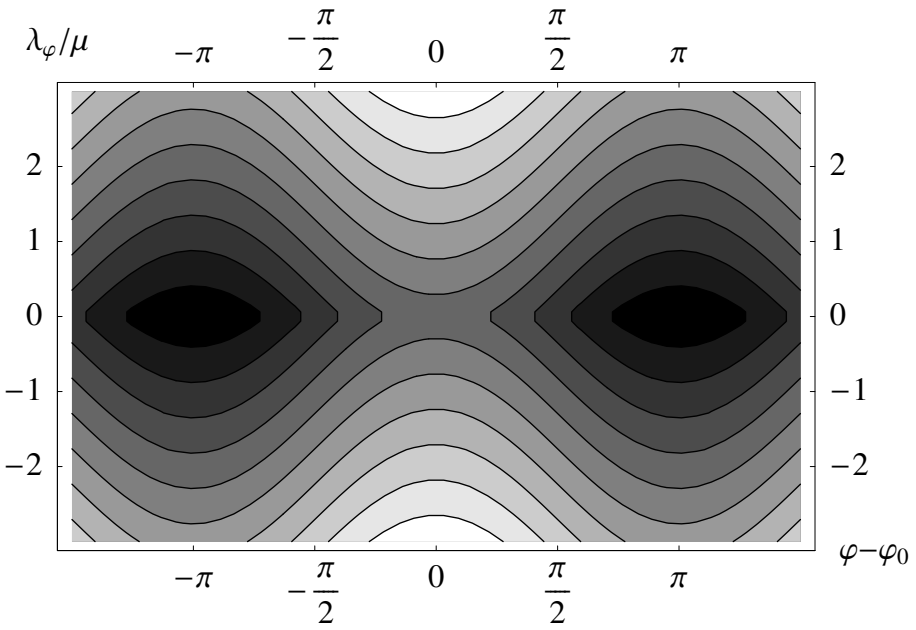
FOI-R--0961--SE
July 2003
1650-1942

Scientific Report

David A. Anisi

Optimal Motion Control of a Ground Vehicle



System Technology Division
SE-172 90 STOCKHOLM
Sweden

Swedish Defence Research Agency
System Technology Division
SE-172 90 STOCKHOLM
Sweden

FOI-R--0961--SE
July 2003
1650-1942

Scientific Report

David A. Anisi

Optimal Motion Control of a Ground Vehicle

Issuing organization	Report number, ISRN	Report type
Swedish Defence Research Agency System Technology Division SE-172 90 STOCKHOLM Sweden	FOI-R--0961--SE	Scientific Report
	Research area code: Combat	
	Month year: July 2003	Project no.: E6003
	Customers code: Commissioned Research	
	Sub area code: Weapons and Protection	
Author/s (editor/s): David A. Anisi	Project manager: Peter Alvå	
	Approved by: Monica Dahlén	
	Sponsoring agency: Swedish Armed Forces	
	Scientifically and technically responsible: John W.C. Robinson	

Report title

Optimal Motion Control of a Ground Vehicle

Abstract

This report investigates how Optimal Control and above all, the Pontryagin Maximum Principle (PMP), can be used to find algorithms for steering a mobile platform between two pre-specified points in the state space.

First the problem of finding time-optimal paths for Dubins' as well as Reeds-Shepp's car models is considered. They turn out to be the concatenation of circular arcs of maximum curvature and straight line segments, all tangentially connected (i.e. essentially bang-bang solutions). For each of these car models, a sufficient family of paths $\mathcal{F}$, is presented. In addition, an algorithm for synthesizing the optimal path, i.e. to pinpoint the global optimum inside $\mathcal{F}$, is provided. The nature of the presented algorithm is geometrical, which makes it highly suitable for numerical computations.

We then consider the problem of generating smoother, more flexible and pliable paths. This helps us to reduce the impairment of the steering device, but also raises the robustness with respect to any possible uncertainties. Despite some rewarding simulation results, the presented concept turns out to suffer from severe numerical instability properties. Upon a more careful investigation, it turns out that the problem at hand is singular. Finally, in an effort to reduce the numerical difficulties, an alternative approach, *viz.* the Method of Perturbation is adopted. Taking the synthesized time-optimal paths as the starting point, the idea is to study how a small change in the design parameter influences the generated paths. However, this approach turns out to be numerically divergent as well.

Keywords

Unmanned Ground Vehicles, Optimal Control, Motion Planning.

Further bibliographic information	Language
	English
ISSN: 1650-1942	Pages: 74
Distribution: By sendlist	Price Acc. to pricelist Security classification Unclassified

Utgivare	Rapportnummer, ISRN	Klassificering
Totalförsvarets forskningsinstitut Avdelningen för Systemteknik SE-172 90 STOCKHOLM Sweden	FOI-R--0961--SE	Vetenskaplig rapport
	Forskningsområde: Bekämpning	
	Månad, år: Juli 2003	Projektnummer: E6003
	Verksamhetsgren: Uppdragsfinansierad verksamhet	
	Delområde: VVS med styrda vapen	

Författare/redaktör	Projektledare
David A. Anisi	Peter Alvå
	Godkänd av: Monica Dahlén
	Uppdragsgivare/kundbeteckning: Försvarsmakten
	Tekniskt och/eller vetenskapligt ansvarig: John W.C. Robinson

Rapportens titel

Optimal styrning av markfordon

Sammanfattning

Denna rapport undersöker möjligheten att tillämpa optimal styrning och då framförallt, Pontryagins Maximumprincip (PMP), till att framställa styrlagar som styr en mobil plattform mellan fördefinierade punkter i tillståndsrummet.

Först riktar vi uppmärksamheten mot Dubins och Reeds-Shepps problem och studerar hur tidsoptimala banor för dessa två bilmodeller ser ut. Dessa banor visar sig bestå av cirkelbågar med maximal krökning och raka linjer som tangerar varandra i beröringspunkterna. Tillräckliga familjen av banor, $\mathcal{F}$, presenteras för var och en av dessa två bilmodeller. Dessutom presenteras en algoritm för att hitta den tidsoptimala banan inuti $\mathcal{F}$. Algoritmen bygger på rent geometriska resonemang vilket gör den synnerligen lämplig för numeriska tillämpningar.

Därefter studerar vi möjligheten att generera mjukare och mer flexibla banor. Detta för att minska slitaget på styranordningen och höja robustheten för eventuella osäkerheter i omvärldsbilden. Trots en del uppmuntrande simuleringsresultat, visar sig vårt tillvägagångsätt vara numeriskt instabilt. Efter en noggrann undersökning, sluter vi oss till att det studerade problemet är singulärt. Slutligen, för att undvika de numeriska hindren, begagnar vi störningsräkning. Med utgångspunkt i de tidsoptimala lösningarna, studerar vi hur en liten förändring i design parametern (och därmed i hamiltonianen), visar sig i termer av de genererade banorna. Dock divergerar även denna metod numeriskt.

Nyckelord

Obemmanade markfordon, optimal reglering, banplanering.

Övriga bibliografiska uppgifter	Språk
	Engelska

ISSN	Antal sidor
1650-1942	74

Distribution	Pris / Sekretess
Enligt missiv	Pris: Enligt prislista Sekretess: Öppen

Contents

Acknowledgments

This Master's thesis would not have been initialized without the hand of Associate Professor Xiaoming Hu, my supervisor at the Royal Institute of Technology (KTH), Sweden. I am indebted for this initiative.
Of the people at FOI, I especially would like to thank my supervisor Johan Hamberg, for all his support and encouragement. People that are willing to put all their time (and even sacrifice their coffee-breaks) to discuss whatever is on a student's mind, should be praised. These thanks extend to Anders Lennartsson, Petter Ögren and John Robinson for their thoughtful reading and constructive criticisms of this report.
Finally, according to Abraham Maslov's *hierarchy of needs*, self-actualization, which I believe the experience of earning a degree should be classified as, is impossible unless a person's more basic needs such as belongingness and love needs are satisfied. Therefore, it would be most ungrateful of me, not to direct a collective gratitude towards my beloved family and friends. I thank you all for being cheerful and patient all through these 17 years of education.

1. Introduction

1.1 Overview and Objectives

This report, written at the Institute of Autonomous Systems at the Swedish Defence Research Agency (FOI), is a Master's thesis, spanning over six months and finalizing a M.Sc. degree in Engineering Physics. Supervisor at FOI has been Johan Hamberg, while Xiaoming Hu has been the counterpart at the Royal Institute of Technology (KTH), from where the master of science degree is earned.

The main objective is to investigate how the Pontryagin Maximum Principle, a tool from the area of Optimal Control, can be used in order to find appropriate control laws for steering a mobile platform between two prescribed points in the state space. Any possible ambition to design an autonomous system, consisting of one single or a group of cooperating vehicles, boils down to being able to plan the motion of the system. Thus, path planning is a most fundamental issue for designing an Unmanned Ground Vehicle (UGV). This work is hence a brick in a modularized project at FOI, aiming at designing a control system, consisting of multiple UGVs.

This report presents results on how to generate time-optimal as well as feasible paths interconnecting two given states - feasible in the sense that it has to obey both the kinematic- and numerical limitations - since in autonomous robot applications, because of the partially unknown environment and unexpected events, the ability to perform path re-planning on-line is of key importance. In addition, the problem of generating nearly time-optimal, but nevertheless smoother, more flexible and pliable paths for a ground vehicle is considered. As will be illustrated, such paths raise the robustness with respect to any possible uncertainties in the environment.

1.2 Report Outline

The following is a structural outline of this report:
Chapter 2 (**Robot Motion Planning**) and chapter 3 (**Optimal Control**), present the mathematical hull and introduce the notions used in the rest of the report.

Chapters 4 and 5 are devoted to the study of **Dubins'** and **Reed-Shepp's car models** respectively. The time-optimal paths for these two car models will turn out to be the concatenation of circular arcs of maximum curvature and straight line segments, all tangentially connected (i.e. essentially bang-bang solutions). Next, the problem of generating nearly time-optimal but nevertheless smoother, more flexible and pliable paths for these systems is considered. Finding these **nearly time-optimal paths**, which is the scope of chapter 6, is carried out by making an appropriate and cunning choice for the integral cost function and, as a byproduct, provides us with a subtle and implicit way of handling input saturation.

Despite some rewarding simulation results, it turns out that the presented concept suffers from severe numerical instability properties. The origin of this undesirable behavior is located and a course of treatment is presented. Nevertheless, due to the singular property of the problem, the ambition to construct arbitrary flexible paths by means of the Pontryagin Maximum Principle must be dismissed.

Finally, in an effort to reduce the numerical difficulties that the shooting methods brings, an alternative approach, *viz.* the **Method of Perturbation**, is adopted.

Taking the synthesized time-optimal paths as the starting point, the idea is to study how a small change in the design parameter ε (and hence in the Hamiltonian function) influences the optimal solution.

In **Appendix A**, the idea behind the shooting method as well as an algorithm for using this technique is presented. In **Appendix B** the transformation of an optimal control problem with a free terminal time, into a equivalent one but defined on a fixed end-time interval, is considered.

1.3 Reader's Guide

In this section, for each and every of the chapters to come, a more detailed and descriptive disposition is provided.

Robot Motion Planning

Initially, the **basic concepts** used in the robotics community are introduced (section 2.1). Everything ranging from state variables and admissible control, to the Lie bracket and Control Lie Algebra... The notion of reachability, accessibility and controllability are also defined herewithin. This part sets the foundation for the rest of this report and should serve as a section of reference.

Next, we consider **constrained systems** and introduce the concept of *nonholonomy* (section 2.2). We establish integrability criteria for a single Pfaffian constraint and present theoretical results on how the presence of nonholonomic constraints affect the controllability properties of a control system.

Following that, the kinematic models for various **mobile platforms** are investigated (section 2.3). The so called *uni-cycle, car-like* and *front-wheeled car* models will be considered. Two conditions that restrict the kinematic abilities of these mobile platforms, namely the "rolling without slipping" and "bounded turning radius" assumptions, are imposed. We show how these two assumptions translate into a nonholonomic Pfaffian equality and an inequality constraint. In addition, the effect of them on a platform's controllability properties are investigated. In the subsection 2.3.1, we carry out the **controllability analysis** for all three car models.

Optimal Control

We start this chapter by formulating the notion of an **optimal control problem** that is general enough to correspond to the problems that will be considered in chapters 4 through 6. Next, in section 3.1, the Pontryagin Maximum principle (**PMP**) is presented for this stated optimal control problem where we also present a stepwise algorithm for applying PMP to an optimal control problem. Finally, we discuss some possible **generalizations** to the aforementioned problem and the implications they bring (section 3.2). The following generalizations are to be made:

- non-autonomous systems
- initial and/or final configurations are not specified, but are instead required to belong to smooth manifolds.
- the cost function includes, in addition to the integral term, an initial and/or final cost term.

Dubins' car

This chapter is devoted to Dubins' problem which can be described as finding the shortest continuously differentiable path between two given points taken by a car, for which the starting and ending directions are specified. In addition,

we assume that the car is moving with unit speed and subject to a minimum turning radius constraint.

First, we formulate Dubins' problem as a minimum-time problem and **characterize the optimal control** by applying PMP to it (section 4.1). Every time-optimal path will be shown to consist of circular arcs of maximum curvature and straight line segments, all tangentially connected. These basic path segments' duration and mutual order however, is a much more delicate matter. In order to restrict the candidates for optimality, a set of **sufficient family of paths**, $\mathcal{F}_D$, is presented (section 4.2). Then finally an algorithm for **synthesizing the optimal path**, i.e. to pinpoint the global optimum inside $\mathcal{F}_D$, is presented (section 4.3). The nature of the presented algorithm is geometrical, which makes it highly suitable for numerical computations. Then, by adopting preferred path following technique, the platform is to follow this synthesized path.

Reeds-Shepp's car

Reed-Shepp's problem is almost identical to that of Dubins', with one exception; Reed-Shepp's car is able to perform backwards motion as well, i.e. now the velocity $v \in \{-1, 1\}$. Consequently, the outline of this chapter is similar to the previous one. Firstly we **characterize the optimal control** (essentially bang-bang solutions), then present the **sufficient family** of paths, $\mathcal{F}_{RS}$, and finally discuss some results regarding **optimal path synthesis**.

Nearly Time-Optimal Paths

Initially, we make a remark that in those cases where we are entitled to dictate an arbitrary integral cost function, this choice should clearly be made with the some practical, control-motivated reasons in mind (section 6.1). Our objectives when making that choice, are three-fold:

- generate nearly time-optimal solutions
- obtain smoother, more flexible and pliable paths
- handle input saturation in a convenient manner.

Then, a handful of more or less fruitful trials and some rewarding simulation results are presented in section 6.2. But at the final hour, the unstable numerical properties of the problem at hand, become to tangible. Once the origin of this behavior is located (section 6.3), we may conclude that the presented concept has to be disregarded. The generated paths are too "non-optimal" to be classified as "nearly time-optimal" solutions.
Finally, in an effort to reduce the numerical difficulties that the shooting methods brings, an alternative approach, *viz.* the **Method of Perturbation**, is adopted (section 6.4). By making a Taylor expansion about the synthesized time-optimal paths, we are able to study the first order contribution from a change in the design parameter, ε, to the change in the appearance of the generated paths.

Appendix A

The idea behind the shooting method, which is the most well-tried and dependable technique for solving two point boundary values problems, is discussed in appendix A. In addition, an algorithm for adopting this technique is presented.

Appendix B

When employing numerical DE solvers in e.g. MATLAB, the duration of the time interval in which the numerical integration should dure, have to be prescribed. This naturally contradicts the fact that we have an optimal control problem with free terminal time at hand. To set this right, we utilize a time

transformation, which equalizes the free terminal time problem with a fixed end-time problem, defined on the interval [0, 1].

2. Robot Motion Planning

Robot motion planning arises as a natural and vital subproblem of the noble ambition to design an autonomous system. Once implemented, it lifts the question of robot control to a higher level, where the input descriptions will specify the nature of the task to be carried out, rather than *how* to do it. From a control perspective, if our objective is to get hold of a cold bottle of beer, we find it more desirable and convenient to simply tell the robot to 'Duff, please!' rather than passing explicit and highly detailed maneuver schemes to it. This involves generating an optimal, as well as feasible path interconnecting two given points in the state space - feasible in the sense that it has to obey both the kinematic and numerical limitations - since in autonomous robot applications, the ability to perform path re-planning on-line is of key importance.

The outline of this chapter is as follows: In section 2.1 we introduce some basic notions used in the robotics community, which also serve as foundation and source of reference, for the chapters to come. In section 2.2 we consider systems, on which a set of kinematic constraints are imposed and introduce the concept of *nonholonomic* systems. We will see that the main consequence of a nonholonomic constraint, is that an arbitrary path in the admissible state space does not necessarily correspond to a feasible trajectory for the robot. This is due to the restriction that the nonholonomic constraint imposes on the set of accessible velocities at each time instant, a fact that will have consequences for our efforts to plan the motion of the robot.
Next, in section 2.3, we present and discuss some models describing the kinematic properties of mobile platforms. The *unicycle*, *car-like* and the *front-wheeled car* models, are going to be investigated. In addition, the controllability properties of these three models will be examined.

2.1 Basic Concepts

Consider a system, fully characterized by a finite number of real variables $x_1, x_2, \ldots, x_n$, satisfying a given set of equality or inequality constraints [1]. If our system is to represent a vehicle or a rocket, these real variables might specify its position and orientation in space, but also more delicate matters, such as the amount of remaining fuel or covered distance, might occasionally be suitable to consider. These real numbers are called *state variables* or *generalized coordinates*. It is convenient to think about the state of the system as a point $x = (x_1, x_2, \ldots, x_n)$ in an n-dimensional space. A point that fulfills all the constraints on the state variables is called an admissible state. The set of all admissible states is called *state space* and denoted $\mathcal{X}$.

Next, we shall introduce the state dynamics and the control. The *state dynamics*, is a prescribed set of rules that governs the time evolution of the state variables. It indicates how the state variables will change over time, given a current state and current input. It is mathematically appealing and physically motivated, to focus our attention on systems whose behavior are governed by nonlinear ordinary differential

[1] The classification of any possible constraint and their effect on the system are to be explicitly considered in section 2.2

equations (ODEs)

$$\dot{x}(t) = f(x(t), u(t)) \tag{2.1}$$

where $f(x,u) \in \mathcal{C}^\infty$ is the *system dynamics* and $u = (u_1, u_2, \ldots, u_m)$ is the *control vector* that can be thought of as a point in an m-dimensional space, the *control space*. More generally, we shall assume that the control u, belongs to a prescribed subset $\mathcal{U} \subseteq \mathbb{R}^m$ and that the *control manifold*, $\mathcal{U}$, is constant, i.e. independent of current state x and time t. In most control problems, the constraints on the control u, commonly arise from technological limitations. For instance, a rocket's thrust magnitude as well as a car's turning radius, are bounded. This type of limitations typically restrict the control space to polyhedra or polyhedral cones in $\mathbb{R}^m$. One (often overlooked) way of handling such control constraints, is replacing $\mathcal{U}$ by a non-constrained control space. Not seldom, one encounters the following type of control constraint

$$|u_i| \leq 1, \qquad i \in \{1, 2, \ldots, m\}$$

Then by defining $u_i = \cos \tilde{u}_i$, we have moved the control into a space in which $\tilde{u}_i \in \mathbb{R}$, thus unconstrained. The price that one has to pay for using this transformation trick, is that, apart from the obvious redundancy, any possible intuitive or physical interpretation of the control input might be lost. However, in chapter 6, we will provide a more refined and implicit way of handling control constraints.
Another restriction imposed on the control is that $u(t)$ has to be a piecewise continuous function. A control that fulfills both of the criteria mentioned, i.e. belongs to the control manifold $\mathcal{U}$, and is piecewise continuous, is called an *admissible control*. Assuming that the system dynamics is a smooth function, i.e. belongs to C^∞, specifying an admissible control on a time interval $[t_i, t_f]$, corresponds to fully determining the time history of the state vector in that time interval. The time evolution of $x(t)$ is called the *path* or *state trajectory*, along which the system moves. The state trajectory is denoted $x(\cdot)$ while the corresponding control function $u(t), t \in [t_i, t_f]$, is denoted as $u(\cdot)$. If the path stays within the system's state space $\mathcal{X}$, we call it an *admissible path*.
Concerning the system dynamics particularized by equation (2.1), we shall pay extra attention to systems, governed by *control affine system dynamics* of form

$$\dot{x}(t) = f(x(t), u(t)) = \Upsilon(x(t)) \cdot u(t) + \Upsilon_0(x(t)) \tag{2.2}$$

where $\Upsilon_0(x(t)) \in \mathbb{R}^n$ is called the drift[2] and $\Upsilon(x(t)) \in \mathbb{R}^{n \times m}$ is a matrix with *vector fields* Υ_i, $i = 1 \ldots m$, as its columns. A vector field Υ, is a function that, with each point x in the state space $\mathcal{X}$, associates a vector belonging to that point's tangent space, $T_x\mathcal{X}$.

Definition 1 (Tangent Space and Tangent Bundle) *The* tangent space *at a point x, $T_x\mathcal{X}$, is defined as the space of all possible velocities of trajectories, passing through the point x. Throughout this report, the tangent space is identical with the space $\mathbb{R}^n$ of column vectors.*
Originating from this, we define the tangent bundle *as*

$$T\mathcal{X} = \bigcup_x T_x\mathcal{X}.$$

Definition 2 (Vector Field) *A vector field Υ, on a manifold $\mathcal{X}$, is defined as a map $\Upsilon : \mathcal{X} \ni x \mapsto \Upsilon(x) \in T_x\mathcal{X}$.*

Notice that $f(x,u) \in T_x\mathcal{X}$, is a special vector that for a fixed u, represents the possible infinitesimal change in the state variables with respect to time. It will be convenient to define the set $f(x,\mathcal{U})$ of all vectors $f(x,u)$, $u \in \mathcal{U}$. This set spans a subspace of $T_x\mathcal{X}$.

[2]The mobile platforms considered in chapters 5 and 6 are examples of *drift-less linear systems*, while Dubins' car (chapter 4) is an example of an *affine system with drift*.

Definition 3 (Distribution) *A* distribution *is an assignment $\Delta(x)$ of a subspace of $T_x\mathcal{X}$ for each $x \in \mathcal{X}$. Given the set of vector fields $\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_m$, the distribution $\Delta \subseteq T_x(\mathcal{X})$ is defined by*

$$\Delta = span\{\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_m\}.$$

The rank of a distribution at a point, equals the maximum number of linearly independent vectors among Υ_i, $i \in \{1, 2, \ldots, m\}$ at this point.

A distribution Δ, can thus be considered to assign a vector space to each point x, namely the space of all accessible velocities at the configuration represented by x. From definition 3, it follows that any vector field in Δ, can be expressed as a linear combination of the Υ_is, which, when independent, serve as a basis for Δ.

The subspaces $\Delta(x)$, typically[3] do not join or fit together in a coherent manner to form the tangent bundle of a smooth submanifold of $\mathcal{X}$. Figuratively spoken, if $\Delta(x)$ is represented by a plate or a clinker (this metaphor will turn out to be a quite striking one, cf. page 22), these clinkers will sometimes be joined to form a neat and smooth wall. Other times, they will appear as an unsorted and thorny pile of bricks. The distribution is then not the tangent bundle of a state submanifold. In such cases, if the task is to transfer the system between two prescribed configurations X_i and X_f, it is possible to reach final points X_f in a set of higher dimension than $\dim \Delta(X_i)$. To study this more formally, one of the most fundamental operations than can be performed on vector fields, namely the *Lie bracket*, has to be introduced.

Definition 4 (Lie Bracket) *Given two vector fields X and Y, another vector field, called the* Lie bracket *and denoted by $[X, Y]$, can be defined. The Lie bracket is computed by*

$$[X, Y] = D\,Y \cdot X - D\,X \cdot Y \tag{2.3}$$

where $D\,X$ and $D\,Y$ are the Jacobian matrices of X and Y respectively, i.e.

$$D\,X = \begin{pmatrix} \frac{\partial X_1}{\partial x_1} & \frac{\partial X_1}{\partial x_2} & \cdots & \frac{\partial X_1}{\partial x_n} \\ \frac{\partial X_2}{\partial x_1} & \frac{\partial X_2}{\partial x_2} & \cdots & \frac{\partial X_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial X_n}{\partial x_1} & \frac{\partial X_n}{\partial x_2} & \cdots & \frac{\partial X_n}{\partial x_n} \end{pmatrix} \quad \textit{and} \quad D\,Y = \begin{pmatrix} \frac{\partial Y_1}{\partial x_1} & \frac{\partial Y_1}{\partial x_2} & \cdots & \frac{\partial Y_1}{\partial x_n} \\ \frac{\partial Y_2}{\partial x_1} & \frac{\partial Y_2}{\partial x_2} & \cdots & \frac{\partial Y_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial Y_n}{\partial x_1} & \frac{\partial Y_n}{\partial x_2} & \cdots & \frac{\partial Y_n}{\partial x_n} \end{pmatrix}$$

Lie brackets have the following two basic properties

1. $[X, Y] = -[Y, X]$ *(skew-symmetric)*
2. $[[X, Y], Z] + [[Y, Z], X] + [[Z, X], Y] = 0$ *(the Jacobi identity)*

Remark 1 It is sometimes desirable to evaluate the Lie bracket component-wise. In such cases, the i^{th} component is conveniently evaluated as

$$[X, Y]_i = \sum_{p=1}^{n} \left(X_p \frac{\partial Y_i}{\partial x_p} - Y_p \frac{\partial X_i}{\partial x_p} \right)$$

Example 1 (Lie Bracket) As a basic example, which however is of interest in section 2.3.1, we calculate the Lie bracket between

$$\Upsilon_1 = \begin{pmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{pmatrix} \quad \text{and} \quad \Upsilon_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

[3] In the presence of nonholonomic constraints to be more specific (cf. section 2.2)

Using definition 4 we get

$$[\Upsilon_1, \Upsilon_2] = D\Upsilon_2 \cdot \Upsilon_1 - D\Upsilon_1 \cdot \Upsilon_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & -\sin\varphi \\ 0 & 0 & \cos\varphi \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \sin\varphi \\ -\cos\varphi \\ 0 \end{pmatrix}$$

Taking iteratively Lie brackets of all the vector fields of a system, motion vectors are generated which lie outside of Δ. A legitimate question in such cases, is to ask how to determine all the final points, that are reachable from a given initial point. This involves determining the systems *reachability* and *controllability* properties, which will be discussed in more detail in section 2.2. We shall now give a more precise definition of these notions. But before then, let us introduce a most fundamental concept in controllability theory, *viz.* the so-called *Control Lie Algebra*.
To make the idea of a Control Lie Algebra concrete, consider the set of all vector fields, that can be generated by taking Lie brackets $[\Upsilon_i, \Upsilon_j]$ of a systems all vector fields Υ_i and Υ_j, where $i, j \in \{1, 2, \ldots, m\} (i \neq j)$. Next, consider taking Lie brackets of the newly created vector fields with each other as well as with the original vector fields, Υ_is. This process is to be repeated indefinitely, by iteratively applying the Lie bracket operation to the new vector fields, until a vector space is obtained that is closed under the Lie bracket operation. A set is said to be closed under the Lie bracket operation if, whenever X and Y belongs the set, it follows that $[X, Y]$ also belongs to it. We then define the Control Lie Algebra to be the set of all vector fields, obtained by this iterative process. Stated more concisely, the Control Lie Algebra is the distribution generated by $\Upsilon_1, \ldots, \Upsilon_m$, and all their Lie brackets recursively computed.

Definition 5 (Control Lie Algebra) *The Control Lie Algebra associated with a distribution Δ, is denoted $CLA(\Delta)$ and is the smallest distribution containing Δ that is closed under the Lie bracket operation.*

Finding a basis of the CLA, is generally a tedious process. There exist however several systematic approaches for generating such a basis, one of which is called the Phillip-Hall basis (see [21]).

We may now proceed to define the concepts of *reachability*, *accessibility* and *controllability*. In each case, three different types will be considered, *viz. exact-time, small-time* and *finite-time*. In addition, each of these concepts will appear in a more restricted version (termed *local*), where we, not only pay attention to what points are reachable, accessible or controllable, but also restrict the corresponding state trajectory $x(\cdot)$ to stay in a *prescribed neighborhood* of the point.

Definition 6 (Reachability) *The set of* exact-time reachable *points from x, is defined as*

$$\mathcal{R}^*(x, T) = \{\hat{x} \in \mathcal{X} | \text{ there exists an admissible } u(\cdot) \text{ such that } x(0) = x \text{ and } x(T) = \hat{x}\}$$

Originating from this definition, we have:

$$\begin{array}{lll} \textit{set of} \text{ small-time reachable } \textit{points from } x & \mathcal{R}(x, T) & = \bigcup_{t \leq T} \mathcal{R}^*(x, t) \\ \textit{set of} \text{ finite-time reachable } \textit{points from } x & \mathcal{R}(x) & = \bigcup_{t \in \mathbb{R}^+} \mathcal{R}^*(x, t) \end{array}$$

Definition 7 (Local Reachability) *The set of* locally exact-time reachable *points from x, is defined as*

$$\begin{aligned} \mathcal{R}^*(x, T, N) = \{\hat{x} \in N | & \text{ there exists an admissible } u(\cdot) \text{ such that } x(0) = x, x(T) = \hat{x} \\ & \text{and the corresponding trajectory } x(\cdot) \in N\}, \end{aligned}$$

where N is a prescribed *neighborhood of x.*

Originating from this definition, we have:

set of locally small-time reachable *points from* x $\quad \mathcal{R}(x,T,N) = \bigcup_{t\leq T} \mathcal{R}^*(x,t,N)$

set of locally finite-time reachable *points from* x $\quad \mathcal{R}(x,N) = \bigcup_{t\in\mathbb{R}^+} \mathcal{R}^*(x,t,N)$

An alternative presentation of the definition of reachability can be found in table 2.1

	Exact-time	Small-time	Finite-time
Reachability	$\mathcal{R}^*(x,T) = \{\hat{x} \in \mathcal{X} \mid$ there exists an admissible $u(\cdot)$ such that $x(0) = x$ and $x(T) = \hat{x}\}$.	$\mathcal{R}(x,T) = \bigcup_{t\leq T} \mathcal{R}^*(x,t)$	$\mathcal{R}(x) = \bigcup_{t\in\mathbb{R}^+} \mathcal{R}^*(x,t)$
Local Reachability	$\mathcal{R}^*(x,T,N) = \{\hat{x} \in N \mid$ there exists an admissible $u(\cdot)$ such that $x(0) = x, x(T) = \hat{x}$ and the corresponding state trajectory $x(\cdot) \in N\}$.	$\mathcal{R}(x,T,N) = \bigcup_{t\leq T} \mathcal{R}^*(x,t,N)$	$\mathcal{R}(x,N) = \bigcup_{t\in\mathbb{R}^+} \mathcal{R}^*(x,t,N)$

Table 2.1: Definition of reachability.

We shall proceed by defining accessibility and controllability. Because of the similarities in the formulation of these definitions, they will be presented by means of table 2.2. As an illustrative example of how the table should be interpreted, we consider the definition of *locally exact-time controllable* systems. Then, from the first column of the last row of table 2.2, it follows that a system is called locally exact-time controllable, if for all $x \in \mathcal{X}$, $\mathcal{R}^*(x,T,N)$ contains a full neighborhood of x, for all $T > 0$ and any prescribed neighborhood N.

	Exact-time	Small-time	Finite-time
Accessible	for all $x \in \mathcal{X}$, $\mathcal{R}^*(x,T)$ contains a non-empty open set for all $T > 0$.	for all $x \in \mathcal{X}$, $\mathcal{R}(x,T)$ contains a non-empty open set for all $T > 0$.	for all $x \in \mathcal{X}$, $\mathcal{R}(x)$ contains a non-empty open set.
Locally Accessible	for all $x \in \mathcal{X}$, $\mathcal{R}^*(x,T,N)$ contains a non-empty open set for all $T > 0$ and any prescribed neighborhood N.	for all $x \in \mathcal{X}$, $\mathcal{R}(x,T,N)$ contains a non-empty open set for all $T > 0$ and any prescribed neighborhood N.	for all $x \in \mathcal{X}$, $\mathcal{R}(x,N)$ contains a non-empty open set for any prescribed neighborhood N.
Controllable	for all $x \in \mathcal{X}$, $\mathcal{R}^*(x,T)$ contains a full neighborhood of x for all $T > 0$.	for all $x \in \mathcal{X}$, $\mathcal{R}(x,T)$ contains a full neighborhood of x for all $T > 0$.	for all $x \in \mathcal{X}$, $\mathcal{R}(x)$ contains a full neighborhood of x.
Locally Controllable	for all $x \in \mathcal{X}$, $\mathcal{R}^*(x,T,N)$ contains a full neighborhood of x for all $T > 0$ and any prescribed neighborhood N.	for all $x \in \mathcal{X}$, $\mathcal{R}(x,T,N)$ contains a full neighborhood of x for all $T > 0$ and any prescribed neighborhood N.	for all $x \in \mathcal{X}$, $\mathcal{R}(x,N)$ contains a full neighborhood of x for any prescribed neighborhood N.

Table 2.2: Definition of accessibility and controllability.

Remark 2 What we call *exact-time* accessibility/ controllability, is sometimes referred to as *strong* accessibility/ controllability.

Next we shall present some results regarding determining a system's accessibility and controllability properties. Proposition 1 gives us a powerful tool for determining a systems accessibility properties. Notice that since locally exact-time accessibility is the most stringent type of accessibility considered, it implies the other five types of accessibilities, presented in table 2.2.

Proposition 1 (Locally Exact-time Accessibility) *(cf. [8] page 68, or [16] page 86)*
If an affine control system satisfies $CLA(\Delta) = T\mathcal{X}$*, then the system is* locally exact-time accessible.

The simplest approach to show small-time controllability for the special important case of *affine systems* is by studying the linearized system. It should be noted that small-time controllability naturally implies controllability and even (small-time) accessibility.

Proposition 2 (Small-time Controllability) *(cf. [16] page 75 or [21] page 511)*
If an affine system of form (2.2), is drift-free at $\hat{x}$ *(i.e.* $\Upsilon_0(\hat{x}) = 0$*) and the linearization at* $\hat{x}$ *and* $u = 0$

$$\dot{z} = \frac{\partial \Upsilon_0}{\partial x}(\hat{x})z + \Upsilon(\hat{x})u$$

is controllable (i.e. satisfies the Kalman rank condition), then the system is small-time controllable from $\hat{x}$*.*

However, no conclusion can be drawn about the controllability properties of the nonlinear system, if the linearized counterpart fails to be controllable. In such cases, one of the propositions presented in the very end of section 2.2 might be applicable.

2.2 Constrained Systems

In the sequel, we shall re-express the controllability properties of control affine systems in the dual language of constraints. We do not, however, introduce the formalism of differential forms and exterior calculus. Hence, we are to discuss two distinctly different types of constraints that might be imposed on a system, namely *holonomic* and *nonholonomic* constraints. The former are characterized by algebraic equality or inequality equations in terms of the state variables and are used to restrict the motion of the system to a smooth submanifold of the state space, $\mathcal{X}$. Obstacles and certain technical limitations, are suitably expressed by means of these *configuration-level* constraints. Nonholonomic constraints on the contrary, correlates the state variables and their time-derivatives, i.e. the generalized velocities, in a fundamentally non-integrable manner and are a *kinematic-level* type of limitations. These constraints, do not reduce the dimension of $\mathcal{X}$, but restrict the distribution of accessible velocities at a configuration to a submanifold of $T_x\mathcal{X}$, the tangent space at configuration x. Let us now discuss both of these types of constraints in greater detail.

Initially, suppose that a scalar constraint of the form

$$F(x, t) = 0 \tag{2.4}$$

applies to the motion of a system. Then, assuming that $F(x, t)$ is continuously differentiable, the Implicit Function Theorem theoretically ensures that equation (2.4) can be used to express one of the generalized coordinates in terms of the others and thereby be eliminated. Nevertheless, in practice, it may be difficult to solve implicit constraint equations for a certain state variable. This elimination however, restricts the set of admissible state variables to a smooth submanifold of $\mathcal{X}$. This submanifold is then the system's actual state space. Once one of the state variables is eliminated, the $(n-1)$ remaining state variables define the system's reduced state vector, x_{red}.

The system's reduced state space $\mathcal{X}_{\text{red}}$, is hence a $(n-1)$-dimensional submanifold of $\mathcal{X}$. This abolishes the system's redundancy properties. Any constraint, having the form of equation (2.4), is called a *holonomic equality constraint*. Generalizing this, if there are k *independent* configurational equality constraints imposed on the system, the reduced state space $\mathcal{X}_{\text{red}}$, will be a submanifold of $\mathcal{X}$ with dimension $(n-k)$ (see [1]).
A constraint of the form $F(x,t) \leq 0$, is termed a *holonomic inequality constraint*. These type of constraints, merely restrict a system's admissible state space to a limited submanifold of $\mathcal{X}$, without reducing its dimension. They are typically used for putting bounds on certain state variables or describing any possible obstacles present in $\mathcal{X}$.
Summarizing what has been said thus far, we state the following definition of a holonomic constraint (see [7])

Definition 8 (Holonomic Constraint) *A constraint that restricts the motion of a system to a smooth submanifold in the state space, is called a* holonomic *constraint.*

When the generalized velocity of a system satisfies an equality- or inequality condition that cannot be integrated or written as an equivalent condition on the configuration-level, the system is called a nonholonomic system. Hence, nonholonomic constraints can be expressed in terms of non-integrable relations between several state variable and their time-derivatives. A scalar constraint of form

$$G(x, \dot{x}, t) = 0 \tag{2.5}$$

where we assume that G is a smooth function, is said to be holonomic if it is integrable, i.e. equation (2.5) can be rewritten as a family of constraints in the form of equation (2.4). Otherwise the constraint is called a *nonholonomic equality constraint*. A scalar nonholonomic equality constraint, restricts the distribution of accessible velocities at a configuration $\Delta(x)$, to a $(n-1)$ - dimensional subspace of $T_x\mathcal{X}$, the tangent space at x, this, without affecting the dimension of the accessibility set. More generally, if there are l independent nonholonomic equality constraints imposed on the system, $\Delta(x)$ will be a subspace of $T_x\mathcal{X}$, with dimension $(n-l)$.
As a special important case (see section 2.3), we consider the case when G expresses a Pfaffian constraint, that is

$$G(x, \dot{x}, t) = g(x)\dot{x} = 0 \tag{2.6}$$

where $g(x) \in \mathbb{R}^{m \times n}$. If we are able to find the potential of $g(x)$, that is find a $f(x)$ such that $\nabla f(x) = g(x)$, then we might equivalently write equation (2.6) as $f(x) = \textit{constant}$, which is a constraint on the configuration-level. In such case, we have translated the kinematic-level constraint specified in equation (2.6), into a family of holonomic constraints. Notice that all configuration-level constraints of the form $f(x) = \textit{constant}$, can be written as an equivalent Pfaffian constraint $f'(x)\,\dot{x} = 0$, by a differentiation with respect to time. The converse is however not true, which is exactly the point that distinguishes, holonomic- and nonholonomic constraints. Stated otherwise, a Pfaffian constraint that is not equivalent to a family of constraints on the configuration-level, is called a nonholonomic constraint.
Finally, a non-integrable constraint of the form

$$G(x, \dot{x}, t) \leq 0 \tag{2.7}$$

is called a *nonholonomic inequality constraint*. It has the effect of restricting $\Delta(x)$, to a submanifold of $T_x\mathcal{X}$. Notice though that, usually, the rank of the distribution of accessible velocities, is unchanged, so that $\operatorname{rank}\Delta(x) = \dim T_x\mathcal{X} = n$.
We may now define the notion of nonholonomic constraints.

Definition 9 (Nonholonomic Constraint) *A* non-integrable *scalar constraint of the form*

$$G(x, \dot{x}, t) \underset{(\leq)}{=} 0 \tag{2.8}$$

where $G(x, \dot{x}, t)$ is a smooth function, is called a nonholonomic *equality (inequality) constraint.*

The reduction of the set of accessible velocities for nonholonomic systems, severely complicates the task of motion planning. In fact, motion planning in the presence of nonholonomic constraints, is fundamentally different from motion planning for holonomic systems. For the latter, a set of independent generalized coordinates can be found, and thus an arbitrary motion in the system's reduced state space, $\mathcal{X}_{\text{red}}$, is feasible. This is since x_{red}, serves as a basis that spans $\mathcal{X}_{\text{red}}$ and thus enables control of all the state variables. In contrast, for a nonholonomic system, a set of independent generalized coordinates attracts attention by its absence. By virtue of this, not every motion is feasible, but only those which satisfy the instantaneous nonholonomic constraint (2.8). This implies that the main consequence of a nonholonomic constraint, is that an arbitrary path in the admissible state space does not necessarily correspond to a feasible trajectory for the system. With this in mind, it is easy to see why this fact plays a leading role in our struggle to plan the motion of a nonholonomic robot.

The presence of kinematic constraints raises two main questions. First, how do we determine whether a given kinematic constraint is nonholonomic or not? The second issue is to determine whether the imposed nonholonomic constraints, in addition to restricting the distribution of accessible velocities, also reduce the set of configurations reachable from the initial configuration. At this point, we only know that they do not reduce the dimension of this set, since otherwise it would be integrable. So the second question at hand is; does the presence of nonholonomic constraints, affect the controllability properties of the system?
Concerning the characterization of a constraint's integrability properties, we consult [11] and establish the fact that for a set of kinematic constraints that are linear in $\dot{x}$, the Frobenius Integrability Theorem gives a necessary and sufficient condition for their integrability (see e.g. [11], page 415). As a corollary, in the case of a single Pfaffian constraint[4], the following is presented:

Proposition 3 (Determination of Integrability of Pfaffian Constraints) *A scalar linear kinematic constraint*

$$g(x)\dot{x} = \sum_{p=1}^{m} g_p(x)\dot{x}_p = 0$$

is holonomic iff the following relation holds for any $i, j, k \in [1, m]$ such that $1 \leq i < j < k \leq m$:

$$A_{ijk} = g_i\Big(\frac{\partial g_k}{\partial x_j} - \frac{\partial g_j}{\partial x_k}\Big) + g_j\Big(\frac{\partial g_i}{\partial x_k} - \frac{\partial g_k}{\partial x_i}\Big) + g_k\Big(\frac{\partial g_j}{\partial x_i} - \frac{\partial g_i}{\partial x_j}\Big) = 0$$

Remark 3 In the tree dimensional case (i.e. $m = 3$), this necessary and sufficient condition, reads $A_{123} = g \cdot [\nabla \times g] = 0$.

Proposition 4 is used to determine whether the $g(x)$ is proportional to the gradient of some function $f(x)$, i.e. whether there exist a function $f(x)$ such that $g(x) = h(x)\nabla f(x)$. This is to be compared with Poincaré's lemma, which instead lays information about the existence of a potential function to $g(x)$, i.e. the existence of a function $f(x)$ such that $g(x) = \nabla f(x)$.

[4]This case is of particular interest for us, since as will be shown in section 2.3, there is a linear scalar nonholonomic constraint imposed on the mobile platforms considered.

Proposition 4 (Poincaré's Lemma) *The coefficients, $g(x)$, of a scalar linear kinematic constraint*

$$g(x)\dot{x} = \sum_{p=1}^{m} g_p(x)\dot{x}_p = 0$$

can be written as a gradient $g(x) = \nabla f(x)$, of some function $f(x)$, iff the following relation holds for any $i, j \in [1, m]$:

$$\Big(\frac{\partial g_i}{\partial x_j} - \frac{\partial g_j}{\partial x_i}\Big) = 0$$

These results provide an effective characterization of holonomy (and hence nonholonomy) for a single Pfaffian constraint.

One way to understand and discuss the effect of nonholonomy on the controllability of a robot, is to regard the robot's velocity vector $\dot{x}$, as the control vector. As illuminated earlier, the distribution of accessible velocities at a configuration $\Delta(x)$, which represents the set of possible motions that can be executed from each configuration, is reduced by virtue of nonholonomy. Hence, for nonholonomic systems, the accessible control space does not coincide with the system's tangent space at x. As an example, we know that a car can not perform crab-like motions, i.e. move sideways[5]. A limitation that causes major problems for many drives trying to parallel-park their car. Nevertheless, from our everyday life experiences, we might attest that this implication does not influence the set of reachable destinations. Hence, there exist examples where the set of reachable configurations are unaffected, although the system is unable to reach all the neighboring configurations directly. Can this empirical observation be generalized?
The following propositions, give powerful results on the controllability properties of affine systems.

Proposition 5 (Small-time Controllability) *(cf. [10], page 106)*
An affine system with no restriction on the size of the control is small-time controllable *if the rank of the control Lie algebra $CLA(\Delta)$ equals n (the dimension of $\mathcal{X}$), for all $x \in \mathcal{X}$.*

Proposition 6 (Controllability) *(cf. [10], page 106)*
A drift-less affine system remains controllable *(but not necessarily small-time controllable) if*

a) $CLA(\Delta) = T\mathcal{X}$ *and*

b) *the convex hull of the control set $\mathcal{U}$ contains the origin in $\mathbb{R}^m$*

By virtue of the foregoing two propositions, checking the controllability properties of a system requires the analysis of the Control Lie Algebra associated with it. Checking the Lie Algebra Rank Condition (LARC), on a control system, is a very fundamental and useful tool for determining a systems controllability properties. In section 2.3.1, the controllability analysis of a number of kinematic models for mobile platforms, will be carried out by means of LARC.
When considering the Reeds-Shepp's car in chapter 5, we will need results on controllability properties of drift-less symmetric systems that fulfill LARC. This will be introduced next. If we define a system to be *symmetric* if every trajectory run backwards in time is also a trajectory for the system, it follows that a necessary condition for possessing symmetry properties, is that the control set $\mathcal{U}$ is symmetric, so that

[5]More specifically, we require the linear velocity to be orthogonal to the axis connecting the rear wheels, and as will be shown in section 2.3, this restriction translates into a linear scalar nonholonomic equality constraint.

both u and $-u$ are admissible inputs[6]. In addition, the system have to be drift-less. With this terminology, the following is valid:

Proposition 7 (Controllability of Affine Drift-less Symmetric Systems) *(cf. [16], page 82) If the vector fields of a drift-less symmetric control system of form*

$$\dot{x} = \Upsilon(x)u$$

have LARC at all $x \in \mathcal{X}$*, then it is* locally controllable.

2.3 Mobile Platforms

Several interesting sets of differential equations, specifying the system dynamics for objects that move by rolling wheels, can be defined. The kinematic models that we are going to consider are the so-called *unicycle*, *car-like* and *front-wheeled car* models.

Unicycle Robot Model

Referring to figure 2.1, the two control inputs for the unicycle robot model, are the linear- and the lateral velocities. The former, which is also referred to as the longitudinal velocity, is denoted by v, while the latter, which represents the robot's angular velocity, is denoted by ω. To fully characterize the state of the robot, we need a minimum of three state variables. Two of them are required to specify the position of the robot in $\mathbb{R}^2$ and one to specify its orientation. The point (x, y), is a reference point on the platform (throughout this report, the rear axle's midpoint), while φ is used to denote the robot's orientation.[7] The position vector $(x, y) \in \mathbb{R}^2$, while the orientation angle is a real number modulo 2π, or equivalently, a member of S^1, the unit circle in the plane. Consequently, the state space of the robot becomes $\mathcal{X} = \mathbb{R}^2 \times S^1$.

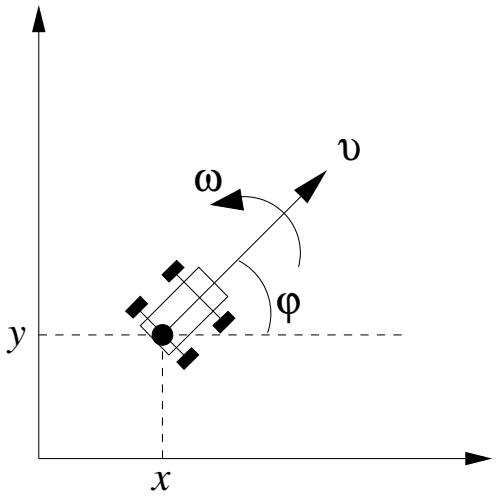


Figure 2.1: The unicycle robot model.

The task now is to represent the motion of the mobile platform as a set of differential equation such that

[6]Notice that this requirement is stronger than and therefore implies condition b) in proposition 6.

[7]Be careful of notation collision. So far, x has denoted a general state vector, but might henceforth occasionally refer to a robot's x-position in $\mathbb{R}^2$.

$$\begin{aligned} \dot{x} &= f_1(x, y, \varphi, \upsilon, \omega) \\ \dot{y} &= f_2(x, y, \varphi, \upsilon, \omega) \\ \dot{\varphi} &= f_3(x, y, \varphi, \upsilon, \omega). \end{aligned} \tag{2.9}$$

The platform's angular velocity specifies the time-evolution of the orientation angle φ, so that $\dot{\varphi} = \omega$. Imposing the "rolling without slipping assumption" on the robot wheels (see assumption 1, page 20 for more details), means that in an infinitely-small time interval, the point (x, y), must move in the direction of the orientation angle, φ. Simple vector algebra then, yields that $\dot{x} = \upsilon \cos\varphi$ and $\dot{y} = \upsilon \sin\varphi$. Summarizing this, we get

$$\begin{aligned} \dot{x} &= \upsilon \cos\varphi \\ \dot{y} &= \upsilon \sin\varphi \\ \dot{\varphi} &= \omega. \end{aligned} \tag{2.10}$$

In order to make the notations more consistent with what has been presented previously (cf. equation (2.2)), we set the vector of the state variables $x = (x, y, \varphi)$ and re-write equation (2.10) as

$$\dot{x} = \begin{pmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{pmatrix} \upsilon + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega = \Upsilon_1 \, \upsilon + \Upsilon_2 \, \omega = \Upsilon \cdot u, \tag{2.11}$$

where $u = [\upsilon \; \omega]^T$ and Υ has Υ_i as its i^{th} column.

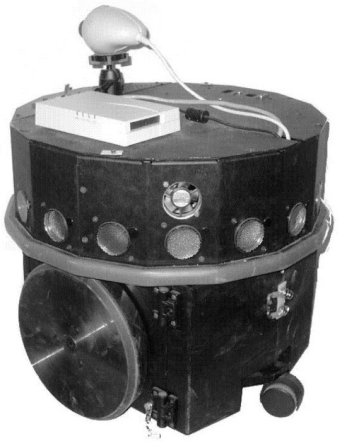


Figure 2.2: Most robots of unicycle-type have two actively controlled steering wheels and a castor wheel. The picture illustrates Nomadic Technologies Scout.

Among the kinematic models to be presented in this section, the unicycle robot model, is the most simplified, this is so because it gives direct control over the orientation of the vehicle. If $\upsilon = 0$ is an admissible control, then a unicycle robot is free to rotate unrestricted, while standing still in (x, y). Consequently a unicycle robot is able to follow any continuous path. However, in the case of the unicycle car models considered in chapter 4 and 5 (Dubins'- and Reed-Shepp's car models), the control domain in the linear direction does not include the origin, so that $\upsilon = 0$ is not a member of the admissible control set, $\mathcal{U}$. This excludes the possibility to rotate the car while standing still. As a matter of fact, solutions where you rotate around freely

in one spot, turn out to be non-optimal and does not occur even in the case when such motions are admissible to be executed. This is concluded from studying time-optimal paths for the convexified Reeds-Shepp's car (see chapter 5 and 6), for which $v = 0$ indeed belongs to $\mathcal{U}$.

Car-like Robot Model

From a driver's point of view, a vehicle has two control possibilities: the accelerator/brake pedal and the steering wheel. The accelerating factor is the linear velocity v, while the steering wheel specifies the angle between the front wheels and the main direction of the vehicle. We define this as the *steering angle* δ, which is our second control variable (see figure 2.3). In practise, the two front wheels are seldom exactly parallel, why we might set δ to be the average of these two angles. Augmenting the steering angle, obviously captures the characteristics of a car even better and results in a refined model compared with the unicycle model.

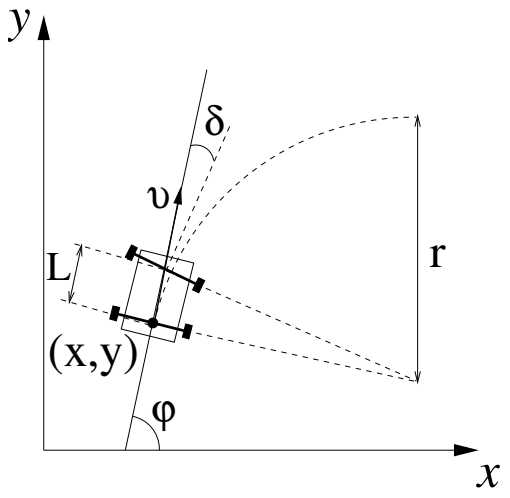


Figure 2.3: The car-like robot model.

The car-like robot model, has the same state variables as the unicycle robot, i.e. $x = (x, y, \varphi)$. Again, the task is to find a set of differential equations f_1, f_2 and f_3 (cf. equation (2.9)), that properly describes the kinematic properties of the car.
Since the "rolling without slipping" assumption is valid even in this case, the motion in the $\mathbb{R}^2$-plane is dictated by the same equations as before, so that $\dot{x} = v\cos\varphi$ and $\dot{y} = v\sin\varphi$.
Proceeding to the time evolution of the orientation angle $\dot{\varphi}$, let s denote the distance traveled by the vehicle. Then $\dot{s} = v$, which is the speed of the car. As shown in figure 2.3, r represents the radius of a circle that will be traversed by (x, y), when the steering angle is fixed. Consequently $\mathrm{d}s = r\,\mathrm{d}\varphi$. From simple trigonometry, we have $\tan\delta = \frac{L}{r}$, which implies

$$\mathrm{d}\varphi = \frac{\tan\delta}{L}\,\mathrm{d}s.$$

Dividing by $\mathrm{d}t$ and using the fact that $\dot{s} = v$, yields

$$\dot{\varphi} = \frac{v}{L}\tan\delta.$$

Thus, the system dynamics for the car-like robot is

$$\begin{aligned} \dot{x} &= v\cos\varphi \\ \dot{y} &= v\sin\varphi \\ \dot{\varphi} &= \frac{v}{L}\tan\delta. \end{aligned} \tag{2.12}$$

Since the linear velocity appears in the time evolution of all the state variables, this car model does not have the possibility to make arbitrary rotations while standing still in $\mathbb{R}^2$. It is only able to follow paths that are at least continuously differentiable.

Front-wheeled Car Model

In the car-like robot model (equation (2.12)), we had the steering angle as an input and assumed to have direct control over its value. This corresponds to being able to move the front wheels instantaneously, which obviously contradicts our intuition about cars in general, and in many real-life applications this assumption is an unrealistic one. Whenever the value of δ changes discontinuously, the path traced out in the plane by (x, y), will have a discontinuity in its curvature. To put this right and make a car model that only generates smooth paths (i.e. belongs to C^2), we might add the steering angle as an extra state variable and consider its derivative to be one of the input signals. Introducing this integrator-chain, results in a delay in the lateral control, such that δ is only allowed to change its values in a continuous manner.
We thus have to consider a four-dimensional state space, in which each state is represented as $x = (x, y, \delta, \varphi)$. The system dynamics for the front-wheeled car model then becomes

$$\begin{aligned} \dot{x} &= v\cos\varphi \\ \dot{y} &= v\sin\varphi \\ \dot{\delta} &= \omega \\ \dot{\varphi} &= \frac{v}{L}\tan\delta \end{aligned} \tag{2.13}$$

where, as usual, v represents the linear velocity, while ω now is the angular velocity of the steering angle. By setting

$$\Upsilon_1 = \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \\ 0 \\ \frac{\tan(\delta)}{L} \end{pmatrix} \quad \text{and} \quad \Upsilon_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$

we may re-write equation (2.13) as

$$\dot{x} = \Upsilon_1\, v + \Upsilon_2\, \omega = \Upsilon \cdot u \tag{2.14}$$

where, as before, $u = [v\ \omega]^T$ and Υ has Υ_i as its i^{th} column.

Although it holds true that the method of introducing the former input signal δ, as a state variable and rather consider its time derivative to be the new control input, gives a more realistic model for describing a real car, one should ask weather such approach is justified, as long as we do not take similar action for the linear control, v. Is it reasonable to consider that we are able to dictate the instant value of the linear velocity of a car? Because of its weight, the inertia along the platform's main axis, is generally orders of magnitude larger than the resistance existing in the steering device, why it may be motivated to introduce an integrator-chain in the linear direction as well. But on the other hand, allowing the steering angle to change instantaneously, corresponds to matter moving in a discontinuous manner in

$\mathcal{X}$, which is physically offensive. This is however not the case when v is allowed to change its value discontinuously.

For a summary of what has been said about the kinematic modeling of mobile platforms, consider table 2.3

Model name	Unicycle Robot Model	Car-like Model	Front-wheeled Car
System Dynamics	$\dot{x} = v\cos\varphi$ $\dot{y} = v\sin\varphi$ $\dot{\varphi} = \omega$	$\dot{x} = v\cos\varphi$ $\dot{y} = v\sin\varphi$ $\dot{\varphi} = \frac{v}{L}\tan\delta$	$\dot{x} = v\cos\varphi$ $\dot{y} = v\sin\varphi$ $\dot{\delta} = \omega$ $\dot{\varphi} = \frac{v}{L}\tan\delta$
Path Smoothness	C	C^1	C^2

Table 2.3: The system dynamics and the required smoothness of the generated paths, for the unicycle, car-like and the front-wheeled car models.

Figure 2.4: The autonomous ground vehicle at FOI.

Generally, we make the following two assumptions about the platform's kinematic abilities:

Assumption 1 (Rolling Without Slipping) *It is customary to assume that the robot wheels do not slip. This assumption is legitimate in all moderate-speed scenarios.*

Assumption 2 (Maximum Steering Angle) *We further assume that we have a limitation on the maximum steering angle* $|\delta| \leq \delta_{\max}$*, meaning that the turning radius for our car is lower bounded.*

Let us examine the effects of these two assumptions. Assuming pure rolling contact between the wheels and the ground, i.e. no slipping, the velocity of (x, y) is always orthogonal to the axis connecting the rear wheels, or equivalently, parallel to the main axis of the vehicle, which naturally coincide with the direction of v. Hence we have

$$\dot{x} = v\cos\varphi \quad \text{and} \quad \dot{y} = v\sin\varphi.$$

Eliminating υ, we get

$$\dot{x}\sin\varphi - \dot{y}\cos\varphi = 0. \tag{2.15}$$

In order to be able to characterize the integrability properties of equation (2.15), we re-write it in the following form

$$(\sin\varphi, -\cos\varphi, 0)\cdot\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\varphi}\end{pmatrix} = g(x)\,\dot{x} = 0. \tag{2.16}$$

Then, we deduce from remark 3 that equation (2.15) would be holonomic (and hence integrable) if and only if $A = g\cdot[\nabla\times g] = 0$. Now since in our case

$$A = g\cdot[\nabla\times g] = (\sin\varphi, -\cos\varphi, 0)\cdot(-\sin\varphi, \cos\varphi, 0) = -1 \neq 0$$

regardless of the orientation angle φ, we conclude that the no slipping assumption, imposes a scalar linear nonholonomic equality constraint, restricting the space of accessible velocities to a 2-dimensional subspace of $T_x\mathcal{X}$ (see figure 2.5).

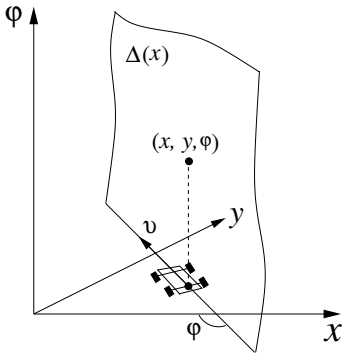


Figure 2.5: In the presence of the "rolling without slipping" assumption, the velocity vector $(\dot{x}, \dot{y})$, must lie in the direction of υ and is therefore restricted to the 2-dim hyper surface $\Delta(x)$.

Proceeding to the second assumption, our objective is to re-write the constraint $|\delta| \leq \delta_{\max}$ as a kinematic-level type of constraint. From figure 2.3 we understand that $\tan\delta_{max} = \frac{L}{R}$, where R denotes the radius of the circle traced out by (x, y) when the steering angle $\delta = \delta_{max}$. Also, we have (cf. table 2.3) $\dot{\varphi} = \frac{\upsilon}{L}\tan\delta$. Then it follows that

$$|\arctan(\frac{\dot{\varphi}L}{\upsilon})| = |\delta| \leq \delta_{\max} = \arctan(\frac{L}{R})$$

which implies

$$|\dot{\varphi}| \leq \frac{|\upsilon|}{R} \tag{2.17}$$

or, equivalently, by expressing $|\upsilon| = \sqrt{\dot{x}^2 + \dot{y}^2}$

$$R^2\dot{\varphi}^2 - \dot{x}^2 - \dot{y}^2 \leq 0.$$

Due to the fact that in all real-life applications, there must exist an upper bound on the value of the linear velocity, we might re-write equation (2.17) as

$$|\dot{\varphi}| \leq \frac{1}{R} \tag{2.18}$$

where we have re-scaled the maximum admissible velocity to one. To get an expression for $\varphi(t)$, we calculate

$$|\varphi(t) - \varphi(0)| = \left|\int \dot{\varphi}(t)\,\mathrm{d}t\right| \leq \int |\dot{\varphi}(t)|\,\mathrm{d}t \leq \frac{t}{R}, \tag{2.19}$$

where the last inequality follows from equation (2.18). Equation (2.19) has been sketched in figure 2.6, from where we might conclude that φ, must lie within a isosceles triangle with top-angle 2α.

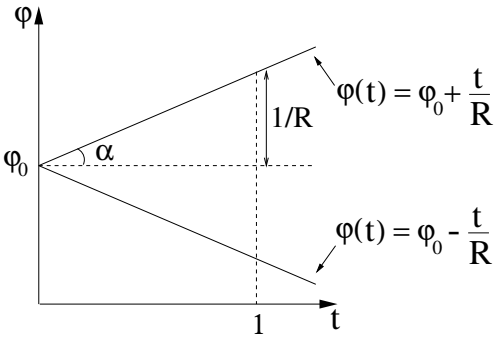


Figure 2.6: The maximum steering angle assumption, entraps $\varphi(t)$ in a triangle with $\alpha = \arctan\frac{1}{R}$.

Hence, assumption 2 restricts the set of accessible velocity vectors to a two-sided cone with angle 2α, where $\alpha = \arctan\frac{1}{R}$. However, it does not reduce the dimension of $\Delta(x)$, but merely restricts it to a subset of $T_x\mathcal{X}$.
Finally, the augmented effect of both our assumptions, is shown in figure 2.7. We observe that these two assumptions, restrict the set of accessible velocity vectors to the intersection between the 2-dimensional plane dictated by the "rolling without slipping" assumption (illustrated in figure 2.5) and a two-sided cone that origins from the "maximum steering angle" assumption.

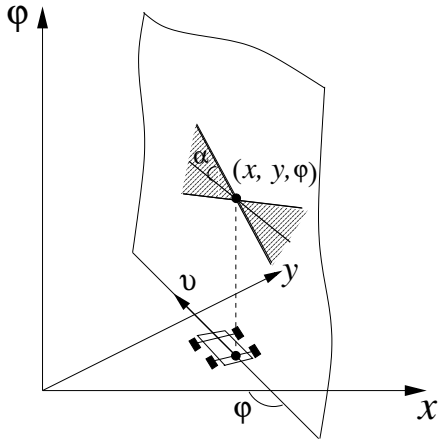


Figure 2.7: The shadowed area represents the restricted set of accessible velocity vectors, for a non-slipping vehicle with bounded steering angle.

2.3.1 Controllability Analysis We will now, by utilizing the results presented in section 2.2 (proposition 5 to be more precise), show that all the models presented,

are small-time controllable. This determination will be carried out by checking the Lie Algebra Rank Condition (LARC) for each of the three models. Notice though that all of these results only hold true when there are no bounds on the input signals, so that $\mathcal{U} = \mathbb{R}^m$.

Unicycle Robot Model

Let us consider the two admissible controls $u_1 = [1\ 0]^T$ and $u_2 = [0\ 1]^T$. Using the notations used in equation (2.11), the associated vector fields are Υ_1 and Υ_2. The Lie bracket them between, is (see definition 4)

$$[\Upsilon_1, \Upsilon_2] = \begin{pmatrix} \sin\varphi \\ -\cos\varphi \\ 0 \end{pmatrix}$$

We can immediately observe that $[\Upsilon_1, \Upsilon_2]$ is linearly independent from Υ_1 and Υ_2, by noting that the determinant of the matrix

$$\begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ 0 & 0 & 1 \\ \sin\varphi & -\cos\varphi & 0 \end{pmatrix}$$

is non-zero for all (x, y, φ). This implies that the dimension of $CLA(\Delta) = 3$. Hence, the unicycle robot model fulfills LARC and is by virtue of proposition 5, small-time controllable.

Car-like Robot Model

Firstly, we wish to re-write the system dynamics for the car-like robot model (equation (2.12)), as an affine control system. Then, by setting $\hat{\omega} = \frac{v}{L}\tan\delta$, we get

$$\begin{aligned} \dot{x} &= v\cos\varphi \\ \dot{y} &= v\sin\varphi \\ \dot{\varphi} &= \hat{\omega}, \end{aligned}$$

which is to be recognized as the system dynamics for the unicycle robot model. However, by doing this we have introduced a subtle difference which involves that $\hat{\omega}$ is no longer a free variable but has to equal zero whenever $v = 0$, i.e. the $\hat{\omega}$-axes in the $\hat{\omega} - v$ plane, is non-admissible. This holds true whenever there are no restrictions imposed on δ. But considering that we only require the control input to be a piecewise continuous function, we realize that it does not matter if the control is not allowed on a null set. Consequently, from what has been said in the previous paragraph about the controllability properties of the unicycle robot model, we conclude that the car-like robot model is small-time controllable as well.

Front-wheeled Car Model

We recall from equation (2.14), that the system dynamic for this model was

$$\dot{x} = \Upsilon_1(x)\, v + \Upsilon_2(x)\, \omega,$$

where

$$\Upsilon_1 = \begin{pmatrix} \cos\varphi \\ \sin\varphi \\ 0 \\ \frac{\tan\delta}{L} \end{pmatrix} \quad \text{and} \quad \Upsilon_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

We then have

$$[\Upsilon_1, \Upsilon_2] = D\Upsilon_2 \cdot \Upsilon_1 - D\Upsilon_1 \cdot \Upsilon_2 = -\begin{pmatrix} 0 & 0 & 0 & -\sin\varphi \\ 0 & 0 & 0 & \cos\varphi \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1+\tan^2\delta}{L} & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = -\begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1+\tan^2\delta}{L} \end{pmatrix}$$

and

$$\begin{aligned}[\Upsilon_1, [\Upsilon_1, \Upsilon_2]] &= -\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2(\tan\delta+\tan^3\delta)}{L} & 0 \end{pmatrix}\begin{pmatrix} \cos\varphi \\ \sin\varphi \\ 0 \\ \frac{\tan\delta}{L} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & -\sin\varphi \\ 0 & 0 & 0 & \cos\varphi \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1+\tan^2\delta}{L} & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1+\tan^2\delta}{L} \end{pmatrix} \\ &= \begin{pmatrix} -\sin\varphi(\frac{1+\tan^2\delta}{L}) \\ \cos\varphi(\frac{1+\tan^2\delta}{L}) \\ 0 \\ 0 \end{pmatrix}.\end{aligned}$$

In order to check the linear dependency of the four vectors $\Upsilon_1, \Upsilon_2, [\Upsilon_1, \Upsilon_2]$ and $[\Upsilon_1, [\Upsilon_1, \Upsilon_2]]$, we calculate

$$\begin{vmatrix} \cos\varphi & \sin\varphi & 0 & \frac{\tan\delta}{L} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{1+\tan^2\delta}{L} \\ -\sin\varphi(\frac{1+\tan^2\delta}{L}) & \cos\varphi(\frac{1+\tan^2\delta}{L}) & 0 & 0 \end{vmatrix} = -\Big(\frac{1+\tan^2\delta}{L}\Big)^2,$$

which is $\neq 0$, for all configurations $(x, y, \delta, \varphi) \in \mathcal{X}$. Thus, the system (2.14), fulfills LARC in every point in the state space, and is thereby small-time controllable.

3. Optimal Control

We start this chapter by formulating an *optimal control problem* that is general enough to correspond to the problems that will be considered in chapters 4 through 6. The main purpose of this chapter is then to present a method for characterizing the optimal control associated with such a problem. This is the scope of section 3.1. The solution method presented here, was first presented in 1957 by the Soviet mathematician L.S. Pontryagin [18] and is called the Pontryagin Maximum Principle (PMP).
In section 3.2, we take a look at more general optimal control problems and the implications of these generalizations on the results presented in section 3.1.
It must be emphasized that there exist other approaches to solve optimal control problems, where Dynamic Programming plays a leading role. For an extensive survey of this method, consult [9].

Control problems typically concern finding a (not necessarily unique) control law $u(\cdot)$, which transfers the system in finite time from a given initial state X_i, to a given final state X_f. This transition is to occur along an admissible path, i.e. $x(\cdot) \in \mathcal{X}$ and respects all kinematic constraints imposed on it. Here $\mathcal{X}$ denotes, in accordance with what has been introduced in sections 2.1 and 2.2, the state space of the robot. We further assume that $u(\cdot)$ is admissible, i.e. is piecewise continuous and belongs to $\mathcal{U}$, the admissible control space.
Let there now be a rule which assigns a unique, real-valued number to each of these transfers. Such a rule can be viewed as the transition cost between X_i and X_f along an admissible path, completely specified by $u(\cdot)$. *Optimal control* then, concerns specifying this rule and thereby providing a systematic method for selecting the "best", or "optimal" control law, according to some prescribed cost functional.
An optimal control problem may have the following form

$$\text{minimize} \int_0^{t_f} \mathcal{L}(x(t), u(t))\, \mathrm{d}t \tag{3.1}$$

subject to

$\dot{x}(t) = f(x(t), u(t))$	system dynamics	(3.2)
$x(0) = X_i$	initial condition	
$x(t_f) = X_f$	final condition	
$x(\cdot) \in \mathcal{X} \subseteq \mathbb{R}^n$	state constraint	
$u(\cdot) \in \mathcal{U} \subseteq \mathbb{R}^m$	control constraint	
$t_f \in (0, \infty)$	final time (to be optimized)	
where		
$\mathcal{L} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$	integral cost function	
$f : \mathcal{X} \times \mathcal{U} \to T\mathcal{X}$	system vector field	

where the integrand $\mathcal{L}$, assigns a cost to the state- and control trajectories. We assume (cf. section 2.1) that the system dynamics $f \in C^\infty$. Notice that since the

stated optimal control problem is invariant under time-translation (since neither f nor $\mathcal{L}$ depends explicitly on the time parameter, t), the value of initial time may be selected arbitrary, hence there is no loss of generality to choose $t_i = 0$.

3.1 The Pontryagin Maximum Principle

In this section we shall state a slimmed-down version of the Pontryagin Maximum Principle (PMP) that covers the optimal control problem (3.1). More general problems, for instance those involving terminal cost terms or non-autonomous systems, can be re-stated and solved by means of this more basic result as well (see e.g. [9], [13], [17] or [18]). One should however be aware of how these generalizations modify the proposition to be presented and take that into consideration. These topics will be covered in this chapters last section, 3.2.

Given the problem statement (3.1), the Pontryagin approach is to introduce a vector of *auxiliary variables* λ and construct a *Hamiltonian function*

$$H(x, \lambda) = \max_u \mathscr{H}(x, \lambda, u), \quad \text{where } \mathscr{H}(x, \lambda, u) = \lambda^T f - \mathcal{L}. \tag{3.3}$$

The following proposition then, gives us *necessary* condition for optimality of a control law.

Proposition 8 (PMP) *In order for u^* to be an optimal solution of (3.1), the following are necessary conditions:*

1. *There exists a non-zero adjoint variable λ, such that $\dot{\lambda} = -\frac{\partial H}{\partial x}$*
2. $u^* = \arg\max \mathscr{H}(x, \lambda, u), \qquad \forall t \in [0, t_f]$
3. $H(x, \lambda) = 0, \qquad \forall t \in [0, t_f]$

Remark 4 In the reference literature, the Hamiltonian (3.3), is sometimes defined as $\mathscr{H}(x, \lambda, u) = \lambda^T f + \lambda_0 \mathcal{L}$. In that case, PMP supplies us with the additional constraint $\lambda_0 = constant \leq 0$. Then we have:

Case 1: $(\lambda_0 = 0)$ This pathological case arises usually by virtue of lack of controllability or other related problems and will therefore not be considered in this report. However, when presenting the sufficient family of paths for Dubins', as well as Reeds-Shepp's car models (in chapter 4 and 5 respectively), the time optimal paths associated with the pathological case are implicitly considered in the sense that they turn out to be included in the solutions that correspond to the the $\lambda_0 < 0$ case.

Case 2: $(\lambda_0 < 0)$ Set $\tilde{\lambda} = [\lambda_0 \ \lambda]$ and $\tilde{f} = [\mathcal{L} \ f]$, allowing us to write the Hamiltonian as $H = \tilde{\lambda}\tilde{f}$. The adjoint equation then becomes $\dot{\tilde{\lambda}} = -\frac{\partial H}{\partial \tilde{x}}$. Now, since the Hamiltonian H is linear in $\tilde{\lambda}$, the adjoint equation is linear in $\tilde{\lambda}$ as well, and can therefore be arbitrary re-scaled, meaning that we can put $\lambda_0 = -1$ without any loss of generality.

Remark 5 The linearity of the adjoint equation is also the reason for requiring λ to be a *non-zero* vector, since an adjoint vector λ such that $\lambda(t) = 0$ for some t, in fact satisfies $\lambda(t) = 0$ for all t.

Remark 6 If we fix the final time t_f, i.e. prescribe the arrival time, the second condition becomes: $H = constant$. Knowledge about the value of the Hamiltonian is replaced by knowledge of the transfer time.

To present a roughly sketched motivation for the appearance of the necessary conditions specified by proposition 8, let us focus on the integral cost function and the system dynamics in problem (3.1). If, for the sake of clarity, the arguments are disregarded, (3.1) can be written as

$$\text{minimize} \int_0^{t_f} [\mathcal{L} + \lambda^T(\dot{x} - f)]\,\mathrm{d}t \tag{3.4}$$

subject to the constraints specified in (3.2). Notice in particular the constraint on the generalized velocity $\dot{x} = f$, which makes the newly introduced parentheses equal to zero. We have thus added a term to the value function, which the constraint forces to be equal to zero, hence this paraphrase is valid for all λs. Using the definition of the Hamiltonian function (3.3), we shall proceed by expressing the value function J as

$$J = \int_0^{t_f} [\lambda^T \dot{x} - \mathscr{H}]\,\mathrm{d}t.$$

We proceed by considering a small variation in the control Δu and the free terminal time Δt, causing a variation Δx in x, together with a variation $\Delta\lambda$ in the arbitrary λ. The first order contribution to the change in J, will then be

$$\Delta J = \int_0^{t_f} \Big(\Delta\lambda\dot{x} + \lambda^T\Delta\dot{x} - \frac{\partial\mathscr{H}}{\partial x}\Delta x - \frac{\partial\mathscr{H}}{\partial\lambda}\Delta\lambda - \frac{\partial\mathscr{H}}{\partial u}\Delta u\Big)\,\mathrm{d}t + \Delta t\Big[\lambda^T\dot{x} - \mathscr{H}\Big]_{t=t_f} \tag{3.5}$$

Paying extra attention to the second term of the integrand and applying integration by parts to it, yields

$$\int_0^{t_f} \lambda^T\Delta\dot{x}\,\mathrm{d}t = \Big[\lambda^T\Delta x\Big]_{t=t_f} - \int_0^{t_f} \dot{\lambda}\Delta x\,\mathrm{d}t, \tag{3.6}$$

the equality following from the fact that the initial time is fixed and at this point the initial state is given, i.e. $[\Delta x]_{t=0} = 0$. Combining (3.5) and (3.6) gives

$$\begin{aligned}\Delta J \;=\; & \int_0^{t_f} \Big((\dot{x} - \frac{\partial\mathscr{H}}{\partial\lambda})\Delta\lambda - (\dot{\lambda} + \frac{\partial\mathscr{H}}{\partial x})\Delta x - \frac{\partial\mathscr{H}}{\partial u}\Delta u\Big)\,\mathrm{d}t \\ & + \lambda^T\Big[\dot{x}\Delta t + \Delta x\Big]_{t=t_f} - \Delta t\Big[\mathscr{H}\Big]_{t=t_f}\end{aligned} \tag{3.7}$$

At an optimal point, we must have $\Delta J = 0$. Let's see what necessary conditions must be fulfilled at such a point. From the definition of the Hamiltonian, it follows that $\frac{\partial\mathscr{H}}{\partial\lambda} = f$ and by virtue of the systems dynamic constraint imposed on problem (3.4), the first term of the integrand is vanishing. Proceeding to the second term; after realizing that we have the full freedom to dictate the time-evolution of the adjoint variable λ, which is devised and introduced by *us*, we might use this freedom constructively by requiring that λ has to be governed by a differential equation such that the second term in equation (3.7) is vanishing as well, that is

$$\dot{\lambda} = -\frac{\partial\mathscr{H}}{\partial x}. \tag{3.8}$$

Next, since Δu is unconstrained and we do not want ΔJ to be negative (that would contradict the optimality of the solution), it follows that in order for a point to be optimal the control must necessarily satisfy (3.9)

$$\frac{\partial\mathscr{H}}{\partial u} = 0, \tag{3.9}$$

explicitly assuming that $\mathscr{H}$ is a differentiable function. However, by employing a more solid proof (cf. [18]), condition (3.9) can be strengthened to $u^* = \arg\max \mathscr{H}$ for all t in the domain, allowing it to have discontinuities.
We know that the final state, which due to the variation in time, is reached at $t = t_f + \Delta t$, is given, i.e. $[\Delta x]_{t=t_f+\Delta t} = 0$. Alternatively, we write the first order contribution to the end-point variation as

$$\begin{aligned}\Big[\Delta x\Big]_{t=t_f+\Delta t} &= \Big[\Delta x\Big]_{t=t_f} + \int_{t_f}^{t_f+\Delta t} f(x(t), u^*(t))\,\mathrm{d}t \\ &= \Big[\Delta x\Big]_{t=t_f} + f(x(t_f), u^*(t_f))\Delta t \\ &= \Big[\Delta x\Big]_{t=t_f} + \Big[\dot{x}\Big]_{t=t_f}\Delta t \\ &= \Big[\Delta x + \dot{x}\Delta t\Big]_{t=t_f} \end{aligned} \tag{3.10}$$

which is to be recognized as the parenthesis that accompany λ^T in equation (3.7). Hence, even this term is vanishing.
Finally, since the variation in final time Δt is arbitrary, it follows that at an optimal point

$$\Big[\mathscr{H}\Big]_{t=t_f} = 0. \tag{3.11}$$

Putting it all together, we have seen that a point is optimal only if

$$\begin{cases} \dot{\lambda} = -\frac{\partial \mathscr{H}}{\partial x} \\ \frac{\partial \mathscr{H}}{\partial u} = 0 \\ \Big[\mathscr{H}\Big]_{t=t_f} = 0 \end{cases} \tag{3.12}$$

Let us now study the time evolution of $\mathscr{H}$. We know that $\mathscr{H}$ is a function of the state vector x, the vector of auxiliary variables λ and the control u, which themselves are functions of time. Therefore the time-derivative of $\mathscr{H}$ becomes

$$\dot{\mathscr{H}} = \frac{\partial \mathscr{H}}{\partial x}\dot{x} + \frac{\partial \mathscr{H}}{\partial \lambda}\dot{\lambda} + \frac{\partial \mathscr{H}}{\partial u}\dot{u}. \tag{3.13}$$

Compiling (3.12) and (3.3) we have

$$\begin{cases} \frac{\partial \mathscr{H}}{\partial x} = -\dot{\lambda} \\ \frac{\partial \mathscr{H}}{\partial \lambda} = f = \dot{x} \\ \frac{\partial \mathscr{H}}{\partial u} = 0 \end{cases} \tag{3.14}$$

which together with (3.13) yields $\dot{\mathscr{H}} = 0$, i.e. $\mathscr{H} = \textit{constant}$ and from the last constraint in (3.12), we obtain that

$$\mathscr{H} = 0, \quad \forall t \in [0, t_f]. \tag{3.15}$$

Since we have chosen u such that $\mathscr{H}$ is optimal ($\frac{\partial \mathscr{H}}{\partial u} = 0$), it follows from (3.3), that we are permitted to draw the same conclusion about the Hamiltonian function $H(x, \lambda) = \max_u \mathscr{H}(x, \lambda, u)$. Hence

$$H = 0, \quad \forall t \in [0, t_f]. \tag{3.16}$$

By introducing the auxiliary variables and defining the Hamiltonian as appearing in (3.3), Pontryagin transferred the optimal control problem (3.1) to a problem in its

cotangent bundle, with the same optimal value. The problem of finding the optimal control, then reduces to determinating the pointwise maximum of the Hamiltonian function. It must be emphasized that the Pontryagin Maximum Principle merely provides *necessary* conditions for optimality. PMP must be viewed as a method for finding candidates for optimal control. To extract the global optimum, we must rule out the non-optimal candidates. If existence is guaranteed and the number of candidates are reasonable, simple comparison will do the work.

From the problem statement (3.1) and proposition 8, in conjunction with what has been said in the previous paragraph, it can be concluded that the following steps are to be taken in order to apply PMP to an optimal control problem:

Step 1 Define the Hamiltonian:

$$\mathscr{H}(x, \lambda, u) = \lambda^T f - \mathcal{L}$$

Step 2 Perform the pointwise maximization:

$$\begin{aligned} u^* &= \arg\max_{u \in \mathcal{U}} \mathscr{H}(x, \lambda, u) \quad \text{for all} \quad t \in [0, t_f] \\ H(x, \lambda) &= \mathscr{H}(x, \lambda, u^*) \end{aligned} \tag{3.17}$$

Step 3 Solve the Two Point Boundary Value Problem (TPBVP):

$$\begin{aligned} \dot{x} &= \ \ \frac{\partial H(x,\lambda)}{\partial \lambda} \\ \dot{\lambda} &= -\frac{\partial H(x,\lambda)}{\partial x} \end{aligned} \tag{3.18}$$

where $x(0) = X_i$ and $x(t_f) = X_f$.

Step 4 Compare the candidates given by PMP to extract the global optimum.

In this line of action, the major difficulty is undoubtedly to solve the TPBVP, which is a mixed boundary value problem, i.e. the boundary condition for some of the parameters are given at both the initial and final time, while other parameters are unrestricted. In section 3.2, we will see that in more general problems, for instance when we require the initial- and/or final state to belong to smooth manifolds, solving TPBVP will become an even more delicate problem. The reason for this is that these generalizations impose, what we call a transversality condition on the auxiliary variables (cf. equation (3.20) and (3.22)). As we shall see in chapter 6, the success of solving the mixed boundary value problem depends to a large extent on whether or not it is possible to obtain closed-form solution of the Hamiltonian system of differential equations (3.18). If no closed-form solution is available, we must take recourse in numerical methods such as the shooting method. This approach is based on successive improvements of the unspecified initial or terminal condition. Although conceptually simple, this approach suffers from the serious drawback of it being hard to find good initial estimates of the unspecified parameters. This fact will be treated more thoroughly in chapter 6.

3.2 PMP Generalizations

Thus far we have stated and presented a solution method for a very special optimal control problem, namely when we strive to control an autonomous system[1] between

[1]The term autonomous, refers to the fact that there is no explicit time dependence in the system dynamics nor the integral cost function.

two a priori given points in the state space, while minimizing a cost function in integral form (3.1). This problem can be extended and modified in various ways. The following generalizations are to be discussed in greater detail below:

- initial and/or final configurations are not specified, but are required to belong to smooth manifolds.
- the cost function includes, in addition to the integral term, an initial and/or final cost term.
- the system dynamics or the optimal control problem itself, are explicitly time-dependent.

First off, rather than specifying the initial and/or final state, one might instead find it more appropriate to require them to belong to smooth manifolds. This wish is highly motivated, since in some applications, rather than designating a fixed value to a specific state variable, it is more relevant to specify an interval for it. In the formerly mentioned rocket example (see section 2.1), where the amount of remaining fuel (henceforth denoted as ι), could serve as an eligible state variable, it might be hard to specify boundary values for it. We have an ambition not to hump any extra weight, so we want the amount of remaining fuel to be as close to zero as possible at the arrival time. But what about the initial value? We do not know the length of the optimal rocket trajectory, which makes it impossible to calculate how much fuel we should take on board at launching moment. Therefore, we might prescribe reasonable minimum and maximum values on $\iota(0)$, so that we require $\iota(0) \in [\iota_{\min}, \iota_{\max}]$.

We shall assume, that the prescribed set of allowed states at the initial- and final time, M_i and M_f, are both smooth manifolds. A $(n-k)$-dimensional smooth manifold is given by

$$M = \{x \in \mathbb{R}^n : \Omega(x) = 0\}, \qquad \text{where} \quad \Omega(x) = \begin{bmatrix} \Omega_1(x) \\ \vdots \\ \Omega_k(x) \end{bmatrix} \tag{3.19}$$

that is, it is the intersection of k smooth hyper-surfaces[2], whose equations are $\Omega_i(x) = 0, \;\; i = 1, 2, \ldots, k$. Furthermore we shall assume that the gradients $\nabla\Omega_i(x)$ are linearly independent so that M possesses a unique tangent plane at every point $x \in M$. This is equivalent to requiring that $\nabla\Omega(x)$, the Jacobian matrix of $\Omega(x)$, has full rank k.

Upon introducing the possibility to control a system between two smooth manifolds, instead of between two given states, we have to re-state proposition 8 in the following manner:

Proposition 9 (PMP - Endpoint Manifolds) *Suppose u^* transfers the system from a state in the initial manifold M_i to a state in the final manifold M_f with minimum cost (i.e. it is optimal). Then the following are necessary conditions:*

1. *There exists a non-zero adjoint variable λ, such that $\dot{\lambda} = -\frac{\partial H}{\partial x}$*
2. $u^* = \arg\max \mathscr{H}(x, \lambda, u) \qquad \forall t \in [t_i, t_f]$
3. $H(x, \lambda) = 0 \qquad \forall t \in [t_i, t_f]$
4. *The vector $\lambda(t)$, at $t = t_i$ and $t = t_f$ respectively, is orthogonal to the end-point manifolds M_i and M_f.*

[2] A surface described by $\Omega_i(x) = 0$ can be shown to be smooth on a domain if $\Omega_i(x)$ is continuously differentiable and $\nabla\Omega_i(x)$ does not vanish on that domain.

The last added condition is also called the *transversality condition* imposed on λ and is denoted as $\lambda \perp M$. More precisely, it expresses the condition that λ has to be perpendicular to every vector that belongs to the end points tangent spaces. That is

$$\lambda^T \varpi = 0, \qquad \forall\, \varpi : \nabla\Omega(x^*)\varpi = 0 \tag{3.20}$$

This transversality condition is to hold at both the initial and final time instant. It is also notable that, although the last condition looks like an addendum to proposition 8, it can be interpreted as when the end manifolds consist of only one single point, then *all* λ-vectors are normal to that point's tangent space and thereby fulfills the transversality condition. That explains why λ appeared to be unrestricted in proposition 8 when both the initial and final states were given.

Remark 7 Remark 4-6 apply to proposition 9 as well.

Next up, is to discuss the generality of the cost function given in (3.1). The integral cost function $\mathcal{L}$, assigns a cost to the state and control trajectories. It is however not hard to think of cases when in addition to this, we might want to inflict a penalty on, or with a different point of view, privilege some special end point configurations. Recurring to the state variable that expressed the amount of remaining fuel in the rocket control example, it was desirable to have $\iota(t_f) \gtrsim 0$. To express this desire, it is recommended to add $\Psi_f(x(t_f)) = \iota^2(t_f)$, to the cost function which is to be minimized[3]. This motivates generalizing the cost function given in (3.1), by appending two end point cost functions to it, so that it becomes

$$J = \Psi_i(x(t_i)) + \int_{t_i}^{t_f} \mathcal{L}(x(t), u(t))\,\mathrm{d}t + \Psi_f(x(t_f)), \tag{3.21}$$

where Ψ_i is the cost that penalize deviation from the desired initial set of configurations, while Ψ_f is the corresponding final cost. These two boundary point costs are assumed to be continuous differentiable functions.
We are now set to re-state proposition 9 so that it applies to optimal control problems between two manifolds, in presence of end point cost terms:

Proposition 10 (PMP - Endpoint Manifolds and Costs) *Suppose u^* transfers the system from a state in the initial manifold M_i to a state in the final manifold M_f while minimizing the cost function J given in (3.21) (i.e. it is optimal). Then the following are necessary conditions:*

1. *There exists a non-zero adjoint variable λ, such that $\dot{\lambda} = -\frac{\partial H}{\partial x}$*
2. $u^* = \arg\max \mathscr{H}(x, \lambda, u) \quad \forall t \in [t_i, t_f]$
3. $H(x, \lambda) = 0 \;\; \forall t \in [t_i, t_f]$
4. *The vector $[\lambda(t) + \nabla\Psi(x^*(t))]$ at $t = t_i$ and $t = t_f$ respectively, is orthogonal to the end manifolds M_i and M_f.*

Another equivalent formulation of this modified transversality condition is

$$[\lambda(t) + \nabla\Psi(x^*(t))]^T \varpi = 0 \qquad \forall\, \varpi : \nabla\Omega(x^*(t))\varpi = 0 \quad t \in \{t_i, t_f\} \tag{3.22}$$

Remark 8 Remark 4-6 on proposition 8 apply to proposition 10 as well.

[3]The non-negativity constraint imposed on ι is to be taken care of when defining $\mathcal{X}$ and can therefore be neglected here.

So far, we have treated *autonomous* systems which are transferred between points on two *fixed* manifolds in state space. We have also disregarded the possibility that the integral cost $\mathcal{L}$ or the end-point costs, might be explicitly time dependent. We shall now see how the maximum principle is modified when the system is non-autonomous and the integral- and boundary point costs, as well as the end point manifolds are time-dependent. Hence, we shall state the maximum principle for the following optimal control problem

$$\min \left[\Psi_i(t_i, x(t_i)) + \int_{ti}^{tf} \mathcal{L}(t, x(t), u(t))\, \mathrm{d}t + \Psi_f(t_f, x(t_f)) \right] \tag{3.23}$$

subject to

$\dot{x}(t) = f(t, x(t), u(t))$	non-autonomous system dynamics
$x(t_i) \in M_i(t_i)$	time-dependent initial condition
$x(t_f) \in M_f(t_i)$	time-dependent final condition
$x(\cdot) \in \mathcal{X} \subseteq \mathbb{R}^n$	state constraint
$u(\cdot) \in \mathcal{U} \subseteq \mathbb{R}^m$	control constraint
$t_f \in (t_i, \infty)$	
where	
$\mathcal{L} : \mathbb{R} \times \mathcal{X} \times \mathcal{U} \to \mathbb{R}$	Lagrange/penalty function
$f : \mathbb{R} \times \mathcal{X} \times \mathcal{U} \to T\mathcal{X}$	dynamics
$\Psi_i, \Psi_f : \mathbb{R} \times \mathcal{X} \to$	initial- and terminal costs
$M_i(t), M_f(t)$	smooth manifolds

In addition to this, as always, we assume that $u(\cdot)$ is a piecewise continuous function. Here-below, proposition 11 shows that for time dependent optimal control problems, the Hamiltonian is no longer a constant.

Proposition 11 (PMP - General) *Suppose u^* is an optimal solution to optimal control problem (3.23). Then the following are necessarily conditions:*

1. *There exists a non-zero adjoint variable $\lambda(t)$, such that $\dot{\lambda}(t) = -\frac{\partial H(t,x(t),\lambda(t))}{\partial x}$*
2. $u^* = \arg\max \mathscr{H}(x, \lambda, u) \quad \forall t \in [0, t_f]$
3. *The Hamiltonian $H(t, x(t), \lambda(t)) = \max_u \mathscr{H}(t, x(t), \lambda(t), u(t))$ satisfies:*
   $H(t, x(t), \lambda(t)) = \left[H(t_f, x(t_f), \lambda(t_f)) - \int_t^{t_f} \frac{\partial H(\tau, x(\tau), \lambda(\tau))}{\partial \tau} \mathrm{d}\tau \right], \quad t \in [t_i, t_f]$
   where the first term equals:
   $H(t_f, x(t_f), \lambda(t_f)) = -\sum_{i=1}^{k} \varpi_i \frac{\partial \Omega_i(t_f, x(t_f))}{\partial t} - \frac{\partial \Psi(t_f, x(t_f))}{\partial t}$
4. *The vector $[\lambda(t) + \frac{\partial \Psi(t,x(t))}{\partial x}]$ at $t = t_i$ and $t = t_f$ respectively, is orthogonal to the end manifolds M_i and M_f.*

Remark 9 Remark 4-5 made for proposition (8) apply to proposition 11 as well.

It is conceivable to formulate even more general control problems, for instance those involving time or state dependent $\mathcal{X}$ or $\mathcal{U}$. However, by means of a set of high-level operations, it is possible to re-formulate them so that the above stated proposition applies. To investigate the technical conditions under which these transformations may occur, consult [17].

4. Dubins' car

Dubins' problem[1] in $\mathbb{R}^2$ can be stated as follows: Given two points $X_i, X_f \in \mathbb{R}^2$ and two normalized velocity vectors V_i, V_f, find a curve $\gamma : [0, T] \to \mathbb{R}^2$ such that

1. $\gamma(s)$ is a continuously differentiable curve, parameterized by arclength.
2. $\|\dot{\gamma}\| = 1$ and satisfies the Lipschitz condition

$$\|\dot{\gamma}(s_2) - \dot{\gamma}(s_1)\| \leq \kappa_0 \, |s_2 - s_1|$$

   for all s_1, s_2 in the domain of γ.
3. γ satisfies the boundary conditions:

$$\gamma(0) = X_i, \quad \dot{\gamma}(0) = V_i, \quad \gamma(T) = X_f, \quad \dot{\gamma}(T) = V_f.$$

4. Among all possible curves satisfying 1-3, γ has shortest length, i.e. T is minimal.

Now, from 2, since Lipschitz functions are differentiable almost everywhere, it follows that γ has a curvature $\kappa = \|\ddot{\gamma}\|$, a.e. and that the curvature is upper bounded by κ_0. For convenience, this bound might be re-scaled to 1.

A very common interpretation to Dubins' problem concerns finding the shortest continuously differentiable path between two given points taken by a car, for which the starting and ending directions are specified. In addition, we assume that the car is moving with unit speed and subject to a minimum turning radius constraint.

This chapter's disposition is as follows; we start by formulating Dubins' problem as an optimal control problem of the form appearing in the beginning of chapter 3. Secondly, by applying the PMP results presented in section 3.1, we characterize the optimal control. It will be shown that optimal paths for "Dubins' car", consist of the concatenation of three distinctly different path segments; a straight line segment and two circular arcs of maximum curvature. These circular arcs correspond to the right- and left turn respectively. The task of determining how to combine these three pieces of elementary path segments, is a much more delicate matter. This difficulty was first solved by L.E. Dubins himself in [4], where he prescribed the set of *sufficient family* of optimal paths. In 1991, these results were slightly improved by Sussman and Tang in a very readable article [24]. It must also be mentioned that similar results were obtained independently and presented almost simultaneously by Boissonnat, Cerezo and Leblond in [2].

Upon presenting the characteristics of the sufficient family of paths for Dubins' car, $\mathcal{F}_D$, we make a remark that possessing a finite, and as a matter of fact small set of paths, which includes the optimal one, is clearly a very pleasant situation. The procedure of optimal path synthesis then boils down to computing the length of all the candidates and by comparison, selecting the minimal length path for the car - a trivial task for any modern computer. This is exactly what we take advantage of

[1] This problem was originally studied in the 1880's by Andrei A. Markov, who was looking for optimal railway constructions across the Russian landscape.

next, where we outline a procedure for motion planning for Dubins' car by, rather thoroughly, present an algorithm for finding the time-optimal path for Dubins' car. The nature of this algorithm is geometrical, which makes it highly suitable for numerical computations. The robot then ought to follow this synthesized path, using preferred path following technique (consult for instance [5] or [6]).

Formulating Dubins' problem as an Optimal Control problem In accordance with the problem formulation and interpretation presented above, our objective is to find the shortest possible continuously differentiable path inter-connecting two given points in $\mathbb{R}^2$, with prescribed tangent vectors at the end points. Thus, the quantity that we wish to minimize is

$$\int_0^T \sqrt{\dot{x}^2 + \dot{y}^2}\,\mathrm{d}t = \int_0^T |v|\,\mathrm{d}t \tag{4.1}$$

where (x, y) is the position of the vehicle and v is the linear velocity of the car. The unit speed assumption then, clearly makes this equivalent to a time optimal problem. Considering the car's dynamics, the unicycle robot model (cf. page 16) will be exploited. Hence, recurring to the optimal control problem formulation in the beginning of chapter 3, Dubins' problem can be stated as

$$\begin{array}{ll}
\text{minimize } \int_0^T 1\mathrm{d}t & \\
\text{subject to} & \\
\dot{x}(t) = [\cos\varphi(t), \sin\varphi(t), \omega(t)]^T & \text{system dynamics} \\
x(0) = [x_i, y_i, \varphi_i]^T = X_i & \text{initial configuration} \\
x(T) = [x_f, y_f, \varphi_f]^T = X_f & \text{final configuration} \\
x(\cdot) \in \mathcal{X} = \mathbb{R}^2 \times S^1 & \text{state constraint} \\
\omega(\cdot) \in [-1, 1] & \text{lateral velocity constraint} \\
T \in (0, \infty) & \text{final time (to be minimized)}
\end{array} \tag{4.2}$$

4.1 Characterizing the Optimal Control

Initially we follow the step-wise instructions given in section 3.1, describing how to solve optimal control problems with PMP. We define the Hamiltonian

$$\mathscr{H}(x, \lambda, \omega) = \lambda^T f - \mathcal{L} = \lambda_x \cos\varphi + \lambda_y \sin\varphi + \lambda_\varphi \omega - 1. \tag{4.3}$$

Next step is to recognize the optimal control ω^*, as the pointwise maximizing argument of the Hamiltonian

$$\omega^* = \underset{|\omega|\leq 1}{\arg\max}\ \mathscr{H}(x, \lambda, \omega) = \underset{|\omega|\leq 1}{\arg\max}\ \lambda_\varphi \omega. \tag{4.4}$$

We comfortably extract that the optimal control basically depends on the sign of λ_φ

$$\begin{cases} \lambda_\varphi < 0 \ \Rightarrow\ \omega^* = \ \ 1 & \text{corresponds to making a } \textit{left} \text{ turn} \\ \lambda_\varphi > 0 \ \Rightarrow\ \omega^* = -1 & \text{corresponds to making a } \textit{right} \text{ turn.} \end{cases} \tag{4.5}$$

But what happens if the *switching function* $\lambda\varphi \equiv 0$? If the switching function vanishes at a finite and isolated number of instants, this situation may be ignored, since the value of ω^* at these distinct points, neither affect the optimal solution nor the optimal cost. Contrary to this, if λ_φ vanishes on a non-zero time-interval $\mathcal{I}$, we must make further investigations. The switching function vanishing on a non-zero time interval is normally referred to as the so called *singular case*, about which it is often said

that "the pointwise maximization step of PMP does not provide any information about the optimal control". But in our particular case, it turns out that we are able to extract a considerable amount of supplementary information about the singular optimal control. Assume that $\lambda_\varphi \equiv 0$ on $\mathcal{I}$, then

$$\mathscr{H}|_{(\lambda_\varphi \equiv 0)} = \lambda_x \cos\varphi + \lambda_y \sin\varphi - 1 = 0, \tag{4.6}$$

where the last equality origins from the fact that, in accordance with proposition 8, the value of the Hamiltonian function for autonomous systems with unspecified terminal time is a known constant and equals zero.

Now, since there is no explicit $x-$ or $y-$dependence in $\mathscr{H}$, it follows from the adjoint equations that λ_x and λ_y are cyclic, i.e. constants. Since λ is a non-zero vector, (4.6) can only hold iff $\varphi(t)$ is a constant for all $t \in \mathcal{I}$, which imply that $\dot{\varphi}(t) = \omega(t) = 0$ in the non-zero time interval, $\mathcal{I}$. Constant orientation angle φ, naturally corresponds to advancing by driving on a straight line segment.
Before proceeding, let us summarize what we know thus far about the optimal control for Dubins' car

$$\omega^*(t) = \begin{cases} -1 & \text{if } \lambda_\varphi(t) < 0 \quad \text{(turn right)} \\ 0 & \text{if } \lambda_\varphi(t) = 0 \quad \text{(go straight)} \\ 1 & \text{if } \lambda_\varphi(t) > 0 \quad \text{(turn left)} \end{cases} \tag{4.7}$$

Notational Conventions

At this point, it is suitable to introduce some unified notation to describe these three different types of path segments. Let S denote the *straight line* segments and B denote the circular arcs, or *bends* that the optimal path consists of. At a sub-level, B includes both the right- and left turn, and whenever we wish to refer to them distinctly, they will be labeled as R and L. The duration of the time-interval spent on each path segment, which clearly coincide with its length, will be indicated using subscripts. Note that this does not exclude the possibility that a segment may have zero duration, i.e. be absent. So that, for instance $L_{\frac{\pi}{2}} S_{\mathfrak{a}} B_{\mathfrak{b}}$ might refer to, among others, any of the path segments in figure 4.1.

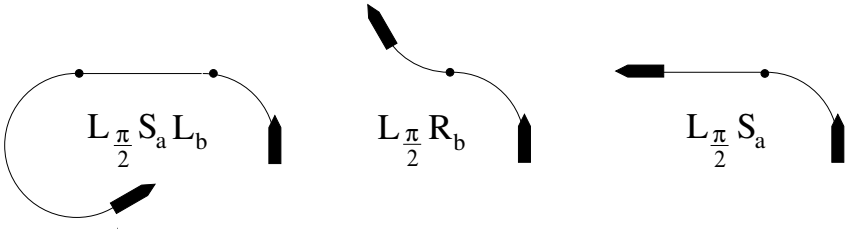


Figure 4.1: Examples of $L_{\frac{\pi}{2}} S_{\mathfrak{a}} B_{\mathfrak{b}}$ paths for Dubins' car.

4.2 Sufficient Family of Paths

While it is now clear that every time-optimal path for Dubins' car consists of a number of circular arcs of maximum curvature, in concatenation with straight line segments, the question of these path segments mutual order, number and duration are still open. When making an inquiry into this matter, it turns out that classical optimal control theory, to which PMP belongs, do not suffice to fully unveil this problem. The reason for this is, as indicated in section 3.1, PMP is a local criterion and merely provides *necessary* condition for optimality, which is often too weak and allows a large set of potential solutions. Furthermore, it can be laborious to extract

concrete information from PMP and it is hard to know when all accessible information is deduced. For these reasons, PMP alone is not an adequate tool to dissect Dubins' problem. Hence it has to be augmented by other techniques and concepts. The issue of the combination of the three basic path segments, was first solved by L.E Dubins himself in [4]. However, in [24], Sussman and Tang elegantly exhibit the power of the more modern geometric optimal control theory and, almost as a byproduct, extract the characteristics of time-optimal paths for Dubins', as well as Reeds-Shepp's car (chapter 5). Their main result on the structure of time-optimal paths for Dubins' car, which in fact is a slight improvement of the results obtained by Dubins in [4], is formulated in the following proposition.

Proposition 12 *Every time-optimal path for Dubins' car in* $\mathbb{R}^2$*, is of form* $B_{\mathfrak{a}}S_{\mathfrak{b}}B_{\mathfrak{c}}$ *or* $B_{\mathfrak{d}}B_{\mathfrak{e}}B_{\mathfrak{f}}$*, where the subscripts lay information about the duration of each segment. These time-indices must obey the following constraints:*

(i)	$\mathfrak{a}, \mathfrak{c} \in [0, 2\pi)$
(ii)	$\mathfrak{b} > 0$
(iii)	$\mathfrak{e} \in (\pi, 2\pi)$
(iv)	$\min\{\mathfrak{d}, \mathfrak{f}\} < \mathfrak{e} - \pi$
(v)	$\max\{\mathfrak{d}, \mathfrak{f}\} \leq \mathfrak{e}$

Naturally there is a non-negativity constraint imposed on the time-indices as well and as indicated before, it is admissible that one or two of these indices are vanishing on each path.

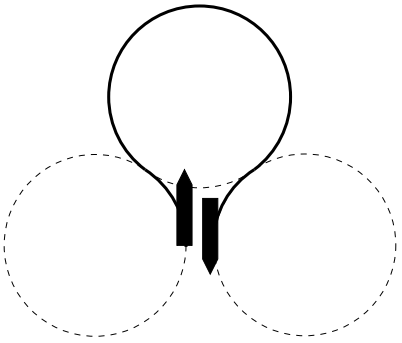


Figure 4.2: Illustration of a BBB type of path for Dubin's car.

We will proceed by defining the concept of sufficient family:

Definition 10 (Sufficient Family) *A collection of paths* $\mathcal{F}$*, is said to be a* sufficient family *for a problem if, given any two points* X_i *and* X_f*, there exists an element in* $\mathcal{F}$ *that goes from* X_i *to* X_f *and is both admissible and optimal.*

Since this definition merely requires that $\mathcal{F}$ contains the optimal path, we conclude that the substance of $\mathcal{F}_D$ is given by proposition 12. This enables us to present the characteristics of the sufficient family for Dubins' car.

Property 1 (Sufficient Family for Dubins' Car) *The following six elements are the constituents of* $\mathcal{F}_D$*:*

1. $R_{\mathfrak{a}}S_{\mathfrak{b}}R_{\mathfrak{c}}$
2. $R_{\mathfrak{a}}S_{\mathfrak{b}}L_{\mathfrak{c}}$
3. $L_{\mathfrak{a}}S_{\mathfrak{b}}R_{\mathfrak{c}}$

4. $L_{\mathfrak{a}} S_{\mathfrak{b}} L_{\mathfrak{c}}$
5. $R_{\mathfrak{d}} L_{\mathfrak{e}} R_{\mathfrak{f}}$
6. $L_{\mathfrak{d}} R_{\mathfrak{e}} L_{\mathfrak{f}}$

where the time-indices obey the constraints specified in proposition 12.

Possessing $\mathcal{F}_D$, genuinely facilitates any effort to execute on-line path planning, since it adds very strong and tight bounds on the number of eligible candidates. This being an negligible number, plays a crucial role for real-life control scenarios, since searching for an optimum among ten or so candidates, is a minor task for any modern computer. How to perform this search is the scope of the next section, but as for now, we can establish the fact that once we have synthesized the optimal path, we are to follow it. This is done by adopting an appropriate path following scheme, for instance the virtual vehicle approach (see e.g. [5] or [6]). This completes our search for an optimal as well as numerically feasible motion planner. The path planning part consists of path synthesis within the set of sufficient family (cf. section 4.3), while path following is made possible adopting for instance the virtual vehicle technique given in [5].

4.3 Optimal Path Synthesis

Although it is very satisfying to be able to restrict the search for optimal paths to the set of sufficient family, the ultimate goal we want to reach is the exact specification of the optimal control law, allowing us to control the car between any two arbitrary points in the state space. The truly interesting issue is to depict the instructions or directives that have to be passed to the car for optimally linking every initial point X_i to a given terminal point, X_f. It is notable that our system being left-invariant, i.e. the system dynamics is indifferent for arbitrary translational and/or rotational transformation, we might without loss of generality choose X_f to be the origin. Another reason for doing so is that, due to the lack of symmetry properties that the unit speed assumption conveys, in most cases, the optimal path from X_i to X_f differs from the one leading from X_f to X_i.

Let $\mu(x)$ denote a function that captures these directives, i.e. the solution of $\dot{x} = f(x, \mu(x))$ is the optimal path interconnecting x and the origin, $\mathcal{O}$. The domain of $\mu(x)$ is required to be the state space $\mathcal{X}$ and its range, the control set $\mathcal{U}$. Due to Pontryagin, a more formal definition of μ, the so called *synthesis function*, can be found in [18]. For a given optimal control problem, we call the problem of finding $\mu(x)$ a *synthesis problem*, while generating the corresponding solution (i.e. the optimal path), is referred to as *path synthesis*. So that path synthesis, basically provides a systematic way to select an optimal path between two arbitrary configurations, inside the sufficient family. To make these concepts more concrete, consider the following example.

Example 2 (Path Synthesis) Consider the problem of steering Dubins' car between $X_i = [0,\, 4,\, \pi]^T$ and $X_f = [0,\, 0,\, 0]^T$. The two end point configurations as well as the optimal path can be seen in figure 4.3.

The explicit instructions for following the optimal path can simply be given as: "Turn left for $\frac{\pi}{2}$ time units (t.u.), then go straight for 2 t.u. and finally, make a left turn for another $\frac{\pi}{2}$ t.u.". These directives are captured by defining the *synthesis function* at X_i as

$$\mu(X_i) = \begin{cases} 0 & \text{if} \quad t \in [\frac{\pi}{2},\ \frac{\pi}{2} + 2] \\ 1 & \text{else} \end{cases}$$

The corresponding *optimal path synthesis* from X_i, is then represented by $L_{\frac{\pi}{2}} S_2 L_{\frac{\pi}{2}}$. Finally, in order to obtain the complete synthesis, the synthesis function and the optimal path synthesis have to be defined for all initial points $X_i \in \mathcal{X}$.

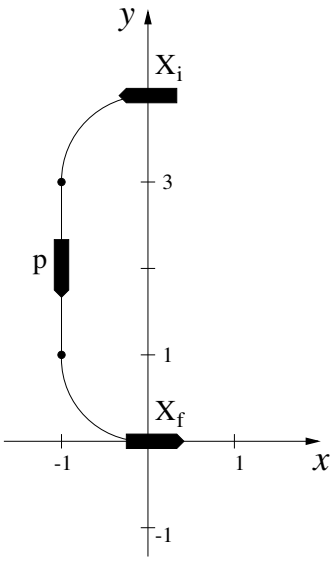


Figure 4.3: Optimal path interconnecting $X_i = [0,\ 4,\ \pi]^T$ and $X_f = [0,\ 0,\ 0]^T$.

Another customary and natural requirement on a synthesis, is that it is "memoryless", meaning that the instructions given by the synthesis function do not depend on the car's past history. For instance, the synthesis function at p in figure 4.3, should be

$$\mu(p) = \begin{cases} 0 & \text{if} \quad t \in [0,\ 1] \\ 1 & \text{else} \end{cases}$$

regardless if p is chosen as initial point, or is just passed cause it belongs to the path synthesis from X_i. This requirement has very close bonds to the principle of optimality. This is because μ completely determines the optimal path synthesis, and we know from the principle of optimality that an optimal solution is also optimal over any subinterval.

One must realize that the major difficulty in synthesis, mentioned at the very end of Example 2, is to define the synthesis for all admissible initial points. How is that viable? As for the Dubins' problem, we note that three pieces of such elementary geometrical shapes as straight lines and circular arcs, can be matched together by means of basic geometric reasoning. The insignificant number of candidates makes this approach adequate. However, by completing the local reasoning of PMP with more global geometric arguments, and making use of the problem's reduction properties, it is possible to construct the domains of validity for each type of path and even get analytic expressions for their boundaries. This work has been elegantly carried out by Souéres and Laumond in [22] for Reeds-Shepp's car, and jointly with Boissonnat and Bui in [3], in the case of Dubins' car. The partition of the state space varies continuously w.r.t. the initial orientation angle φ_i and gives rise to an efficient way of selecting the optimal path within the sufficient family. We will return to the results from these works later in chapter 5, where we study Reeds-Shepp's car. But as mentioned earlier, a purely geometrical approach suffice to synthesize the optimal control whenever the number of constituents are less or equal to three, as it is for the Dubins' car. In the sequel, our proposed algorithm for selecting the optimal path inside $\mathcal{F}_D$ will be presented. Paying attention to proposition 12, it is clear that whenever the euclidean distance between the initial and final configurations is greater than $6R$, where R is the minimum turning radius, the BBB type of paths are to be disregarded. The reason for this being that the final configuration then clearly is

beyond the range of any BBB path, making the $6R$ limitation, a self-evident but not necessarily tight upper bound. Now, assuming that R is much smaller than the distances we are interested in moving the car, we are entitled to, for the time being, overlook the BBB type of paths when presenting the algorithm.

Proposed Algorithm for Finding the Time-Optimal Path for Dubins' Car
Initial data are the given end point configurations $X_i = (x_i, y_i, \varphi_i)$ and $X_f = (x_f, y_f, \varphi_f)$. The notations used in the algorithm can be interpreted by means of figure 4.4. In addition, whenever what is stated, applies to both the initial and final points, the subscript specifying the end point will be dismissed, so for instance, r_c refers indistinctly to both r_{ci} and r_{cf}.

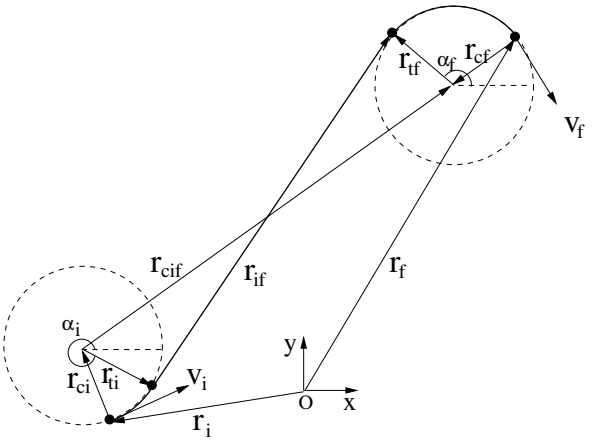


Figure 4.4: Used notations.

Step 1 From the given data, create the position vectors r and the velocity vectors V, as defined in figure 4.4.

Step 2 Find the center of the two circles at each end point, i.e. construct r_{c_+} and r_{c_-} by rotating V, $\pm 90°$ as shown in figure 4.5.

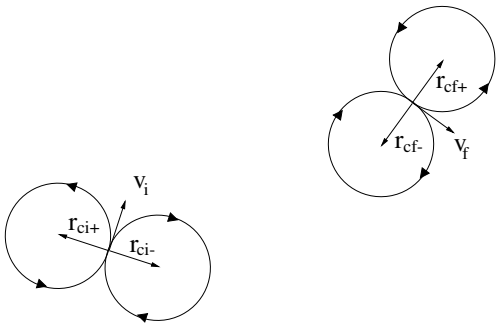


Figure 4.5: The center of the end point circles are found by rotating $V, \pm 90°$. Notice the enforced motion directions on the circles.

Now, from any of the two circles at the initial point, it is possible to construct four tangents to any of the two final point circles. But considering that the circles are directed, i.e. the circles centered at r_{c_-} and r_{c_+}, are only admissible for clockwise and anti-clockwise motion respectively, entitles us to rule out three of these four

tangents. This is done by imposing three constraints that the tangents have to fulfill, namely a *parallelity*, a *perpendicularity* and a *regularity constraint*. The source of these constraints originates from the following reasoning:
We start by defining $\alpha \in [0, 2\pi)$, as the angle between the horizontal axes and the sought tangent vector r_t. Notice that the prescribed unit length of the tangent vector, equalizes the search for α and r_t. We further define the *direction* of a circle as $d = v \times r_c$. By this means, we are able to compare the directions of two arbitrary circles that are to be connected, since

$$\begin{cases} d_i \cdot d_f = \ \ 1 & \Longrightarrow \quad \text{equi-directed circles} \\ d_i \cdot d_f = -1 & \Longrightarrow \quad \text{counter-directed circles} \end{cases}$$

The reason why this quantity is relevant is enlightened in figure 4.6, where it is shown how two of the four common tangents can be instantly dismissed by requiring the tangent vectors r_{ti} and r_{tf} to be parallel if the direction of the circles coincide, and to be anti-parallel otherwise. This requirement is more appropriately formulated as the following *parallelity constraint* on α:

$$|\alpha_f - \alpha_i| = \begin{cases} 0 & \text{equi-directed circles} \\ \pi & \text{counter-directed circles} \end{cases} \tag{4.8}$$

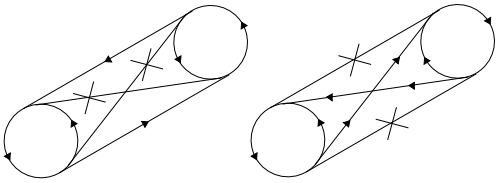


Figure 4.6: Two of the tangents can instantly be disregarded by imposing a parallelity constraint upon r_t.

Unfortunately, constraint (4.8) does not preclude the existence of non-admissible solutions of the form appearing in figure 4.7. Their presence is eliminated by the following *perpendicularity constraint*

$$r_t \perp r_{if}. \tag{4.9}$$

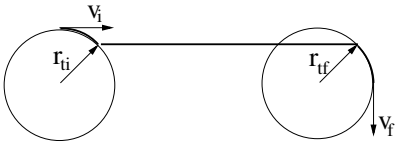


Figure 4.7: The parallelity of the tangent vector, is not a sufficient condition for path feasibility. The thicker line clearly is not a continuously differentiable path.

To exclude the third tangent, we draw our attention to figure 4.8. As we can see on the right-hand side, selecting the incorrect tangent vector, results in a non-smooth and thereby non-feasible solution. In order for a tangent vector to be regular, i.e. to have the proper orientation and structure such that the resulting path has both the feasible orientation and smoothness, it must satisfy the following *regularity constraint*

$$[d \times r_t] \cdot r_{if} > 0. \tag{4.10}$$

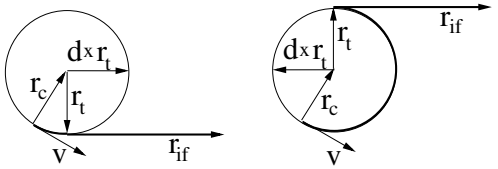


Figure 4.8: For admissible paths, $d \times r_t$ must not be anti-parallel to r_{if}.

Summarizing these three constraints, we set $d_i \cdot d_f = D$ and get a system of nonlinear equations for α

$$\begin{cases} |\alpha_f - \alpha_i| + (\frac{D-1}{2})\pi & = & 0 & \text{parallelity} \\ \big[\cos\alpha, \sin\alpha\big] \cdot r_{cif} + 1 - D & = & 0 & \text{perpendicularity} \\ D\big[\sin\alpha, \cos\alpha\big] \cdot r_{if} & > & 0 & \text{regularity} \end{cases} \tag{4.11}$$

We are now ready to state the third step of the algorithm.

Step 3 From each of the two circles at the starting point, find a tangent connecting it to each of the two end point circles, such that it satisfies (4.11). Practically, the most convenient way to achieve this, is in a two phase manner; firstly we find all the candidates that are solutions to the non-linear equation system consisting of the two upper constraints in (4.11), upon which we check regularity and select the proper solution by means of the last constraint.

As a final step, we are to calculate the length of each of the four remaining path candidates and appoint the optimal solution. Every BSB type of path, has three constituents. In accordance with proposition 12, let $\mathfrak{a}$ denote the length of the first circular path segment, $\mathfrak{b}$ the length of the straight tangent interconnecting the circles, and finally, $\mathfrak{c}$ denote the length of the final circle segment. We then trivially have

$$\mathfrak{b} = \|r_{if}\|. \tag{4.12}$$

As for the length of the circular path segments, we must not be as heedless. Let $\langle a, b\rangle$ denote the angle between two vectors, a and b. Judging from figure 4.4, it might be tempting to set $\mathfrak{a} = \langle r_{ti}, -r_{ci}\rangle$ and $\mathfrak{c} = \langle r_{tf}, -r_{cf}\rangle$. However, since the angle between two vectors, is usually calculated as $\langle a, b\rangle = \arccos\big[\frac{a \cdot b}{\|a\|\,\|b\|}\big]$, and the inverse cosine function's range is $[0, \pi]$, this procedure returns the smallest, or inner angle between the vectors. But as it can be noted on the right-hand side in figure 4.9, for some configurations, we are instead interested in the outer angle, so that $\mathfrak{a} = 2\pi - \langle r_{ti}, -r_{ci}\rangle$ and $\mathfrak{c} = 2\pi - \langle r_{tf}, -r_{cf}\rangle$.

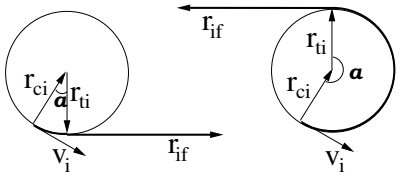


Figure 4.9: For some configurations, the outer angle between r_t and $-r_c$ is to be considered.

But what condition specifies this? It turns out that the decisive condition varies depending on which of the end points we are considering. At the initial point, whenever $\langle r_{ti}, V_i\rangle$ is greater than $\frac{\pi}{2}$, we spend more than π t.u. on the circular path and therefore have to calculate the outer angle, while $\langle r_{tf}, V_f\rangle < \frac{\pi}{2}$ is the proper condition at the final point.

Having clarified this, we can calculate the length of the circular path segments according to the pseudo-code given in (4.13).

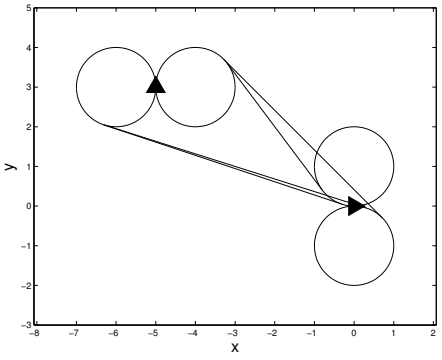
Length of initial circle, $\mathfrak{a}$	Length of final circle, $\mathfrak{c}$
if $\langle r_{ti}, V_i \rangle < \frac{\pi}{2}$	if $\langle r_{tf}, V_f \rangle > \frac{\pi}{2}$
$\mathfrak{a} = \langle r_{ti}, -r_{ci} \rangle$	$\mathfrak{c} = \langle r_{tf}, -r_{cf} \rangle$
else	else
$\mathfrak{a} = 2\pi - \langle r_{ti}, -r_{ci} \rangle$	$\mathfrak{c} = 2\pi - \langle r_{tf}, -r_{cf} \rangle$

(4.13)

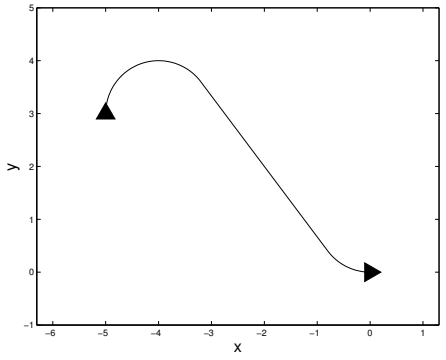
Step 4 Calculate the length of each of the four remaining candidates by taking the sum of $\mathfrak{a}, \mathfrak{b}$ and $\mathfrak{c}$ and by comparison appoint the optimal solution. The time indices, $\mathfrak{a} - \mathfrak{c}$ are to be calculated as put forward in (4.12) and (4.13).

It must be mentioned, that by defining α as the angle between r_c and r_t, instead of the angle between the horizontal axes and r_t, we would have avoided the implication we encountered when trying to calculate the length of the circular paths segments. The price of that being, a less neat parallelity constraint on α in the previous step.

Results from simulations made in Matlab, using the above described algorithm can be seen in figure 4.10.



(a) Candidates for optimal path after the third step. The task is to steer from initial configuration $X_i = (-5, 3, \frac{\pi}{2})$, to final configuration $X_f = (0, 0, 0)$



(b) In the last step, we appoint the optimal solution by calculating and comparing the length of each of the four remaining paths shown in figure 4.10(a).

Figure 4.10: Simulation results using the above described geometrical algorithm, for finding the time-optimal path for Dubins' problem.

5. Reeds-Shepp's car

In 1990, Reeds and Shepp [20], extended Dubins' pioneering work, which was the scope of the previous chapter, by augmenting the car with a reverse gear. The problem that Reeds and Shepp (henceforth abbreviated as RS) considered, is best interpreted as finding the shortest continuous differentiable path interconnecting two given points, taken by a car with prescribed end-point directions. As a further constraint, it is assumed that the car has limited steering angle and is moving with constant speed in the plane but, contrary to Dubins' car, has the possibility to perform backwards motions as well. Considering the car dynamics, the unicycle robot model (2.10) will be exploited. The allowance of reversed movement is the only point that separates the RS-car from that of Dubins. Hence, recurring to the optimal control problem formulation in chapter 3, RS's problem can be stated as

$$
\begin{array}{ll}
\text{minimize } \int_0^T 1 \mathrm{d}t & \\
\text{subject to} & \\
\dot{x}(t) = [\upsilon(t)\cos\varphi(t), \upsilon(t)\sin\varphi(t), \omega(t)]^T & \text{system dynamics} \\
x(0) = [x_i, y_i, \varphi_i]^T = X_i & \text{initial configuration} \\
x(T) = [x_f, y_f, \varphi_f]^T = X_f & \text{final configuration} \\
x(\cdot) \in \mathcal{X} = \mathbb{R}^2 \times S^1 & \text{state constraint} \\
\upsilon(\cdot) \in \{-1, 1\} & \text{linear velocity constraint} \\
\omega(\cdot) \in [-1, 1] & \text{lateral velocity constraint} \\
T \in (0, \infty) & \text{final time (to be minimized.)}
\end{array}
\tag{5.1}
$$

Although appearing insignificant, allowing the car to perform backwards motions, turns out to have implications on various issues, including the car's controllability and symmetry properties. For instance, since RS's problem is symmetric, it is possible to use LARC (cf. proposition 7) in order to show that it is locally controllable from everywhere in $\mathcal{X}$. On the contrary, Dubins' car is a system with a drift, controllable but not locally controllable from everywhere. This is, of course, due to the inability to move backwards, hence it takes, for instance $2\pi - \epsilon$ units of time to steer from $X_i = (0, 0, 0)$ to $X_f = (-\epsilon, 0, 0)$. The differences in symmetry properties are also natural. In the case of the RS-car, since the linear velocity might change sign, we can persuade ourselves that the optimal path leading from X_i to X_f is also optimal for the problem of optimally steering from X_f to X_i. It is possible to follow every path backwards. This is clearly not the case for Dubins' car, therefore the optimal path leading from X_i to X_f is always different from the one leading in the opposite direction.
However, as will be seen in the sequel, Dubins' and the RS-car, share a significant amount of common ground as well. Consider for instance the RS-car between two consecutive cusp-points, i.e. a subsection of a optimal path for which the linear velocity is kept constant, $\upsilon = \pm 1$. Every time-optimal path for the RS-car between

two cusps is clearly also optimal for Dubins' car and hence belongs to $\mathcal{F}_D$, the sufficient family of paths for Dubins' problem. Notice that this observation is not conversive, since by executing backwards movement, the RS-car possesses an extended set of motion abilities. A path γ may be optimal for Dubins' problem, but by using the possibility to move backwards, a RS-car might execute a motion along a path $\hat{\gamma}$ which is shorter than γ but non-feasible for Dubins' car.

The outline of this chapter will, to a large extend, resemble that of the previous one. In section 5.1, we start by applying PMP to problem (5.1) to characterize the optimal control. As in the case of Dubins' car, we are to establish the fact that every time-optimal solution to problem (5.1), consists of the concatenation of straight line segment and circular arcs or bends of maximum curvature. But considering the possibility to move both forward and backward, as well as to turn in both directions, we have a total of six elementary path types which are to be combined in an appropriate manner to construct the optimal path (cf. figure 5.1). The problem of these elementary path segments' mutual order, number and duration is solved in section 5.2 where we utilize the results in [24] and [22] to define the sufficient family of paths, $\mathcal{F}_{RS}$, and synthesize the optimal path, linking any given initial configuration to the origin.

5.1 Optimal Control Characterization

As in the case of Dubins' car, we outset by defining the Hamiltonian

$$\begin{aligned} \mathscr{H}(x,\lambda,u) &= \lambda^T f - \mathcal{L} = \lambda_x \upsilon\cos\varphi + \lambda_y \upsilon \sin\varphi + \lambda_\varphi \omega - 1 \\ &= \upsilon[\lambda_x\cos\varphi + \lambda_y \sin\varphi] + \omega\lambda_\varphi - 1 = \upsilon\,\sigma_\upsilon + \omega\,\sigma_\omega - 1 \end{aligned} \tag{5.2}$$

Next, we are to determine the optimal control as the pointwise maximizing argument of the Hamiltonian

$$\upsilon^* = \underset{|\upsilon|=1}{\arg\max}\ \mathscr{H} = \underset{|\upsilon|=1}{\arg\max}\ \upsilon\,\sigma_\upsilon \tag{5.3}$$

$$\omega^* = \underset{|\omega|\le 1}{\arg\max}\ \mathscr{H} = \underset{|\omega|\le 1}{\arg\max}\ \omega\,\sigma_\omega. \tag{5.4}$$

Thus, the optimal control is *bang-bang* and essentially depends on the sign of the *switching functions*, σ_υ and σ_ω respectively. The argumentation for obtaining the optimal control in the lateral direction ω^*, is identical to the one outlined in the case of Dubins' car. Therefore, referring to equation (4.7) and the preceding reasoning, we recognize the optimal control in the lateral direction as

$$\omega^*(t) = \begin{cases} -1 & \text{if } \sigma_\omega < 0 \quad \text{(turn right)} \\ 0 & \text{if } \sigma_\omega = 0 \quad \text{(go straight)} \\ 1 & \text{if } \sigma_\omega > 0 \quad \text{(turn left)} \end{cases} \tag{5.5}$$

Proceeding to the optimal control in the linear direction υ^*, we wish to extract some information about the *singular case*, that is when $\sigma_\upsilon \equiv 0$ on a non-zero time interval $\mathcal{I}$.

$$\sigma_\upsilon \equiv 0 \quad \forall t \in \mathcal{I} \quad \Longrightarrow \quad \varphi \equiv -\arctan\frac{\lambda_x}{\lambda_y} \quad \forall t \in \mathcal{I} \tag{5.6}$$

Now, since there is no explicit x- or y-dependence in $\mathscr{H}$, it follows from the adjoint equations that λ_x and λ_y are cyclic, i.e. constants. Equation (5.6) then tells us that φ must be constant for all $t \in \mathcal{I}$. Constant orientation angle φ, obviously corresponds to moving along a straight line. From (5.5) we deduce that along a straight line, the switching function in the lateral direction σ_ω is also equal to zero. With both the switching functions vanishing, (5.2) yields $\mathscr{H} = -1$. This is a contradiction to

the third optimality condition stated in proposition 8, namely that the Hamiltonian function for an autonomous system with free terminal time is a constant and equalizes zero. Therefore we conclude that $\sigma_\upsilon \neq 0$ except at possibly a finite and distinct number of points, which are not going to affect neither the optimal control nor the optimal cost. Thus we might disregard the singular case in linear direction and obtain

$$\upsilon^*(t) = \text{sign}\ \sigma_\upsilon = \text{sign}\ [\lambda_x(t)\cos\varphi(t) + \lambda_y(t)\sin\varphi(t)] \tag{5.7}$$

Remark 10 The fact that the two switching functions σ_υ and σ_ω cannot vanish simultaneously, excludes the possibility that υ and ω switch at the same time.

Notational Conventions

Let us next take a look at the set of motions that the RS-car is able to execute and introduce a nomenclature for them. Between any two consecutive cusp-points, i.e. when we change the sign of υ, the RS-car's range of motion abilities, coincide with Dubins' car. Hence it seems convenient to reclaim the notational conventions introduced at the end of section 4.1 and complement it with some additional notation. Re-stating what has been introduced earlier, let S denote a *straight line* segment, R a *right turn*, while L signifies a *left turn*. We use B, whenever we wish to refer to the *bends* indistinctly, i.e. $B = \{R, L\}$. As before, we indicate the duration of each path segment, by using subscripts.
Proceeding to the the RS-car significant type of motions, let the direction of motion be specified by a superscript on a path segment. For forward motions, $'+'$ will be used, while $'-'$ will denote backward motions. The six basic path segments that are the constituents of every time-optimal path for the RS-car, are denoted as $R^{\pm}, L^{\pm}$ and $S^{\pm}$. Notice in figure 5.1 that the letters R/L, refer to the direction in which the velocity vector of the car is turning. Occasionally we may want to stress the fact that the velocity υ, changes sign. These cusp-points, will be denoted as $\curlywedge$. Augmenting this with remark 10, made above about the absence of cases when υ and ω switches simultaneously, enables us to refine and restrict our notations further. As an example, the notation B^+BB might refer to 8 different path combinations, while $B^+\curlywedge B\curlywedge B$ includes only 2 cases.

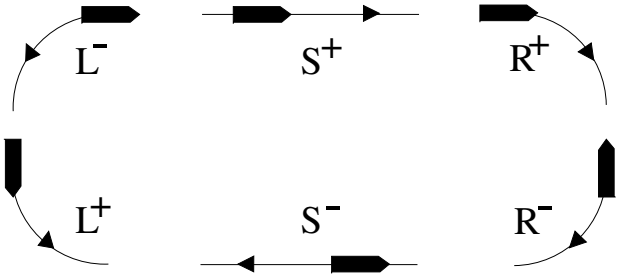


Figure 5.1: The six elementary path types for the RS-car.

5.2 Sufficient Family and Path Synthesis

The notion of *sufficient family* and *path synthesis* was introduced earlier in sections 4.2 and 4.3 respectively. For a brief overview of these concepts, consult these sections.

Contrary to Dubins, Reeds and Shepp did not provide a solid proof for the existence of optimal paths in [20]. Instead, they based their reasoning on specific *ad hoc* arguments from differential calculus and geometry, resulting in the sufficient family

of paths for their car, $\mathcal{F}_{RS}$. Their approach to find the right set of elements for the sufficient family, was based on the comparison of the length of various non-optimal paths. Starting with an admissible path γ from X_i to X_f, they made a direct study of the possibility of replacing γ with a shorter path, until it cannot be shortened further and happens to be of the special kinds appearing in $\mathcal{F}_{RS}$. The computer played a fundamental role in their discovery of the results and the following quote made by Reeds and Shepp in [20] (page 390), might illuminate the empirical spirit in that paper and explain why their efforts cannot serve as a framework for further studies.

> "There is another solution which we have never observed to be optimal and so doubtless can be discarded."

One of the main problems encountered in this pioneer work, was that since the control set for the RS-car is non-convex, the standard existence results for time-optimal trajectories from optimal control theory (i.e. Fillipov's theorem), no longer apply directly. This problem was set right by Sussman and Tang in [24], by replacing the non-convex control set $\mathcal{U} = \{-1, 1\}$ by its convex hull $\hat{\mathcal{U}} = [-1, 1]$ resulting in what the authors name the "Convexified Reeds-Shepp" problem (CRS), which then lends itself directly to the use of Follipov's theorem (cf. figure 5.2).

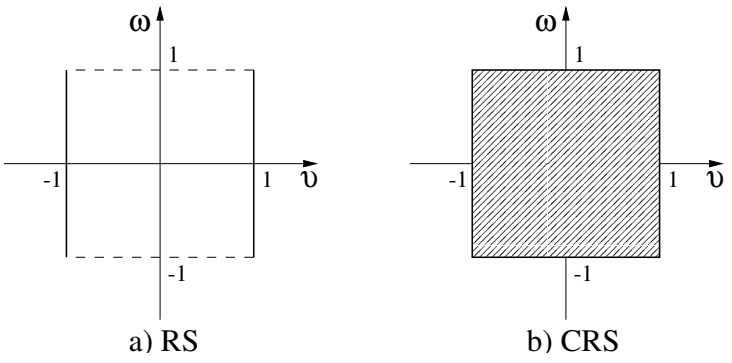


Figure 5.2: The control domains for Reeds-Shepp (RS) and Convexified Reeds-Shepp (CRS) problem.

The price one have to pay for using this finesse is that nothing guarantees that the obtained time-optimal paths are RS-admissible. This must be investigated explicitly. However, it turns out that the sufficient family of paths for CRS's problem, consists solely of S or B pieces. Therefore, proving regularity of the CRS solutions for the RS-car, will be done *a posteriori* by direct inspection.

The following proposition gathers the main results presented in [24] on time-optimal paths for the RS-car.

Proposition 13 *Let γ denote the time-optimal path for RS's problem in $\mathbb{R}^2$. Then γ is the concatenation of at most five pieces of elementary path segments (B or S pieces), of which at most four are B's and at most one is S. The possible combinations are*

(i) $B_{\mathfrak{a}}^{+} \curvearrowleft B_{\mathfrak{b}} \curvearrowleft B_{\mathfrak{c}}$
where $(\mathfrak{a} + \mathfrak{b} + \mathfrak{c}) \in [0, \pi]$

(ii) $B_{\mathfrak{a}} B_{\mathfrak{b}} B_{\mathfrak{c}}$ *or* $B_{\mathfrak{a}} B_{\mathfrak{b}} B_{\mathfrak{b}} B_{\mathfrak{c}}$
where $\mathfrak{a}, \mathfrak{c} \in [0, \mathfrak{b}]$, $\mathfrak{b} \in [0, \frac{\pi}{2}]$, *and the switchings alternate between* υ *and* ω.

(iii) $B_{\mathfrak{a}} S_{\mathfrak{b}} B_{\mathfrak{c}}$
where $\mathfrak{a}, \mathfrak{c} \in [0, \frac{\pi}{2}], \mathfrak{b} > 0$

(iv) $B_{\mathfrak{a}} \curvearrowleft B_{\frac{\pi}{2}} S_{\mathfrak{b}} B_{\frac{\pi}{2}} \curvearrowleft B_{\mathfrak{c}}$
where $\mathfrak{a}, \mathfrak{c} \in [0, \frac{\pi}{2}], \mathfrak{b} > 0$, *and the two middle* B*'s must be of different types.*

(v) $B_{\mathfrak{a}} \curvearrowleft B_{\frac{\pi}{2}} S_{\mathfrak{b}} B_{\mathfrak{c}}$ *or* $B_{\mathfrak{c}} S_{\mathfrak{b}} B_{\frac{\pi}{2}} \curvearrowleft B_{\mathfrak{a}}$
where $\mathfrak{b} > 0$ *and if all the* B*'s are similar,* $\mathfrak{a} + \mathfrak{c} \in [0, \frac{\pi}{2}]$, *else* $\mathfrak{a}, \mathfrak{c} \in [0, \frac{\pi}{2}]$

Remark 11 In [20], the original paper by Reeds and Shepp, group *(i)* was presented as $B_{\mathfrak{a}} \curvearrowleft B_{\mathfrak{b}} \curvearrowleft B_{\mathfrak{c}}$. This contributed with four of the 48 three-parameter path combinations, that formed their set of sufficient family. In [24] however, due to a more powerful lemma, the need for combinations $L^- L^+ L^-$ and $R^- R^+ R^-$ are ruled out by Sussman and Tang. This lowers the above mentioned 48 to a 46.

Remark 12 Notice that although some of the items consist of up to five pieces of elementary path types, all the groups have three undetermined time indices. This issue will become important when we want to synthesize the optimal path (see section 5.2).

Remark 13 The constraints on the time indices within group *(i)* and *(v)*, are due to Boissonnat *et al.* (see [2] for more details).

Now that we are familiar with the form of every optimal solution to RS's problem, let us characterize $\mathcal{F}_{RS}$ by specifying its substance.

Property 2 (Sufficient Family for Reeds-Shepp's Car) *Group (i)-(v) presented in proposition 13, are the constituents of* $\mathcal{F}_{RS}$.

It should be mentioned that it is possible to express $\mathcal{F}_{RS}$ in some less accurate but more well-presented ways. The contrary is also valid. Reeds' and Shepp's most general representation is: "$\mathcal{F}_{RS} = BBSBB$, where one or more of the arcs or segments may vanish". However, we found the representation in proposition 13, an appropriate compromise between legibility and correctness.

Optimal Path Synthesis

The concept and notion of *path synthesis* was introduced in section 4.3. Concisely, we ought to determine the optimal path linking any initial configuration X_i in the entire state space $\mathcal{X}$ to the origin. In the same section, we also presented an algorithm for finding the time-optimal path for Dubins' car model. The algorithm is based on geometrical reasonings and might be used whenever the candidate for optimal path, consists of no more than three pieces of elementary path segments. Although the possibility of generalizing the algorithm to enclose more general path combinations has not been investigated in any greater detail, it does seem viable. This belief is based on the fact that, although some of the candidates for optimal solution to RS's problem consist of up to five pieces of elementary path segments, none of them have more than three undetermined time indices. In other words, all the candidates for optimal solution, have three or less degrees of freedom.
For now though, we might utilize the achievements of Souéres and Laumond in [22]. There, they elegantly construct the domains of validity for each type of path and even get analytic expressions for their boundaries. These results have been gathered by completing the local reasoning of PMP with more global geometric arguments.

By making use of the RS's problem's reduction- and symmetry properties, they were able to narrow their study to the quadrant enclosed between two axes of symmetry. Moreover, they established that it suffices to only consider positive or negative values of φ_i. These observations significantly reduce the sufficient family. The outcome of this conducted research is a partition of the state space which varies continuously with respect to φ_i and gives rise to an efficient way of pinpointing the optimal path given X_i. To synthesize the optimal path for steering the car to the origin, it suffices to determine in which cell the starting point is located. The cell-decomposition of the planes P_{φ_i} are shown on pages 132-134 in [12], for several different values of φ_i. It is notable that the 46 path types never appear simultaneously in a plane P_{φ_i}. Table 5.1 presents boundaries on φ_i, for which each group of path exists.

group	path type	intervals of validity
(i)	$B_{\mathfrak{a}}^{+} \curlywedge B_{\mathfrak{b}} \curlywedge B_{\mathfrak{c}}$	$\varphi_i \in [-\pi, \pi]$
(ii)	$B_{\mathfrak{a}} B_{\mathfrak{b}} B_{\mathfrak{c}}$	$\varphi_i \in [-\pi, \pi]$
	$B_{\mathfrak{a}} B_{\mathfrak{b}} B_{\mathfrak{b}} B_{\mathfrak{c}}$	$\varphi_i \in [-\frac{2\pi}{3}, \frac{2\pi}{3}]$
(iii)	$B_{\mathfrak{a}} S_{\mathfrak{b}} B_{\mathfrak{c}}$	$\varphi_i \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ if $sign(\omega)$ changes
		$\varphi_i \in [-\pi, \pi]$ if $sign(\omega)$ is constant
(iv)	$B_{\mathfrak{a}} \curlywedge B_{\frac{\pi}{2}} S_{\mathfrak{b}} B_{\frac{\pi}{2}} \curlywedge B_{\mathfrak{c}}$	$\varphi_i \in [-2 \operatorname{arcot}(2), 2 \operatorname{arcot}(2)]$
(v)	$B_{\mathfrak{a}} \curlywedge B_{\frac{\pi}{2}} S_{\mathfrak{b}} B_{\mathfrak{c}}$ or	$\varphi_i \in [-\pi, \pi]$ if $sign(\omega)$ changes
	$B_{\mathfrak{c}} S_{\mathfrak{b}} B_{\frac{\pi}{2}} \curlywedge B_{\mathfrak{a}}$	$\varphi_i \in [-\pi, -\frac{\pi}{2}] \cup [\frac{\pi}{2}, \pi]$ if $sign(\omega)$ is constant

Table 5.1: Intervals of validity for the constituents of $\mathcal{F}_{RS}$

In some cells the optimal directives for reaching the origin is not uniquely determined, upon which an impartial choice can be made. Consequently the optimal path synthesis for the RS-car is not unique. Moreover, we have said that the cell-decomposition of the configurations space, reduces the path synthesis procedure to pinpointing the cell in which the starting point is located. Thereby we are able to recognize the optimal path type that steers the car to the origin. But one last obstacle, namely to determine the duration of each time index in the optimal path type, must not be overlooked. This issue still calls for a numerical algorithms similar to the one presented in section 4.3. However, the computational complexity of such an algorithm is highly reduced due to the very strong bounds that the partition of the plane P_{φ_i} induces.

6. Nearly Time-Optimal Paths

In the previous two chapters, the problem of finding time-optimal paths for Dubins' and Reeds-Shepp's car has been considered. By using the Pontryagin Maximum Principle (PMP), we concluded that such paths are made of circular arcs and straight line segments, all tangentially connected. These time-optimal solutions suffer from some drawbacks. Their discontinuous curvature profile, together with the wear and impairment on the control equipment that the bang-bang solutions induce, calls for "smoother" and more supple reference paths to follow. Our main tool for generating such nearly time-optimal paths is to make an appropriate and cunning choice of the integral cost function, $\mathcal{L}$. What objectives to consider when making that choice and a handful of more or less fruitful examples of suitable $\mathcal{L}$s are presented in section 6.1. A nearly time-optimal, but nevertheless smoother and more pliable solution, also brings more flexibility and robustness, in the presence of uncertainty. All the above mentioned advantages are illustrated in section 6.2 where some simulation results are presented. It turns out however, that the presented concept suffers from severe numerical instability properties. The origin of this undesirable behavior is revealed in section 6.3, where we establish the fact that the problem at hand is singular.
Finally, in an effort to reduce the numerical difficulties that the shooting methods brings, an alternative approach, *viz.* the **Method of Perturbation**, is adopted in section 6.4. By making a Taylor expansion about the synthesized time-optimal paths, we are able to study the first order contribution from a change in the design parameter, ε, to the change in the appearance of the generated paths.
Throughout this chapter, the convexified RS-car (CRS), will be exploited. That is a unicycle robot model, with a convexified control set (cf. page 46). It should also be clarified that whenever we wish to refer to the linear- or the lateral control indistinctly, we use u, so that $u \in \{v, \omega\}$.

6.1 Selection of the Lagrangian

Initially, we establish the fact that we have the full freedom to make a cunning choice for an arbitrary integral cost function, $\mathcal{L}$, for an optimal control problem (3.1). This freedom should be used constructively by selecting $\mathcal{L}$, with some practical, control-motivated reasons in mind. We consider three main objectives, individually contributing to the Lagrangian.

1. We have the obvious intention of reaching the terminal configuration as fast as possible. Hence, we wish to eliminate the solution candidates, in which unnecessarily amount of time is being wasted. This target is, as in the case of time-optimal control problems, reached by including an *integral constant*, $\mathcal{L}_1 = 1$ in the integral cost function, $\mathcal{L}$.

2. In order to avoid, or at least reduce, the drawbacks of time-optimal, bang-bang solutions, we wish to obtain smoother and more pliable paths. This issue involves introducing a *penalty* for steering the car with control values close to their boundary points (see figure 6.1). The penalty function enables us to encourage a more prudent and moderate driving style, avoid bang-bang solutions

and thereby obtain more supple paths. The embedded flexibility in such paths, also results in increased robustness in the presence of uncertainty. The most intuitive contribution of this objective to $\mathcal{L}$, is any even polynomial, for instance $\mathcal{L}_2 = \omega^2$.

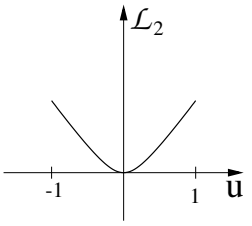


Figure 6.1: Moderate driving style is encouraged by means of a penalty function.

3. As discussed in chapter 3, technological limitations, such as the car's turning radius, normally leave us with a restricted control domain. Our third and last objective when selecting $\mathcal{L}$, is to handle input saturations in a convenient manner. A direct approach, involves adopting an integral cost function, having the form of a well (see figure 6.2). For instance

$$\mathcal{L}_3 = \begin{cases} 0 & \text{if } u \in [-1,\, 1] \\ \infty & \text{else} \end{cases} \tag{6.1}$$

provides a solid *barrier*, in which the control function is allowed to take its values. This idea is exploited in the two examples below. However, as illustrated in the up-following discussion, by making a more clever and cunning selection of $\mathcal{L}$, we show a more subtle and implicit way to handle input restrictions.

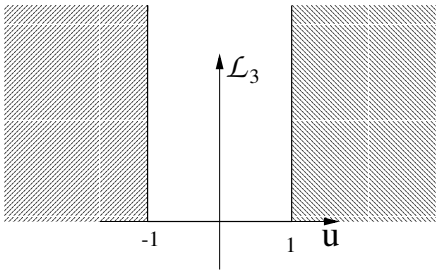


Figure 6.2: The well function, provides a restricting barrier for the domain of u.

The naive and adverse way to bond the contribution of our three objectives together, is to simply add them up, that is to set $\mathcal{L} = \sum_{i=1}^{3} \mathcal{L}_i$. We might however observe that by using superposition, the penalizing properties of $\mathcal{L}_2$ can be welded together with the off-barriering characteristics of $\mathcal{L}_3$ and expressed by means of a common cost function, $\mathcal{L}_0$. For instance, $\mathcal{L}_0 = u \tan u$ will have the same impact on an optimal control problem as $\mathcal{L}_2$ and $\mathcal{L}_3$ traced out in figure 6.1 and 6.2 respectively and can with advantage replace them both (cf. example 3).

To obtain even more flexibility and versatility, let us introduce, what we call a *design parameter* ε, in $\mathcal{L}_0$. Then by adjusting ε, we will be able to decide how close to the time-optimal solution we wish to find ourselves. Setting $\varepsilon = 0$ yields $\mathcal{L} = \mathcal{L}_1 = 1$ and thus equals this with the time-optimal problems considered in chapter 4 and 5. The outcome of this, will naturally be the same, i.e. the same set of time-optimal

solutions that were presented in the above mentioned chapters. Then by gradually tuning ε up, we penalize the usage of the boundary points of the control domain more and more, and hence put more value into a moderate driving style. This of course occurs at the expense of time-optimality.

For reasons of presentation, we will follow the natural line of thought when striving to find an appropriate choice of integral cost function $\mathcal{L}$. We will start with the most natural and intuitive guess, extract what is to be learned and proceed to more and more cunning and implicit choices of $\mathcal{L}$.

Example 3 (Taking $\mathcal{L}_0$ as a hybrid between a penalty function and a well.) Motivated by the foregoing discussion about our preferences when selecting Lagrangian and their contributions to it, it seems tempting to choose $\mathcal{L}_0$ as a hybrid between a penalty function and a well. Intuitively, this should capture the sought penalizing and restricting properties of $\mathcal{L}_2$ and $\mathcal{L}_3$. Plenty of such very natural, but nevertheless fruitless, attempts were made. A representative example of that being $\mathcal{L}_0 = \varepsilon\, u \tan u$. The polynomial term corresponds to the sought penalizing properties, while the tangent function provides a barrier which serves as the control domain (see figure 6.3).

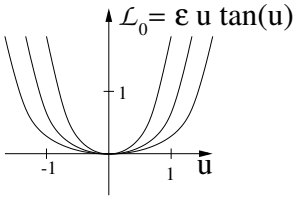


Figure 6.3: The Lagrangian for different values on ε when $\mathcal{L}_0$ is a hybrid between a penalty function and a well.

Let us examine what the characterization of the optimal control boils down to in this case. With $\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_1 = \varepsilon\, u \tan u + 1$, the Hamiltonian becomes

$$\mathscr{H} = \lambda^T f - \mathcal{L} = \lambda_1 v \cos\varphi + \lambda_y v \sin\varphi + \lambda_\varphi\, \omega - \varepsilon\, v \tan v - \varepsilon\, \omega \tan\omega - 1 \tag{6.2}$$

Maximizing equation (6.2) pointwise gives

$$\begin{cases} \frac{\partial \mathscr{H}}{\partial v} = \lambda_x \cos\varphi + \lambda_y \sin\varphi - \varepsilon\,[\tan v + \frac{v}{\cos^2 v}] = 0 \\ \frac{\partial \mathscr{H}}{\partial \omega} = \lambda_\varphi - \varepsilon\,[\tan\omega + \frac{\omega}{\cos^2 \omega}] = 0 \end{cases} \tag{6.3}$$

which is an implicit system of equations, for v and ω. Without any analytical expression of the pointwise optimal control, it is hard to proceed any further.

Example 4 (An analytically solvable example and a source of inspiration) Another intuitively justified candidate for an appropriate integral cost function, is

$$\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_1 = \frac{\varepsilon}{2\,(1-u^2)} + 1 \tag{6.4}$$

which is sketched in figure 6.4 and undeniably possesses both the desired penalizing and restricting properties. We will see, that this choice of integral cost function, will lead to analytically solvable equations for the optimal control. However, by means of a simple but cunning approximation trick, we will proceed and get extremely perspicuous and tidy expressions for the optimal control, simultaneously possessing all the desired qualities we seek. This approximation then, will serve as a source of inspiration for the section to follow, where we derive the same results without

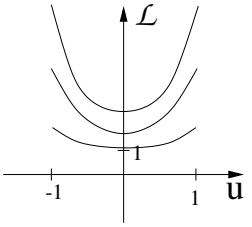


Figure 6.4: Integral cost function used in example 4, sketched for different values on ε.

employing the approximation step.
The Hamiltonian in this case becomes

$$\mathscr{H} = \lambda^T f - \mathcal{L} = \lambda_x v \cos\varphi + \lambda_y v \sin\varphi + \lambda_\varphi \omega - \frac{\varepsilon}{2\,(1-v^2)} - \frac{\varepsilon}{2\,(1-\omega^2)} - 1 \quad (6.5)$$

Maximizing this Hamiltonian pointwise, involves finding the roots of its partial derivatives

$$\frac{\partial \mathscr{H}}{\partial v} = \sigma - \frac{\varepsilon v}{(1-v^2)^2} = 0 \quad (6.6)$$

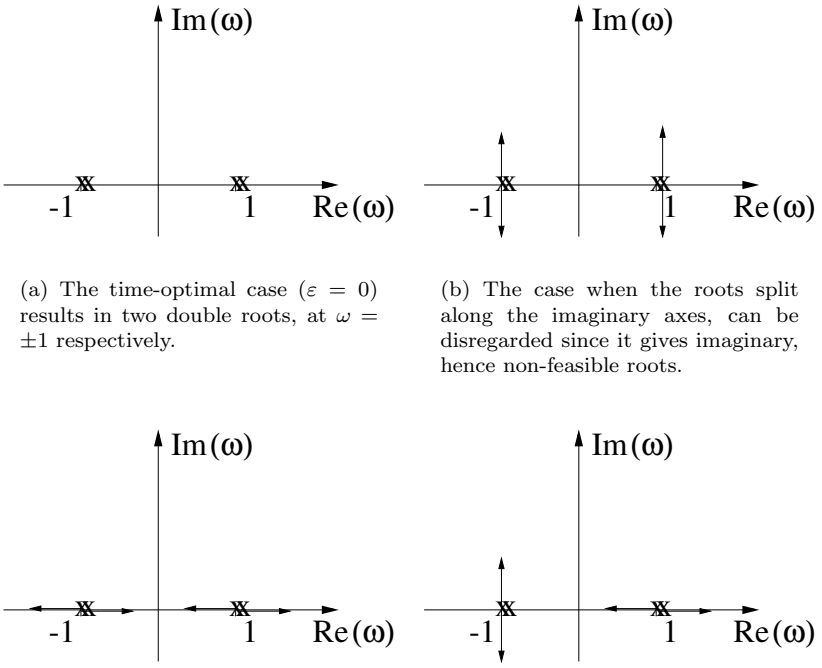
$$\frac{\partial \mathscr{H}}{\partial \omega} = \lambda_\varphi - \frac{\varepsilon \omega}{(1-\omega^2)^2} = 0 \quad (6.7)$$

where $\sigma = \lambda_x \cos\varphi + \lambda_y \sin\varphi$. By virtue of the similarities between equations (6.6) and (6.7), it suffices to study either one. Consider then the partial derivate w.r.t. the lateral velocity and initially, assume $\lambda_\varphi \neq 0$ (the $\lambda_\varphi \equiv 0$ case, will be examined later). Then, we are able to write equation (6.7) as

$$(1-\omega^2)^2 = \frac{\varepsilon}{\lambda_\varphi}\,\omega \quad (6.8)$$

In the time-optimal case, when $\varepsilon = 0$, we trivially have $(1-\omega^2)^2 = 0$, meaning that ω has two double roots in ± 1 respectively (see figure 6.5(a)). Now, what happens to the roots and how do they evolve as the design parameter ε is gradually increased? We know that the roots of ω are complex conjugated, i.e. are symmetric with respect to the real axis. Then, there exist only three possible loci for ω's roots, ω^*. Figure [6.5(b)-(d)] illustrate these three different cases. The situation in figure 6.5(b) can be disregarded directly, since it gives four imaginary, hence non-feasible roots.
In order to be able to select between the remaining two cases (i.e. one or two real roots in the restricted control interval, $\omega \in [-1,\, 1]$), we refer to figure 6.6. There, the intersection points of the functions $f_1(\omega) = (1-\omega^2)^2$ and $f_2(\omega) = \frac{\varepsilon}{\lambda_\varphi}\,\omega$ obviously correspond to the roots of equation (6.8). Hence, we conclude that we only have one real root in the admissible interval, or equivalently that the pointwise maximization of the Hamiltonian has a unique solution. In addition, we have found out that there is another real root, but outside the domain of the control and two roots with non-zero imaginary parts.

In the time-optimal case ($\varepsilon = 0$), the slope of $f_2(\omega)$ is zero, hence the two possible values for ω^* are ± 1. This coincides with the results presented in chapter 4 and 5 ($\omega^* = \text{sign}\,\lambda_\varphi$) and naturally corresponds to the minimum radius left- and right turn respectively. This holds for all non-zero values on λ_φ. But as we set $\varepsilon \neq 0$, only the quadrant of location of our only real and admissible root, is determined by sign λ_φ. Its exact value is indefinite but nevertheless, continuously varying with λ_φ. We further note that the boundary values of the lateral control, are not reached unless



(a) The time-optimal case ($\varepsilon = 0$) results in two double roots, at $\omega = \pm 1$ respectively.

(b) The case when the roots split along the imaginary axes, can be disregarded since it gives imaginary, hence non-feasible roots.

(c) If both double roots split along the real axis, we get two roots in the admissible control set $|\omega| \leq 1$.

(d) This last case, results in one admissible root.

Figure 6.5: Root loci.

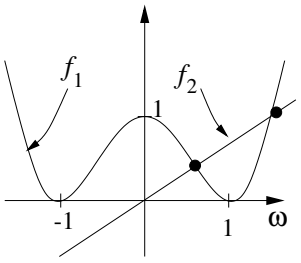


Figure 6.6: The roots of equation (6.8) correspond to the intersection points of $f_1(\omega) = (1-\omega^2)^2$ and $f_2(\omega) = \frac{\varepsilon}{\lambda_\varphi}\,\omega$.

$\lambda_\varphi \to \infty$, excluding the occurrence of bang-bang solutions. These are all desirable properties. Now, what happens as $\lambda_\varphi \to 0$ and we cross the border line instead? Will ω^* remain continuous even at this point? As $\lambda_\varphi \to 0$, the straight line $f_2(\omega)$, becomes more and more vertically aligned, meaning that $\omega^* \to 0$. To see what happens as we cross the imaginary axis, we take a look at the case when $\lambda_\varphi = 0$. Setting $\lambda_\varphi = 0$ in equation (6.7) yields $\varepsilon\,\omega = 0$ and since ε is a free variable, we get $\omega^* = 0$. Hence, we conclude that ω^* varies continuously as λ_φ evolves and changes sign.

Although it is fully possible to analytically solve equation (6.7) for ω^*, the resulting expressions looks anything but tidy. Therefore, we approximate $f_1(\omega),\ \omega \in [-1, 1]$ in figure 6.6, with the upper half of a unit circle. With this approach, expressing the lateral control as a function of ε, reduces to finding the point of intersect between a straight line and a semi-circle. Referring to figure 6.7 and considering the fact that $\tan\vartheta$ equals the direction of the straight line $\frac{\varepsilon}{\lambda_\varphi}$, we get

$$\omega^* = \cos\vartheta = \frac{1}{\sqrt{\tan^2\vartheta + 1}} = \frac{1}{\sqrt{(\frac{\varepsilon}{\lambda_\varphi})^2 + 1}} = \frac{\lambda_\varphi}{\sqrt{\varepsilon^2 + \lambda_\varphi^2}} \tag{6.9}$$

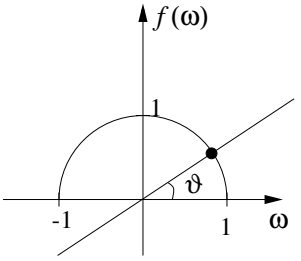


Figure 6.7: The circle segment approximates $f_1(\omega),\ \omega \in [-1, 1]$ in figure 6.6 and leads to some awarding results.

We observe that by setting $\varepsilon = 0$ in equation (6.9), we get $\omega^* = \text{sign}\ \lambda_\varphi$, which is to be recognized as the optimal control we had in the time-optimal case. Then by gradually tuning the design parameter up, we damp the fluctuating behavior of the optimal control whenever λ_φ changes sign, and thereby replace the bang-bang solutions with smoother and more pliable paths (cf. figure 6.8).

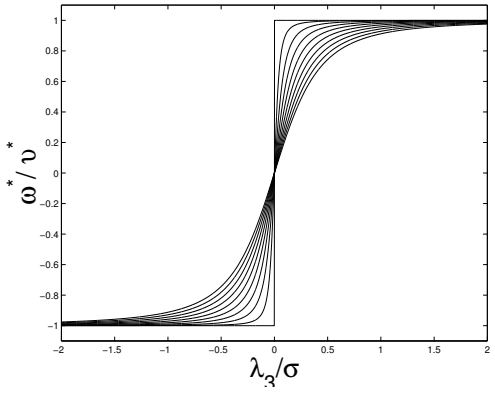


Figure 6.8: The ε-dependence of the optimal control. ε varying between 0 and 0.5.

Analogous reasoning applies for the linear control v^*, with σ playing the role of λ_φ. The summarizing results of this example will therefore be

$$\boxed{\begin{array}{rcl} \upsilon^* &=& \frac{\sigma}{\sqrt{\varepsilon^2+\sigma^2}} \\ \omega^* &=& \frac{\lambda_\varphi}{\sqrt{\varepsilon^2+\lambda_\varphi^2}} \end{array}}$$

Motivated by the perspicuous results and their desirable properties obtained in example 4, we make the following choice for Lagrangian

$$\mathcal{L} = 1 - \varepsilon\sqrt{1-u^2}, \tag{6.10}$$

its circular form, evidently influenced by the approximation made in the previous example. The Hamiltonian then becomes

$$\mathscr{H} = \lambda^T f - \mathcal{L} = \lambda_x \upsilon\cos\varphi + \lambda_y \upsilon\sin\varphi + \lambda_\varphi\,\omega + \varepsilon\sqrt{1-\upsilon^2} + \varepsilon\sqrt{1-\omega^2} - 1 \tag{6.11}$$

and its pointwise maximization gives

$$\frac{\partial\mathscr{H}}{\partial\upsilon} = \sigma - \frac{\varepsilon\upsilon}{\sqrt{1-\upsilon^2}} = 0 \implies \sqrt{1-\upsilon^2} = \frac{\varepsilon}{\sigma}\,\upsilon \tag{6.12}$$

$$\frac{\partial\mathscr{H}}{\partial\omega} = \lambda_\varphi - \frac{\varepsilon\omega}{\sqrt{1-\omega^2}} = 0 \implies \sqrt{1-\omega^2} = \frac{\varepsilon}{\lambda_\varphi}\,\omega, \tag{6.13}$$

where $\sigma = \lambda_x\cos\varphi + \lambda_y\sin\varphi$. The solutions of both (6.12) and (6.13), involve finding the point of intersection between a straight line, with a known slope ($\frac{\varepsilon}{\sigma}$ or $\frac{\varepsilon}{\lambda_\varphi}$) and the upper half of a unit circle and is best interpreted by means of figure 6.7. In accordance with the calculations made in example 4, we obtain

$$\boxed{\begin{array}{rcl} \upsilon^* &=& \frac{\sigma}{\sqrt{\varepsilon^2+\sigma^2}} \\ \omega^* &=& \frac{\lambda_\varphi}{\sqrt{\varepsilon^2+\lambda_\varphi^2}} \end{array}} \tag{6.14}$$

Reflecting upon what this choice of the Lagrangian has resulted in, we are able to pinpoint some distinguished characteristics; firstly, notice that $\mathcal{L}_0$ in this case, conceptually differs from the ones considered so far. This one merely satisfies our penalizing objective, while no solid control restricting barrier, is imposed. Nevertheless, we can conclude from the expressions for the optimal controls (equation (6.14)), that even this requirement is met. For all non-zero values on the design parameter ε, $|u^*| < 1$, which is the saturation boundary for the optimal control. This provides us with a more subtle and implicit approach for handling control constraints.
Secondly, we observe that we have the possibility to make a continuous and arbitrary adjustment of the design parameter, in order to damp the fluctuating behavior of the optimal control pertaining to time-optimal solutions. This introduced flexibility, is illustrated in figure 6.8 and enables us to designedly avoid time-optimal solutions and their drawbacks while obtaining more supple paths.
Yet another advantage is the smoothing effect of u^* on the systems Hamiltonian function H. This benefit is to be discussed more thoroughly in section 6.3.

6.2 Simulation Results

In addition to the systems dynamics that governs the time evolution of the state vector and the optimal control law specified by equation (6.14), the time evolution of the auxiliary variables will be of interest, when we carry out our simulations. The TPBVP that we ought to consider is

Two Point Boundary Value Problem (TPBVP)

$$\begin{cases} \dot{x} = \frac{\partial H}{\partial \lambda_x} = v^* \cos\varphi \\ \dot{y} = \frac{\partial H}{\partial \lambda_y} = v^* \sin\varphi \\ \dot{\varphi} = \frac{\partial H}{\partial \lambda_\varphi} = \omega^* \end{cases} \qquad \begin{cases} \dot{\lambda_x} = -\frac{\partial H}{\partial x} = 0 \\ \dot{\lambda_y} = -\frac{\partial H}{\partial y} = 0 \\ \dot{\lambda_\varphi} = -\frac{\partial H}{\partial \varphi} = v^*[\lambda_y \cos\varphi - \lambda_x \sin\varphi] \end{cases}$$
$$\text{s.t. } x(0) = X_i \text{ and } x(T) = X_f \tag{6.15}$$

where

$$\boxed{\begin{array}{lcl} v^* & = & \frac{\sigma}{\sqrt{\varepsilon^2+\sigma^2}} \\ \omega^* & = & \frac{\lambda_\varphi}{\sqrt{\varepsilon^2+\lambda_\varphi^2}} \end{array}}$$

Since the state variables are specified at both the initial and final time instant, while the auxiliary variables are unrestricted, we have a mixed boundary value problem at hand. Taking resource in the numerical methods, the so called *shooting method* is the most well-tried and dependable technique. The idea behind and the algorithm for this method, is discussed and presented in appendix A. Also, when employing numerical DE solvers in e.g. Matlab, the duration of the time interval in which the numerical integration should last, have to be prescribed. This naturally contradicts the fact that we have a free arrival time, which is to be optimized. To set this right, we utilize the time transformation described in appendix B.

Figure 6.9 and 6.10 are the outcome of our first simulation and serve as a perfect example of how choosing $\mathcal{L}$ as prescribed by (6.10) results in such expressions for optimal input that meets all our three objectives specified on page 49. The task is to steer the car between $X_i = [-2,\ 2,\ -\frac{\pi}{2}]$ and $X_f = [0,\ 0,\ -\frac{\pi}{2}]$. The time-optimal path is obtained by setting $\varepsilon = 0$ and is sketched with a thicker line in figure 6.9. However, by tuning ε up gradually, we are able to digress from this time-optimal (bang-bang) solution in a continuous and controlled manner.

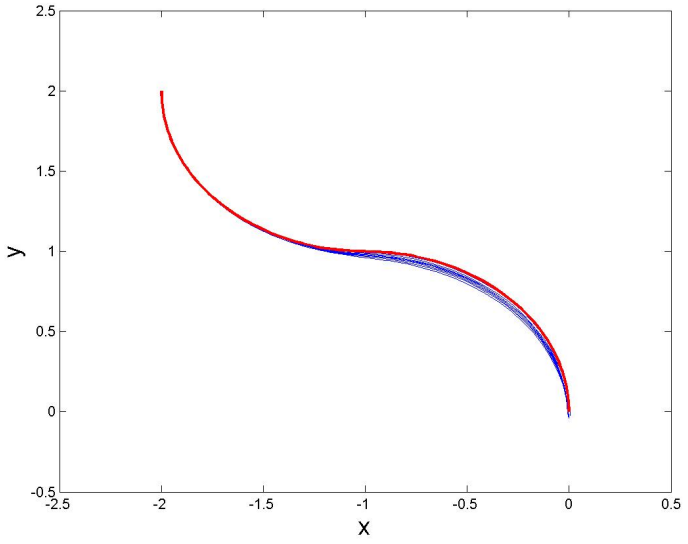


Figure 6.9: By tuning ε up gradually, we digress from the time-optimal (bang-bang) solution. In this particular case, ε varies in the interval [0.1, 0.5].

The corresponding plot of $\omega^*(t)$ in figure 6.10, shows how the fluctuating behavior of the optimal input is damped gradually for higher values on ε. This helps us to reduce

the impairment of the steering device, but also raises the robustness with respect to any possible uncertainties.

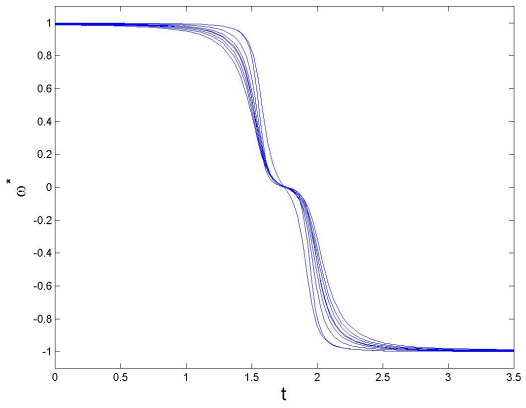


Figure 6.10: The effect of ε on the optimal input, $\omega^*(t)$.

This robustness is illustrated in greater detail in our next simulation where we wish to study the flexibility of the mobile platform with respect to a change in a *prescribed* time of arrival, $\widehat{T}$. Such flexibility is of great importance when simultaneous rendezvous problems for a team of platforms are considered. In such cases, a common time over target (TOT) is determined which both ensures simultaneous intercept and is feasible for each platform on the team. However, due to incomplete or even incorrect information about the surrounding world, some of the team members might not be able to reach the target upon the agreed time. Consequently, it is of interest to determine whether the remaining team members are able to postpone their time of arrival and agree upon a new feasible TOT. Notice that since the time of arrival is *prescribed*, the integral constant $\mathcal{L}_1 = 1$ might as well be excluded from the integral cost function, $\mathcal{L}$. It is so because this term only contributes with an additive constant in the value function J. However, for the sake of consistency, it might be motivated to include the integral constant in $\mathcal{L}$.

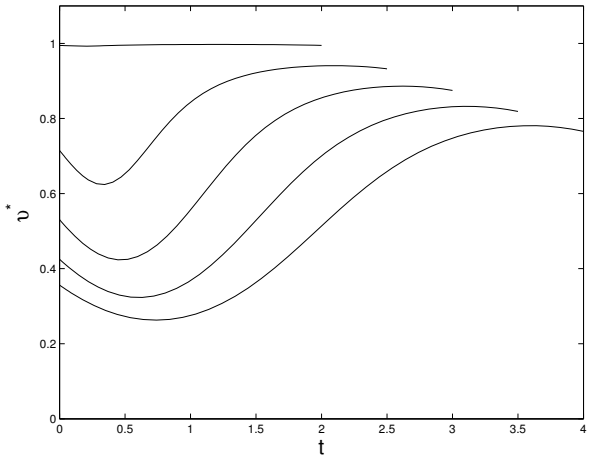


Figure 6.11: As the time of arrival sets to higher values, a more moderate driving style is being adopted. By decreasing the value of the linear velocity, the car "wastes time" and is thereby able to adjust its time of arrival within a considerable interval of time.

Referring to figure 6.11 and 6.12, the task in all five trials is to steer between the same two prescribed configurations. However, the prescribed time of arrival, $\widehat{T}$, varies. As seen in figure 6.11, when $\widehat{T}$ is set to 2, the linear velocity $v^*(t)$, almost takes its highest value, i.e. equals 1, during the entire time interval - this in order to be able to make it to the final configuration at the prescribed time of arrival. But as we set $\widehat{T}$ to higher and higher values, we note how the linear velocity decreases and a more moderate driving style is being adopted.
The corresponding control in the the lateral direction $\omega^*(t)$, can be seen in figure 6.12, where we note that for $\widehat{T} = 2$, $\omega^*(t)$ is relatively bang-bang. But as the value of the arrival time increases, the fluctuating behavior of the angular velocity is being damp, resulting in much smoother paths.

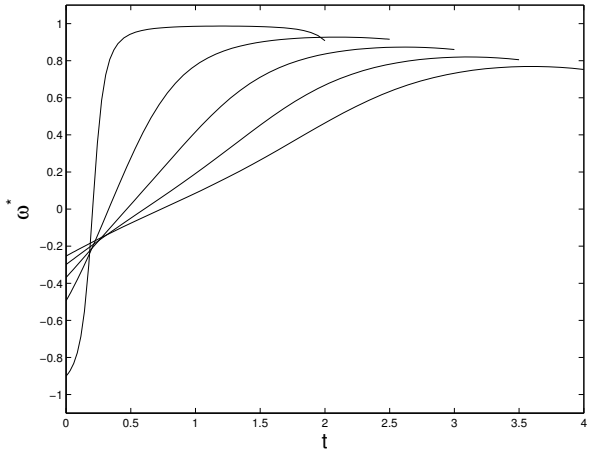


Figure 6.12: As the time of arrival is postponed, the generated paths become smoother.

These two abovementioned simulations, illustrate that the proposed control law meets all our requirements and fulfills all our objectives thus far. However, the numerical computations in the shooting method, turns out to be divergent at some instances. This is to be illustrated in our next simulation. Once again, the resistance and flexibility against a change in $\widehat{T}$ is being investigated, but this time the outcome is not as successful. Figure 6.13 shows the generated paths when the task is to steer the platform from the initial configuration $X_i = [0,\ 0,\ 0]$, to the final configuration $X_f = [1.5,\ 2,\ \frac{\pi}{2}]$, while $\widehat{T}$ varies between 3 and 10. For $\widehat{T} < 8$, the corresponding paths succeed to interconnect X_i and X_f by making a smooth left turn. But for arrival times $\widehat{T} > 8$, the generating paths fail to end up in the prescribed final configuration. This is because it is extremely hard to make a initial guess of the auxiliary variables $\lambda(0)$ such that $x(\widehat{T}) = X_f$ (see appendix A for more details). In the next section, we are to locate the origin of this undesirable behavior.

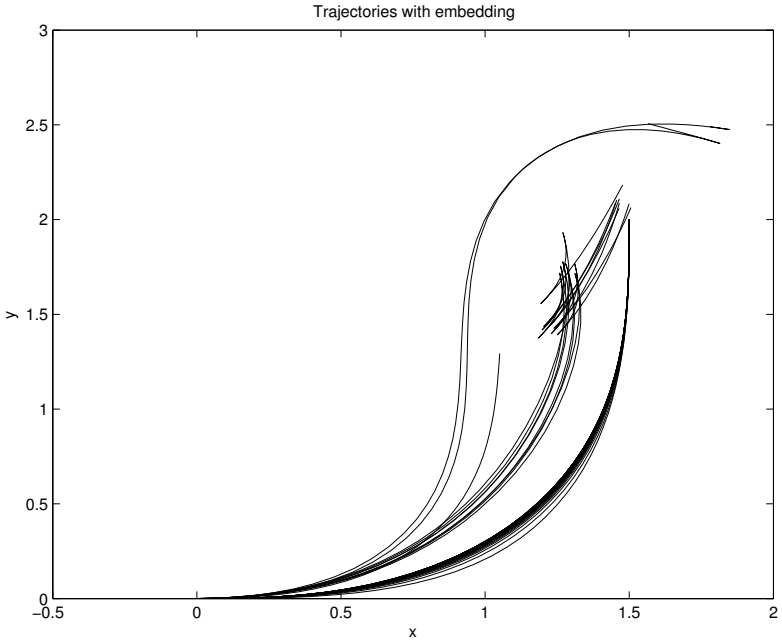


Figure 6.13: The generated paths when the prescribed time of arrival varies between 3 and 10 time units. Because of the numerical instability of the shooting method, some of the paths fail to end up in the prescribed final configuration, $X_f = (1.5,\ 2,\ \frac{\pi}{2})$.

6.3 The Singular Property of the Problem

In order to get a glimpse beneath the surface and gain some insight about the reasons for the shooting method to diverge, we recur to Dubins' problem which was the scope of chapter 4.

Initially, compiling equation (4.3) and (4.7) together with the definition of the Hamiltonian function (cf. equation (3.3)), we get

$$H(x,\lambda) = \mathscr{H}(x,\lambda,\omega^*) = \lambda_x \cos\varphi + \lambda_y \sin\varphi + |\lambda_\varphi| - 1. \tag{6.16}$$

Since neither of the state variables specifying the position of the platform (i.e. x or y) is included in H, the corresponding auxiliary variables, λ_x and λ_y, are cyclic, that is they are time constants. Then by setting

$$\begin{cases} \lambda_x = -\mu\cos\varphi_0 \\ \lambda_y = \ \ \mu\sin\varphi_0, \end{cases}$$

we are able to write equation (6.16) as

$$H = \mu\cos(\varphi - \varphi_0) + |\lambda_\varphi| - 1.$$

Now, from the PMP chapter, or more precisely, proposition 8 in conjunction with remark 6, we conclude that the Hamiltonian is constant on a full trajectory - that is, the level curves of H correspond to different trajectories for the mobile platform. Let us therefore pay attention to these. The level curves of the system are sketched by means of *Mathematica* in figure 6.14. The reason for rewriting the cyclic auxiliary variables (λ_x and λ_y) in terms of μ and φ_0 might be more obvious now, since the constant φ_0 solely adjusts the horizontal alignment of the level curves, while the constant μ specifies their depth. We are thus able to study the full motion of the mobile platform by just paying attention to the time evolution of two variables, namely $\varphi(t)$ and $\lambda_\varphi(t)$.

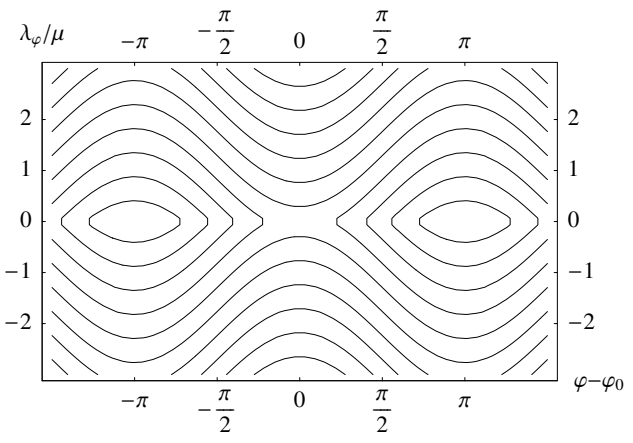


Figure 6.14: The level curves of the Hamiltonian correspond to different trajectories of the mobile platform. Generating a path for Dubins' car by choosing an appropriate $\lambda(0)$ is a singular problem.

Referring to figure 6.14, let us demonstrate the singular property of the problem by considering how the generation of a typical BSB path for Dubins' car is carried out. The orientation of the platform at the initial point, $\varphi(0)$, is known, so that we know where on the φ-axis we are at $t = 0$. We then ought to follow a level curve until we reach a switch-point, i.e. λ_φ becomes zero. This corresponds to the first circular movement. In order to generate the straight line segment in the BSB type

of path, we are then to stop at this particular point for an arbitrary amount of time, namely as long as the S segment endures. This because on a straight line segment, the orientation angle is constant while λ_φ equals zero. Following this, in a controlled manner, we are to decide which of the level curves to follow, the one that leads to higher values on φ (left turn) or the one leading in the opposite direction, resulting in a right turn. Generating such a motion by just finding an appropriate value on $\lambda_\varphi(0)$ is of course a singular problem.
Same conclusion can be drawn from studying figure 6.15 where the level surface of the Hamiltonian can be seen. We note that two smooth surfaces are seamed together at a joint, centered at the axis along $\lambda_\varphi = 0$. Hence the Hamiltonian has different derivatives on different sides of this joint, making the right hand side (the dynamics) of the TPBVP (6.15) a discontinuous function. Then it follows from standard results on differential equations, that in the case of a discontinuous dynamics, not even existence of a solution is guaranteed, even less its optimality or uniqueness. It is a widespread idea that all the information about the motion of a mobile platform lies in the initial values of the auxiliary variables $\lambda(0)$. We have however shown that this does not hold true in all cases and that a more careful analysis of the system properties must be carried out in order to be able to draw any conclusions about that matter.

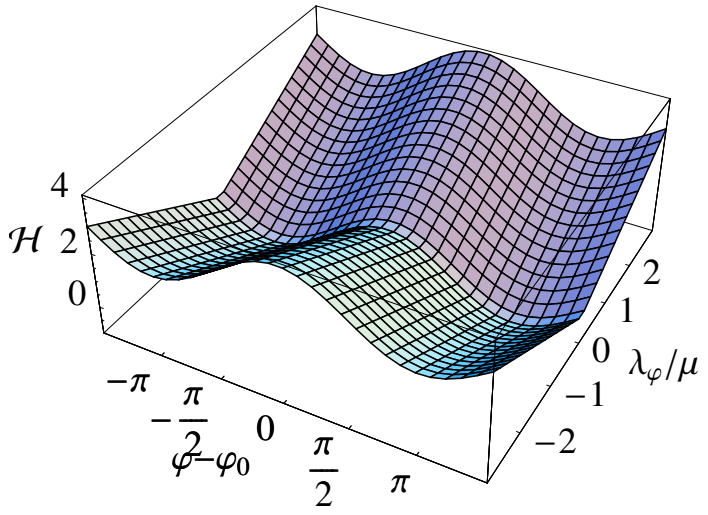


Figure 6.15: The level surface of the Hamiltonian function for Dubins' problem.

Let our next study involve the Hamiltonian of the CRS car, i.e. when we utilize the control law specified by equation (6.14). Combining this with equation (6.11) yields the following expression for the Hamiltonian function for the CRS car

$$\begin{aligned} H_{\mathrm{CRS}}(x,\lambda) &= \mathscr{H}(x,\lambda,u^*) \\ &= \frac{\sigma^2}{\sqrt{\varepsilon^2+\sigma^2}} + \frac{\lambda_\varphi^2}{\sqrt{\varepsilon^2+\lambda_\varphi^2}} + \varepsilon\sqrt{1-\frac{\sigma^2}{\varepsilon^2+\sigma^2}} + \varepsilon\sqrt{1-\frac{\lambda_\varphi^2}{\varepsilon^2+\lambda_\varphi^2}} - 1 \\ &= \sqrt{\varepsilon^2+\sigma^2} + \sqrt{\varepsilon^2+\lambda_\varphi^2} - 1. \end{aligned} \tag{6.17}$$

To fully understand the advantage with (6.17), let us present the expression for the Hamiltonian for the RS-car and make a comparison. Putting equations (5.2), (5.5) and (5.7) together, we get

$$H_{\mathrm{RS}}(x,\lambda) = \mathscr{H}(x,\lambda,u^*) = |\sigma| + |\lambda_\varphi| - 1$$

Comparing this with equation (6.17), we see that by introducing ε, we have abolished the discontinuous properties of H_{RS}, which does not have a sharp joint for $\varepsilon > 0$ and is a continuously differentiable function. Possessing a continuously differentiable Hamiltonian is noticeable, since it improves and rectifies the numerical issues of the optimal control problem. However, our ambition is to produce *nearly time-optimal* paths, hence we are concerned with rather small values on the design parameter ε. As a result, the corresponding improvements in the numerical properties of the problem at hand, will be comparatively small. In order to get a significant improvement and obtain numerically stable algorithms for real-life applications, we have to tune ε up to considerable values. That in turn, penalizes the input signals correspondingly and results in paths that differ too much from the time-optimal paths to be classified as "nearly time-optimal". Similar paths might more conveniently be generated by any "non-optimal" planning scheme, such as the cell-decomposition or visibility graph method (cf. [11]).

We have also investigated the possible approach of choosing $\mathcal{L}$ differently, so that better numerical properties are obtained for relatively low values on ε. A representative candidate for such a choice is $\mathcal{L} = \varepsilon \ln(1-u^2)+1$. Notice that since $\lim_{u\to\pm 1} \mathcal{L} = -\infty$, this choice of integral cost function, in accordance with the ones considered in examples 3 and 4, provides a solid barrier, in which the control function is allowed to take its values. The common characteristic of suitable Lagrangians for this approach, is that they put a considerable penalty on the input signal even for relatively low values on ε. The net outcome of the simulations adopting such Lagrangian is however similar to the ones aforementioned. There is obviously a trade-off between the numerical stability of the problem and the magnitude of the penalty imposed on the control function on one hand, and yet another one between the imposed penalty and the appearance of the generated paths. The bottom line is that, even in the case when we do possess a continuously differentiable H, it is *extremely* hard to find appropriate starting values for $\lambda(0)$ which take us to the prescribed final configuration, X_f. Moreover, even in the case that these values were given to us by some omnipotent and above all generous being, due to the singular property of the problem, we would most probably run into trouble, trying to apply numerical evaluations on the TPBVP.

6.4 An Alternative Approach: the Method of Perturbation

Once we had observed the difficulties we had while trying to solve TPBVP (6.15) numerically, but before we had determined its singular properties, we realized that an alternative approach for generating nearly time-optimal paths would be of great interest. With inspiration from Hamiltonian Perturbation Theory, a paradigm from the sphere of Analytical Mechanics, our objective is to study the first order contribution of introducing the design parameter ε, to the shape of the synthesized time-optimal paths. The basic idea is to introduce and recognize an explicit ε-dependence in all variables. Adopting this view involves considering the time-optimal solutions as an unperturbated case, while a change in ε (and hence in the Hamiltonian function) is regarded as a perturbation, whose time-evolution is to be determined (cf. figure 6.16). The analysis of the design parameter's influence on a system is not a trivial task. How will a small change in the design parameter ε, effect the time-evolution of the configuration- and the auxiliary variables?
Two different approaches have been considered, but because of the similarities in the outcome of them, we will herewithin only present one of these. This approach, which is the conceptually simplest one, starts off from the Taylor expansion of the time-optimal (unperturbated) solution. The objective is then to get a first order differential equation (DE) for the derivatives, which is to be solved with conventional methods, either analytically or numerically. Although it is possible to do this in a

recursive matter, for derivatives of arbitrary order, resulting in higher order approximation of the contribution of ε, it is seldom practically motivated to go beyond the second order. Hence, we will restrict ourselves to get a linear approximation of the solution, i.e. calculating the rate of change of the state- and auxiliary variables with respect to the design parameter ε (cf. figure 6.16).

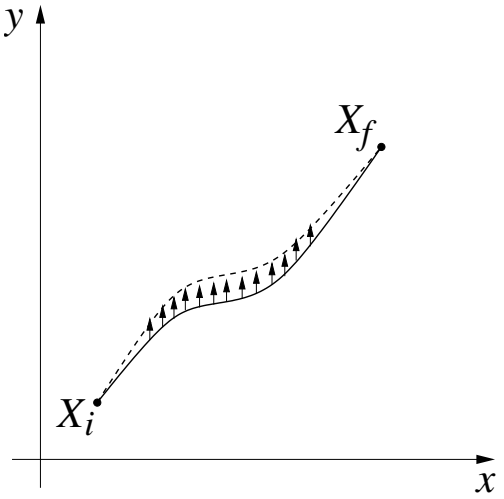


Figure 6.16: The path sketched with solid line is to represent an unperturbated solution, while the dashed line is the perturbed counterpart. A first order approximation of ε's effect on the time-optimal paths is to determine the time-evolution of $[X \;\; \Lambda]^T$, which corresponds to evaluating the value of the small arrows in the figure.

Illustrating this, if we express the time-optimal solution in terms of its Taylor expansion

$$\begin{pmatrix} x(t,\varepsilon) \\ \lambda(t,\varepsilon) \end{pmatrix} = \begin{pmatrix} x(t) \\ \lambda(t) \end{pmatrix} + \varepsilon \begin{pmatrix} X(t) \\ \Lambda(t) \end{pmatrix} + \mathscr{O}(\varepsilon^2),$$

a first order approximation of ε's effect on $[x \;\; \lambda]^T$ is to determine the time evolution of $[X \;\; \Lambda]^T$. This time evolution, serves as an indicator of the change in the shape of the unperturbated solution, originating from an infinite-small change in ε. It corresponds to evaluating the value of the small arrows in figure 6.16. Remember that what we now seek is a DE for $X(t)$ and $\Lambda(t)$.
Upon the introduction of the explicit ε-dependence, Hamilton's system of equations (3.18), becomes

$$\begin{aligned} \frac{\partial}{\partial t}x(t,\varepsilon) &= H'_\lambda(x(t,\varepsilon),\lambda(t,\varepsilon),\varepsilon) \\ \frac{\partial}{\partial t}\lambda(t,\varepsilon) &= -H'_x(x(t,\varepsilon),\lambda(t,\varepsilon),\varepsilon). \end{aligned} \tag{6.18}$$

Taking the ε-derivative of (6.18) gives us the desired (linear) DE for $X(t)$ and $\Lambda(t)$.

$$\begin{aligned} \dot{X}(t) &= H''_{\lambda x}X(t) + H''_{\lambda\lambda}\Lambda(t) + H''_{\lambda\varepsilon} \\ \dot{\Lambda}(t) &= -H''_{xx}X(t) - H''_{x\lambda}\Lambda(t) - H''_{x\varepsilon} \end{aligned} \tag{6.19}$$

Before we proceed, let us make a small digression into the world of finite dimensional systems theory, to see how to solve systems of differential equations of the type appearing in (6.19).

Introducing $\xi(t) = \begin{pmatrix} X(t) \\ \Lambda(t) \end{pmatrix}$, enables us to write (6.19) in the following more compact form

$$\dot{\xi} = A\,\xi + B, \tag{6.20}$$

with

$$A = \begin{pmatrix} H''_{\lambda x} & H''_{\lambda\lambda} \\ -H''_{xx} & -H''_{x\lambda} \end{pmatrix} \text{ and}$$
$$B = \begin{pmatrix} H''_{\lambda\varepsilon} \\ -H''_{x\varepsilon} \end{pmatrix}.$$

The general solution of such systems of differential equations is (cf. [14])

$$\xi(t) = \Phi(t, t_i)\xi(t_i) + \int_{t_i}^{t_f} \Phi(t, s)B(s)\mathrm{d}s, \tag{6.21}$$

where Φ is the so called transition matrix function. How to determine the transition matrix function can be a delicate issue. For time-invariant systems however, i.e. constant A and B[1], determining the transition matrix function becomes much simpler in that it can be expressed in the terms of the matrix exponential and becomes

$$\Phi(t, s) = e^{A(t-s)}. \tag{6.22}$$

To make the idea of this approach even more concrete, let us now apply this method to a simple optimal control problem. In order to demonstrate and bring out the full taste of it, we have designedly chosen to formulate the optimal control problem such that matters are kept as simple as possible.

Example 5 (Method of Perturbation) In the following, an illustrative optimal control problem of form (3.1) will be considered with

$$\begin{aligned} &\mathcal{L}(x, u) = \frac{x^2 + u^2}{2} + \varepsilon u \\ &\dot{x} = f(x, u) = u \\ &x(0) = 0,\ x(1) = 1. \end{aligned} \tag{6.23}$$

Applying PMP to this yields (cf. section 3.1),

$$\frac{\partial \mathscr{H}}{\partial u} = 0 \quad \Rightarrow \quad u^* = \lambda - \varepsilon,$$

resulting in

$$H(x, \lambda, \varepsilon) = \frac{\lambda^2 - x^2}{2} - \varepsilon\lambda + \frac{\varepsilon^2}{2}.$$

Here above, the necessary tools for solving a linear DE of type (6.20) are provided, let us then start off smoothly by calculating the second order partial derivatives of $H(x, \lambda, \varepsilon)$ and setting $\varepsilon = 0$ (since we the Taylor expansion is along the time-optimal trajectory). These calculations yield

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
$$B = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

[1] This result can in fact be generalized to time-varying cases for which all $A(t)$ commute.

Now, using this A and B in (6.21) and (6.22) leaves us with the complete solution of $[X \;\; \Lambda]^T$ as a function of time

$$\begin{pmatrix} X(t) \\ \Lambda(t) \end{pmatrix} = \begin{pmatrix} \cosh(t) & \sinh(t) \\ \sinh(t) & \cosh(t) \end{pmatrix} \begin{pmatrix} X(t_i) \\ \Lambda(t_i) \end{pmatrix} + \begin{pmatrix} -\sinh(t) \\ 1-\cosh(t) \end{pmatrix}. \tag{6.24}$$

This is the time function of a first order approximation for the perturbation that introducing ε in (6.23) conveys.

Returning to our main problem, we make an observation that the perturbation of the time-optimal path, $X(t)$, have to equal zero at the initial, as well as the final time instance, that is, we require $X(0)$ and $X(T)$ to vanish - this, because are to steer between two *prescribed* configurations. On the contrary, there are no constraints on $\Lambda(t)$ at corresponding time instances. Hence, what the proposed approach boils down to, is yet another TPBVP, this time in terms of X and Λ, instead of the state vector and the vector of auxiliary variables. The only remaining question to be answered is, whether this TPBVP is easier to solve than the previous one (i.e. (6.15)). In order to illuminate this issue and apply the above mentioned method to our particular problem, we recur to equation (6.17) to get the expression for the Hamiltonian function for the convexified Reeds-Shepp's car. We thus have

$$H_{\mathrm{CRS}}(x,\lambda) = \sqrt{\varepsilon^2+\sigma^2} + \sqrt{\varepsilon^2+\lambda_\varphi^2} - 1,$$

where $\sigma = \lambda_x \cos\varphi + \lambda_y \sin\varphi$. Now, to get A and B (cf. equation (6.20)), we have to calculate the second order partial derivatives of this Hamiltonian, but we might beforehand notice that since it is only ε^2 that occurs in H_{CRS}, the first order approximation of ε's effect on the optimal paths will be vanishing. Hence, we either have to make a second order approximation (i.e. calculate the time evolution of yet another term in the Taylor series expansion), or more conveniently, noticing that since ε is an *arbitrary* chosen constant, we might as well make a substitution $\varepsilon_0 = \varepsilon^2$. Now, calculating the second order partial derivatives of the Hamiltonian function and setting $\varepsilon_0 = 0$ (since we the Taylor expansion is along the time-optimal trajectory) yields

$$A = \begin{pmatrix} 0 & 0 & -\sin\varphi & 0 & 0 & 0 \\ 0 & 0 & \cos\varphi & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma & \sin\varphi & -\cos\varphi & 0 \end{pmatrix}$$

$$B = -\frac{1}{2}\begin{pmatrix} 0 & 0 & \frac{\rho}{\sigma^2} & \frac{\cos\varphi}{\sigma^2} & \frac{\sin\varphi}{\sigma^2} & \frac{1}{\lambda_\varphi^2} \end{pmatrix}^T,$$

where ρ is the time derivative of σ, i.e. $\rho = -\lambda_x \sin\varphi + \lambda_y \cos\varphi$. It should be noticed that we no longer are dealing with a time-invariant system, so that it is no longer possible to calculate the transition matrix function by means of equation (6.22). We thus have to take resource in some numerical method two solve the TPBVP at hand. However, we might predict the outcome of such an effort by noticing that along an time-optimal trajectory ($\varepsilon_0 = 0$), the last component of B obviously goes to infinity as we approach a switching point, i.e. $\lambda_\varphi \to 0$. This makes the time evolution of the DE (6.20) divergent and consequently, the proposed line of action have to be disregarded.

7. Conclusions and Future Work

Initially, the problem of finding time-optimal paths for Dubins' and Reeds-Shepp's (RS) car models is considered. Upon the presentation of the sufficient family of paths for both these models, a geometric algorithm for finding the time-optimal path for Dubins' car is provided. Although the possibility of generalizing this algorithm to include more general path combinations and thereby covering the RS-car has not been investigated in any greater detail, it does seem viable. This belief is based on the fact that all candidates for optimal solution to RS problem, have the same number of undetermined time indices as Dubins' problem, namely three. Once this extension is carried out, we ought to adopt preferred path following technique in order to follow this synthesized time-optimal path.

When considering the problem of generating nearly time-optimal paths for the CRS car model, we concluded that PMP, augmented with the shooting method, is not an adequate approach. This is due to the singular properties of the associated TP-BVP. It is a widespread belief that all the information about the motion of a mobile platform lies in the initial values of the auxiliary variables, $\lambda(0)$. We have however shown that this does not hold true in all cases and that a more careful analysis of the system properties must be carried out in order to be able to draw any conclusions about that matter.

We were however able to rectify the numerical difficulties to some extend by either significantly increasing the value of the design parameter ε, or choosing another type of Lagrangian. The common characteristic of suitable Lagrangians for this line of action, is that they put a considerable penalty on the input signal even for relatively low values on ε. There is obviously a trade-off between the numerical stability of the problem and the magnitude of the penalty imposed on the control function on one hand, and yet another one between the imposed penalty and the appearance of the generated paths. Hence, the generated paths when striving to get a numerically stable algorithm for real-life applications, are too unlike the time-optimal bang-bang solutions, to be classified as "nearly time-optimal" paths and thus have to be disregarded. Similar paths might more conveniently be generated by any "non-optimal" planning scheme, such as the cell-decomposition or visibility graph method (cf. [11]). The insignificant number of variables involved in the treacherously simple problems we have studied, turn them into perspicuous and invaluable textbook cases, from where we are able to extract a considerable amount of information while striving to put our theoretical knowledge in practice. For this reason, it might be motivated to go beyond the case of Ground Vehicles and extend the conducted study to the three-dimensional case and thus study the problem of finding time-optimal paths for Aerial Vehicles and missiles. Dubins problem in $\mathbb{R}^3$ is not as well-studied, although some rather spotty and counterintuitive results have been presented. It was for instance an open conjecture whether optimal paths in $\mathbb{R}^3$ are again concatenations of straight lines and circle segments, until this was disproved by Sussman in [23]. Dubins' problem can be generalized to several space dimensions in a number of different ways. Important contributions have been given in [15], [19] and [23].

A. The Shooting Method

We pay attention to the TPBVP (6.15). It is alternatively formulated as the problem of finding an initial condition for the auxiliary variables $\lambda(0)$, such that $A(t_f) = x(t_f) - X_f = 0$. The shooting method approach, is based on successive improvements of the unspecified boundary conditions (in our case the λ's). The iterative procedure continues until all the specified boundary conditions are met, i.e. $A(t_f) = 0$.
To apply shooting to a TPBVP, the following stepwise algorithm is used to update the initial condition for λ (see [9]):

Step 1 Make a qualified initial guess $\lambda(0) = \lambda_0$.

Step 2 Integrate the system

$$\begin{cases} \dot{x} = \quad H_\lambda & x(0) = X_i \\ \dot{\lambda} = -H_x & \lambda(0) = \lambda_0 \end{cases} \tag{A.1}$$

while choosing u such that $H_u = 0$. In our case, this corresponds to choosing v^* and ω^*, in accordance with equation (6.14).

Step 3 Compute the final error $A(t_f) = x(t_f) - X_f$.

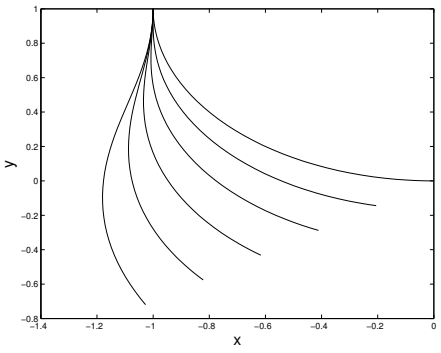
Step 4 Update the initial condition $\lambda_0 = \lambda_0 + \eta\, \Phi(0, t_f)\, A(t_f)$, where the transition matrix $\Phi(0, t_f) = \left[\frac{\partial A(t_f)}{\partial \lambda(0)}\right]^{-1}$, transfers a perturbation in $\lambda(0)$ to a perturbation in the final error. Moreover, η is a suitably chosen step-length.

Step 5 Iterate steps 2-4, until $|A(t_f)| < tol$.

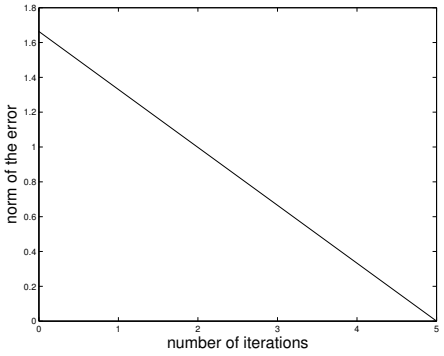
The transition matrix in step 4, can be calculated, either by means of numerical differentiation, or by linearizing the system (A.1) and the final error $A(t_f)$.

Remark 14 Generalizing this, if not all the state variables at the final time are specified, then $\lambda(t_f)$ has to fulfill a so called transversality condition (cf. section 3.2). Then the final error will, instead of being solely a function of the state vector, become a function of both the final state vector and final vector of auxiliary variables, i.e. $A(x(t_f), \lambda(t_f))$.

Although conceptually simple and quite reliable (using shooting method, satellites were launched in the 1950s), this method's main drawbacks are two folded; firstly it can be crucial, as seen in section 6.2, to find good initial estimates on the unspecified parameters. Secondly, the linearized system used in the calculation of the transition matrix $\Phi(0, t_f)$, may be gravely ill-conditioned and might result in divergent calculations (see e.g. [9]).



(a) Generated trajectories by iteratively applying shooting method five times. Here, the initial- and final configurations are $X_i = (-1, 1, -\frac{\pi}{2})$ and $X_f = (0, 0, 0)$ respectively.



(b) The corresponding error for each iteration.

Figure A.1: Illustration of the shooting method

B. Time Transformation of Optimal Control Problems

When employing numerical methods in e.g. Matlab, the duration of the time interval in which the numerical integration should dure, have to be prescribed. This naturally contradicts the fact that we have a free arrival time, which is to be optimized. To set this right, we utilize the following time transformation.

Consider the following dynamical system and its cost functional

$$\begin{aligned} &\min \textstyle\int_{t_i}^{t_f} \mathcal{L}(x(t), u(t))\,\mathrm{d}t \\ &\text{subject to} \\ &\dot{x}(t) = f(x(t), u(t)) \\ &x(t_i) = X_i \end{aligned} \tag{B.1}$$

Suppose problem (B.1) is defined on a free end-time interval $[t_i,\, t_f]$. Assume additionally, that we are able to put an (not necessarily tight) upper bound on the optimized arrival time t_f^*, i.e. find a constant $\widehat{T} \in \mathbb{R} : t_f^* \leq \widehat{T}$. Initially, define a new time variable $\tau \in [0,\, 1]$ and let the old time variable t be a linear transformation of it, that is set $t(\tau) = a\tau + t_i$ where $a \in \mathbb{R}$ is a constant. Next, introduce two new state variables

$$\begin{cases} \alpha(\tau) = t(\tau) = a\tau + t_i \;\text{ and} \\ \beta(\tau) = \frac{\mathrm{d}t(\tau)}{\mathrm{d}\tau} = a. \end{cases}$$

Then, one has the following new dynamical system defined on the *fixed* end-time interval $[0,\, 1]$:

$$\begin{cases} x^{\dagger}(\tau) = \frac{\mathrm{d}x}{\mathrm{d}t} \cdot \frac{\mathrm{d}t}{\mathrm{d}\tau} = f(x(\tau), u(\tau)) \cdot \beta(\tau) \\ \alpha^{\dagger}(\tau) = \beta(\tau) \\ \beta^{\dagger}(\tau) = 0 \end{cases}$$

where the symbol $\dagger$, refers to the operation of differentiation with respect to τ. Introduce now an extended state vector $\tilde{x}$ and a modified Lagrangian L

$$\begin{cases} \tilde{x} = [x^T(\tau), \alpha(\tau), \beta(\tau)]^T \in \mathbb{R}^{n+2} \\ L(\tilde{x}(\tau), u(\tau)) = \mathcal{L}(x(\tau), u(\tau)) \cdot \beta(\tau). \end{cases}$$

By virtue of this, we are able to re-formulate problem (B.1) as

$$\begin{aligned} &\min \textstyle\int_0^1 L(\hat{x}(\tau), u(\tau))\,\mathrm{d}t \\ &\text{subject to} \\ &\dot{\tilde{x}}(\tau) = [f(x(\tau), u(\tau)) \cdot \beta(\tau), \beta(\tau), 0]^T \\ &\tilde{x}(0) = [X_i, 0, a]^T, \end{aligned} \tag{B.2}$$

where the constant that specifies the rate of time-growth, $a \in [t_i, \widehat{T}]$. This is the alternative representation of problem (B.1), defined on the fixed end-time interval [0, 1].

Remark 15 The generalization of this time transformation, to comprise non-autonomous systems, is straightforward.

References

[1] Jérôme Barraquand and Jean-Claude Latombe. "On Nonholonomic Mobile Robots and Optimal Maneuvering". *Proceedings of IEEE International Symposium on Intelligent Control*, pages 340–347, Sep. 1989.

[2] J.D. Boissonnat, A. Cerezo, and J. Leblond. "Shortest paths of bounded curvature in the plane". In *IEEE International Conference on Robotics and Automation*, Nice, France, 1992.

[3] X.-N. Bui, J.D. Boissonnat, P. Souéres, and J.P. Laumond. "Shortest Path Synthesis for Dubins Non-holonomic Robot". *Proceedings, IEEE International Conference on Robotics and Automation*, 1:2–7, May. 1994.

[4] L.E. Dubins. "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents". *American Journal of Mathematics*, 79:497–516, 1957.

[5] M. Egerstedt, X. Hu, and A. Stotsky. "Control of Mobile Platforms Using a Virtual Vehicle Approach". *IEEE Transactions on Automatic Control*, 46(11), Nov. 2001.

[6] Magnus Egerstedt. *"Motion Planning and Control of Mobile Robots"*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2000.

[7] Petter Ögren. *"Formations and Obstacle Avoidance in Mobile Robot Control"*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2003.

[8] Alberto Isidori. *Nonlinear Control Systems, Third Edition*. Springer-Verlag, 1995.

[9] Ulf Jönsson, Claes Trygger, and Petter Ögren. "Optimal Control". Lecture notes, Optimization and Systems Theory, Royal Institute of Technology(KTH), 2003.

[10] Velimir Jurdjevic. *Geometric Control Theory*. Cambridge Univesity Press, 1997.

[11] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[12] Jean-Paul Laumond, editor. *Robot Motion Planning and Control*. LAAS-CNRS, Toulouse, France, Aug. 1997.

[13] George Leitman. *An Introduction to Optimal Control*. McGraw-Hill, Inc., 1966.

[14] Anders Lindquist and Janne Sand. "An Introduction to Mathematical System Theory". Lecture notes, Division of Optimization and Systems Theory, Royal Institute of Technology(KTH), 1996.

[15] Dirk Mittenhuber. "Dubins' problem is intrinsically three-dimensional". *European Series in Applied and Industrial Mathematics: Controle, Optimisation et Calcul de Variation*, 3:1–22, 1998.

[16] H. Nijmeijer and A.J. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, 1990.

[17] Carina Norberg Vooren. "Differential Geometric Aspects of Optimal Control". Technical report, FOI-R--0521--SE, Swedish Defence Research Agency (FOI), Jan. 2002.

[18] L.S. Pontryagin et al. *The Mathematical Theory of Optimal Processes*. New York: John Wiley & Sons, Inc., 1962.

[19] Felipe Monroy Péres. *"Non-Euclidean Dubins' problem: A control theoretic approach"*. PhD thesis, University of Toronto, 1995.

[20] J.A. Reeds and L.A. Shepp. "Optimal paths for a car that goes both forwards and backwards". *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[21] Shankar Sastry. *Nonlinear Systems: Analysis, Stability and Control*. Springer-Verlag, 1999.

[22] Philippe Souéres and Jean-Paul Laumond. "Shortest Path Synthesis for a Car-Like Robot". *IEEE Transactions on Automatic Control*, 41(5):672–688, May. 1996.

[23] Héctor J. Sussman. "Shortest 3-dimensional paths with a prescribed curvature bound.". *Proceedings of the 1995 IEEE Conference on Decision and Control*, pages 3306–3312, 1995.

[24] Héctor J. Sussman and Guoqing Tang. "Shortest paths for the Reeds-Shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control". Technical Report Report SYCON-91-10, Department of Mathematics, Rutgers University, 1991.