UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXIV Ciclo

Optimal motion planning
of
wheeled mobile robots

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Aurelio Piazzi

Dottorando: *Gabriele Lini*

January 2012

Alla mia famiglia,
per il costante
sostegno.

Contents

List of Figures

List of Tables

Introduction

This thesis presents some results obtained during my PhD course Dottorato in Tecnologie dell'Informazione, at the Università di Parma, Dipartimento di Ingegneria dell'Informazione, in the three years period 2009-2012. The work has been focused on the problem of the time-optimal motion control of wheeled autonomous systems, such as unicycle robots, automatic guided vehicles (AGVs), car-like vehicles and truck and trailer (or one-trailer) systems.

The aim is to obtain a control that provides a *smooth* motion of the unmanned vehicle in minimum-time. In order to do that, it is necessary to plan a path with an appropriate geometric continuity, and two time-optimal input signals of velocity and steering angle continuous with their derivatives. Moreover, a feedback controller must be adopted to guarantee the robustness of the overall control scheme. Final result of the thesis can be viewed as the synthesis of various methods for hybrid feedforward/feedback control for a wide class of wheeled mobile robots. Figure 1 presents a conceptual scheme that summarizes the idea behind the hybrid feedforward/feedback control, which is the final result of the work done along the three years of study and research.

Path planning and velocity planning can be completely independent to each other, on condition that:

1) the planned path has an appropriate geometric continuity and satisfies geometric interpolating conditions at the path endpoints, and

2) the velocity is a C^1-function satisfying interpolating conditions (on distance, velocity and accelerations) at the endpoints of the planned time-interval.

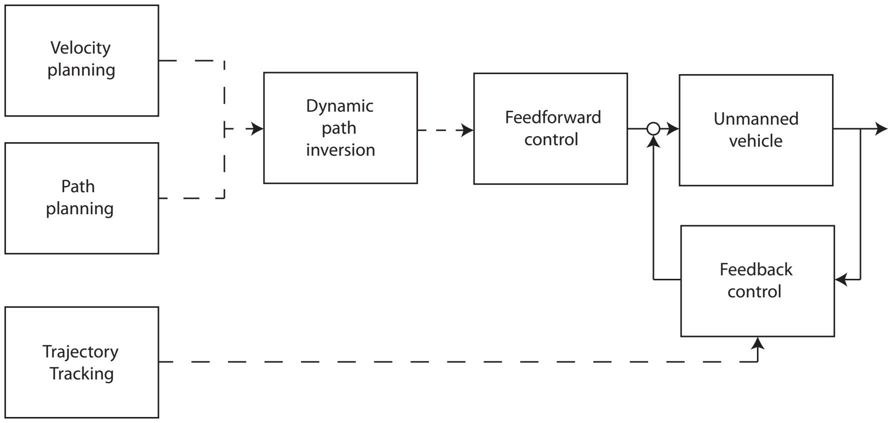


Figure 1: The overall architecture for the optimal motion control of the wheeled vehicle.

Indeed, given a sufficiently smooth path, a dynamic inversion procedure can be applied to determine the feedforward control inputs of the autonomous vehicle still maintaining freedom in the planning of velocity input.

Hence, the thesis first shows some methods that permit to plan a path and optimal input signals which lead to a minimum-time smooth motion for a variety of automatic guided systems in nominal conditions (i.e. no noise affects the systems). Secondly, it is shown how guarantee the tracking of the planned trajectory by means of a feedback control, when the system is affected by additive noise.

The very first part of the thesis (chapter 1) faces the time-optimal velocity planning with arbitrary boundary conditions for an automatic guided vehicle. Initially, only a constraint on the maximum value of the jerk (i.e. the velocity second derivative) is considered. The addressed minimum-time planning problem has been recast into an input-constrained minimum-time reachability control problem with respect to a suitable state-space system, where the control input is actually the sought jerk of the velocity planning. By virtue of the well-known Pontryagin's Maximum Principle the optimal input-constrained control is then a bang-bang function. An algebraic approach to obtain this optimal solution has been devised and a new algorithm to compute the bang-bang jerk profile is exposed. This problem has been reconsidered introducing constraints also on the maximum values of the velocity and acceleration. In this case the Pontryagin's Maximum Principle does not ensure the existence of the time-optimal control. Sufficient conditions, guaranteeing the existence of a solution to the minimum-time constrained planning problem, are exposed. The time-optimal control is not a classic bang-bang function, but it shall be a generalized bang-bang. The problem has been faced through discretization and the obtained solution is based on a sequence of linear programming feasibility checks, depending on motion constraints and boundary conditions.

Chapter 2 presents two methods for the path planning of car-like and one trailer vehicles. It is shown how plan paths with an appropriate geometric continuity by resolving a geometric interpolation. In particular, the geometric interpolation problem, which has infinite dimension, has been recast into a polynomial interpolation problem (a finite dimension problem), by means of the $\boldsymbol{\eta}$-splines. The shaping of this kind of spline depends on a vector of parameters called "eta", and on the boundary conditions. It is then presented a multi-optimization process to optimally choose these free parameters, with the aim to plan trajectory that respect bounds on curvature and curvature derivative, ensuring avoidance of the obstacles in the "real" workspace. In the case of the car-like vehicle, applications to the autonomous parking problem are presented.

In chapter 3, the dynamic path inversion block (cf. figure 1) is outlined by introducing a procedure that permits to obtain a minimum-time steering control input for an automatic guided vehicle (AGV). One can consider to have just planned a path and a time-optimal velocity profile exploiting the techniques introduced in the first two chapter. The optimal steering input signal for the AGV is obtained with a dynamic inversion on the planned path, based on some geometric properties of the path itself, and of the AGV kinematic system. Similar procedure can be easily determined for the other vehicles, such as the car-like and the one-trailer.

Finally, chapter 4 proposes two methods for the trajectory tracking for autonomous systems affected by additive noise. Both methods are thought for cases where continuous-time or high-frequency revelation of the system state or output is not possible or not economical and only low-frequency feedback is practicable. The implemented solutions to this trajectory tracking problem, relies on iterative replanning methods to compute a new reference trajectory, used to generate the feedforward inverse command velocities that help in reducing the tracking errors. For both techniques explicit closed-form bounds on the tracking error are provided.

Chapter 1

Minimum-time velocity planning

Plans are only good intentions unless
they immediately degenerate into hard work

– Peter Drucker

In the wide field of vehicles autonomous navigation, significant research efforts have been dedicated to the problem of optimal motion planning. The problem of motion planning for autonomous guided vehicles is a well known and studied issue in robotics, see for example the recent books [1] and [2]. This chapter propose techniques for minimum-time velocity planning with arbitrary boundary conditions, considering two different cases: one with only constraint on the maximum absolute value of the jerk (i.e the velocity second derivative), and one with constraints also on the maximum absolute value of the acceleration and velocity. The minimum-time velocity planning is cast in the context of the so-called path-velocity decomposition [3] using the iterative steering navigation technique [4, 5].

The first two sections briefly introduce the optimal control theory, with particular attention to the linear time-optimal problem. For more details on this arguments see, for example, books [6, 7].

The third section presents a procedure for the synthesis of a velocity C^1-function that permits in minimum-time and with a bounded jerk to interpolate given velocity and acceleration at the time planning interval endpoints and to travel a given distance. The condition on the maximum jerk value permits to obtain a smooth velocity profile [8]. The addressed minimum-time planning problem will be recast into an input-constrained minimum-time reachability control problem with respect to a suitable state-space system, where the control input is actually the sought jerk of the velocity planning. By virtue of the well-known Pontryagin's Maximum Principle the optimal input-constrained control is then a *bang-bang* function.

Finally, a solution for the constrained minimum-time velocity planning is presented. In this case, the time-optimal solution is not a classic bang-bang function, but it shall be a *generalized* bang-bang function [9]. The minimum-time transition is obtained by discretizing the continuous-time model and formulating an equivalent discrete-time optimization problem solved by means of linear programming techniques. More precisely, boundary conditions and problem constraints are expressed by linear inequalities on a column vector $\mathbf{u}$, representing the input signal (i.e the jerk) at sampling times. Hence, the minimum-time planning problem is reformulated as a feasibility test for a linear programming problem, and the minimum number of steps required to complete the given transition can be found through a simple bisection algorithm. The use of linear programming techniques for solving minimum-time problems for linear discrete-time systems subject to bounded inputs dates back to Zadeh [10]. Subsequently, many contributions have appeared focusing on various improvements. For example a faster algorithm is proposed in [11]. For what concerns time-optimal control for continuous-time systems, a related result, under different hypotheses, is presented in [12].

1.1 Optimal control theory

Optimal control is the process of determining control and state trajectories for a dynamic system over a period of time, in order to minimize a performance index. The method is closely related in its origins to the theory of *calculus of variations* and it is largely due to the work of Richard Bellman [13], and Lev Pontryagin *et al.* [14]. Optimal control and its ramifications have found applications in many different fields, including aerospace, process control, robotics, bioengineering, economics, and it continues to be an active research area within control theory.

1.1.1 Problem statement and notation

Consider optimal problems defined by the *constraint set* C, a subset of the tangent bundle of a smooth manifold M, and a *cost function* f, that is a real-valued function having C as its domain. A *trajectory of* C is an absolutely continuous curve $x(t) \in M$ such that $\frac{dx}{dt}(t) \in C$ for almost all t in the domain of x. The total cost of x is defined as

$$\int_0^T f\left(\frac{dx}{dt}(t)\right) dt\,,$$

where $[0, T]$ denotes the domain of x. Given any two points x_0 and x_f in M, the *optimal trajectory* of C is the one which connects x_0 to x_f and whose total cost is minimal among all such trajectories of C.

The considered sets C admit sections of the form $\xi = F(\pi(\xi), u_1, \dots, u_m)$, where $(u_1, \dots, u_m)$ takes values in a fixed set $U \in \mathbb{R}^m$, π indicates the natural projection from TM onto M, and ξ is an arbitrary point of C. Then, the trajectory velocity $\frac{dx}{dt}$ is parametrized by the controls $u_1, \dots, u_m$, and its total cost can be expressed as

$$\int_0^T c(x(t), u(t))dt = \int_0^T f \circ F(x(t), u(t))dt\,.$$

In a given section of C, the trajectories of C that connects two given points x_0 and x_f in a finite time T, coincide with the solution curves $x(t)$ of the differential system

$$\begin{cases} \frac{dx}{dt} = F(x(t), u(t), \dots, u_m(t)) \\ x(0) = x_0 \\ x(T) = x_f \, . \end{cases}$$

Under suitable smoothness assumptions on F, each control function $u(t)$ determine a unique solution curve, so the problem of finding the optimal trajectories of C is converted to one of finding the controls that give rise to the optimal trajectory and that is an *optimal control problem.*

We shall need additional notation. For any matrix C, C' indicates its transpose, while $span(C)$ represents the set of all the eigenvalues of C. For any vector space E, its dual is denoted by $E^*.E$ can be regarded as a subspace of $(E^*)^*$ through the correspondence $e \to g(e)$ for any $e \in E$ and $g \in E^*$. When E is finite-dimensional, $E = (E^*)^*$. Recall that a linear mapping $L : E \to E^*$ is said to be symmetric if L is equal to its dual mapping L^*.

1.2 Linear time-optimal problem

The process of transferring one state into another along a trajectory of a given differential system such that the time of transfer is minimal is known as the *minimal-time problem*, and it is one of the basic concerns of optimal control theory. Consider the linear time-invariant system,

$$\frac{dx}{dt} = Ax + Bu \, , \tag{1.1}$$

with $x \in M \subset \mathbb{R}^n$ and $u \in U_c \subset \mathbb{R}^m$, where A and B are constant matrices of order $n \times n$ and $n \times m$ respectively. Let system (1.1) be defined in a real, finite-dimensional vector space M in which the control functions are restricted to a compact and convex neighborhood U_c of the origin, in a finite-dimensional control space U, and also assume that (1.1) is controllable and that control functions are measurable. A trajectory is defined by the pair (x, u), in which x is an absolutely continuous curve of some time interval $[0, T]$, $T > 0$, that satisfies (1.1) almost everywhere in $[0, T]$.

Definition 1 *A trajectory (x, u) is called time-optimal on an interval $[0, T]$ if for any other trajectory (y, v) of (1.1) defined on its interval $[0, S]$, for which $y(0) = x(0)$ and $y(S) = x(T)$, S is larger than or equal to T.*

Theorem 1 *For any time-optimal trajectory (x, u) on an interval $[0, T]$*

a) *the terminal point $x(T)$ belongs to the boundary $\partial\mathcal{A}(x(0), T)$ of the set of reachable points from $x(0)$ at $t = T$ of system (1.1);*

b) *any point b that belongs to the boundary of the set reachable from the origin at time T is the terminal point of a time-optimal trajectory on the interval $[0, T]$.*

Proof. If $x(T)$ belonged to the interior of $\mathcal{A}(x(0), T)$, then $x(T)$ would also belong to the interior of $\mathcal{A}(x(0), T - \epsilon)$, for some $\epsilon > 0$, which is not possible, because that would violate the time optimality of (x, u) on the time-interval $[0, T]$. This argument proves part a).

To prove b), note that for any $T > 0$, points on the boundary of $\mathcal{A}(0, T)$ cannot be reached in a time shorter than T. On the other hand $\mathcal{A}(0, T)$ is compact for each $T > 0$. Therefor, for each b on $\partial\mathcal{A}(0, T)$ there exists a trajectory (x, u) defined on the time-interval $[0, T]$ such that $x(0) = 0$ and $x(T) = b$. It follows by the foregoing argument that (x, u) is time-optimal on $[0, T]$. □

1.2.1 The maximum principle

For the minimum-time control problems, the Pontryagin maximum principle provides the necessary and the sufficient conditions for optimality. The reader is recommended to consult [6, pp. 305–306] for the proof of the theorem and other details.

Theorem 2 (Pontryagin's Maximum Principle) *Any time-optimal trajectory $(\bar{x}, \bar{u})$ on an interval $[0, T]$ is the projection of an integral curve $(\bar{x}, \bar{p}, \bar{u})$ of the Hamiltonian vector field $\vec{H}$ associated with $H(x, p, u) = -p_0 + p(Ax + Bu)$, with p_0 equal to either 0 or 1, such that*

a) $H(\bar{x}(t), \bar{p}(t), \bar{u}(t)) = \max_{u \in U_c} H(\bar{x}(t), \bar{p}(t), u)$ *for almost all* t *in* $[0, T]$*;*

b) $H(\bar{x}(t), \bar{p}(t), \bar{u}(t)) = 0$ *almost everywhere in* $[0, T]$*;*

c) $\bar{p}(t) \neq 0$ *for any* t*, if* $p_0 = 0$.

Proof. See [6, pp. 305–306]. □

Remark The following remarks are helpful for clarify some important aspects and consequences of the maximum principle:

1. H should be regarded as a function on $T^*M = M \times M^*$ parametrized by both the choice of a control function and the value of p_0.

2. Assume that $u(t)$ is a given measurable control function with values in U_c. Each integral curve $\sigma(t) = (x(t), p(t))$ of the Hamiltonian vector field $\vec{H}$ associated with $H(x, p, u(t)) = -p_0 + p(Ax + Bu(t))$, when expressed in canonical coordinates, satisfies the following pair of differential equations:
$$\frac{dx}{dt} = Ax(t) + Bu(t)\,, \qquad \frac{dp}{dt} = -A^*p(t)\,.$$

3. The maximality condition a) of theorem 2 is equivalent to $\bar{p}(t)B\bar{u}(t) = \max_{u \in U_c} \bar{p}(t)Bu$ for almost all t in $[0, T]$.

1.2.2 Bang-bang principle for scalar systems

The *bang-bang* principle says that the optimal controls take the most advantage of possible control action at each instant. The name is motivated by the particular case of a control space given by the interval $U_c = [u^-, u^+]$, where optimal controls must switch between the minimal and maximal values u^- and u^+. There are various theorems that make this principle rigorous. Here, the simplest one is reported, as Sontag stated in [7, pp. 436–437].

Theorem 3 (Weak bang-bang) *Assume that the matrix pair* (A, B) *is controllable. Let* $\bar{u}$ *be a control steering system* (1.1) *from an initial state* x_0 *to a final state* x_f *in minimal time* $T > 0$*. Then,* $\bar{u} \in \partial U_c$ *for almost* t *in* $[0, T]$.

Proof. The proof directly derives from the application of the Pontryagin's maximum principle (see [7, pp. 436–437]). □

Thanks to theorem 2 it is possible to state that the time-optimal control $\bar{u}$ is unique and it is also possible determine its structure (for a more rigorous treatment see [7] and [15]).

We specialize now to single input systems ($m = 1$), and write b instead of B in (1.1). In general $U_c = [u^-, u^+]$, but we will take, in order to simplify the exposition, $u^- = -1$ and $u^+ = 1$. Assume that the pair (A, b) is controllable. For each two states x_0 and x_f, there is a unique time-optimal control $\bar{u}$ steering x_0 to x_f, and there is a nonzero vector $\gamma \in \mathbb{R}^n$ such that

$$\bar{u}(t) = \mathrm{sgn}(\gamma' e^{-tA} b)\,, \tag{1.2}$$

for all $t \notin S_{\gamma,T}$, where

$$S_{\gamma,T} = \left\{t \in [0,T] :\; \gamma' e^{-tA} b = 0\right\}\,,$$

is a finite set. This means that the optimal control $\bar{u}$ is a piecewise constant function, which switches between values -1 and 1. The following proposition permits to determine the number of switchings in the case of system matrix A has only real eigenvalues.

Proposition 1 *Suppose that the matrix A has only n real eigenvalues,i.e.*

$$span(A) \in \mathbb{R}\,.$$

Then, for each γ, b and T, $S_{\gamma,T}$ as at most $n-1$ elements, whereby any time-optimal control for system (1.1) *as no more than $n-1$ switchings.*

Proof. This proposition derives directly from the application of the Pontryagin's maximum principle to the time-optimal control of a scalar system. Reader can find several proofs of this proposition (see, for example [7, 15]).

1.3 Minimum-time velocity planning with arbitrary boundary conditions

This section introduces and explains the approach presented in [16], which solves the minimum-time velocity planning problem with arbitrary boundary conditions and a constraint on the maximum jerk value. The obtained optimal-time solution, based on Pontryagin's Maximum Principle, is a smooth planning with continuous velocities and accelerations. The devised algebraic algorithm to solve this minimum-time planning problem is well suited to be implemented within the framework of path-velocity decomposition for autonomous navigation.

1.3.1 Problem statement and the structure of the optimal solution

The following definition will be used along this paper.

Definition 2 *A function $f : \mathbb{R} \to \mathbb{R}$, $t \to f(t)$ has a PC^2 continuity, and we write $f(t) \in PC^2$ if*

a) $f(t) \in C^1(\mathbb{R})$,

b) $f(t) \in C^2(\mathbb{R} - \{t_1, t_2, \dots\})$,

c) $\exists \lim_{t \to t_i^-} D^2 f(t)$, $\exists \lim_{t \to t_i^+} D^2 f(t)$, $i = 1, 2, \dots$

where $\{t_1, t_2, \dots\}$ is a set of discontinuity instants.

The problem is to plan a minimum-time smooth velocity profile $v(t) \in PC^2$ while a given constraint on the maximum jerk value j_M is guaranteed and the initial and final conditions on the velocity and acceleration are arbitrarily assigned. Formally:

$$\min_{v \in PC^2} t_f \,, \tag{1.3}$$

such that

$$\int_0^{t_f} v(\xi) d\xi = s_f \,, \tag{1.4}$$

$$v(0) = v_0 \ , \quad v(t_f) = v_f \ , \tag{1.5}$$

$$\dot{v}(0) = a_0 \ , \quad \dot{v}(t_f) = a_f \ , \tag{1.6}$$

$$|\ddot{v}(t)| \leq j_M \ , \quad \forall t \in [0, t_f] \ , \tag{1.7}$$

where $s_f > 0$, $j_M > 0$ and v_0, v_f, a_0, $a_f \in \mathbb{R}$ are arbitrary velocity and acceleration boundary conditions. s_f is the total length of the path and t_f is the travelling time to complete this path. The solution of the above problem is $\bar{v}(t) \in PC^2$ with associated minimum-time $\bar{t}_f$.

The minimum-time planning problem (1.3)-(1.7) can be easily recast to an input-constrained minimum-time control problem with respect to a suitable state-space system. Indeed consider the jerk $\ddot{v}(t)$ as the control input $u(t)$ of a cascade of three integrators as depicted in figure 1.1.

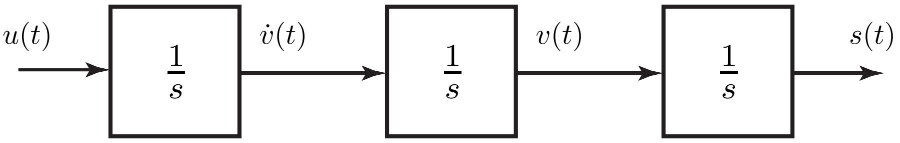


Figure 1.1: The system model for velocity planning.

Introducing the state $\mathbf{x}(t)$ as the column vector

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} := \begin{bmatrix} s(t) \\ v(t) \\ \dot{v}(t) \end{bmatrix} ,$$

the system is represented by the differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,u(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t) \ . \tag{1.8}$$

Hence, problem (1.3)-(1.7) is equivalent to find a time-optimal control $\bar{u}(t)$ that brings system (1.8) from the initial state $\mathbf{x}(0) = [0\, v_0\, a_0]'$ to the final state

$\mathbf{x}(\bar{t}_f) = [s_f \; v_f \; a_f]'$ in minimum time $\bar{t}_f$, while satisfying the input constraint

$$|\bar{u}(t)| \leq j_M \, , \quad \forall t \in [0, \bar{t}_f] \; .$$

In sections 1.2.1 and 1.2.2 it has been exposed that the Pontryagin's maximum principle gives a necessary and sufficient condition for this class of problems. Moreover, it has been shown that in the case of a linear scalar system the time-optimal control $\bar{u}(t)$ is a bang-bang function. In our case it will be a piecewise constant function that switches between the $-j_M$ and $+j_M$. Finally, another information on the optimal control structure is obtained from proposition 1. Considering that system (1.8) has three null eigenvalues we deduce, by virtue of proposition 1, that the time-optimal jerk $\bar{u}(t)$ has at most two switching instants. Hence, the general structure of the optimal $\bar{u}(t)$ is depicted in figure 1.2 where $u_M \in \{-j_M, +j_M\}$ and $0 \leq t_1 \leq t_2 \leq \bar{t}_f$ with $\bar{t}_f > 0$.

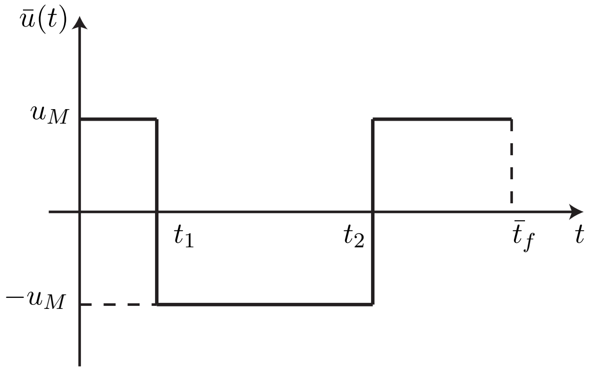


Figure 1.2: An example of the minimum-time control (jerk) profile.

1.3.2 The algebraic solution

It has been shown above the structure of the time-optimal control $\bar{u}(t)$. In the following, an algebraic approach will be exposed to exactly determine this optimal jerk profile.

Exploiting the boundary conditions (1.3)-(1.6), the problem is to find the switching time values t_1 and t_2, the minimum time $\bar{t}_f$ and the sign of the jerk initial value $\bar{u}(0)$, while satisfying the constraint $0 \le t_1 \le t_2 \le \bar{t}_f \quad \text{with} \quad \bar{t}_f > 0$. From the boundary condition (1.6) on the final acceleration value we know that

$$a_0 + \int_0^{\bar{t}_f} \bar{u}(\xi) d\xi = a_f \, .$$

Integrating the optimal jerk profile on the three intervals, the following relation is obtained

$$a_0 + \int_0^{t_1} u_M d\xi + \int_{t_1}^{t_2} (-u_M) d\xi + \int_{t_2}^{\bar{t}_f} u_M d\xi = a_f \, ,$$

and finally a first linear equation in t_1, t_2 and $\bar{t}_f$ is found

$$2\, u_M\, t_1 - 2\, u_M\, t_2 + u_M\, \bar{t}_f = a_f - a_0 \, . \tag{1.9}$$

The acceleration profile $x_3(t)$ is obtained by integrating the optimal jerk according to

$$x_3(t) = a_0 + \int_0^{t} \bar{u}(\xi) d\xi \, , \quad \forall t \in [0, \bar{t}_f] \, ,$$

that results in the following equation

$$x_3(t) = \begin{cases} a_0 + u_M\, t & t \in [0, t_1] \\ a_0 + 2\, u_M\, t_1 - u_M\, t & t \in [t_1, t_2] \\ a_0 + 2\, u_M\, t_1 - 2\, u_M\, t_2 + u_M\, t & t \in [t_2, \bar{t}_f] \, . \end{cases} \tag{1.10}$$

Now, by virtue of the boundary condition (1.5), the following relation is deduced

$$v_0 + \int_0^{\bar{t}_f} x_3(\xi) d\xi = v_f \, ,$$

hence, from (1.10), one obtains

$$v_0 + \int_0^{t_1} (a_0 + u_M\, \xi) d\xi + \int_{t_1}^{t_2} (a_0 + 2\, u_M\, t_1 - u_M\, \xi) d\xi \\ + \int_{t_2}^{\bar{t}_f} (a_0 + 2\, u_M\, t_1 - 2\, u_M\, t_2 + u_M\, \xi) d\xi = v_f \, .$$

Finally, a quadratic equation in t_1, t_2 and $\bar{t}_f$ is found

$$-u_M\, t_1^2 + 2\, u_M\, t_1\, \bar{t}_f + u_M\, t_2^2 - 2\, u_M\, t_2\, \bar{t}_f + \frac{1}{2}\, u_M\, \bar{t}_f^2 + a_0\, \bar{t}_f = v_f - v_0\,. \quad (1.11)$$

Integrating the acceleration function $x_3(t)$ as follows

$$x_2(t) = v_0 + \int_0^t x_3(\xi) d\xi\,, \quad \forall t \in [0, \bar{t}_f]\,,$$

the velocity profile $x_2(t)$ is obtained

$$x_2(t) = \begin{cases} v_0 + a_0\, t + \frac{1}{2}\, u_M\, t^2 & t \in [0, t_1] \\ \\ v_0 + a_0\, t + 2\, u_M\, t_1\, t - u_M\, t_1^2 - \frac{1}{2}\, u_M\, t^2 & t \in [t_1, t_2] \\ \\ \frac{1}{2}\, u_M\, t^2 - u_M\, t_1^2 + u_M\, t_2^2 + 2\, u_M\, t_1\, t & \\ -2\, u_M\, t_2\, t + a_0\, t + v_0 & t \in [t_2, \bar{t}_f]\,. \end{cases} \quad (1.12)$$

By virtue of the boundary condition (1.4), the following relation holds

$$\int_0^{\bar{t}_f} x_2(\xi) d\xi = s_f\,,$$

then, from (1.12), we deduce

$$\begin{aligned} &\int_0^{t_1} (v_0 + a_0\, \xi + \frac{1}{2}\, u_M\, \xi^2) d\xi + \int_{t_1}^{t_2} (v_0 + a_0\, \xi + 2\, u_M\, t_1\, \xi - u_M\, t_1^2 \\ &\quad -\frac{1}{2}\, u_M\, \xi^2) d\xi + \int_{t_2}^{\bar{t}_f} (\frac{1}{2}\, u_M\, \xi^2 - u_M\, t_1^2 + u_M\, t_2^2 + 2\, u_M\, t_1\, \xi \\ &\quad\quad -2\, u_M\, t_2\, \xi + a_0\, \xi + v_0) d\xi = s_f\,. \end{aligned}$$

Finally, the last cubic equation in t_1, t_2 and $\bar{t}_f$ is given by

$$\begin{aligned} &\frac{1}{3}\, u_M\, t_1^3 - u_M\, t_1^2\, \bar{t}_f + u_M\, t_1\, \bar{t}_f^2 - \frac{1}{3}\, u_M\, t_2^3 + u_M\, t_2^2\, \bar{t}_f - u_M\, t_2\, \bar{t}_f^2 \\ &\quad + \frac{1}{6}\, u_M\, \bar{t}_f^3 + \frac{1}{2}\, a_0\, \bar{t}_f^2 + v_0\, \bar{t}_f = s_f\,. \end{aligned} \quad (1.13)$$

The time-optimal velocity profile is planned by solving the nonlinear algebraic system given by equations (1.9), (1.11) and (1.13).

Here, we consider the case of positive initial jerk (i.e. $u_M = +j_M$). From equation (1.9) follows

$$t_1 = t_2 - \frac{1}{2}\,\bar{t}_f^2 + \frac{1}{2}\,\frac{a_f - a_0}{j_M}\,. \tag{1.14}$$

By substituting relation (1.14) in (1.11) the relation below holds

$$t_2 = \frac{\left[\frac{3}{4}\,j_M\,\bar{t}_f^2 - \frac{1}{2}\,(3\,a_f - a_0)\,\bar{t}_f + \frac{1}{4\,j_M}\,(a_f - a_0)^2 + v_f - v_0\right]}{j_M\,\bar{t}_f - a_f + a_0}\,. \tag{1.15}$$

By substitution of (1.14) and (1.15) in (1.13), a quartic equation in $\bar{t}_f$ unknown is obtained

$$\begin{aligned}
&\frac{1}{32}\,u_M^2\,t_3^4 + \frac{1}{8}\,u_M\,(a_0 - a_f)\,t_3^3 + \left(\frac{1}{2}\,u_M\,(v_0 + v_f) - \frac{1}{16}\,(a_0^2 + a_f^2) - \frac{3}{8}\,a_0\,a_f\right)t_3^2 \\
&+ \left(\frac{1}{8}\,\frac{a_0\,a_f}{u_M}(a_0 - a_f) - \frac{1}{24}\,\frac{a_0^3 - a_f^3}{u_M} + a_0\,v_f - a_f\,v_0 - u_M\,s_f\right)t_3 - \frac{1}{96}\frac{a_0^4 + a_f^4}{u_M^2} \\
&\quad + \frac{1}{24}\,\frac{a_0\,a_f}{u_M^2}(a_0^2 + a_f^2) - \frac{1}{16}\,\frac{a_0^2\,a_f^2}{u_M^2} - \frac{1}{2}\,(v_0^2 + v_f^2) + v_0\,v_f - a_0\,s_f + a_f\,s_f = 0\,.
\end{aligned} \tag{1.16}$$

In the case of negative initial jerk (i.e. $u_M = -j_M$), the optimal solution can be found by changing the sign of j_M in (1.9), (1.11) and (1.13) and then applying the same procedure exposed above. In sake of simplicity the three equations system for this case is omitted.

The optimal degenerate case

Consider a positive initial jerk value (i.e. $u_M = +j_M$). A solution of the three equations system (1.9), (1.11) and (1.13) exists only if the following relation holds (see (1.15))

$$j_M\,\bar{t}_f - a_f + a_0 \neq 0\,. \tag{1.17}$$

If (1.17) is not verified, follows that

$$a_0 + j_M \, \bar{t}_f = a_f \, ,$$

which corresponds to the optimal degenerate solution expressed by

$$t_1 = t_2 = 0 \, , \quad \bar{t}_f = \frac{a_f - a_0}{j_M} \, . \tag{1.18}$$

Hence, by virtue of condition $\bar{t}_f > 0$ the following inequality must hold

$$a_f > a_0 \, .$$

The optimal degenerate jerk is

$$\bar{u}(t) = j_M \, , \quad \forall t \in [0, \bar{t}_f] \, . \tag{1.19}$$

Note that solution (1.18) satisfies equation (1.9). Integrating (1.19) one deduces the acceleration function

$$x_3(t) = a_0 + j_M \, t \, , \quad \forall t \in [0, \bar{t}_f] \, .$$

In the same way the optimal velocity function is obtained

$$x_2(t) = v_0 + \int_0^t x_3(\xi) d\xi = v_0 + a_0 \, t + \frac{1}{2} \, j_M \, t^2 \, , \quad \forall t \in [0, \bar{t}_f] \, ,$$

and then the optimal space function is given by

$$x_1(t) = \int_0^t x_2(\xi) d\xi = v_0 \, t + \frac{1}{2} \, a_0 \, t^2 + \frac{1}{6} \, j_M \, t^3 \, , \quad \forall t \in [0, \bar{t}_f] \, .$$

If $t = \bar{t}_f$, by virtue of the boundary conditions (1.3) and (1.4) follows that

$$v_0 + a_0 \, \bar{t}_f + \frac{1}{2} \, j_M \, \bar{t}_f^2 = v_f \, , \tag{1.20}$$

and

$$v_0 \, \bar{t}_f + \frac{1}{2} \, a_0 \, \bar{t}_f^2 + \frac{1}{6} \, j_M \, \bar{t}_f^3 = s_f \, . \tag{1.21}$$

By substituting relation (1.18) in (1.20) the relation below is deduced

$$\frac{1}{2}\,\frac{a_f^2 - a_0^2}{j_M} + v_0 - v_f = 0\,. \tag{1.22}$$

Then, by substituting relation (1.18) in (1.21) the following equation holds

$$\frac{1}{6}\,\frac{a_f^2}{j_M^2} - \frac{2}{3}\,\frac{a_0^3}{j_M^2} - \frac{1}{2}\,\frac{a_0^2\,a_f}{j_M^2} + \frac{v_0\,a_0}{j_M} - \frac{v_0\,a_f}{j_M} - s_f = 0\,. \tag{1.23}$$

Relations (1.22) and (1.23) must be satisfied in the degenerate case. Note that they are exactly the second and the third equation of system (1.9), (1.11), (1.13) when it has solution (1.18).

In case of initial negative jerk (i.e. $u_M = -j_M$), the optimal degenerate solution is

$$\bar{u}(t) = -j_M\,, \quad \forall t \in [0, \bar{t}_f]\,,$$

corresponding to

$$t_1 = t_2 = 0\,, \quad \bar{t}_f = \frac{a_0 - a_f}{j_M}\,. \tag{1.24}$$

This degenerate case emerges with

$$a_0 > a_f\,,$$

and the following relations hold

$$\frac{1}{2}\,\frac{a_0^2 - a_f^2}{j_M} + v_0 - v_f = 0\,, \tag{1.25}$$

and

$$\frac{1}{6}\,\frac{a_f^2}{j_M^2} - \frac{2}{3}\,\frac{a_0^3}{j_M^2} - \frac{1}{2}\,\frac{a_0^2\,a_f}{j_M^2} - \frac{v_0\,a_0}{j_M} + \frac{v_0\,a_f}{j_M} - s_f = 0\,. \tag{1.26}$$

1.3.3 The minimum-time algorithm

The Minimum-Time Velocity Planning (**MTVP**) algorithm is presented by exploiting the algebraic solution exposed in subsection 1.3.2. This algorithm must verifies if a positive or a negative jerk degenerate solution exists; after that, if a degenerate solution was not found it checks the generic cases of initial positive and negative jerk solutions. Hence, the **MTVP** algorithm can be synthesized as follows:

```
begin
if $a_f > a_0$ then
  procedure PJDS;
end
if $a_f < a_0$ then
  procedure NJDS;
end
procedure PJS;
procedure NJS;
end
```

Then, the **MTVP** algorithm is composed of four separated procedures: the Positive Jerk Degenerate Solution (**PJDS**), the Negative Jerk Degenerate Solution (**NJDS**), the Positive Jerk Solution (**PJS**) and the Negative Jerk Solution (**NJS**). Let us describe these procedures in detail.

Procedure PJDS

This procedure starts if $a_f > a_0$, because is not possible to have a degenerate solution with positive initial jerk (i.e. $u_M = +j_M$) if $a_f \leq a_0$. If conditions (1.22) and (1.23) are verified the positive jerk degenerate solution (1.18) is imposed and the **MTVP** algorithm is stopped, otherwise the algorithm execution returns to the main program. The procedure is as follows:

```
begin
if $\frac{1}{2}\,\frac{a_f^2-a_0^2}{j_M} + v_0 - v_f = 0$ and
  $\frac{1}{6}\,\frac{a_f^2}{j_M^2} - \frac{2}{3}\,\frac{a_0^3}{j_M^2} - \frac{1}{2}\,\frac{a_0^2\,a_f}{j_M^2} + \frac{v_0\,a_0}{j_M} - \frac{v_0\,a_f}{j_M} - s_f = 0$ then
  $[t_1, t_2, \bar{t}_f] = [0, 0, \frac{a_f - a_0}{j_M}]$ ;
  exit
else
  return
end
```

Procedure NJDS

This procedure is dual to the **PJDS** one. If $a_f < a_0$ and conditions (1.25) and (1.26) are verified, the negative jerk degenerate solution (1.24) is imposed and the main program is stopped.

```
begin
if $\frac{1}{2}\frac{a_0^2 - a_f^2}{j_M} + v_0 - v_f = 0$ and
  $\frac{1}{6}\frac{a_f^2}{j_M^2} - \frac{2}{3}\frac{a_0^3}{j_M^2} - \frac{1}{2}\frac{a_0^2\,a_f}{j_M^2} - \frac{v_0\,a_0}{j_M} + \frac{v_0\,a_f}{j_M} - s_f = 0$ then

  $[t_1, t_2, \bar{t}_f] = [0, 0, \frac{a_0 - a_f}{j_M}]$ ;
  exit
else
  return
end
```

Procedure PJS

First, all the positive real roots of quartic equation (1.16) are computed and stored in an array **T**. Then expressions (1.14) and (1.15) are used to determine a feasible solution. If three values of t_1, t_2, and $\bar{t}_f$ satisfying inequalities $0 \leq t_1 \leq t_2 \leq \bar{t}_f$ are found the minimum-time velocity planning solution is obtained and the main program is stopped.

```
begin
Compute the positive real roots of
   equation (1.16), $\mathbf{T} = [t_{f1}, t_{f2}, \ldots, t_{fl}]$ with $(l \leq 4)$ ;
if $\mathbf{T}$ is empty then
   return
for $i = 1, \ldots, l$ do
  $t_{2i} = \frac{\left[\frac{3}{4}\, j_M\, t_{fi}^2 - \frac{1}{2}\,(3\, a_f - a_0)\, t_{fi} + \frac{1}{4\, j_M}\,(a_f - a_0)^2 + v_f - v_0\right]}{j_M\, t_{fi} - a_f + a_0}$ ;

  if $0 \leq t_{2i} \leq t_{fi}$ then
    $t_{1i} = t_{2i} - \frac{1}{2}\, t_{fi}^2 + \frac{1}{2}\, \frac{a_f - a_0}{j_M}$ ;

    if $0 \leq t_{1i} \leq t_{2i}$ then
      $[t_1, t_2, \bar{t}_f] = [t_{1i}, t_{2i}, t_{3i}]$ ;
      exit
    else
      continue
  else
    continue
return
end
```

Procedure NJS

This procedure is dual to the **PJS** one. The quartic equation to start with is the modified (1.16) where j_M is substituted by $-j_M$. Then all the positive real solutions of this equation are computed and a feasible solution is sought.

```
begin
In equation (1.16) do the substitution $j_M \leftarrow -j_M$
   and compute the positive real roots,
  $\mathbf{T} = [t_{f1}, t_{f2}, \ldots, t_{fl}]$ with $(l \leq 4)$ ;
```

```
if $\mathbf{T}$ is empty then
   return
for $i = 1, \ldots, l$ do
   $t_{2i} = \frac{\left[\frac{3}{4}\, j_M\, t_{fi}^2 - \frac{1}{2}\,(3\, a_f - a_0)\, t_{fi} + \frac{1}{4\, j_M}\,(a_f - a_0)^2 + v_f - v_0\right]}{j_M\, t_{fi} - a_f + a_0}$ ;

   if $0 \leq t_{2i} \leq t_{fi}$ then
      $t_{1i} = t_{2i} - \frac{1}{2}\, t_{fi}^2 + \frac{1}{2}\, \frac{a_f - a_0}{j_M}$ ;

      if $0 \leq t_{1i} \leq t_{2i}$ then
         $[t_1, t_2, \bar{t}_f] = [t_{1i}, t_{2i}, t_{3i}]$ ;
         exit
      else
         continue
   else
      continue
return
end
```

1.3.4 Simulations results

Example 1: consider the following data: $s_f = 3,25$ m, $j_M = 0,5$ m/s^3, $v_0 = 0$ m/s, $a_0 = 0$ m/s^2, $v_f = 2,25$ m/s and $a_f = 1,5$ m/s^2. Exploiting the **MTVP** algorithm described in subsection 1.3.3 the following optimal solution is obtained:

$$u_M = +j_M \quad t_1 = 1 \text{ s} \quad t_2 = 3 \text{ s} \quad \bar{t}_f = 7 \text{ s}$$

The jerk, acceleration, velocity and space profiles, for this case, are depicted in figure 1.3.

Example 2: let be the case of: $s_f = 8,42$ m, $j_M = 0,25$ m/s^3, $v_0 = 1$ m/s, $a_0 = 0,5$ m/s^2, $v_f = 2,75$ m/s and $a_f = 0$ m/s^2. The optimal solution is the following:

$$u_M = +j_M \quad t_1 = 1 \text{ s} \quad t_2 = \bar{t}_f = 4 \text{ s}$$

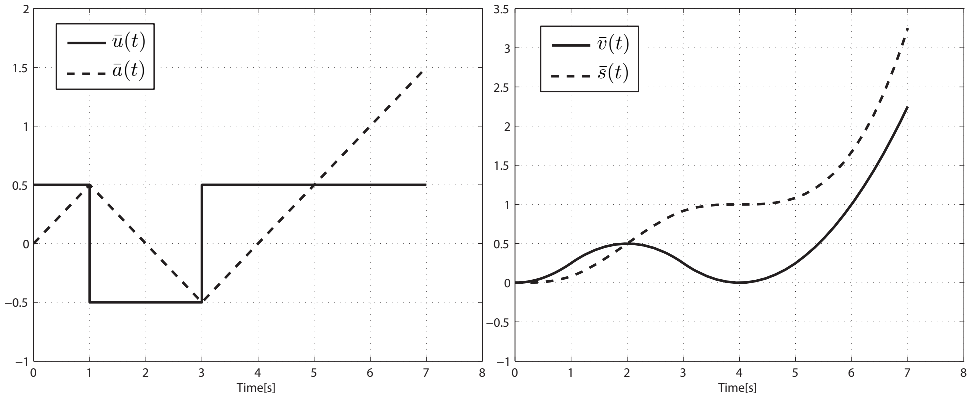


Figure 1.3: The optimal profiles of jerk $\bar{u}(t)$, acceleration $\bar{a}(t)$, velocity $\bar{v}(t)$, and space $\bar{s}(t)$ for example 1.

See figure 1.4 for the optimal $\bar{u}(t)$, $\bar{a}(t)$, $\bar{v}(t)$ and $\bar{s}(t)$ profiles.

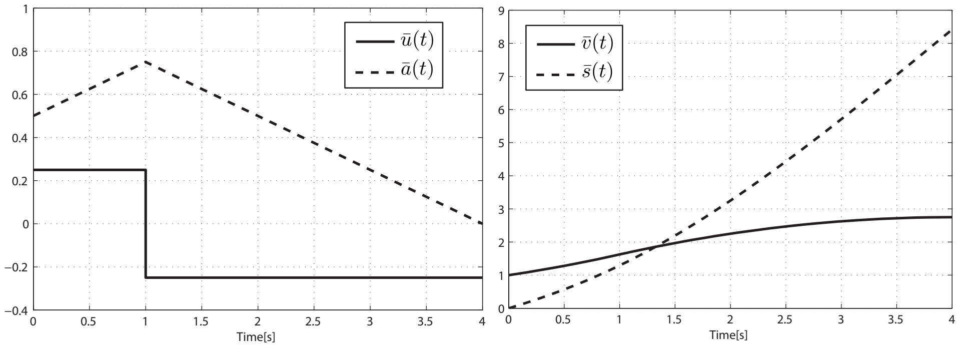


Figure 1.4: The optimal profiles of jerk $\bar{u}(t)$, acceleration $\bar{a}(t)$, velocity $\bar{v}(t)$, and space $\bar{s}(t)$ for example 2.

1.4 Minimum-time constrained velocity planning

This section explains a procedure which has appeared for the first time in [17]. The proposed method solves again the minimum-time velocity planning problem with generic initial and final boundary conditions for the velocity and the acceleration but with constrains not only on the jerk but on velocity and acceleration too.

This minimum-time planning problem is relevant in the context of robotic autonomous navigation, where the iterative steering supervisor periodically replans the future mobile robot motion starting from current position, velocity and acceleration conditions. The problem is faced through discretization and its solution is based on a sequence of linear programming feasibility checks, depending on motion constraints and boundary conditions.

1.4.1 Problem statement and sufficient condition

The faced problem is the minimum-time planning of a smooth velocity profile $v(t) \in PC^2([0, t_f])$ (see definition 2), where t_f represents the travelling minimum-time along a given path whose length is equal to s_f, respecting given velocity, acceleration, and jerk constraints. Formally:

$$\min_{v \in PC^2} t_f \,, \tag{1.27}$$

such that

$$\int_0^{t_f} v(\xi) d\xi = s_f \,, \tag{1.28}$$

$$v(0) = v_0 \,, \quad v(t_f) = v_f \,, \tag{1.29}$$

$$\dot{v}(0) = a_0 \,, \quad \dot{v}(t_f) = a_f \,, \tag{1.30}$$

$$|v(t)| \leq v_M \,, \quad \forall t \in [0, t_f] \,, \tag{1.31}$$

$$|\dot{v}(t)| \leq a_M \,, \quad \forall t \in [0, t_f] \,, \tag{1.32}$$

$$|\ddot{v}(t)| \leq j_M \,, \quad \forall t \in [0, t_f] \,, \tag{1.33}$$

where $s_f, v_M, a_M, j_M \in \mathbb{R}_+$ and $v_0, v_f, a_0, a_f \in \mathbb{R}$ are given boundary conditions. For the special case of zero boundary conditions (i.e. $v_0 = v_f = 0$, $a_0 = a_f = 0$) a closed form solution has been provided by [18]. Remark that in our context of iterative autonomous navigation, it is crucial to consider generic boundary conditions on initial and final velocities and accelerations.

Such as in section 1.3 the problem is recasted into a minimum-time control problem with respect to a suitable state-space system. Indeed consider again the jerk $\ddot{v}(t)$ as the control input $u(t)$ of the cascade of three integrators as depicted in figure 1.1. The system equations are still given by (1.8). Constraints (1.31), (1.32) and (1.33) will be considered as two state constraints and an input bound respectively. Hence, problem (1.27)-(1.33) is equivalent to finding a time-optimal control $\bar{u}(t)$ that brings system (1.8) from the initial state $\mathbf{x}(0) = [0\, v_0\, a_0]'$ to the final state $\mathbf{x}(\bar{t}_f) = [s_f\, v_f\, a_f]'$ in minimum time $\bar{t}_f$, while satisfying the following constraints

$$|x_2(t)| \leq v_M\,, \quad \forall t \in [0, \bar{t}_f]\,, \tag{1.34}$$

$$|x_3(t)| \leq a_M\,, \quad \forall t \in [0, \bar{t}_f]\,, \tag{1.35}$$

and

$$|\bar{u}(t)| \leq j_M\,, \quad \forall t \in [0, \bar{t}_f]\,. \tag{1.36}$$

In the case of constrained state, it is not guarantee that a time-optimal control $\bar{u}(t)$ exists. The existence of solution $\bar{u}(t)$ of problem(1.27)-(1.33) depends on the values of the initial state $\mathbf{x}_0$, the final state $\mathbf{x}_f$, and it also depends on the constraints (1.34)-(1.36). To guarantee the existence of the optimal control $\bar{u}(t)$, these values must respect four sufficient conditions as stated in the following result.

Proposition 2 *The minimum-time optimal control $\bar{u}(t)$, solution of problem* (1.27)-(1.33)*, from initial state $\mathbf{x}(0) = [0\, v_0\, a_0]'$ to final state $\mathbf{x}(\bar{t}_f) = [s_f\, v_f\, a_f]'$ exists if the following sufficient conditions are satisfied:*

$$|v_0| \leq v_M\,, \qquad |v_f| \leq v_M\,, \tag{1.37}$$

$$|a_0| \leq a_M\,, \qquad |a_f| \leq a_M\,, \tag{1.38}$$

$$\textit{if} \quad a_0 \geq 0 \quad \textit{then} \quad v_0 + \frac{1}{2}\frac{a_0^2}{j_M} \leq v_M\,, \tag{1.39}$$

$$\textit{if} \quad a_0 < 0 \quad \textit{then} \quad v_0 - \frac{1}{2}\,\frac{a_0^2}{j_M} \geq 0\,, \tag{1.40}$$

$$\textit{if} \quad a_f \geq 0 \quad \textit{then} \quad v_f - \frac{1}{2}\,\frac{a_f^2}{j_M} \geq 0\,, \tag{1.41}$$

$$\textit{if} \quad a_f < 0 \quad \textit{then} \quad v_f + \frac{1}{2}\,\frac{a_f^2}{j_M} \leq v_M\,, \tag{1.42}$$

and

$$s_f \geq s_{ref}\,, \tag{1.43}$$

where s_{ref} is a reference distance depending on the problem data which is defined below by a four-step procedure:

1.

$$s_1 := \frac{v_0\,|a_0|}{j_M} + \frac{1}{3}\,\frac{a_0^3}{j_M^2} \quad \textit{and} \quad v_1 := v_0 + \operatorname{sgn}(a_0)\,\frac{1}{2}\,\frac{a_0^2}{j_M}\,.$$

2.

$$s_2 := \frac{v_f\,|a_f|}{j_M} - \frac{1}{3}\,\frac{a_f^3}{j_M^2} \quad \textit{and} \quad v_2 := v_f - \operatorname{sgn}(a_f)\,\frac{1}{2}\,\frac{a_f^2}{j_M}\,.$$

3. *if* $\sqrt{j_M\,|v_1 - v_2|} \leq a_M$ *then*

$$v_{ref} := \max(v_1, v_2)\,,$$

$$s_c := \frac{2\,v_{ref}\,\sqrt{j_M\,|v_1 - v_2|}}{j_M} - \frac{[j_M\,|v_1 - v_2|]^{3/2}}{j_M^2}\,,$$

else

$$s_c := \frac{1}{2}\,\frac{|v_1^2 - v_2^2|}{a_M} + \frac{1}{2}\,\frac{a_M\,(v_1 + v_2)}{j_M}\,.$$

4. $s_{ref} := s_1 + s_c + s_2\,.$

Proof. The argument of the proof uses the equivalence of problem (1.27)-(1.33) with the constrained control problem (1.34)-(1.36). Specifically, it shall be found a control input $u(t)$ that brings the state from $[0\ v_0\ a_0]'$ to $[s_f\ v_f\ a_f]'$ while satisfying the imposed state constraints. Obviously, if this input exists, then the optimal one will exists too.

Consider the case $a_0 \geq 0$. If conditions (1.37), (1.38) and (1.39) on initial state $\mathbf{x}(0)$ hold, it is possible to apply a control function $u(t) = -j_M$ which brings the acceleration $x_3(t)$ to zero before the velocity $x_2(t)$ exceeds its boundary value v_M. In fact, if $u(t) = -j_M$ with $t \in [0, t_1]$ (where t_1 is the critical time where the acceleration became null) the following result is true

$$x_3(t) = a_0 + \int_0^t u(\xi) d\xi = a_0 - j_M \, t \, . \tag{1.44}$$

But in $t = t_1$ we have $x_3(t_1) = 0$, so it is possible to obtain the critical time value

$$t_1 = \frac{a_0}{j_M} \, . \tag{1.45}$$

Integrating equation (1.44) in $[0, t_1]$, it follows that

$$x_2(t) = v_0 + \int_0^t x_3(\xi) d\xi = v_0 + a_0 \, t - \frac{1}{2} \, j_M \, t^2 \, . \tag{1.46}$$

In $t = t_1$, by substituting relation (1.45) in (1.46), the value of $v_1 = x_2(t_1)$ is obtained

$$v_1 = v_0 + \frac{1}{2} \frac{a_0^2}{j_M} \, ,$$

then, by virtue of condition (1.39) we know that $v_1 \leq v_M$ and constraint (1.34) is satisfied. The traveled space at time t_1 is

$$s_1 = \int_0^{t_1} x_2(\xi) d\xi = \frac{v_0 \, a_0}{j_M} + \frac{1}{3} \frac{a_0^3}{j_M^2} \, . \tag{1.47}$$

Consider the case of $a_f < 0$; if conditions (1.37), (1.38) and (1.42) are verified, the final state $x(t_f)$ can be reached by applying the control function $u(t) = -j_M$, with $t \in [t_2, t_f]$. The acceleration function is given by

$$x_3(t) = \int_{t_2}^t u(\xi) d\xi = -j_M \, (t - t_2) \, , \tag{1.48}$$

and in $t = t_f$ we have $x_3(t_f) = a_f$, so it is possible to obtain

$$t_f - t_2 = -\frac{a_f}{j_M} \, . \tag{1.49}$$

By integrating equation (1.48) in $[t_2, t_f]$, we get

$$x_2(t) = v_2 + \int_{t_2}^{t_f} x_3(\xi) = v_2 - \frac{1}{2}\, j_M\, (t_f - t_2)^2 \,. \tag{1.50}$$

In $t = t_f$, by substituting relation (1.49) in (1.50), the value of $v_2 = x_2(t_2)$ is

$$v_2 = v_f + \frac{1}{2}\,\frac{a_f^2}{j_M}\,,$$

then, by virtue of condition (1.42), constraint (1.34) holds. The traveled space for $t \in [t_2, t_f]$ is

$$s_2 = \int_{t_2}^{t_f} x_2(\xi) d\xi = -\frac{v_f\, a_f}{j_M} + \frac{1}{3}\,\frac{a_f^3}{j_M^2}\,. \tag{1.51}$$

If $v_1 = v_2$, the total traveled space is $s_f = s_1 + s_2$, where s_1 and s_2 are given by (1.47) and (1.51) respectively, then condition (1.43) is verified.

Consider the case of $v_1 > v_2$: by defining t_c as the time instant when $x_3(t_c) = -a_c$, where $-a_c$ is the acceleration minimum value, and if $a_c \leq a_M$, it is possible to interpolate v_1 and v_2 with the following control jerk function:

$$\begin{cases} u(t) = -j_M & t \in [t_1, t_c] \\ u(t) = j_M & t \in [t_c, t_2]\,, \end{cases}$$

where $t_c - t_1 = t_2 - t_c$. Then, for $u(t) = -j_M$ in $t \in [t_1, t_c]$ the acceleration function is given by

$$x_3(t) = \int_{t_1}^{t} u(\xi) d\xi = -j_M\, (t - t_1)\,. \tag{1.52}$$

But for $t = t_c$ we have $x_3(t_c) = -a_c$ so it is possible to obtain

$$t_c - t_1 = \frac{a_c}{j_M}\,. \tag{1.53}$$

By integrating equation (1.52) one deduces the velocity function

$$x_2(t) = v_1 + \int_{t_1}^{t} x_3(\xi) d\xi = v_1 - \frac{1}{2}\, j_M\, (t - t_1)^2 \,. \tag{1.54}$$

In $t = t_c$, by substituting (1.53) in (1.54), the velocity is given by

$$x_2(t_c) = v_1 - \frac{1}{2}\,\frac{a_c^2}{j_M}\,.$$

The distance s_3 covered in the time-interval $[t_1, t_c]$ is deduced as follows,

$$s_3 = \int_{t_1}^{t_c} x_2(\xi) d\xi = v_1\,\frac{a_c}{j_M} - \frac{1}{6}\,\frac{a_c^3}{j_M^2}\,. \tag{1.55}$$

By applying the same procedure in the time-interval $[t_2, t_c]$, with $u(t) = j_M$, the following result is obtained

$$x_2(t_c) = v_2 + \frac{1}{2}\,\frac{a_c^2}{j_M}\,,$$

while the traveled space s_4 is given by

$$s_4 = v_1\,\frac{a_c}{j_M} - \frac{5}{6}\,\frac{a_c^3}{j_M^2}\,.$$

In $t = t_c$ we have

$$v_1 - \frac{1}{2}\,\frac{a_c^2}{j_M} = v_2 + \frac{1}{2}\,\frac{a_c^2}{j_M}\,, \tag{1.56}$$

and solving equation (1.56) for a_c, the following equality holds

$$a_c = \sqrt{j_M\,(v_1 - v_2)}\,. \tag{1.57}$$

The distance s_c, covered in the time-interval $[t_1, t_2]$, is given by

$$s_c = s_3 + s_4 = \frac{2\,v_1\,\sqrt{j_M\,(v_1 - v_2)}}{j_M} - \frac{[j_M\,(v_1 - v_2)]^{3/2}}{j_M^2}\,, \tag{1.58}$$

where a_c was substituted with relation (1.57). For the time-interval $[0, t_f]$, the total traveled space is $s_f = s_1 + s_c + s_2$, where s_1, s_2 and s_c are given by relations (1.47), (1.51) and (1.58) respectively, then condition (1.43) is verified.

Finally consider $v_1 > v_2$ and $a_c = \sqrt{j_M\,(v_1 - v_2)} > a_M$. In this case it will exists a time-interval $[t_{c1}, t_{c2}]$, where acceleration $x_1(t)$ will be equal to its minimum value $-a_M$, while the control function will be $u(t) = 0$. In $t = t_{c1}$ and in $t = t_{c2}$, the velocity values are given by

$$x_2(t_{c1}) = v_1 - \frac{1}{2}\frac{a_M^2}{j_M},$$

and

$$x_2(t_{c2}) = v_2 + \frac{1}{2}\frac{a_M^2}{j_M},$$

respectively. Moreover, in $t = t_{c2}$ the following relation holds

$$v_2 + \frac{1}{2}\frac{a_M^2}{j_M} = v_1 - \frac{1}{2}\frac{a_M^2}{j_M} - a_M\,(t_{c2} - t_{c1})\,. \tag{1.59}$$

From equation (1.59) the following equality is obtained

$$t_{c2} - t_{c1} = \frac{v_1 - v_2}{a_M} - \frac{a_M}{j_M}\,. \tag{1.60}$$

The traveled space in $[t_{c1}, t_{c2}]$ is given by

$$s_5 = \frac{1}{2}\,(v_1 + v_2)\,(t_{c2} - t_{c1})\,, \tag{1.61}$$

and by substituting relation (1.60) in (1.61) it is possible to obtain

$$s_5 = \frac{1}{2}\frac{(v_1^2 - v_2^2)}{a_M} - \frac{1}{2}\frac{a_M\,(v_1 + v_2)}{j_M}\,. \tag{1.62}$$

The distance s_c covered for $t \in [t_1, t_2]$ is obtained by summing s_3 and s_5, given by (1.55) and (1.62) respectively, with $s_4 = \frac{v_2\,a_M}{j_M} + \frac{1}{6}\frac{a_M^3}{j_M^2}$, and it results to be

$$s_c = \frac{1}{2}\frac{(v_1^2 - v_2^2)}{a_M} + \frac{1}{2}\frac{a_M\,(v_1 + v_2)}{j_M}\,. \tag{1.63}$$

Then, the total traveled space is $s_f = s_1 + s_c + s_3$, where s_1, s_2 and s_c are given by (1.47), (1.51) and (1.63) respectively, and condition (1.43) is verified. The other sufficient conditions can be proved in the same way saw above. □

1.4.2 An approximated solution using discretization

This subsection shows how to find a numerically approximated solution of problem (1.27)-(1.33) by discretization of system (1.8). The technique that will be introduced, exploits the result presented by Consolini and Piazzi in [19], which shows that, given a continuous-time system, an approximated optimal control can be found through the following procedure:

1. find the discretized system with sampling period T_s;
2. find the optimal input sequence $\bar{u}(k)$;
3. use for the continuous-time system the input function $u(t)$ obtained from the discrete-time sequence with a zero-order hold

   $$u(t) = \bar{u}_{T_s}\left(\lfloor \frac{t}{T_s} \rfloor\right) \,,$$

   where $T_s \in \mathbb{R}$ is the sampling period and $\forall x \in \mathbb{R}$,

   $$\lfloor x \rfloor = \max\{z \in \mathbb{Z} \,:\, z \leq x\} \,,$$

   denotes the integer part of x.

As shown in [19], when $T_s \to 0$ the approximated solution converges to the optimal continuous-time solution.

The optimal discrete-time control sequence $\bar{u}(t)$ can be found by means of linear programming. In fact, in the discrete-time case, the constraints (1.34)-(1.36) can be represented as linear inequalities and the minimum number of steps needed for the requested transition can be found through a sequence of feasibility tests of a linear programming problem.

The matrices of the equivalent discrete-time system are the following ones:

$$\mathbf{A}_d = e^{\mathbf{A}\,T_s} = \begin{bmatrix} 1 & T_s & \frac{1}{2}T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \,,$$

and

$$\mathbf{B}_d = \mathbf{f}(\mathbf{A}, T_s)\,\mathbf{B} = \left(\int_0^{T_s} e^{\mathbf{A}\,\tau} d\tau\right)\mathbf{B} = \begin{bmatrix} \frac{1}{6}T_s^3 \\ \frac{1}{2}T_s^2 \\ T_s \end{bmatrix},$$

where T_s is the sampling period. Then, the discrete-time system is

$$\mathbf{x}(k+1) = \mathbf{A}_d\,\mathbf{x}(k) + \mathbf{B}_d\,\mathbf{u}(k)\,, \tag{1.64}$$

whose solution is given by

$$\mathbf{x}(k) = \mathbf{A}_d^k\,\mathbf{x}_0 + \sum_{j=0}^{k-1} \mathbf{A}_d^{k-1-j}\mathbf{B}_d\,\mathbf{u}(j)\,, \tag{1.65}$$

where

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}.$$

Define the control vector $\mathbf{u} \in \mathbb{R}^{k_f}$ as follows

$$\mathbf{u} = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(k_f - 1) \end{bmatrix},$$

from (1.36) it follows that it must be

$$-u_M \cdot \mathbf{1}_{k_f} \le \mathbf{u} \le u_M \cdot \mathbf{1}_{k_f}\,,$$

where $\mathbf{1}_{k_f}$ denotes the k_f-dimensional vector whose components are all equal to 1. The velocity constraint for discrete-time system is given by

$$-v_M \le x_2(k) \le v_M\,, \quad \text{with} \quad k = 0, \ldots, k_f - 1\,. \tag{1.66}$$

From equation (1.65), velocity sequence $x_2(k)$ can be written as follows

$$\begin{aligned} x_2(k) &= \mathbf{C}_1\,\mathbf{x}(k) \\ &= \mathbf{C}_1\left[\mathbf{A}_d^k\,\mathbf{x}_0 + \sum_{j=0}^{k-1}\mathbf{A}_d^{k-1-j}\mathbf{B}_d\,\mathbf{u}(j)\right] \\ &= \mathbf{C}_1\,\mathbf{A}_d^k\,\mathbf{x}_0 + \sum_{j=0}^{k-1}\mathbf{C}_1\mathbf{A}_d^{k-1-j}\mathbf{B}_d\,\mathbf{u}(j)\,, \end{aligned} \tag{1.67}$$

where

$$\mathbf{C}_1 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

By substituting (1.67) in (1.66), the following relation is obtained

$$-v_M - \mathbf{C}_1\,\mathbf{A}_d^k\,\mathbf{x}_0 \leq \sum_{j=0}^{k-1}\mathbf{C}_1\mathbf{A}_d^{k-1-j}\mathbf{B}_d\,\mathbf{u}(j) \leq v_M - \mathbf{C}_1\,\mathbf{A}_d^k\,\mathbf{x}_0\,,$$

with $k = 0, \ldots, k_f - 1$. Set $\mathbf{v}_c = v_M \cdot \mathbf{1}_f$, then the inequality on velocity constraint (1.66) become

$$-\mathbf{v}_c - \mathbf{G}_1 \leq \mathbf{H}_1\,\mathbf{u} \leq \mathbf{v}_c - \mathbf{G}_1\,,$$

where $\mathbf{G}_1 \in \mathbb{R}^{k_f}$ and $\mathbf{H}_1 \in \mathbb{R}^{k_f \times k_f}$ are given by

$$\mathbf{G}_1 = \begin{bmatrix} \mathbf{C}_1\,\mathbf{x}_0 \\ \mathbf{C}_1\,\mathbf{A}_d\,\mathbf{x}_0 \\ \mathbf{C}_1\,\mathbf{A}_d^2\,\mathbf{x}_0 \\ \vdots \\ \mathbf{C}_1\,\mathbf{A}_d^{k_f-1}\,\mathbf{x}_0 \end{bmatrix},$$

and

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{C}_1\,\mathbf{B}_d & \mathrm{O} & \cdots & \mathrm{O} \\ \mathbf{C}_1\,\mathbf{A}_d\,\mathbf{B}_d & \ddots & \ddots & \mathrm{O} \\ \mathbf{C}_1\,\mathbf{A}_d^2\,\mathbf{B}_d & \ddots & \ddots & \mathrm{O} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{C}_1\,\mathbf{A}_d^{k_f-1}\,\mathbf{B}_d & \cdots & \cdots & \mathbf{C}_1\,\mathbf{B}_d \end{bmatrix}.$$

The acceleration constraint for discrete-time system (1.64) is given by

$$-a_M \le x_3(k) \le a_M\,, \quad \text{with} \quad k = 0, \ldots, k_f - 1\,. \tag{1.68}$$

Set $\mathbf{a}_c = a_M \cdot \mathbf{1}_f$ and

$$\mathbf{C}_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},$$

then, constraint (1.68) is written as

$$-\mathbf{a}_c - \mathbf{G}_2 \le \mathbf{H}_2\,\mathbf{u} \le \mathbf{a}_c - \mathbf{G}_2\,,$$

where $\mathbf{G}_2 \in \mathbb{R}^{k_f}$ and $\mathbf{H}_2 \in \mathbb{R}^{k_f \times k_f}$ are given by

$$\mathbf{G}_2 = \begin{bmatrix} \mathbf{C}_2\,\mathbf{x}_0 \\ \mathbf{C}_2\,\mathbf{A}_d\,\mathbf{x}_0 \\ \mathbf{C}_2\,\mathbf{A}_d^2\,\mathbf{x}_0 \\ \vdots \\ \mathbf{C}_2\,\mathbf{A}_d^{k_f-1}\,\mathbf{x}_0 \end{bmatrix},$$

and

$$\mathbf{H}_2 = \begin{bmatrix} \mathbf{C}_2\,\mathbf{B}_d & \mathrm{O} & \cdots & \mathrm{O} \\ \mathbf{C}_2\,\mathbf{A}_d\,\mathbf{B}_d & \ddots & \ddots & \mathrm{O} \\ \mathbf{C}_2\,\mathbf{A}_d^2\,\mathbf{B}_d & \ddots & \ddots & \mathrm{O} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{C}_2\,\mathbf{A}_d^{k_f-1}\,\mathbf{B}_d & \cdots & \cdots & \mathbf{C}_2\,\mathbf{B}_d \end{bmatrix}.$$

The interpolation condition on final state can be written as follows

$$\mathbf{x}_f = \mathbf{x}(k_f) = \begin{bmatrix} x_1(k_f) \\ x_2(k_f) \\ x_3(k_f) \end{bmatrix} = \begin{bmatrix} s_f \\ v_f \\ a_f \end{bmatrix}. \tag{1.69}$$

From equation (1.65) we have

$$\mathbf{x}_f = \mathbf{A}_d^{k_f}\,\mathbf{x}_0 + \sum_{j=0}^{k_f-1} \mathbf{A}_d^{k_f-1-j}\mathbf{B}_d\,\mathbf{u}(j)\,, \tag{1.70}$$

then, by substituting equation (1.70) in (1.69) we obtain the final state interpolation condition as follows

$$\mathbf{H}_{eq}\,\mathbf{u} = \mathbf{x}_f - \mathbf{A}_d^{k_f}\,\mathbf{x}_0\,,$$

where $\mathbf{H}_{eq} \in \mathbb{R}^{3\times k_f}$ is given by

$$\mathbf{H}_{eq} = \left[\begin{array}{cccc} \mathbf{A}_d^{k_f-1}\,\mathbf{B}_d & \mathbf{A}_d^{k_f-2}\,\mathbf{B}_d & \cdots & \mathbf{B}_d \end{array}\right].$$

In conclusion given a number of steps k_f, there exists an input vector $\mathbf{u}$ for which the constraints on velocity, acceleration and jerk, and the final interpolation condition are satisfied if and only if the following linear programming problem is feasible

$$\begin{cases} -u_M\cdot\mathbf{1}_{k_f} \le \mathbf{u} \le u_M\cdot\mathbf{1}_{k_f} \\ -\mathbf{v}_c - \mathbf{G}_1 \le \mathbf{H}_1\,\mathbf{u} \le \mathbf{v}_c - \mathbf{G}_1 \\ -\mathbf{a}_c - \mathbf{G}_2 \le \mathbf{H}_2\,\mathbf{u} \le \mathbf{a}_c - \mathbf{G}_2 \\ \mathbf{H}_{eq}\,\mathbf{u} = \mathbf{x}_f - \mathbf{A}_d^{k_f}\,\mathbf{x}_0\,. \end{cases} \tag{1.71}$$

1.4.3 The bisection algorithm

The minimum number of steps $\bar{k}_f$ and the associated optimal discrete-time control sequence $\bar{u}(k)$, with $k = 0, \ldots, \bar{k}_f - 1$, can be determined by means of a sequence of linear programming feasibility tests, defined by (1.71), through a simple bisection algorithm. The Minimum-Time Control algorithm (**MTC**) is summarized as follows:

```
begin
    $k_f \leftarrow 1;$
    $l \leftarrow 0;$
    while $\backsim \boldsymbol{LPP}$ do
      $l \leftarrow k_f$
      $k_f \leftarrow 2\,k_f$
    end
```

```
  $h \leftarrow k_f$;
  while $h - l > 1$ do
    $k_f \leftarrow \lfloor \frac{h+l}{2} \rfloor$;
      if $\backsim$ $\boldsymbol{LPP}$ then
        $l \leftarrow k_f$;
      else
        $h \leftarrow k_f$;
    end
  $k_f^* \leftarrow h$;
  $u^*(k) \leftarrow \mathbf{u}$;
end
```

In **MTC** algorithm ***LPP*** denotes a linear programming procedure that solves problem (1.71), which, if a feasible solution exists, returns the solution sequence $\mathbf{u}$ and the number of steps k; if the problem is feasible it also returns a Boolean true value.

The algorithm performances strongly depend on the used sampling time. By reducing T_s, which means sampling the continuous-time system with an higher frequency, the dimension of the resulting linear programming problem increases, thus causing an increment of the total computational time. Considering the computational complexity, Karmarkar has shown in [20] that a linear programming problem can be solved by means of an interior-point algorithm with running time proportional to $n^{3.5}$, where n is the number of inequalities. In our case this would means that each feasibility test would require a time proportional to $n_s^{3.5}$, where n_s is the total number of samples. The complexity of the bisection search, with respect to the minimum number of samples, is given by $O(\log n_s)$, therefore the total complexity of the proposed algorithm is given by $O(n_s^{3.5}\ \log n_s)$. For more details on the algorithm complexity see [21].

1.4.4 Simulations results

Example 1: consider the following interpolation conditions and constraints:

- initial state

$$\mathbf{x}_0 := \begin{bmatrix} s_0 \\ v_0 \\ a_0 \end{bmatrix} := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- final state

$$\mathbf{x}_f := \begin{bmatrix} s_f \\ v_f \\ a_f \end{bmatrix} := \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

- problem constraints

$$v_M = 0,65 \text{ m/s} \quad a_M = 0.5 \text{ m/s}^2 \quad j_M = 0.5 \text{ m/s}^3$$

The jerk, acceleration, velocity and space profiles, obtained by means of the **MTC** algorithm, are depicted in figure 1.5.

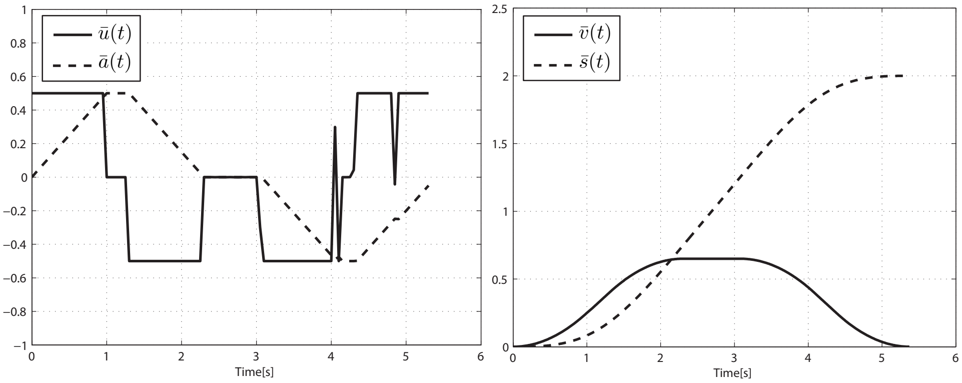


Figure 1.5: The pseudo-optimal profiles of jerk $\bar{u}(t)$, acceleration $\bar{a}(t)$, velocity $\bar{v}(t)$, and space $\bar{s}(t)$ for example 1.

Example 2: consider the following problem:

- initial state

$$\mathbf{x}_0 := \begin{bmatrix} s_0 \\ v_0 \\ a_0 \end{bmatrix} := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- final state

$$\mathbf{x}_f := \begin{bmatrix} s_f \\ v_f \\ a_f \end{bmatrix} := \begin{bmatrix} 2 \\ 1 \\ 0,25 \end{bmatrix}$$

- problem constraints

$$v_M = 1,5 \text{ m/s} \quad a_M = 0.6 \text{ m/s}^2 \quad j_M = 0.5 \text{ m/s}^3$$

The jerk, acceleration, velocity and space profiles, obtained in this case, are depicted in figure 1.6.

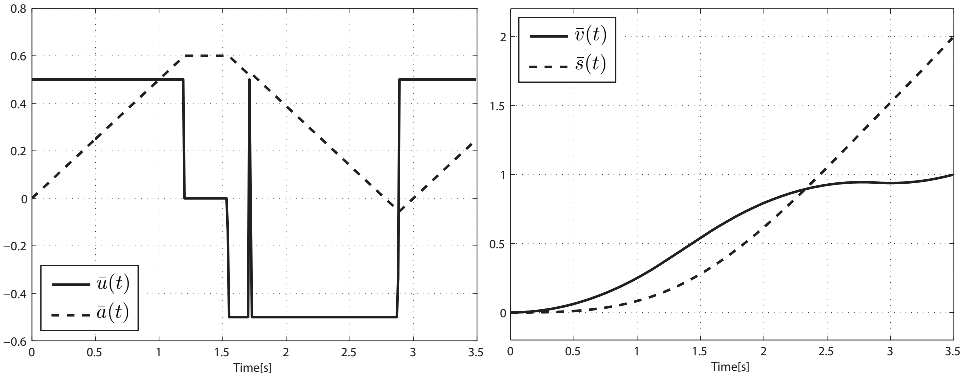


Figure 1.6: The pseudo-optimal profiles of jerk $\bar{u}(t)$, acceleration $\bar{a}(t)$, velocity $\bar{v}(t)$, and space $\bar{s}(t)$ for example 2.

Example 3: the problem data are given by:

- initial state

$$\mathbf{x}_0 := \begin{bmatrix} s_0 \\ v_0 \\ a_0 \end{bmatrix} := \begin{bmatrix} 0 \\ 1 \\ -0.5 \end{bmatrix}$$

- final state

$$\mathbf{x}_f := \begin{bmatrix} s_f \\ v_f \\ a_f \end{bmatrix} := \begin{bmatrix} 2,167 \\ 0,5 \\ 0,5 \end{bmatrix}$$

- problem constraints

$$v_M = 1 \text{ m/s} \quad a_M = 0.5 \text{ m/s2} \quad j_M = 0.5 \text{ m/s}^3$$

Figure 1.7 shows optimal solution obtained by means of the **MTC** algorithm.

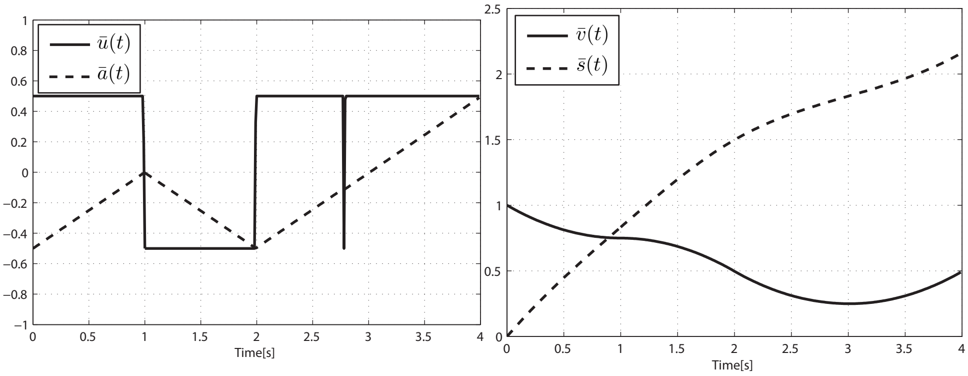


Figure 1.7: The pseudo-optimal profiles of jerk $\bar{u}(t)$, acceleration $\bar{a}(t)$, velocity $\bar{v}(t)$, and space $\bar{s}(t)$ for example 3.

Chapter 2

Path generation and autonomous parking

A goal without a plan
is just a wish.

– Antoine de Saint-Exupery

In this chapter the problem of the path planning for nonholonomic vehicles is discussed. The two methods presented in the following are well suited for their implementation into the framework of autonomous parking of autonomous vehicles.

Fist section proposes a multi-optimization approach to the autonomous parking of car-like vehicles [22]. It uses a polynomial curve primitive, the $\boldsymbol{\eta}^3$-spline, to build up intrinsically feasible path maneuvers over which to minimize with a weighted sum method the total length of parking paths and the moduli of the maximum path curvature and curvature derivative. The approach takes into account the mandatory constraint of obstacle avoidance and maximal steering angle and the constraint of maximal curvature derivative which is a selectable limit to ensure the desired smoothness of the parking paths.

Simulation results are included for a garage parking example.

Section 2.2 addresses the smooth path generation of a truck and trailer vehicle (cf. [23]). It is shown how the fourth-order geometric continuity of the trailer path (continuity of the unit tangent vector, curvature, and first and second derivatives of curvature) is associated to the vehicle's smooth control inputs (velocity and steering of the truck). Then, taking into account the nonholonomic constraints of the articulated vehicle, the path generation can be performed by the introduction of the $\boldsymbol{\eta}^4$-spline. This is a ninth-order polynomial curve primitive that can interpolate given Cartesian points with associated arbitrary unit tangent vector, curvature, and first and second derivatives of curvature. The $\boldsymbol{\eta}^4$-spline depends on a set of eight (eta) parameters that can be freely chosen to change the path shape without changing the interpolations conditions at the path endpoints. Completeness, minimality, and symmetry of the $\boldsymbol{\eta}^4$-spline are established. An example on a parking maneuver of the articulated vehicle is presented and the pertinent optimal path planning is also discussed.

2.1 Multi-optimization of $\boldsymbol{\eta}^3$-splines for autonomous parking

This section proposes a multi-optimization approach to the autonomous parking of car-like vehicles. Focusing on the planning of motion maneuvers of car-like vehicles, the parking problem can be theoretically introduced as follows: given an initial configuration and a final configuration of the vehicle, find a path joining the initial and final configurations such that: 1) the path is collision-free, i.e. the vehicle on the path avoids any collision with all the obstacles of the parking scenario (other cars, walls, curbs, etc.); 2) the path is feasible (or admissible), i.e. the vehicle on the path satisfies the differential constraints of the vehicle model (the nonholonomic constraints) and the actuator constraints (such as e.g. the bound on the maximal steering angle of the front wheels).

The parking problem without differential and actuator constraints becomes the so-called piano mover's problem which is a classic problem in the motion planning literature (cf. the book [24] and the extensive references included). When the parking problem formulation is complete with both requirements 1) and 2), the approaches exposed in the literature are mainly based on a two-step procedure: First, a collision-free path that ignores differential (and actuator) constraints is determined. Then this path is suitably modified in order to accommodate to the constraints. In such a way, the first step just requires to pick up a solution technique for the piano mover's problem and in the second step *ad hoc* smoothing techniques or local steering methods are devised to accomplish a complete solution.

The two-step procedure was first proposed by Laumond *et al.* in [25] and subsequently several variants appeared [26–28] (also cf. [29] and references herein included).

The solution proposed in this section, first addresses the parking problem as a smooth feedforward control problem where the vehicle's sought control inputs, the linear velocity and the front wheel steering angle, are C^1-signals, i.e. continuous time functions admitting derivatives that are still continuous. Then, the introduction of the concept of third-order geometric continuity of Cartesian paths and the procedure of dynamic path inversion as exposed in [5] permits the feedforward control problem to be reduced to a purely geometric problem followed by a velocity planning problem. This geometric problem regards the search of a sequence of feasible paths connecting the initial vehicle configuration to the final one while satisfying all the the required constraints (obstacle avoidance, maximum steering angle, etc.). In this context, a path is feasible if it is a G^3-path, i.e. a path that has continuity, along the curve, of the unit-tangent vector, curvature, and derivative of the curvature with respect to the arc length (cf. subsection 2.1.1).

2.1.1 The smooth parking problem

We consider an autonomous parking problem for the car-like vehicle depicted in figure 2.1. The Cartesian coordinates of the rear-axle middle-point are denoted

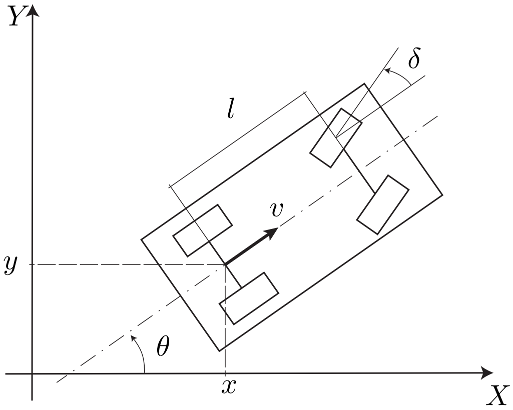


Figure 2.1: The car-like vehicle on the Cartesian plane.

by x, y and θ is the vehicle orientation angle with respect to the X axis. The distance between the rear-axle and the front-axle is l. With the usual modeling assumptions (no-slippage of the wheels, rigid body, etc.) the following nonlinear kinematic model of the car-like vehicle can be deduced:

$$\begin{cases} \dot{x}(t) = v(t)\,\cos\theta(t) \\ \dot{y}(t) = v(t)\,\sin\theta(t) \\ \dot{\theta}(t) = \frac{1}{l}\,v(t)\,\tan\delta(t)\,, \end{cases} \tag{2.1}$$

where the vehicle control inputs are $v(t)$ and $\delta(t)$, the velocity of the rear-axle middle-point and the steering angle of the front wheels respectively. Recall definition 7 of G^3-paths, that will be used along this chapter.

In order to obtain a smooth motion control, inputs v and δ must be functions with C^1 continuity, i.e. continuous functions with continuous first derivatives. A connection between smooth inputs and paths of the car-like vehicle is established by the following result.

Proposition 3 *Assign any $T > 0$. If a Cartesian path Γ is generated by the car-like vehicle described by system* (2.1)*, with inputs $v(t), \delta(t) \in C^1([0,T])$ where $v(t) \neq 0$ and $|\delta(t)| < \frac{\pi}{2}$ $\forall t \in [0,T]$, then Γ is a G^3-path. Conversely,* *given any G^3-path Γ there exist inputs $v(t), \delta(t) \in C^1([0,T])$ with $v(t) \neq 0$ and $|\delta(t)| < \frac{\pi}{2}$ $\forall t \in (0,T)$, and initial conditions such that the path generated by* (2.1) *coincides with the given* Γ.

Proof. It follows from an analogous result presented in [5] for unicycle mobile robots. □

Instrumentals to our approach to path planning for the autonomous parking of car-like vehicles are the following concepts of *configuration vector* and corresponding *configuration space.*

Definition 3 *The coordinate position (considering the middle-point of the rear-axle) and orientation of the vehicle with respect to the Cartesian plane $\{X, Y\}$ and the steering angle δ compose the* configuration vector *as follows:*

$$\mathbf{q} \doteq \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \doteq \begin{bmatrix} x \\ y \\ \theta \\ \delta \end{bmatrix} \in \mathcal{Q}\,, \tag{2.2}$$

where $\mathcal{Q} \doteq \mathbb{R}^2 \times [0, 2\pi[\, \times [-\delta_M,\, +\delta_M]$, is the configuration space*; herein δ_M is the maximum allowed value of the steering angle.*

In the parking scenario, the occupancy area of the car-like vehicle is denoted by $\mathcal{A}$ which is normally a rectangle moving in the Cartesian plane $\{X, Y\}$, referred as the *parking space* $\mathcal{P}$. The car body $\mathcal{A}$ occupies a portion area of $\mathcal{P}$ that depends on the configuration vector $\mathbf{q}$, i.e. $\mathcal{A} = \mathcal{A}(\mathbf{q}) \subset \mathcal{P}$. In the parking space there are also the obstacles $\mathcal{B}_i$, $i = 1, 2, \ldots n$, (see figure 2.2) considered as convex polygons without loss of generality. Recall that a non-convex polygon can be always decomposed in two or more convex polygons.

The parking problem can be introduced as a smooth feedforward control problem for model (2.1), i.e. the problem of devising inputs $v(t), \delta(t) \in C^1$, for which the vehicle starting from a given configuration $\mathbf{q}_s = [x_s\, y_s\, \theta_s\, \delta_s]'$ reaches an assigned final or *goal* configuration $\mathbf{q}_g = [x_g\, y_g\, \theta_g\, \delta_g]'$ while avoiding all the obstacles and satisfying at any time the constraint $|\delta(t)| \leq \delta_M$. The sought

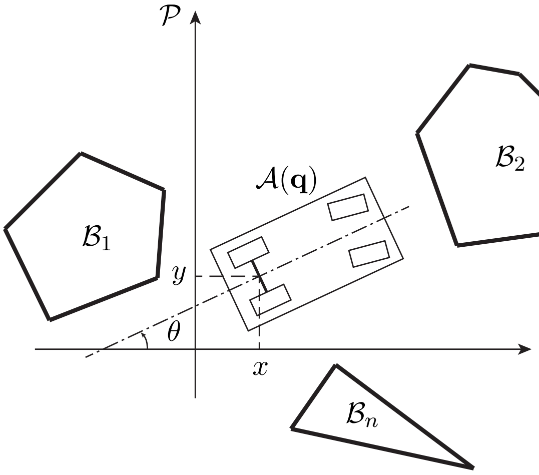


Figure 2.2: Parking space $\mathcal{P}$ with car $\mathcal{A}(\mathbf{q})$ and obstacles $\mathcal{B}_i$, $i = 1, \dots, n$.

feedforward control may admit maneuvers, i.e. changes of sign in the vehicle velocity $v(t)$, so that when the velocity is positive the car performs a forward movement whereas when it is negative we have a car's backward movement.

On the grounds of proposition 3 and of the (dynamic) path inversion concept [5] introduced in the precedent chapter, the smooth parking feedforward control problem can be reduced to a purely geometric problem, to be more specific a purely Cartesian G^3-path planning problem followed by a velocity planning on the determined paths. This means determining a sequence of (feasible) G^3-paths $\{\Gamma_1, \Gamma_2, \dots \Gamma_h\}$ (h is the number of parking paths) that the vehicle can exactly follow by applying feedforward inputs $v(t), \delta(t)$ where $v(t) \in C^1$ can be freely designed with the constraint of having zero velocity and zero acceleration at the the start and at the end of each path Γ_i. The steering input on the path Γ_i can be simply determined by (cf. [5] and [30])

$$\delta(t) = \pm \arctan(l\kappa_i(s))|_{s=\int_{t_i}^{t} v(\xi)d\xi} ,$$

for a forward (+) or backward (-) movement. Herein $\kappa_i(s)$ is the curvature function of arc length s and t_i is the time instant at the beginning of Γ_i.

In the following, a path Γ to be followed by the vehicle with a forward or backward movement will be denoted by Γ^+ or Γ^- respectively. Therefore, a sequence of paths $\{\Gamma_1, \Gamma_2, \dots \Gamma_h\}$ is actually $\{\Gamma_1^+, \Gamma_2^-, \dots \Gamma_h^+\}$ or $\{\Gamma_1^-, \Gamma_2^+, \dots \Gamma_h^-\}$ if h is odd, and $\{\Gamma_1^+, \Gamma_2^-, \dots \Gamma_h^-\}$ or $\{\Gamma_1^-, \Gamma_2^+, \dots \Gamma_h^+\}$ if h is even. In the introduced sequence of paths we see an alternation of forward and backward paths, i.e. a forward path Γ_i^+ is followed by a backward Γ_{i+1}^- or viceversa. Any pair of subsequent paths $\{\Gamma_i^+,\Gamma_{i+1}^-\}$ or $\{\Gamma_i^-,\Gamma_{i+1}^+\}$ is made of paths that meet each other at a common Cartesian point corresponding to a configuration vector $\mathbf{q}_i$ $(i = 1, \dots h-1)$ which is still common for the vehicle at the end of path Γ_i and at the start of Γ_{i+1} in case of no steering at standstill, i.e. the case when $\dot{\delta}(t) = 0$ if $v(t) = 0$.

When the vehicle parking problem can be solved without maneuvers we have just one G^3-path Γ_1^+ or Γ_1^- $(h = 1)$ to determine and optimize (see figure 2.3). If no solution can be found with one path because of the obstacles

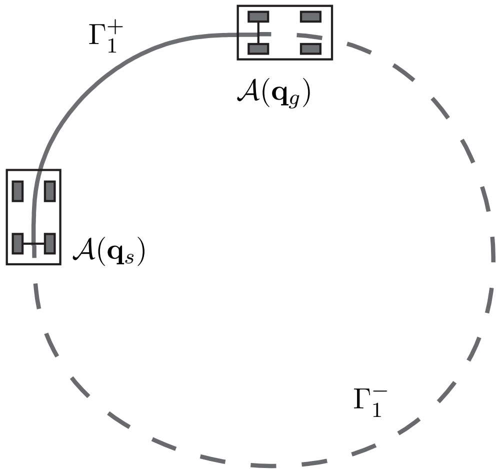


Figure 2.3: The vehicle from $\mathbf{q}_s$ to $\mathbf{q}_g$ with forward path Γ_1^+ or backward Γ_1^-.

and the limitation given by the maximum steering angle δ_M, a solution may be sought with two chained paths $\{\Gamma_1^+, \Gamma_2^-\}$ or $\{\Gamma_1^-, \Gamma_2^+\}$ $(h = 2)$. In this case there is one motion inversion of the vehicle or, in other words, one maneuver to complete the parking task. On the parking space, Γ_1 and Γ_2 meet at a cusp point whose Cartesian coordinates are given by the first two components of configuration vector $\mathbf{q}_1$. In figure 2.4, the case of two maneuvers ($h = 2$) is depicted. When also with $h = 2$ no solution is found we can try with more paths. Figure 2.5 shows the case of three maneuvers $h = 3$.

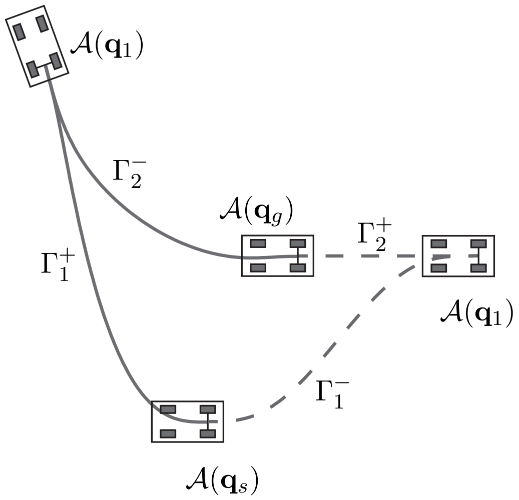

$\mathcal{A}(\mathbf{q}_1)$
Γ_2^-
$\mathcal{A}(\mathbf{q}_g)$
Γ_2^+
Γ_1^+
$\mathcal{A}(\mathbf{q}_1)$
Γ_1^-
$\mathcal{A}(\mathbf{q}_s)$

Figure 2.4: The two-paths sequences $\{\Gamma_1^+, \Gamma_2^-\}$ and $\{\Gamma_1^-, \Gamma_2^+\}$ for the parking planning.

The G^3-paths Γ_i, $i = 1, \ldots, h$ composing the sequence $\{\Gamma_1, \Gamma_2, \ldots \Gamma_h\}$ must satisfy specific interpolation conditions at the endpoints of each Γ_i (cf. subsection 2.1.2) in order to guarantee the overall feasibility of the planned paths. In particular considering that the vehicle starts at the given configuration $\mathbf{q}_s = [x_s\, y_s\, \theta_s\, \delta_s]'$ it follows that the starting point of Γ_1 satisfies:

- Cartesian coordinates are $(x_s\, y_s)$;

- direction angle of the unit-tangent vector is θ_s;

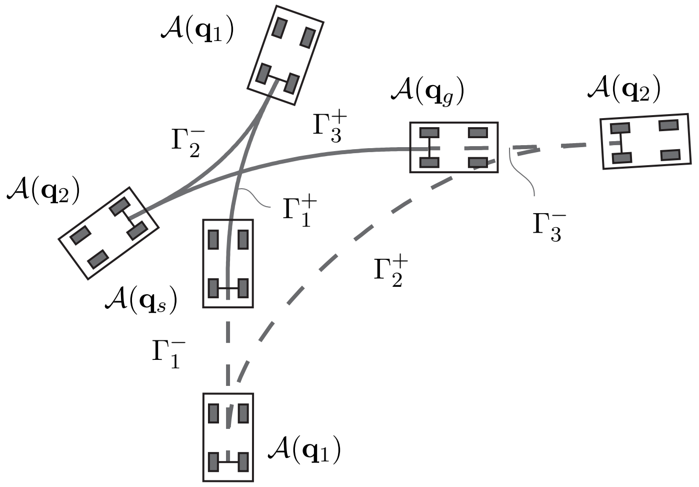

$\mathcal{A}(\mathbf{q}_1)$
$\mathcal{A}(\mathbf{q}_g)$
$\mathcal{A}(\mathbf{q}_2)$
Γ_2^-
Γ_3^+
$\mathcal{A}(\mathbf{q}_2)$
Γ_1^+
Γ_3^-
Γ_2^+
$\mathcal{A}(\mathbf{q}_s)$
Γ_1^-
$\mathcal{A}(\mathbf{q}_1)$

Figure 2.5: The three-paths sequences $\{\Gamma_1^+, \Gamma_2^-, \Gamma_3^+\}$ and $\{\Gamma_1^-, \Gamma_2^+, \Gamma_3^-\}$ for the parking planning.

- scalar curvature κ_s is given by (cf. [5, 31])

$$\kappa_s = \begin{cases} \frac{1}{l}\tan\delta_s & \text{if } \Gamma_1 = \Gamma_1^+ \\ -\frac{1}{l}\tan\delta_s & \text{if } \Gamma_1 = \Gamma_1^- \, ; \end{cases} \tag{2.3}$$

- the derivative of the scalar curvature with respect to the arc length, $\dot{\kappa}_s$ can be freely chosen.

Analogously, the vehicle arrives finally at the goal configuration $\mathbf{q}_g = [x_g \, y_g \, \theta_g \, \delta_g]'$ for which the endpoint of Γ_h satisfies:

- Cartesian coordinates are $(x_g \, y_g)$;

- direction angle of the unit-tangent vector is θ_g;

- scalar curvature κ_g is given by

$$\kappa_g = \begin{cases} \frac{1}{l}\tan\delta_g & \text{if } \Gamma_h = \Gamma_h^+ \\ -\frac{1}{l}\tan\delta_g & \text{if } \Gamma_h = \Gamma_h^- \, ; \end{cases} \tag{2.4}$$

- the derivative of the scalar curvature with respect to the arc length, $\dot{\kappa}_g$ is a free parameter of the planning problem.

The smooth parking problem considered in this paper can be introduced as follows.

Problem 1 (Multi-optimization of a sequence of G^3-paths for the smooth parking problem) *Given the number h of paths, consider the space $\mathcal{F}_h$ of all the sequences of G^3-paths $\{\Gamma_1^+, \Gamma_2^-, \dots \Gamma_h\}$ (or $\{\Gamma_1^-, \Gamma_2^+, \dots \Gamma_h\}$) such that this sequence:*

a) *is feasible as a whole, i.e. there exist feedforward controls $v(t), \delta(t) \in C^1$ for which the vehicle of model* (2.1) *follows the path sequence exactly, and*

b) *connects the given initial configuration $\mathbf{q}_s$ to the final configuration $\mathbf{q}_g$.*

Find the path sequence in $\mathcal{F}_h$ that minimizes the indexes

- *the maximum value of the absolute curvature on the h paths,*
- *the maximum value of the absolute curvature derivative on the h paths, and*
- *the total length of the h paths $\Gamma_1, \Gamma_2, \dots \Gamma_h$*

subject to the following constraints

1) *avoidance of all the obstacles $\mathcal{B}_1, \mathcal{B}_2 \dots \mathcal{B}_n$ along the paths $\Gamma_1, \Gamma_2, \dots \Gamma_h$;*

2) *{maximum value of the absolute curvature on the h paths} $\leq \kappa_M$;*

3) *{maximum value of the absolute curvature derivative on the h paths}$\leq \dot{\kappa}_M$;*

4) *avoidance of steering at standstill;*

where $\kappa_M = \frac{1}{l} \tan \delta_M$ and $\dot{\kappa}_M$ is a freely chosen bound for the absolute value of the curvature derivative.

Remark It is worth noting the differences among the constraints of problem 1. Constraints 1) and 2) are *hard* constraints related to obstacle avoidance and maximal steering angle (which is a vehicle's mechanical constraint) respectively, whereas constraints 3) and 4) are *soft* constraints related to path smoothness and parking modality respectively. In particular, if steering at standstill is admitted, the fourth constraint, which is considered in this exposition, can be removed without changing the proposed overall approach to the parking problem.

The constrained multi-optimization of problem 1 is a search in the infinite-dimensional space $\mathcal{F}_h$. In the next subsection, an approximation scheme based on $\boldsymbol{\eta}^3$-splines will make possible to reduce the search into a finite-dimensional space for which standard parameter optimization can be used.

2.1.2 Shaping paths sequence with $\boldsymbol{\eta}^3$-splines

The $\boldsymbol{\eta}^3$-splines (cf. in [32]) are an effective tool to approximate Cartesian paths with third-order geometric continuity. Indeed, they can interpolate a sequence of Cartesian points over which unit-tangent vector, curvature, and curvature derivative can be arbitrarily assigned. A single $\boldsymbol{\eta}^3$-spline is a seventh-order polynomial curve

$$\mathbf{p}(u;\boldsymbol{\eta}) = [p_x(u)\, p_y(u)]' , \quad u \in [0,1] , \tag{2.5}$$

$$p_x(u) = \sum_{i=0}^{7} \alpha_i u^i , \qquad p_y(u) = \sum_{i=0}^{7} \beta_i u^i , \tag{2.6}$$

that depends on a six-dimensional vector $\boldsymbol{\eta}$ (the *eta* parameter vector) and interpolates the data vectors $\mathbf{c}_a = [x_a\, y_a\, \theta_a\, \kappa_a\, \dot{\kappa}_a]'$ and $\mathbf{c}_b = [x_b\, y_b\, \theta_b\, \kappa_b\, \dot{\kappa}_b]'$, at the curve endpoints $\mathbf{p}(0;\boldsymbol{\eta})$ and $\mathbf{p}(1;\boldsymbol{\eta})$ respectively: $(x_a\, y_a)$ and $(x_b\, y_b)$ are the Cartesian coordinates of the endpoints, θ_a and θ_b are the direction angles of the unit-tangent vectors, κ_a and κ_b are the scalar curvatures, and $\dot{\kappa}_a$ and $\dot{\kappa}_b$ are the derivatives of the scalar curvatures with respect to the arc length. The $\boldsymbol{\eta}$ is a free vector in $\mathbb{R}^2_+ \times \mathbb{R}^4$ that can be used to shape the resulting path while maintaining the interpolation conditions at the endpoints. The complete closed-form expressions of the $\boldsymbol{\eta}^3$-spline are reported in [32] and [33].

Here, we propose to use a simplified version of the $\boldsymbol{\eta}^3$-spline that only depends on the first two components of vector $\boldsymbol{\eta}$ (actually the most important ones, cf. section V of [32]) while the remaining components are set to zero. Specifically, in this case $\boldsymbol{\eta}$ is redefined as the two-dimensional vector $[\eta_a \, \eta_b]' \in \mathbb{R}^2_+$ where its positive components are the mathematical velocities of the curve at the endpoints, i.e. $\eta_a = \|\dot{\mathbf{p}}(0;\boldsymbol{\eta})\|$ and $\eta_b = \|\dot{\mathbf{p}}(1;\boldsymbol{\eta})\|$. The corresponding simplified closed-form expressions of coefficients α_i, β_i, $i = 0, 1, \ldots, 7$, appearing in (2.5) and (2.6) are detailed below:

$$
\begin{aligned}
&\alpha_0 = x_a, \quad \alpha_1 = \eta_a \cos\theta_a, \\
&\alpha_2 = -\frac{1}{2}\eta_a^2 \kappa_a \sin\theta_a, \quad \alpha_3 = -\frac{1}{6}\eta_a^3 \dot{\kappa}_a \sin\theta_a, \\
&\alpha_4 = 35\,(x_b - x_a) - 20\eta_a \cos\theta_a + \left(5\kappa_a + \frac{2}{3}\eta_a \dot{\kappa}_a\right)\eta_a^2 \sin\theta_a - 15\eta_b \cos\theta_b \\
&\quad - \left(\frac{5}{2}\kappa_b - \frac{1}{6}\eta_b \dot{\kappa}_b\right)\eta_b^2 \sin\theta_b, \\
&\alpha_5 = -84(x_b - x_a) + 45\eta_a \cos\theta_a - (10\kappa_a + \eta_a \dot{\kappa}_a)\eta_a^2 \sin\theta_a + 39\eta_b \cos\theta_b \\
&\quad + \left(7\kappa_b - \frac{1}{2}\eta_b \dot{\kappa}_b\right)\eta_b^2 \sin\theta_b, \\
&\alpha_6 = 70(x_b - x_a) - 36\eta_a \cos\theta_a + \left(\frac{15}{2}\kappa_a + \frac{2}{3}\eta_a \dot{\kappa}_a\right)\eta_a^2 \sin\theta_a - 34\eta_b \cos\theta_b \\
&\quad - \left(\frac{13}{2}\kappa_b - \frac{1}{2}\eta_b \dot{\kappa}_b\right)\eta_b^2 \sin\theta_b, \\
&\alpha_7 = -20(x_b - x_a) + 10\eta_a \cos\theta_a - \left(2\kappa_a + \frac{1}{6}\eta_a \dot{\kappa}_a\right)\eta_a^2 \sin\theta_a + 10\eta_b \cos\theta_b \\
&\quad + \left(2\kappa_b - \frac{1}{6}\eta_b \dot{\kappa}_b\right)\eta_b^2 \sin\theta_b, \\
&\beta_0 = y_a, \quad \beta_1 = \eta_a \sin\theta_a, \\
&\beta_2 = \frac{1}{2}\eta_a^2 \kappa_a \cos\theta_a, \quad \beta_3 = \frac{1}{6}\eta_a^3 \dot{\kappa}_a \cos\theta_a, \\
&\beta_4 = 35\,(y_b - y_a) - 20\eta_a \sin\theta_a - \left(5\kappa_a + \frac{2}{3}\eta_a \dot{\kappa}_a\right)\eta_a^2 \cos\theta_a - 15\eta_b \sin\theta_b
\end{aligned}
$$

$$
\begin{aligned}
&+\left(\frac{5}{2}\kappa_b-\frac{1}{6}\eta_b\dot\kappa_b\right)\eta_b^2\cos\theta_b,\\
\beta_5 &= -84(y_b-y_a)+45\eta_a\sin\theta_a+(10\kappa_a+\eta_a\dot\kappa_a)\eta_a^2\cos\theta_a+39\eta_b\sin\theta_b\\
&-\left(7\kappa_b-\frac{1}{2}\eta_b\dot\kappa_b\right)\eta_b^2\cos\theta_b,\\
\beta_6 &= 70(y_b-y_a)-36\eta_a\sin\theta_a-\left(\frac{15}{2}\kappa_a+\frac{2}{3}\eta_a\dot\kappa_a\right)\eta_a^2\cos\theta_a-34\eta_b\sin\theta_b\\
&+\left(\frac{13}{2}\kappa_b-\frac{1}{2}\eta_b\dot\kappa_b\right)\eta_b^2\cos\theta_b,\\
\beta_7 &= -20(y_b-y_a)+10\eta_a\sin\theta_a+\left(2\kappa_a+\frac{1}{6}\eta_a\dot\kappa_a\right)\eta_a^2\cos\theta_a+10\eta_b\sin\theta_b\\
&-\left(2\kappa_b-\frac{1}{6}\eta_b\dot\kappa_b\right)\eta_b^2\cos\theta_b\,.
\end{aligned}
$$

The infinite-dimensional space $\mathcal{F}_h$ of problem 1 can be approximated with a finite-dimensional space by using $\boldsymbol{\eta}^3$-splines. Consider an element of $\mathcal{F}_h$, i.e. a sequence of G^3-paths $\{\Gamma_1^+,\Gamma_2^-,\ldots,\Gamma_h\}$ (or $\{\Gamma_1^-,\Gamma_2^+,\ldots,\Gamma_h\}$), then each Γ_i^+ or Γ_i^-, will be approximated by a single (simplified) $\boldsymbol{\eta}^3$-spline denoted as $\mathbf{p}_i^+(u;\boldsymbol{\eta}_i)$ or $\mathbf{p}_i^-(u;\boldsymbol{\eta}_i)$. Hence, the sequence of $\boldsymbol{\eta}^3$-splines

$$\{\mathbf{p}_1^+(u;\boldsymbol{\eta}_1),\mathbf{p}_2^-(u;\boldsymbol{\eta}_2),\ldots,\mathbf{p}_h(u;\boldsymbol{\eta}_h)\}\,,$$

or

$$\{\mathbf{p}_1^-(u;\boldsymbol{\eta}_1),\mathbf{p}_2^+(u;\boldsymbol{\eta}_2),\ldots,\mathbf{p}_h(u;\boldsymbol{\eta}_h)\}\,,$$

will be used to set up the multi-optimization for the parking path planning.

The simplified spline $\mathbf{p}_i(u;\boldsymbol{\eta}_i)$ is defined by the interpolating conditions $\mathbf{c}_{a,i}=[x_{a,i}\,y_{a,i}\,\theta_{a,i}\,\kappa_{a,i}\,\dot\kappa_{a,i}]'$ and $\mathbf{c}_{b,i}=[x_{b,i}\,y_{b,i}\,\theta_{b,i}\,\kappa_{b,i}\,\dot\kappa_{b,i}]'$ at the path endpoints and by the parameter vector $\boldsymbol{\eta}_i=[\eta_{a,i}\,\eta_{b,i}]'$.

Remark In the proposed approximating scheme, a path Γ_i is actually approximated by $\mathbf{p}_i\left([0,1];\boldsymbol{\eta}_i\right)$, i.e. the Cartesian image over interval $[0,1]$ of the $\boldsymbol{\eta}^3$-spline curve $\mathbf{p}_i\left(u;\boldsymbol{\eta}_i\right)$. In the following, to simplify notation the same symbol $\mathbf{p}_i\left(u;\boldsymbol{\eta}_i\right)$ or even $\mathbf{p}_i$ is used to denote both the parametric curve and the corresponding path.

The parking sequence of $\boldsymbol{\eta}^3$-splines $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_h\}$ can satisfies the conditions a) and b) and the constraint 4) of problem 1 by a proper assignment of the interpolation conditions. These assignments are exemplified below for the cases $h = 1$, 2.

Case $h = 1$ with $\{\mathbf{p}_1^+(u; \boldsymbol{\eta}_1)\}$ (one forward movement of the vehicle): The vehicle starts at configuration $\mathbf{q}_s$ and arrives at configuration $\mathbf{q}_g$ (cf. (2.3) and (2.4)). Hence, the spline parameters can be set as follows:

$$\mathbf{p}_1^+(u; \boldsymbol{\eta}_1) : \begin{cases} \mathbf{c}_{a,1} = \begin{bmatrix} x_s \\ y_s \\ \theta_s \\ \frac{1}{l} \tan \delta_s \\ z_1 \end{bmatrix}, \; \mathbf{c}_{b,1} = \begin{bmatrix} x_g \\ y_g \\ \theta_g \\ \frac{1}{l} \tan \delta_g \\ z_2 \end{bmatrix}, \\ \boldsymbol{\eta}_1 = [z_3 \, z_4]', \end{cases}$$

where $z_1, z_2 \in [-\dot{\kappa}_M, \dot{\kappa}_M]$ and $z_3, z_4 \in \mathbb{R}_+$ indicate the free variables to be optimized. These are packed in the vector $\mathbf{z} = [z_1 \, z_2 \, z_3 \, z_4]'$ that belongs to the search space $\mathcal{Z} := [-\dot{\kappa}_M, \dot{\kappa}_M]^2 \times \mathbb{R}_+^2$.

Case $h = 1$ with $\{\mathbf{p}_1^-(u; \boldsymbol{\eta}_1)\}$ (one backward movement of the vehicle): The spline parameters can be set as follows:

$$\mathbf{p}_1^-(u; \boldsymbol{\eta}_1) : \begin{cases} \mathbf{c}_{a,1} = \begin{bmatrix} x_s \\ y_s \\ \theta_s + \pi \\ -\frac{1}{l} \tan \delta_s \\ z_1 \end{bmatrix}, \; \mathbf{c}_{b,1} = \begin{bmatrix} x_g \\ y_g \\ \theta_g + \pi \\ -\frac{1}{l} \tan \delta_g \\ z_2 \end{bmatrix}, \\ \boldsymbol{\eta}_1 = [z_3 \, z_4]', \end{cases}$$

where $\mathbf{z} = [z_1 \, z_2 \, z_3 \, z_4]' \in \mathcal{Z} = [-\dot{\kappa}_M, \dot{\kappa}_M]^2 \times \mathbb{R}_+^2$.

Case $h = 2$ with $\{\mathbf{p}_1^+(u; \boldsymbol{\eta}_1), \mathbf{p}_2^-(u; \boldsymbol{\eta}_2)\}$ (one forward movement plus a backward one): All the spline parameters can be set as follows

$$\mathbf{p}_1^+(u;\boldsymbol{\eta}_1) : \begin{cases} \mathbf{c}_{a,1} = \begin{bmatrix} x_s \\ y_s \\ \theta_s \\ \frac{1}{l}\tan\delta_s \\ z_1 \end{bmatrix}, \; \mathbf{c}_{b,1} = \begin{bmatrix} z_9 \\ z_{10} \\ z_{11} \\ z_{12} \\ z_2 \end{bmatrix}, \\ \boldsymbol{\eta}_1 = [z_5 \, z_6]', \end{cases}$$

$$\mathbf{p}_2^-(u;\boldsymbol{\eta}_2) : \begin{cases} \mathbf{c}_{a,2} = \begin{bmatrix} z_9 \\ z_{10} \\ z_{11} + \pi \\ -z_{12} \\ z_3 \end{bmatrix}, \; \mathbf{c}_{b,2} = \begin{bmatrix} x_g \\ y_g \\ \theta_g + \pi \\ -\frac{1}{l}\tan\delta_g \\ z_4 \end{bmatrix}, \\ \boldsymbol{\eta}_2 = [z_7 \, z_8]', \end{cases}$$

where the free variables are z_i, $i = 1, \dots 12$, and they form the vector $\mathbf{z} \in \mathcal{Z}$ with $\mathcal{Z} := [-\dot{\kappa}_M, \dot{\kappa}_M]^4 \times \mathbb{R}_+^4 \times \mathbb{R}^2 \times [0, 2\pi) \times [-\kappa_M, \kappa_M]$ which is a twelve-dimensional search space.

Case $h = 2$ with $\{\mathbf{p}_1^-(u;\boldsymbol{\eta}_1), \mathbf{p}_2^+(u;\boldsymbol{\eta}_2)\}$ (one backward movement plus a a forward one): similarly to the previous case, all the parameters can be set as follows

$$\mathbf{p}_1^-(u;\boldsymbol{\eta}_1) : \begin{cases} \mathbf{c}_{a,1} = \begin{bmatrix} x_s \\ y_s \\ \theta_s + \pi \\ -\frac{1}{l}\tan\delta_s \\ z_1 \end{bmatrix}, \; \mathbf{c}_{b,1} = \begin{bmatrix} z_9 \\ z_{10} \\ z_{11} \\ z_{12} \\ z_2 \end{bmatrix}, \\ \boldsymbol{\eta}_1 = [z_5 \, z_6]', \end{cases}$$

$$\mathbf{p}_2^+(u;\boldsymbol{\eta}_2) : \left\{ \begin{array}{c} \mathbf{c}_{a,2} = \begin{bmatrix} z_9 \\ z_{10} \\ z_{11}+\pi \\ -z_{12} \\ z_3 \end{bmatrix}, \; \mathbf{c}_{b,2} = \begin{bmatrix} x_g \\ y_g \\ \theta_g \\ \frac{1}{l}\tan\delta_g \\ z_4 \end{bmatrix}, \\ \boldsymbol{\eta}_2 = [z_7 \, z_8]'. \end{array} \right.$$

When $h > 2$, the spline parameters can be set up similarly as in the presented cases. Table 2.1 reports the dimension and structure of the search space $\mathcal{Z}$ as a function of h. In particular, when the parking is done with h splines, the dimension of the search space is $8h - 4$: every added spline increases of 8 the dimension of $\mathcal{Z}$.

Remark The proposed approximation scheme replaces each path Γ_i of sequence $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_h\}$ with only one $\boldsymbol{\eta}^3$-spline to avoid excessive increasing of the dimension of the search space $\mathcal{Z}$. Yet, it would be possible within the same proposed framework to improve the approximation by using two or more $\boldsymbol{\eta}^3$-splines for each Γ_i.

2.1.3 Setting up the multi-optimization

In this section the multi-optimization of problem 1 is dealt with the substitution of the infinite-dimensional space $\mathcal{F}_h$ with the finite-dimensional parameter space $\mathcal{Z}$ introduced in the previous section. This corresponds to do the searching for multi-optimization on the sequences of simplified $\boldsymbol{\eta}^3$-splines $\{\mathbf{p}_1(u;\boldsymbol{\eta}_1), \mathbf{p}_2(u;\boldsymbol{\eta}_2), \ldots, \mathbf{p}_h(u;\boldsymbol{\eta}_h)\}$ instead of the sequences of G^3-paths introduced in subsection 2.1.2.

The three indexes to be minimized using the standard weighted sum method [34] are (cf. problem 1): the maximum value of the curvature modulus on the h splines, the maximum value of the absolute value of the curvature derivative (with respect to the arc length) on the h splines, and the total length of the h splines. These indexes are respectively denoted by κ_{max}, $\dot{\kappa}_{max}$, and s_{tot} and depend on the parameter vector $\mathbf{z} \in \mathcal{Z}$. They can be determined as follows (the

h	dim($\mathcal{Z}$)	$\mathcal{Z}$
1	4	$[-\dot{\kappa}_M, \dot{\kappa}_M]^2 \times \mathbb{R}_+^2$
2	12	$[-\dot{\kappa}_M, \dot{\kappa}_M]^4 \times \mathbb{R}_+^4 \times \mathbb{R}^2 \times$ $[0, 2\pi) \times [-\kappa_M, \kappa_M]$
3	20	$[-\dot{\kappa}_M, \dot{\kappa}_M]^6 \times \mathbb{R}_+^6 \times \mathbb{R}^4 \times$ $[0, 2\pi)^2 \times [-\kappa_M, \kappa_M]^2$
$\vdots$	$\vdots$	$\vdots$
h	$8h-4$	$[-\dot{\kappa}_M, \dot{\kappa}_M]^{2h} \times \mathbb{R}_+^{2h} \times \mathbb{R}^{2(h-1)} \times$ $[0, 2\pi)^{h-1} \times [-\kappa_M, \kappa_M]^{h-1}$

Table 2.1: Dimension and structure of the search space $\mathcal{Z}$.

dependencies on $\mathbf{z}$ are omitted for simplicity and $\mathbf{p}_i(u;\boldsymbol{\eta}_i) \equiv [p_{x,i}(u)\, p_{y,i}(u)]'$, $i = 1, \ldots, h$, cf. (2.5)):

$$\kappa_{max} \doteq \max_{i=1,\ldots h} \kappa_{max,i}\,, \tag{2.7}$$

where $(i = 1, \ldots, h)$

$$\kappa_{max,i} \doteq \max_{u\in[0,1]} |\kappa_i(u)|\;,$$

and

$$\kappa_i(u) = \frac{\dot{p}_{x,i}(u)\ddot{p}_{y,i}(u) - \ddot{p}_{x,i}(u)\dot{p}_{y,i}(u)}{(\dot{p}_{x,i}^2(u) + \dot{p}_{y,i}^2(u))^{\frac{3}{2}}}\,,$$

is the scalar curvature of spline $\mathbf{p}_i(u;\boldsymbol{\eta}_i)$;

$$\dot{\kappa}_{max} \doteq \max_{i=1,\ldots h} \dot{\kappa}_{max,i}\,, \tag{2.8}$$

where $(i = 1, \ldots, h)$

$$\dot{\kappa}_{max,i} \doteq \max_{u\in[0,1]} \left|\frac{d\kappa_i}{ds}(u)\right|\;,$$

and

$$\frac{d\kappa_i}{ds}(u) = \frac{\dot{p}_{x,i}\dddot{p}_{y,i} - \dddot{p}_{x,i}\dot{p}_{y,i}}{(\dot{p}_{x,i}^2 + \dot{p}_{y,i}^2)^2} - 3\frac{(\dot{p}_{x,i}\ddot{p}_{y,i} - \ddot{p}_{x,i}\dot{p}_{y,i})(\dot{p}_{x,i}\ddot{p}_{x,i} + \dot{p}_{y,i}\ddot{p}_{y,i})}{(\dot{p}_{x,i}^2 + \dot{p}_{y,i}^2)^3},$$

is the derivative of the curvature of spline $\mathbf{p}_i(u;\boldsymbol{\eta}_i)$ with respect to the arc length (for brevity the dependency on u is omitted in the right side of the above relation);

$$s_{tot} \doteq \sum_{i}^{h} s_{tot,i}\,, \tag{2.9}$$

where

$$s_{tot,i} \doteq \int_0^1 [\dot{p}_{x,i}^2(\xi) + \dot{p}_{y,i}^2(\xi)]^{1/2} d\xi\,.$$

The constraint of obstacle avoidance is dealt with the concept of *occupancy span* of the vehicle along a path planning:

Definition 4 *The occupancy span of the vehicle along the spline sequence* $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_h\}$ *is the set defined as*

$$\mathcal{S} \doteq \bigcup_{i=1}^{n} \mathcal{S}_i\,,$$

where

$$\begin{aligned}\mathcal{S}_i \doteq \{\boldsymbol{p} \in \mathcal{P} :\, \boldsymbol{p} \in \mathcal{A}(\boldsymbol{q})\,, q_1 = p_{x,i}(u),\, q_2 = p_{y,i}(u),\\ q_3 = \arg(\dot{p}_{x,i}(u) + j\dot{p}_{y,i}(u))\,, u \in [0,1]\}\ .\end{aligned}$$

Note that the occupancy span depends on $\mathbf{z} \in \mathcal{Z}$, i.e. $\mathcal{S} \equiv \mathcal{S}(\mathbf{z})$. Define the *obstacle region* $\mathcal{O}$ as the union of all the obstacles, i.e. $\mathcal{O} \doteq \cup_{i=1}^{n}\mathcal{B}_i$ and the vehicle avoids all the obstacles along a path planning if and only if the intersection of $\mathcal{S}(\mathbf{z})$ and $\mathcal{O}$ is the empty set (cf. constraint (2.11) below).

Now the nonlinear constrained multiobjective optimization problem for the geometric planning of autonomous parking can be stated as follows:

Problem 2 (Multi-optimization of a sequence of $\boldsymbol{\eta}^3$-splines for the smooth parking problem) *Given the number h of paths, consider the parameter space* $\mathcal{Z}$ *that defines the sequences* $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \ldots, \mathbf{p}_h\}$ *(or* $\{\mathbf{p}_1^-, \mathbf{p}_2^+, \ldots, \mathbf{p}_h\}$*)*

according to the interpolating scheme exposed in section 2.1.3. *Then, the posed problem is* ($\lambda_1, \lambda_2, \lambda_3 \geq 0$ *and* $\lambda_1 + \lambda_2 + \lambda_3 = 1$)*:*

$$\min_{\mathbf{z}\in\mathcal{Z}} \lambda_1 \kappa_{max}(\mathbf{z}) + \lambda_2 \dot{\kappa}_{max}(\mathbf{z}) + \lambda_3 s_{tot}(\mathbf{z}) \,, \tag{2.10}$$

subject to

$$\mathcal{S}(\mathbf{z}) \cap \mathcal{O} = \varnothing \,, \tag{2.11}$$

$$\kappa_{max}(\mathbf{z}) \leq \kappa_M \,, \tag{2.12}$$

$$\dot{\kappa}_{max}(\mathbf{z}) \leq \dot{\kappa}_M \,. \tag{2.13}$$

The coefficients λ_1, λ_2, and λ_3 of the composite index to be minimized in (2.10) can be freely chosen in order to weight the smoothness of the resulting maneuver paths (which is related to low values of both κ_{max} and $\dot{\kappa}_{max}$) versus the minimization of s_{tot}, the total length of the parking paths.

Remark Note that the possible constraint of avoiding steering at vehicle's standstill does not appear in the constraints (2.12)-(2.13) because it is plainly enforced by proper assignment of the geometric interpolating conditions on the $\boldsymbol{\eta}^3$-splines.

Obstacle avoidance constraint (2.11) can be equivalently reduced to an equality constraint by computing the *maximal collision area* of the vehicle along the spline sequence:

$$\text{mca} \doteq \max_{i=1,\ldots,h} \text{mca}_i \,, \tag{2.14}$$

$$\begin{aligned} \text{mca}_i \doteq \max_{u\in[0,1]} \{ & \text{area}\,(\mathcal{A}(\boldsymbol{q}) \cap \mathcal{O}) \,:\, q_1 = p_{x,i}(u), \\ & q_2 = p_{y,i}(u),\, q_3 = \arg(\dot{p}_{x,i}(u) + j\dot{p}_{y,i}(u))\} \;. \end{aligned}$$

Constraint (2.11) is therefore equivalent to

$$\text{mca}(\boldsymbol{z}) = 0 \,,$$

and in such a way problem 2 becomes a constrained minimization problem for which a standard penalty method [35] can take into account all the constraints so as to reduce the whole multi-optimization to the minimization of just one index. In a real-time scenario for autonomous parking, fast local minimization algorithms can be then implemented to solve problem 2 provided that the following data is readily available: (1) the number h of splines; (2) the maneuver sequence to prefer $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \ldots, \mathbf{p}_h\}$ or $\{\mathbf{p}_1^-, \mathbf{p}_2^+, \ldots, \mathbf{p}_h\}$; (3) an initial estimate of the parameter vector $\boldsymbol{z}$. Reasonably, this data can be determined by using look-up tables that can be constructed off-line by extensive optimizations such as those based on methods of stochastic global multi-objective optimization [36].

2.1.4 Simulation results

Example 1: Firstly, an example of *garage parking* maneuver in a constrained environment is considered for a standard compact car with wheelbase and maximum steering angle of the front wheels $l = 2.3$ m and $\delta_M = 0.464$ rad. Hence, the maximum curvature of the car paths is $\kappa_M = \frac{1}{l}\tan\delta_M = 0.218$ m^{-1}. The allowed maximum absolute value of the curvature derivative with respect to the arc length is chosen as $\dot{\kappa}_M = 2.50$ m^{-2}. The origin of the Cartesian plane $\mathcal{P}$ is chosen to be inside the parking lot that the car has to reach. The car has start configuration

$$\mathbf{q}_s = [x_s\; y_s\; \theta_s\; \delta_s]' = [7\;\; -6\;\; 3\pi/4\;\; 0]'\,,$$

and the final goal configuration, which corresponds to a *front car parking* mode (i.e. the car can only reach the goal configuration with a forward final motion because of the surrounding obstacles (cf. figure 2.6), is

$$\mathbf{q}_g = [x_g\; y_g\; \theta_g\; \delta_g] = [0.7\; 0\; \pi\; 0]'\,.$$

The multi-optimizations for solving this parking problem are set up with weights $\lambda_1 = 0.5$, $\lambda_2 = 0.2$, and $\lambda_3 = 0.3$. All the possible spline sequences to be considered up to three splines are the following (the arguments of the

$\boldsymbol{\eta}^3$-splines are omitted for compactness):

$$\begin{aligned} h = 1 \ &: \qquad \{\mathbf{p}_1^+\},\ \{\mathbf{p}_1^-\}; \\ h = 2 \ &: \qquad \{\mathbf{p}_1^+,\mathbf{p}_2^-\},\ \{\mathbf{p}_1^-,\mathbf{p}_2^+\}; \\ h = 3 \ &: \qquad \{\mathbf{p}_1^+,\mathbf{p}_2^-,\mathbf{p}_3^+\},\ \{\mathbf{p}_1^-,\mathbf{p}_2^+,\mathbf{p}_3^-\}. \end{aligned}$$

The sequences $\{\mathbf{p}_1^-\}$, $\{\mathbf{p}_1^+,\mathbf{p}_2^-\}$, $\{\mathbf{p}_1^-,\mathbf{p}_2^+,\mathbf{p}_3^-\}$ have to be discarded due to the fact that the last spline has to be covered with a car's forward movement (front car parking). Hence the topologically possible sequences are: $\{\mathbf{p}_1^+\}$, $\{\mathbf{p}_1^-,\mathbf{p}_2^+\}$, $\{\mathbf{p}_1^+,\mathbf{p}_2^-,\mathbf{p}_3^+\}$. Parking with $\{\mathbf{p}_1^+\}$ is not feasible because the multi-optimization (2.10) fails to satisfy all the required constraints (2.11)-(2.13). Instead, both sequences $\{\mathbf{p}_1^-,\mathbf{p}_2^+\}$ and $\{\mathbf{p}_1^+,\mathbf{p}_2^-,\mathbf{p}_3^+\}$ lead to feasible parking maneuvers.

For the two splines maneuver the multi-optimization of $\{\mathbf{p}_1^-,\mathbf{p}_2^+\}$ leads to a Pareto optimal solution

$$\bar{\mathbf{z}} \in \mathcal{Z} = [-2.5, 2.5]^4 \times \mathbb{R}_+^4 \times \mathbb{R}^2 \times [0, 2\pi) \times [-0.218, 0.218]\,,$$

for which $\kappa_{max}(\bar{\mathbf{z}}) = 0.143$ m^{-1}, $\dot{\kappa}_{max}(\bar{\mathbf{z}}) = 0.260$ m^{-2}, $s_{tot}(\bar{\mathbf{z}}) = 22.8$ m. This solution is depicted with graphic simulation in figure 2.6. Plots of curvature and curvature derivative are reported in figure 2.7.

For the three splines maneuver the multi-optimization of $\{\mathbf{p}_1^+,\mathbf{p}_2^-,\mathbf{p}_3^+\}$ leads to solution

$$\bar{\mathbf{z}} \in \mathcal{Z} = [-2.5, 2.5]^6 \times \mathbb{R}_+^6 \times \mathbb{R}^4 \times [0, 2\pi)^2 \times [-0.218, 0.218]^2\,,$$

for which $\kappa_{max}(\bar{\mathbf{z}}) = 0.168$ m^{-1}, $\dot{\kappa}_{max}(\bar{\mathbf{z}}) = 0.704$ m^{-2}, $s_{tot}(\bar{\mathbf{z}}) = 25$ m. This solution is depicted figure 2.8, while curvature and curvature derivative are reported in figure 2.9.

Example 2: As second example, a *parallel parking* maneuver in a constrained environment is considered with the same data for the dynamic model and for the constraints, given for the precedent example. The car has start and final configurations

$$\mathbf{q}_s = [x_s\ y_s\ \theta_s\ \delta_s]' = [-2.5\ 2.5\ \pi\ 0]'\,,$$

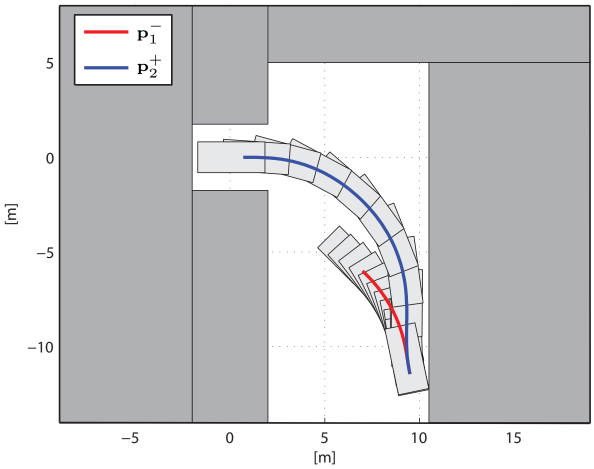


Figure 2.6: Optimal parking with two-spline maneuver $\{\mathbf{p}_1^-, \mathbf{p}_2^+\}$ in example 1.

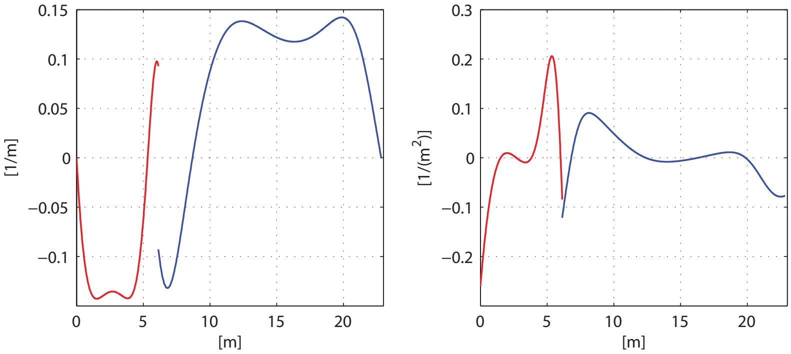


Figure 2.7: Plots of curvature and curvature derivative as functions of the arc length along the entire optimal spline maneuver $\{\mathbf{p}_1^-, \mathbf{p}_2^+\}$ in example 1.

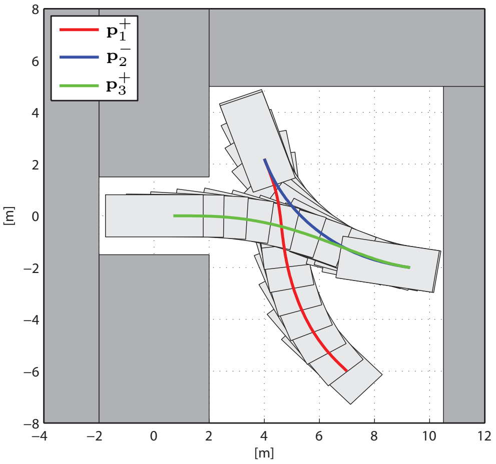


Figure 2.8: Optimal parking with three-spline maneuver $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \mathbf{p}_3^+\}$ in example 1.

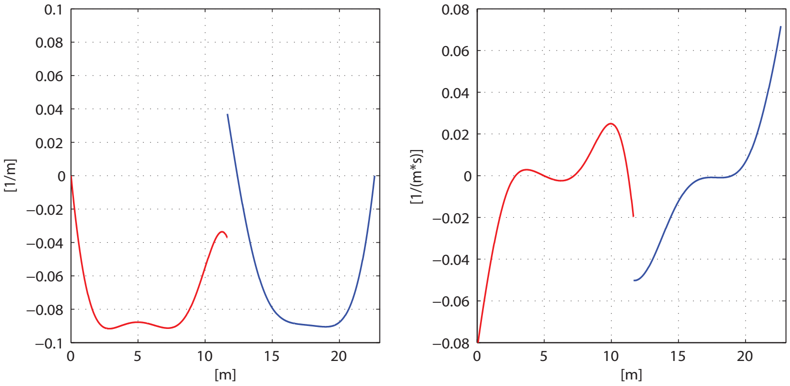


Figure 2.9: Plots of curvature and curvature derivative as functions of the arc length along the entire optimal spline maneuver $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \mathbf{p}_3^+\}$ in example 1.

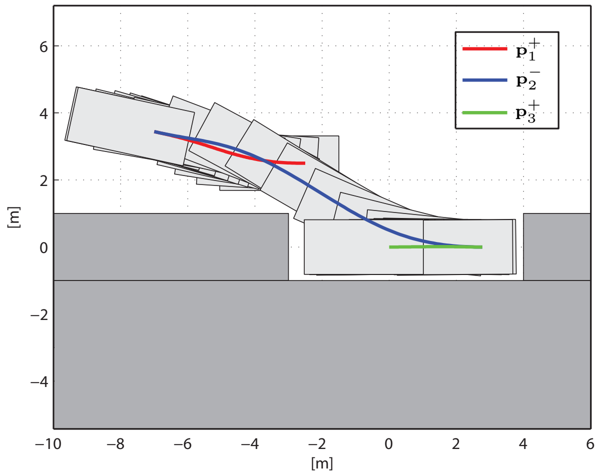


Figure 2.10: Optimal parking with three-spline maneuver $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \mathbf{p}_3^+\}$ in example 2.

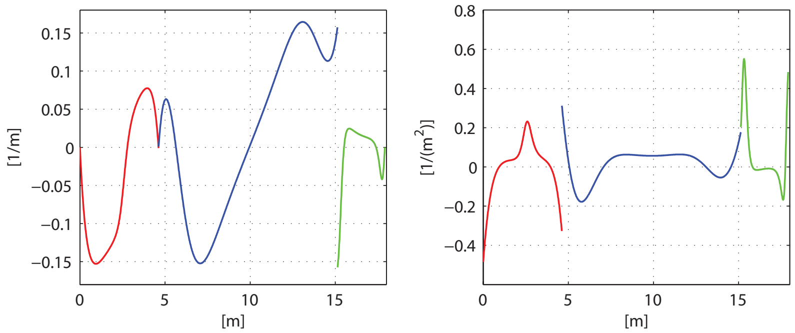


Figure 2.11: Plots of curvature and curvature derivative as functions of the arc length along the entire optimal spline maneuver $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \mathbf{p}_3^+\}$ in example 2.

and

$$\mathbf{q}_g = [x_g \; y_g \; \theta_g \; \delta_g] = [0 \; 0 \; \pi \; 0]' \, ,$$

respectively. Setting $\lambda_1 = 0.3$, $\lambda_2 = 0.2$, and $\lambda_3 = 0.5$, sequence $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \mathbf{p}_3^+\}$ is the first one results to be feasible. The optimal solution

$$\bar{\mathbf{z}} \in \mathscr{Z} = [-2.5, 2.5]^6 \times \mathbb{R}_+^6 \times \mathbb{R}^4 \times [0, 2\pi)^2 \times [-0.218, 0.218]^2 \, .$$

for the sequence $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \mathbf{p}_3^+\}$ gives the results: $\kappa_{max}(\bar{\mathbf{z}}) = 0.165\,\mathrm{m}^{-1}$, $\dot{\kappa}_{max}(\bar{\mathbf{z}}) = 0.551\ \mathrm{m}^{-2}$, $s_{tot}(\bar{\mathbf{z}}) = 17.9$ m. This solution is depicted with graphic simulation in figure 2.10. Plots of curvature and curvature derivative are reported in figure 2.11.

2.2 Path generation for a truck and trailer vehicle

In this section a method for the smooth path generation of a truck and trailer vehicle is presented. The advantages and potentialities in achieving full or partial autonomy in the guidance of automated vehicles are a strong motivation to improve current technologies and methodologies. Focusing on the motion automation of articulated vehicles, the present work addresses the need to generate high quality drive paths for an automated truck and trailer vehicle. This need can arise in a variety of applications (e.g. in industry, agriculture, mining, etc. [37, 38]).

Considering the usual kinematic model of a truck and trailer vehicle, this section presents a new trajectory generation method in which the feedforward (i.e. open-loop) control can steer the vehicle from an initial configuration to a final one, while permitting free shaping of the trailer path connecting these configurations. With this method, the feedforward controls, i.e. the truck velocity and the steering angle of the front wheels, are smooth C^1-signals, the initial and final configurations are arbitrary and the connecting path is modeled by using a new curve primitive, the $\boldsymbol{\eta}^4$-spline.

The problem of nonholonomic trajectory generation for an n-trailer vehicle (i.e. an articulated vehicle consisting of a truck towing n trailers) was considered and solved in [39] by using three distinct classes of control inputs: sinusoids, piece-wise constants, and polynomials. This method relies on, by coordinate transformations, the conversion of the n-trailer system into a Goursat normal form and then into the corresponding *chained form* [40] for which the controllability problem (i.e. the problem of steering between system configurations) is solved by feedforward control. Then, by reversing the transformations the actual system inputs are obtained; however in this reversing singularities may appear so that the desired control is not guaranteed to be obtained in all planning cases. Moreover, the method does not account for any flexibility in direct shaping or modeling the Cartesian paths of the trailers and the truck.

This section proposes a path generation methodology for the smooth feedforward control of the truck and trailer vehicle within the framework of path-velocity decomposition [3]. A result presented in the following subsections (proposition 4) shows that the path generated by the vehicle trailer is a G^4-path [32, 33] (i.e. a path which has fourth-order geometric continuity) if and only if, excluding kinematic singularities, the velocity and the steering functions of the truck are C^1-functions.

Fourth-order geometric continuity accounts for the continuity along the curve of the path itself, the unit tangent vector, the curvature, and the first and second order curvature derivatives with respect to the arc length. Therefore, when pursuing the smooth feedforward control of the articulated vehicle, path planning can be pertinently done with G^4-paths. This naturally leads to the polynomial G^4-interpolating problem on the Cartesian plane.

The section presents a complete solution to this interpolating problem. The solution is the $\boldsymbol{\eta}^4$-spline which is a ninth-order polynomial curve interpolating Cartesian points with associated arbitrary G^4-data (unit tangent vector, curvature, first and second derivatives of curvature). The $\boldsymbol{\eta}^4$-spline generalizes the $\boldsymbol{\eta}^2$-spline and $\boldsymbol{\eta}^3$-spline previously presented in the precedent sections. The $\boldsymbol{\eta}^4$-spline is a curve primitive that depends on set of 8 parameters, which can be freely chosen to modify the path shape without changing the interpolation conditions at the path endpoints.

2.2.1 Smooth feedforward control of the truck and trailer vehicle

Consider a truck an trailer vehicle with the trailer supposed to be joined to the truck at the midpoint of its rear axle. See figure 2.12 where a schematic plan view of the articulated vehicle on a Cartesian frame $\{x,\, y\}$ is depicted. We indicate with couple $(x_1,\, y_1)$ the coordinates of the axle midpoint of the trailer and with θ_1 its orientation angle with respect to the x axis. The truck actuates the motion by the velocity v of the rear wheels and by the steering angle δ of the front wheels. The distance between the front axle and the rear axle of the truck is d_0, whereas the distance between the trailer axle and the rear axle of the truck is d_1. With the usual modeling assumptions of rigid body of the truck and the trailer and of no-slippage of the wheels, the following nonlinear kinematic model of the articulated vehicle can be deduced

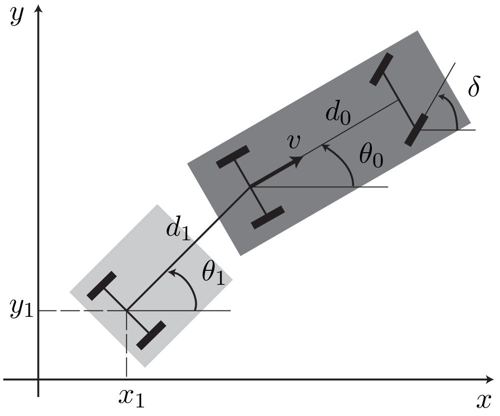


Figure 2.12: Schematic of a truck and trailer vehicle.

$$\left\{\begin{array}{lcl} \dot{x}_1 & = & v\,\cos(\theta_0-\theta_1)\,\cos\theta_1 \\ \dot{y}_1 & = & v\,\cos(\theta_0-\theta_1)\,\sin\theta_1 \\ \dot{\theta}_0 & = & \frac{v}{d_0}\,\tan\delta \\ \dot{\theta}_1 & = & \frac{v}{d_1}\,\sin(\theta_0-\theta_1)\,. \end{array}\right. \tag{2.15}$$

We saw in the precedent sections that in this context it is convenient to use the *extended state* of model (2.15), or *configuration* of the articulated vehicle, which is defined as the state plus the inputs and their derivatives:

$$(x_1, y_1, \theta_0, \theta_1, v, \dot{v}, \delta, \dot{\delta})\,. \tag{2.16}$$

The following definition will be used along this section:

Definition 5 ($\mathbf{G^k}$-curve, $\mathbf{k \geq 2}$) *A curve $\mathbf{p}(u)$, with $u \in [u_0, u_1]$, has k-th order geometric continuity, and we say $\mathbf{p}(u)$ is a G^k-curve, if $\mathbf{p}(u)$ is a G^{k-1}-curve, $\frac{d^k}{du^k}\mathbf{p}(u) \in PC([u_0, u_1])$, and the $(k-2)$-th order derivative of the curvature with respect to the arc length is continuous along the curve, i.e. $\frac{d^{k-2}}{ds^{k-2}}\kappa(u) \in C^0([u_0, u_1])$.*

The G^k-continuity of curves can be naturally extended to Cartesian paths as follows:

Definition 6 ($\mathbf{G^k}$-paths) *A given set of points of a Cartesian plane is a G^k-path if there exists a parametric G^k-curve whose image is the given path.*

We stated above that, in order to obtain a smooth vehicle motion, inputs $v(t)$ and $\delta(t)$ must be C^1-functions. Such a continuity of these vehicle inputs is linked to the fourth-order geometric continuity of the trailer path as stated by the following proposition.

Proposition 4 *Assign any $t_f > 0$. For model (2.15), consider smooth inputs $v(t), \delta(t) \in C^1([0, t_f])$, with $v(t) \neq 0$, $|\delta(t)| < \frac{\pi}{2}$ and initial conditions such that $|\theta_0(t) - \theta_1(t)| < \frac{\pi}{2}$, $\forall t \in [0, t_f]$. Then the path generated by model (2.15), i.e. $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}([0, t_f])$, is a G^4-path. Conversely, given a G^4-path Γ there exist smooth inputs $v(t), \delta(t) \in C^1([0, t_f])$ with $v(t) \neq 0$, $|\delta(t)| < \frac{\pi}{2}$, $\forall t \in [0, t_f]$ and initial conditions for which $|\theta_0(t) - \theta_1(t)| < \frac{\pi}{2}$, $\forall t \in [0, t_f]$ and the path generated by system (2.15) coincides with the given Γ, i.e. $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}([0, t_f]) \equiv \Gamma$.*

Proof. Let us demonstrate the first part of the proposition. The solution of the differential equations (2.15) leads to trajectory $[x_1(t)\, y_1(t)]'$, $t \in [0, t_f]$ which is a regular Cartesian curve. Indeed, its derivative $[\dot{x}_1(t)\, \dot{y}_1(t)]^T$ never vanishes over $[0, t_f]$ because $v(t) \neq 0$ and $|\theta_0(t) - \theta_1(t)| < \frac{\pi}{2}$, $\forall t \in [0, t_f]$.

The unit tangent vector of curve $[x_1(t)\, y_1(t)]'$ can be expressed as

$$\boldsymbol{\tau}(t) = \frac{[\dot{x}_1(t)\, \dot{y}_1(t)]'}{\sqrt{\dot{x}_1^2(t) + \dot{y}_1^2(t)}} = \operatorname{sgn}(v(t)) \begin{bmatrix} \cos\theta_1(t) \\ \sin\theta_1(t) \end{bmatrix} . \tag{2.17}$$

Hence, the unit tangent vector $\boldsymbol{\tau}$ is continuous over the trailer curve because $\theta_1(t)$ is continuous in $[0, t_f]$.

As known from the theory of planar curves [41], the scalar curvature κ is given by the derivative of the tangent angle θ_1 with respect to the arc length s, where $s = \int_0^t (\dot{x}_1^2(\xi) + \dot{y}_1^2(\xi))^{\frac{1}{2}} d\xi$. It can be expressed as follows

$$\begin{aligned} \kappa &= \frac{d\theta_1}{ds} = \frac{d\theta_1}{dt} \frac{1}{\frac{ds}{dt}} = \dot{\theta}_1 \frac{1}{(\dot{x}_1^2 + \dot{y}_1^2)^{\frac{1}{2}}} \\ &= \frac{v}{d_1} \sin(\theta_0 - \theta_1) \frac{1}{|v| \cos(\theta_0 - \theta_1)} \\ &= \operatorname{sgn}(v) \frac{1}{d_1} \tan(\theta_0 - \theta_1) . \end{aligned} \tag{2.18}$$

For the continuity of the state variables θ_0 and θ_1, curvature κ is continuous in $[0, t_f]$ too. The derivative of the scalar curvature κ is given by

$$\frac{d\kappa}{ds} = \frac{1}{d_1 \cos^3(\theta_0 - \theta_1)} \left[\frac{1}{d_0} \tan\delta - \frac{1}{d_1} \sin(\theta_0 - \theta_1) \right] . \tag{2.19}$$

The curvature derivative $\frac{d\kappa}{ds}$ is then continuous along the curve because θ_0, θ_1 and δ are continuous in $[0, t_f]$. Finally, the second derivative of the curvature can be expressed as follows

$$\begin{aligned} \frac{d^2\kappa}{ds^2} = &\frac{\dot{\delta}}{|v| d_0 d_1 \cos^2\delta \cos^4(\theta_0 - \theta_1)} - \operatorname{sgn}(v) \frac{\frac{1}{d_0} \tan\delta - \frac{1}{d_1} \sin(\theta_0 - \theta_1)}{d_1^2 \cos^3(\theta_0 - \theta_1)} \\ &+ \operatorname{sgn}(v) \frac{3 \left[\frac{1}{d_0} \tan\delta - \frac{1}{d_1} \sin(\theta_0 - \theta_1) \right]^2 \sin(\theta_0 - \theta_1)}{d_1 \cos^5(\theta_0 - \theta_1)} . \end{aligned} \tag{2.20}$$

Again, from the continuity of the state variables θ_0 and θ_1 and from the hypothesis $v, \delta \in C^1([0, t_f])$, the second derivative of the curvature with respect to the arc length is continuous in $[0, t_f]$. This shows that curve $[x_1(t)\, y_1(t)]'$ is a G^4-curve, hence the image $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}([0, t_f])$ is a G^4-path.

In order to prove the converse part of the proposition, consider the G^4-curve $\mathbf{p}(s)$, where s is the arc length on Γ and $\mathbf{p}([0, s_f]) \equiv \Gamma$ with s_f being the total arc length of Γ. We choose the following initial conditions

$$\begin{cases} \begin{bmatrix} x_1(0) \\ y_1(0) \end{bmatrix} = \mathbf{p}(0) \\ \theta_0(0) = \arg \frac{d\mathbf{p}}{ds}(0) + \arctan(d_1 \kappa(0)) \\ \theta_1(0) = \arg \frac{d\mathbf{p}}{ds}(0)\,, \end{cases} \tag{2.21}$$

where $\frac{d\mathbf{p}}{ds}(s)$ and $\kappa(s)$ are the unit tangent vector and the curvature of $\mathbf{p}(s)$ respectively.

Also consider any $v_1(t) \in C^1([0, t_f])$ such that $v_1(t) > 0$, $\forall t \in [0, t_f]$ and

$$\int_0^{t_f} v_1(\xi) d\xi = s_f\,.$$

Then define the control inputs as

$$v(t) = v_1(t) \left[1 + d_1^2 \kappa^2(s)\right]^{\frac{1}{2}} \Big|_{s=\int_0^t v_1(\xi)d\xi} \tag{2.22}$$

and

$$\delta(t) = \arctan \left[\frac{d_0 \kappa}{(1 + d_1^2 \kappa^2)^{\frac{1}{2}}} + \frac{d_0 d_1 \frac{d\kappa}{ds}}{(1 + d_1^2 \kappa^2)^{\frac{3}{2}}} \right] \Bigg|_{s=\int_0^t v_1(\xi)d\xi} \,. \tag{2.23}$$

Obviously, $v(t) \neq 0$, $\forall t \in [0, t_f]$ and $v(t) \in C^1([0, t_f])$ because $v_1 \in C^1([0, t_f])$ and $\kappa \in C^1([0, s_f])$. Moreover, $|\delta(t)| < \frac{\pi}{2}$, $\forall t \in [0, t_f]$ and $\delta(t) \in C^1([0, t_f])$ because $\kappa \in C^2([0, s_f])$ (indeed $\mathbf{p}(s)$ is a G^4-curve).

Explicit solutions of system (2.15) can be given for θ_0 and θ_1 as follows:

$$\theta_0(t) = \theta_0(0) + \int_0^t \frac{v(r)}{d_0} \tan \delta(r)\, dr\,, \tag{2.24}$$

$$\theta_1(t) = \theta_0(t) - \arctan\left[d_1 \kappa(s)\right]\big|_{s=\int_0^t v_1(\xi)d\xi}\,. \tag{2.25}$$

Straightforwardly, solution (2.24) satisfies the third equation of system (2.15). By explicit derivation of solution (2.25) and some computations the fourth equation of system (2.15) is also verified and

$$\dot{\theta}_1(t) = v_1(t)\kappa(s)|_{s=\int_0^t v_1(\xi)d\xi}\,,\ t \in [0, t_f]\,. \tag{2.26}$$

From (2.25) evidently the inequality $|\theta_0(t) - \theta_1(t)| < \frac{\pi}{2}$, $\forall t \in [0, t_f]$ follows. The last point to prove is

$$\begin{bmatrix} x_1(t) \\ y_1(t) \end{bmatrix} = \mathbf{p}(s)|_{s=\int_0^t v_1(\xi)\,d\xi}\,,\ t \in [0, t_f]\,. \tag{2.27}$$

First note that

$$\theta_1(t) = \arg \left.\frac{d\mathbf{p}}{ds}\right|_{s=\int_0^t v_1(\xi)\,d\xi}\,, \tag{2.28}$$

and recall that

$$\kappa = \frac{d}{ds}(\arg \boldsymbol{\tau})\,, \tag{2.29}$$

because $\theta_1(0) = \arg \frac{d\mathbf{p}}{ds}(0)$ (cf. conditions (2.21)) and the derivatives of both sides of (2.28) coincide (cf. (2.29) and (2.26)):

$$\begin{aligned} \frac{d}{dt} \arg \left.\frac{d\mathbf{p}}{ds}\right|_{s=\int_0^t v_1(\xi)\,d\xi} &= \frac{d}{ds} \arg \left.\frac{d\mathbf{p}}{ds}\right|_{s=\int_0^t v_1(\xi)\,d\xi} \cdot \frac{ds}{dt} \\ &= \kappa(s)|_{s=\int_0^t v_1(\xi)\,d\xi} \cdot v_1(t) = \dot{\theta}_1(t)\,. \end{aligned}$$

In turn, identity (2.27) holds because $[x_1(0)\,y_1(0)]' = \mathbf{p}(0)$ (cf. conditions (2.21)) and derivatives of the sides of (2.27) are equal to each other. Indeed, by virtue of (2.22) and (2.25)

$$v_1(t) = v(t)\cos(\theta_0(t) - \theta_1(t))\,,$$

so that

$$\begin{aligned}
\frac{d}{dt}\mathbf{p}(s)\Big|_{s=\int_0^t v_1(\xi)d\xi} &= \frac{d\mathbf{p}}{ds}\Big|_{s=\int_0^t v_1(\xi)\,d\xi} \cdot \frac{ds}{dt} \\
&= \left[\begin{array}{c} \cos\arg\frac{d\mathbf{p}}{ds} \\ \sin\arg\frac{d\mathbf{p}}{ds} \end{array}\right]\Bigg|_{s=\int_0^t v_1(\xi)d\xi} \cdot v_1(t) \\
&= \left[\begin{array}{c} \cos\theta_1(t) \\ \sin\theta_1(t) \end{array}\right] v(t)\cos(\theta_0(t)-\theta_1(t)) = \left[\begin{array}{c} \dot{x}_1(t) \\ \dot{y}_1(t) \end{array}\right],
\end{aligned}$$

the last equality being derived from the first two equations of system (2.15). □

The provided proof of proposition 4 is fully constructive. Indeed, it provides the dynamic path inversion procedure to determine the feedforward inverse control to drive the articulated vehicle from a given configuration to a target configuration, along a G^4-path. This path can be any desired G^4-path provided that the path endpoints have Cartesian coordinates, unit tangent vector, curvature, and first and second derivatives of curvature in accordance with the current vehicle configuration (cf. (2.17)-(2.20)). Hence, the generation of a G^4-path for the articulated vehicle must ensure interpolating conditions at the endpoints up to the second derivative of the curvature. This is the problem that is addressed, in a polynomial setting, in the next subsection.

2.2.2 The η^4-splines

Considered the result relative to the smooth feedforward control of the truck and trailer vehicle as exposed in the previous section (proposition 4), the following interpolation problem in the Cartesian plane is introduced.

Problem 3 *Determine the minimal order polynomial curve which interpolates two given endpoints $\mathbf{p}_A = [x_A\ y_A]'$ and $\mathbf{p}_B = [x_B\ y_B]'$ with associated unit tangent vectors defined by angles θ_A and θ_B, scalar curvatures κ_A and κ_B, curvature derivatives $\dot{\kappa}_A$, $\dot{\kappa}_B$ and second-order derivatives of the curvature $\ddot{\kappa}_A$, $\ddot{\kappa}_B$ (both derivatives are defined with respect to the arc length) (see figure* 2.13*).*

Assume that interpolating data $\mathbf{p}_A$, $\mathbf{p}_B \in \mathbb{R}^2$, $\theta_A, \theta_B \in [0, 2\pi)$, κ_A, $\kappa_B \in \mathbb{R}$, $\dot{\kappa}_A$, $\dot{\kappa}_B \in \mathbb{R}$ *and* $\ddot{\kappa}_A$, $\ddot{\kappa}_B \in \mathbb{R}$ *can be arbitrarily assigned.*

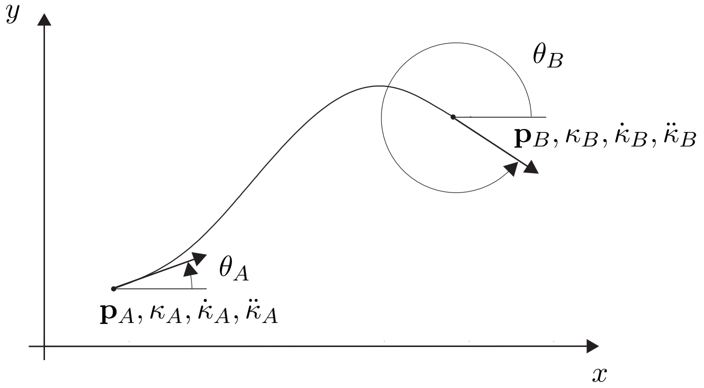

y
θ_B
$\mathbf{p}_B, \kappa_B, \dot{\kappa}_B, \ddot{\kappa}_B$
θ_A
$\mathbf{p}_A, \kappa_A, \dot{\kappa}_A, \ddot{\kappa}_A$
x

Figure 2.13: The polynomial G^4-interpolating problem.

The provisional solution for the above interpolating problem is given by a ninth-order polynomial curve $\mathbf{p}(u) = [\alpha(u)\ \beta(u)]'$, $u \in [0, 1]$ defined as follows

$$\alpha(u) = \sum_{i=0}^{9} \alpha_i u^i, \tag{2.30}$$

$$\beta(u) = \sum_{i=0}^{9} \beta_i u^i, \tag{2.31}$$

where coefficients α_i, β_i $i = 0, \ldots, 9$ are to be determined according to the above interpolating problem. As known from the theory of planar curves, the unit tangent vector $\boldsymbol{\tau}$ and curvature κ can be expressed as

$$\boldsymbol{\tau}(u) = \frac{[\dot{\alpha}\ \dot{\beta}]'}{(\dot{\alpha} + \dot{\beta})^{1/2}}\,, \tag{2.32}$$

$$\kappa(u) = \frac{\dot{\alpha}\ddot{\beta} - \ddot{\alpha}\dot{\beta}}{(\dot{\alpha} + \dot{\beta})^{3/2}}\,. \tag{2.33}$$

Deduction of the first and second derivative of the curvature κ with respect to the arc length leads to the following formulae:

$$\frac{d\kappa}{ds}(u) = \frac{(\dot{\alpha}\dddot{\beta} - \dddot{\alpha}\dot{\beta})(\dot{\alpha}^2 + \dot{\beta}^2) - 3(\dot{\alpha}\ddot{\beta} - \ddot{\alpha}\dot{\beta})(\dot{\alpha}\ddot{\alpha} + \dot{\beta}\ddot{\beta})}{(\dot{\alpha}^2 + \dot{\beta}^2)^3}, \tag{2.34}$$

$$\begin{aligned}\frac{d^2\kappa}{ds^2}(u) = \Big[&(\dot{\alpha}\ddddot{\beta} - \ddddot{\alpha}\dot{\beta} + \ddot{\alpha}\dddot{\beta} - \dddot{\alpha}\ddot{\beta})(\dot{\alpha}^2 + \dot{\beta}^2)^2 - 7(\dot{\alpha}\dddot{\beta} - \dddot{\alpha}\dot{\beta})(\dot{\alpha}\ddot{\alpha} + \dot{\beta}\ddot{\beta}) \\ &(\dot{\alpha}^2 + \dot{\beta}^2) - 3(\dot{\alpha}\ddot{\beta} - \ddot{\alpha}\dot{\beta})(\ddot{\alpha}^2 + \ddot{\beta}^2 + \dot{\alpha}\dddot{\alpha} + \dot{\beta}\dddot{\beta})(\dot{\alpha}^2 + \dot{\beta}^2) \\ &+ 18(\dot{\alpha}\ddot{\beta} - \ddot{\alpha}\dot{\beta})(\dot{\alpha}\ddot{\alpha} + \dot{\beta}\ddot{\beta})^2\Big] \frac{1}{(\dot{\alpha}^2 + \dot{\beta}^2)^5}.\end{aligned} \tag{2.35}$$

The imposition of the G^4-interpolating conditions of the above problem on the endpoints of $\mathbf{p}(u)$ leads to the following relations:

$$\mathbf{p}(0) = \mathbf{p}_A, \tag{2.36}$$

$$\mathbf{p}(1) = \mathbf{p}_B, \tag{2.37}$$

$$\dot{\mathbf{p}}(0) = \eta_1 \begin{bmatrix} \cos\theta_A \\ \sin\theta_A \end{bmatrix}, \tag{2.38}$$

$$\dot{\mathbf{p}}(1) = \eta_2 \begin{bmatrix} \cos\theta_B \\ \sin\theta_B \end{bmatrix}, \tag{2.39}$$

$$\kappa(0) = \kappa_A, \tag{2.40}$$

$$\kappa(1) = \kappa_B, \tag{2.41}$$

$$\frac{d\kappa}{ds}(0) = \dot{\kappa}_A, \tag{2.42}$$

$$\frac{d\kappa}{ds}(1) = \dot{\kappa}_B, \tag{2.43}$$

$$\frac{d^2\kappa}{ds^2}(0) = \ddot{\kappa}_A, \tag{2.44}$$

$$\frac{d^2\kappa}{ds^2}(1) = \ddot{\kappa}_B. \tag{2.45}$$

Note that relation (2.38) and (2.39), which ensure the interpolation of the unit tangent vectors, are well posed provided that η_1 and η_2 are any positive parameters.

Relations (2.36)-(2.45) form a nonlinear algebraic system of 14 equations in the 20 unknowns α_i, β_i. Hence this system may admit a solution set with 6 degrees of freedom. This solution set can be parametrized according to the introduction of further 6 real parameters $\eta_3, \dots, \eta_8$ defined as follows:

$$\langle \ddot{\mathbf{p}}(0)\,, \begin{bmatrix} \cos\theta_A \\ \sin\theta_A \end{bmatrix} \rangle = \eta_3\,, \tag{2.46}$$

$$\langle \ddot{\mathbf{p}}(1)\,, \begin{bmatrix} \cos\theta_B \\ \sin\theta_B \end{bmatrix} \rangle = \eta_4\,, \tag{2.47}$$

$$\langle \dddot{\mathbf{p}}(0)\,, \begin{bmatrix} \cos\theta_A \\ \sin\theta_A \end{bmatrix} \rangle = \eta_5\,, \tag{2.48}$$

$$\langle \dddot{\mathbf{p}}(1)\,, \begin{bmatrix} \cos\theta_B \\ \sin\theta_B \end{bmatrix} \rangle = \eta_6\,, \tag{2.49}$$

$$\langle \ddddot{\mathbf{p}}(0)\,, \begin{bmatrix} \cos\theta_A \\ \sin\theta_A \end{bmatrix} \rangle = \eta_7\,, \tag{2.50}$$

$$\langle \ddddot{\mathbf{p}}(1)\,, \begin{bmatrix} \cos\theta_B \\ \sin\theta_B \end{bmatrix} \rangle = \eta_8\,. \tag{2.51}$$

Equations (2.36)-(2.45) and (2.46)-(2.51) form an algebraic system of 20 equations in the 20 unknowns α_i, β_i, $i = 0, \dots, 9$ that depends on the real parameters $\eta_1, \eta_2 \in \mathbb{R}_+$ and $\eta_3, \dots, \eta_8 \in \mathbb{R}$. This parameters can be packed to form the eta vector $\boldsymbol{\eta} := [\eta_1 \ \dots \ \eta_8]'$ belonging to the parameter space $\mathcal{H} := \mathbb{R}^2_+ \times \mathbb{R}^6$. From equations (2.36) and (2.38) we determine

$$\begin{gathered} \alpha_0 = x_A\,, \quad \beta_0 = y_A\,, \\ \alpha_1 = \eta_1 \cos\theta_A\,, \quad \beta_1 = \eta_1 \sin\theta_A\,. \end{gathered} \tag{2.52}$$

Equations (2.37) and (2.39) lead to the linear equations

$$\alpha(1) = \sum_{i=0}^{9} \alpha_i = x_B\,, \quad \beta(1) = \sum_{i=0}^{9} \beta_i = y_B\,, \tag{2.53}$$

$$\dot{\alpha}(1) = \sum_{i=1}^{9} i\,\alpha_i = \eta_2 \cos\theta_B\,, \quad \dot{\beta}(1) = \sum_{i=1}^{9} i\,\beta_i = \eta_2 \sin\theta_B\,. \tag{2.54}$$

From equation (2.40) and solution α_1, β_1 given by (2.52) we obtain

$$-2\eta_1 \sin\theta_A \alpha_2 + 2\eta_1 \cos\theta_A \beta_2 = \eta_1^3 \kappa_A \,, \tag{2.55}$$

and from (2.46)

$$2\cos\theta_A \alpha_2 + 2\sin\theta_A \beta_2 = \eta_3 \,. \tag{2.56}$$

Equations (2.55) and (2.56) give the solutions

$$\alpha_2 = \frac{1}{2}\eta_3 \cos\theta_A - \frac{1}{2}\eta_1^2 \kappa_A \sin\theta_A \,, \tag{2.57}$$

$$\beta_2 = \frac{1}{2}\eta_3 \sin\theta_A - \frac{1}{2}\eta_1^2 \kappa_A \cos\theta_A \,. \tag{2.58}$$

Taking into account relation (2.34), equation (2.42) becomes

$$(6\alpha_1\beta_3 - 6\beta_1\alpha_3)\eta_1^2 - 12(\alpha_1\beta_2 - \alpha_2\beta_1)(\alpha_1\alpha_2 + \beta_1\beta_2) = \eta_1^6 \dot{\kappa}_A \,, \tag{2.59}$$

and from (2.48) we obtain

$$6\cos\theta_A \alpha_3 + 6\sin\theta_A \beta_3 = \eta_5 \,. \tag{2.60}$$

By substitution of solutions (2.52), (2.57), and (2.58), equations (2.59), (2.60) form a linear algebraic system in the unknowns α_3, β_3 which has a unique solution because the determinant of its coefficient matrix is equal to $6\eta_1^3$ and it differs from zero on the assumption $\eta_1 > 0$. This solution is given by

$$\alpha_3 = -\left(\frac{1}{2}\eta_1\eta_3\kappa_A + \frac{1}{6}\eta_1^3\dot{\kappa}_A\right)\sin\theta_A + \frac{1}{6}\eta_5\cos\theta_A \,, \tag{2.61}$$

$$\beta_3 = \left(\frac{1}{2}\eta_1\eta_3\kappa_A + \frac{1}{6}\eta_1^3\dot{\kappa}_A\right)\cos\theta_A + \frac{1}{6}\eta_5\sin\theta_A \,. \tag{2.62}$$

Using relation (2.35), equation (2.44) becomes

$$\begin{gathered} 12(2\alpha_1\beta_4 - 2\alpha_4\beta_1 + \alpha_2\beta_3 - \alpha_3\beta_2)\eta_1^4 - 84(\alpha_1\beta_3 - \alpha_3\beta_1)(\alpha_1\alpha_2 + \beta_1\beta_2)\eta_1^2 \\ -12(\alpha_1\beta_2 - \alpha_2\beta_1)(2\alpha_2^2 + 2\beta_2^2 + 3\alpha_1\alpha_3 + 3\beta_1\beta_3)\eta_1^2 \\ +144(\alpha_1\beta_2 - \alpha_2\beta_1)(\alpha_1\alpha_2 + \beta_1\beta_2)^2 = \eta_1^{10}\ddot{\kappa}_A \,, \end{gathered} \tag{2.63}$$

and from (2.50) we have

$$24\cos\theta_A\alpha_4 + 24\sin\theta_A\beta_4 = \eta_7\,. \tag{2.64}$$

By substitution of solutions (2.52), (2.57), (2.58), (2.61), and (2.62), the above equations (2.63), (2.64) are a linear algebraic system in the unknowns α_4, β_4. There exists a unique solution given by

$$\begin{aligned}\alpha_4 = -\bigg(&\frac{1}{6}\eta_1\eta_5\kappa_A + \frac{1}{4}\eta_1^2\eta_3\dot{\kappa}_A + \frac{1}{8}\eta_1^4\kappa_A^3 + \frac{1}{24}\eta_1^4\ddot{\kappa}_A \\ &- \frac{1}{8}\eta_3\kappa_A\bigg)\sin\theta_A + \frac{1}{24}\eta_7\cos\theta_A\,,\end{aligned} \tag{2.65}$$

$$\begin{aligned}\beta_4 = \bigg(&\frac{1}{6}\eta_1\eta_5\kappa_A + \frac{1}{4}\eta_1^2\eta_3\dot{\kappa}_A + \frac{1}{8}\eta_1^4\kappa_A^3 + \frac{1}{24}\eta_1^4\ddot{\kappa}_A \\ &+ \frac{1}{8}\eta_3\kappa_A\bigg)\cos\theta_A + \frac{1}{24}\eta_7\sin\theta_A\,,\end{aligned} \tag{2.66}$$

because the coefficient matrix of system (2.63), (2.64) is nonsingular (the determinant of this matrix is $24\eta_1^5$ which differs from zero because $\eta_1 > 0$). By substituting relations (2.54) into equation (2.41) we obtain

$$\eta_2\cos\theta_B\,\ddot{\beta}(1) - \eta_2\sin\theta_B\,\ddot{\alpha}(1) = \eta_2^3\kappa_B\,, \tag{2.67}$$

and from (2.47)

$$\cos\theta_B\,\ddot{\alpha}(1) + \sin\theta_B\,\ddot{\beta}(1) = \eta_4\,. \tag{2.68}$$

The linear system given by equations (2.67) and (2.68) admits the unique solution (its coefficient matrix is nonsingular because it is equal to $-\eta_2$ that differs from zero by assumption):

$$\ddot{\alpha}(1) = \eta_4\cos\theta_B - \eta_2^2\kappa_B\sin\theta_B, \tag{2.69}$$

$$\ddot{\beta}(1) = \eta_4\sin\theta_B + \eta_2^2\kappa_B\cos\theta_B. \tag{2.70}$$

Using relations (2.54), (2.69), (2.70) into equation (2.43) we have

$$\eta_2^3\cos\theta_B\,\dddot{\beta}(1) - \eta_2^3\sin\theta_B\,\dddot{\alpha}(1) = \eta_2^6\dot{\kappa}_B + 3\eta_2^4\eta_4\kappa_B \tag{2.71}$$

and from (2.49)

$$\cos\theta_B \dddot{\alpha}(1) + \sin\theta_B \dddot{\beta}(1) = \eta_6. \tag{2.72}$$

The determinant of coefficient matrix of linear equations (2.71), (2.72) is $-\eta_2^3$ so that the following unique solution holds:

$$\dddot{\alpha}(1) = \eta_6 \cos\theta_B - \left(\eta_2^3 \dot{\kappa}_B + 3\eta_2\eta_4\kappa_B\right) \sin\theta_B\,, \tag{2.73}$$

$$\dddot{\beta}(1) = \eta_6 \sin\theta_B + \left(\eta_2^3 \dot{\kappa}_B + 3\eta_2\eta_4\kappa_B\right) \cos\theta_B\,. \tag{2.74}$$

By substituting relations (2.54), (2.69), (2.70), (2.73), (2.74) into equation (2.45) we obtain

$$\begin{gathered}\eta_2^5 \cos\theta_B \ddddot{\beta}(1) - \eta_2^5 \sin\theta_B \ddddot{\alpha}(1) = \\ \eta_2^{10} \ddot{\kappa}_B + 3\eta_2^4\kappa_B^3 + 3\eta_4^2\kappa_B + 4\eta_2\eta_6\kappa_B + 6\eta_2^2\eta_4\dot{\kappa}_B\,.\end{gathered} \tag{2.75}$$

and from (2.51)

$$\cos\theta_B \ddddot{\alpha}(1) + \sin\theta_B \ddddot{\beta}(1) = \eta_8\,. \tag{2.76}$$

Again, the pair of linear equations (2.75) and (2.76) admits a unique solution (the determinant of the coefficient matrix is $-\eta_2^5$) which is reported below:

$$\ddddot{\alpha}(1) = \eta_8 \cos\theta_B - \left[3\left(\eta_4^2\kappa_B + \eta_2^4\kappa_B^3\right) + 4\eta_2\eta_4\kappa_B + 6\eta_2^2\eta_4\dot{\kappa}_B + \eta_2^5\ddot{\kappa}_B\right] \sin\theta_B\,, \tag{2.77}$$

$$\ddddot{\beta}(1) = \eta_8 \sin\theta_B + \left[3\left(\eta_4^2\kappa_B + \eta_2^4\kappa_B^3\right) + 4\eta_2\eta_4\kappa_B + 6\eta_2^2\eta_4\dot{\kappa}_B + \eta_2^5\ddot{\kappa}_B\right] \cos\theta_B\,. \tag{2.78}$$

By collecting the relations defining $\alpha(1)$, $\dot{\alpha}(1)$, $\ddot{\alpha}(1)$, $\dddot{\alpha}(1)$, and $\ddddot{\alpha}(1)$ (cf. (2.53), (2.54), (2.69), (2.73), (2.77)) the following linear system in the unknowns α_5,

α_6, α_7, α_8, is α_9 is obtained:

$$\begin{cases} \alpha_5 + \alpha_6 + \alpha_7 + \alpha_8 + \alpha_9 = x_B - \alpha_0 - \alpha_1 - \alpha_2 - \alpha_3 - \alpha_4 \\ 5\alpha_5 + 6\alpha_6 + 7\alpha_7 + 8\alpha_8 + 9\alpha_9 = \eta_2 \cos\theta_B - \alpha_1 - 2\alpha_2 - 3\alpha_3 - 4\alpha_4 \\ 20\alpha_5 + 30\alpha_6 + 42\alpha_7 + 56\alpha_8 + 72\alpha_9 = \eta_4 \cos\theta_B - \eta_2^2 \kappa_B \sin\theta_B \\ \qquad\qquad - 2\alpha_2 - 6\alpha_3 - 12\alpha_4 \\ 60\alpha_5 + 120\alpha_6 + 210\alpha_7 + 336\alpha_8 + 504\alpha_9 = \eta_6 \cos\theta_B \\ \qquad - \left(\eta_2^3 \dot\kappa_B + 3\eta_2\eta_4\kappa_B\right) \sin\theta_B - 6\alpha_3 - 24\alpha_4 \\ 120\alpha_5 + 360\alpha_6 + 840\alpha_7 + 1680\alpha_8 + 3024\alpha_9 = \eta_8 \cos\theta_B \\ - \left[\eta_2^5 \ddot\kappa_B + 3\left(\eta_2^4 \kappa_B^3 + \eta_4^2 \kappa_B\right) + 4\eta_2\eta_4\kappa_B + 6\eta_2^2\eta_4\dot\kappa_B\right] \cos\theta_B - 24\alpha_4 \,. \end{cases} \tag{2.79}$$

Similarly, by collecting the relations defining $\beta(1)$, $\dot\beta(1)$, $\ddot\beta(1)$, $\dddot\beta(1)$, and $\ddddot\beta(1)$ (cf. (2.53), (2.54), (2.70), (2.74), (2.78)) the following linear system in the unknowns β_5, β_6, β_7, β_8, and β_9 holds:

$$\begin{cases} \beta_5 + \beta_6 + \beta_7 + \beta_8 + \beta_9 = y_B - \beta_0 - \beta_1 - \beta_2 - \beta_3 - \beta_4 \\ 5\beta_5 + 6\beta_6 + 7\beta_7 + 8\beta_8 + 9\beta_9 = \eta_2 \sin\theta_B - \beta_1 - 2\beta_2 - 3\beta_3 - 4\beta_4 \\ 20\beta_5 + 30\beta_6 + 42\beta_7 + 56\beta_8 + 72\beta_9 = \eta_4 \sin\theta_B + \eta_2^2 \kappa_B \cos\theta_B \\ \qquad\qquad - 2\beta_2 - 6\beta_3 - 12\beta_4 \\ 60\beta_5 + 120\beta_6 + 210\beta_7 + 336\beta_8 + 504\beta_9 = \eta_6 \sin\theta_B \\ \qquad + \left(\eta_2^3 \dot\kappa_B + 3\eta_2\eta_4\kappa_B\right) \cos\theta_B - 6\beta_3 - 24\beta_4 \\ 120\beta_5 + 360\beta_6 + 840\beta_7 + 1680\beta_8 + 3024\beta_9 = \eta_8 \sin\theta_B \\ + \left[\eta_2^5 \ddot\kappa_B + 3\left(\eta_2^4 \kappa_B^3 + \eta_4^2 \kappa_B\right) + 4\eta_2\eta_4\kappa_B + 6\eta_2^2\eta_4\dot\kappa_B\right] \sin\theta_B - 24\beta_4 \,. \end{cases} \tag{2.80}$$

The above linear systems (2.79), (2.80) have the same coefficient matrix whose determinant is 288. Hence, a unique solution can be deduced for all the unknowns. The explicit expressions of all coefficients α_i, β_i, $i = 0, \ldots, 9$ are not reported for brevity (see [23] for more details). The resulting polynomial curve is denoted by $\mathbf{p}(u;\, \boldsymbol{\eta})$ and it is called $\boldsymbol{\eta}^4$-spline.

Proposition 5 (Completeness) *The $\boldsymbol{\eta}^4$-spline $\mathbf{p}(u; \boldsymbol{\eta})$ satisfies any given set of interpolating data $\mathbf{p}_A$, θ_A, κ_A, $\dot\kappa_A$, $\ddot\kappa_A$ and $\mathbf{p}_B$, θ_B, κ_B, $\dot\kappa_B$, $\ddot\kappa_B$ for* *all* $\boldsymbol{\eta} \in \mathcal{H}$*. Conversely, given any ninth-order polynomial curve* $\mathbf{q}(u)$*,* $u \in [0,1]$ *with* $\dot{\mathbf{q}}(0) \neq \mathbf{0}$ *and* $\dot{\mathbf{q}}(1) \neq \mathbf{0}$ *which satisfies a given set of interpolating conditions* $\mathbf{p}_A$*,* θ_A*,* κ_A*,* $\dot{\kappa}_A$*,* $\ddot{\kappa}_A$ *and* $\mathbf{p}_B$*,* θ_B*,* κ_B*,* $\dot{\kappa}_B$*,* $\ddot{\kappa}_B$*, there exists a parameter vector* $\boldsymbol{\eta} \in \mathcal{H}$ *such that* $\mathbf{p}(u;\boldsymbol{\eta})$ *coincides with* $\mathbf{q}(u)$*.*

Proof. It has been shown that system (2.36)-(2.51) is equivalent to a linear system whose coefficient matrix is nonsingular, provided that $\eta_1 > 0$ and $\eta_2 > 0$. Hence, the solution provided by the $\boldsymbol{\eta}^4$-spline is unique and satisfies any given set of interpolating data $\mathbf{p}_A$, θ_A, κ_A, $\dot{\kappa}_A$, $\ddot{\kappa}_A$ and $\mathbf{p}_B$, θ_B, κ_B, $\dot{\kappa}_B$, $\ddot{\kappa}_B$ for all $\boldsymbol{\eta} \in \mathcal{H}$. Define $\eta_1 := \|\dot{\mathbf{q}}(0)\|$ and $\eta_2 := \|\dot{\mathbf{q}}(1)\|$, so that $\eta_1, \eta_2 > 0$ by hypothesis. Define $\boldsymbol{\tau}_A := [\cos\theta_A \, \sin\theta_A]'$, $\boldsymbol{\tau}_B := [\cos\theta_B \, \sin\theta_B]'$ and set the parameters $\eta_3, \ldots, \eta_8$ according to

$$\begin{aligned} \eta_3 &:= \langle \ddot{\mathbf{q}}(0)\,,\, \boldsymbol{\tau}_A \rangle, \quad \eta_5 := \langle \dddot{\mathbf{q}}(0)\,,\, \boldsymbol{\tau}_A \rangle, \quad \eta_7 := \langle \ddddot{\mathbf{q}}(0)\,,\, \boldsymbol{\tau}_A \rangle, \\ \eta_4 &:= \langle \ddot{\mathbf{q}}(1)\,,\, \boldsymbol{\tau}_B \rangle, \quad \eta_6 := \langle \dddot{\mathbf{q}}(1)\,,\, \boldsymbol{\tau}_B \rangle, \quad \eta_8 := \langle \ddddot{\mathbf{q}}(1)\,,\, \boldsymbol{\tau}_B \rangle. \end{aligned}$$

Having defined all the eta parameters, consider the algebraic system (2.36)-(2.51) with the given set of interpolating conditions $\mathbf{p}_A$, θ_A, κ_A, $\dot{\kappa}_A$, $\ddot{\kappa}_A$ and $\mathbf{p}_B$, θ_B, κ_B, $\dot{\kappa}_B$, $\ddot{\kappa}_B$. The unknowns are the coefficients of a ninth-order polynomial curve $\mathbf{p}(u)$. Hence, there exists a unique solution, the $\boldsymbol{\eta}^4$-spline $\mathbf{p}(u;\boldsymbol{\eta})$, that must coincides with $\mathbf{q}(u)$. □

Property 1 (Minimality) *The* $\boldsymbol{\eta}^4$*-spline* $\mathbf{p}(u;\boldsymbol{\eta})$ *is the minimal order polynomial curve interpolating any arbitrarily given set of data* $\mathbf{p}_A, \mathbf{p}_B \in \mathbb{R}^2$*,* $\theta_A, \theta_B \in [0, 2\pi)$*,* $\kappa_A, \kappa_B \in \mathbb{R}$*,* $\dot{\kappa}_A, \dot{\kappa}_B \in \mathbb{R}$*, and* $\ddot{\kappa}_A, \ddot{\kappa}_B \in \mathbb{R}$*.*

Proof. Proposition 5 shows that the $\boldsymbol{\eta}^4$-spline $\mathbf{p}(u;\boldsymbol{\eta})$ is the family of all polynomial curves, till to the ninth order, interpolating any given G^4-data. Hence, if an eighth or lower order polynomial curve interpolating any assigned set of boundary condition exists, it must coincide with $\mathbf{p}(u;\boldsymbol{\eta})$ for some appropriate $\boldsymbol{\eta} \in \mathcal{H}$. Consider the following boundary conditions (leading to a so-called lane-change path):

$$\begin{gathered} \mathbf{p}_A = [0\ 0]', \mathbf{p}_B = [2\ 1]',\ \theta_A = \theta_B = 0, \kappa_A = \kappa_B = 0, \\ \dot{\kappa}_A = \dot{\kappa}_B = 0,\ \ddot{\kappa}_A = \ddot{\kappa}_B = 0, \end{gathered}$$

and evaluate the $\boldsymbol{\eta}^4$-spline using its coefficients (cf. [23]):

$$\begin{aligned}
\alpha(u;\boldsymbol{\eta}) = {} & \eta_1 u + \frac{1}{2}\eta_3 u^2 + \frac{1}{6}\eta_5 u^3 + \frac{1}{24}\eta_7 u^4 + \Bigg[252 - 70\eta_1 - 56\eta_2 - \frac{35}{2}\eta_3 \\
& + \frac{21}{2}\eta_4 - \frac{5}{2}\eta_5 - \eta_6 - \frac{5}{24}\eta_7 + \frac{1}{24}\eta_8\Bigg] u^5 + \Bigg[-840 + 224\eta_1 + 196\eta_2 + \frac{105}{2}\eta_3 \\
& - \frac{77}{2}\eta_4 + \frac{20}{3}\eta_5 + \frac{23}{6}\eta_6 + \frac{5}{12}\eta_7 - \frac{1}{6}\eta_8\Bigg] u^6 + [1080 - 280\eta_1 - 260\eta_2 - 63\eta_3 \\
& + 53\eta_4 - \frac{15}{2}\eta_5 - \frac{11}{2}\eta_6 - \frac{5}{12}\eta_7 + \frac{1}{4}\eta_8\Bigg] u^7 + [-630 + 160\eta_1 + 155\eta_2 + 35\eta_3 \\
& - \frac{65}{2}\eta_4 + 4\eta_5 + \frac{7}{2}\eta_6 + \frac{5}{24}\eta_7 - \frac{1}{6}\eta_8\Bigg] u^8 + [140 - 35\eta_1 - 35\eta_2 \\
& - \frac{15}{2}\eta_3 + \frac{15}{2}\eta_4 - \frac{5}{6}\eta_5 - \frac{5}{6}\eta_6 - \frac{1}{24}\eta_7 + \frac{1}{24}\eta_8\Bigg] u^9 ,
\end{aligned}$$

$$\beta(u;\boldsymbol{\eta}) = 126u^5 - 420u^6 + 540u^7 - 315u^8 + 70u^9 .$$

Evidently, $\beta(u;\boldsymbol{\eta})$ is a strict ninth-order polynomial that does non depend on $\boldsymbol{\eta}$. Thus, it is not possible to interpolate the given data with an eighth or lower order polynomial curve. □

Proposition 5 and property 1 make evident that the found $\boldsymbol{\eta}^4$-spline is the complete solution the posed G^4-interpolating problem. The $\boldsymbol{\eta}^4$-spline present itself as a family of polynomial curves parametrized by eta parameters $\eta_1, \ldots, \eta_8$. A relevant property of this parametrization is the symmetry.

Property 2 (Symmetry) *Assume* $\eta_1 = \eta_2 = v \in \mathbb{R}_+$, $\eta_3 = -\eta_4 = w \in \mathbb{R}$, $\eta_5 = \eta 6 = z \in \mathbb{R}$, $\eta_7 = -\eta_8 = t \in \mathbb{R}$ *and define* $\boldsymbol{\eta} = [v \; v \; w \; -w \; z \; z \; t \; -t]'$. *Moreover, consider* $\theta_A = \theta_B = \theta \in [0, 2\pi)$, $\kappa_A = -\kappa_B = \kappa \in \mathbb{R}$, $\dot\kappa_A = \dot\kappa_B = \dot\kappa \in \mathbb{R}$, $\ddot\kappa_A = -\ddot\kappa_B = \ddot\kappa \in \mathbb{R}$. *Then, for any* $\mathbf{p}_A$ *and* $\mathbf{p}_B$, *curve* $\mathbf{p}(u;\boldsymbol{\eta})$ *satisfies the following symmetry relation*

$$\mathbf{p}(u;\boldsymbol{\eta}) = \mathbf{p}_A + \mathbf{p}_B - \mathbf{p}(1-u;\boldsymbol{\eta}) \tag{2.81}$$

$\forall u \in [0,1]$, $\forall v \in \mathbb{R}_+$, $\forall w, z, t \in \mathbb{R}$, $\forall \theta \in [0, 2\pi)$, *and* $\forall \kappa, \dot\kappa, \ddot\kappa \in \mathbb{R}$.

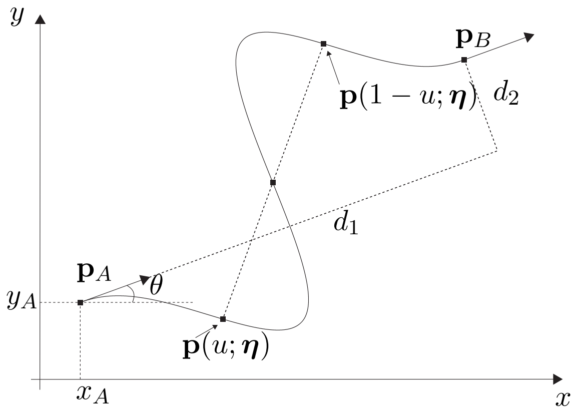

y
$\mathbf{p}_B$
$\mathbf{p}(1-u;\boldsymbol{\eta})$
d_2
d_1
$\mathbf{p}_A$
θ
y_A
$\mathbf{p}(u;\boldsymbol{\eta})$
x_A
x

Figure 2.14: Symmetry of the $\boldsymbol{\eta}^4$-spline.

Proof. It is always possible to find $d_1, d_2 \in \mathbb{R}$ such that (cf. figure 2.14)

$$\mathbf{p}_B = \mathbf{p}_A + d_1 \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} + d_2 \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} .$$

Curve $\mathbf{p}(u;\ \boldsymbol{\eta})$, evaluated by means of its coefficients and the assigned interpolating conditions, can be expressed as

$$\begin{aligned}
\mathbf{p}(u;\boldsymbol{\eta}) = {} & \begin{bmatrix} x_A \\ y_A \end{bmatrix} + v \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} u + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \left\{ \frac{1}{2} \begin{bmatrix} w \\ \kappa v^2 \end{bmatrix} u^2 \right. \\
& + \frac{1}{6} \begin{bmatrix} z \\ \dot\kappa v^3 + 3\kappa v w \end{bmatrix} u^3 + \frac{1}{24} \begin{bmatrix} t \\ \ddot\kappa v^4 + 6\dot\kappa v^2 w + 4\kappa v z + 3\kappa w^2 + 3\kappa^3 v^4 \end{bmatrix} u^4 \\
& + \begin{bmatrix} 126 d_1 - 126 v - 28 w & -\frac{7}{2} z - \frac{1}{4} t \\ 126 d_2 - \frac{1}{4}\ddot\kappa v^4 - \frac{7}{2}\dot\kappa v^3 - \frac{3}{2}\dot\kappa v^2 w - 28\kappa v^2 & -\frac{3}{4}\kappa w^2 - \kappa v z - \frac{21}{2}\kappa v w - \frac{3}{4}\kappa^3 v^4 \end{bmatrix} u^5 \\
& + \begin{bmatrix} -420 d_1 + 420 v + 91 w & +\frac{21}{2} z + \frac{7}{12} t \\ -420 d_2 + \frac{7}{12}\ddot\kappa v^4 + \frac{21}{2}\dot\kappa v^3 + \frac{7}{2}\dot\kappa v^2 w + 91\kappa v^2 & +\frac{7}{4}\kappa w^2 + \frac{7}{3}\kappa v z + \frac{63}{2}\kappa v w + \frac{7}{4}\kappa^3 v^4 \end{bmatrix} u^6 \\
& + \begin{bmatrix} 540 d_1 - 540 v - 116 w & -13 z - \frac{2}{3} t \\ 540 d_2 - \frac{2}{3}\ddot\kappa v^4 - 13\dot\kappa v^3 - 4\dot\kappa v^2 w - 116\kappa v^2 & -2\kappa w^2 - \frac{8}{3}\kappa v z - 39\kappa v w - 2\kappa^3 v^4 \end{bmatrix} u^7
\end{aligned}$$

$$+\begin{bmatrix} -315d_1+315v+\frac{135}{2}w & +\frac{15}{2}z+\frac{3}{8}t \\ -315d_2+\frac{3}{8}\ddot{\kappa}v^4+\frac{15}{2}\dot{\kappa}v^3+\frac{9}{4}\dot{\kappa}v^2w+\frac{135}{2}\kappa v^2 & +\frac{9}{8}\kappa w^2+\frac{3}{2}\kappa vz+\frac{45}{2}\kappa vw+\frac{9}{8}\kappa^3v^4 \end{bmatrix} u^8$$
$$\left. +\begin{bmatrix} 70d_1-70v-15w & -\frac{5}{3}z-\frac{1}{12}t \\ 70d_2-\frac{1}{12}\ddot{\kappa}v^4-\frac{5}{3}\dot{\kappa}v^3-\frac{1}{2}\dot{\kappa}v^2w-15\kappa v^2 & -\frac{1}{4}\kappa w^2-\frac{1}{3}\kappa vz-5\kappa vw-\frac{1}{4}\kappa^3v^4 \end{bmatrix} u^9 \right\}, \tag{2.82}$$

Now, use (2.82) to evaluate $\mathbf{p}(u;\boldsymbol{\eta})+\mathbf{p}(1-u;\boldsymbol{\eta})$. Some algebraic manipulations are required to obtain

$$\mathbf{p}(u;\boldsymbol{\eta})+\mathbf{p}(1-u;\boldsymbol{\eta}) = 2\begin{bmatrix} x_A \\ y_A \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \mathbf{p}_A+\mathbf{p}_B\,,$$

and conclude that, evidently, (2.81) holds $\forall u \in [0,1]$, $\forall v \in \mathbb{R}_+$, $\forall w,z,t \in \mathbb{R}$, $\forall \theta \in [0,2\pi)$, and $\forall \kappa, \dot{\kappa}, \ddot{\kappa} \in \mathbb{R}$. □

A variety of curve primitives (circular arc, clotoids, cubic spirals, etc.) can be approximated by the $\boldsymbol{\eta}^4$-spline (as shown in [32,33] for the $\boldsymbol{\eta}^3$-spline). The significant case relative to the line segment primitive, as illustrated by property below.

Property 3 (Line segment generation) *Let be given any pair of Cartesian point $\mathbf{p}_A$, $\mathbf{p}_B$ with $\mathbf{p}_A \neq \mathbf{p}_B$. Define $\theta := \arg(\mathbf{p}_B-\mathbf{p}_A)$ and set $\theta_A=\theta_B=\theta$, $\kappa_A=\kappa_B=0$, $\dot{\kappa}_A=\dot{\kappa}_B=0$, $\ddot{\kappa}_A=\ddot{\kappa}_B=0$. Then, $\mathbf{p}(u;\ \boldsymbol{\eta})$ is a line segment $\forall \boldsymbol{\eta} \in \mathcal{H}$.*

Proof. Define $d := \|\mathbf{p}_B-\mathbf{p}_A\|$. Hence

$$\mathbf{p}_B = \mathbf{p}_A + d\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}\,,$$

and the $\boldsymbol{\eta}^4$-spline with the assigned interpolating condition can be expressed as follows

$$\mathbf{p}(u;\boldsymbol{\eta}) = \begin{bmatrix} x_A \\ y_A \end{bmatrix} + f(u;\boldsymbol{\eta})\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}\,, \tag{2.83}$$

where $f(u;\boldsymbol{\eta})$ is the following function

$$\begin{aligned} f(u;\boldsymbol{\eta}) &= \eta_1 u + \frac{1}{2}\eta_3 u^2 + \frac{1}{6}\eta_5 u^3 + \frac{1}{24}\eta_7 u^4 + \left[126d - 70\eta_1 - 56\eta_2 - \frac{35}{2}\eta_3 \right. \\ &\left. + \frac{21}{2}\eta_4 - \frac{5}{2}\eta_5 - \eta_6 - \frac{5}{24}\eta_7 + \frac{1}{24}\eta_8\right] u^5 + \left[-420d + 224\eta_1 + 196\eta_2 + \frac{105}{2}\eta_3 \right. \\ &\left. - \frac{77}{2}\eta_4 + \frac{20}{3}\eta_5 + \frac{23}{6}\eta_6 + \frac{5}{12}\eta_7 - \frac{1}{6}\eta_8\right] u^6 + [540d - 280\eta_1 - 260\eta_2 - 63\eta_3 \\ &\left. + 53\eta_4 - \frac{15}{2}\eta_5 - \frac{11}{2}\eta_6 - \frac{5}{12}\eta_7 + \frac{1}{4}\eta_8\right] u^7 + [-315d + 160\eta_1 + 155\eta_2 + 35\eta_3 \\ &\left. - \frac{65}{2}\eta_4 + 4\eta_5 + \frac{7}{2}\eta_6 + \frac{5}{24}\eta_7 - \frac{1}{6}\eta_8\right] u^8 + \left[70d - 35\eta_1 - 35\eta_2 - \frac{15}{2}\eta_3 \right. \\ &\left. + \frac{15}{2}\eta_4 - \frac{5}{6}\eta_5 - \frac{5}{6}\eta_6 - \frac{1}{24}\eta_7 + \frac{1}{24}\eta_8\right] u^9. \end{aligned}$$

It is easy to verify that $f(0;\boldsymbol{\eta}) = 0$ and $f(1;\boldsymbol{\eta}) = d$. Thus, equation (2.83) proves that $\mathbf{p}(u;\boldsymbol{\eta})$ belongs to the segment line joining $\mathbf{p}_A$ with $\mathbf{p}_B$ $\forall \boldsymbol{\eta} \in \mathcal{H}$. □

2.2.3 A path planning example

Consider a parking maneuver for an automated truck and trailer vehicle in an unobstructed environment. The composed vehicle starts from the initial configuration

$$(x_1, y_1, \theta_0, \theta_1, v, \dot{v}, \delta, \dot{\delta}) = (18, 3, \frac{3}{4}\pi, \frac{\pi}{2}, 0, 0, \frac{\pi}{12}, 0)\,,$$

and with a forward movement reaches the final configuration $(0, 0\pi, \pi, 0, 0, 0, 0)$ (as usual, Cartesian coordinates are expressed in meters [m] and angles in radians [rad]). The truck vehicle has wheelbase $d_0 = 3$ m and the distance between the trailer axle and the truck joint is $d_1 = 4$ m. The path of the trailer at the endpoints must have $[x_A\ y_A]' = [18\ 3]'$, $\theta_A = \frac{\pi}{2}$ and $[x_B\ y_B]' = [0\ 0]'$, $\theta_B = \pi$.

From formulae (2.18), (2.19) we also deduce that $\kappa_A = -0.25$ m^{-1}, $\dot{\kappa}_A = 0.1882$ m^{-2} and $\kappa_B = 0$, $\dot{\kappa}_B = 0$. The second derivatives of curvature at the endpoints, $\ddot{\kappa}_A$ and $\ddot{\kappa}_B$, can be freely chosen according to relation (2.20) because the truck and trailer is at rest in the initial and final configurations ($v = 0$ and $\dot{\delta} = 0$). Hence, the parking path which refers to the trailer axis midpoint can be planned using an $\boldsymbol{\eta}^4$-spline $\mathbf{p}(u, \boldsymbol{\eta})$ with the above determined interpolating conditions. The actual shape of this $\boldsymbol{\eta}^4$-spline depends on 10 free parameters ($\ddot{\kappa}_A, \ddot{\kappa}_B \in \mathbb{R}$, $\eta_1, \eta_2 \in \mathbb{R}_+$, $\eta_3, \ldots, \eta_8 \in \mathbb{R}$) and this gives a significant flexibility in achieving a satisfactory parking maneuver. The most important parameters influencing the path shape are $\ddot{\kappa}_A, \ddot{\kappa}_B$ and "curve velocities" η_1, η_2 (cf. (2.38), (2.39)). See figures 2.15, 2.16, 2.17, and 2.18 which depict families of paths according to the following settings. For all $\boldsymbol{\eta}^4$-spline families we have

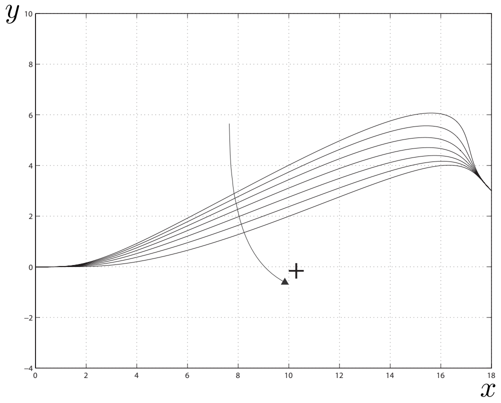


Figure 2.15: The $\boldsymbol{\eta}^4$-spline with $\ddot{\kappa}_A$ varying in $[-3,\ 3]$.

$\eta_3 = \cdots = \eta_8 = 0$ and $\ddot{\kappa}_A \in [-3,\ 3]$, $\ddot{\kappa}_B = 0$, $\eta_1 = \eta_2 = 10$ (figure 2.15), $\ddot{\kappa}_A = 0$, $\ddot{\kappa}_B \in [-3,\ 3]$, $\eta_1 = \eta_2 = 10$ (figure 2.16), $\ddot{\kappa}_A = \ddot{\kappa}_B = 0$, $\eta_1 \in [4,\ 25]$, $\eta_2 = 10$ (figure 2.17), and $\ddot{\kappa}_A = \ddot{\kappa}_B = 0$, $\eta_1 = 10$, $\eta_2 \in [4,\ 25]$ (figure 2.18).

The other shaping parameters are η_3, η_4, the curve acceleration projected on the unit tangent vectors at the endpoints of the $\boldsymbol{\eta}^4$-spline (cf. (2.46), (2.47)), η_5, η_6, the curve jerk at the curve endpoints (cf. (2.48), (2.49)), and η_7, η_8, the curve jerk derivatives at the path endpoints (cf. (2.50), (2.51)).

The freedom in selecting the free parameters leads to pose an optimal path planning problem. A sensible index to minimize is the maximum of the steering

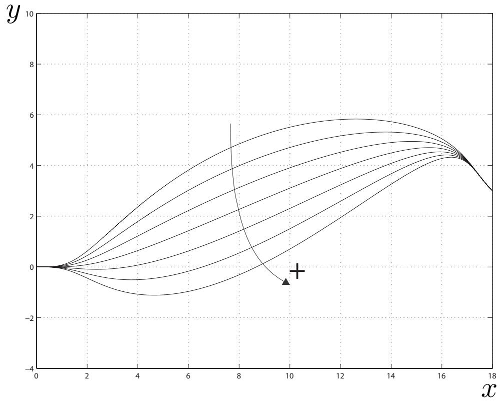


Figure 2.16: The $\boldsymbol{\eta}^4$-spline with $\ddot{\kappa}_B$ varying in $[-3,\ 3]$.

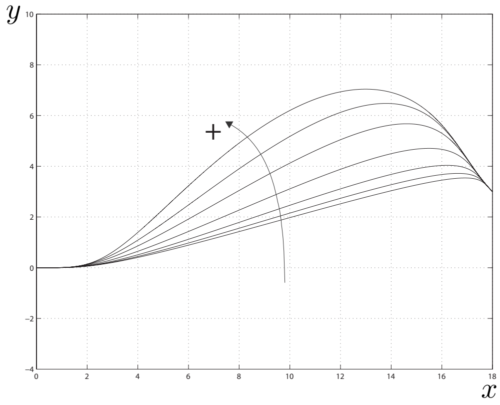


Figure 2.17: The $\boldsymbol{\eta}^4$-spline with η_1 varying in $[4,\ 25]$.

angle modulus over the whole maneuver path:

$$\min_{\ddot{\kappa}_A, \ddot{\kappa}_B \in \mathbb{R},\ \boldsymbol{\eta} \in \mathcal{H}} \delta_{\mathrm{max}}\,, \tag{2.84}$$

where $\delta_{\mathrm{max}} := \max_{s \in [0,\ s_f]} |\delta(s)|$, and s_f denotes the total length of the $\boldsymbol{\eta}^4$-spline. The steering angle as a function of the curvilinear abscissa, $\delta(s)$ can be

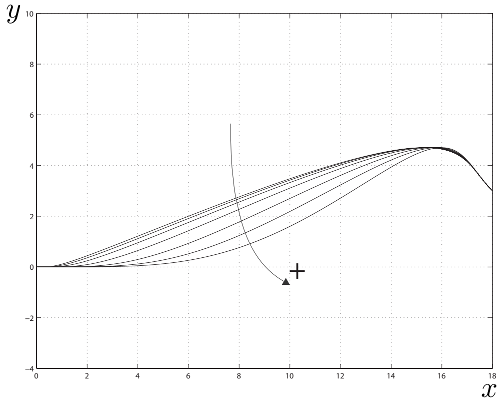


Figure 2.18: The $\boldsymbol{\eta}^4$-spline with η_2 varying in [4, 25].

uniquely determined by relations (2.18) and (2.19):

$$\delta(s) = \arctan\left[\frac{d_0}{(1+d_1^2\kappa^2(s))^{1/2}}\kappa(s) + \frac{d_0 d_1}{(1+d_1^2\kappa^2(s))^{3/2}}\frac{d\kappa}{ds}(s)\right]. \qquad (2.85)$$

The above formula (2.85), which is a generalization of the well-known relation $\delta(s) = \arctan[d_0\kappa(s)]$ for car-like vehicles without trailers, is the basis of the dynamic path inversion approach to the feedforward of the truck and trailer vehicle [5]. The optimal minimax problem (2.84) can be reduced to a standard minimization by a sweeping discretization over $u \in [0,\ 1]$, the curve parameter of spline $\mathbf{p}(u;\boldsymbol{\eta})$. Using local optimization, the following results have been obtained: the optimal maximum steering is $\bar{\delta}_{\max} = 0.6197$ rad (corresponding to 35°.51) which is archived with $\ddot{\kappa}_A = -0.0783$ m^{-2}, $\ddot{\kappa}_B = -0.124$ m^{-2}, $\bar{\eta}_1 = 35.14$, $\bar{\eta}_2 = 22.73$, $\bar{\eta}_3 = 70.40$, $\bar{\eta}_4 = -0.5326$, $\bar{\eta}_5 = -1367$, $\bar{\eta}_6 = -17.42$, $\bar{\eta}_7 = 7013$, and $\bar{\eta}_8 = 214.6$. The corresponding length of the $\boldsymbol{\eta}^4$-spline is $\bar{s}_f = 25.74$ m. These results are depicted in figures 2.19 and 2.20. In particular, figure 2.20 reports both the optimal path of the trailer and the corresponding path of the truck.

It is useful for real-time applications to reduce the computational burden associated to problem (2.84). This can be achieved by adopting the heuristic

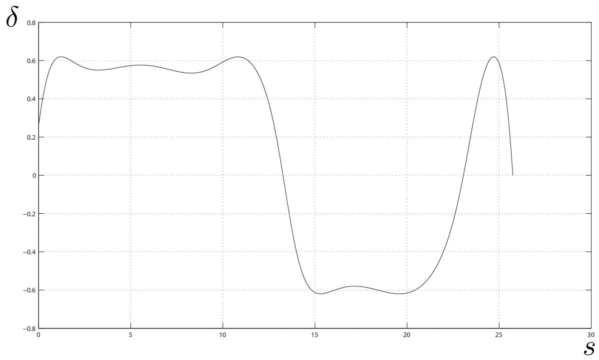


Figure 2.19: The optimal steering $\bar{\delta}(s)$ for problem (2.84).

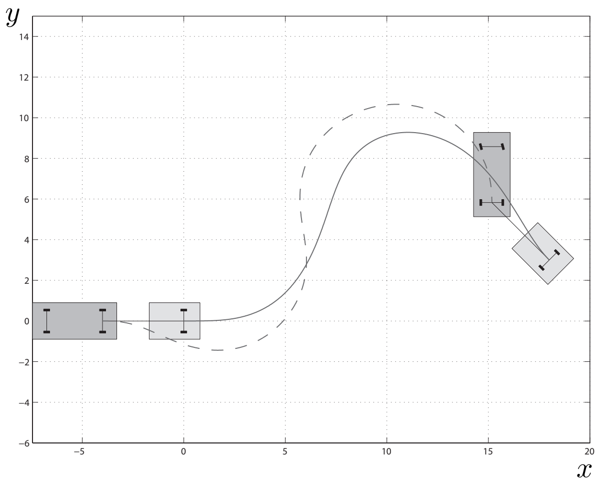


Figure 2.20: The optimal maneuver paths for problem (2.84).

of setting to zero all the eta parameters from η_3 to η_8 [30, 32, 33]. In such a way, the considered problem becomes

$$\min_{\ddot{\kappa}_A,\ddot{\kappa}_B\in\mathbb{R},\,\eta_1>0,\,\eta_2>0} \delta_{\max}\,. \tag{2.86}$$

The found solution for problem (2.86) is $\bar{\delta}_{\max} = 0.7309$ (or 41°.88) which corresponds to the parameters $\bar{\ddot{\kappa}}_A = -0.0353$ m^{-2}, $\bar{\ddot{\kappa}}_B = -0.0825$ m^{-2}, $\bar{\eta}_1 = 30.33$, and $\bar{\eta}_2 = 17.31$; the total spline length is $\bar{s}_f = 24.20$ m (see figures 2.21 and 2.22).

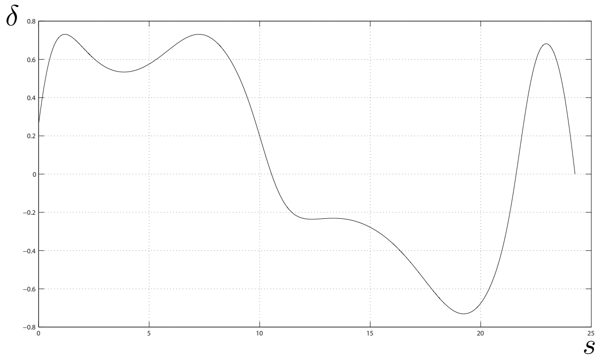


Figure 2.21: The optimal steering $\bar{\delta}(s)$ for problem (2.86).

Another index which can be appropriately minimized is the total spline length s_f and also considering the previous index $\delta_{\max}$, it emerges a multi-objective optimization that can be posed as follows:

$$\min_{\ddot{\kappa}_A,\ddot{\kappa}_B\in\mathbb{R},\,\boldsymbol{\eta}\in\mathcal{H}} \{\lambda_1\delta_{\max} + \lambda_2 s_f\} \tag{2.87}$$

where $\lambda_1, \lambda_2 \geq 0$ and $\lambda_1 + \lambda_2 = 1$. The weight coefficients λ_1 and λ_2 can be freely chosen to set a trade-off between $\delta_{\max}$ and s_f. For example by choosing $\lambda_1 = 0.95$ and $\lambda_2 = 0.05$, the found solution for (2.87) is the following: $\lambda_1\bar{\delta}_{\max} + \lambda_2\bar{s}_f = 1.836$ with $\bar{\delta}_{\max} = 0.6761$ (or 38°.74), $\bar{s}_f = 23.88$ m. The corresponding optimal parameters are $\bar{\ddot{\kappa}}_A = -0.0642$ m^{-2}, $\bar{\ddot{\kappa}}_B = -0.1401$ m^{-2}, $\bar{\eta}_1 = 28.14$, $\bar{\eta}_2 = 23.18$, $\bar{\eta}_3 = 19.28$, $\bar{\eta}_4 = 1.054$, $\bar{\eta}_5 = -737.7$, $\bar{\eta}_6 = -18.08$,

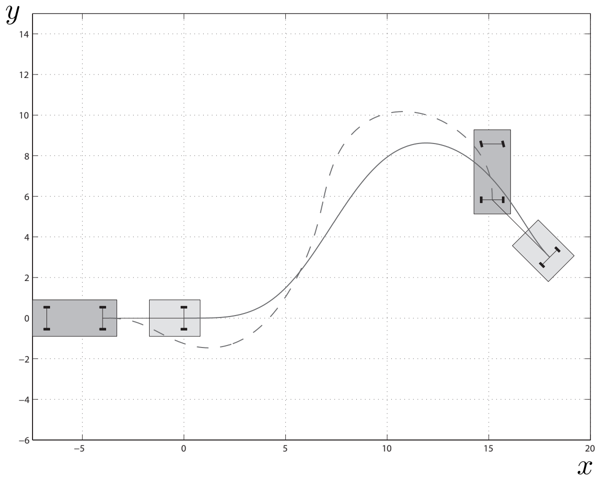


Figure 2.22: The optimal maneuver paths for problem (2.86).

$\bar{\eta}_7 = 885.3$, and $\bar{\eta}_8 = 346.3$. It is worth noting that the optimal distance $\bar{s}_f$ of multi-optimization (2.87) is reduced (that is, improved) of 7.2% with respect to the $\bar{s}_f$ of the single optimization (2.84) whereas $\bar{\delta}_{\mathrm{max}}$ of (2.87) is increased of 9.1% with respect to the $\bar{\delta}_{\mathrm{max}}$ of (2.84). This may be useful as far as the increasing of $\bar{\delta}_{\mathrm{max}}$ is compatible with the mechanical limit of the truck and trailer steering angle. As previously done for the single optimization, the simplified multi-optimization problem is

$$\min_{\dddot{\kappa}_A, \dddot{\kappa}_B \in \mathbb{R},\, \eta_1 > 0,\, \eta_2 > 0} \{\lambda_1 \delta_{\mathrm{max}} + \lambda_2 s_f\} \,. \tag{2.88}$$

With $\lambda_1 = 0.95$ and $\lambda_2 = 0.05$ the solution of (2.88) is $\lambda_1 \bar{\delta}_{\mathrm{max}} + \lambda_2 \bar{s}_f = 1.873$ with $\bar{\delta}_{\mathrm{max}} = 0.7564$ (or 43°.34), $\bar{s}_f = 23.09$ m. The corresponding parameters are $\dddot{\kappa}_A = -0.0341$ m^{-2}, $\dddot{\kappa}_B = -0.1069$ m^{-2}, $\bar{\eta}_1 = 26.05$, and $\bar{\eta}_2 = 15.78$ (see figures 2.23 and 2.24).

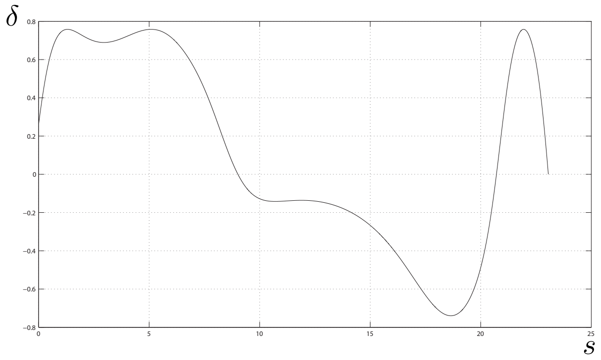


Figure 2.23: The optimal steering $\bar{\delta}(s)$ for multi-optimization problem (2.88)

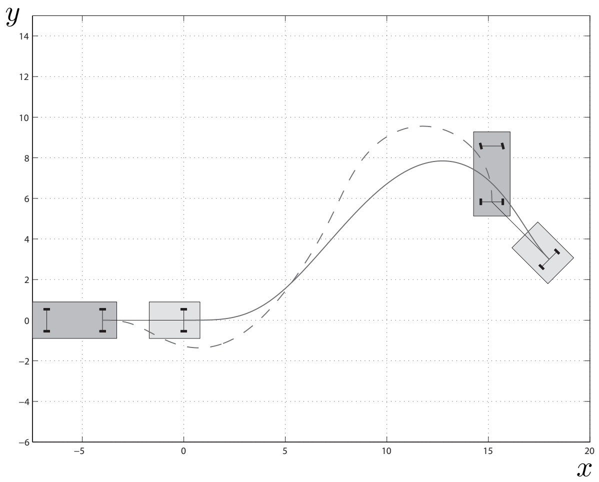


Figure 2.24: The optimal maneuver paths for multi-optimization problem (2.88)

Chapter 3

Time-optimal dynamic path inversion

Strong lives are motivated
by dynamic purposes

– Kenneth Hildebrand

Nowadays, the handling of materials and parts through Automatic Guided Vehicles (AGVs) is of increasing importance in the automation and logistics of factories and warehouses. The absence of human intervention in the normal operations of the AGVs permits to optimize by design the performances and specifically to pursue a motion planning to achieve fastest movements with full respect of all the pertinent constraints [42, 43]. Considering the more general scenario of trajectory planning of wheeled mobile robots, the basic problem of minimum-time planning between two robot configurations has been addressed with 1) unobstructed environments and 2) obstructed environments with obstacle to be avoided (respectively cf. [44] and [45] and references therein reported). The former case has been mainly dealt with the Pontryagin Maximum Principle (PMP) whereas with the latter, that is more difficult, a variety of sub-optimal or approximating techniques has been proposed e.g. potential functions, sampling methods such as Probabilistic Road Maps (PRMs), Rapidly-Exploring Dense Trees (RDTs), etc. Focusing on the special case of time-optimal (or minimum-time) trajectory planning on specific, desired paths, the use of the path-velocity decomposition [3] permits to reduce the planning to a suitable optimal velocity problem. This was the approach pursued by Prado et al. [46] who presented a sub-optimal method based on path segmentation to achieve a smooth velocity planning suitable for both static and dynamic environments.

This chapter presents a solution for the problem of time-optimal trajectory planning of an AGV on a given feasible path while respecting velocity, acceleration and jerk constraints. Moreover, this planning must connect two arbitrary dynamic configurations of the AGV, i.e. at the start and at the end of the planning the AGV may not be at rest. A key to solve the problem is to recast it as a dynamic path inversion problem.

3.1 Introduction to dynamic inversion

The dynamic inversion technique has been recently developed for the synthesis of high performance control systems [47, 48]. The idea behind this method is the inversion of the dynamical system in order to find an input that generates the desired output. Figure 3.1 shows a possible control scheme based on system inversion, which is essentially an open-loop control.

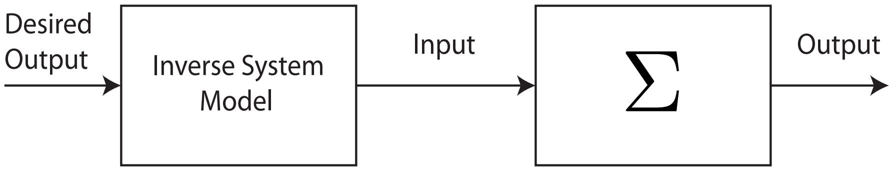


Figure 3.1: Dynamic inversion based control.

In many cases, once the desired output is known in advance, it is possible to perform a stable inversion, i.e. to determine a corresponding bounded (noncausal) input. The actual control system design can be centered on a feedforward/feedback scheme (see figure 3.2) where the feedforward control is determined through stable inversion and a feedback regulator handles modeling and signal errors [49]. The majority of the works pursuing this approach deals with nonlinear and nonminimum-phase systems and the emphasis is on algorithmic procedures to perform a stable inversion on a given output function.

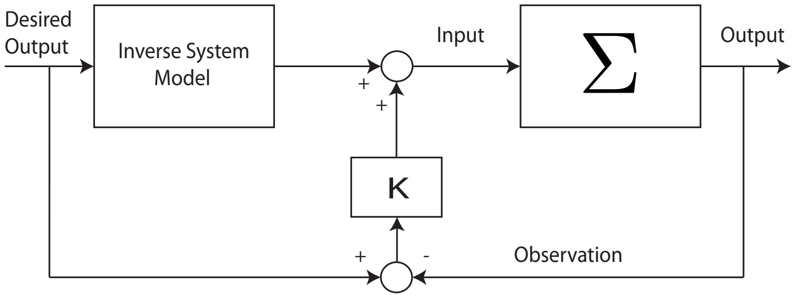


Figure 3.2: Feedforward/feedback control scheme.

3.1.1 Input-output dynamic path inversion

For a wide class of dynamic systems, the inversion problem can also be posed as a stable dynamic *input-output path inversion*. Dynamic path inversion, which was introduced in [5], is the problem, given a desired path on the output space, of finding the control inputs that generate the desired path. We said above that it is a variant of the more studied dynamic (signal) inversion which is the problem of finding the control inputs that generate the desired signal outputs [49–51].

The idea is to consider the output signal $y(t)$ as a function parametrization of a path Γ in the output space $\mathbb{R}^p$. For a given time interval $[0, T]$, the path Γ is defined as the image of output function (i.e. $\Gamma = y([0, t_1])$). This problem can be formally stated as follows:

Problem 4 *Given a path* $\Gamma \subset \mathbb{R}^p$ *and a traveling time* $T > 0$*, find initial conditions and input* $u(t)$ *for which the system output* $y(t)$ *satisfies*

$$y([0, T]) = \Gamma$$

This problem is quite general and especially relevant for the motion control of nonholonomic wheeled vehicles, and it has a strong connection with *differential flatness* [52, 53].

Roughly speaking, a system with n scalar inputs is said to be differentially flat if there exist n outputs $y_1, \ldots, y_n$ for which the system variables (i.e. the states and the inputs) can be algebraically expressed as functions of the outputs and their derivatives, till a finite order. A more rigorously definition of the flatness, is given in the next chapter. When the system is differentially flat, the dynamic path inversion problem is relatively easy to solve.

3.2 Time-optimal dynamic path inversion for an automatic guided vehicle

This section presents the work appeared in [31], which faces time-optimal trajectory planning of an automatic guided vehicle (AGV) on a given feasible path while respecting velocity, acceleration and jerk constraints. A theoretical result shows the connection, for the AGV, between the geometric continuity of its paths and the smoothness of its control inputs (linear velocity and steering angle of the AGV motor wheel). The solution hence proposed for the optimal planning is based on a dynamic path inversion algorithm, for which first the optimal velocity profile is determined and then the optimal steering signal is derived from a geometrical construction.

3.2.1 Kinematic model and problem statement

A typical wheeled automatic guided vehicle has forks for handling materials, two passive wheels and a motor wheel. See figure 3.3 where a schematic plan

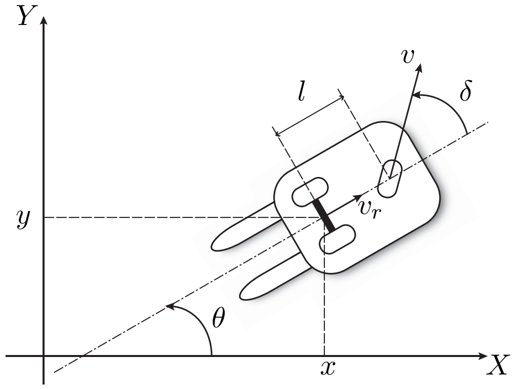


Figure 3.3: A wheeled AGV on a Cartesian plane.

view of an AGV and a Cartesian reference frame are depicted. As usual, x and y indicate the Cartesian coordinates of the AGV rear-axle middle-point and θ is the vehicle orientation angle with respect of the x-axis. The motion of the AGV is actuated by the forward motor wheel whose linear velocity is v and δ is the steering angle; l is the distance between the rear-axle and the forward wheel's hub. With the usual modeling assumptions of no-slippage, rigid body and nonholonomic constraints the following nonlinear kinematic model of the AGV can be deduced [54]:

$$\begin{cases} \dot{x}(t) = v(t)\,\cos\theta(t)\,\cos\delta(t) \\ \dot{y}(t) = v(t)\,\sin\theta(t)\,\cos\delta(t) \\ \dot{\theta}(t) = \frac{1}{l}\,v(t)\,\sin\delta(t)\,. \end{cases} \tag{3.1}$$

The linear velocity v and the steering angle δ are the AGV control inputs. The following definition will be used along this chapter:

Definition 7 *A Cartesian path* Γ *has third order geometric continuity, and we say* Γ *is a* G^3*-path, if its scalar curvature is continuous and the derivative with respect to the arc length of the curvature is continuous on the path too (for more details see* [5]*).*

In order to obtain a smooth motion control, inputs v and δ must be functions with C^1 continuity, i.e. they are continuous functions with continuous derivatives. A connections between smooth inputs and paths of the AGV is established by proposition below (recall proposition 3).

Proposition 6 *Assign any $T > 0$. If a Cartesian path Γ is generated by the AGV with inputs $v(t), \delta(t) \in C^1([0, T])$ where $v(t) \neq 0$ and $|\delta(t)| < \frac{\pi}{2}$, $\forall t \in [0, T]$, then Γ is a G^3-path. Conversely, given any G^3-path Γ then exist inputs $v(t), \delta(t) \in C^1([0, T])$ with $v(t) \neq 0$ and $|\delta(t)| < \frac{\pi}{2}$, $\forall t \in [0, T]$, and initial conditions such that the path generated by the AGV coincides with the given Γ.*

Proof. proposition above can be deduced by a similar result proposed by Guarino Lo Bianco *et al.* in [5]. □

Instrumental to our approach to optimal motion control of AGVs is the definition of an "extended state" of system (3.1) that also comprises the control functions and their first derivatives:

$$\left\{x(t), y(t), \theta(t), v(t), \dot{v}(t), \delta(t), \dot{\delta}(t)\right\} .$$

Then, the following time-optimal dynamic path inversion (TOPI) problem can be posed.

Problem 5 (TOPI problem) *Given an assigned G^3-path Γ, determine the control functions $v(t)$, $\delta(t) \in PC^2$ (see definition 2) such that system (3.1) travels exactly on path Γ in minimum-time $\bar{t}_f$ from initial extended state (at time $t = 0$)*

$$\mathcal{A} := \left\{x_A, y_A, \theta_A, v_A, \dot{v}_A, \delta_A, \dot{\delta}_A\right\} ,$$

to final extended state (at time $t = \bar{t}_f$)

$$\mathcal{B} := \left\{x_B, y_B, \theta_B, v_B, \dot{v}_B, \delta_B, \dot{\delta}_B\right\} ,$$

satisfying the following constraints

$$0 \leq v(t) \leq v_M , \quad \forall t \in [0, \bar{t}_f] ,$$

$$|\dot{v}(t)| \leq a_M \,, \quad \forall t \in [0, \bar{t}_f] \,,$$
$$|\ddot{v}(t)| \leq j_M \,, \quad \forall t \in [0, \bar{t}_f] \,,$$

where $v_M, a_M, j_M > 0$ *are given bounds.*

Hence, in order to give a solution to TOPI problem, it is preliminarily necessary to determine a desired G^3-path that satisfies the interpolating data deduced from the extended states $\mathcal{A}$ and $\mathcal{B}$ [32]. Let us introduce the following relations

$$v_r(t) = v(t)\, \cos\delta(t)\,, \tag{3.2}$$

$$\dot{v}_r(t) = \dot{v}(t)\, \cos\delta(t) - v(t)\, \dot{\delta}(t)\, \sin\delta(t)\,, \tag{3.3}$$

$$\omega(t) = \frac{1}{l}\, v(t)\, \sin\delta(t)\,, \tag{3.4}$$

$$\dot{\omega}(t) = \frac{1}{l}\, \dot{v}(t)\, \sin\delta(t) + \frac{1}{l}\, v(t)\, \dot{\delta}(t)\, \cos\delta(t)\,, \tag{3.5}$$

where $v_r(t)$ and $\dot{v}_r(t)$ denote the linear velocity and acceleration of the AGV rear-axle middle-point, and $\omega(t)$ and $\dot{\omega}(t)$ denote the angular velocity and acceleration of the AGV. From [5], the curvature and its derivative with respect to the arclength, k_A and $\dot{k}_A$ in $t = 0$, and k_B and $\dot{k}_B$ in $t = \bar{t}_f$, are given by

$$k_A = \frac{\omega_A}{v_{rA}}\,, \qquad \dot{k}_A = \frac{\dot{\omega}_A\, v_{rA} - \omega_A\, \dot{v}_{rA}}{v_{rA}^3}\,, \tag{3.6}$$

and

$$k_B = \frac{\omega_B}{v_{rB}}\,, \qquad \dot{k}_B = \frac{\dot{\omega}_B\, v_{rB} - \omega_B\, \dot{v}_{rB}}{v_{rB}^3}\,, \tag{3.7}$$

where $\omega_A = \omega(0)$, $v_{rA} = v_r(0)$, $\omega_B = \omega(\bar{t}_f)$ and $v_{rB} = v_r(\bar{t}_f)$. By substituting relations (3.2)-(3.5) in (3.6)-(3.7), the following equations are obtained

$$k_A = \frac{1}{l}\, \tan\delta_A\,, \tag{3.8}$$

$$\dot{k}_A = \frac{1}{l}\, \frac{\dot{\delta}_A}{v_A\, \cos^3\delta_A}\,, \tag{3.9}$$

and

$$k_B = \frac{1}{l}\, \tan\delta_B\,, \tag{3.10}$$

$$\dot{k}_B = \frac{1}{l} \, \frac{\dot{\delta}_B}{v_B \, \cos^3 \delta_B} \, . \tag{3.11}$$

On the extended states $\mathcal{A}$ and $\mathcal{B}$ we impose the assumptions

$$|\delta_A| < \pi/2 \, , \quad \text{and} \quad |\delta_B| < \pi/2 \, .$$

Therefore, relations (3.9) and (3.11) indicate that there exist two definite forbidden cases

$$\{v_A = 0\} \wedge \{\dot{\delta}_A \neq 0\} \, , \qquad \{v_B = 0\} \wedge \{\dot{\delta}_B \neq 0\} \, ,$$

which are considered as further assumptions on the TOPI problem. On the other hand if $v_A = 0$ and $\dot{\delta}_A = 0$ and similarly $v_B = 0$ and $\dot{\delta}_B = 0$, then $\dot{k}_A$ and $\dot{k}_B$ can be arbitrarily assigned and this improves the design freedom in shaping the Γ path for the AGV.

Hence, the G^3-path Γ must satisfy at the endpoints the interpolations conditions shown in figure 3.4, i.e. the initial and final Cartesian points of Γ have (x_A, y_A) and (x_B, y_B) as coordinates, θ_A and θ_B as unit-tangent directions, k_A and k_B as curvatures, $\dot{k}_A$ and $\dot{k}_B$ as curvature derivatives respectively.

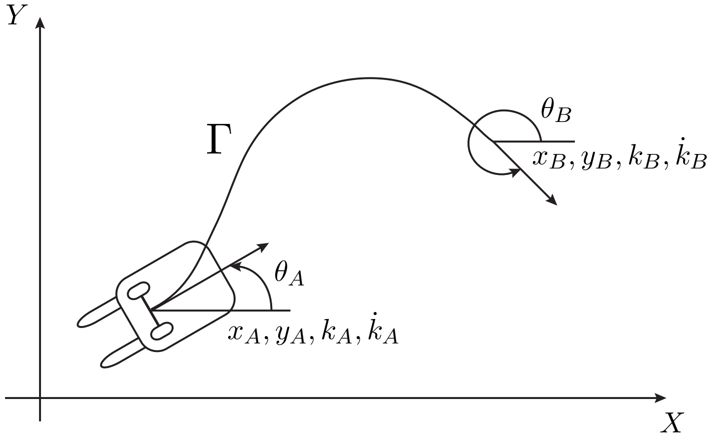


Figure 3.4: The interpolations conditions at the endpoints of path Γ.

This interpolation problem can be easily solved using the $\boldsymbol{\eta}^3$-splines [32,33], introduced in the precedent chapter, which are seventh-order polynomial curves with free design parameters (the $\boldsymbol{\eta}$ vector) to shape the desired path intercourse between the endpoints.

Remark In the following subsections, path Γ denotes the Cartesian path generated by the rear-axle middle-point, i.e. by $(x(t), y(t))$. Below another relevant path of the AGV, denoted by Γ_f, is introduced. Γ_f is the path generated by the AGV forward motor wheel.

3.2.2 The dynamic path inversion algorithm

The time-optimal control functions $\bar{v}(t)$ and $\bar{\delta}(t)$, which permit the AGV to follow the given path Γ in minimum-time, will be obtained by a dynamic path inversion procedure.

Note that functions $\bar{v}(t)$ and $\bar{\delta}(t)$, solution of the TOPI problem, are associated to the actuated motor wheel of the AGV (see figure 3.3), so that the inversion procedure will need to determine the path Γ_f of the forward wheel which is geometrically linked to Γ. Knowledge of Γ_f and its total distance s_f allows to apply the path-velocity decomposition method [3] to the TOPI problem so that the velocity $\bar{v}(t)$ will be computed independently from $\bar{\delta}(t)$ by setting the minimum-time constrained velocity planning discussed in section 1.4. Then the optimal steering $\bar{\delta}(t)$ will be determined by exploiting the geometric properties of model (3.1) relative to paths Γ and Γ_f.

The dynamic path inversion algorithm can be then described in the following three steps:

1. Determine the path Γ_f of the forward wheel. Consider the following parametrization of path Γ (as customary using $\boldsymbol{\eta}^3$-splines)

$$\begin{aligned} \mathbf{p}(u):\quad & [0,1] \to \mathbb{R}^2 \\ & u \to \mathbf{p}(u)\,. \end{aligned}$$

The unit tangent vector $\boldsymbol{\tau}(u)$ of Γ is given by

$$\boldsymbol{\tau}(u) = \frac{\dot{\mathbf{p}}(u)}{\|\dot{\mathbf{p}}(u)\|}\,,$$

and a parametrization of path Γ_f can be obtained as follows

$$\mathbf{p}_f(u) = \mathbf{p}(u) + l\ \boldsymbol{\tau}(u),\ u \in [0,1], \tag{3.12}$$

where l is the distance between the rear-axle middle point and the forward wheel. Figure 3.5 depicts the geometric relation between paths Γ and Γ_f.

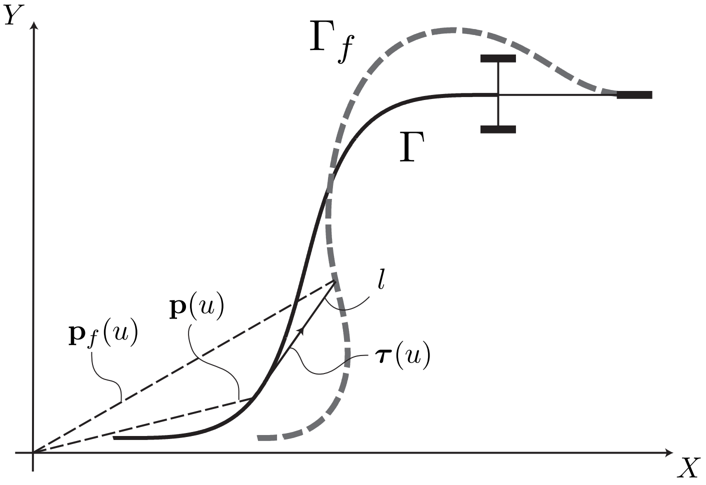


Figure 3.5: Geometric construction of the forward path Γ_f.

Compute the total distance s_f to be travelled by the forward wheel on Γ_f:

$$s_f = \int_0^1 \|\dot{\mathbf{p}}_f(u)\| du\,. \tag{3.13}$$

2. Determine the minimum-time velocity $\bar{v}(t)$ by solving the constrained problem widely exposed in section 1.4.

3. Determine the optimal steering function $\bar{\delta}(t)$ by solving the following equation system:

$$\begin{cases} \int_0^t \bar{v}(\xi) d\xi = \int_0^u \|\dot{\mathbf{p}}_f(\xi)\| d\xi \\ \bar{\delta}(t) = \arg \boldsymbol{\tau}_f(u) - \arg \boldsymbol{\tau}(u) \, . \end{cases} \tag{3.14}$$

The geometrical meaning of this determination is depicted in figure 3.6.

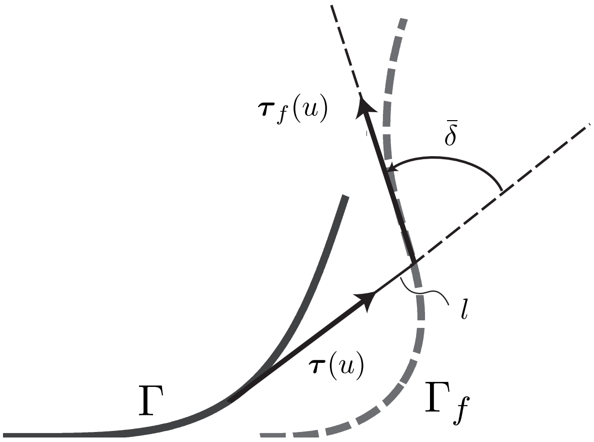


Figure 3.6: Geometrical interpretation of equation system (3.14).

Remark The velocity planning problem leads to a smooth velocity profile (see section 1.4) that is easy to implement on an actuator drive because velocity and acceleration are continuous and the jerk (the time-derivative of acceleration) is limited and constrained as desired (by setting the bound j_M). Also note that the constraint $v(t) \geq 0$ imposes that the automatic guided vehicle does not go backward on the desired path.

3.2.3 Example

Consider an AGV with $l = 1.1$ m, the distance between the motor wheel and the rear-axle, and constraints on the actuation of the motor wheel given by

$v_M = 3$ m/s, $a_M = 1$ m/s^2, and $j_M = 0.5$ m/s^3.

It is desired a minimum-time transition between the extended states $\mathcal{A}$ and $\mathcal{B}$ given by (measures are expressed in m, m/s, m/s^2, rad, rad/s):

$$\mathcal{A} = \left\{x_A, y_A, \theta_A, v_A, a_A, \delta_A, \dot{\delta}_A\right\} = \{0, 0, 0, 1, -1, 0, 0\} \ .$$

$$\mathcal{B} = \left\{x_B, y_B, \theta_B, v_B, a_B, \delta_B, \dot{\delta}_B\right\} = \{16, 8, 0, 3, 0, 0, 0\} \ .$$

The desired Cartesian path Γ between (x_A, y_A) and (x_B, y_B) is an S-shaped path that can be easily determined by interpolation with the $\boldsymbol{\eta}^3$-splines [32]. The interpolation data are $(\theta_A, k_A, \dot{k}_A)$ and $(\theta_B, k_B, \dot{k}_B)$ for which $\theta_A = 0$ and $\theta_B = 0$ from the assigned extended states $\mathcal{A}$ and $\mathcal{B}$ and $k_A = 0$, $\dot{k}_A = 0$ and $k_B = 0$, $\dot{k}_B = 0$ as it follows from relations (3.8)-(3.11).

Path Γ is then an $\boldsymbol{\eta}^3$-spline, a seventh order polynomial curve, whose free parameters are chosen according to the heuristic rule suggested in [30, 32]:

$$\boldsymbol{\eta} = (\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6) = (d, d, 0, 0, 0, 0) \, ,$$

where $d = \|(x_A - x_B, y_A - y_B)\| = 17.89$ is the Euclidean distance between (x_A, y_A) and (x_B, y_B). Path Γ is the blue one depicted in figure 3.7.

To determine the time-optimal controls $\bar{v}(t)$ and $\bar{\delta}(t)$ which are the solution to the TOPI problem we use the dynamic path inversion algorithm described in three steps in subsection 3.2.2.

Step 1: The path Γ_f of the forward motor wheel is computed according to (3.12). It is depicted in figure 3.7. The length of Γ_f is $s_f = 19.12$ m according to (3.13).

Step 2: The existence of the time-optimal velocity is guaranteed by the fulfilment of the sufficient conditions of proposition 2. Indeed, conditions (1.37) and (1.38) are immediately satisfied. Because $a_A < 0$ and $a_B \geq 0$, we check conditions (1.40) and (1.41) respectively: $v_A + \frac{1}{2}\frac{a_A^2}{j_M} = 0 \geq 0$ and $v_B - \frac{1}{2}\frac{a_B^2}{j_M} = 3 \geq 0$. Application of the four-step procedure of proposition 2 determines $s_{ref} = 8.17$ m so that the last inequality (1.43) is also satisfied: $s_f \geq s_{ref}$. Hence, since

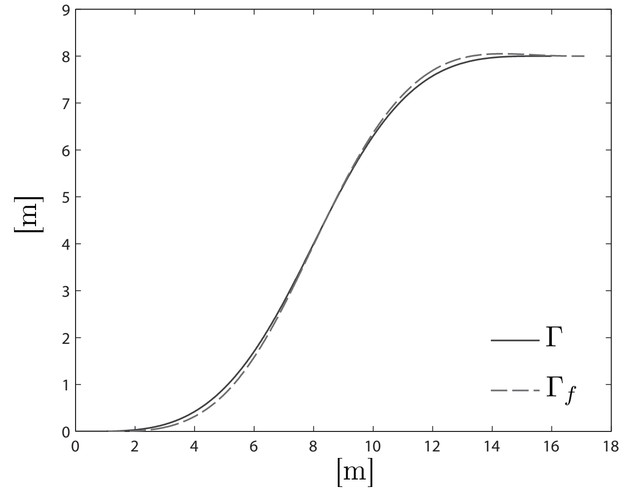


Figure 3.7: The planned path Γ and the associated forward path Γ_f of the AGV.

the constrained minimum-time velocity problem has solution, the TOPI problem has solution too by virtue of the path dynamic inversion algorithm of subsection 3.2.2.

The approximated determination of $\bar{v}(t)$ is gained with the procedure detailed in subsection 1.4.2 and its profile is shown in figure 3.8. It has been chosen the sampling time $T = 0.01$ s and the linear programming routine has run using MOSEK [55]. The resulting minimum-time for the transition of the AGV from $\mathcal{A}$ to $\mathcal{B}$ along Γ is $\bar{t}_f = 10.64$ s.

Step 3: The optimal steering control $\bar{\delta}(t)$ is determined by solving (3.14) with a sweeping discretization on parameter $u \in [0, 1]$. The result is shown in figure 3.9. Many simulations of the motion of the AGV have been performed. In particular, the minimum-time transition of the AGV between the extended states $\mathcal{A}$ and $\mathcal{B}$ along Γ has been simulated by using the found $\bar{v}(t)$ and $\bar{\delta}(t)$. The adopted approximations are good enough to ensure a tracking of the planned trajectory with negligible errors.

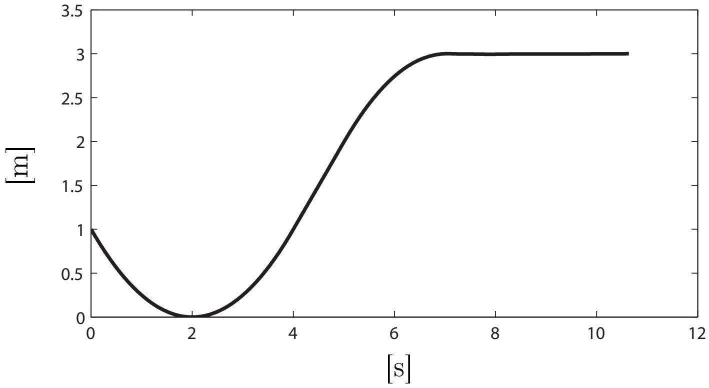


Figure 3.8: The optimal velocity profile $\bar{v}(t)$.

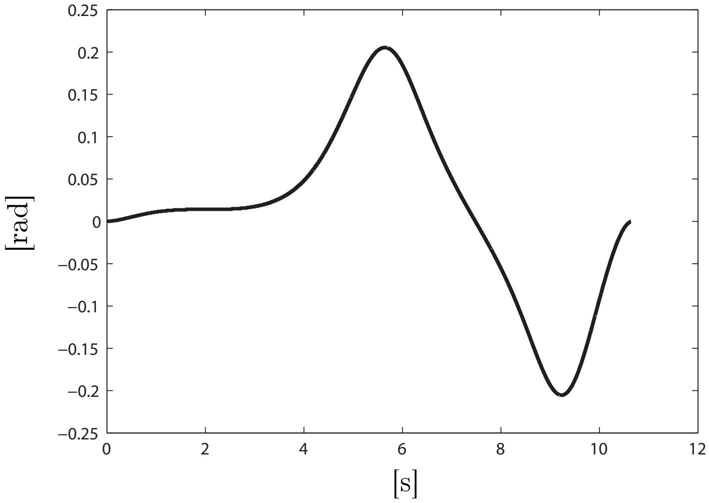


Figure 3.9: The optimal steering control $\bar{\delta}(t)$.

Chapter 4

Replanning methods for the trajectory tracking

It is a bad plan that admits
of no modification.

– Publilius Syrus

As known, a fundamental problem in control theory for automation is output tracking [56]. Given a desired signal or reference on the output variable of a controlled dynamical system, the problem is to appropriately manipulate the input of the system in such a way that the actual output follows as close as possible the desired reference. The classic solution approach prescribes the design of a feedback controller that can asymptotically zero the tracking error [57–59]. When the desired output is known in advance, an alternative tracking control strategy is inversion-based control. It is a feedforward/feedback strategy where the feedforward is determined by stable input-output inversion and the feedback is activated by a controller whose input is the error between the reference state and the actual state [49, 50, 60]. A variant of this strategy considers the application of a feedback controller first to reduce the effects of unmodeled dynamics or uncertainties on the controlled system and then the closed-loop dynamics is inverted by the stable inversion procedure [61].

Both the classic approach to output tracking and the newer inversion-based one expect the continuous-time availability of the measured output or the measured state of the controlled system. However, there are cases where continuous-time or high-frequency revelation of the system state or output is not possible or not economical and only low-frequency feedback is practicable. The resulting control framework is then an hybrid feedforward/feedback scheme where the controlled system is commanded by feedforward (i.e. open-loop) inputs that are periodically updated to compensate or reduce the tracking error. This paradigm has been pioneered in [62,63] for the robust stabilization of nonlinear driftless and chained systems; an application was also proposed for the lane following control of a vision-based autonomous car [30].

In the first section, within the framework of hybrid feedforward/feedback control schemes we propose a trajectory tracking problem of a WMR modeled by a unicycle model affected by norm-bound noise (cf. [64]). Given a desired, feasible Cartesian trajectory to be tracked by the WMR, the proposed control scheme uses a recursive convex replanning method to compute a new reference trajectory whenever the WMR's state is real-time available at a frequency assigned by the replanning time period T (cf. section 4.1.1). Then, this new reference trajectory that is still feasible is used to generate the feedforward inverse command velocities that help in reducing the tracking errors (see figure 4.5). If the replanning period T is sufficiently small relative to the noise magnitude, explicit closed-form bounds on the global tracking error are provided (cf. corollary 1). In such a way a "practical" tracking convergence to the desired trajectory is achieved.

Second section presents the output tracking of a nonlinear flat system affected by additive noise on its state derivative (cf. [65, 66]). More specifically, we consider a controlled system whose performance output is a flat output of the system itself [67]. A desired output signal is sought on the actual output by using a feedforward inverse input that is periodically updated using the

observation of the full system state acquired at intervals of period T. The proposed method is actually an iterative output replanning that uses the desired output trajectory and the updated state to replan an output trajectory whose inverse input helps in reducing the tracking error. This iterative replanning exploits the Hermite interpolating polynomials to achieve an overall arbitrarily smooth input and a tracking error that can be made arbitrarily small under mild assumptions if the sampling period T is sufficiently small.
Notation: If x is a $\mathcal{C}^l(\mathbb{R}, \mathbb{R}^n)$ function, $x^{(l)}$ denotes the derivative of x of order l, $x(t_0^+)$ and $x(t_0^-)$ denotes, respectively, $\lim_{x \to t_0^+} x(t)$ and $\lim_{x \to t_0^-} x(t)$. For any vector $v \in \mathbb{R}^n$, $(v)_i$ denotes its i-th component.

4.1 Recursive convex replanning

This section considers the Cartesian trajectory tracking of wheeled mobile robots to be performed by an hybrid control scheme with feedforward inverse control and a state feedaback that is only updated periodically and relies on a recursive convex replanning of the reference trajectory. This approach applied to the standard unicycle model is shown to maintain its efficacy also in presence of noise or unmodeled robot dynamics. Explicit, sufficient conditions are provided to ensure global boundedness of the tracking error. Finally, experimental results are presented using Lego Mindstorm mobile robots.

4.1.1 Trajectory tracking for the unicycle

Here, the recursive tracking approach discussed in this section is presented in the case of the kinematic unicycle. Consider the following model for the unicycle (see figure 4.1)

$$\left\{ \begin{array}{rcl} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} & = & v(t) \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} + \eta(t) \\ \dot{\theta} & = & \omega(t) + \eta_\theta(t)\,, \end{array} \right. \tag{4.1}$$

where $(x, y) \in \mathbb{R}^2$ is the position of the center of the unicycle, θ is the orientation angle and v, ω are the velocity control inputs, and set $z = (x, y, \theta)$. Functions η and η_θ are noise terms that satisfy the bounds $\forall t \in \mathbb{R}$

$$\begin{cases} \|\eta(t)\| & \leq & N\,, \\ |\eta_\theta(t)| & \leq & N_\theta\,. \end{cases} \tag{4.2}$$

When the noise terms are not present, (i.e. $N = 0$ and $N_\theta = 0$) system (4.1) is called the *nominal unicycle.*

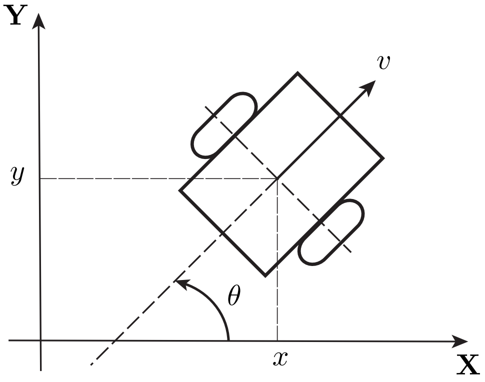


Figure 4.1: Schematic of a unicycle mobile robot.

Consider a reference trajectory γ_0 defined as follows.

Assumption 1 *Let* $\gamma_0 : \mathbb{R}_+ \to \mathbb{R}^2$ *be a reference trajectory with* C^3 *continuity such that:*

a) $0 < V_m \leq \|\dot{\gamma_0}(t)\| \leq V_M\,,$

b) $\|\ddot{\gamma_0}(t)\| \leq A_M\,.$

Exact tracking of γ_0 is achieved when, $\forall t \geq 0$,

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \gamma_0(t)\,.$$

The following straightforward result characterizes completely the exact tracking problem for the nominal unicycle.

Property 4 *Exact tracking is achieved for the nominal unicycle* (4.1)*, i.e.*

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \gamma_0(t)\,, \quad \forall t \geq 0\,,$$

if and only if the following conditions hold:

a) $\begin{pmatrix} x(0) \\ y(0) \end{pmatrix} = \gamma_0(0)\,,$

b) $v(0) \begin{pmatrix} \cos\theta(0) \\ \sin\theta(0) \end{pmatrix} = \dot{\gamma}_0(0)\,,$

c) $v(t) = \|\dot{\gamma}_0(t)\|, \forall t \geq 0\,,$

d) $\omega(t) = \frac{d}{dt}\arg(\dot{\gamma}_0(t))\,, \forall t \geq 0\,.$

Conditions a), b) imply that the initial conditions must be such that at the initial time the unicycle lies at the beginning of the curve with orientation angle parallel to the tangent vector to the curve γ_0. Conditions c), d) actually define the controls that must be used to exactly track the given reference. These controls are *feedforward* velocity input signals because depend only on the reference γ_0.

Remark Having chosen a C^3-trajectory reference, i.e. a trajectory function that is continuous with its derivatives till to the third order, we obtain by means of c) and d) smooth velocities $v(t)$, $\omega(t)$ with continuous accelerations, i.e. $v, \omega \in C^1(\mathbb{R}_+)$. A weaker condition to still ensure continuous accelerations is to assume $\gamma_0 \in C^2(\mathbb{R}_+)$ and γ_0 is a G^3-curve, i.e. a curve with third order geometric continuity (continuity along the curve of the tangent vector, curvature, and derivative of the curvature with respect to the arc length) [5].

Obviously, using feedforward control only, defined by c) and d), the tracking error may grow unbounded if $N > 0$, $N_\theta > 0$. In order to keep the error bounded one may use continuous-time feedback control. In this section another approach is considered, based on an idea similar to *iterative steering* (see [4]). The method consists in using at all times the feedforward controls given by c), d) but the reference trajectory is periodically replanned. When $t \in [0,T]$, γ_0 is used as reference trajectory, for $t \in [T,2T]$ a different curve γ_1 is used and, in general the reference trajectory γ_i is used for $t \in [iT,(i+1)T]$. Each references γ_i is defined *recursively* with respect to γ_{i-1} in such a way to keep the tracking error limited. Before explaining in detail the overall feedforward/feedback strategy, the replanning operator to be used to construct each reference γ_i from γ_{i-1} is defined as follows:

Definition 8 (Replanning Operator) *Let be given a (current) reference trajectory $\gamma : [t_0,+\infty) \to \mathbb{R}^2$ and a robot's state $z_0 = (x_0, y_0, \theta_0)$. Define a new reference trajectory $\gamma_{z_0,t_0,\gamma} : [t_0,+\infty) \to \mathbb{R}^2$ according to the convex replanning:*

$$\begin{aligned}\gamma_{z_0,t_0,\gamma}(t) = \lambda(t-t_0)\,[(x_0,y_0) + R(e_\theta(t_0))(\gamma(t)-\gamma(t_0))] \\ +(1-\lambda(t-t_0))\,\gamma(t)\ ,\end{aligned} \tag{4.3}$$

where

- $\lambda : \mathbb{R}_+ \to [0,1]$ *is a monotone decreasing C^3-function with $\lambda(0) = 1$, $D^i\lambda(0) = 0$, $i = 1,2,3$ and $\lim_{t\to+\infty}\lambda(t) = 0$;*
- $R(x) = \begin{bmatrix} \cos x & -\sin x \\ \sin x & \cos x \end{bmatrix}$ *is the rotation matrix;*
- $e_\theta(t_0) = \theta_0 - \arg\dot\gamma(t_0)$ *is the heading angle error at time t_0.*

The curve $\gamma_1 = \gamma_{z_0,t_0,\gamma_0}$ is a C^3-function and enjoys the following properties

$$\gamma_1(t_0) = (x_0, y_0)\,,$$

$$\arg\dot\gamma_1(t_0) = \theta_0\,,$$

$$\lim_{t\to\infty} \gamma_1(t) - \gamma_0(t) = 0\,.$$

In other words, trajectory γ_1 at t_0 is equal to $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ and its derivative has the direction given by θ_0. Asymptotically γ_1 converges to γ_0 and the rate of convergence is controlled by the monotone decreasing function λ. Remark that the replanned curve γ_1 is determined trough a linear convex combination, weighted by $\lambda(t)$, of function γ_0 and another trajectory obtained by rotating and translating γ_0 itself, as depicted in figure 4.2. For instance, one may choose λ using C^3-transition polynomials [48] and setting the transition time equals to $2T$:

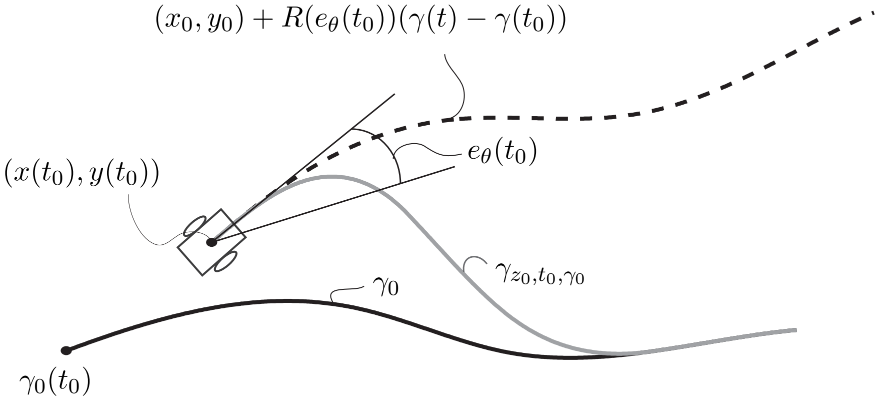

$(x_0, y_0) + R(e_\theta(t_0))(\gamma(t) - \gamma(t_0))$
$e_\theta(t_0)$
$(x(t_0), y(t_0))$
$\gamma_{z_0,t_0,\gamma_0}$
γ_0
$\gamma_0(t_0)$

Figure 4.2: Convex replanning.

$$\begin{aligned} \lambda(t) &= 1 - 35\left(\frac{t}{2T}\right)^4 + 84\left(\frac{t}{2T}\right)^5 \\ &- 70\left(\frac{t}{2T}\right)^6 + 20\left(\frac{t}{2T}\right)^7, \ t \in [0, 2T], \\ \lambda(t) &= 0, \ t > 2T. \end{aligned} \tag{4.4}$$

The graph of this function is reported in figure 4.3.

The motion control method can be summarized as follows (it is assumed that a), b) of property 4 hold).

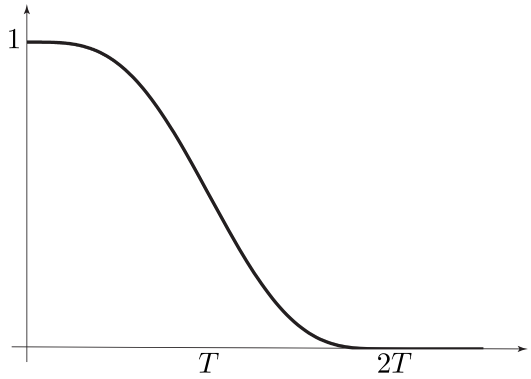


Figure 4.3: The C^3-transition polynomial $\lambda(t)$.

1) For $t \in [0, T]$, where $T > 0$ is the *replanning time*, the control functions are given by c), d) (in property 4)

2) For $t \in [iT, (i+1)T]$, with $i = 1, 2, \dots$, the control velocities are defined by

$$u(t) = \|\dot{\gamma}_i(t)\|\,, \tag{4.5}$$

$$\omega(t) = \frac{d}{dt}\arg(\dot{\gamma}_i(t))\,, \tag{4.6}$$

where $\gamma_i(t)$ is the trajectory determined via the convex replanning operator (4.3):

$$\gamma_i = \gamma_{z(iT), iT, \gamma_{i-1}}\,. \tag{4.7}$$

That is, for $t \in [iT, (i+1)T]$, an open loop control is applied, that would drive the nominal system from state $\begin{pmatrix} x(iT) \\ y(iT) \end{pmatrix}$ with orientation $\theta(iT)$, to reference trajectory γ_{i-1}. Therefore the reference trajectory γ_i is defined recursively with respect to trajectory γ_{i-1}, as shown in figure 4.4.

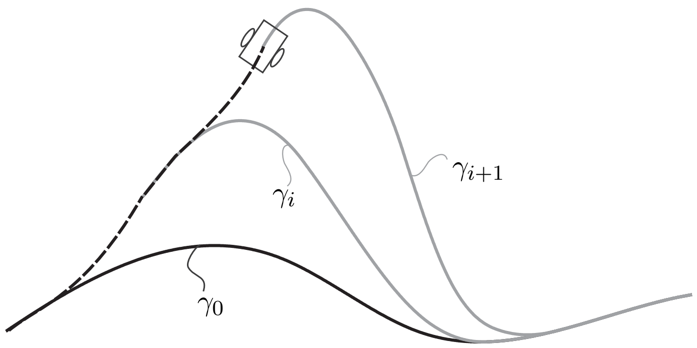


Figure 4.4: Recursive generation of reference trajectories.

The overall control scheme is depicted in figure 4.5 where the Recursive Convex Replanning Operator block takes care of the iterative trajectory generation and the Inverse Control Operator block computes the actual control by means of differential relations (4.5),(4.6).

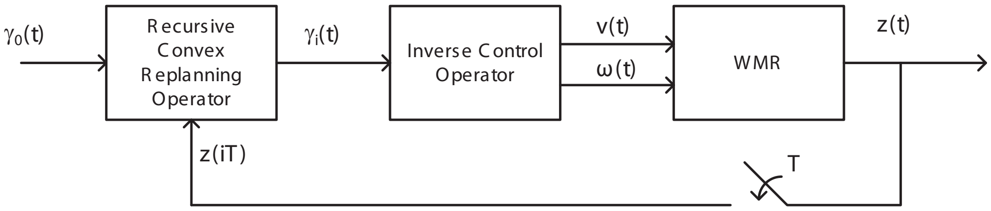


Figure 4.5: The hybrid feedforward/feedback scheme for the trajectory tracking of wheeled mobile robots.

The control method just outlined draws on the idea of iterative state steering (see [4]), the main difference lies in the fact that each replanned trajectory is defined recursively with respect to the previous one. With respect to the iterative state steering, this method has the following significant differences:

- if the noise functions η and η_θ affecting system (4.1) are zero during time interval $[iT, (i+1)T]$ the replanned trajectory coincides with the previous one, i.e. $\gamma_{i+1} = \gamma_1$. No replanning is actually performed in absence of noise.

- The replanning does not affect the control smoothness as ω and v remain C^1-functions, linear and angular accelerations remain continuous. Actually, these control functions could be made arbitrarily regular by choosing sufficiently regular reference γ_0 and function λ.
- Even if a direct comparison is difficult, the sufficient conditions for applying this method are somehow weaker than the one appearing in [4] since it is not required that the tracking error decreases in norm after the replanning time T (see (c) of assumption 1 in [4]).

In this section, this method will be analyzed, conditions will be found that allow keeping the tracking error limited and bounds will be provided. The problem that will be solved is therefore the following.

Problem 6 *Find conditions on trajectory γ_0, replanning time T and noise magnitude that guarantee that the tracking error is bounded, and find an estimate on the error norm.*

In section 2.1.2, this problem will be considered for a general class of systems that includes the unicycle model (4.1). The main result of this work (proposition 7), when applied to the case of the unicycle vehicle with function λ defined as in (4.5), brings to the following result.

Corollary 1 *Consider control laws given by* (4.5) *and* (4.6) *and let $\lambda(t)$ be given by* (4.4). *If $T < \frac{32}{83 N_\theta}$ then the following bounds hold*

$$\|\dot\gamma_i(t) - \dot\gamma_0(t)\| \leq \bar V_2 := \frac{\frac{83}{32}\, T\, N_\theta\, V_M + \left(\frac{T}{2}^2\, N_\theta + T\, N\right)}{1 - \frac{83}{32}\, T\, N_\theta}\,, \tag{4.8}$$

$$\|\gamma_i(t) - \gamma_0(t)\| \leq \left(1 + \frac{T}{4}\right) T\, N_\theta(\bar V_2 + V_M) + \frac{T}{2} N\,. \tag{4.9}$$

This result means that if the product of the replanning time T and the noise bound N_θ is sufficiently small, then the difference between the replanned curves γ_i and the reference curve γ_0 is bounded (the tracking error has similar bounds). Obviously, the provided bounds grow as the replanning time T increases and decrease with the noise bounds N, N_θ. Exact tracking is guaranteed only when $N = 0$ and $N_\theta = 0$.

4.1.2 Recursive tracking in a general setting

In this subsection we introduce the recursive tracking problem in a more general setting and present a technical result (proposition 7) which will permit to find tracking bounds for the case of the unicycle vehicle discussed in section 4.1.1.

Consider system

$$\begin{cases} \dot{z}(t) &= f(z(t), u(t)) + \eta(t) \\ z(t_0) &= z_0\,, \end{cases} \tag{4.10}$$

where $z(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and η is a noise term that satisfies the following constrain

$$\|\eta(t)\| \leq N \quad \forall t \in \mathbb{R}\,, \tag{4.11}$$

with $N \in \mathbb{R}_+$. As in the case of the unicycle, when $N = 0$, the system above is called the *nominal system* (4.10). Define as *feasible trajectory* a reference function which can be exactly tracked by the nominal system (4.10):

Definition 9 *A continuous function $\gamma_0 : \mathbb{R} \to \mathbb{R}^n$ is a feasible trajectory for* (4.10) *with control u_0, if the following differential equation is satisfied*

$$\dot{\gamma}_0(t) = f(\gamma_0(t), u_0(t)), \quad t \geq 0\,. \tag{4.12}$$

The following is the fundamental assumption for defining a recursive iterative tracking. For every feasible system trajectory γ, every initial state $\bar{z}$ and time $\bar{t}$, it is assumed that there exists a feasible replanned trajectory that brings asymptotically the state from $\bar{z}$ to the reference γ.

Assumption 2 *If γ_0 is a feasible trajectory for* (4.10) *then* $\forall \bar{z} \in \mathbb{R}^n$ *and* $\bar{t} \in \mathbb{R}$ *there exist continuous functions* $u_{\bar{z},\bar{t},\gamma_0} : [\bar{t}, +\infty) \to \mathbb{R}^m$ *and* $\gamma_{\bar{z},\bar{t},\gamma_0} : [\bar{t}, +\infty) \to \mathbb{R}^n$*, such that*

$$\begin{cases} \dot{\gamma}_{\bar{z},\bar{t},\gamma_0}(t) &= f(\gamma_{\bar{z},\bar{t},\gamma_0}(t), u_{\bar{z},\bar{t},\gamma_0}(t)) \\ \gamma_{\bar{z},\bar{t},\gamma_0}(\bar{t}) &= \bar{z}\,, \end{cases} \tag{4.13}$$

and

$$\lim_{t\to+\infty} \gamma_{\bar{z},\bar{t},\gamma_0}(t) - \gamma_0(t) = 0\,. \tag{4.14}$$

Assumption 2 allows defining a recursive iterative control (as has been done in the case of the unicycle vehicle in section 4.1.1) in the following way.
Control Law: Given a reference trajectory γ_0, the control function $\bar{u}$ for system (4.1) is defined as follows

$$\begin{array}{l} \bar{u}(t) = u_0(t), \text{ if } t \in [0, T] \\ \bar{u}(t) = u_{z(iT),iT,\gamma_{i-1}}(t) \text{ if } t \in [iT, (i+1)T]\,, \end{array} \tag{4.15}$$

where

$$\begin{array}{rcl} \dot{z}(t) &=& f(z(t), \bar{u}(t)) \\ \gamma_i(t) &=& \gamma_{z(iT),iT,\gamma_{i-1}}(t), \ i > 0\,. \end{array} \tag{4.16}$$

The following defines a particular class of positive definite operators, similar to Lyapunov functions.

Definition 10 *Let n be a positive integer, then $U : \mathbb{R}^n \to \mathbb{R}$, is a seminorm if the following conditions hold*

1. $V(0) = 0\,;$

2. $V(z) \geq 0, \forall z \in \mathbb{R}^n\,;$

3. $V(z_1 + z_2) \leq V(z_1) + V(z_2),\ \forall z_1, z_2 \in \mathbb{R}^n\,.$

Moreover $V = (V_1, V_2, \ldots, V_l) : \mathbb{R}^n \to \mathbb{R}^l$ is a vector of seminorms if each component V_i is a seminorm.

Notation: for any relational operator $<_R$ and $x, y \in \mathbb{R}^n$, $x <_R y$ means $x_i <_R y_i$, $i = 1, \ldots, n$.

Definition 11 *Given a function* $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ *and a seminorm* U*, we say that system* (4.10) *is* (U, φ)*-bounded, if, when* $\bar{\gamma}$ *is a feasible trajectory with control* $\bar{u}$ *and* z *is the solution of the following system*

$$\begin{cases} \dot{z}(t) &= f(z(t), \bar{u}(t)) + \eta(t) \\ z(t_0) &= \bar{\gamma}(t_0)\,, \end{cases}$$

then, $\forall t \geq t_0$

$$U(z(t) - \bar{\gamma}(t)) \leq \varphi(t - t_0)\,. \tag{4.17}$$

The following proposition is the main result of this section.

Proposition 7 *Let* V *be a vector of seminorms and* U *a seminorm,* γ_0 *a feasible trajectory for* (4.10)*, with control function* u_0*. Let* $z(t)$ *and* γ_i *be defined according to* (4.15)*,* (4.16)*. Let function* $\Phi : \mathbb{R}^l \times \mathbb{R} \times \mathbb{R}^l \to \mathbb{R}^l$ *be such that*

$$V(\bar{\gamma}_{z_0,t_0,\gamma}(t) - \gamma(t)) \leq \Phi(U(z(t_0) - \gamma(t_0)), t - t_0, W(\gamma - \gamma_0))\,, \tag{4.18}$$

and Φ *is monotone increasing with respect to each component of the argument* W*, defined as* $W(\gamma) = \sup_{t \in \mathbb{R}} V(\gamma(t))$*. Moreover, assume that there exists a function* $\varphi(t)$*, such that* (4.10) *is* (U, φ)*-bounded. If there exists* $\bar{V} \in \mathbb{R}^l$ *such that*

$$\bar{V} \geq \sum_{k=1}^{+\infty} \Phi(\varphi(T), t - kT, \bar{V})\,, \tag{4.19}$$

then, $\forall t \in \mathbb{R}$ *and* $\forall i \in \mathbb{N}$*,*

$$V(\gamma_i(t) - \gamma_0(t)) \leq \bar{V}\,. \tag{4.20}$$

Proof. Proposition 7 can be proved by induction as follows. Consider first $i = 0$, in this case inequality (4.20) holds since, by 1) of definition 10,

$$V(\gamma_0(t) - \gamma_0(t)) = V(0) = 0 \leq \bar{V}\,.$$

Moreover assume that (4.20) is verified for $i = 0, 1, \ldots, l-1$, then from (4.18) and 2) of definition 10 the following relation is obtained

$$\begin{aligned} V(\gamma_l(t) - \gamma_0(t)) &= V\left(\sum_{k=1}^{l}(\gamma_k(t) - \gamma_{k-1}(t))\right) \leq \sum_{k=1}^{l} V(\gamma_k(t) - \gamma_{k-1}(t)) \\ &\leq \sum_{k=1}^{l} \Phi\left(U(z(kT) - \gamma_{k-1}(kT)), t - kT, W\left(\gamma_{k-1}(t) - \gamma_0(t)\right)\right) . \end{aligned} \tag{4.21}$$

From (4.17), with $\bar{\gamma} = \gamma_{k-1}$ and $t_0 = (k-1)T$, $\forall k = 1, \ldots, l$, the following inequality holds

$$U(z(kT) - \gamma(kT)) \leq \varphi(T) .$$

Since by the inductive hypothesis relation (4.20) is true for $i = 0, 1, \ldots, l-1$, $\forall t \in \mathbb{R}$:

$$W(\gamma_{k-1}(t) - \gamma_0(t)) \leq \bar{V} ,$$

therefore, the following inequality is obtained

$$V(\gamma_l(t) - \gamma_0(t)) \leq \sum_{k=1}^{l} \Phi(\varphi(T), t - kT, \bar{V}) . \tag{4.22}$$

and finally, combining (4.22) and (4.19), it follows that

$$V(\gamma_i(t) - \gamma_0(t)) \leq \bar{V} .$$

□

Remark that instead of finding separately a function Φ and φ which satisfy (4.18) and (4.17), one can find directly the composite function $\Phi(\varphi(T), t - t_0, W(\gamma - \gamma_0))$ which appears in (4.19), as will be done for the unicycle vehicle.

The idea behind proposition (7) is the following. The key element for finding bounds for trajectories γ_i defined in (4.16) consists in finding the function $\Phi(\varphi(T), t - t_0, W(\gamma - \gamma_0))$, which provides bounds on the norm at time t of the difference of a curve replanned at t_0 with the previous one (γ), as a function of the replanning time T, the time elapsed since the parameterization ($t - t_0$) and the maximum value of the norms of the difference between γ and the reference curve γ_0.

4.1.3 Application to the tracking problem for the unicycle

In this subsection proposition 7 is applied to the tracking problem for the unicycle vehicle, introduced in section 4.1.1.

The following lemma estimates the error on the feed-forward control of system (4.1) caused by the noise terms.

Lemma 1 *Consider system* (4.1)*, assume that a) and b) in property* 4 *hold and that the controls u and ω are given by c) and d). Then the following inequalities hold*

$$|\theta(t) - \arg(\gamma_0(t))| \leq N_\theta t\,, \tag{4.23}$$

$$\left\| \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} - \gamma_0(t) \right\| \leq \frac{t^2}{2} N_\theta V_M + Nt\,. \tag{4.24}$$

Proof. Define $e_\theta(t) = \theta(t) - \arg(\dot\gamma_0(t))$ and $e(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} - \gamma_0(t)$, then $\dot e_\theta(t) = \eta_\theta(t)$ and $|\dot e_\theta(t)| \leq N_\theta$, from which (4.23) is obtained. Moreover $\dot e = v \begin{pmatrix} \cos\theta(t) - \cos(\arg\dot\gamma(t)) \\ \sin\theta(t) - \sin(\arg\dot\gamma(t)) \end{pmatrix} + \eta$ and $\|\dot e(t)\| \leq V_M \sqrt{2}\sqrt{1-\cos e_\theta} + N$. Since $\cos x \geq 1 - \frac{x^2}{2}$, then $\|\dot e(t)\| \leq V_M \frac{t^2 N_\theta}{2} + Nt$, from which (4.24) follows. □

The following result represents the direct application of proposition 7 to the case of the unicycle.

Proposition 8 *Consider system* (4.1)*, where the control u is defined by* (4.5)*-*(4.6) *and the reference function γ_0 satisfies assumption* 1*. Moreover suppose that*

$$\chi = TN_\theta \left[\sum_{i=0}^{+\infty} \lambda(iT) + \sum_{i=0}^{+\infty} |\dot\lambda(iT)| iT \right] < 1\,.$$

Define

$$\begin{aligned} \bar V_2 = (1-\chi)^{-1} & \left(TN_\theta V_M TN_\theta \left(\sum_{i=1}^{+\infty} \lambda(Ti) + \dot\lambda(Ti) T_i \right) \right. \\ & \left. + \left(\frac{TN_\theta}{2} + NT \right) \sum_{i=0}^{+\infty} \lambda(T_i) \right)\,, \end{aligned} \tag{4.25}$$

$$\bar{V}_1 = \left(\frac{T^2}{2} N_\theta(\bar{V}_2 + V_M) + NT\right) \sum_{i=1}^{+\infty} \lambda(Ti) + TN_\theta(\bar{V}_2 + V_M) \sum_{i=1}^{+\infty} \lambda(Ti)\,, \quad (4.26)$$

$$\begin{aligned} \bar{V}_3 = \left(1 - TN_\theta \sum_{i=0}^{+\infty} \lambda(iT)\right)^{-1} & \left\{ T\,N_\theta\,(\bar{V}_2 + V_M) \left(\sum_{i=1}^{+\infty} \ddot{\lambda}(iT) + \dot{\lambda}(iT) \right) \right. \\ & \left. + A_M\,T\,N_\theta \sum_{i=1}^{+\infty} \lambda(iT) + \left(\frac{T^2}{2}\,N_\theta + T\,N\right) \sum_{i=1}^{+\infty} \ddot{\lambda}(iT) \right\}\,, \end{aligned} \quad (4.27)$$

and suppose that the following condition is verified

$$V_m < \bar{V}_2\,,$$

then the following inequalities hold, $\forall i \in \mathbb{N}$ *and* $\forall t \geq iT$,

$$\|\gamma_i(t) - \gamma_0(t)\| \leq \bar{V}_1\,, \quad (4.28)$$

$$\|\dot{\gamma}_i(t) - \dot{\gamma}_0(t)\| \leq \bar{V}_2\,, \quad (4.29)$$

$$\|\ddot{\gamma}_i(t) - \ddot{\gamma}_0(t)\| \leq \bar{V}_3\,. \quad (4.30)$$

Moreover the controls defined by (4.5) *and* (4.6) *satisfy the following bounds,* $\forall i \in \mathbb{N}$, $\forall t \geq iT$,

$$u(t) \in [V_m - \bar{V}_2, V_M + \bar{V}_2]\,, \quad (4.31)$$

$$\|\omega(t)\| \leq \frac{A_M + \bar{V}_3}{V_m - \bar{V}_2}\,. \quad (4.32)$$

Proof. Since, input functions $u(t)$, $\omega(t)$ defined in (4.5) and (4.6) are C^1 and respectively C^0, then the extended state $z = \{x, y, \theta, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}\}$ is well defined. Set $V = (V_1, V_2, V_3)$ with $V_1 = \|(x, y)\|$, $V_2 = \|(\dot{x}, \dot{y})\|$ and $V_3 = \|(\ddot{x}, \ddot{y})\|$. Remark that V satisfies definition 10. In order to use proposition 7, we now define the function $\Phi = (\Phi_1, \Phi_2, \Phi_3)$ such that (4.18) holds. To define Φ_1, consider the following bound

$$\begin{aligned} V_1(\gamma_{z(t_0),t_0,\gamma} - \gamma) = {} & \|\lambda(t - t_0)\,\{R(e_\theta(t_0))\,[\gamma(t) - \gamma(t_0)] \\ & + \gamma(t_0) + e_\gamma(t_0)\} + [1 - \lambda(t - t_0)]\,\gamma(t) - \gamma(t)\| \\ = {} & \|\lambda(t - t_0)\,\{[R(e_\theta(t_0)) - I]\,[\gamma(t) - \gamma(t_0)] + e_\gamma(t_0)\}\|\,, \end{aligned}$$

together with $\|R(_\theta(t_0)) - I\| \leq |e_\theta(t_0)|$ and

$$\begin{aligned}\|\gamma(t) - \gamma(t_0)\| \leq (t - t_0) \sup_{t \geq t_0} \|\dot{\gamma}(t)\| &\leq (t - t_0) \left[V_M + \sup_{t \geq t_0} \|\dot{\gamma}(t) - \dot{\gamma}_0(t)\| \right] \\ &\leq (t - t_0) \left[V_M + W_2(\gamma - \gamma_0) \right].\end{aligned}$$

Therefore, by lemma 1, we find the bound

$$\begin{aligned}V_1(\gamma_{z(t_0),t_0,\gamma} - \gamma) &\leq \lambda(t - t_0) \, \{|e_\theta(t_0)| \, (t - t_0)[W_2(\gamma - \gamma_0) + V_M] + |e_\gamma(t_0)|\} \\ &\leq \Phi_1(T, t - t_0, W(\gamma - \gamma_0)).\end{aligned}$$

Analogously

$$\begin{aligned}V_2(\gamma_{z(t_0),t_0,\gamma} - \gamma) = \|\dot{\lambda}(t - t_0) \, \{[R(e_\theta(t_0)) - I][\gamma(t) - \gamma(t_0)] + e_\gamma(t_0)\} + \dot{\gamma}(t) \\ \{\lambda(t - t_0) \, [R(e_\theta(t_0)) - I]\}\| \leq \Phi_2(T, t - t_0, W(\gamma - \gamma_0)).\end{aligned}$$

Finally

$$\begin{aligned}V_3(\gamma_{z(t_0),t_0,\gamma} - \gamma) = \|\ddot{\lambda}(t - t_0) \, \{[R(e_\theta(t_0)) - I][\gamma(t) - \gamma(t_0) + e_\gamma(t_0)]\} \\ + \dot{\lambda}(t - t_0) \, [R(e_\theta(t_0)) - I]\dot{\gamma}(t) + \ddot{\gamma}(t) \, \{1 + \lambda(t - t_0) \, [R(e_\theta(t_0) - I]\} \\ + \dot{\gamma}(t) \, \ddot{\lambda}(t - t_0) \, [R(e_\theta(t_0) - I] - \ddot{\gamma}(t)\| \leq \Phi_3(T, t - t_0, W(\gamma - \gamma_0)).\end{aligned}$$

From (4.25), (4.26) and (4.27) it follows that, for $k = 1, 2, 3$

$$\bar{V}_k \geq \sum_{i=1}^{+\infty} \Phi_k(T, iT, \bar{V}_k),$$

and, by (4.20) of proposition 7, relations (4.28), (4.29) and (4.30) hold. Moreover, $\forall t \in [iT, (i + 1)T]$

$$\begin{aligned}u(t) = \|\dot{\gamma}_i(t)\| &= \|\dot{\gamma}_0(t) + \dot{\gamma}_i(t) - \dot{\gamma}_0(t)\| \\ \in &\left[V_m - \sup_{t \geq iT} \{\|\dot{\gamma}_i(t) - \dot{\gamma}_0(t)\|\}, V_M + \sup_{t \geq iT} \{\|\dot{\gamma}_i(t) - \dot{\gamma}_0(t)\|\} \right] \\ &\subset \left[V_m - \bar{V}_2, V_M + \bar{V}_2 \right],\end{aligned}$$

hence (4.31) holds. Furthermore,

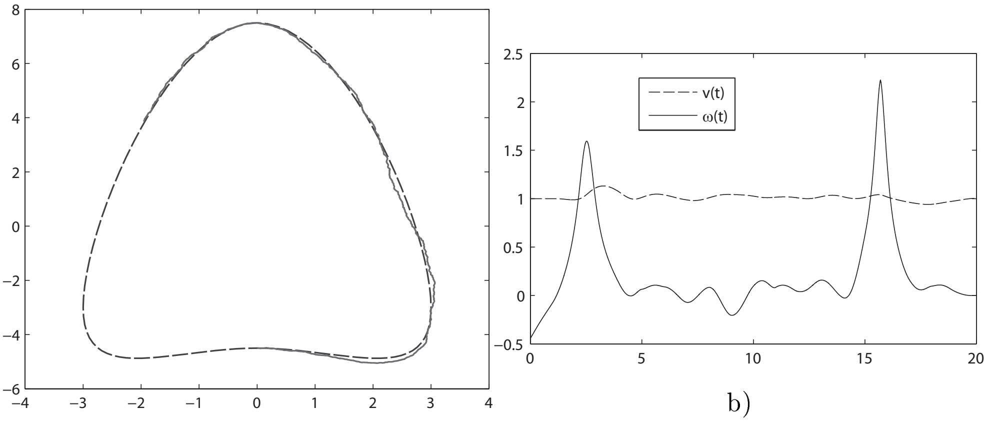
$$\begin{aligned}|\omega(t)| = |\frac{d}{dt}\arg(\dot{\gamma}_i(t))| &\leq \frac{|\det[\ddot{\gamma}_i(t), \dot{\gamma}_i(t)]|}{\|\dot{\gamma}_i(t)\|^2} \leq \frac{\|\ddot{\gamma}_i(t)\|}{\|\dot{\gamma}_i(t)\|} \\ &\leq \frac{\|\ddot{\gamma}_0(t)\| + \sup_{t\geq iT}\{\|\ddot{\gamma}_i(t) - \ddot{\gamma}_0(t)\|\}}{\|\dot{\gamma}_0(t)\| - \sup_{t\geq iT}\{\|\dot{\gamma}_i(t) - \dot{\gamma}_0(t)\|\}} \leq \frac{A_M + \bar{V}_2}{V_m - \bar{V}_2},\end{aligned}$$

therefore (4.32) holds and the proof of proposition 8 is complete. □

Corollary 1 follows from proposition 8 when λ is given by (4.4).

4.1.4 Simulation results

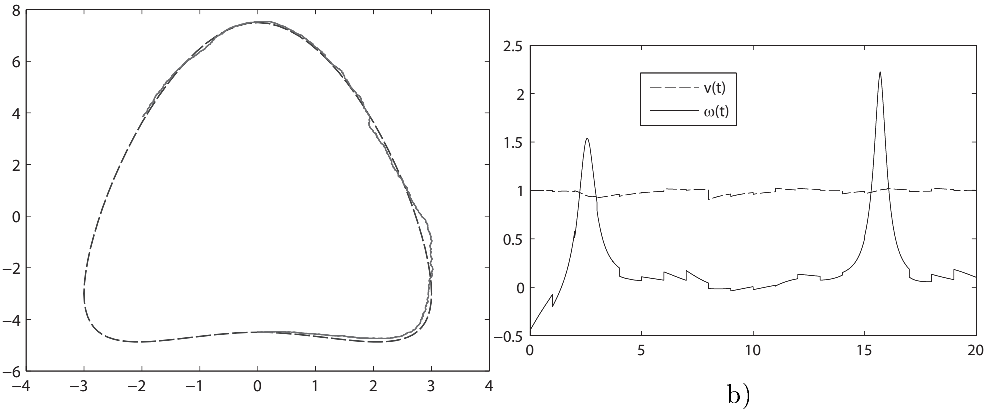
The method presented in subsection 4.1.1 has been compared with the controller for the unicycle presented in [68, p.809]. We have assumed that the state is measured only at regular intervals $T = 1$ s, which represents also the replanning time for our algorithm. The state appearing in the feedback control law presented in [68] is obtained through a discontinuous open loop observer which is updated at each observation time. The gain in this controller have been set to have controls signals of magnitude similar to the method of section 4.1.1. As reference trajectory we have considered a periodic spline followed with constant speed 1 m/s. The noise bounds appearing in (4.2) are given by $N = 0.5\sqrt{2}$ and $N_\theta = 0.5$. The obtained results are presented in figures 4.6 and 4.7. The two methods showed a similar performance in terms of tracking error. However, the control method presented in this section has the advantage of providing overall continuous input signals whereas the control signals of the classic controller are discontinuous (even if this is a consequence of having used a discontinuous observer). Our method has the advantage of guaranteeing an arbitrary class of continuity in the input signals. Moreover, it is not an *ad hoc* solution for the unicycle, since it can be applied in principle to any system satisfying the conditions presented in section 4.1.2.



a)

b)

Figure 4.6: a) The robot trajectory and b) the control inputs for the recursive method.



a)

b)

Figure 4.7: a) The robot trajectory and b) the control inputs for the method presented by Samson.

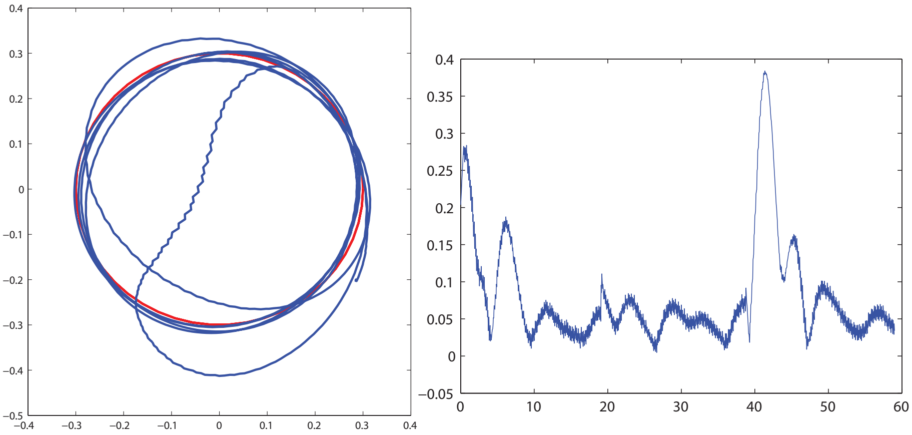
4.1.5 Experimental results

We have implemented an experimental setting for the method presented in section 4.1.1. A mobile robot built with Lego Mindstorm NXT pieceshas been used. The traction is provided by two front wheels, a passive rear castor wheel is used to prevent the robot from falling over. The inputs variable are ω_l and ω_r, the angular velocity of left and right wheels. Set $v = r\frac{\omega_l+\omega_r}{2}$ and $\omega = \frac{r}{L}\frac{\omega_l-\omega_r}{2}$, where r is the driving wheels radius and L is the distance between the two wheels. After this substitution this differential drive robots can be described with the unicycle model (4.1).

Two red markers of different sizes have been placed on the robot and the system state (x, y, θ) is measured ten times per second through a Unibrain firewire camera, using standard computer vision techniques. A personal computer running MATLAB contains a systems observer for finding the robot state and implements the recursive controller presented in (4.5), (4.6) and (4.7). The control signals are computed and sent to the wheeled robot via Bluetooth. The replanning time has been set to $T = 0.8$ s. This experimental setting is characterized by some difficulties, in particular the Bluetooth transmission introduces in the control loop a delay of 80 ms, and the wheels occasionally experiment slippage.

Figure 4.8-a) shows the experimental results obtained when the reference trajectory is a circle of radius equal to 30 cm, followed with a constant speed of 0.2 m/s. The red line represents the reference trajectory γ_0 and the blue line the robot observed position. In the middle of the test the robot has been moved with a rod to test the robustness of the controller, this explain the large transient error present in the figure. In figure 4.8-b), the norm of the (x, y) component of the tracking error is showed; the spike on time $t = 40$ it is due to the test of the robustness of the controller.

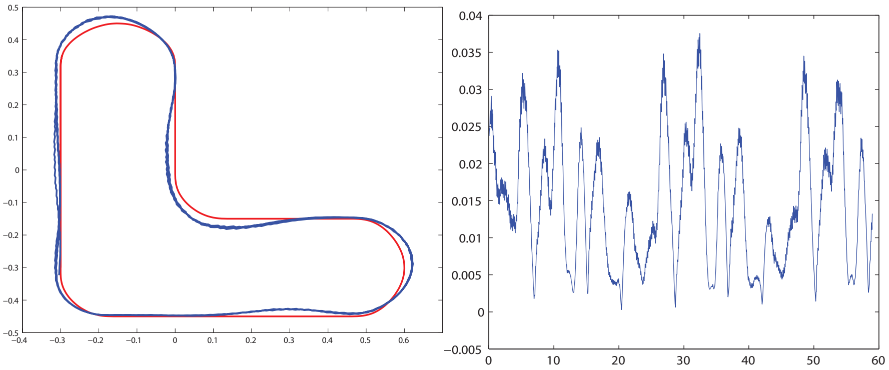
Figure 4.9-a) shows another experiment where the desired trajectory is a spline which has been reparameterized with constant speed 0.15 m/s. The associated tracking error is shown in figure 4.9-b). Remark that the evaluation of functions γ_i in (4.7) require the use of a recursive function. If function λ



b)

a)

Figure 4.8: a) Reference and actual trajectory for a circle b) the norm of the (x, y) component of the tracking error with respect to time.



a)

b)

Figure 4.9: a) Reference and actual trajectory for a composite spline b) the norm of the (x, y) component of the tracking error with respect to time.

reaches 0 in finite time τ, then the maximum order of recursion is given by the ratio $\frac{\tau}{T}$ (recall that T is the replanning time). Since the order of recursion is deterministic, the proposed control law can be implemented in a real time controller. Parameter T must be carefully chosen. In fact, on one hand, by (4.8), (4.9), reducing T improves the tracking performances. On the other hand, it increases the ratio $\frac{\tau}{T}$, the number of recursions needed to implement the controller and the computational effort.

4.2 Iterative output replanning for flat systems

The section considers the output tracking problem for nonlinear systems whose performance output is also a flat output of the system itself. A desired output signal is sought on the actual performance output by using a feedforward inverse input that is periodically updated with discrete-time feedback of the sampled state of the system. The proposed method is based on an iterative output replanning that uses the desired output trajectory and the sampled state to replan an output trajectory whose inverse input helps in reducing the tracking error. This iterative replanning exploits the Hermite interpolating polynomials to achieve an overall arbitrarily smooth input and a tracking error that can be made arbitrarily small if the state sampling period is sufficiently small and mild assumptions are considered. Some simulation results are presented for the cases of an unicycle and a one-trailer system affected by additive noise.

4.2.1 Problem statement

Consider the nonlinear controlled system

$$\dot{x} = f(x, u) , \tag{4.33}$$

with $x \in C(\mathbb{R}, \mathbb{R}^n)$, $u \in C(\mathbb{R}, \mathbb{R}^m)$. System (4.33) is *flat* if there exists an output function y such that the system state $x(t)$ and the input $u(t)$ can be written as a function of y and its derivatives up to a finite order, evaluated at time t. More precisely the following definition can be given (see [52]).

Definition 12 *System* (4.33) *is flat if there exist a flat output y of dimension m, two integers r and s and mappings ψ from $\mathbb{R}^n \times \mathbb{R}^{m(s+1)}$ to $\mathbb{R}^m$, of rank m in a suitably chosen output subset, and (ϕ_0, ϕ_1) from $\mathbb{R}^{m(r+2)}$ to $\mathbb{R}^n \times \mathbb{R}^m$, of rank $m+n$ in a suitable open subset, such that*

$$y = (y_1, \ldots, y_m) = \psi(x, u, \dot{u}, \ldots, u^{(s)})\,, \tag{4.34}$$

implies that

$$\begin{aligned} x &= \phi_0(y, \dot{y}, \ldots, y^{(r)})\,, \\ u &= \phi_1(y, \dot{y}, \ldots, y^{(r+1)})\,, \end{aligned} \tag{4.35}$$

the differential equation $\frac{d\phi_0}{dt} = f(\phi_0, \phi_1)$ being identically satisfied.

In this way, function ϕ_0 represents the state x with the output y and its derivatives up to the order r. Function ϕ_1 represents the input u with the output and its derivatives up to the order $r+1$.

For simplicity, for a C^n function f we use the notation $\bar{f}^n = (f, f^{(1)}, \ldots, f^{(n)})$, to denote the ordered set containing function f and its time derivatives up to the order n.

If ϕ_1 is sufficiently regular, differentiating (4.35), one obtains functions ϕ_i, such that, for any $i \geq 1$

$$u^{(i-1)} = \phi_i(\bar{y}^{r+i})\,, \tag{4.36}$$

i.e., the input derivatives can be expressed as a function of the output and its derivatives. Similarly, if ψ is sufficiently regular, differentiating (4.34), one obtains functions ψ_i, such that, for any $i \geq 0$

$$y^{(i)} = \psi_i(x, \bar{u}^{s+i})\,, \tag{4.37}$$

with $\psi_0 = \psi$, where ψ is given in (4.34). Combining (4.36) and (4.37), the following identity holds $\forall i \geq 1$

$$u^{(i-1)} = \phi_i(\psi_0(x, \bar{u}^s), \ldots, \psi_{r+i}(x, \bar{u}^{(s+r+i)}))\,. \tag{4.38}$$

It is well known that tracking and motion planning problems can be easily solved for flat systems, see for instance chapter 7 of [52]. In this section we study the tracking problem for system (4.33), in presence of a bounded disturbance added to the nominal velocity of the state:

$$\dot{x}(t) = f(x(t), u(t)) + \eta(t)\,, \tag{4.39}$$

where η is a disturbance signal such that

$$\|\eta(t)\| \leq N,\ \forall t \in \mathbb{R}\,. \tag{4.40}$$

The performance output of system (4.39) is given by

$$y = \psi(x, u)\,. \tag{4.41}$$

We assume that y is a flat output for system (4.39) when no noise is present (i.e., $\eta = 0$). In this case, from (4.34) and (4.35), it follows that the output signal y satisfies

$$y = \psi(\phi_0(\bar{y}^r), \phi_1(\bar{y}^{r+1}))\,. \tag{4.42}$$

Note that the form (4.39) may be restrictive since the disturbance η enters as a pure additive term. This form does not include, for instance, cases in which a disturbance multiplies the state x or the input u.

We assume that the full system state is acquired periodically, with a sampling period equal to $T > 0$. In this way, the feedback control relies on the discrete-time observed sequence $x(kT)$, $k \in \mathbb{N}$. For instance, this assumption is reasonable when the system state is obtained through a camera, using computer vision techniques. In this case, a sampling time of $T = 0.1$ seconds would be a typical situation.

We study an iterative output replanning technique for controlling system (4.39), based on Hermite interpolating polynomials, similar in spirit to the iterative state steering method presented in [4]. Roughly speaking, the method is the following. A sufficiently regular reference output trajectory y_d is assigned in advance. During each time interval $[kT, (k+1)T[$, a replanned output y_p is computed such that

1. y_p corresponds through (4.35) to an initial state which is the same as $x(kT)$, i.e.

$$x(kT) = \phi_0(y_p(kT), \dot{y}_p(kT), \ldots, y_p^{(r)}(kT)) .$$

2. the replanned output y_p converges to the desired one y_d at time $(k+1)T$, i.e., $y_p((k+1)T) = y_d((k+1)T)$.

The control is given according to (4.35), $\forall t \in [kT, (k+1)T[$, by

$$u(t) = \phi_1(y_p(t), \dot{y}_p(t), \ldots, y_p^{(r+1)}(t)) .$$

Since the system is affected by additive noise and in interval $[kT, (k+1)T[$ open loop control is used, at time $(k+1)T$ the system output $y((k+1)T)$ is different from $y_d((k+1)T)$. Hence, the above step is repeated, finding a new replanned trajectory y_p, that would drive the output of the nominal system to y_d at time $(k+2)T$. Again, for the presence of noise, at time $(k+2)T$ the actual system output is different from the reference trajectory and a new trajectory is replanned. Since the replanned trajectories converge to the reference y_d, the system output is driven towards the desired output and the tracking error is kept limited despite the presence of a disturbance. This method is illustrated in figure 4.10, while figure 4.11 shows the corresponding control scheme.

We prove that the tracking error can be made arbitrarily small if the replanning time T is chosen sufficiently small. Moreover, we show that the replanned output y_p can be chosen in such a way to have an arbitrary degree of continuity on the resulting input function.

4.2.2 An Hermite interpolation problem

Consider the following problem.

Problem 7 (Replanning problem) *Given flat system* (4.33)*, an output reference trajectory, $y_d \in C^{r+l}(\mathbb{R}, \mathbb{R}^m)$, an initial state $x_0 \in \mathbb{R}^n$ and initial values for the input and its derivatives $u_0, u_0^{(1)}, \ldots, u_0^{(l-1)}$, find an output reference trajectory $y_p \in C^{r+l}(\mathbb{R}, \mathbb{R}^m)$ such that the following properties hold*

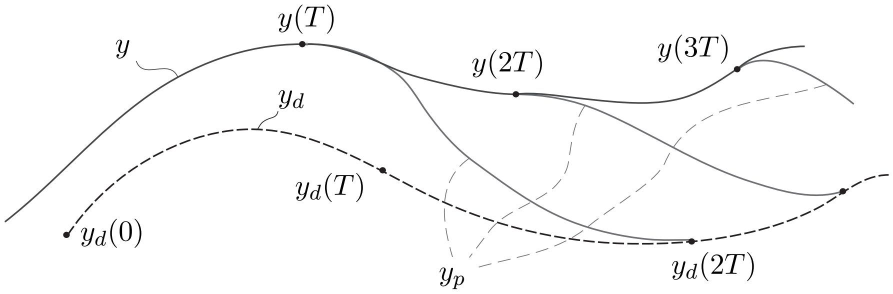


Figure 4.10: The iterative replanning method. The figure shows the reference output trajectory y_d, the actual system output y and the replanned trajectory y_p.

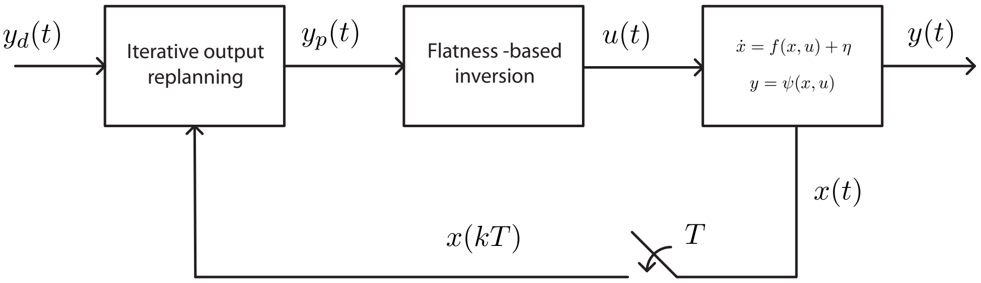

$y_d(t)$
Iterative output replanning
$y_p(t)$
Flatness -based inversion
$u(t)$
$\dot{x} = f(x, u) + \eta$
$y = \psi(x, u)$
$y(t)$
$x(t)$
$x(kT)$
T

Figure 4.11: The iterative control scheme for the trajectory tracking of a flat system.

a) $\phi_0(\bar{y}_p^r(0)) = x_0$, *i.e., x_0 is the initial state of the system trajectory that has y_p as output,*

b) $u_0^{(i)} = \phi_{i+1}(\bar{y}_p^{r+i+1}(0))$, *$i = 0, \ldots, l-1$ i.e., $u_0^{(i)}$ is the initial value of the i-th derivative of the control for the system trajectory which has y_p as output,*

c) $y_p(t) = y_d(t)$, *$\forall t \geq T$, where T is a given positive constant, i.e. function y_p converges to y_d at time T.*

For any $l \in \mathbb{N}$, let $\Psi_0, \Psi_1, \ldots, \Psi_{r+l}$, be vectors in $\mathbb{R}^m$ and set matrix $\Psi = (\Psi_0, \Psi_1, \ldots, \Psi_{r+l})$. Consider the interpolation problem of determining a function $\pi_{\Psi,T} \in C^{r+l}([0,T], \mathbb{R}^m)$ that satisfies the two conditions

$$\frac{d^i}{dt^i}\pi_{\Psi,T}(0) = \Psi_i,\ i = 0, \ldots, r+l\,, \tag{4.43}$$

$$\frac{d^i}{dt^i}\pi_{\Psi,T}(T) = 0,\ i = 0, \ldots, r+l\,. \tag{4.44}$$

Condition (4.43) requires that function $\pi_{\Psi,T}$ have the first $r+l$ derivatives equal to the columns of Ψ at time $t = 0$, while condition (4.44) requires that all derivatives up to the $(r+l)$-th be equal to 0 at time $t = T$.

This problem belongs to the class of Hermite interpolation problems, which have been widely studied in interpolation literature. Its solution can be written in the form

$$(\pi_{\Psi,T})_i(t) = \sum_{k=0}^{r+l} A_{T,k}(t)\,(\Psi_k)_i\,, \tag{4.45}$$

where the Hermite interpolation function $A_{T,k}$ is the minimum degree polynomial that satisfies conditions

$$\frac{d^i}{dt^i}A_{T,k}(0) = \delta_{i-k}, \quad \frac{d^i}{dt^i}A_{T,k}(T) = 0\,,$$

where

$$\delta_i = \begin{cases} 1 \text{ if } i = 0\,, \\ 0 \text{ otherwise.} \end{cases}$$

These polynomials have degree $2(r+l+1)$ and can be computed in closed form using a result presented in [69]:

$$A_{T,k}(t) = (t-T)^{r+l+1}\,\frac{t^k}{k!}\sum_{i=0}^{r-k+l}\frac{(-t)^i}{i!}\,\frac{(r+i+l)!}{(-T)^{r+l+i+1}\,(r+l)!}\,. \tag{4.46}$$

These polynomials satisfy the following inequality, $\forall t \in [0,T]$

$$|A_{T,k}(t)| \leq \frac{T^k}{k!}\sum_{i=0}^{r-k+l}\frac{(r+i+l)!}{i!(r+l)!}\,.$$

Expression (4.45) implies that, for any $\tilde{T} > 0$, there exists a constant C, such that, $\forall T \in [0,\tilde{T}]$, $\forall t \in [0,T]$

$$|\pi_{\Psi,T}(t)| \leq C\|\Psi\|\,. \tag{4.47}$$

Figure 4.12 shows some of the Hermite polynomials $A_{T,k}$. We use Hermite polynomials for defining the replanned trajectory y_p. To this end, define function

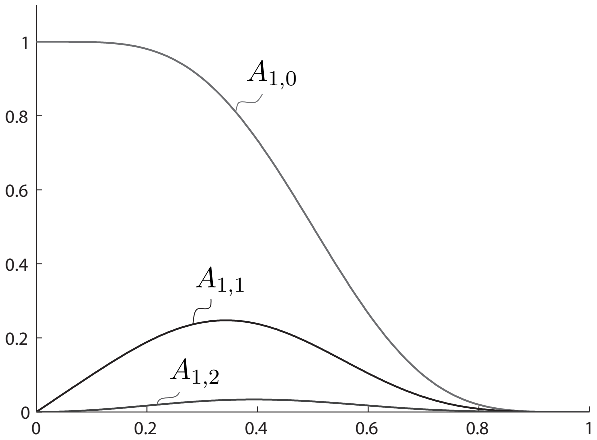


Figure 4.12: The first three Hermite polynomials for $r+l=4$ and $T=1$.

$$\begin{aligned}\bar{\psi} : \mathbb{R}^n \times \mathbb{R}^{m(r+l+1)} &\to \mathbb{R}^{m\times(r+l+1)}\\ (z, v, v_1, \ldots, v_{r+l}) &\rightsquigarrow (\Phi_0, \Phi_1, \ldots, \Phi_{r+l})\,,\end{aligned} \tag{4.48}$$

such that

$$\Phi_k = \psi_k(z, v_0, v_1, \ldots, v_k)\ , k = 0, \ldots, r+l\,, \tag{4.49}$$

where ψ_k is defined in (4.37). In this way Φ_k represents the k-th derivative of the output obtained when the system state is z and the input and its derivatives are given by v_i, $i = 0, \ldots, k$. Finally define the reference trajectory y_p as follows

$$y_p(t) = \begin{cases} y_d(t) + \pi_{\Psi,T}(t) & \text{if } 0 \leq t < T \\ y_d(t) & \text{if } t \geq T\,, \end{cases}$$

where

$$\Psi = \bar{\psi}(x_0, u_0^{(0)}, \ldots, u_0^{(r+l)}) - \bar{y}_d^{r+l}(0)\,. \tag{4.50}$$

Function y_p solves the *replanning problem*, since

1. it belongs to C^{r+l},
2. it satisfies properties a) and b) because of conditions (4.43) and (4.50),
3. it satisfies property c) because of condition (4.44).

Remark In this way, x_0 represents the initial state corresponding to output function y_p and $u_0, u_0^{(1)}, \ldots, u_0^{(r+l)}$ represent the initial input and the initial input derivatives up to the degree $r+l$.
The use of Hermite interpolation allows to define replanned trajectories that corresponds to arbitrary conditions on the initial state, the initial input and its derivatives.

4.2.3 Iterative control law

Using the replanning method described in the previous sections, the control law for system (4.39) is defined as follows

$$u(t) = \phi_1(\bar{y}_p^{r+1}(t))\,, \tag{4.51}$$

where ϕ_1 is given in (4.35) and, for $t \in]kT, (k+1)T]$

$$y_p(t) = y_d(t) + \pi_{\Psi(k),T}(t - kT)\,, \tag{4.52}$$

with

$$\Psi(0) = \bar{\psi}(x_0, u_0, u_0^{(1)}, \ldots, u_0^{(r+l)}) - \bar{y}_d^{r+l}(0)\,, \tag{4.53}$$

and, for $k > 0$,

$$\Psi(k) = \bar{\psi}(x(kT^-), \bar{u}^{r+l}(kT^-)) - \bar{y}_d^{r+l}(kT)\,. \tag{4.54}$$

In (4.53), x_0 represents the initial state $x(0)$ and the assigned constants u_0, $u_0^{(1)}, \ldots, u_0^{(r+l)}$ are the initial control input with its derivatives. In other words, in time interval $]kT,(k+1)T]$ it is used the control function u that would drive the nominal system (4.33) along the reference trajectory $y_p(t)$. This trajectory is computed by adding the polynomial function $\pi_{\Psi(k),T}$ to the reference trajectory y_d. In this way the replanned trajectory y_p satisfies the properties

a) $\phi_0(\bar{y}_p^r(kT)) = x(kT)$, i.e. $x(kT)$ is the value at time kT of the state trajectory that corresponds to y_p,

b) $\bar{y}_p^{r+l}((k+1)T^-) = \bar{y}_d^{r+l}((k+1)T)$, i.e. the replanned trajectory is the same as the desired trajectory at time $(k+1)T$.

4.2.4 Main results

A relevant property is that the resulting control function u is C^{l-1} continuous as shown in the following proposition.

Proposition 9 *The control function u defined in* (4.51) *belongs to class C^{l-1}.*

Proof. Since y_p is of class C^{r+l} in the open sets $]kT,(k+1)T[$ according to (4.36), the control function belongs to C^{l-1} in the the union of intervals $]kT,(k+1)T[$, $k \in \mathbb{N}$. It remains to prove C^{l-1} continuity on kT, $k \in \mathbb{N}$. Since system (4.33) is flat, by definition (4.51) and taking into account (4.36) it follows that, $\forall k \in \mathbb{N}$,

$$u^{(i)}(kT^+) = \phi_{i+1}(\bar{y_p}^{r+i+1}(kT^+)), \quad i = 0, \ldots, l-1\,,$$

moreover, by conditions (4.37), (4.54)

$$y_p^{(i)}(kT^+) = \psi_i(x(kT^-), \bar{u}^{s+i}(kT^-)),\; i = 0, \ldots, l\,,$$

therefore by (4.38), $u^{(i)}(kT^+) = u^{(i)}(kT^-)$, $\forall i = 0, \ldots, l-1$, $\forall k \in \mathbb{N}$, which proves C^{l-1} continuity. □

Remark With regards to proposition 9, it is worth noting that integer l is, in practice, a free parameter provided that a sufficiently smooth desired trajectory $y_d \in \mathcal{C}^{r+l}$ is designed. Consequently, this implies that the control input of the proposed method can be chosen as smooth as necessary or desired.

The main result of this paper requires the following Lipschitz assumption on function (4.33).

Assumption 3 *Given flat system* (4.33)*, there exist constants* $0 < L_f, L_\psi \in \mathbb{R}$ *for which* $\forall x_1, x_2 \in \mathbb{R}^n, u \in \mathbb{R}^m$

$$\|f(x_1, u) - f(x_2, u)\| \leq L_f \|x_1 - x_2\|\,,$$

and the associated function $\bar{\psi}$ *(see* (4.48)*) satisfies the following condition,* $\forall x_1, x_2 \in \mathbb{R}^n$ *and* $u_0, \ldots, u_{r+l} \in \mathbb{R}^m$

$$\|\bar{\psi}(x_1, u_0, \ldots, u_{r+l}) - \bar{\psi}(x_2, u_0, \ldots, u_{r+l})\| \leq L_\psi \|x_1 - x_2\|\,.$$

The following theorem states that it is always possible to choose a replanning time T, sufficiently small, such that the output tracking error is lower than any given positive constant ϵ.

Theorem 4 *Let* $\dot{x}(t) = f(x(t), u(t))$ *be a control system with flat output* (4.41)*, such that assumption 3 is satisfied. Let* $y_d \in C^{r+l}(\mathbb{R}, \mathbb{R}^m)$ *be a reference trajectory for the flat output* ψ*. Consider the differential system*

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) + \eta(t) \\ x(0) = x_0\,, \end{cases} \tag{4.55}$$

where $\|\eta(t)\| \leq N$*,* $\forall t \in \mathbb{R}$*, and the initial state* x_0 *is such that there exists an initial control* u_0 *for which* $y_d(0) = \psi(x_0, u_0)$*. Then, for any* $\epsilon > 0$*, there exists* $T > 0$ *such that the solution of* (4.55) *with control function* u *given by* (4.51) *satisfies*

$$\|\psi(x(t), u(t)) - y_d(t)\| \leq \epsilon,\ \forall t \geq 0\,.$$

The following lemma will be used in the proof of theorem 4.

Lemma 2 *Let $L_f \in \mathbb{R}$ be such that $\forall x_1, x_2 \in \mathbb{R}^n$ and $\forall t \in \mathbb{R}$*

$$\|f(x_1,t) - f(x_2,t)\| \leq L_f \, \|x_1 - x_2\| \, , \tag{4.56}$$

and let x and x_r be the solutions of

$$\begin{cases} \dot{x}(t) = f(x,t) + \eta(t) \\ x(0) = x_0 \, , \end{cases} \quad \textit{and} \quad \begin{cases} \dot{x}_r(t) = f(x_r,t) \\ x_r(0) = x_0 \, , \end{cases}$$

with $x_0 \in \mathbb{R}^n$ and $\|\eta(t)\| \leq N$, $\forall t \in \mathbb{R}$. Then

$$\|x(t) - x_r(t)\| \leq \frac{e^{L_f t} - 1}{L_f} N \, . \tag{4.57}$$

Proof. By hypothesis (4.56) the following differential inequality is satisfied

$$\frac{d\|x(t) - x_r(t)\|}{dt} \leq \|f(x(t),t) + \eta(t) - f(x_r(t),t)\| \leq L_f \|x(t) - x_r(t)\| + N \, .$$

Inequality (4.57) follows from the Comparison Lemma, solving the corresponding linear differential equation in the variable $\|x - x_r\|$. □

Proof of theorem 4. For any $k \in \mathbb{N}$, $\forall t \in [kT, (k+1)T[$, let x_r be the solution of the following differential system

$$\begin{cases} \dot{x}_r(t) = f(x_r(t), u(t)) \\ x_r(kT) = x(kT) \, , \end{cases} \tag{4.58}$$

where the control u is given by (4.51). System $f(x,u)$ is flat and by definition 12 the associated function ϕ_0 satisfies the differential equation

$$\frac{d\phi_0(\bar{y}_p^r)}{dt}(t) = f(\phi_0(\bar{y}_p^r(t)), \phi_1(\bar{y}_p^{r+1}(t))) \, . \tag{4.59}$$

By construction, the replanned trajectory y_p satisfies $x(kT) = \phi_0(\bar{y}_p^r(kT))$. Since u is defined by (4.51), differential equation (4.58) can be rewritten as

$$\begin{cases} \dot{x}_r(t) = f(x_r(t), \phi_1(\bar{y}_p^{r+1}(t))) \\ x_r(kT) = x(kT) \, . \end{cases}$$

Hence, for $t \in [kT, (k+1)T)[$, $x_r(t)$ is the solution of the same differential equation (4.59) as ϕ_0, therefore $x_r(t) = \phi_0(\bar{y}_p^r(t))$. Consequently, by (4.42), $\psi(x_r(t), u(t)) = \psi(\phi_0(\bar{y}_p^r), \phi_1(\bar{y}_p^{r+1})) = y_p(t)$.

Moreover, using lemma 2,

$$\|y(t) - y_p(t)\| = \|\psi(x, u) - \psi(x_r, u)\| \leq L_\psi \|x - x_r\| \leq L_\psi \frac{e^{L_f T} - 1}{L_f} N\,.$$

Set T_1 sufficiently small such that $L_\psi \frac{e^{L_f T_1} - 1}{L_f} N \leq \frac{\epsilon}{2}$, so that $\|y(t) - y_p(t)\| \leq \frac{\epsilon}{2}$, $\forall t > 0$.

Remark that, $\forall k \in \mathbb{N}$,

$$\begin{aligned} &\|\bar{\psi}(x(kT), \bar{u}^{r+l}(kT)) - \bar{\psi}(x_r(kT), \bar{u}^{r+l}(kT))\| \\ &\leq L_\psi \|x(kT) - x_r(kT)\| \leq L_\psi \frac{e^{L_f T} - 1}{L_f} N\,, \end{aligned}$$

therefore, by (4.47) and assumption 3, $\forall t \in [kT, (k+1)T[$

$$\begin{aligned} \|y_p(t) - y_d(t)\| &= \|\pi_{\bar{\psi}(x(kT), u(kT), \ldots, u^{(r+l)}(kT)) - \bar{y}_d^{r+l}(kT), T}(t)\| \\ &\leq C\|\bar{\psi}(x(kT), u(kT), \ldots, u^{(r+l)}(kT)) - \bar{y}_d^{r+l}(kT)\| \\ &= CL_\psi \|x(kT) - x_r(kT)\| \leq CL_\psi \frac{e^{L_f T} - 1}{L_f} N\,. \end{aligned}$$

Choose then T_2 such that $CL_\psi \frac{e^{L_f T_2} - 1}{L_f} N \leq \frac{\epsilon}{2}$, so that $\|y_p(t) - y_d(t)\| \leq \frac{\epsilon}{2}$, $\forall t > 0$. Finally set $T = \min\{T_1, T_2\}$, then the thesis holds since, $\forall t > 0$, $\|y(t) - y_d(t)\| \leq \|y(t) - y_p(t)\| + \|y_p(t) - y_d(t)\| \leq \epsilon$. □

4.2.5 Simulation results for the case of a unicycle

This section shows some simulation results obtained by applying the method presented in the above subsections to a unicycle system illustrated in figure 4.1. Here, the unicycle kinematics (4.1) of section 4.1 is proposed again with the only substitution of y with z.

$$\left\{ \begin{array}{rcl} \begin{pmatrix} \dot{x} \\ \dot{z} \end{pmatrix} & = & v(t) \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} + \eta(t) \\ \dot{\theta} & = & \omega(t) + \eta_\theta(t)\,, \end{array} \right. \tag{4.60}$$

In this example we want that the unicycle follows a C^2 curve with continuous control inputs $v(t)$, $\omega(t) \in C^0$. We have assumed that the state is periodically measured with sample time $T = 1$ s. The performance output is given by $y = (x, z)$, which is a flat output with $r = 1$ for the nominal unicycle (4.60). In order to obtain a C^0 control input, by proposition 9 we set $l = 1$. In this way the Hermite interpolating polynomials (4.46) have degree $2(r + l + 1) = 6$.

As reference trajectory we have considered a periodic spline y_d followed with constant speed 1 m/s. The noise bounds appearing in (4.2) are given by $N = 0.5\sqrt{2}$ and $N_\theta = 0.5$. The obtained result is presented in figure 4.13. The control inputs and the error functions are depicted in figure 4.14 and 4.15, where the position error $e(t)$ and the angular error $e_\theta(t)$ are defined as follows

$$e(t) = \left(\begin{array}{c} x(t) \\ z(t) \end{array} \right) - y_d(t)\,,$$

$$e_\theta(t) = \theta(t) - \arg \dot{y}_d(t).$$

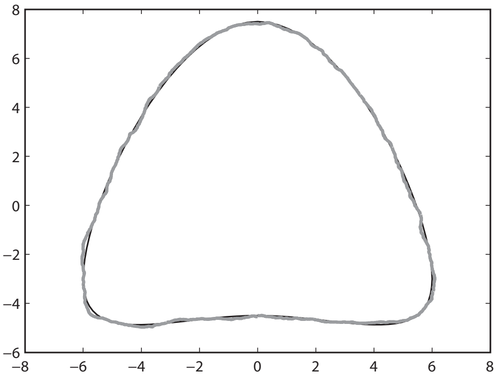


Figure 4.13: Simulation results for unicycle system with the iterative replanning method.

Figure (4.16) shows the reference trajectory y_d, the replanned one y_p and the actual unicycle output y.

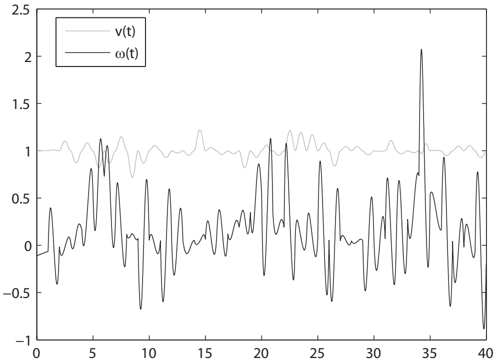


Figure 4.14: The control inputs for the iterative replanning method applied to the unicycle system.

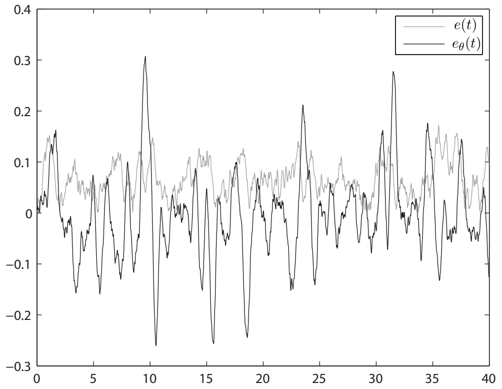


Figure 4.15: The error functions for the iterative replanning method applied to the unicycle system.

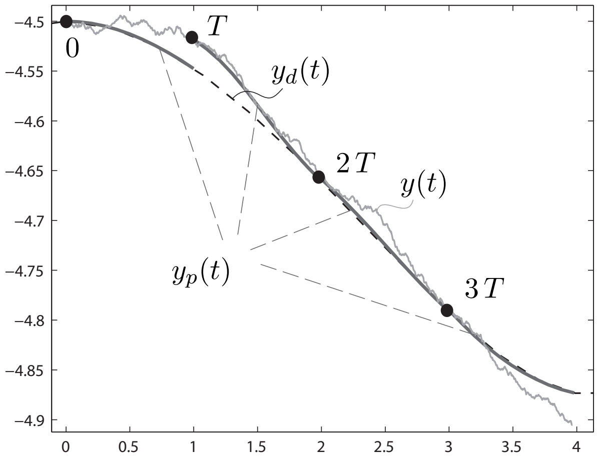


Figure 4.16: The reference trajectory y_d, the replanned one y_p and the actual unicycle output y, for the unicycle example.

Comparison with a well known method

The proposed method has been compared with the same one used for the comparison in subsection 4.1.5 (cf. [68, p.809]), with an observation time set to be $T = 1$ s. The gains in the Samson's controller have been set to obtain control signals of magnitude similar to the introduced method. The results obtained with this method are shown in figures 4.17, 4.18 and 4.19.

The two methods shows a similar performance in terms of tracking error. However, the control method presented in this section has the advantage of providing input signals of an arbitrary degree of continuity, whereas the control signals of the classic controller are discontinuous (even if this is a consequence of having used a discontinuous observer). Moreover, the iterative output replanning method has the advantage of being applicable to any system with a flat performance output.

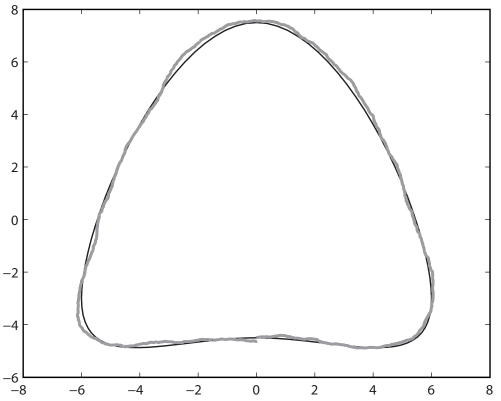


Figure 4.17: Simulation results for the unicycle with the Samson's method.

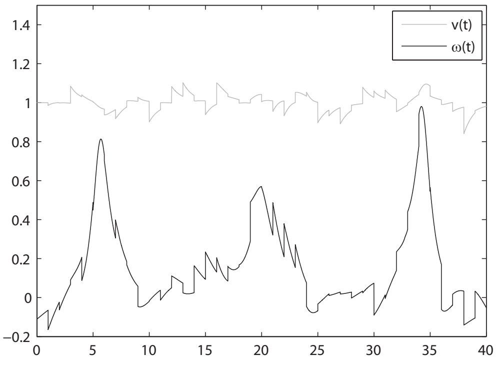


Figure 4.18: The control inputs for the Samson's method applied to the unicycle system.

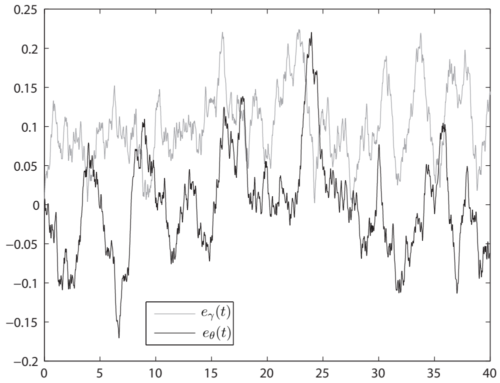


Figure 4.19: The error functions for the Samson's method applied to the unicycle system.

4.2.6 Simulation results for the case of a one-trailer system

The iterative output replanning method has been applied to a *truck and trailer*, or *one trailer*, system (see section 2.2). With respect to the coordinates (x, z) of the middle point of the trailer rear axle, the well-known equations for this system are given by

$$\left\{\begin{array}{rcl} \dot{x} & = & v\cos\theta_1\,\cos(\theta_0-\theta_1)+\eta_x(t) \\ \dot{z} & = & v\sin\theta_1\,\cos(\theta_0-\theta_1)+\eta_z(t) \\ \dot{\theta}_0 & = & \omega+\eta_{\theta_0}(t) \\ \dot{\theta}_1 & = & \frac{v}{d}\,\sin(\theta_0-\theta_1)+\eta_{\theta_1}(t)\,, \end{array}\right.$$

where θ_0 and θ_1 are the orientation angles of the pulling truck and of the trailer respectively, and d represents the distance between the rear point (x, z) of the trailer and the joint point on the truck. The control input function v is the truck longitudinal velocity while ω represent its angular velocity. As shown in [53], this system is flat ($r = 2$) with respect to the performance output $y = (x, z)$. In this case to limit the degree of the interpolation polynomials, we have chosen $l = 0$, obtaining therefore a discontinuous control by proposition 9.

In this simulation, the noise terms have been chosen such that $|\eta_x(t)|$, $|\eta_z(t)|$, $|\eta_{\theta_0}(t)|$, $|\eta_{\theta_1}(t)| \leq 0.2$. The value of the distance d is set to be equal to 1 m.

Truck pulling a trailer

As reference trajectory y_d we have considered the same C^2 periodic spline used for the unicycle example, characterized by a positive constant speed of 1 m/s. The obtained results are presented in figures 4.20, 4.21 and 4.22, where $e_\theta(t) = \theta_1(t) - \arg \dot{y}_d(t)$.

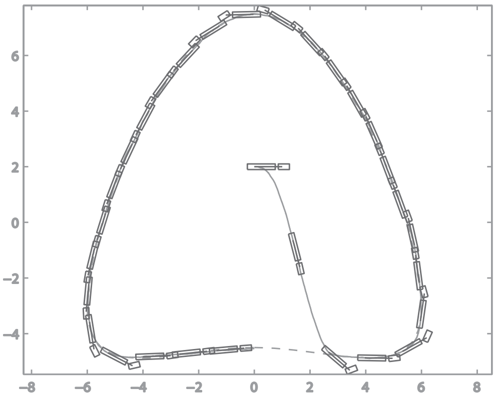


Figure 4.20: Tracking results for the one-trailer system on a periodic spline, in the truck pulling trailer case.

As in the case of the unicycle, the iterative output replanning method shows a good performance in terms of tracking error.

Truck pushing a trailer

As known, the trajectory tracking of trailer system in reverse gear is more difficult than in forward driving, because it is unstable (see [70]). The reference trajectory is again the periodic spline used in the previous examples, with the difference, in this case, of being characterized by a constant negative speed of -1 m/s. The obtained results are presented in figures 4.23, 4.24 and 4.25.

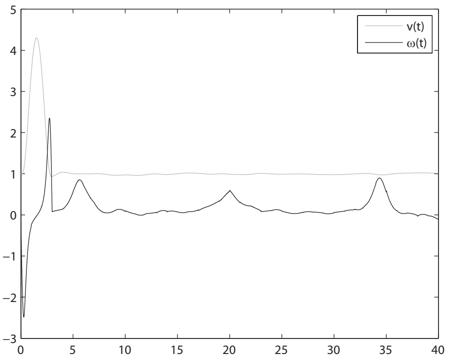
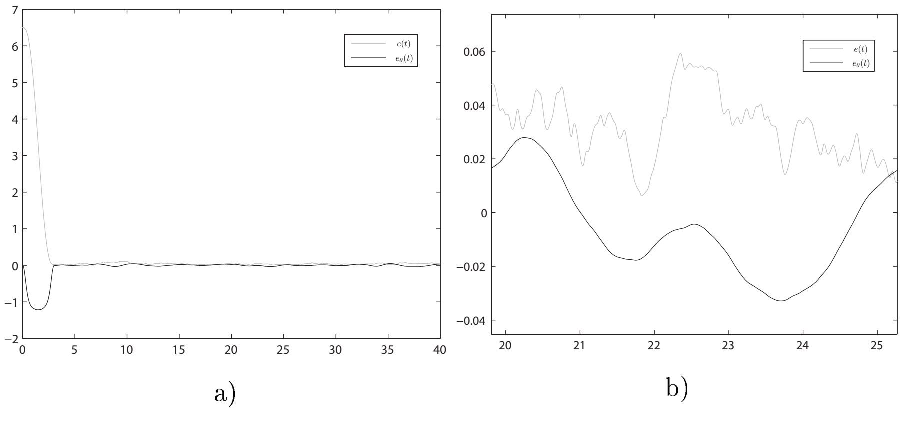


Figure 4.21: The control inputs for the iterative replanning method applied to the one-trailer system, in the truck pulling trailer case.



a)

b)

Figure 4.22: a) The error functions for the iterative replanning method applied to the one-trailer system, in the truck pulling trailer case and b) a close up of it on a smaller time interval.

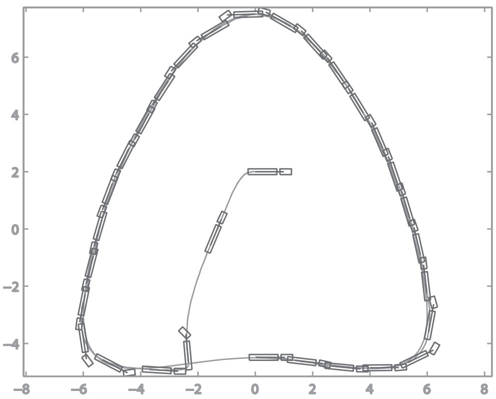


Figure 4.23: Tracking results for the one-trailer system on a periodic spline, in the truck pushing trailer case.

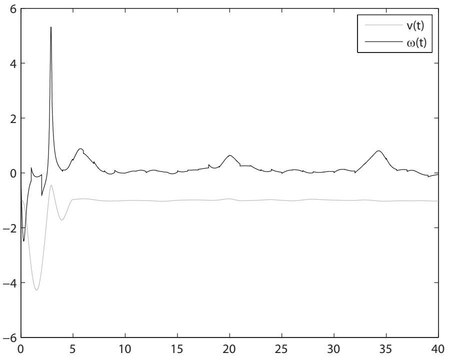


Figure 4.24: The control inputs for the iterative replanning method applied to the one-trailer system, in the truck pushing trailer case.

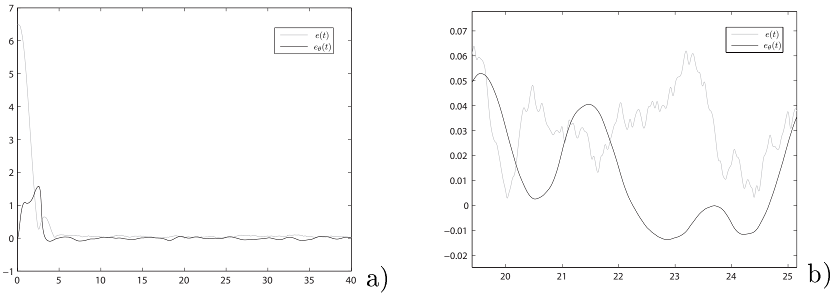


Figure 4.25: a) The error functions for the iterative replanning method applied to the one-trailer system, in the truck pushing trailer case and b) a close up of it on a smaller time interval.

Also in this case, the iterative output replanning method shows a good performance in terms of tracking error.

Conclusions

The thesis has presented some methods useful for the optimal planning and control for the motion of autonomous wheeled vehicles. In particular, the exposed techniques may be applied to the wide class of flat systems (cf. definition 12). Results can be summarized as an hybrid feedforward/feedback control scheme, whose purpose is to guarantee a robust and highly performing control.

High performances are reached out with the planning of time-optimal and continuous velocity profiles and geometrically continuous paths, that lead to a continuous steering input signal. This means that a smooth and optimal motion of the wheeled vehicle can be attained and, in such a way, the vehicle autonomous navigation can perform agile and event-driven maneuvers.

Robustness is achieved by means of iterative trajectory replanning procedures, which guarantee the tracking of the planned trajectory in the presence of noise. It has been proved the existence, for the proposed trajectory planning methods, of closed-form bounds on the tracking error.

Simulation and experimental results obtained during this research point out that the presented methods may be well suited for a real-time implementation provided that some of the required optimizations are done off-line. Indeed, optimal velocity profiles and paths can be generated in real-time using fast local optimization routines based on look-up tables built with off-line optimization.

Bibliography

[1] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[2] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer, Berlin, Germany, 2008.

[3] K. Kant and S.W. Zucker. Toward efficient trajectory planning: the path-velocity decomposition. *Int. J. of Robotics Research*, 5(3):72–89, 1986.

[4] P. Lucibello and G. Oriolo. Robust stabilization via iterative state steering with an application to chained-form systems. *Automatica*, 37(1):71–79, January 2001.

[5] C. Guarino Lo Bianco, A. Piazzi, and M. Romano. Smooth motion generation for unicycle mobile robots via dynamic path inversion. *IEEE Trans. on Robotics*, 20(5):884–891, October 2004.

[6] Velimir Jurdjevic. *Geometric Control Theory*. Cambridge University Press, Cambridge, UK, 1996.

[7] E.D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer-Verlag, New York, second edition, 1998.

[8] E. A. Croft, B. Benhabib, and R. G. Fenton. Near-time optimal robot motion planning for on-line applications. *Journal of Robotic Systems*, 12(8):553–567, March 1995.

[9] L. Consolini and A. Piazzi. Generalized bang-bang control for feedforward constrained regulation. *Automatica*, 45(10):2234–2243, 2009.

[10] L. A. Zadeh. On optimal control and linear programming. *IRE Transactions on Automatic Control*, pages 45–46, 1962.

[11] G. Bashein. A simple algorithm for on-line computation of optimal controls. *IEEE Transactions on Automatic Control*, 1971.

[12] S. Kim, D. Choi, and I. Ha. A comparison principle for state-constrained differential inequalities and its application to time-optimal contol. *IEEE Transactions on Automatic Control*, 2005.

[13] R.E. Bellman. Dynamic programming and lagrange multipliers. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 42, pages 767–769, October 1956.

[14] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenco. *The Mathematical theory of optimal processes.* Wiley, New York, USA, 1962.

[15] A. A. Agrachev and Yu. L. Sachkov. *Control Theory from the Geometric Viewpoint.* Springer-Verlag, Berlin, Germany, 2003.

[16] G. Lini and A. Piazzi. Minimum-time velocity planning with arbitrary boundary conditions. In *Robot Motion and Control 2009*, volume 396 of *Lecture Notes in Control and Information Sciences*, pages 287–296. Springer Berlin and Heidelberg, 2009.

[17] G. Lini, L. Consolini, and A. Piazzi. Minimum-time constrained velocity planning. In *Proceedings of the 2009 17th Mediterranean Conference on Control and Automation*, pages 748–753, Thessaloniki, Greece, July 2009.

[18] R. D. Peters. Ideal lift kinematics: Complete equations for plotting optimum motion. *Proceedings of ELEVCONŠ95 (The International Associa-*

tion of Elevator Engineers), 1995. republished by Elevator World, April 1996 and by Elevatori, May/June 1996.

[19] L. Consolini and A. Piazzi. Generalized bang-bang control for feedforward constrained regulation. *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006. Università degli Studi di Parma.

[20] N. Karmakar. A new polynomial-time algorithm for linear programming. *Report. AT&T Bell Laboratories*, 1984.

[21] L. Consolini, O. Gerelli, C. Guarino Lo Bianco, and A. Piazzi. Flexible joints control: A minimum-time feed-forward technique. *Mechatronics, Elsevier*, pages 348–356, October 2008.

[22] G. Lini, L. Consolini, and A. Piazzi. Multi-optimization of $\boldsymbol{\eta}^3$-splines for autonomous parking. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference, (CDC-ECC), To appear on*, Orlando, Florida, December 2011.

[23] F. Ghilardelli, G. Lini, and A. Piazzi. Path generation using $\boldsymbol{\eta}^4$-splines for a truck and trailer vehicle. Technical Report TSC01/11, Dipartimento di Ingegneria dell'Informazione, Università di Parma (Italy), May 2011.

[24] J.-C. Latombe. *Robot motion planning.* Kluwer, 1991.

[25] J.-P. Laumond, P.E. Jacobs, M. Täix, and R. Murray. A motion planner for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 10(5):577–593, October 1994.

[26] A.W. Divelbiss and J.T.Wen. A path space approach to nonholonomic motion planning in the presence of obstacles. *IEEE Transaction on Robotics and Automation*, 13(3):443–651, June 1997.

[27] Y. Chitour. A continuation method for motion-planning problems. *ESAIM: Control, Optimisation and Calculus of Variations*, 12(1):139–168, 2006.

[28] B. Müller, J. Deutscher, and S. Grodde. Continuous curvature trajectory design and feedforward control for parking a car. *IEEE Transactions on Control Systems Technology*, 15(3):541–553, May 2007.

[29] J. Minguez, F. Lamiraux, and J.-P. Laumond. Motion planning and obstacle avoidance. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 827–852. Springer, 2008.

[30] A. Piazzi, C. Guarino Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi. Quintic G^2-splines for the iterative steering of vision-based autonomous vehicles. *IEEE Trans. on Intelligent Transportations Systems*, 3(1):27–36, March 2002.

[31] G. Lini and A. Piazzi. Time-optimal dynamic path inversion for an automatic guided vehicle. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, pages 5264–5269, Atlanta, GA, December 2010.

[32] A. Piazzi, C. Guarino Lo Bianco, and M. Romano. $\boldsymbol{\eta}^3$-splines for the smooth path generation of wheeled mobile robots. *IEEE Transactions on Robotics*, 23(5):1089–1095, October 2007.

[33] A. Piazzi, C. Guarino Lo Bianco, and M. Romano. Smooth path generation for wheeled mobile robots with $\boldsymbol{\eta}^3$-splines. In F. Casolo, editor, *Motion Control*, chapter 4, pages 75–96. In-Teh sciyo.com, 2010.

[34] K.M. Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, 1998.

[35] D.G. Luenberger. *Linear and Nonlinear Programming.* Addison–Wesley, Reading, Massachusetts, second edition, 1989.

[36] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms.* Wiley, New York, 2001.

[37] F. Lamiraux, J.-P. Laumond, C. Van Geem, D. Boutonnet, and G. Raust. Trailer truck trajectory optimization: the transportation of components for the Airbus A380. *Robotics Automation Magazine, IEEE*, 12(1):14–21, March 2005.

[38] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist. Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles. *Automation Science and Engineering, IEEE Transactions on*, 7(1):167–172, January 2010.

[39] D. Tilbury, R.M. Murray, and S. Shankar Sastry. Trajectory generation for the N-trailer problem using Goursat normal form. *Automatic Control, IEEE Transactions on*, 40(5):802–819, May 1995.

[40] O.J. Sørdalen. Conversion of the kinematics of a car with n trailers into a chained form. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, volume 1, pages 382–387, May 1993.

[41] Dirk J. Struik. *Lectures on classical differential geometry*. Dover Publications, Mineola N.Y., USA, second edition, 1988.

[42] J. Chung and J.M.A. Tanchoco. Material handling automation in production and warehouse systems. In S.Y. Nof, editor, *Springer Handbook of Automation*, pages 961–980. Springer, 2009.

[43] T. Le-Anh and M.B.M. De Koster. A review of design and control of automated guided vehicle systems. *European Journal of Operational Research*, 171(1):1–23, 2006.

[44] H. Wang, Y. Chen, and P. Souères. A geometric algorithm to compute time-optimal trajectories for a bidirectional steered robot. *IEEE Transactions on Robotics*, 25(2):399–413, 2009.

[45] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, and L. Kavraki. *Principles of Robot Motion: Theory, Algorithms, and Implementation*. The MIT Press, 2005.

[46] M. Prado, A. Simon, E. Carabias, A. Perez, and F. Ezquerro. Optimal velocity planning of wheeled mobile robots on specific paths in static and dynamic environments. *Journal of Robotic Systems*, 20(12):737–754, 2003.

[47] B. Widrow and E. Walach. *Adaptive Inverse Control*. Prentice-Hall, Upper Saddle River, NJ, 1995.

[48] A. Piazzi and A. Visioli. Optimal noncausal set-point regulation of scalar systems. *Automatica*, 37(1):121–127, January 2001.

[49] S. Devasia, D. Chen, and B. Paden. Nonlinear inversion-based output tracking. *IEEE Transactions on Automatic Control*, AC-41(7):930–942, July 1996.

[50] L.R. Hunt, G. Meyer, and R. Su. Noncausal inverses for linear systems. *IEEE Transaction on Automatic Control*, AC-41:608–611, 1996.

[51] Q. Zoua and S. Devasia. Precision preview-based stable-inversion for nonlinear nonminimum-phase systems: The vtol example. *Automatica 43 (2007) 117 - 127*, 43:117–127, 2007.

[52] J. Lévine. *Analysis and Control of Nonlinear Systems: a Flatness-based Approach*. Springer, 2009.

[53] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61(6):1327–1361, 1995.

[54] P.F. Muir and C.P. Neuman. Kinematic modeling of wheeled mobile robot. *Journal of Robotic System*, 4(2):281–340, April 1987.

[55] E.D. Anderson and K.D. Anderson. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In H. Frenk, K. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, pages 197–232. Kluwer Academic Publishers, 2000.

[56] A. Isidori. Control theory for automation: fundamentals. In S.Y. Nof, editor, *Springer Handbook of Automation*, pages 147–172. Springer, 2009.

[57] B.A. Francis and W.M. Wonham. The internal model principle of control theory. *Automatica*, 12:457–465, 1976.

[58] G. Basile and G. Marro. *Controlled and Conditioned Invariants in Linear System Theory.* Prentice-Hall, Englewoood Cliffs, New Jersey, 1992.

[59] A. Isidori. *Nonlinear Control Systems.* Springer, London, third edition edition, 1995.

[60] S. Devasia. Should model-based inverse inputs be used as feedforward under plant uncertainty? *IEEE Transactions on Automatic Control*, 47(11):1865–1871, November 2002.

[61] A. Piazzi and A. Visioli. Optimal inversion-based control for the set-point regulation of nonminimum-phase uncertain scalar systems. *IEEE Transactions on Automatic Control*, 46(10):1654–1659, October 2001.

[62] M.K. Bennani and P. Rouchon. Robust stabilization of flat and chained systems. In *Proc. of the European Control Conference*, pages 2642–2646, Rome, Italy, 5-8 September 1995.

[63] P. Morin and C. Samson. Exponential stabilization of nonlinear driftless systems with robustness to unmodeled dynamics,. *ESAIM: Control, Optimisation and Calculus of Variations*, 4:1–35, April 1999.

[64] M. Argenti, L. Consolini, G. Lini, and A. Piazzi. Recursive convex replanning for the trajectory tracking of wheeled mobile robots. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation, (ICRA)*, pages 4916–4921, Anchorage, Alaska, May 2010.

[65] L. Consolini, L. Lini, and A. Piazzi. Iterative output replanning for flat systems affected by additive noise. In *Proceedings of 49th IEEE Con-*

ference on Decision and Control (CDC), pages 6248–6253, Atlanta, GA, December 2010.

[66] L. Consolini, L. Lini, and A. Piazzi. Hermite polynomials for iterative output replanning for flat systems affected by additive noise. *Asian Journal of Control, To appear on*, 2012.

[67] H. Sira-Ramirez and S.K. Agrawal. *Differentially Flat Systems.* Marcel Dekker, New York, 2004.

[68] P. Morin and C. Samson. Motion control of wheeled mobile robots. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, chapter 34. Springer, Berlin, Germany, 2008.

[69] A. Spitzbart. A generalization of hermite's interpolation formula. *The American Mathematical Monthly*, 67(1):42–46, 1960.

[70] Timothy V. Fossum and Gilbert N. Lewis. A mathematical model for trailer-truck jackknifing. *SIAM Review*, 23(1):pp. 95–99, 1981.

Acknowledgements

It is a pleasure to thank the many people who made this thesis possible.

I would like to thank prof. Aurelio Piazzi, my tutor, for giving me advice and motivation and for the trust that has shown in my work.

It is difficult to overstate my gratitude to prof. Luca Consolini. With his enthusiasm, his inspiration, and his great efforts to explain things clearly and simply, he helped to make control theory fun for me. Future development of my research could be the study of a method to solve the open and incredibly difficult problem of optimally organize his desktop.

I wish to thank all my friends in Parma, especially Giuseppe “The Doctor” De Iaco, for helping me get through the difficult times, for all the emotional support, and above all for the good times (as Al Green says).

Special thank goes to my girlfriend, Daniela “Fossi” Bradascio... It’s the most romantic thing I’ve ever done, isn’t it?

Last, and most important, I wish to thank my parents. They raised me, supported me, taught me, and loved me. To them I dedicate this thesis.