

Transform  
PIZ  
for Aut

L  
Claudi

# Trajectory Planning for Automatic Machines and Robots

Luigi Biagiotti · Claudio Melchiorri

# Trajectory Planning for Automatic Machines and Robots



Dr. Luigi Biagiotti  
DII, University of Modena  
and Reggio Emilia  
Via Vignolese 905  
41100 Modena  
Italy  
[luigi.biagiotti@unimore.it](mailto:luigi.biagiotti@unimore.it)

Prof. Claudio Melchiorri  
DEIS, University of Bologna  
Via Risorgimento 2  
40136 Bologna  
Italy  
[cmelchiorri@deis.unibo.it](mailto:cmelchiorri@deis.unibo.it)

ISBN: 978-3-540-85628-3

e-ISBN: 978-3-540-85629-0

Library of Congress Control Number: 2008934462

© Springer-Verlag Berlin Heidelberg 2008

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

*Cover design:* Erich Kirchner, Heidelberg, Germany

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

[springer.com](http://springer.com)

*To Francesca and Morena*

---

## Preface

This book deals with the problems related to planning motion laws and trajectories for the actuation system of automatic machines, in particular for those based on electric drives, and robots. The problem of planning suitable trajectories is relevant not only for the proper use of these machines, in order to avoid undesired effects such as vibrations or even damages on the mechanical structure, but also in some phases of their design and in the choice and sizing of the actuators. This is particularly true now that the concept of “electronic cams” has replaced, in the design of automatic machines, the classical approach based on “mechanical cams”.

The choice of a particular trajectory has direct and relevant implications on several aspects of the design and use of an automatic machine, like the dimensioning of the actuators and of the reduction gears, the vibrations and efforts generated on the machine and on the load, the tracking errors during the motion execution.

For these reasons, in order to understand and appreciate the peculiarities of the different techniques available for trajectory planning, besides the mathematical aspects of their implementation also a detailed analysis in the time and frequency domains, a comparison of their main properties under different points of view, and general considerations related to their practical use are reported.

For these reasons, we believe that the contents of this book can be of interest, besides for students of Electrical and Mechanical Engineering courses, also for engineers and technicians involved in the design and use of electric drives for automatic machines.

We would like to thank all the persons and colleagues which have contributed to this book. In particular, we would like to thank Claudio Bonivento, for the initial suggestions and motivations, and Alberto Tonielli for the discussions on electric drives and their use. The colleagues and friends Roberto Zanasi, Cesare Fantuzzi, and Alessandro De Luca have contributed not only with several constructive comments, but also with the development of some of the algorithms presented in this book.

## VIII Preface

Finally, the help of all the students that have worked on these arguments developing software and executing experimental activities, as well as the co-operations and discussions with technicians and engineers of several industries with their problems related to the design, control, and trajectory planning for automatic machines, are gratefully acknowledged.

Bologna,  
June 2008

Luigi Biagiotti  
Claudio Melchiorri

---

# Contents

<b>1</b>	<b>Trajectory Planning</b>	<b>1</b>
1.1	A General Overview on Trajectory Planning	1
1.2	One-dimensional Trajectories	3
1.3	Mechanical Cams and Electronic Cams	4
1.4	Multi-dimensional Trajectories	6
1.5	Contents and Structure of this Book	8
1.6	Notation	10

---

## Part I Basic Motion Profiles

---

<b>2</b>	<b>Analytic Expressions of Elementary Trajectories</b>	<b>15</b>
2.1	Polynomial Trajectories	15
2.1.1	Linear trajectory (constant velocity)	17
2.1.2	Parabolic trajectory (constant acceleration)	18
2.1.3	Trajectory with asymmetric constant acceleration	21
2.1.4	Cubic trajectory	23
2.1.5	Polynomial of degree five	26
2.1.6	Polynomial of degree seven	28
2.1.7	Polynomials of higher degree	30
2.2	Trigonometric Trajectories	42
2.2.1	Harmonic trajectory	42
2.2.2	Cycloidal trajectory	43
2.2.3	Elliptic trajectory	45
2.3	Exponential Trajectories	47
2.4	Trajectories Based on the Fourier Series Expansion	51
2.4.1	Gutman 1-3	53
2.4.2	Freudenstein 1-3	54
2.4.3	Freudenstein 1-3-5	55

<b>3</b>	<b>Composition of Elementary Trajectories</b>	59
3.1	Linear Trajectory with Circular Blends	59
3.2	Linear Trajectory with Parabolic Blends (Trapezoidal)	62
3.2.1	Trajectory with preassigned acceleration	65
3.2.2	Trajectory with preassigned acceleration and velocity	65
3.2.3	Synchronization of several trapezoidal trajectories	66
3.2.4	Trajectory through a sequence of points	67
3.2.5	Displacement time of a trapezoidal trajectory	69
3.2.6	Trajectory with assigned durations $T$ and $T_a$	69
3.2.7	Trajectory with non-null initial and final velocities	70
3.3	Linear Trajectory with Polynomial Blends	76
3.4	Trajectory with Double S Velocity Profile	79
3.4.1	Computation of the trajectory for $q_1 > q_0$	88
3.4.2	Computation of the trajectory for $q_1 < q_0$	90
3.4.3	Double S with null initial and final velocities	90
3.4.4	On-line computation of the double S trajectory	93
3.4.5	Displacement time of a double S trajectory	101
3.4.6	Double S trajectory with assigned duration of the different phases	102
3.5	Fifteen Segments Trajectory	107
3.6	Piecewise Polynomial Trajectory	117
3.7	Modified Trapezoidal Trajectory	119
3.8	Modified Sinusoidal Trajectory	124
3.9	Modified Cycloidal Trajectory	127
3.10	Constant Velocity/Acceleration Trajectories with Cycloidal or Harmonic Blends	133
3.10.1	Constraints on the velocity profile	133
3.10.2	Constraints on the acceleration profile	135
3.10.3	Minimum-time trajectories	140
3.11	Trajectories with Constant Acceleration and Cycloidal/Cubic Blends	144
<b>4</b>	<b>Multipoint Trajectories</b>	151
4.1	Interpolation by Polynomial Functions	151
4.2	Orthogonal Polynomials	155
4.3	Trigonometric Polynomials	164
4.4	Cubic Splines	166
4.4.1	Computation of the coefficients for assigned initial and final velocities	169
4.4.2	Periodic cubic splines	172
4.4.3	Cubic splines with assigned initial and final velocities: computation based on the accelerations	175
4.4.4	Cubic splines with assigned initial and final velocities and accelerations	177
4.4.5	Smoothing cubic splines	180

4.4.6	Choice of the time instants and optimization of cubic splines .....	188
4.5	B-spline Functions for Trajectories with High Degree of Continuity .....	194
4.6	Nonlinear Filters for Optimal Trajectory Planning .....	208
4.6.1	Online trajectory planner with velocity, acceleration and jerk constraints .....	209
4.6.2	Online trajectory planner with velocity and acceleration constraints .....	216

## Part II Elaboration and Analysis of Trajectories

<b>5</b>	<b>Operations on Trajectories</b> .....	223
5.1	Geometric Modification of a Trajectory.....	223
5.2	Scaling in Time .....	228
5.2.1	Kinematic scaling .....	230
5.2.2	Dynamic Scaling .....	236
5.3	Synchronization of Trajectories .....	241
<b>6</b>	<b>Trajectories and Actuators</b> .....	245
6.1	Trajectories and Electric Motors.....	245
6.1.1	Trajectories and choice of the actuator .....	247
6.2	Characteristics of the Motion Profiles .....	250
6.2.1	Comparison between trapezoidal and double S trajectories .....	256
<b>7</b>	<b>Dynamic Analysis of Trajectories</b> .....	265
7.1	Models for Analysis of Vibrations.....	265
7.1.1	Linear model with one degree of freedom .....	266
7.1.2	Linear model with $n$ degrees of freedom .....	267
7.1.3	Nonlinear model with one degree of freedom .....	269
7.1.4	Nonlinear model with $n$ degrees of freedom .....	270
7.2	Analysis of the Trajectories in the Time Domain .....	271
7.3	Analysis of the Trajectories in the Frequency Domain .....	285
7.3.1	Frequency spectrum of some elementary trajectories ..	287
7.3.2	Numerical computation of the frequency spectrum of generic trajectories .....	294
7.3.3	Harmonic content of periodic trajectories .....	299
7.3.4	Scaling and frequency properties of a trajectory ..	303
7.4	Frequency Modifications of Trajectories .....	304
7.4.1	Polydyne and splinedyne functions .....	305
7.4.2	Input filtering and shaping .....	318
7.4.3	Feedforward based on the inversion of the plant dynamics .....	330

---

**Part III Trajectories in the Operational Space**

---

**8 Multidimensional Trajectories and Geometric Path**

<b>Planning</b> . . . . .	341
8.1 Introduction . . . . .	341
8.1.1 Continuity of the geometric path and continuity of the trajectory . . . . .	343
8.1.2 Global and local interpolation/approximation . . . . .	346
8.2 Orientation of the Tool . . . . .	347
8.2.1 Case of independent position and orientation . . . . .	347
8.2.2 Case of position and orientation coupled . . . . .	353
8.3 Definition of the Geometric Path Through Motion Primitives . . . . .	356
8.4 Global Interpolation . . . . .	359
8.4.1 Definition of the set $\{\bar{u}_k\}$ . . . . .	359
8.4.2 Cubic B-spline interpolation . . . . .	360
8.5 Global Approximation . . . . .	364
8.5.1 Knots choice . . . . .	366
8.6 A Mixed Interpolation/Approximation Technique . . . . .	368
8.7 Smoothing Cubic B-splines . . . . .	371
8.7.1 Smoothing B-splines with assigned start/end points and directions . . . . .	373
8.8 B-spline Functions for Trajectories with High Degree of Continuity . . . . .	376
8.9 Use of Nurbs for Trajectory Generation . . . . .	391
8.10 Local Interpolation with Bézier Curves . . . . .	393
8.10.1 Computation of the tangent and curvature vectors . . . . .	394
8.10.2 Cubic Bézier curves interpolation . . . . .	395
8.10.3 Quintic Bézier curves interpolation . . . . .	400
8.11 Linear Interpolation with Polynomial Blends . . . . .	406

**9 From Geometric Paths to Trajectories** . . . . . 415

9.1 Introduction . . . . .	415
9.2 Constant Scaling . . . . .	416
9.3 Generic Motion Law . . . . .	418
9.4 Constant Feed Rate . . . . .	421
9.5 Generic Feed Rate Profile . . . . .	424
9.6 Integration of Geometric Path and Motion Law for Complex 3D Tasks . . . . .	429
9.6.1 Linear trajectory with polynomial blends . . . . .	429
9.6.2 B-spline trajectory . . . . .	440
9.6.3 Smoothing B-spline trajectory . . . . .	445
9.6.4 B-spline approximation of a trajectory based on motion primitives . . . . .	449

---

**Part IV Appendices**


---

<b>A Numerical Issues .....</b>	457
A.1 Parameters of normalized polynomials $q_N(\tau)$ .....	457
A.2 Parameters of the Trajectory ‘4-3-4’ .....	461
A.3 Solution of the Equation $M \ k = q$ .....	461
A.4 Efficient Evaluation of Polynomial Functions .....	463
A.5 Numerical Solution of Tridiagonal Systems .....	464
A.5.1 Tridiagonal systems .....	464
A.5.2 Cyclic tridiagonal systems .....	465
<b>B B-spline, Nurbs and Bézier curves .....</b>	467
B.1 B-spline Functions .....	467
B.1.1 B-spline basis functions .....	467
B.1.2 Definition and properties of B-splines .....	471
B.1.3 Evaluation of a B-spline curve .....	474
B.1.4 Derivative of a B-spline curve .....	475
B.1.5 Conversion from B-form to Piecewise Polynomial form (pp-form) .....	479
B.2 Definition and Properties of Nurbs .....	481
B.3 Definition and Properties of Bézier Curves .....	483
B.3.1 Evaluation of a Bézier curve .....	484
B.3.2 Derivatives of a Bézier curve .....	486
<b>C Representation of the Orientation .....</b>	489
C.1 Rotation Matrices .....	489
C.1.1 Elementary rotation matrices .....	490
C.2 Angle-Axis Representation .....	490
C.3 Euler Angles .....	491
C.4 Roll-Pitch-Yaw Angles .....	493
<b>D Spectral Analysis and Fourier Transform .....</b>	495
D.1 Fourier Transform of a Continuous Time Function .....	495
D.1.1 Main properties of the Fourier transform .....	496
D.2 Fourier Series of a Periodic Continuous Function .....	497
D.3 Fourier Transform of a Discrete Time Function .....	498
D.3.1 Discrete Fourier transform .....	499
D.4 Fourier Analysis of Signals Using DFT (and FFT) .....	500
<b>References .....</b>	503
<b>Index .....</b>	509

# 1

---

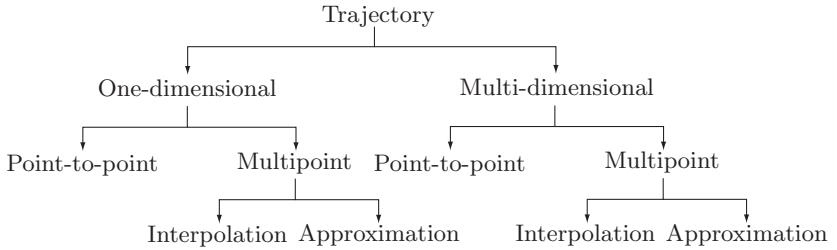
## Trajectory Planning

This book deals with the problem of *trajectory planning*, i.e. of the computation of desired motion profiles for the actuation system of automatic machines. Because of their wide use, only electric drives are considered here, and their motion is defined in the context of the real-time control of automatic machines with one or more actuators, such as packaging machines, machine-tools, assembly machines, industrial robots, and so on. In general, for the solution of this problem some specific knowledge about the machine and its actuation system is also required, such as the *kinematic model* (*direct and inverse*) (usually the desired movement is specified in the *operational space*, while the motion is executed in the *actuation space* and often these domains are different) and the *dynamic model of the system* (in order to plan suitable motion laws that allow to execute the desired movement with proper loads and efforts on the mechanical structure). Moreover, for the real-time execution of the planned motion, it is necessary to define proper *position/velocity control algorithms*, in order to optimize the performances of the system and to compensate for disturbances and errors during the movements, such as saturations of the actuation system. Several techniques are available for planning the desired movement, each of them with peculiar characteristics that must be well known and understood. In this book, the most significant and commonly adopted techniques for trajectory planning are illustrated and analyzed in details, taking into account the above mentioned problems.

---

### 1.1 A General Overview on Trajectory Planning

Basically, the trajectory planning problem consists in finding a relationship between two elements belonging to different domains: time and space. Accordingly, the trajectory is usually expressed as a parametric function of the time,



**Fig. 1.1.** Main trajectory categories.

which provides at each instant the corresponding desired position. Obviously, after having defined this function, also other aspects related to its implementation must be considered, such as time discretization (automatic machines are controlled by digital control systems), saturation of the actuation system, vibrations induced on the load, and so on.

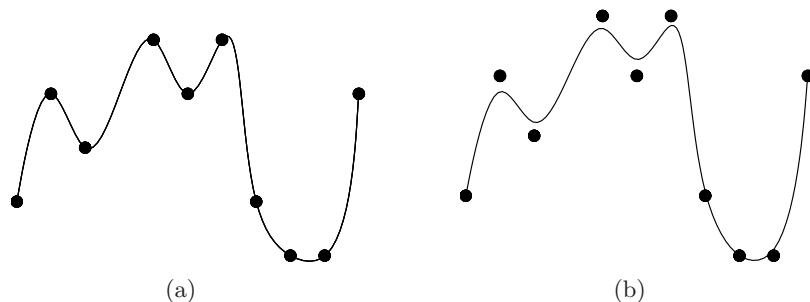
As shown in Fig. 1.1, the main distinction among the various categories of trajectories consists in the fact that they can be one- or multi-dimensional. In the first case they define a position for a one degree-of-freedom (dof) system, while in the latter case a multidimensional working space is considered. From a formal point of view, the difference between these two classes of trajectories consists in the fact that they are defined by a scalar ( $q = q(t)$ ) or a vectorial ( $\mathbf{p} = \mathbf{p}(t)$ ) function. However, the differences are deeper if one considers the approaches and the tools used in the two cases for their computation. Between one- and multi-dimensional trajectories, there is a class of trajectories with intermediate characteristics, namely single-axis motion laws applied to a multi-axis system, composed by several actuators arranged in a so-called master-slave configuration. In this case the motions of the single actuators, although one-dimensional, cannot be designed separately but must be properly coordinated/synchronized<sup>1</sup>.

In this book, the design of one-dimensional trajectories is firstly considered. Then, the problem of their coordination/synchronization is addressed and, finally, the planning of motions in the three-dimensional space is taken into account.

The techniques reported in this book, both for one-dimensional and multi-dimensional trajectories, are also classified depending on the fact that the desired motion is defined by assuming initial and final points only (*point-to-point trajectories*) or by considering also a set of intermediate via-points which must properly interpolated/approximated (*multipoint trajectories*). In

---

<sup>1</sup> In the literature, the two terms “coordination” and “synchronization” are used as synonyms [1]



**Fig. 1.2.** Interpolation (a) and approximation (b) of a set of data points.

the former case, a complex motion is obtained by joining several<sup>2</sup> point-to-point trajectories which are individually optimized by considering for each of them the initial and final boundary conditions on velocity, acceleration, etc., and the constraints on their maximum values. Conversely, in the case of multipoint trajectories), by specifying the intermediate points it is possible to define arbitrarily complex motions and the trajectory is found as the solution of a global optimization problem which depends on the conditions imposed on each via-point and on the overall profile. Moreover, it is possible to adopt different criteria for the definition of the motion profile on the basis of the given via-points, which are not necessarily crossed by the trajectory. In particular, two types of fitting can be distinguished:

- *Interpolation*: the curve crosses the given points for some values of the time, Fig. 1.2(a).
- *Approximation*: the curve does not pass exactly through the points, but there is an error that may be assigned by specifying a prescribed tolerance, Fig. 1.2(b).

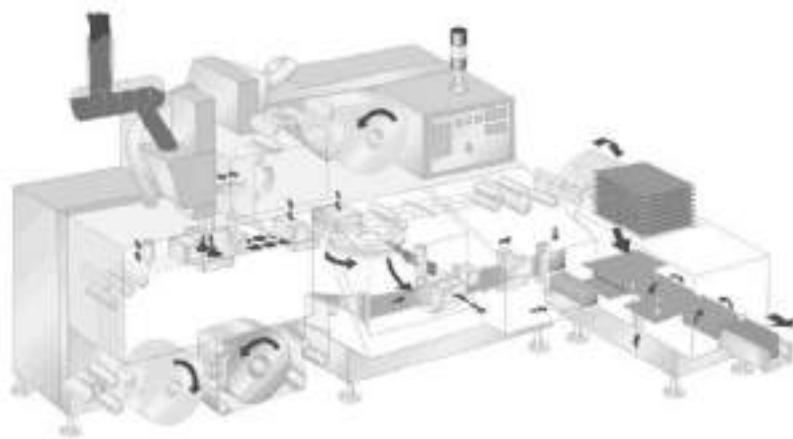
The latter approach can be useful when, especially in multi-dimensional trajectories, a reduction of the speed/acceleration values along the curve is desirable, at the expense of a lower precision.

## 1.2 One-dimensional Trajectories

Nowadays the design of high speed automatic machines, whose actuation systems is mainly based on electric drives, generally involves the use of several actuators distributed in the machine and of relatively simple mechanisms, see for example Fig. 1.3, where the sketch of a packaging machine is reported. About twenty motion axes are present in a machine of this type. The so-called *electronic cams* and *electronic gears* are employed for the generation

---

<sup>2</sup> At least two (three) segments are necessary for a typical periodic motion composed by a rise and a return phase (and, in case, by a dwell phase).

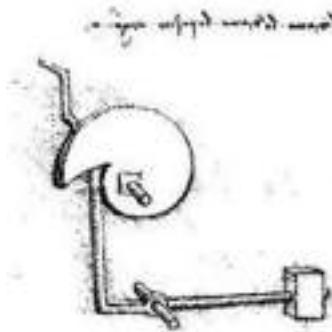


**Fig. 1.3.** Sketch of an automatic machine for tea packaging (courtesy IMA).

of motion where needed, in place of a single or few actuators and complex kinematic chains. In this manner, more flexible machines can be obtained, able to cope with the different production needs required from the market, [2]. In this context, the problem of trajectory planning has assumed more and more importance [3] since, once the displacement and its duration have been defined, the choice of the modality of motion from the initial to the final point has important implications with respect to the sizing of the actuators, the efforts generated on the structure, and the tracking capabilities of the specified motion (tracking error). Therefore, it is necessary to carefully consider the different types of point-to-point trajectories which could be employed with a specific system (actuation and load). As a matter of fact, for given boundary conditions (initial and final positions, velocities, accelerations, etc.) and duration, the typology of the trajectory has a strong influence on the peak values of the velocity and acceleration in the intermediate points, as well as on the spectral content of the resulting profile. For this reason, in the first part of the book the most common families of trajectories used in the industrial practice are described, providing their analytical expression. Then, these trajectories are analyzed and compared, by taking into account both the frequency aspects and the achievable performances for the overall machine.

### 1.3 Mechanical Cams and Electronic Cams

*Mechanical cams* have a very long history. Although some authors trace back their origin even to the Paleolithic age, as referred in [4], certainly Leonardo da Vinci can be considered as one of the first pioneers of the 'modern' de-



**Fig. 1.4.** A mechanical cam designed by Leonardo da Vinci.

sign of cam mechanisms, with his design of some machines based on these mechanisms, Fig. 1.4.

In the last decades, mechanical cams have been widely used in automatic machines for transferring, coordinating and changing the type of motion from a *master* device to one or more *slave* systems, Fig. 1.5. With reference to Fig. 1.6 the body C, the *cam*, is supposed to move at a constant rotational velocity, and therefore its angular position  $\theta$  is a linear function of time. The body F, the *follower*, has an alternative motion  $q(\theta)$  defined by the profile of the cam. The design of mechanical cams, especially for planar mechanisms, has been extensively and carefully investigated, and a wide literature is available on this topic, see for example [4, 5, 6, 7, 8, 9].

As already mentioned, *mechanical cams* are nowadays substituted more and more often by the so-called *electronic cams*. The goal is to obtain more

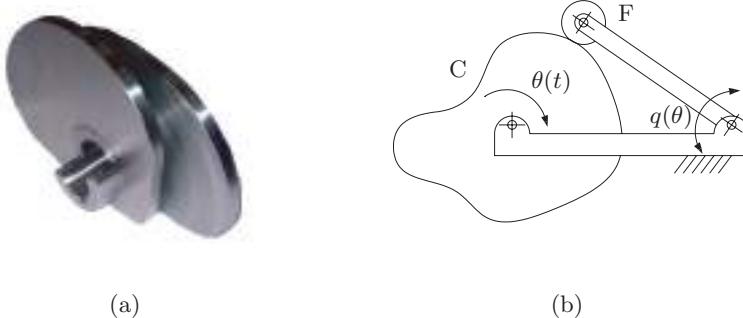


(a)



(b)

**Fig. 1.5.** Mechanical cams, part of an automatic machine (courtesy IMA).



**Fig. 1.6.** (a) A mechanical cam; (b) working principle of a simple mechanical cam (C) with the follower (F).

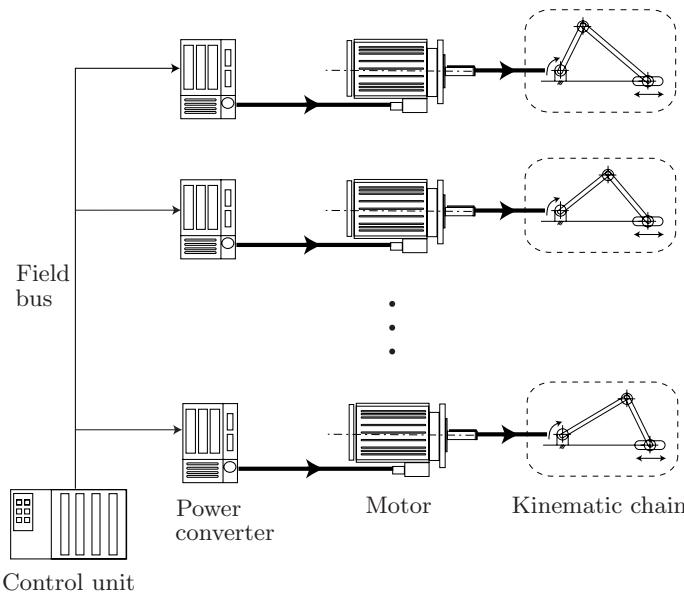
flexible machines, with improved performances, easy to be re-programmed, and possibly at lower costs. With electronic cams, the motion  $q(t)$  is directly obtained by means of an electric actuator, properly programmed and controlled to generate the desired motion profile. Therefore, the need of designing cams to obtain the desired movement has been progressively replaced by the necessity of planning proper trajectories for electric motors.

In multi-axis machines based on mechanical cams, the synchronization of the different axes of motion is simply achieved by connecting the slaves to a single master (the coordination is performed at the mechanical level), while in case of electronic cams the problem must be considered in the design of the motion profiles for the different actuators (the synchronization is performed at the software level, see Fig. 1.7). A common solution is to obtain the synchronization of the motors by defining a master motion, that can be either virtual (generated by software) or real (the position of an actuator of the machine), and then by using this master position as “time” (i.e. the variable  $\theta(t)$  in Fig. 1.6(b)) for the other axes.

## 1.4 Multi-dimensional Trajectories

Properly speaking, the term *trajectory* denotes a path in the three-dimensional space. For example, the Merriam-Webster dictionary defines the trajectory as “the curve that a body describes in space”, [10].

Although in the case of a machine composed by several motors each of them can be independently programmed and controlled (control in the *joint space*), many applications require a coordination among the different axes of motion with the purpose of obtaining a desired multi-dimensional trajectory in the *operational space* of the machine. This is the case of tool machines used to cut, mill, drill, grind, or polish a given workpiece, or of robots which



**Fig. 1.7.** Structure of a multi-axis system based on electronic cams.

must perform tasks in the three-dimensional space, such as spot welding, arc welding, handling, gluing, etc.

In these applications, it is necessary to specify

1. The *geometric path*  $\mathbf{p} = \mathbf{p}(u)$  to be followed, including also the orientation along the curve.
2. The modality by means of which the geometric path must be tracked, that is the *motion law*  $u = u(t)$ .

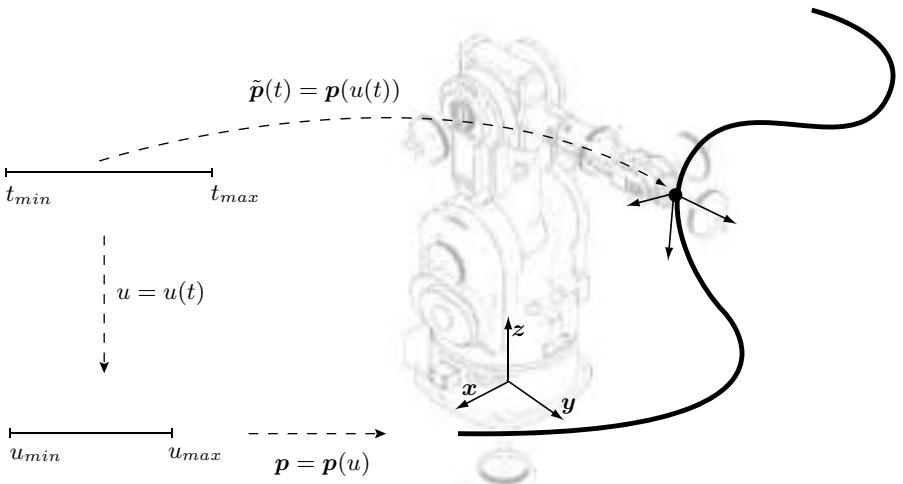
The curve followed by the end effector must be designed on the basis of the constraints imposed by the task (e.g. the interpolation of a given set of via-points), while the determination of the motion law descends from other constraints, such as the imposition of the conditions on the maximum velocities, accelerations, and torques that the actuation system is able to provide.

From the composition of the geometric path and of the motion law the complete trajectory is obtained

$$\tilde{\mathbf{p}}(t) = \mathbf{p}(u(t))$$

as shown in Fig. 1.8. Once the desired movement is specified, the inverse kinematics<sup>3</sup> of the mechanism is employed to obtain the corresponding trajectory in the actuation (joint) space, where the motion is generated and controlled.

<sup>3</sup> The *direct kinematics* of a mechanical device is a (nonlinear) function  $\mathbf{q} \rightarrow \mathbf{p} = \mathbf{f}(\mathbf{q})$  mapping the joint positions  $\mathbf{q} = [q_1, q_2, \dots, q_n]^T$  (i.e. the actuators' positions) to the corresponding position/orientation  $\mathbf{p}$  of a specific point of the



**Fig. 1.8.** A multi-dimensional trajectory defined in the working space of an industrial robot (courtesy COMAU).

## 1.5 Contents and Structure of this Book

A relevant, detailed bibliography is available for the problem of moving parts of automatic machines by means of mechanical cams, and in particular for the problem of the determination of the best cam profile in order to obtain the desired motion at the load. As already mentioned, among the numerous and good reference books, one can refer for example to [4, 5, 6, 7, 8, 9]. On the other hand, a similar bibliography concerning the solution of the same problems by means of electric actuators is not currently available. These problems, although in a rather simplified fashion, are partially faced in robotics [11, 12, 13], but limited to the illustration of simple motion profiles and planning of operational space trajectories.

In this book, the main problems related to the planning of trajectories in the joint space are discussed, with particular reference to electric actuators for automatic machines. The case of trajectories defined in the operational space is also considered, discussing the interpolation and approximation techniques for planning motions in the 3D space.

Specifically, the following topics are illustrated:

### Part 1 Basic motion profiles

- Chapter 2. The basic functions for defining simple trajectories are illustrated: polynomial, trigonometric, exponential and based on the Fourier

---

machine in the operational space. The *inverse kinematics* is the inverse function  $\mathbf{p} \rightarrow \mathbf{q} = \mathbf{f}^{-1}(\mathbf{p}) = \mathbf{g}(\mathbf{p})$ .

series expansion. The main properties of these basic functions are presented and discussed.

- Chapter 3. More complex trajectories are presented, defined in order to obtain specific characteristics in terms of motion, velocity, acceleration, such as the trapezoidal or the double S.
- Chapter 4. Trajectories interpolating a set of via-points are presented. In particular, the interpolation by means of polynomial functions, the cubic splines, the B-splines, and techniques for the definition of “optimal” (i.e. minimum time) trajectories are illustrated.

## **Part 2 Elaboration and analysis of trajectories**

- Chapter 5. The problems of kinematic and dynamic “scaling” of a trajectory are discussed. Comments on the synchronization of several motion axes are given.
- Chapter 6. The trajectories are analyzed and compared by taking into account the effects produced on the actuation system. For this purpose, the maximum and the root mean square values of the velocities and accelerations, consequence of the different motion profiles, are taken into account.
- Chapter 7. The trajectories are analyzed by considering their frequency properties and their influence on possible vibration phenomena in the mechanical system.

## **Part 3 Trajectories in the operational space**

- Chapter 8. The problem of trajectory planning for automatic machines, and in particular for robot manipulators, is considered in the operational space. The basic tools to solve this problem are illustrated, along with some examples.
- Chapter 9. The problem of the analytical composition of the geometric path with the motion law is considered in detail. The goal is to define parametric functions of time so that given constraints on velocities, accelerations, and so on, are satisfied.

Four appendices close the book, with details about some aspects related to the computational issues for one-dimensional trajectories, namely efficient polynomial evaluation, matrix inversion and so on (Appendix A), the B-spline, Nurbs and Bézier definitions and properties (Appendix B), the tools for the definition of the orientation in three-dimensional space (Appendix C), and the spectral analysis of analog and digital signals (Appendix D).

## 1.6 Notation

In this book, the following notation is adopted.

### One-dimensional trajectories:

$q(t)$	: position profile
$t$	: independent variable, that can be either the “time” (as normally assumed in the book) or the angular position $\theta$ of the master in a system based on electronic cams
$q^{(1)}(t), \dot{q}(t)$	: time-derivative of the position (velocity profile)
$q^{(2)}(t), \ddot{q}(t)$	: time-derivative of the velocity (acceleration profile)
$q^{(3)}(t), \dddot{q}(t)$	: time-derivative of the acceleration (jerk profile)
$q^{(4)}(t)$	: time-derivative of the jerk (snap, jounce or ping profile)
$s(t)$	: spline function
$q_k(t)$	: $k$ -th position segment ( $k = 0, \dots, n - 1$ ) in multi-segment trajectories
$\tilde{q}(t')$	: reparameterization of $q(t)$ (scaling in time), $\tilde{q}(t') = q(t)$ with $t = \sigma(t')$
$t_0, t_1$	: initial and final time instants in point-to-point motions
$T$	: total duration of a point-to-point trajectory ( $T = t_1 - t_0$ )
$q_0, q_1$	: initial and final via-points in point-to-point motions
$h$	: total displacement ( $h = q_1 - q_0$ )
$q_k$	: $k$ -th via-points ( $k = 0, \dots, n$ ) in multipoint trajectories
$t_k$	: $k$ -th time instant ( $k = 0, \dots, n$ ) in multipoint trajectories
$T_k$	: duration of the $k$ -th segment ( $T_k = t_{k+1} - t_k$ ) in multi-segment trajectories
$v_0, v_1$	: initial and final velocity in point-to-point motions
$a_0, a_1$	: initial and final acceleration in point-to-point motions
$j_0, j_1$	: initial and final jerk in point-to-point motions
$v_0, v_n$	: initial and final velocity in multipoint motions
$a_0, a_n$	: initial and final acceleration in multipoint motions
$j_0, j_n$	: initial and final jerk in multipoints motions
$v_{max}$	: maximum speed value
$a_{max}$	: maximum acceleration value
$j_{max}$	: maximum jerk value

### Multi-dimensional trajectories:

$\mathbf{p}(u)$	: geometric path
$p_x, p_y, p_z$	: $x-$ , $y-$ , $z-$ components of the curve $\mathbf{p}$
$u$	: independent variable for parametric functions describing a geometric path
$u(t)$	: function of time defining the motion law
$\mathbf{p}^{(1)}(u)$	: derivative of the position (tangent vector) with respect to $u$
$\mathbf{p}^{(2)}(u)$	: derivative of the tangent vector (curvature vector) with respect to $u$
$\mathbf{p}^{(i)}(u)$	: $i$ -th time-derivative of the geometric path $\mathbf{p}(u)$
$\mathbf{p}_k(u)$	: $k$ -th curve segment ( $k = 0, \dots, n - 1$ ) in multi-segment trajectories
$s(u)$	: B-spline function
$n(u)$	: Nurbs function
$b(u)$	: Bézier function
$\tilde{\mathbf{p}}(t)$	: position trajectory obtained by composing the geometric path with the motion law, $\tilde{\mathbf{p}}(t) = \mathbf{p}(u) \circ u(t)$
$\tilde{p}_x, \tilde{p}_y, \tilde{p}_z$	: $x-, y-, z-$ components of the trajectory $\tilde{\mathbf{p}}$ as a function of the time $t$
$\tilde{\mathbf{p}}^{(i)}(t)$	: $i$ -th derivative of the trajectory ( $i = 1$ velocity, $i = 2$ acceleration, etc.)
$\tilde{p}_x^{(i)}, \tilde{p}_y^{(i)}, \tilde{p}_z^{(i)}$	: $x-, y-, z-$ components of $\tilde{\mathbf{p}}^{(i)}$
$\hat{\mathbf{p}}(\hat{u})$	: parameterization of the function $\mathbf{p}(u)$ , $\hat{\mathbf{p}}(\hat{u}) = \mathbf{p}(u) \circ u(\hat{u})$
$q_k$	: $k$ -th via-points ( $k = 0, \dots, n$ ) in multipoints trajectories
$R_k$	: rotation matrix defining the orientation at the $k$ -th via-point
$t_k$	: tangent vector at the generic $k$ -th via-point
$\bar{u}_k$	: $k$ -th “time instant” ( $k = 0, \dots, n$ ) in multipoints trajectories
$t_0, t_n$	: tangent vectors at the initial and final points in multipoints motions
$n_0, n_n$	: curvature vectors at the initial and final points in multipoints motions
$G^h$	: class of functions with geometric continuity up to the order $h$

$\mathbb{N}$	: set of natural numbers
$\mathbb{R}$	: set of real numbers
$\mathbb{C}$	: set of complex numbers
$m$	: scalar number
$ m $	: absolute value
$\mathbf{m}$	: vector
$ \mathbf{m} $	: vector norm
$\mathbf{m}^T$	: transpose of the vector $\mathbf{m}$
$\mathbf{M}$	: matrix
$ \mathbf{M} $	: matrix norm
$ \mathbf{M} _F$	: Frobenius norm of matrix $\mathbf{M}$
$\text{tr}(\mathbf{M})$	: trace of matrix $\mathbf{M}$
$\text{diag}\{m_1, \dots, m_{n-1}\}$	: diagonal matrix
$\omega$	: angular frequency
$T_s$	: sampling time
$C^h$	: class of functions continuous up to the $h$ -th derivative
$\text{floor}(\cdot)$	: integer part function
$\text{sign}(\cdot)$	: sign function
$\text{sat}(\cdot)$	: saturation function
$m!$	: factorial operator

Sometimes, these symbols have different meanings. Where not explicitly indicated, the new meaning is clear from the context.

For the sake of simplicity, the numerical values used in this book are considered dimensionless. In this manner, the mathematical expressions can be applied without changes to several practical cases, with different physical dimensions. In particular, positions may refer to meters, degrees, radians, ...; velocities may then refer to meters/second, degrees/second, ...; and so on.

Finally, it is worth noticing that, without loss of generality, the algorithms for one-dimensional trajectories assume that  $q_1 > q_0$ , and therefore the desired displacement  $h = q_1 - q_0$  is always positive. If this is not the case, the basic motion profiles are unchanged, while the motions based on composition of elementary trajectories (described in Ch. 3) require the adoption of the procedure reported in Sec. 3.4.2.

## **Part I**

---

### **Basic Motion Profiles**

## Analytic Expressions of Elementary Trajectories

The basic trajectories are illustrated, classified into three main categories: polynomial, trigonometric, and exponential. Trajectories obtained on the basis of Fourier series expansion are also explained. More complex trajectories, able to satisfy desired constraints on velocity, acceleration and jerk, can be obtained by means of a suitable composition of these elementary functions. The case of a single actuator, or axis of motion, is specifically considered. The discussion is general, and it is therefore valid to define both a trajectory in the joint space and a motion law in the operational space, see Chapter 8 and Chapter 9.

---

### 2.1 Polynomial Trajectories

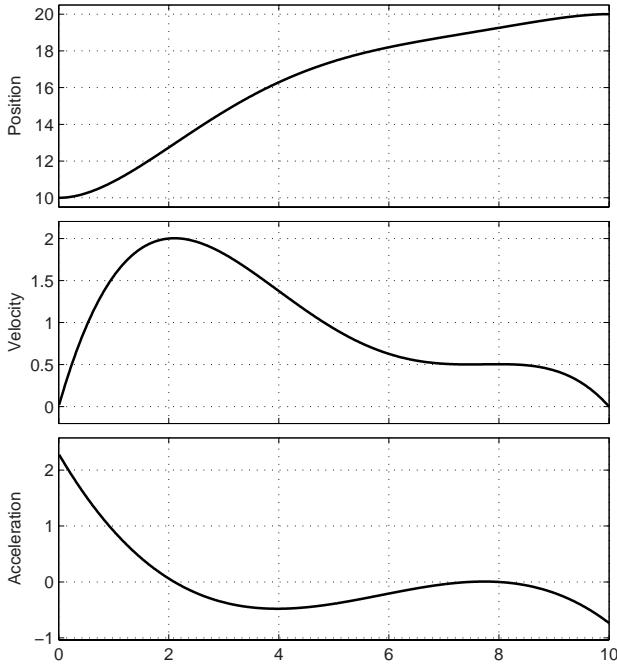
In the most simple case, a motion is defined by assigning the initial and final time instant  $t_0$  and  $t_1$ , and conditions on position, velocity and acceleration at  $t_0$  and  $t_1$ . From a mathematical point of view, the problem is then to find a function

$$q = q(t), \quad t \in [t_0, t_1]$$

such that the given conditions are satisfied. This problem can be easily solved by considering a polynomial function

$$q(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_n t^n$$

where the  $n+1$  coefficients  $a_i$  are determined so that the initial and final constraints are satisfied. The degree  $n$  of the polynomial depends on the number of conditions to be satisfied and on the desired “smoothness” of the resulting motion. Since the number of boundary conditions is usually even, the degree  $n$  of the polynomial function is odd, i.e. three, five, seven, and so on.



**Fig. 2.1.** Position, velocity and acceleration profiles of a polynomial trajectory computed by assigning boundary and intermediate conditions (Example 2.1).

In general, besides initial and final conditions on the trajectory, other conditions could be specified concerning its time derivatives (velocity, acceleration, jerk, ...) at generic instants  $t_j \in [t_0, t_1]$ . In other words, one could be interested in determining a polynomial function  $q(t)$  whose  $k$ -th time-derivative assumes a specific value  $q^{(k)}(t_j)$  at a given instant  $t_j$ . Mathematically, these conditions can be specified as

$$k! a_k + (k+1)! a_{k+1} t_j + \dots + \frac{n!}{(n-k)!} a_n t_j^{n-k} = q^{(k)}(t_j)$$

or, in matrix form, as

$$\mathbf{M} \mathbf{a} = \mathbf{b}$$

where  $\mathbf{M}$  is a known  $(n+1) \times (n+1)$  matrix,  $\mathbf{b}$  collects the given  $(n+1)$  conditions to be satisfied, and  $\mathbf{a} = [a_0, a_1, \dots, a_n]^T$  is the vector of the unknown parameters to be computed. In principle this equation can be solved simply as

$$\mathbf{a} = \mathbf{M}^{-1} \mathbf{b}$$

although, for large values of  $n$ , this procedure may lead to numerical problems. These considerations are analyzed in more details in Chapter 4.

**Example 2.1** Fig. 2.1 shows the position, velocity and acceleration profiles of a polynomial trajectory computed by assigning the following conditions:

$$\begin{aligned} q_0 &= 10, & q_1 &= 20, & t_0 &= 0, & t_1 &= 10, \\ v_0 &= 0, & v_1 &= 0, & v(t=2) &= 2, & a(t=8) &= 0. \end{aligned}$$

There are four boundary conditions (position and velocity at  $t_0$  and  $t_1$ ) and two intermediate conditions (velocity at  $t = 2$  and acceleration at  $t = 8$ ). Note that with six conditions it is necessary to adopt a polynomial at least of degree five. In this case, the coefficients  $a_i$  result

$$\begin{aligned} a_0 &= 10.0000, & a_1 &= 0.0000, & a_2 &= 1.1462, \\ a_3 &= -0.2806, & a_4 &= 0.0267, & a_5 &= -0.0009. \end{aligned}$$

□

### 2.1.1 Linear trajectory (constant velocity)

The most simple trajectory to determine a motion from an initial point  $q_0$  to a final point  $q_1$ , is defined as

$$q(t) = a_0 + a_1(t - t_0).$$

Once the initial and final instants  $t_0$ ,  $t_1$ , and positions  $q_0$  and  $q_1$  are specified, the parameters  $a_0, a_1$  can be computed by solving the system

$$\begin{cases} q(t_0) = q_0 = a_0 \\ q(t_1) = q_1 = a_0 + a_1(t_1 - t_0) \end{cases} \implies \begin{bmatrix} 1 & 0 \\ 1 & T \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \end{bmatrix}$$

where  $T = t_1 - t_0$  is the time duration. Therefore

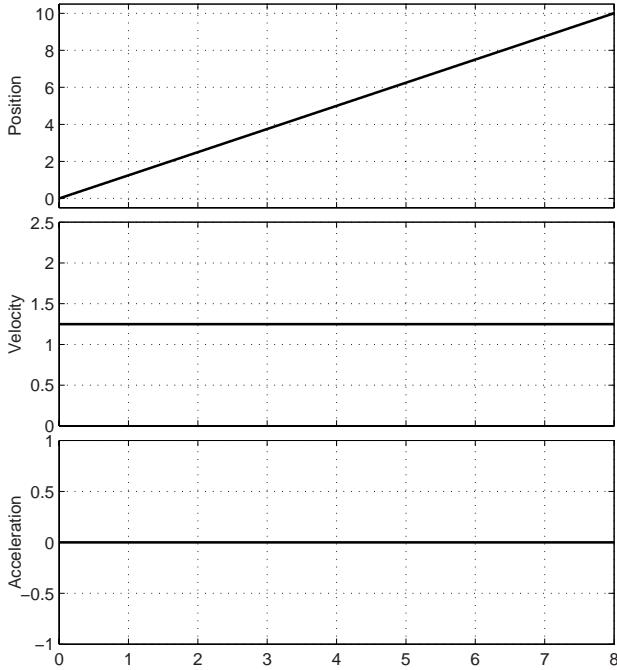
$$\begin{cases} a_0 = q_0 \\ a_1 = \frac{q_1 - q_0}{t_1 - t_0} = \frac{h}{T} \end{cases}$$

where  $h = q_1 - q_0$  is the displacement. The velocity is constant over the interval  $[t_0, t_1]$  and its value is

$$\dot{q}(t) = \frac{h}{T} \quad (= a_1).$$

Obviously, the acceleration is null in the interior of the trajectory and has an impulsive behavior at the extremities.

**Example 2.2** Fig. 2.2 reports the position, velocity and acceleration of the linear trajectory with the conditions  $t_0 = 0$ ,  $t_1 = 8$ ,  $q_0 = 0$ ,  $q_1 = 10$ . Note that at  $t = t_0$ ,  $t_1$ , the velocity is discontinuous and therefore the acceleration is infinite in these points. For this reason the trajectory in this form is not adopted in the industrial practice. □



**Fig. 2.2.** Position, velocity and acceleration of a constant velocity trajectory, with  $t_0 = 0$ ,  $t_1 = 8$ ,  $q_0 = 0$ ,  $q_1 = 10$ .

### 2.1.2 Parabolic trajectory (constant acceleration)

This trajectory, also known as *gravitational trajectory* or *with constant acceleration*, is characterized by an acceleration with a constant absolute value and opposite sign in the acceleration/deceleration periods. Analytically, it is the composition of two second degree polynomials, one from  $t_0$  to  $t_f$  (the flex point) and the second from  $t_f$  to  $t_1$ , see Fig. 2.3.

Let us consider now the case of a trajectory symmetric with respect to its middle point, defined by  $t_f = \frac{t_0+t_1}{2}$  and  $q(t_f) = q_f = \frac{q_0+q_1}{2}$ . Note that in this case  $T_a = (t_f - t_0) = T/2$ ,  $(q_f - q_0) = h/2$ .

In the first phase, the “acceleration” phase, the trajectory is defined by

$$q_a(t) = a_0 + a_1(t - t_0) + a_2(t - t_0)^2, \quad t \in [t_0, t_f].$$

The parameters  $a_0, a_1$  and  $a_2$  can be computed by imposing the conditions of the trajectory through the points  $q_0, q_f$  and the condition on the initial velocity  $v_0$

$$\begin{cases} q_a(t_0) = q_0 = a_0 \\ q_a(t_f) = q_f = a_0 + a_1(t_f - t_0) + a_2(t_f - t_0)^2 \\ \dot{q}_a(t_0) = v_0 = a_1. \end{cases}$$

One obtains

$$a_0 = q_0, \quad a_1 = v_0, \quad a_2 = \frac{2}{T^2}(h - v_0 T).$$

Therefore, for  $t \in [t_0, t_f]$ , the trajectory is defined as

$$\begin{cases} q_a(t) = q_0 + v_0(t - t_0) + \frac{2}{T^2}(h - v_0 T)(t - t_0)^2 \\ \dot{q}_a(t) = v_0 + \frac{4}{T^2}(h - v_0 T)(t - t_0) \\ \ddot{q}_a(t) = \frac{4}{T^2}(h - v_0 T) \end{cases} \text{ (constant).}$$

The velocity at the flex point is

$$v_{max} = \dot{q}_a(t_f) = 2\frac{h}{T} - v_0.$$

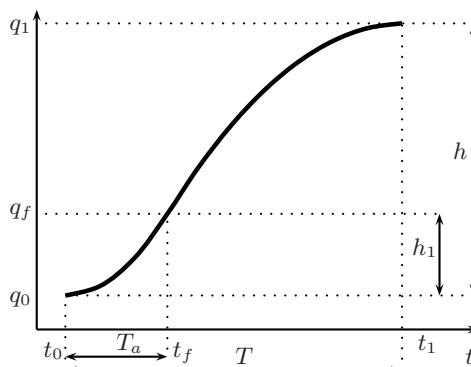
Note that, if  $v_0 = 0$ , the resulting maximum velocity has doubled with respect to the case of the constant velocity trajectory. The jerk is always null except at the flex point, when the acceleration changes its sign and it assumes an infinite value.

In the second part, between the flex and the final point, the trajectory is described by

$$q_b(t) = a_3 + a_4(t - t_f) + a_5(t - t_f)^2 \quad t \in [t_f, t_1].$$

If the final value of the velocity  $v_1$  is assigned, at  $t = t_1$ , the parameters  $a_3, a_4, a_5$  can be computed by means of the following equations

$$\begin{cases} q_b(t_f) = q_f = a_3 \\ q_b(t_1) = q_1 = a_3 + a_4(t_1 - t_f) + a_5(t_1 - t_f)^2 \\ \dot{q}_b(t_1) = v_1 = a_4 + 2a_5(t_1 - t_f) \end{cases}$$



**Fig. 2.3.** Trajectory with constant acceleration.

from which

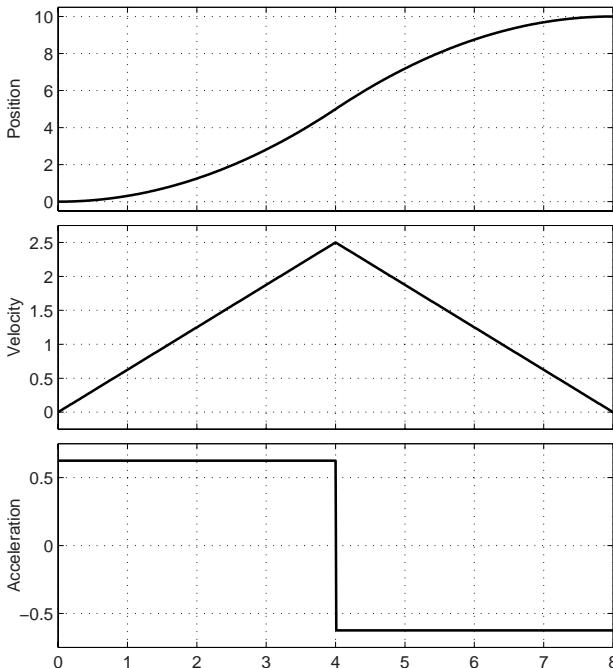
$$a_3 = q_f = \frac{q_0 + q_1}{2}, \quad a_4 = 2\frac{h}{T} - v_1, \quad a_5 = \frac{2}{T^2}(v_1 T - h).$$

The expression of the trajectory for  $t \in [t_f, t_1]$  is

$$\begin{cases} q_b(t) = q_f + \left(2\frac{h}{T} - v_1\right)(t - t_f) + \frac{2}{T^2}(v_1 T - h)(t - t_f)^2 \\ \dot{q}_b(t) = 2\frac{h}{T} - v_1 + \frac{4}{T^2}(v_1 T - h)(t - t_f) \\ \ddot{q}_b(t) = \frac{4}{T^2}(v_1 T - h). \end{cases}$$

Note that, if  $v_0 \neq v_1$ , the velocity profile of this trajectory is discontinuous at  $t = t_f$ .

**Example 2.3** Fig. 2.4 reports the position, velocity and acceleration for this trajectory. The conditions  $t_0 = 0$ ,  $t_1 = 8$ ,  $q_0 = 0$ ,  $q_1 = 10$ ,  $v_0 = v_1 = 0$  have been assigned.  $\square$



**Fig. 2.4.** Position, velocity and acceleration of a trajectory with constant acceleration, with  $t_0 = 0$ ,  $t_1 = 8$ ,  $q_0 = 0$ ,  $q_1 = 10$ .

If the constraint on the position at  $t = t_f$  (i.e.  $q(t_f) = q_f = \frac{q_0+q_1}{2}$ ) is not assigned, the six parameters  $a_i$  may be determined in order to have a continuous velocity profile, i.e.  $\dot{q}_a(t_f) = \dot{q}_b(t_f)$ .

As a matter of fact, by imposing the six conditions

$$\begin{cases} q_a(t_0) = a_0 & = q_0 \\ \dot{q}_a(t_0) = a_1 & = v_0 \\ q_b(t_1) = a_3 + a_4 \frac{T}{2} + a_5 \left(\frac{T}{2}\right)^2 & = q_1 \\ \dot{q}_b(t_1) = a_4 + 2a_5 \frac{T}{2} & = v_1 \\ q_a(t_f) = a_0 + a_1 \frac{T}{2} + a_2 \left(\frac{T}{2}\right)^2 & = a_3 = q_b(t_f) \\ \dot{q}_a(t_f) = a_1 + 2a_2 \frac{T}{2} & = a_4 = \dot{q}_b(t_f) \end{cases}$$

where  $T/2 = (t_f - t_0) = (t_1 - t_f)$ , one obtains

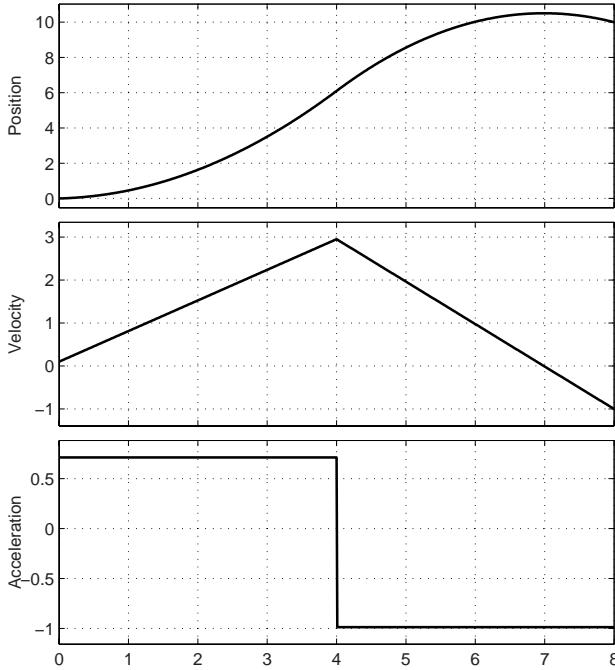
$$\begin{cases} a_0 = q_0 \\ a_1 = v_0 \\ a_2 = \frac{4h - T(3v_0 + v_1)}{2T^2} \\ a_3 = \frac{4(q_0 + q_1) + T(v_0 - v_1)}{8} \\ a_4 = \frac{4h - T(v_0 + v_1)}{2T} \\ a_5 = \frac{-4h + T(v_0 + 3v_1)}{2T^2}. \end{cases}$$

**Example 2.4** Fig. 2.5 reports the position, velocity and acceleration for this trajectory. The conditions  $t_0 = 0$ ,  $t_1 = 8$ ,  $q_0 = 0$ ,  $q_1 = 10$ ,  $v_0 = 0.1$ ,  $v_1 = -1$  have been assigned.  $\square$

### 2.1.3 Trajectory with asymmetric constant acceleration

This trajectory is obtained from the previous one by considering the flex point at a generic instant  $t_0 < t_f < t_1$ , as shown in Fig. 2.3, and not necessarily at  $t = (t_1 + t_0)/2$ . The trajectory is described by the two polynomials

$$\begin{aligned} q_a(t) &= a_0 + a_1(t - t_0) + a_2(t - t_0)^2, & t_0 \leq t < t_f \\ q_b(t) &= a_3 + a_4(t - t_f) + a_5(t - t_f)^2, & t_f \leq t < t_1 \end{aligned}$$



**Fig. 2.5.** Position, velocity and acceleration of a trajectory with constant acceleration and continuous velocity, with  $t_0 = 0$ ,  $t_1 = 8$ ,  $q_0 = 0$ ,  $q_1 = 10$ ,  $v_0 = 0.1$ ,  $v_1 = -1$ .

where the parameters  $a_0, a_1, a_2, a_3, a_4$  and  $a_5$  are obtained by imposing the four conditions on the position and velocity at  $t_0$ ,  $t_1$ , and the two continuity conditions (position and velocity) at  $t_f$ :

$$\begin{cases} q_a(t_0) = a_0 & = q_0 \\ q_b(t_1) = a_3 + a_4(t_1 - t_f) + a_5(t_1 - t_f)^2 = q_1 \\ \dot{q}_a(t_0) = a_1 & = v_0 \\ \dot{q}_b(t_1) = a_4 + 2a_5(t_1 - t_f) & = v_1 \\ q_a(t_f) = a_0 + a_1(t_f - t_0) + a_2(t_f - t_0)^2 = a_3 & (= q_b(t_f)) \\ \dot{q}_a(t_f) = a_1 + 2a_2(t_f - t_0) & = a_4 & (= \dot{q}_b(t_f)). \end{cases}$$

By defining  $T_a = (t_f - t_0)$  and  $T_d = (t_1 - t_f)$ , the resulting parameters are

$$\begin{cases} a_0 = q_0 \\ a_1 = v_0 \\ a_2 = \frac{2h - v_0(T + T_a) - v_1 T_d}{2TT_a} \\ a_3 = \frac{2q_1 T_a + T_d(2q_0 + T_a(v_0 - v_1))}{2T} \\ a_4 = \frac{2h - v_0 T_a - v_1 T_d}{T} \\ a_5 = -\frac{2h - v_0 T_a - v_1(T + T_d)}{2TT_d}. \end{cases}$$

Velocity and acceleration for  $t_0 \leq t < t_f$  are

$$\begin{aligned} \dot{q}_a(t) &= a_1 + 2a_2(t - t_0) = v_0 + \frac{2h - v_0(T + T_a) - v_1 T_d}{TT_a}(t - t_0) \\ \ddot{q}_a(t) &= 2a_2 = \frac{2h - v_0(T + T_a) - v_1 T_d}{TT_a} \end{aligned}$$

while for  $t_f \leq t < t_1$  they result

$$\begin{aligned} \dot{q}_b(t) &= a_4 + 2a_5(t - t_f) = \frac{2h - v_0 T_a - v_1 T_d}{T} - \frac{2h - v_0 T_a - v_1(T + T_d)}{TT_d}(t - t_f) \\ \ddot{q}_b(t) &= 2a_5 = -\frac{2h - v_0 T_a - v_1(T + T_d)}{TT_d}. \end{aligned}$$

Note that, in case  $v_0 = v_1 = 0$ , the value of the maximum velocity is the same as in the previous case (symmetric flex point):

$$v_{max} = \dot{q}_a(t_f) = 2 \frac{h}{T}$$

Obviously, if  $t_f = \frac{t_0+t_1}{2}$  the previous trajectory is obtained.

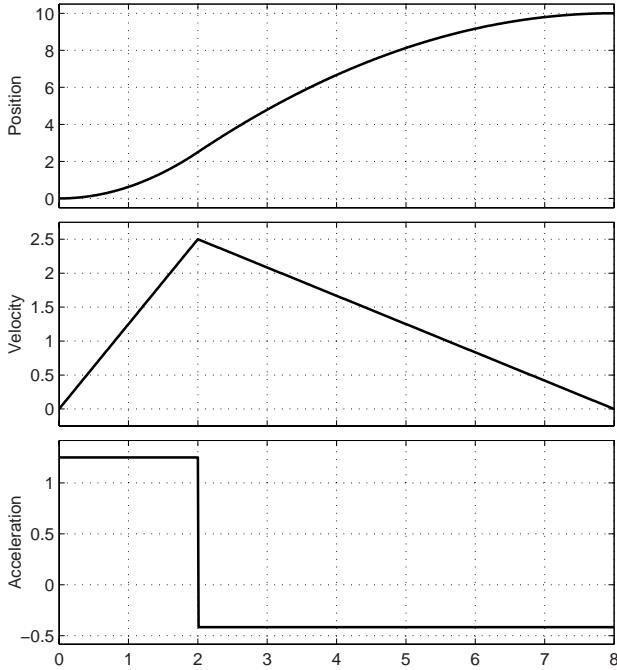
**Example 2.5** Fig. 2.6 shows the position, velocity and acceleration for this trajectory with the same conditions as in the Example 2.3.  $\square$

#### 2.1.4 Cubic trajectory

In case both position and velocity values are specified at  $t_0$  and  $t_1$  ( $q_0, q_1$ , and  $v_0, v_1$  respectively), there are four conditions to be satisfied. Therefore, a third degree polynomial must be used

$$q(t) = a_0 + a_1(t - t_0) + a_2(t - t_0)^2 + a_3(t - t_0)^3, \quad t_0 \leq t \leq t_1 \quad (2.1)$$

and, from the given conditions, the four parameters  $a_0, a_1, a_2, a_3$  are

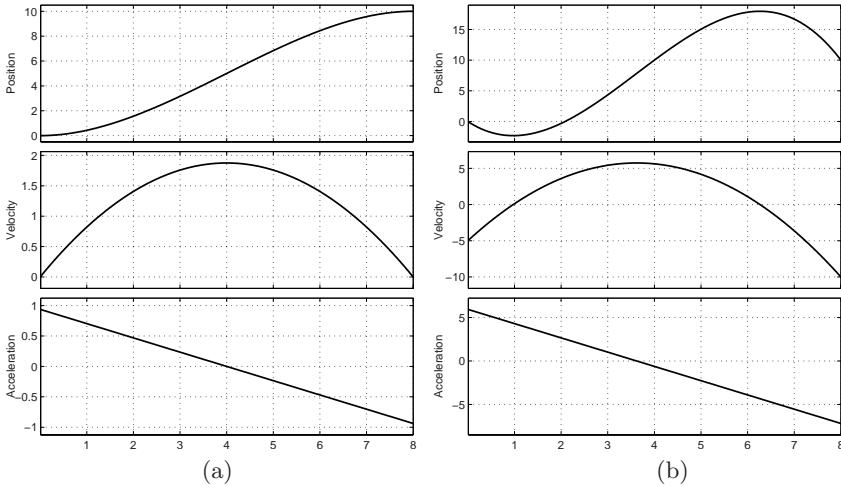


**Fig. 2.6.** Position, velocity and acceleration od a trajectory with asymmetric constant acceleration and  $t_0 = 0, t_1 = 8, t_f = 2, q_0 = 0, q_1 = 10$ .

$$\begin{cases} a_0 = q_0 \\ a_1 = v_0 \\ a_2 = \frac{3h - (2v_0 + v_1)T}{T^2} \\ a_3 = \frac{-2h + (v_0 + v_1)T}{T^3}. \end{cases} \quad (2.2)$$

By exploiting this result, it is very simple to compute a trajectory with continuous velocity through a sequence of  $n$  points. The overall motion is subdivided into  $n-1$  segments. Each of these segments connects the points  $q_k$  and  $q_{k+1}$  at  $t_k, t_{k+1}$  and has initial/final velocity  $v_k, v_{k+1}$  respectively. Then, equations (2.2) are used for each of these segments to define the  $4(n-1)$  parameters  $a_{0k}, a_{1k}, a_{2k}, a_{3k}$ .

**Example 2.6** Fig. 2.7(a) shows position, velocity and acceleration for this trajectory with  $q_0 = 0, q_1 = 10, t_0 = 0, t_1 = 8$  and null initial and final velocities. If these are not null, motion profiles such as those shown in Fig. 2.7(b) are obtained, where the conditions  $v_0 = -5, v_1 = -10$  have been assigned.  $\square$



**Fig. 2.7.** Position, velocity and acceleration of a third degree polynomial trajectory with the conditions  $q_0 = 0$ ,  $q_1 = 10$ ,  $t_0 = 0$ ,  $t_1 = 8$ . In (a) the initial and final velocities are null ( $v_0 = v_1 = 0$ ), while in (b) the values  $v_0 = -5$ ,  $v_1 = -10$  have been assigned.

**Example 2.7** Fig. 2.8 reports the plots of position, velocity and acceleration for a multipoint trajectory with

$$\begin{aligned} t_0 &= 0, & t_1 &= 2, & t_2 &= 4, & t_3 &= 8, & t_4 &= 10, \\ q_0 &= 10, & q_1 &= 20, & q_2 &= 0, & q_3 &= 30, & q_4 &= 40, \\ v_0 &= 0, & v_1 &= -10, & v_2 &= 10, & v_3 &= 3, & v_4 &= 0. \end{aligned}$$

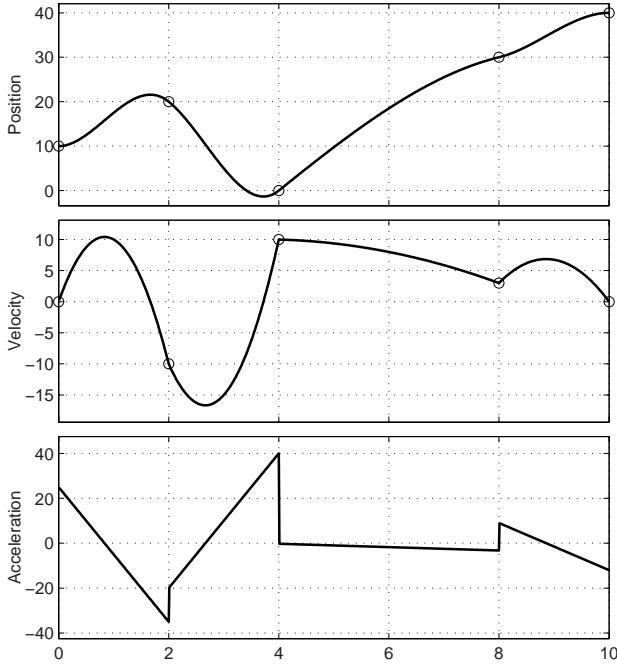
□

In defining a trajectory through a set of points  $q_0, \dots, q_n$ , not always the velocities in the intermediate points are specified. In these cases, suitable values for the intermediate velocities may be determined with heuristic rules such as

$$v_0 \quad \text{(assigned)} \\ v_k = \begin{cases} 0 & \text{sign}(d_k) \neq \text{sign}(d_{k+1}) \\ \frac{1}{2}(d_k + d_{k+1}) & \text{sign}(d_k) = \text{sign}(d_{k+1}) \end{cases} \quad (2.3) \\ v_n \quad \text{(assigned)}$$

where  $d_k = (q_k - q_{k-1})/(t_k - t_{k-1})$  is the slope of the line segment between the instants  $t_{k-1}$  and  $t_k$ , and  $\text{sign}(\cdot)$  is the sign function.

**Example 2.8** The plots obtained with the same sequence of points as in Example 2.7 are reported in Fig. 2.9. In this case, the intermediate velocities are computed with (2.3). □



**Fig. 2.8.** Position, velocity and acceleration for a cubic polynomial through a sequence of points.

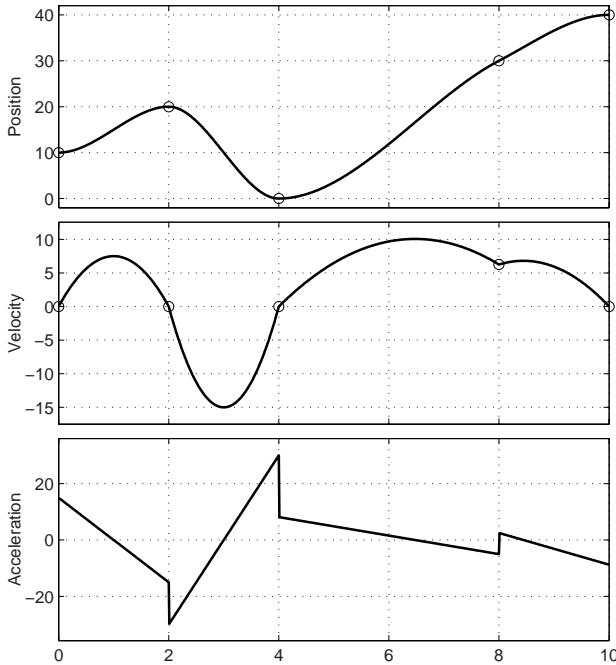
### 2.1.5 Polynomial of degree five

A trajectory through the points  $q_0, \dots, q_n$ , based on third degree polynomials, is characterized by continuous position and velocity profiles, while in general the acceleration is discontinuous, see the examples in Fig. 2.8 and Fig. 2.9.

Although this trajectory is in general “smooth” enough, acceleration discontinuities can generate in some applications undesired effects on the kinematic chains and on the inertial loads. This happens in particular when the minimization of time is of concern, and therefore high acceleration (and velocity) values are assigned, or when relevant mechanical elasticities are present in the actuation system. These aspects are discussed with more details in Chapter 7.

In order to obtain trajectories with continuous acceleration, besides conditions on position and velocity it is also necessary to assign suitable initial and final values for the acceleration. Therefore, since there are six boundary conditions (position, velocity, and acceleration), a fifth degree polynomial must be adopted:

$$q(t) = q_0 + a_1(t - t_0) + a_2(t - t_0)^2 + a_3(t - t_0)^3 + a_4(t - t_0)^4 + a_5(t - t_0)^5 \quad (2.4)$$



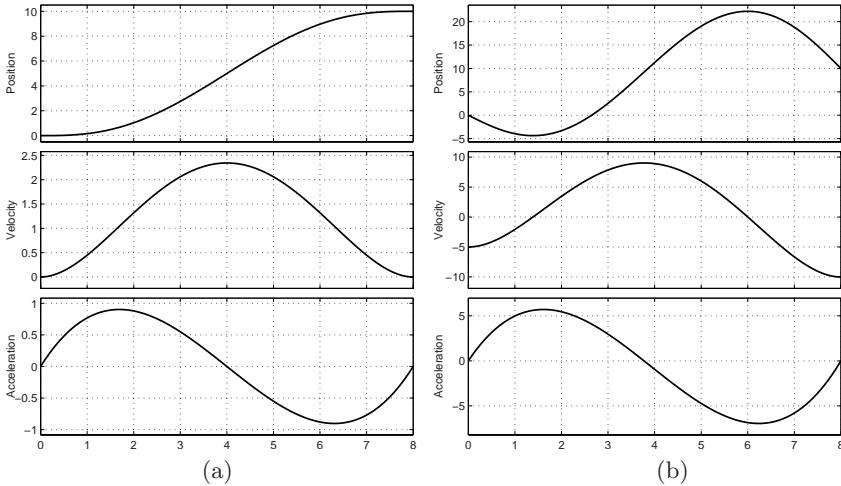
**Fig. 2.9.** Position, velocity and acceleration of a cubic polynomial trajectory through a sequence of points with the intermediate velocities computed according to (2.3).

with the conditions

$$\begin{aligned} q(t_0) &= q_0, & q(t_1) &= q_1 \\ \dot{q}(t_0) &= v_0, & \dot{q}(t_1) &= v_1 \\ \ddot{q}(t_0) &= a_0, & \ddot{q}(t_1) &= a_1. \end{aligned}$$

In this case, by defining  $T = t_1 - t_0$ , the coefficients of the polynomial result

$$\left\{ \begin{array}{l} a_0 = q_0 \\ a_1 = v_0 \\ a_2 = \frac{1}{2}a_0 \\ a_3 = \frac{1}{2T^3}[20h - (8v_1 + 12v_0)T - (3a_0 - a_1)T^2] \\ a_4 = \frac{1}{2T^4}[-30h + (14v_1 + 16v_0)T + (3a_0 - 2a_1)T^2] \\ a_5 = \frac{1}{2T^5}[12h - 6(v_1 + v_0)T + (a_1 - a_0)T^2]. \end{array} \right. \quad (2.5)$$



**Fig. 2.10.** Position, velocity and acceleration of a fifth degree polynomial with  $q_0 = 0$ ,  $q_1 = 10$ ,  $v_0 = v_1 = 0$ ,  $a_0 = a_1 = 0$ ,  $t_0 = 0$ ,  $t_1 = 8$  (a), and  $v_0 = -5$ ,  $v_1 = -10$  (b).

**Example 2.9** A fifth degree trajectory is shown in Fig. 2.10. The initial and final conditions are  $q_0 = 0$ ,  $q_1 = 10$ ,  $v_0 = v_1 = 0$ ,  $a_0 = a_1 = 0$ ,  $t_0 = 0$ ,  $t_1 = 8$  in Fig. 2.10(a), and  $v_0 = -5$ ,  $v_1 = -10$ , in Fig. 2.10(b). Compare these plots with those in Fig. 2.7. Note that, by adopting a cubic polynomial it is not possible to assign boundary values on the acceleration.  $\square$

For a motion through a sequence of points, the considerations illustrated for a third degree polynomial can be applied in the same manner, see eq. (2.3).

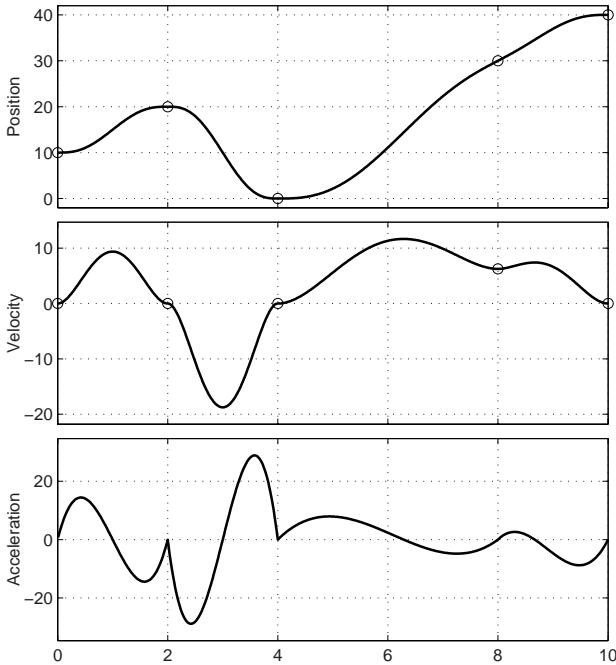
**Example 2.10** Fig. 2.11 reports a fifth degree polynomial, with automatic computation of the intermediate velocities and null intermediate accelerations (compare with Fig. 2.9). Notice the improved “smoothness” in this case.  $\square$

### 2.1.6 Polynomial of degree seven

In particular cases, it might be necessary to define higher degree polynomials in order to obtain smoother profiles. With polynomials of degree seven such as

$$q(t) = a_0 + a_1(t - t_0) + a_2(t - t_0)^2 + a_3(t - t_0)^3 + a_4(t - t_0)^4 + a_5(t - t_0)^5 + a_6(t - t_0)^6 + a_7(t - t_0)^7 \quad (2.6)$$

it is possible to specify eight boundary conditions

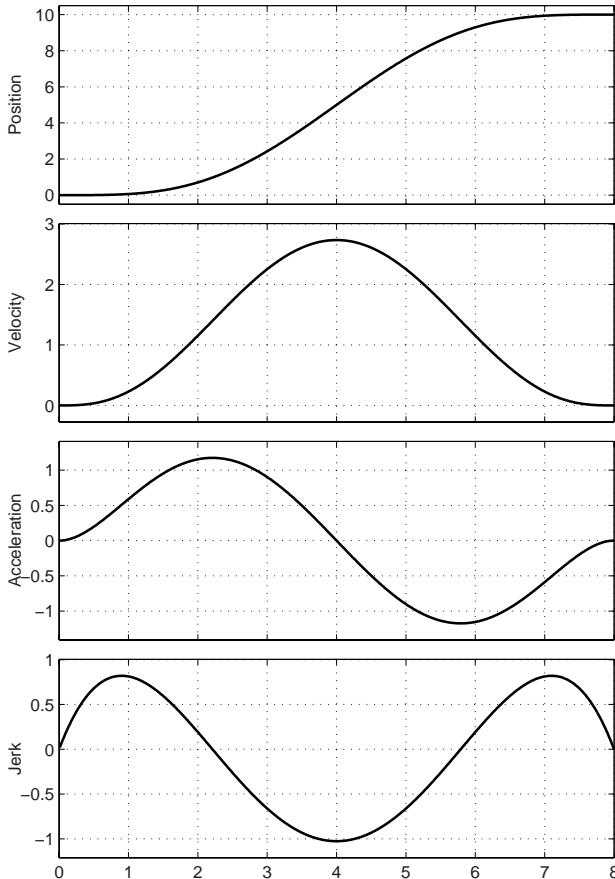


**Fig. 2.11.** Position, velocity and acceleration with a fifth degree polynomial through a sequence of points (compare with Fig. 2.9).

$$\begin{aligned} q(t_0) &= q_0, & \dot{q}(t_0) &= v_0, & \ddot{q}(t_0) &= a_0, & q^{(3)}(t_0) &= j_0, \\ q(t_1) &= q_1, & \dot{q}(t_1) &= v_1, & \ddot{q}(t_1) &= a_1, & q^{(3)}(t_1) &= j_1. \end{aligned}$$

By defining  $T = t_1 - t_0$  and  $h = q_1 - q_0$ , the coefficients  $a_i$ ,  $i = 0, \dots, 7$  are

$$\left\{ \begin{array}{l} a_0 = q_0 \\ a_1 = v_0 \\ a_2 = \frac{a_0}{2} \\ a_3 = \frac{j_0}{6} \\ a_4 = \frac{210h - T[(30a_0 - 15a_1)T + (4j_0 + j_1)T^2 + 120v_0 + 90v_1]}{6T^4} \\ a_5 = \frac{-168h + T[(20a_0 - 14a_1)T + (2j_0 + j_1)T^2 + 90v_0 + 78v_1]}{2T^5} \\ a_6 = \frac{420h - T[(45a_0 - 39a_1)T + (4j_0 + 3j_1)T^2 + 216v_0 + 204v_1]}{6T^6} \\ a_7 = \frac{-120h + T[(12a_0 - 12a_1)T + (j_0 + j_1)T^2 + 60v_0 + 60v_1]}{6T^7}. \end{array} \right.$$



**Fig. 2.12.** Position, velocity, acceleration and jerk of a seventh degree polynomial (compare with Fig. 2.7 and Fig. 2.10).

**Example 2.11** A seventh degree polynomial trajectory is shown in Fig. 2.12, obtained with the boundary conditions  $q_0 = 0$ ,  $q_1 = 10$ ,  $v_0 = v_1 = 0$ ,  $a_0 = a_1 = 0$ ,  $j_0 = 0$ ,  $j_1 = 0$ ,  $t_0 = 0$ ,  $t_1 = 8$ .  $\square$

Obviously, in case of a desired motion through a sequence of points, the considerations illustrated for third and fifth degree polynomials can be applied.

### 2.1.7 Polynomials of higher degree

In particular applications it is necessary to adopt polynomials of high degree in order to impose several constraints, such as boundary conditions on velocity,

acceleration, jerk, snap and even higher order derivatives or conditions in the intermediate points. In these cases, it may be convenient to express the polynomial function of degree  $n$  in normalized form, i.e. as

$$q_N(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + \dots + a_n\tau^n \quad (2.7)$$

with unitary displacement  $h = q_1 - q_0 = 1$  and duration  $T = \tau_1 - \tau_0 = 1$  (for the sake of simplicity it is also assumed  $\tau_0 = 0$ ).

In order to determine the parameters  $a_i$ , it is possible to define an equation of the type

$$\mathbf{M} \mathbf{a} = \mathbf{b} \quad (2.8)$$

where  $\mathbf{a} = [a_0, a_1, a_2, \dots, a_n]^T$ . The vector  $\mathbf{b}$ , containing the boundary conditions on position, velocity, acceleration and so on, is in the form<sup>1</sup>

$$\mathbf{b} = [q_0, v_0, a_0, j_0, \dots, q_1, v_1, a_1, j_1, \dots]^T.$$

Finally, matrix  $\mathbf{M}$  can be easily defined by imposing the boundary conditions on (2.7):

1.  $a_0 = 0$ : polynomial through the first point ( $q_N(0) = 0$ ).
2.  $a_1 = v_0, a_2 = a_0, a_3 = j_0, \dots$ : initial conditions on velocity, acceleration, ...; in general there are  $n_{ci}$  initial conditions on the derivatives of  $q_N(\tau)$ .
3.  $\sum_{i=0}^n a_i = 1$ : polynomial through the last point ( $q_N(1) = 1$ ).
4.  $\sum_{i=1}^n ia_i = v_1$ : final condition on velocity.
5.  $\sum_{i=2}^n i(i-1)a_i = a_1$ : final condition on acceleration.
6.  $\sum_{i=3}^n i(i-1)(i-2)a_i = j_1$ : final condition on jerk.
7.  $\sum_{i=d}^n \frac{i!}{(i-d)!} a_i = c_{d1}$ : final condition on the  $d$ -th derivative of  $q_N(\tau)$  (with  $n_{cf}$  final conditions).

The polynomial  $q_N(\tau)$ , of degree  $n$ , has  $n+1$  coefficients  $a_i$  and therefore matrix  $\mathbf{M}$  has dimensions  $(n+1) \times (n+1)$ , where  $n+1 = n_{ci} + n_{cf} + 2$ . The parameters  $\mathbf{a}$  are determined from  $\mathbf{a} = \mathbf{M}^{-1}\mathbf{b}$ . Note that also for relatively

---

<sup>1</sup> The values of the initial/final velocity, acceleration, ...,  $(v_{Nj}, a_{Nj}, \dots, j = 1, 0)$  are obtained by “normalizing” the corresponding boundary conditions  $v_j, a_j, \dots$  as  $q_{Nj}^{(k)} = \frac{q_j^{(k)}}{h/T^k}$ , being  $q_j^{(k)}$  the given constraint on the derivative of order  $k$  of the desired trajectory  $q(t)$  from  $q_0$  to  $q_1$  ( $h = q_1 - q_0$ ) and of duration  $T$ . For the sake of simplicity, also the normalized boundary conditions  $v_{N0}, v_{N1}, \dots$  are denoted here as  $v_0, v_1, a_0, a_1, \dots$

low values of  $n$  (e.g.  $n = 18, 19, \dots$ ), the computation of  $\mathbf{M}^{-1}$  may give numerical problems due to bad conditioning.

For this reason, if necessary, it is possible to compute the coefficients  $a_i$  with other approaches, more robust from the computational point of view. As a matter of fact, it is possible to exploit the so-called *Bézier/Bernstein form* of polynomials, i.e.

$$q_N(\tau) = \sum_{i=0}^n \binom{n}{i} \tau^i (1-\tau)^{n-i} p_i, \quad 0 \leq \tau \leq 1 \quad (2.9)$$

where  $\binom{n}{i}$  are *binomial coefficients* defined as

$$\binom{n}{i} = \frac{n!}{i!(n-i)!},$$

$\binom{n}{i} \tau^i (1-\tau)^{n-i}$  are the Bernstein basis polynomials, and  $p_i$  are scalar coefficients called *control points*, see also Sec. B.3. Obviously, the expressions (2.7) and (2.9) are equivalent, and it is possible to express a polynomial in both the forms. Accordingly, the relationship between the coefficients  $a_i$  and the parameters  $p_i$  is:

$$a_j = \frac{n!}{(n-j)!} \sum_{i=0}^j \frac{(-1)^{i+j}}{i!(j-i)!} p_i, \quad j = 0, 1, \dots, n, \quad (2.10)$$

see also (B.22). The parameters  $p_i$  in (2.9) can be computed by imposing the boundary conditions on  $q_N(\tau)$ , i.e

$$\begin{aligned} q_N(0) &= 0, & q_N(1) &= 1 \\ \dot{q}_N(0) &= \mathbf{v}_0, & \dot{q}_N(1) &= \mathbf{v}_1 \\ \ddot{q}_N(0) &= \mathbf{a}_0, & \ddot{q}_N(1) &= \mathbf{a}_1 \\ &\vdots & &\vdots \end{aligned} \quad (2.11)$$

An interesting property of the expression (2.9) is that it allows to solve independently the two problems tied to the imposition of boundary conditions at the initial and at the final points (these problems must be solved together if eq. (2.8) is used). As a matter of fact, the derivatives of  $q_N(\tau)$  in (2.9) for  $\tau = 0$  and  $\tau = 1$  are

$$\left\{ \begin{array}{l} \dot{q}_N(0) = n(-p_0 + p_1) \\ \ddot{q}_N(0) = n(n-1)(p_0 - 2p_1 + p_2) \\ \vdots \\ q_N^{(k)}(0) = \frac{n!}{(n-k)!} \sum_{i=0}^k \binom{k}{i} (-1)^{k+i} p_i \end{array} \right. \quad (2.12)$$

and

$$\begin{cases} \dot{q}_N(1) = n(p_n - p_{n-1}) \\ \ddot{q}_N(1) = n(n-1)(p_n - 2p_{n-1} + p_{n-2}) \\ \vdots \\ q_N^{(k)}(1) = \frac{n!}{(n-k)!} \sum_{i=0}^k \binom{k}{i} (-1)^i p_{n-i}. \end{cases} \quad (2.13)$$

As already pointed out, in order to meet all the conditions the degree  $n$  of the polynomial must be at least equal to  $n_{ci} + n_{cf} + 1$ . Note that the problem (2.12) depends only on the value of the first  $n_{ci} + 1$  control points  $p_i$ . Likewise, the problem (2.13) involves only the last  $n_{cf} + 1$  control points.

From (2.12) and the obvious condition  $q_N(0) = q_0$  (in this case  $q_0 = 0$ ) it is possible to define an equation of the type

$$\mathbf{M}_0 \mathbf{p}_0 = \mathbf{b}_0 \quad (2.14)$$

with

$$\mathbf{M}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 3 & -3 & 1 & 0 & 0 & \dots & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & \dots & 0 \\ & & & & \ddots & & & \end{bmatrix}, \quad \mathbf{b}_0 = \begin{bmatrix} 0 \\ \frac{\mathbf{v}_0}{n} \\ \frac{\mathbf{a}_0}{n(n-1)} \\ \frac{\mathbf{j}_0}{n(n-1)(n-2)} \\ \frac{\mathbf{s}_0}{n(n-1)(n-2)(n-3)} \\ \vdots \end{bmatrix}$$

and the vector of the  $n_{ci} + 1$  unknowns  $\mathbf{p}_0 = [p_0, p_1, p_2, \dots, p_{n_{ci}}]^T$ . Note that matrix  $\mathbf{M}_0$  has a triangular structure, and therefore the procedure for its inversion, necessary to find the solution  $\mathbf{p}_0$ , results numerically robust. The last  $n_{cf} + 1$  control points  $\mathbf{p}_1 = [p_n, p_{n-1}, p_{n-2}, \dots, p_{n-n_{cf}}]^T$  are the solution of a system of equations similar to (2.14) (in this case the first equation is  $q_N(1) = q_1 = 1$ ):

$$\mathbf{M}_1 \mathbf{p}_1 = \mathbf{b}_1 \quad (2.15)$$

with

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & -3 & 3 & -1 & 0 & 0 & \dots & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & \dots & 0 \\ & & & & \ddots & & & \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} 1 \\ \frac{\mathbf{v}_1}{n} \\ \frac{\mathbf{a}_1}{n(n-1)} \\ \frac{\mathbf{j}_1}{n(n-1)(n-2)} \\ \frac{\mathbf{s}_1}{n(n-1)(n-2)(n-3)} \\ \vdots \end{bmatrix}.$$

Once all the control points  $\mathbf{p} = [p_0, p_1, \dots, p_{n_{ci}}, p_{n-n_{cf}}, \dots, p_{n-1}, p_n]^T$  in (2.9) are known, it is possible to determine the parameters  $a_i$  in (2.7) according to (2.10).

After the computation of the parameters which define the normalized polynomial  $q_N(\tau)$  either in the form (2.7) or (2.9), the function describing the motion between the two generic points  $(t_0, q_0)$  and  $(t_1, q_1)$  is

$$q(t) = q_0 + q_N(\tau) h, \quad \text{with } \tau = \frac{t - t_0}{T} \quad (2.16)$$

and the velocity, acceleration, ... profiles are

$$\left\{ \begin{array}{l} \dot{q}(t) = \dot{q}_N(\tau) \frac{h}{T} \\ \ddot{q}(t) = \ddot{q}_N(\tau) \frac{h}{T^2} \\ \vdots \\ \frac{d q^d(t)}{d t^d} = \frac{d q_N^d(\tau)}{d \tau^d} \frac{h}{T^d} \end{array} \right. \quad (2.17)$$

see also Sec. 5.2.1.

**Example 2.12** Let us define a polynomial function with the following conditions

$$q_0 = 10, \quad v_0 = 5, \quad a_0 = 0, \quad j_0 = 0, \quad s_0 = 0$$

$$q_1 = 30, \quad v_1 = 0, \quad a_1 = 10, \quad j_1 = 0, \quad s_1 = 0$$

and  $t_0 = 1$ ,  $t_1 = 5$ . In this case, the boundary conditions on the derivatives of the polynomial are 4 at the initial point and 4 at the final point ( $n_{ci} = n_{cf} = 4$ ). Therefore, the degree  $n$  of the polynomial function must be 9. In order to find the coefficients  $p_i$  which define the Bézier/Bernstein polynomial, it is necessary to normalize the constraints. With  $h = q_1 - q_0 = 20$  and  $T = t_1 - t_0 = 4$  the normalized boundary conditions result

$$q_0 = 0, \quad v_0 = 1, \quad a_0 = 0, \quad j_0 = 0, \quad s_0 = 0$$

$$q_1 = 1, \quad v_1 = 0, \quad a_1 = 8, \quad j_1 = 0, \quad s_1 = 0.$$

Therefore, the matrices  $\mathbf{M}_j$  and the vectors  $\mathbf{b}_j$  in (2.14) and (2.15) are respectively

$$\mathbf{M}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ -1 & 3 & -3 & 1 & 0 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}, \quad \mathbf{b}_0 = \begin{bmatrix} 0 \\ \frac{1}{9} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 1 & -3 & 3 & -1 & 0 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} 1 \\ 0 \\ \frac{1}{9} \\ 0 \\ 0 \end{bmatrix}.$$

The control points are

$$\mathbf{p} = \frac{1}{9} [0, 1, 2, 3, 4, 15, 12, 10, 9, 9]^T$$

and the relative normalized trajectory is

$$\begin{aligned} q_N(\tau) = & (1-\tau)^8\tau + 8(1-\tau)^7\tau^2 + 28(1-\tau)^6\tau^3 + 56(1-\tau)^5\tau^4 + \\ & + 210(1-\tau)^4\tau^5 + 112(1-\tau)^3\tau^6 + 40(1-\tau)^2\tau^7 + 9(1-\tau)\tau^8 + \tau^9. \end{aligned}$$

By exploiting (2.10), this trajectory can be rewritten in the standard polynomial form as

$$q_N(\tau) = \tau + 140\tau^5 - 504\tau^6 + 684\tau^7 - 415\tau^8 + 95\tau^9.$$

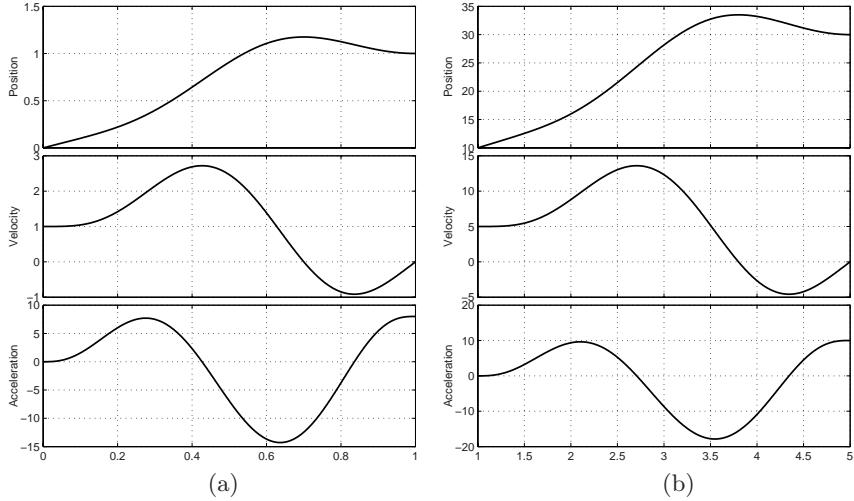
The profiles of position, velocity and acceleration of  $q_N(\tau)$  are shown in Fig. 2.13(a).

Finally, by adopting (2.16) and (2.17), the expression of the desired trajectory with displacement  $h = 20$  and duration  $T = 4$  is obtained. The corresponding profiles of position, velocity and acceleration are shown in Fig. 2.13(b). □

If the standard form (2.7) is assumed, the coefficients of the polynomial  $q(t)$  and of its derivatives can be easily deduced from (2.16) and (2.17) as functions of  $a_i$ ,  $T$ , and  $h$ . As a matter of fact, if we denote with  $b_{i,k}$  the coefficients of  $q^{(k)}(t)$ , i.e.

$$q^{(k)}(t) = \sum_{i=0}^{n-k} b_{i,k} (t - t_0)^i \quad (2.18)$$

the expressions of the position, velocity, acceleration, ... profiles become



**Fig. 2.13.** Normalized polynomial trajectory of degree 9 (a) and corresponding trajectory from  $(t_0, q_0)$  to  $(t_1, q_1)$  (b), Example 2.12.

$$\begin{aligned}
 \text{position: } q(t) &= \sum_{i=0}^n b_{i,0}(t-t_0)^i \rightarrow b_{i,0} = \begin{cases} q_0 + h a_0, & i = 0 \\ \frac{h}{T^i} a_i, & i > 0 \end{cases} \\
 \text{velocity: } \dot{q}(t) &= \sum_{i=0}^{n-1} b_{i,1}(t-t_0)^i \rightarrow b_{i,1} = (i+1) \frac{h}{T^{i+1}} a_{i+1} \\
 \text{acceleration: } \ddot{q}(t) &= \sum_{i=0}^{n-2} b_{i,2}(t-t_0)^i \rightarrow b_{i,2} = (i+1)(i+2) \frac{h}{T^{i+2}} a_{i+2} \\
 &\vdots \\
 \text{d-th derivative: } q^{(d)}(t) &= \sum_{i=0}^{n-d} b_{i,d}(t-t_0)^i \rightarrow b_{i,d} = \frac{(i+d)!}{i!} \frac{h}{T^{i+d}} a_{i+d}.
 \end{aligned} \tag{2.19}$$

Of particular interest is the case of null boundary conditions:

$$\begin{array}{ll}
 v_0 = 0, & v_1 = 0 \\
 a_0 = 0, & a_1 = 0 \\
 j_0 = 0, & j_1 = 0 \\
 \vdots & \vdots
 \end{array}$$

Under this hypothesis the control points, which determine (2.9) and are solution of (2.14) and (2.15), are

$$\mathbf{p} = [\underbrace{0, 0, 0, 0, \dots, 0}_{n_{ci}+1}, \underbrace{1, 1, 1, 1, \dots, 1}_{n_{cf}+1}]^T.$$

The corresponding expression of the coefficients  $a_i$  in (2.7) can be determined from  $\mathbf{p}$  with (2.10). Their values, for polynomials  $q_N(\tau)$  up to degree 21, are reported in Tab. 2.1.

	3	5	7	9	11	13	15	17	19	21
$a_0$	0	0	0	0	0	0	0	0	0	0
$a_1$	0	0	0	0	0	0	0	0	0	0
$a_2$	3	0	0	0	0	0	0	0	0	0
$a_3$	-2	10	0	0	0	0	0	0	0	0
$a_4$	-	-15	35	0	0	0	0	0	0	0
$a_5$	-	6	-84	126	0	0	0	0	0	0
$a_6$	-	-	70	-420	462	0	0	0	0	0
$a_7$	-	-	-20	540	-1980	1716	0	0	0	0
$a_8$	-	-	-	-315	3465	-9009	6435	0	0	0
$a_9$	-	-	-	70	-3080	20020	-40040	24310	0	0
$a_{10}$	-	-	-	-	1386	-24024	108108	-175032	92378	0
$a_{11}$	-	-	-	-	-252	16380	-163800	556920	-755820	352716
$a_{12}$	-	-	-	-	-	-6006	150150	-1021020	2771340	-3233230
$a_{13}$	-	-	-	-	-	924	-83160	1178100	-5969040	13430340
$a_{14}$	-	-	-	-	-	-	25740	-875160	8314020	-33256080
$a_{15}$	-	-	-	-	-	-	-3432	408408	-7759752	54318264
$a_{16}$	-	-	-	-	-	-	-	-109395	4849845	-61108047
$a_{17}$	-	-	-	-	-	-	-	12870	-1956240	47927880
$a_{18}$	-	-	-	-	-	-	-	-	461890	-25865840
$a_{19}$	-	-	-	-	-	-	-	-	-48620	9189180
$a_{20}$	-	-	-	-	-	-	-	-	-	-1939938
$a_{21}$	-	-	-	-	-	-	-	-	-	184756

**Table 2.1.** Per column: coefficients  $a_i$  of the normalized polynomials  $q_N(\tau)$  with degree  $n = 3, 5, \dots, 21$ , with null boundary conditions on their derivatives up to order 10. The degree of the polynomials is  $n = 2n_c + 1$ , being  $n_c$  the number of null initial (and final) conditions.

The polynomial functions obtained in this manner, i.e with null boundary conditions and  $h = 1$ ,  $T = 1$ , have some peculiar properties:

1.  $q_N(\tau) = 1 - q_N(1 - \tau)$ .
2.  $a_0 = a_1 = \dots = a_{n_c} = 0$ .

3.  $a_i \in \mathbb{N}$ .
4.  $\text{sign}(a_{n_{ci}+1}) = 1, \quad \text{sign}(a_{n_{ci}+2}) = -1, \quad \text{sign}(a_{n_{ci}+3}) = 1, \dots$
5.  $\sum_{i=0}^n a_i = 1$ .

From the coefficients of Tab. 2.1 and the above equations (2.19) it is simple to compute the coefficients of the polynomials of the normalized velocity, acceleration,  $\dots$ , profiles (functions  $\dot{q}_N(\tau)$ ,  $\ddot{q}_N(\tau)$ ,  $\dots$ ) or of the polynomials  $q(t)$ ,  $\dot{q}(t)$ ,  $\ddot{q}(t)$ ,  $\dots$  for a generic displacement. The coefficients of  $\dot{q}_N(\tau)$  and  $\ddot{q}_N(\tau)$  are reported in Appendix A.1.

The position, velocity, acceleration and jerk profiles for these polynomials are shown in Fig. 2.14. Note the increasing smoothness of the profiles, and the corresponding higher values for the maximum velocity, acceleration and jerk, whose numerical values are reported in Tab. 2.2, denoted with  $C_v$ ,  $C_a$ , and  $C_j$  respectively.

**Example 2.13** Let us define a polynomial function with the following conditions

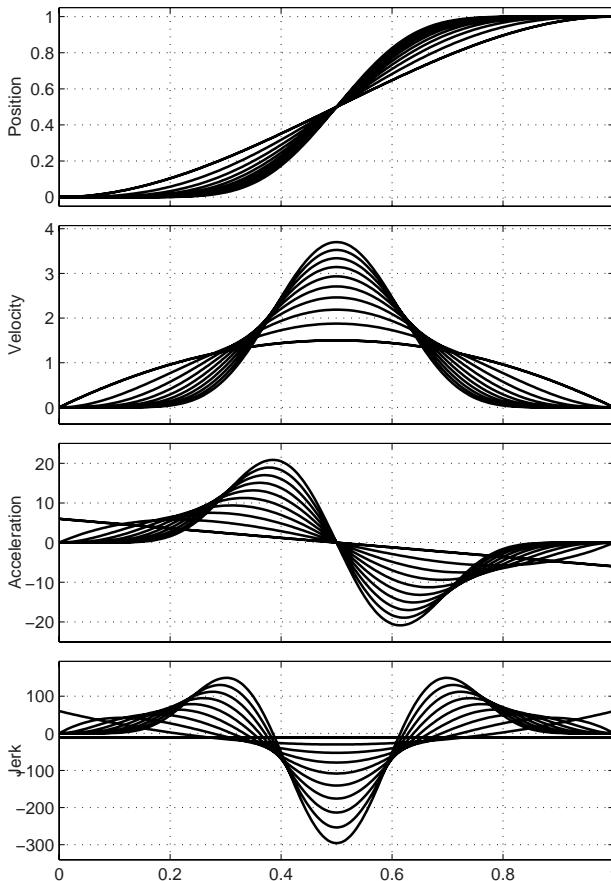
$$\begin{aligned} q_0 &= 10, & v_0 &= 0, & a_0 &= 0, & j_0 &= 0, & s_0 &= 0 \\ q_1 &= 30, & v_1 &= 0, & a_1 &= 0, & j_1 &= 0, & s_1 &= 0 \end{aligned}$$

and  $t_0 = 1$ ,  $t_1 = 5$ . There are 10 conditions to be satisfied, and therefore the polynomial must be at least of degree 9. The expression of the normalized polynomial  $q_N(\tau)$  in the Bézier/Bernstein form (2.9) with null boundary conditions is:

$$q_N(\tau) = 126(1-\tau)^4\tau^5 + 84(1-\tau)^3\tau^6 + 36(1-\tau)^2\tau^7 + 9(1-\tau)\tau^8 + \tau^9.$$

$n$	$C_v$	$\Delta\%$	$C_a$	$\Delta\%$	$C_j$	$\Delta\%$
3	1.5	0	6	0	12	0
5	1.875	25	5.7735	-3.78	60	400
7	2.1875	45.83	7.5132	25.22	52.5	337.5
9	2.4609	64.06	9.372	56.2	78.75	556.25
11	2.707	80.47	11.2666	87.78	108.2813	802.34
13	2.9326	95.51	13.1767	119.61	140.7656	1073.05
15	3.1421	109.47	15.0949	151.58	175.957	1366.31
17	3.3385	122.56	17.018	183.63	213.6621	1680.52
19	3.5239	134.93	18.9441	215.73	253.7238	2014.36
21	3.7001	146.68	20.8723	247.87	296.011	2366.76

**Table 2.2.** Maximum values of velocity ( $C_v$ ), acceleration ( $C_a$ ) and jerk ( $C_j$ ) for normalized polynomials of degree 3 - 21: smoother (higher degree) polynomials present higher velocity and acceleration values. The variations with respect to the 3-rd degree polynomial are also reported.



**Fig. 2.14.** Position, velocity, acceleration and jerk profiles for normalized polynomial functions of degree 3 - 21 with null boundary conditions.

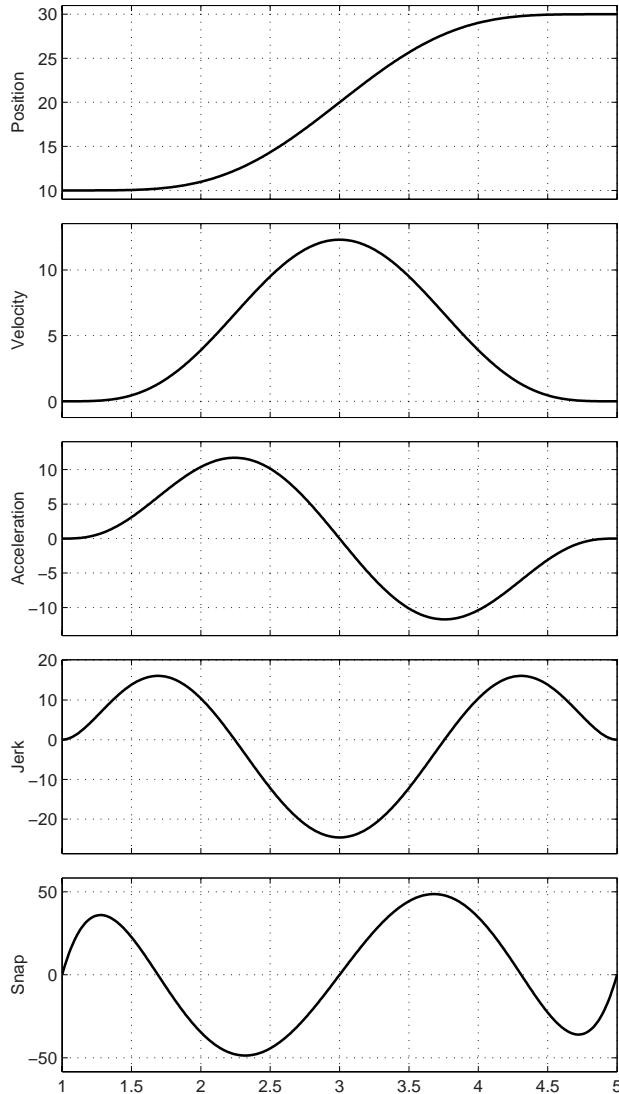
From Tab. 2.1, the coefficients  $\mathbf{a} = [a_0, a_1, \dots, a_9]^T$  of the standard polynomial form are:

$$\mathbf{a} = [0, 0, 0, 0, 0, 126, -420, 540, -315, 70]^T.$$

By using (2.16), the desired trajectory with displacement  $h = 20$  and duration  $T = 4$  is computed as

$$q(t) = 10 + 20(126\tau^5 - 420\tau^6 + 540\tau^7 - 315\tau^8 + 70\tau^9), \quad \text{with } \tau = \left(\frac{t-1}{4}\right).$$

Alternatively, from (2.19), one can directly write the expression of  $q(t)$  and of its derivatives:



**Fig. 2.15.** Polynomial function of degree 9 of Example 2.13.

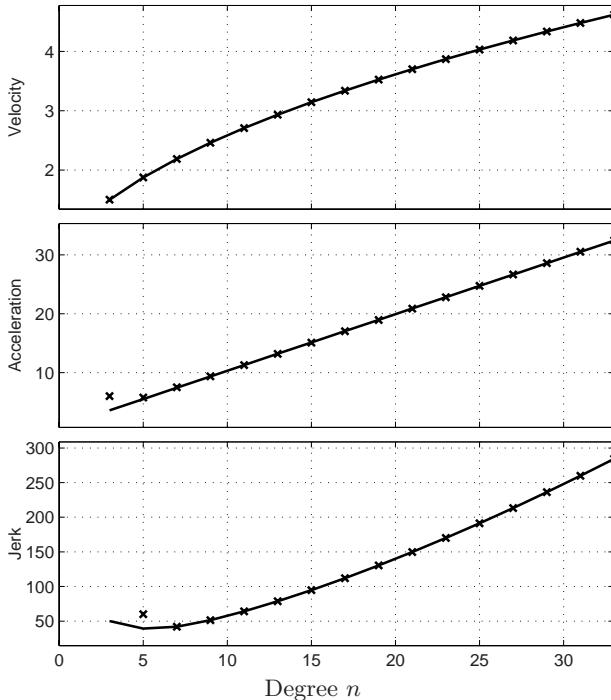
$$\begin{aligned}
 q(t) &= 10 + 20 \frac{126}{45} (t-1)^5 + 20 \frac{-420}{46} (t-1)^6 + 20 \frac{540}{47} (t-1)^7 + \\
 &\quad + 20 \frac{-315}{48} (t-1)^8 + 20 \frac{70}{49} (t-1)^9 \\
 &= 10 + 2.4609(t-1)^5 - 2.0508(t-1)^6 + 0.6592(t-1)^7 + \\
 &\quad - 0.0961(t-1)^8 + 0.0053(t-1)^9
 \end{aligned}$$

$$\begin{aligned}
\dot{q}(t) &= 5 \cdot 20 \frac{126}{4^5} (t-1)^4 + 6 \cdot 20 \frac{-420}{4^6} (t-1)^5 + 7 \cdot 20 \frac{540}{4^7} (t-1)^6 + \\
&\quad + 8 \cdot 20 \frac{-315}{4^8} (t-1)^7 + 9 \cdot 20 \frac{70}{4^9} (t-1)^8 \\
&= 12.3047(t-1)^4 - 12.3047(t-1)^5 + 4.6143(t-1)^6 + \\
&\quad - 0.7690(t-1)^7 + 0.0481(t-1)^8 \\
\ddot{q}(t) &= 5 \cdot 4 \cdot 20 \frac{126}{4^5} (t-1)^3 + 6 \cdot 5 \cdot 20 \frac{-420}{4^6} (t-1)^4 + 7 \cdot 6 \cdot 20 \frac{540}{4^7} (t-1)^5 + \\
&\quad + 8 \cdot 7 \cdot 20 \frac{-315}{4^8} (t-1)^6 + 9 \cdot 8 \cdot 20 \frac{70}{4^9} (t-1)^7 \\
&= 49.2188(t-1)^3 - 61.5234(t-1)^4 + 27.6855(t-1)^5 + \\
&\quad - 5.3833(t-1)^6 + 0.3845(t-1)^7.
\end{aligned}$$

These functions are shown in Fig. 2.15.  $\square$

The maximum value of the velocity, acceleration, jerk, . . . , of a (normalized) polynomial  $q_N(\tau)$  increases with the degree  $n$ , as illustrated in Fig. 2.14 and reported in Tab. 2.2. It is interesting to note, as illustrated in Fig. 2.16, that the rates of growth of  $C_v$ ,  $C_a$  and  $C_j$  are proportional to  $\sqrt{n}$ ,  $n$ , and  $n^2$  respectively.

Although the determination of polynomials in the Bézier/Bernstein form is quite robust from the numerical point of view, for large values of  $n$  (eg. 37, 39, . . . ) the computation of polynomials is in any case affected by relevant numerical errors, and therefore it is advisable to use other functions to define smooth motion profiles, like trigonometric or exponential functions.



**Fig. 2.16.** Maximum values of the velocity, acceleration and jerk profiles of normalized polynomials of degree 3 - 33 with null boundary conditions, plotted as function of  $n$  (x-marks); interpolation with functions depending respectively on  $\sqrt{n}$ ,  $n$ ,  $n^2$  (solid lines).

## 2.2 Trigonometric Trajectories

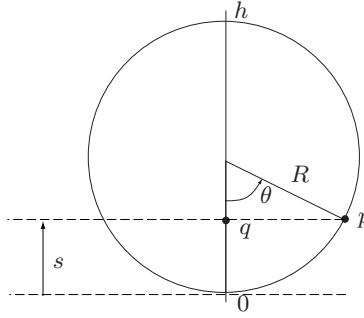
In this section, the analytical expressions of trajectories based on trigonometric functions are described. These trajectories present non-null continuous derivatives for any order of derivation in the interval  $(t_0, t_1)$ . However, these derivatives may be discontinuous in  $t_0$  and  $t_1$ .

### 2.2.1 Harmonic trajectory

An harmonic motion is characterized by an acceleration profile that is proportional to the position profile, with opposite sign. The mathematical formulation of the harmonic motion can be also deduced graphically, see Fig. 2.17.

Let the point  $q$  be the projection on the diameter of point  $p$ . If point  $p$  moves on the circle with constant velocity, the motion of  $q$ , called *harmonic*, is described by

$$s(\theta) = R(1 - \cos \theta) \quad (2.20)$$



**Fig. 2.17.** Geometric construction of the harmonic motion.

where  $R$  is the radius of the circle. In a more general form, the harmonic trajectory can be defined as

$$q(t) = \frac{h}{2} \left( 1 - \cos \frac{\pi(t - t_0)}{T} \right) + q_0 \quad (2.21)$$

with  $h = q_1 - q_0$  and  $T = t_1 - t_0$ , from which

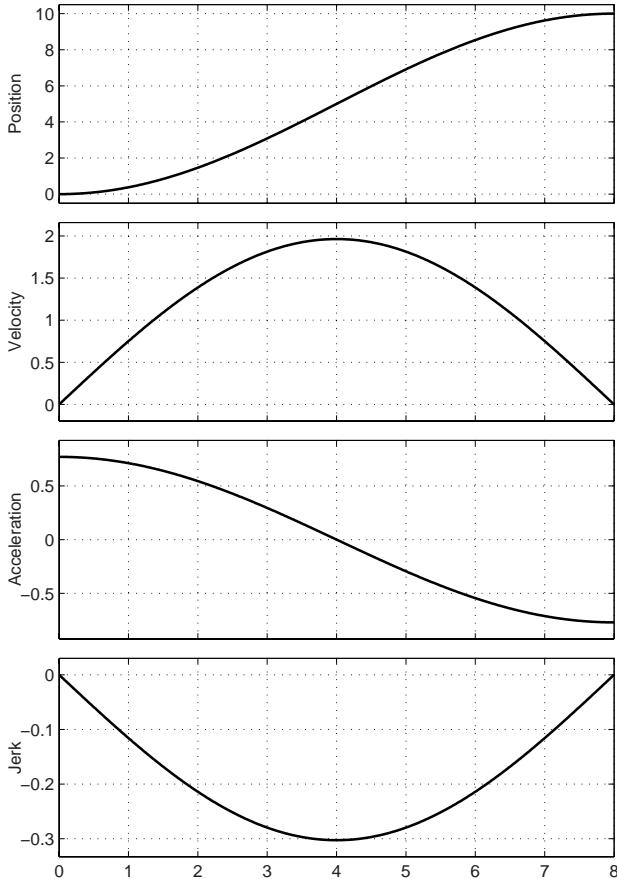
$$\begin{cases} \dot{q}(t) &= \frac{\pi h}{2T} \sin \left( \frac{\pi(t - t_0)}{T} \right) \\ \ddot{q}(t) &= \frac{\pi^2 h}{2T^2} \cos \left( \frac{\pi(t - t_0)}{T} \right) \\ q^{(3)}(t) &= -\frac{\pi^3 h}{2T^3} \sin \left( \frac{\pi(t - t_0)}{T} \right). \end{cases}$$

**Example 2.14** Fig. 2.18 reports the position, velocity, acceleration and jerk of an harmonic trajectory with the conditions  $t_0 = 0$ ,  $t_1 = 8$ ,  $q_0 = 0$ ,  $q_1 = 10$ .  $\square$

## 2.2.2 Cycloidal trajectory

As shown in Fig. 2.18, the harmonic trajectory presents a discontinuous acceleration and, therefore, infinite instantaneous jerk at  $t_0$ ,  $t_1$ . As already discussed, a discontinuous acceleration profile may generate undesired effects when flexible mechanisms are present. A continuous acceleration profile is obtained with the cycloidal trajectory, described by a circle with circumference  $h$  rolling along a line see Fig. 2.19,

$$\begin{aligned} q(t) &= (q_1 - q_0) \left( \frac{t - t_0}{t_1 - t_0} - \frac{1}{2\pi} \sin \frac{2\pi(t - t_0)}{t_1 - t_0} \right) + q_0 \\ &= h \left( \frac{t - t_0}{T} - \frac{1}{2\pi} \sin \frac{2\pi(t - t_0)}{T} \right) + q_0 \end{aligned} \quad (2.22)$$



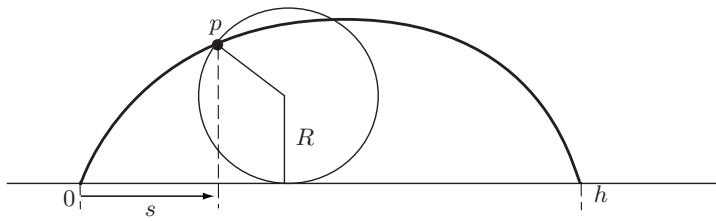
**Fig. 2.18.** Position, velocity, acceleration and jerk of an harmonic trajectory when  $t_0 = 0$ ,  $t_1 = 8$ ,  $q_0 = 0$ ,  $q_1 = 10$ .

from which

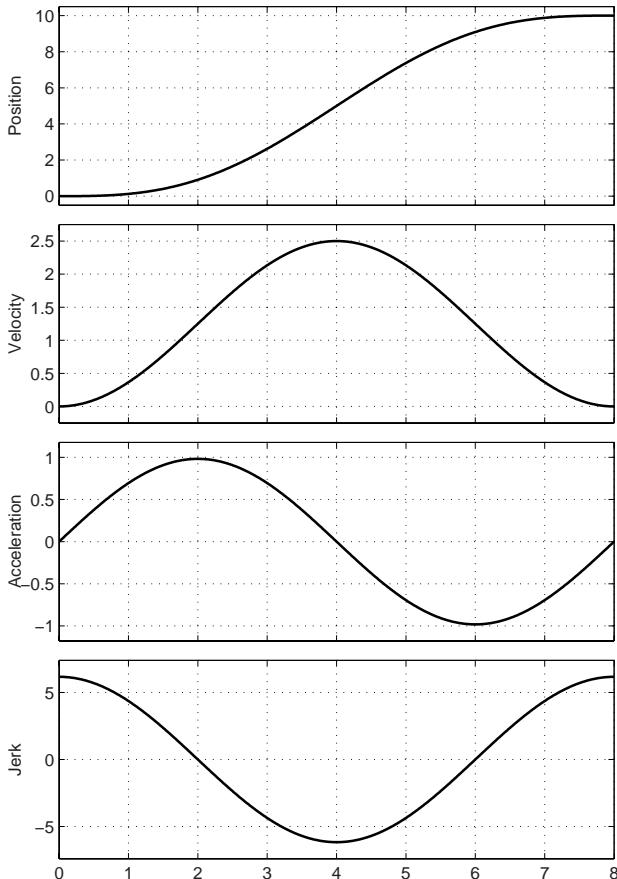
$$\begin{aligned}\dot{q}(t) &= \frac{h}{T} \left( 1 - \cos \frac{2\pi(t-t_0)}{T} \right) \\ \ddot{q}(t) &= \frac{2\pi h}{T^2} \sin \frac{2\pi(t-t_0)}{T} \\ q^{(3)}(t) &= \frac{4\pi^2 h}{T^3} \cos \frac{2\pi(t-t_0)}{T}.\end{aligned}$$

In this case, the acceleration is null in  $t = t_0, t_1$ , and therefore it presents a continuous profile.

**Example 2.15** Fig. 2.20 shows position, velocity, acceleration and jerk for a cycloidal trajectory with the same conditions as in the previous example.  $\square$



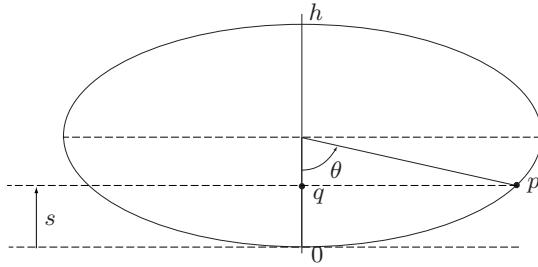
**Fig. 2.19.** Geometric construction of the cycloidal motion.



**Fig. 2.20.** Position, velocity, acceleration and jerk of a cycloidal trajectory with  $t_0 = 0$ ,  $t_1 = 8$ ,  $q_0 = 0$ ,  $q_1 = 10$ .

### 2.2.3 Elliptic trajectory

As shown in Fig. 2.17, the harmonic motion can be obtained graphically by projecting on the diameter a point moving on a circle. An elliptic motion is



**Fig. 2.21.** Geometric construction of the elliptic motion.

obtained by projecting the motion of a point moving on an ellipse on the minor axis, of length equal to the desired displacement  $h = q_1 - q_0$ , see Fig. 2.21. The resulting equation is

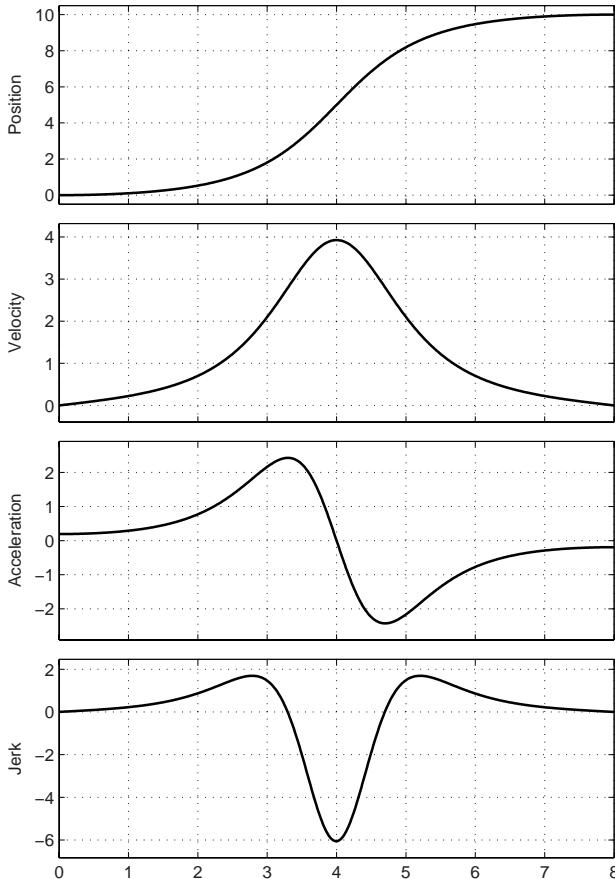
$$q(t) = \frac{h}{2} \left( 1 - \frac{\cos \frac{\pi(t-t_0)}{T}}{\sqrt{1 - \alpha \sin^2 \frac{\pi(t-t_0)}{T}}} \right) + q_0 \quad (2.23)$$

where  $\alpha = \frac{n^2-1}{n^2}$ , and  $n$  is the ratio between the major and minor ellipse axes. The velocity and the acceleration are

$$\begin{aligned} \dot{q}(t) &= \frac{\pi h}{2T} \frac{\sin \frac{\pi(t-t_0)}{T}}{n^2 \sqrt{\left(1 - \alpha \sin^2 \frac{\pi(t-t_0)}{T}\right)^3}} \\ \ddot{q}(t) &= \frac{\pi^2 h}{2T^2} \cos\left(\frac{\pi(t-t_0)}{T}\right) \frac{1 + 2\alpha \sin^2 \frac{\pi(t-t_0)}{T}}{n^2 \sqrt{\left(1 - \alpha \sin^2 \frac{\pi(t-t_0)}{T}\right)^5}}. \end{aligned}$$

Obviously, the harmonic trajectory is obtained by setting  $n = 1$ .

**Example 2.16** Fig. 2.22 shows position, velocity, acceleration and jerk of this trajectory. Fig. 2.23 reports the profiles of position, velocity and acceleration with different choices of  $n$ . Note that the maximum values of velocity and acceleration increase with  $n$ .  $\square$



**Fig. 2.22.** Position, velocity, acceleration and jerk of an elliptic trajectory with  $t_0 = 0$ ,  $t_1 = 8$ ,  $q_0 = 0$ ,  $q_1 = 10$ ,  $n = 2$ .

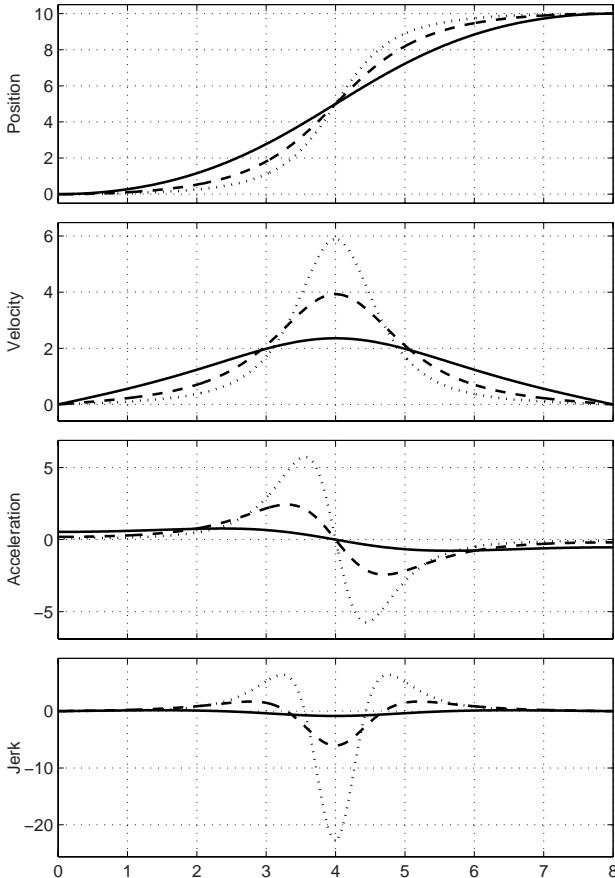
## 2.3 Exponential Trajectories

As discussed in Chapter 7, natural vibrations induced on the machine by the actuation system should always be minimized.

This involves also the choice of proper motion profiles, since discontinuities in the desired trajectory may generate vibrations in the machine due to the induced discontinuities in the applied forces and the elastic effects of the mechanical system itself. Therefore, it may be convenient to introduce trajectories whose smoothness can be adjusted according to the needs, [14].

For this purpose, it is possible to consider an exponential function for the velocity, as

$$\dot{q}(\tau) = v_c e^{-\sigma f(\tau, \lambda)}$$



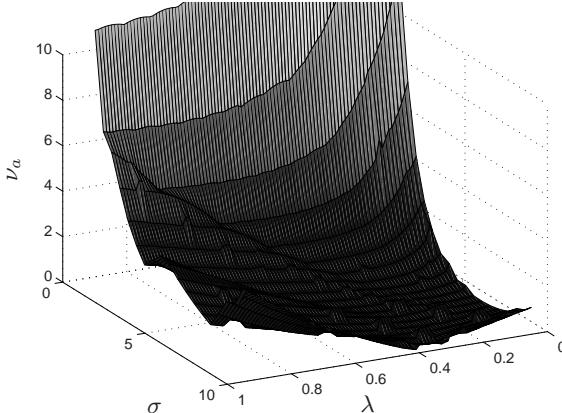
**Fig. 2.23.** Elliptic trajectories when:  $n = 1.2$  (solid),  $n = 2$  (dashed),  $n = 3$  (dotted).

where  $\sigma$  and  $\lambda$  are free parameters. Possible choices for the function  $f(\tau, \lambda)$  are

$$f_a(\tau, \lambda) = \frac{(2\tau)^2}{|1 - (2\tau)^2|^\lambda} \quad \text{or} \quad f_b(\tau, \lambda) = \frac{\sin^2 \pi \tau}{|\cos \pi \tau|^\lambda} .$$

If a normalized motion profile is considered, i.e. with unit displacement and duration, and in particular with the conditions  $q_0 = -0.5$ ,  $q_1 = 0.5$ , and  $\tau_0 = -0.5$ ,  $\tau_1 = 0.5$ , then the constant  $v_c$  can be computed as

$$v_c = \frac{1}{2 \int_0^{\frac{1}{2}} -\sigma f(\tau, \lambda) d\tau} .$$



**Fig. 2.24.** Maximum values of the residual spectrum  $\nu_a$  of the exponential trajectory for different values of  $\sigma$  and  $\lambda$ .

At this point, the normalized motion  $q_N(\tau)$  is defined by the following equations

$$\begin{cases} q_N(\tau) = v_c \int_0^\tau e^{-\sigma f(\tau, \lambda)} d\tau \\ \dot{q}_N(\tau) = v_c e^{-\sigma f(\tau, \lambda)} \\ \ddot{q}_N(\tau) = -v_c \sigma \frac{f(\tau, \lambda)}{d\tau} e^{-\sigma f(\tau, \lambda)}. \end{cases} \quad (2.24)$$

The choice of the function  $f_a(\tau, \lambda)$  or  $f_b(\tau, \lambda)$  has only a little influence on the actual motion profile and therefore, being  $f_a$  simpler from a computational point of view, it is adopted in the following discussion. More important is the choice of  $\sigma$  and  $\lambda$ , whose values may be assigned in order to minimize the maximum amplitude of the high frequency components of the acceleration profile, responsible of vibrations induced in the machine. The maximum values of the residual spectrum  $\nu_a$ <sup>2</sup> of  $\ddot{q}_N$  for frequencies greater than 5 Hz, obtained for several values of the parameters  $\sigma, \lambda$ , are shown in Fig. 2.24.

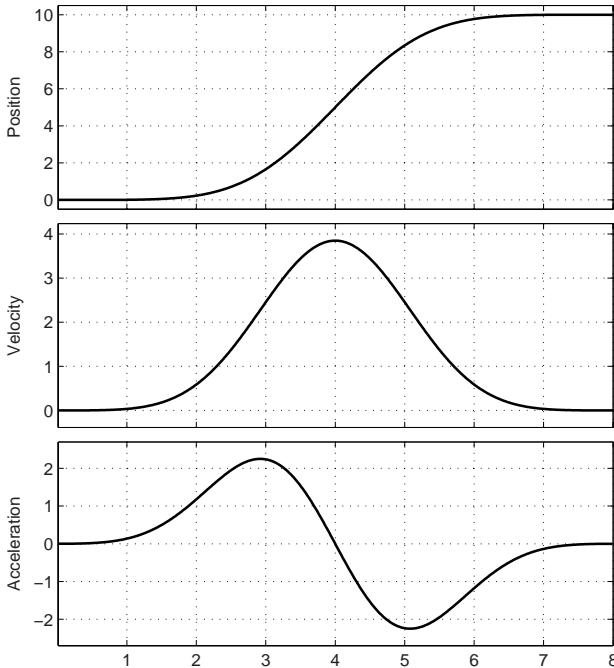
In particular, the numerical values of  $\nu_a$  obtained for some values of  $\sigma$  and with the corresponding  $\lambda$  which minimizes the residual spectrum are reported in Tab. 2.3. It is possible to show that the minimum value  $\nu_{a,min} = 0.018$  is obtained for  $\lambda = 0.20$ ,  $\sigma = 7.1$ , [14].

In case of a trajectory from an initial point  $q_0$  to a final one  $q_1$ , with  $h = q_1 - q_0$ , and time instants  $t_0$  and  $t_1$ , with  $T = t_1 - t_0$ , the actual position  $q(t)$ , velocity  $\dot{q}(t)$  and acceleration  $\ddot{q}(t)$  profiles may be obtained from (2.24)

<sup>2</sup> The residual spectrum is defined here as the maximum amplitude of the frequency spectrum of the acceleration profile for frequencies higher than a given threshold.

$\sigma$	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5
$\lambda$	0.61	0.49	0.41	0.34	0.29	0.25	0.22	0.19	0.18	0.18	0.19	0.28
$\nu_a$	4.364	2.736	1.697	1.034	0.625	0.370	0.217	0.125	0.071	0.039	0.019	0.043

**Table 2.3.** Parameters  $\sigma$  and  $\lambda$  for exponential trajectories and the related maximum amplitude of the frequency content of the acceleration profiles ( $> 5 \text{ Hz}$ ).



**Fig. 2.25.** Position, velocity, acceleration profiles of an exponential trajectory with  $\sigma = 7.1$  and  $\lambda = 0.2$ .

as

$$q(t) = q_0 + h \left( \frac{1}{2} + q_N(\tau) \right), \quad \dot{q}(t) = \frac{h}{T} \dot{q}_N(\tau), \quad \ddot{q}(t) = \frac{h}{T^2} \ddot{q}_N(\tau)$$

with  $\tau = \left( \frac{t - t_0}{T} - 0.5 \right)$ , see also Chapter 5.

**Example 2.17** An exponential trajectory with the conditions

$$q_0 = 0, \quad q_1 = 10, \quad t_0 = 0, \quad t_1 = 8, \quad \lambda = 0.20, \quad \sigma = 7.1$$

is shown in Fig. 2.25. □

**Example 2.18** The exponential trajectories obtained with the conditions

$$q_0 = 0, \quad q_1 = 10, \quad t_0 = 0, \quad t_1 = 8$$

and the parameters  $\sigma, \lambda$  as in Tab. 2.3 are shown in Fig. 2.26.  $\square$

A final comment concerns the computation of eq. (2.24), where an integral function explicitly appears. If the computation of  $q_N(\tau)$  by using integrals, with possibly variable upper bounds, is unsuitable for the online generation of the motion profile, it is possible to adopt a series expansion of the function  $q_N(\tau)$ , truncated at a proper order  $r$ , as

$$\begin{cases} q_N(\tau) = a_0 \tau + \sum_{k=1}^r a_{2k} \sin(2k\pi\tau) \\ \dot{q}_N(\tau) = a_0 + 2\pi \sum_{k=1}^r k a_{2k} \cos(2k\pi\tau) \\ \ddot{q}_N(\tau) = -4\pi^2 \sum_{k=1}^r k^2 a_{2k} \sin(2k\pi\tau) \end{cases}$$

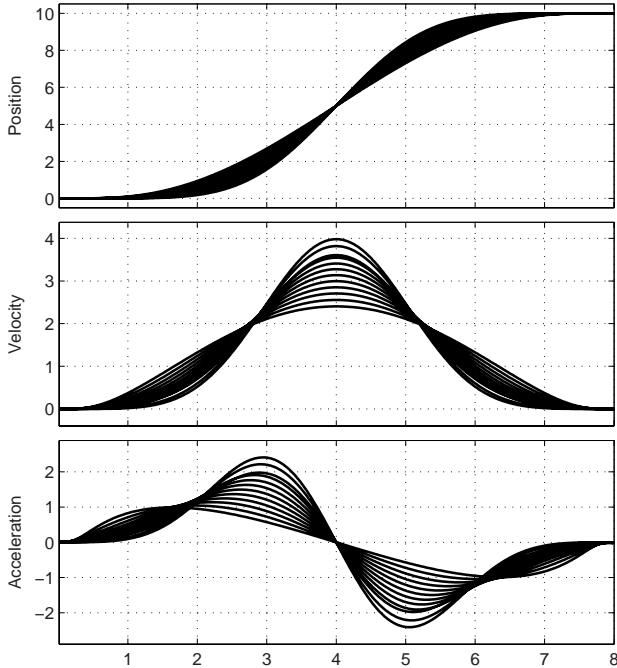
where

$$a_0 = 1, \quad a_{2k} = \frac{2}{\pi k} \int_0^{\frac{1}{2}} \dot{q}(\tau) \cos(2k\pi\tau) d\tau.$$

## 2.4 Trajectories Based on the Fourier Series Expansion

Besides quite obvious conditions about continuity of the position profile and its derivatives up to a given order, and the given boundary constraints, it might be of interest to pursue also other goals. Among the different possibilities, it could be desirable to minimize the amplitude of the acceleration profile, in order to avoid efforts on the load due to inertial forces or vibrational effects of the mechanical structure.

The minimization of the amplitude of the acceleration in general is in contrast with the continuity of the profile: a discontinuous acceleration profile minimizes the peak of acceleration but, on the other hand, may generate oscillations and/or vibrations because of the related discontinuity of the inertial forces. For example, the trapezoidal velocity trajectory (discussed in the following Chapter 3) presents, other conditions being equal, smaller values for the acceleration but, at the same time, an higher harmonic content that usually implies possible vibrations in the mechanical structure. On the contrary, the cycloidal trajectory is characterized by a low harmonic content but presents higher acceleration values. It is possible to define trajectories that represent a compromise between these two opposite features. As an example, trajectories



**Fig. 2.26.** Position, velocity, acceleration profiles of exponential trajectories with  $\sigma$  and  $\lambda$  as in Tab. 2.3.

derived from a Fourier series expansion of the motion profiles illustrated in the previous sections are now considered.

It is well known that a fundamental tool for the analysis in the frequency domain  $\omega$  of a signal  $x(t)$  defined in the time domain is the *Fourier Transform*  $X(\omega) = \mathcal{F}\{x(t)\}$ , see Appendix D. On the other hand, it is worth noticing that trajectories for high speed automatic machines are often a cyclic repetition of a basic motion: therefore, the trajectory  $q(t)$  can be assumed to be periodic. Under this hypothesis,  $q(t)$  can be analyzed by means of the *Fourier series expansion*.

The Fourier series is a mathematical tool often used for analyzing periodic functions by decomposing them into a weighted sum of sinusoidal component functions, sometimes referred to as *normal Fourier modes*, or simply *modes*. Given a piecewise continuous function  $x(t)$ , periodic with period  $T$ , and square-integrable over the interval  $[-T/2, T/2]$ , that is

$$\int_{-T/2}^{T/2} |x(t)|^2 dt < +\infty,$$

the corresponding Fourier series expansion is

$$x(t) = \frac{1}{2}a_0 + \sum_{k=1}^{\infty} [a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t)]$$

where  $\omega_0 = 2\pi/T$  is the fundamental frequency (*rad/sec*) of the function and, for any non-negative integer  $k$ ,

$$\begin{aligned} a_k &= \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cos(k\omega_0 t) dt && \text{are the even Fourier coefficients of } x(t) \\ b_k &= \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin(k\omega_0 t) dt && \text{are the odd Fourier coefficients of } x(t). \end{aligned}$$

An alternative expression of the Fourier series expansion is

$$x(t) = v_0 + \sum_{k=1}^{\infty} v_k \cos(k\omega_0 t - \varphi_k) \quad (2.25)$$

where

$$v_0 = \frac{a_0}{2}, \quad v_k = \sqrt{a_k^2 + b_k^2}, \quad \varphi_k = \arctan\left(\frac{b_k}{a_k}\right).$$

Eq. (2.25) defines the signal as a linear combination of a constant term ( $v_0$ ) and of an infinite number of sinusoidal functions (the *harmonic functions*) at frequencies  $k\omega_0$ ;  $v_k$  represents the weight of the  $k$ -th harmonic function on  $x(t)$ , and  $\varphi_k$  its phase. The *maximum frequency* of the signal corresponds to the maximum  $k$  for which  $v_k \neq 0$  from a practical point of view. On the basis of the Fourier series expansion of a signal, it is then possible to understand its properties in the frequency domain.

The basic idea of the techniques for planning the motion profiles illustrated below is to compute a Fourier series expansion of a function  $q(t)$  defined by one of the methods presented in the previous sections and, then, define a new trajectory  $q_f(t)$  by considering only the first  $N$  terms of the series. In this manner, it is possible to obtain a function that presents specific properties in the frequency domain, see also Sec. 7.3.

### 2.4.1 Gutman 1-3

This trajectory is obtained as Fourier series expansion of the parabolic profile, Sec. 2.1.2, by taking into consideration only the first two elements, [15]:

$$\left\{ \begin{array}{l} q(t) = q_0 + h \left( \frac{(t-t_0)}{T} - \frac{15}{32\pi} \sin \frac{2\pi(t-t_0)}{T} - \frac{1}{96\pi} \sin \frac{6\pi(t-t_0)}{T} \right) \\ \dot{q}(t) = \frac{h}{T} \left( 1 - \frac{15}{16} \cos \frac{2\pi(t-t_0)}{T} - \frac{1}{16} \cos \frac{6\pi(t-t_0)}{T} \right) \\ \ddot{q}(t) = \frac{h\pi}{8T^2} \left( 15 \sin \frac{2\pi(t-t_0)}{T} + 3 \sin \frac{6\pi(t-t_0)}{T} \right) \\ q^{(3)}(t) = \frac{h\pi^2}{4T^3} \left( 15 \cos \frac{2\pi(t-t_0)}{T} + 9 \cos \frac{6\pi(t-t_0)}{T} \right) \end{array} \right.$$

where  $h$  is the displacement and  $T$  the time duration. The maximum acceleration is  $5.15 h/T^2$ , i.e. 28.75% larger than the maximum acceleration of the parabolic trajectory ( $4h/T^2$ ) and, for example, 18.04% smaller than the maximum acceleration of the cycloidal trajectory ( $2\pi h/T^2$ ). On the other hand, the frequency content is lower with respect to the parabolic profile, and higher than the cycloidal one, see Chapter 7.

**Example 2.19** Fig. 2.27 reports the position, velocity, acceleration and jerk for the Gutman 1-3 trajectory with  $h = 20$  and  $T = 10$  ( $q_0 = 0$ ,  $t_0 = 0$ ).  $\square$

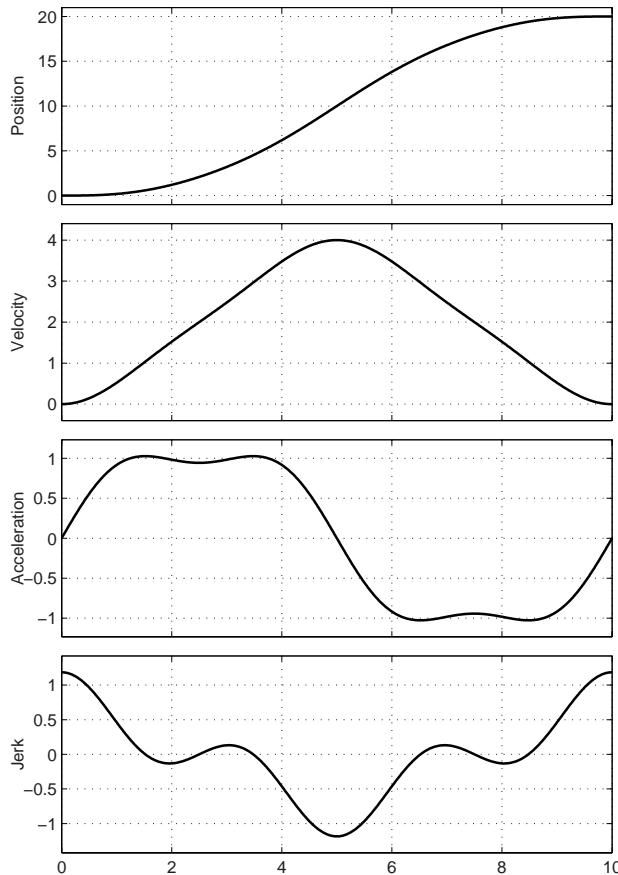
#### 2.4.2 Freudenstein 1-3

As in the previous case, only the first and the third terms of the Fourier series expansion of the parabolic trajectory are considered, but the trajectory is defined as [16]

$$\left\{ \begin{array}{l} q(t) = q_0 + \frac{h(t-t_0)}{T} - \frac{h}{2\pi} \left( \frac{27}{28} \sin \frac{2\pi(t-t_0)}{T} + \frac{1}{84} \sin \frac{6\pi(t-t_0)}{T} \right) \\ \dot{q}(t) = \frac{h}{T} \left( 1 - \frac{27}{28} \cos \frac{2\pi(t-t_0)}{T} - \frac{1}{28} \cos \frac{6\pi(t-t_0)}{T} \right) \\ \ddot{q}(t) = \frac{2\pi h}{T^2} \left( \frac{27}{28} \sin \frac{2\pi(t-t_0)}{T} + \frac{3}{28} \sin \frac{6\pi(t-t_0)}{T} \right) \\ q^{(3)}(t) = \frac{4\pi^2 h}{T^3} \left( \frac{27}{28} \cos \frac{2\pi(t-t_0)}{T} + \frac{9}{28} \cos \frac{6\pi(t-t_0)}{T} \right). \end{array} \right.$$

This trajectory has a maximum acceleration value equal to  $5.39 h/T^2$ , i.e. 34.75% larger than the parabolic profile and 14.22% smaller than the cycloidal trajectory.

**Example 2.20** Fig. 2.28 shows the position, velocity, acceleration and jerk for the Freudenstein 1-3 trajectory, with  $h = 20$  and  $T = 10$  ( $q_0 = 0$ ,  $t_0 = 0$ ).  $\square$

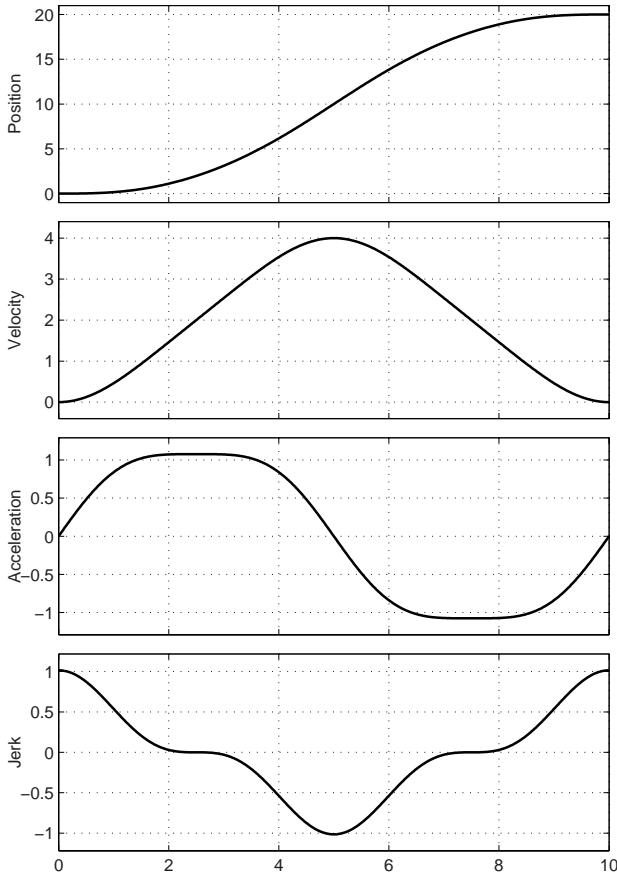


**Fig. 2.27.** Position, velocity, acceleration and jerk of the Gutman 1-3 trajectory with  $h = 20$  and  $T = 10$ .

### 2.4.3 Freudenstein 1-3-5

This trajectory is defined as

$$\begin{cases} q(t) = q_0 + \frac{h(t-t_0)}{T} - \frac{h}{2\pi}\alpha \left( \sin \frac{2\pi(t-t_0)}{T} + \frac{1}{54} \sin \frac{6\pi(t-t_0)}{T} + \frac{1}{1250} \sin \frac{10\pi(t-t_0)}{T} \right) \\ \dot{q}(t) = \frac{h}{T} \left[ 1 - \alpha \left( \cos \frac{2\pi(t-t_0)}{T} + \frac{1}{18} \cos \frac{6\pi(t-t_0)}{T} + \frac{1}{250} \cos \frac{10\pi(t-t_0)}{T} \right) \right] \\ \ddot{q}(t) = \frac{2\pi h}{T^2} \alpha \left( \sin \frac{2\pi(t-t_0)}{T} + \frac{1}{6} \sin \frac{6\pi(t-t_0)}{T} + \frac{1}{50} \sin \frac{10\pi(t-t_0)}{T} \right) \\ q^{(3)}(t) = \frac{4\pi^2 h}{T^3} \alpha \left( \cos \frac{2\pi(t-t_0)}{T} + \frac{1}{2} \cos \frac{6\pi(t-t_0)}{T} + \frac{1}{10} \cos \frac{10\pi(t-t_0)}{T} \right) \end{cases}$$

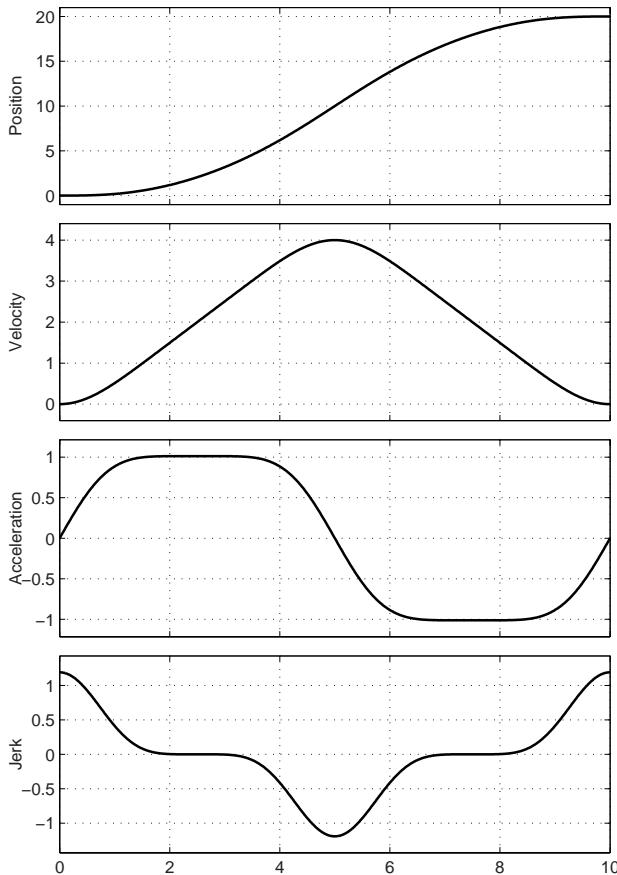


**Fig. 2.28.** Position, velocity, acceleration and jerk of the Freudenstein 1-3 trajectory with  $h = 20$  and  $T = 10$ .

where  $\alpha = \frac{1125}{1192} = 0.9438$ . This trajectory has a maximum acceleration value equal to  $5.06 h/T^2$ , i.e. 26.5% larger than the parabolic motion, and 19.47% smaller than the cycloidal profile.

**Example 2.21** Fig. 2.29 shows the position, velocity, acceleration and jerk for the Freudenstein 1-3-5 trajectory, with  $h = 20$  and  $T = 10$  ( $q_0 = 0, t_0 = 0$ ).  $\square$

If a larger number of terms of the Fourier series expansion is considered, profiles with lower acceleration values but higher frequency components are obtained. As discussed in Chapter 7, this could generate undesired vibrations in the mechanical structure. A compromise has then to be obtained between the requirements of low acceleration values and frequency bandwidth of the corresponding signal. An empiric rule could be to limit the maximum frequency



**Fig. 2.29.** Position, velocity, acceleration and jerk of the Freudenstein 1-3-5 trajectory with  $h = 20$  and  $T = 10$ .

of the trajectory to  $\omega_r/10$ , being  $\omega_r$  the lower resonance frequency of the mechanical structure under consideration. This can be obtained by truncating, for example, the Fourier series expansion to  $N$ , where  $N = \text{floor}\left(\frac{\omega_r/10}{\omega_0}\right)$ , where  $\omega_0 = 2\pi/T$ ,  $T$  is the period of the trajectory, and  $\text{floor}(x)$  is the function which gives the largest integer less than or equal to  $x$ .

## Composition of Elementary Trajectories

Often, useful motion profiles may be obtained as proper combinations of the functions which define the elementary trajectories. In fact, one may be interested in obtaining not only a continuous function with continuous derivatives up to a given order, but also other features, for example minimum values for the maximum acceleration or jerk. Trajectories obtained by a proper composition of the functions illustrated in Chapter 2 are now presented. Some of these are very common in the industrial practice, such as the “trapezoidal velocity” or the “double S” trajectories.

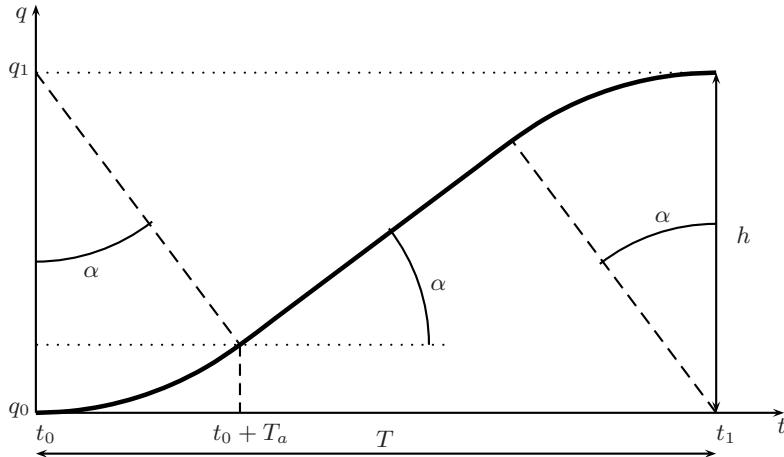
---

### 3.1 Linear Trajectory with Circular Blends

The trajectory with constant velocity, presented in Sec. 2.1.1, cannot be used in practice since it presents discontinuous velocity and acceleration (impulses with infinite amplitude at the beginning and at the end of the motion) profiles. In order to obtain at least a continuous velocity profile, it may be modified by adding circular arcs at the beginning and at the end of the trajectory, as shown in Fig. 3.1. In this case, the circular arcs are characterized by radii equal to the displacement  $h = q_1 - q_0$ , and centers located in  $(0, h)$ ,  $(T, 0)$  respectively, with  $T = t_1 - t_0$ .

The trajectory is divided into three phases: acceleration, constant velocity, and deceleration phase. The acceleration and deceleration phases have the same duration  $T_a = h \sin \alpha$  (with  $\alpha$  defined below), so that the circular arcs are connected with the (tangent) linear segment. The trajectory is described by the following equations:

$$1) \quad t_0 \leq t < t_0 + T_a$$



**Fig. 3.1.** Trajectory with constant velocity and circular blends.

$$\begin{cases} q_a(t) = h \left( 1 - \sqrt{1 - \frac{(t-t_0)^2}{h^2}} \right) + q_0 \\ \dot{q}_a(t) = \frac{t-t_0}{\sqrt{h^2 - (t-t_0)^2}} \\ \ddot{q}_a(t) = \frac{h^2}{\sqrt{[h^2 - (t-t_0)^2]^3}}. \end{cases}$$

2)  $t_0 + T_a \leq t < t_1 - T_a$

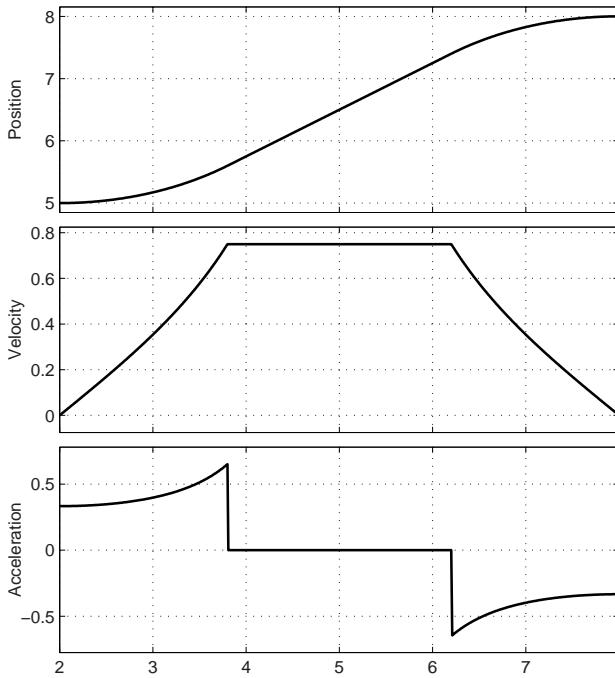
$$\begin{cases} q_b(t) = a_0 + a_1(t - t_0) \\ \dot{q}_b(t) = a_1 \\ \ddot{q}_b(t) = 0. \end{cases}$$

3)  $t_1 - T_a \leq t < t_1$

$$\begin{cases} q_c(t) = q_0 + \sqrt{h^2 - (t_1 - t)^2} \\ \dot{q}_c(t) = \frac{t_1 - t}{\sqrt{h^2 - (t_1 - t)^2}} \\ \ddot{q}_c(t) = -\frac{h^2}{\sqrt{[h^2 - (t_1 - t)^2]^3}}. \end{cases}$$

The parameters  $a_0$  and  $a_1$  are determined on the basis of the continuity conditions of position and velocity at  $t = t_a = t_0 + T_a$ :

$$\begin{cases} q_a(t_a) = q_b(t_a) \\ \dot{q}_a(t_a) = \dot{q}_b(t_a) \end{cases} \quad t_a = t_0 + T_a.$$



**Fig. 3.2.** Position, velocity and acceleration of a trajectory with constant velocity and circular connections when  $t_0 = 2$ ,  $t_1 = 8$ ,  $q_0 = 5$ ,  $q_1 = 8$ .

From the second condition, it follows

$$a_1 = \frac{h \sin \alpha}{\sqrt{h^2 - h^2 \sin \alpha^2}} = \frac{h \sin \alpha}{h \cos \alpha} = \tan \alpha$$

while from the first one

$$\left( 1 - \sqrt{1 - \frac{h^2 \sin \alpha^2}{h^2}} \right) + q_0 = a_0 + h \tan \alpha \sin \alpha$$

and accordingly

$$a_0 = h \frac{\cos \alpha - 1}{\cos \alpha} + q_0.$$

Finally, the parameter  $\alpha$  can be determined with simple geometrical considerations. From Fig. 3.1 one obtains

$$\tan \alpha = \frac{h - 2h(1 - \cos \alpha)}{T - 2h \sin \alpha}$$

from which, by considering only the solution in  $[0, \pi/2]$

$$\alpha = \arccos \frac{2h^2 + T\sqrt{T^2 - 3h^2}}{h^2 + T^2}.$$

Note that, from the assumption on the radius of the circular arcs, equal to  $h$ , the duration  $T$  must satisfy the constraint

$$T \geq \sqrt{3} h = \sqrt{3} (q_1 - q_0).$$

**Example 3.1** Fig. 3.2 reports the position, velocity and acceleration for this trajectory when  $t_0 = 2$ ,  $t_1 = 8$ ,  $q_0 = 5$ ,  $q_1 = 8$ .  $\square$

### 3.2 Linear Trajectory with Parabolic Blends (Trapezoidal)

A very common method to obtain trajectories with a continuous velocity profile is to use linear motions with parabolic blends, characterized therefore by the typical trapezoidal velocity profiles.

These trajectories are divided into three parts. Assuming a positive displacement, i.e.  $q_1 > q_0$ <sup>1</sup>, in the first part the acceleration is positive and constant, and therefore the velocity is a linear function of time and the position is a parabolic curve. In the second part the acceleration is null, the velocity is constant and the position is a linear function of time. In the last part, a constant negative acceleration is present, the velocity decreases linearly and the position is again a polynomial function of degree two, see Fig. 3.3. For these trajectories, the duration  $T_a$  of the acceleration phase is usually assumed equal to the duration  $T_d$  of the deceleration phase.

If  $t_0 = 0$ , the trajectory is computed as follows:

1. **Acceleration phase**,  $t \in [0, T_a]$ . The position, velocity and acceleration are expressed as

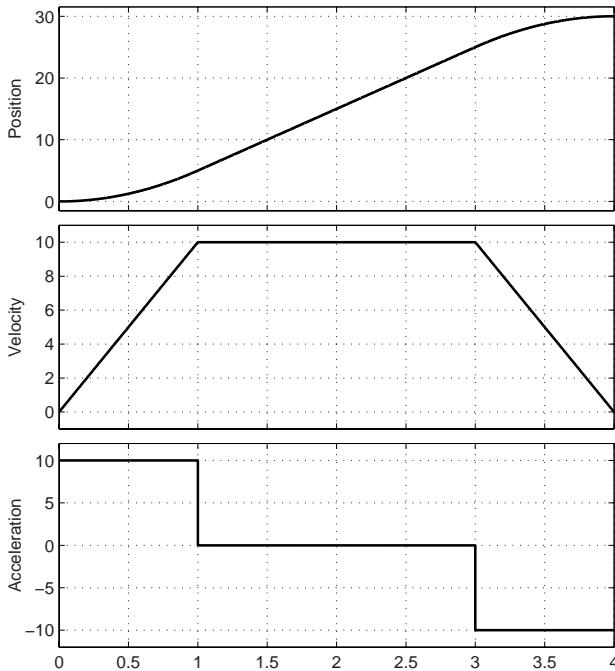
$$\begin{cases} q(t) = a_0 + a_1 t + a_2 t^2 \\ \dot{q}(t) = a_1 + 2a_2 t \\ \ddot{q}(t) = 2a_2. \end{cases} \quad (3.1)$$

The three parameters  $a_0$ ,  $a_1$ , and  $a_2$  are defined by the constraints on the initial position  $q_0$  and velocity  $v_0$ , and on the constant velocity  $v_v$  desired at the end of the acceleration phase. If the initial velocity is set to zero, one obtains

$$\begin{cases} a_0 = q_0 \\ a_1 = 0 \\ a_2 = \frac{v_v}{2T_a}. \end{cases}$$

---

<sup>1</sup> If  $q_1 < q_0$ , all the relations illustrated in this section hold with a proper change of the signs for accelerations and velocities, see Sec. 3.4.2.



**Fig. 3.3.** Position, velocity and acceleration of a linear trajectory with parabolic blends.

In this phase, the acceleration is constant and its value is  $v_v/T_a$ .

2. **Constant velocity phase**,  $t \in [T_a, t_1 - T_a]$ . Position, velocity and acceleration are now defined as

$$\begin{cases} q(t) = b_0 + b_1 t \\ \dot{q}(t) = b_1 \\ \ddot{q}(t) = 0 \end{cases} \quad (3.2)$$

where, for continuity reasons,

$$b_1 = v_v$$

and

$$q(T_a) = q_0 + \frac{v_v T_a}{2} = b_0 + v_v T_a$$

from which

$$b_0 = q_0 - \frac{v_v T_a}{2}.$$

3. **Deceleration phase**,  $t \in [t_1 - T_a, t_1]$ . Position, velocity and acceleration are

$$\begin{cases} q(t) = c_0 + c_1 t + c_2 t^2 \\ \dot{q}(t) = c_1 + 2c_2 t \\ \ddot{q}(t) = 2c_2. \end{cases} \quad (3.3)$$

The parameters are now defined with conditions on the final position  $q_1$  and velocity  $v_v$ , and on the velocity  $v_v$  at the beginning of the deceleration phase. With a null final velocity one obtains

$$\begin{cases} c_0 = q_1 - \frac{v_v t_1^2}{2T_a} \\ c_1 = \frac{v_v t_1}{T_a} \\ c_2 = -\frac{v_v}{2T_a}. \end{cases}$$

In conclusion, considering the general case  $t_0 \neq 0$ , the trajectory (in position) is defined as

$$q(t) = \begin{cases} q_0 + \frac{v_v}{2T_a} (t - t_0)^2, & t_0 \leq t < t_0 + T_a \\ q_0 + v_v \left( t - t_0 - \frac{T_a}{2} \right), & t_0 + T_a \leq t < t_1 - T_a \\ q_1 - \frac{v_v}{2T_a} (t_1 - t)^2, & t_1 - T_a \leq t \leq t_1. \end{cases} \quad (3.4)$$

**Example 3.2** Fig. 3.3 shows the position, velocity and acceleration of a typical trajectory with trapezoidal velocity, with the conditions  $q_0 = 0$ ,  $q_1 = 30$ ,  $t_0 = 0$ ,  $t_1 = 4$ ,  $T_a = 1$ ,  $v_v = 10$ .  $\square$

Note that some additional conditions must be specified in order to determine univocally the trapezoidal trajectory. A typical condition concerns the time-length of the acceleration and deceleration periods  $T_a$ , that must satisfy the obvious condition  $T_a \leq T/2 = (t_1 - t_0)/2$ . Moreover, there might be some conditions on the maximum velocity and acceleration of the actuation system. Obviously, these conditions affect the feasibility of the trajectory.

In any case, the given conditions must satisfy some geometric constraints. In particular, from the velocity continuity condition in  $t = t_0 + T_a$  one obtains

$$a_a T_a = \frac{q_m - q_a}{T_m - T_a}, \quad \text{where} \quad \begin{cases} q_a = q(t_0 + T_a) \\ q_m = (q_1 + q_0)/2 \quad (= q_0 + h/2) \\ T_m = (t_1 - t_0)/2 \quad (= T/2) \end{cases}$$

where  $a_a$  is the constant acceleration value in the first phase, and from (3.4)

$$q_a = q_0 + \frac{1}{2} a_a T_a^2.$$

From these two equations, it is easy to obtain

$$\mathbf{a}_a T_a^2 - \mathbf{a}_a(t_1 - t_0)T_a + (q_1 - q_0) = 0. \quad (3.5)$$

Moreover,

$$v_v = \frac{q_1 - q_0}{t_1 - t_0 - T_a} = \frac{h}{T - T_a}.$$

Any couple  $(\mathbf{a}_a, T_a)$  satisfying (3.5) can be considered. For example,  $T_a$  can be assigned and therefore the acceleration (and velocity) is computed accordingly. If the value  $T_a = (t_1 - t_0)/3$  is chosen, the following values are obtained

$$v_v = \frac{3(q_1 - q_0)}{2(t_1 - t_0)} = \frac{3h}{2T}, \quad \mathbf{a}_a = \frac{9(q_1 - q_0)}{2(t_1 - t_0)^2} = \frac{9h}{2T^2}.$$

If the velocity obtained in this manner is too high for the actuation system, i.e.  $v_v > v_{max}$ , then the parameter  $T_a$  must be decreased (and  $\mathbf{a}_a$  is modified according to (3.5)), or  $T$  (the time duration of the trajectory) must be increased. If the value of the acceleration is too high, i.e.  $\mathbf{a}_a > \mathbf{a}_{max}$ , then  $T_a$  must be increased.

### 3.2.1 Trajectory with preassigned acceleration

Another method for defining this trajectory is to assign the maximum value for the desired acceleration  $\mathbf{a}_a$  and then compute the acceleration/deceleration period  $T_a$ . In fact, from (3.5), if  $\mathbf{a}_a$  is assigned, the acceleration period is

$$T_a = \frac{\mathbf{a}_a(t_1 - t_0) - \sqrt{\mathbf{a}_a^2(t_1 - t_0)^2 - 4\mathbf{a}_a(q_1 - q_0)}}{2\mathbf{a}_a}. \quad (3.6)$$

From this equation also the minimum value for the acceleration is obtained

$$\mathbf{a}_a \geq \frac{4(q_1 - q_0)}{(t_1 - t_0)^2} = \frac{4h}{T^2}. \quad (3.7)$$

If the value  $\mathbf{a}_a = 4h/T^2$  is chosen, then  $T_a = \frac{1}{2}(t_1 - t_0)$  and the parabolic trajectory illustrated in Sec. 2.1.2 is obtained.

### 3.2.2 Trajectory with preassigned acceleration and velocity

A trajectory with desired values for the acceleration and velocity (e.g.  $\mathbf{a}_a = \mathbf{a}_{max}$ ,  $v_v = v_{max}$ ) can be achieved by setting

$$\begin{cases} T_a = \frac{v_{max}}{\mathbf{a}_{max}}, & \text{acceleration time} \\ v_{max}(T - T_a) = q_1 - q_0 = h, & \text{displacement} \\ T = \frac{h\mathbf{a}_{max} + v_{max}^2}{\mathbf{a}_{max}v_{max}}, & \text{total duration} \end{cases} \quad (3.8)$$

and then (with  $t_1 = t_0 + T$ )

$$q(t) = \begin{cases} q_0 + \frac{1}{2}a_{max}(t - t_0)^2, & t_0 \leq t \leq t_0 + T_a \\ q_0 + a_{max}T_a \left( t - t_0 - \frac{T_a}{2} \right), & t_0 + T_a < t \leq t_1 - T_a \\ q_1 - \frac{1}{2}a_{max}(t_1 - t)^2, & t_1 - T_a < t \leq t_1. \end{cases} \quad (3.9)$$

In this case, the linear segment exists if and only if

$$h \geq \frac{v_{max}^2}{a_{max}}.$$

If this condition is not true, then

$$\begin{cases} T_a = \sqrt{\frac{h}{a_{max}}}, & \text{acceleration time} \\ T = 2T_a, & \text{total time} \\ v_{max} = a_{max}T_a = \sqrt{a_{max}h} = \frac{h}{T_a}, & \text{maximum velocity} \end{cases}$$

and ( $t_1 = t_0 + T$ )

$$q(t) = \begin{cases} q_0 + \frac{1}{2}a_{max}(t - t_0)^2, & t_0 \leq t \leq t_0 + T_a \\ q_1 - \frac{1}{2}a_{max}(t_1 - t)^2, & t_1 - T_a < t \leq t_1. \end{cases} \quad (3.10)$$

In this manner, the total duration  $T$  of the motion from  $q_0$  to  $q_1$  is not specified, but is computed on the basis of the specified values for the acceleration and the velocity.

### 3.2.3 Synchronization of several trapezoidal trajectories

If several actuators must be coordinated, all the movements must be the defined according to the slowest one, or the one with the largest displacement. For example, let us consider the case in which the motion of several actuators must be planned with the same constraints, in terms of maximum acceleration and velocities. In this case, the maximum acceleration  $a_{max}$  is imposed for the actuator with the largest displacement, and the acceleration period  $T_a$  and the total duration  $T$  are computed with the above equations. Once these values are determined, the acceleration and velocity of the remaining actuators are computed given the relative displacements  $h_i$ , see Sec. 3.2.6:

$$a_i = \frac{h_i}{T_a(T - T_a)}, \quad v_i = \frac{h_i}{T - T_a}.$$

Then, (3.4) can be used for the computation of each trajectory.

**Example 3.3** Let us consider three actuators with maximum velocity and acceleration values given by  $v_{max} = 20$  and  $a_{max} = 20$ . Three synchronized displacements have to be defined, so that the duration  $T$  is minimized and the acceleration/deceleration periods are the same for the three actuators. The displacements are defined by:

$$\begin{aligned} a) \quad q_{0,a} &= 0, & q_{1,a} &= 50. \\ b) \quad q_{0,b} &= 0, & q_{1,b} &= -40. \\ c) \quad q_{0,c} &= 0, & q_{1,c} &= 20. \end{aligned}$$

The actuator  $a$  has the largest displacement ( $h_a = 50$ ), and therefore the duration  $T$  and the acceleration/deceleration intervals are imposed by this actuator. Since  $h_a > v_{max}^2/a_{max}$ , the linear segment exists and, from eq. (3.8) and the values of  $v_{max}$ ,  $a_{max}$ , it results

$$T_a = 1, \quad T = 3.5.$$

Then, the values of the minimum/maximum velocities and accelerations for the other two actuators can be computed as:

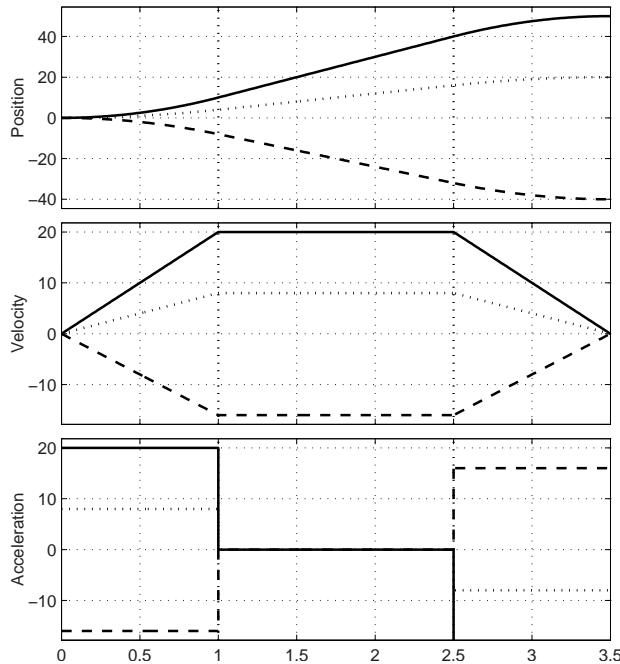
$$\begin{aligned} b) \quad a_b &= \frac{h_b}{T_a(T - T_a)} = -16, & v_b &= \frac{h_b}{T - T_a} = -16. \\ c) \quad a_c &= \frac{h_c}{T_a(T - T_a)} = 8, & v_c &= \frac{h_c}{T - T_a} = 8. \end{aligned}$$

Note that the actuator  $b$  has a negative displacement ( $h_b < 0$ ) and therefore the acceleration/deceleration intervals are switched. Moreover, because of symmetry, it results  $a_{min} = -a_{max}$  and  $v_{min} = -v_{max}$ . The values for the time intervals  $T$ ,  $T_a$ , and the maximum acceleration/velocity of each actuator can be used to plan the three trajectories. The result is shown in Fig. 3.4.  $\square$

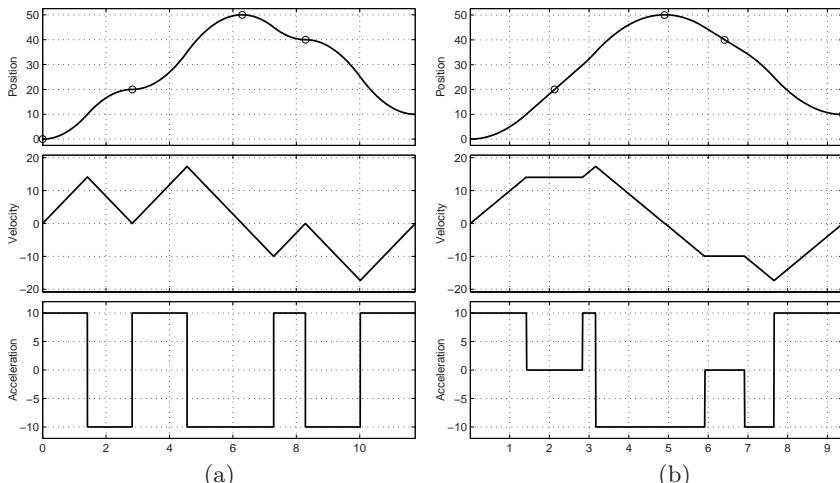
### 3.2.4 Trajectory through a sequence of points

If a trajectory through a sequence of points is planned with the above technique, the resulting motion will present null velocities in the intermediate points. Since this may be unacceptable, the generic intermediate motion between the points  $q_k$  and  $q_{k+1}$  may be “anticipated” in such a way that it starts before the motion between the points  $q_{k-1}$  and  $q_k$  is concluded. This is obtained by adding from the instant  $t_k - T'_{ak}$  the velocity and acceleration profiles of the two segments  $q_{k-1} \div q_k$  and  $q_k \div q_{k+1}$ .

**Example 3.4** Fig. 3.5(a) shows the position, velocity and acceleration profiles of a trapezoidal trajectory through a sequence of points with null intermediate velocities. Fig. 3.5(b) shows the same motion with non-null intermediate velocities. Note that in this latter case the duration  $T$  of the trajectory is shorter.  $\square$



**Fig. 3.4.** Three synchronized trapezoidal trajectories.



**Fig. 3.5.** Position, velocity and acceleration of a trapezoidal trajectory through a sequence of points with null (a) and non-null (b) intermediate velocities.

### 3.2.5 Displacement time of a trapezoidal trajectory

It may be of interest to consider the displacement time  $T$  of a trajectory with trapezoidal velocity profile with null initial and final velocities.

In case the maximum velocity is not reached, a triangular velocity profile is obtained, whose duration is

$$T = 2T_a = 2\sqrt{\frac{h}{a_{max}}}.$$

In this case, the peak value of the velocity is

$$v_{lim} = 2\frac{h}{T} = \frac{h}{T_a}.$$

Conversely, if the maximum velocity is reached, the trajectory duration is

$$T = \frac{h a_{max} + v_{max}^2}{a_{max} v_{max}} = \frac{h}{v_{max}} + T_a \quad (3.11)$$

where  $T_a$  is the duration of the acceleration period:

$$T_a = \frac{v_{max}}{a_{max}}. \quad (3.12)$$

### 3.2.6 Trajectory with assigned durations $T$ and $T_a$

By solving the system composed by equations (3.11) and (3.12) with respect to  $v_{max}$  and  $a_{max}$ , it is possible to find the values of these variables which guarantee given durations  $T$  and  $T_a$  of the overall trapezoidal trajectory and of the acceleration phase. In particular, one obtains

$$\begin{cases} v_{max} = \frac{h}{T - T_a} \\ a_{max} = \frac{h}{T_a(T - T_a)}. \end{cases}$$

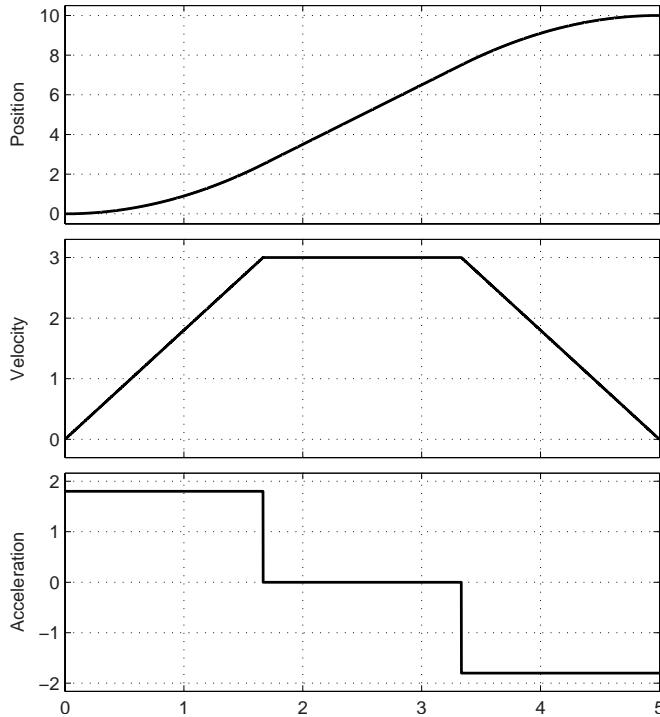
If one assumes that the acceleration period is a fraction of the duration  $T$

$$T_a = \alpha T, \quad 0 < \alpha \leq 1/2$$

the expressions of the maximum speed and acceleration become

$$\begin{cases} v_{max} = \frac{h}{(1 - \alpha)T} \\ a_{max} = \frac{h}{\alpha(1 - \alpha)T^2}. \end{cases}$$

By substituting these values in (3.9), the trajectory is completely defined.



**Fig. 3.6.** Trapezoidal trajectory with prescribed duration  $T$ , and with  $T_a = T/3$ .

**Example 3.5** A trapezoidal trajectory with the boundary conditions

$$q_0 = 0, \quad q_1 = 10, \quad v_0 = 0, \quad v_1 = 0,$$

is computed with the purpose of obtaining a total duration  $T = 5$ . Moreover, it is imposed an acceleration period equal to  $T_a = T/3$ . As a consequence, the values of the velocity and acceleration are

$$v_{max} = 3, \quad a_{max} = 1.8.$$

The trajectory reported in Fig. 3.6 is obtained. □

### 3.2.7 Trajectory with non-null initial and final velocities

The general expression of the trapezoidal trajectory, when non-null initial and final velocities ( $v_0$  and  $v_1$ ) are considered, is

**Acceleration phase**,  $t \in [t_0, t_0 + T_a]$

$$\begin{cases} q(t) = q_0 + v_0(t - t_0) + \frac{v_v - v_0}{2T_a}(t - t_0)^2 \\ \dot{q}(t) = v_0 + \frac{v_v - v_0}{T_a}(t - t_0) \\ \ddot{q}(t) = \frac{v_v - v_0}{T_a} = a_a. \end{cases} \quad (3.13a)$$

**Constant velocity phase,**  $t \in [t_0 + T_a, t_1 - T_d]$

$$\begin{cases} q(t) = q_0 + v_0 \frac{T_a}{2} + v_v \left( t - t_0 - \frac{T_a}{2} \right) \\ \dot{q}(t) = v_v \\ \ddot{q}(t) = 0. \end{cases} \quad (3.13b)$$

**Deceleration phase,**  $t \in [t_1 - T_d, t_1]$

$$\begin{cases} q(t) = q_1 - v_1(t_1 - t) - \frac{v_v - v_1}{2T_d}(t_1 - t)^2 \\ \dot{q}(t) = v_1 + \frac{v_v - v_1}{T_d}(t_1 - t) \\ \ddot{q}(t) = -\frac{v_v - v_1}{T_d} = -a_a. \end{cases} \quad (3.13c)$$

These equations represent a generalization of (3.4). Note that in general the duration  $T_a$  of the acceleration phase and the duration  $T_d$  of the deceleration phase may be different (the case  $q_1 > q_0$  is considered).

### Trajectory with preassigned duration and acceleration

As in case of a trapezoidal trajectory with null initial and final velocities, the parameters in (3.13a)-(3.13c) are determined by imposing some additional conditions. It is possible to assign the maximum value for the desired acceleration ( $a_a = a_{max}$ ) and the desired duration of the trajectory ( $T = t_1 - t_0$ ), and then determine the acceleration/deceleration periods  $T_a/T_d$ .

It is first necessary to check whether the trajectory is feasible or not under the hypothesis  $\ddot{q}(t) \leq a_{max}$ . In case

$$a_{max}h < \frac{|v_0^2 - v_1^2|}{2} \quad (3.14)$$

it is not possible to find a trapezoidal trajectory compliant with the given constraints on initial and final velocities and maximum acceleration, since the displacement  $h$  is too small with respect to  $v_0$  or  $v_1$ . In this case, one should increase the maximum value of the acceleration, or reduce  $v_0$  or  $v_1$ . Obviously, if the initial and final speeds are both null, the trajectory is always feasible.

If the trapezoidal trajectory exists, it is possible to compute the constant velocity as

$$v_v = \frac{1}{2} \left( v_0 + v_1 + a_{max}T - \sqrt{a_{max}^2 T^2 - 4a_{max}h + 2a_{max}(v_0 + v_1)T - (v_0 - v_1)^2} \right)$$

and then the duration of the acceleration/deceleration tracts:

$$T_a = \frac{v_v - v_0}{a_{max}}, \quad T_d = \frac{v_v - v_1}{a_{max}}.$$

Obviously, the total duration of the trajectory must be greater than  $T_a + T_d$ . This can be assured by imposing

$$a_{max}^2 T^2 - 4a_{max}h + 2a_{max}(v_0 + v_1)T - (v_0 - v_1)^2 > 0.$$

Therefore, the acceleration must be larger than a limit value:

$$a_{max} \geq a_{lim} = \frac{2h - T(v_0 + v_1) + \sqrt{4h^2 - 4h(v_0 + v_1)T + 2(v_0^2 + v_1^2)T^2}}{T^2}. \quad (3.15)$$

In case  $a_{max} = a_{lim}$ , the constant velocity tract is not present, and the acceleration phase is immediately followed by the deceleration segment.

**Example 3.6** Figure 3.7 shows the position, velocity and acceleration of two trapezoidal velocity trajectories with non-null initial and final speeds, computed respectively with the constraints

$$T = 5, \quad a_{max} = 10$$

and

$$T = 5, \quad a_{max} = 1.$$

In both cases, initial and final conditions are

$$q_0 = 0, \quad q_1 = 30$$

and

$$v_0 = 5, \quad v_1 = 2.$$

Unfortunately, the constraint  $a_{max} = 1$  does not satisfy the condition (3.15) and therefore, in the second trajectory, the maximum value of the acceleration has been set to  $a_{lim} = 2.1662$ , and the constant velocity tract is not present.

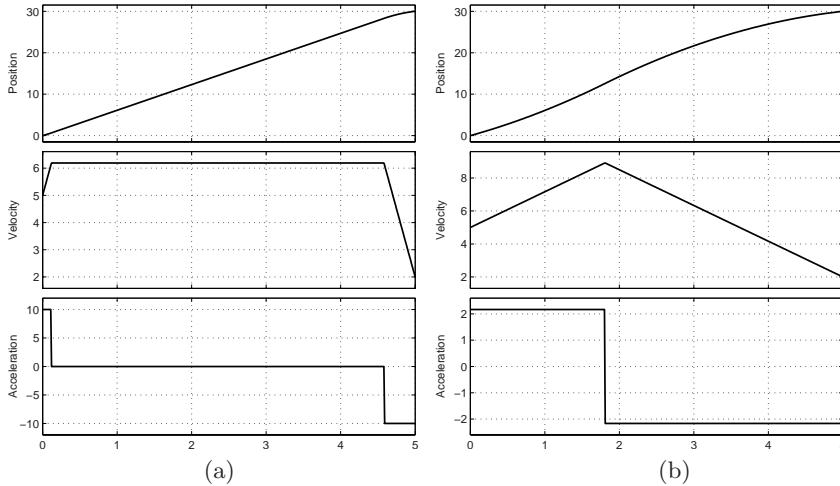
The time intervals which determine the trajectory in the two cases are respectively

$$T_a = 0.119, \quad T_d = 0.419$$

and

$$T_a = 1.8075, \quad T_d = 3.1925.$$

□



**Fig. 3.7.** Position, velocity and acceleration of a trapezoidal trajectory with non-null initial and final velocities computed for a given duration  $T$  and a maximum acceleration  $a_{max}$ , when the constant speed tract is present (a) or not (b).

### Trajectory with preassigned acceleration and velocity

In this case, the maximum values of the acceleration and the velocities are assigned, therefore

$$a_a = a_{max}, \quad v_v \leq v_{max}$$

while the total duration  $T$  of the trajectory is obtained as an output of the algorithm for the determination of the trapezoidal trajectory. At the beginning, it is necessary to check whether the trajectory is feasible or not by means of (3.14). If the trajectory exists, two cases are possible according to the fact that the maximum velocity  $v_{max}$  is reached or not; in this second case only the acceleration and deceleration phases exist, and the constant velocity segment is not present. If the condition

$$h a_{max} > v_{max}^2 - \frac{v_0^2 + v_1^2}{2}$$

is true,  $v_{max}$  is actually reached and maintained during the constant velocity phase (Case 1). Otherwise (Case 2), the maximum velocity of the trajectory is

$$v_{lim} = \sqrt{h a_{max} + \frac{v_0^2 + v_1^2}{2}} < v_{max}.$$

In the two cases, the parameters defining the trajectory can be computed as:

**Case 1:**  $v_v = v_{max}$ .

$$T_a = \frac{v_{max} - v_0}{a_{max}}, \quad T_d = \frac{v_{max} - v_1}{a_{max}},$$

$$T = \frac{h}{v_{max}} + \frac{v_{max}}{2a_{max}} \left( 1 - \frac{v_0}{v_{max}} \right)^2 + \frac{v_{max}}{2a_{max}} \left( 1 - \frac{v_1}{v_{max}} \right)^2.$$

**Case 2:**  $v_v = v_{lim} = \sqrt{h a_{max} + \frac{v_0^2 + v_1^2}{2}} < v_{max}$ .

$$T_a = \frac{v_{lim} - v_0}{a_{max}}, \quad T_d = \frac{v_{lim} - v_1}{a_{max}},$$

$$T = T_a + T_d.$$

If  $h < 0$ , the trajectory can be computed with a method similar to the one explained for double S trajectories in Sec. 3.4.2.

**Example 3.7** In Fig. 3.8 the position, velocity and acceleration of a trapezoidal velocity trajectory with non-null initial and final speeds are shown. In particular, two different situations are considered. In both cases, initial and final conditions are

$$q_0 = 0, \quad q_1 = 30$$

and

$$v_0 = 5, \quad v_1 = 2$$

while the maximum acceleration is  $a_{max} = 10$ . The difference between the trajectories is the value of maximum speed, that in the case a) is  $v_{max} = 10$  while in the case b) is  $v_{max} = 20$ . Because of this constraint, in the former case the maximum speed is reached, and the time intervals which determine the trajectory are

$$T_a = 0.5, \quad T_d = 0.8, \quad T = 3.44.$$

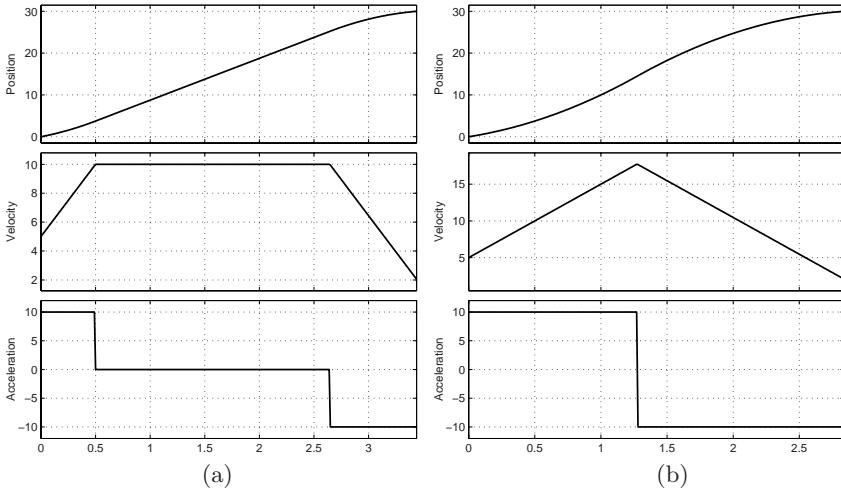
Conversely, in the latter case  $v_{lim} = 17.7 < v_{max}$  and only the acceleration and deceleration phases are present:

$$T_a = 1.27, \quad T_d = 1.57, \quad T = 2.84.$$

□

### Trajectory through a sequence of points

In Sec. 3.2.4 a trajectory through a set of points  $q_k$  with a trapezoidal velocity profile is obtained by superimposing contiguous trapezoidal trajectories, computed by considering null velocities at the internal via-points. Obviously, another possibility is to consider proper initial and final velocities for each segment. In particular, under the hypothesis that in each segment the speed  $v_{max}$  is reached, the velocities in the internal points can be computed as



**Fig. 3.8.** Position, velocity and acceleration of a trapezoidal trajectory with non-null initial and final velocities when the maximum speed  $v_{max}$  is reached (a) or not (b).

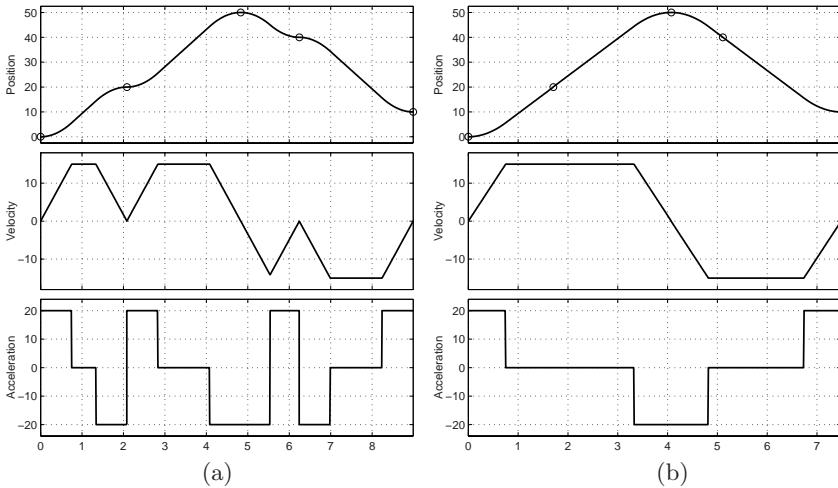
$$\begin{aligned}
 v_0 & \quad (\text{assigned}) \\
 v_k = \begin{cases} 0 & \text{sign}(h_k) \neq \text{sign}(h_{k+1}) \\ \text{sign}(h_k) v_{max} & \text{sign}(h_k) = \text{sign}(h_{k+1}) \end{cases} & \\
 v_n & \quad (\text{assigned})
 \end{aligned} \tag{3.16}$$

being  $h_k = q_k - q_{k-1}$ .

**Example 3.8** In Fig. 3.9 two trapezoidal trajectories passing through the same set of points of Example 3.4 and with the constraints

$$v_{max} = 15, \quad a_{max} = 20$$

are compared when null and non-null intermediate velocities are considered. Note that in this latter case the duration  $T$  of the trajectory is considerably shorter, and that velocity and acceleration profiles are less oscillating.  $\square$

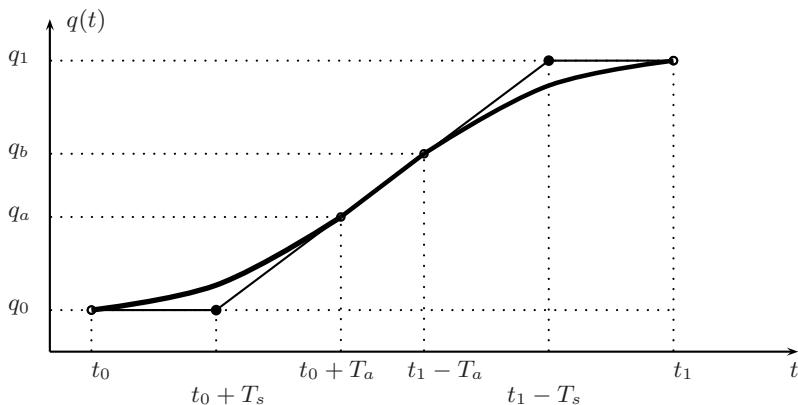


**Fig. 3.9.** Position, velocity and acceleration of a trapezoidal trajectory through a sequence of points with null (a) and non-null (b) intermediate velocities.

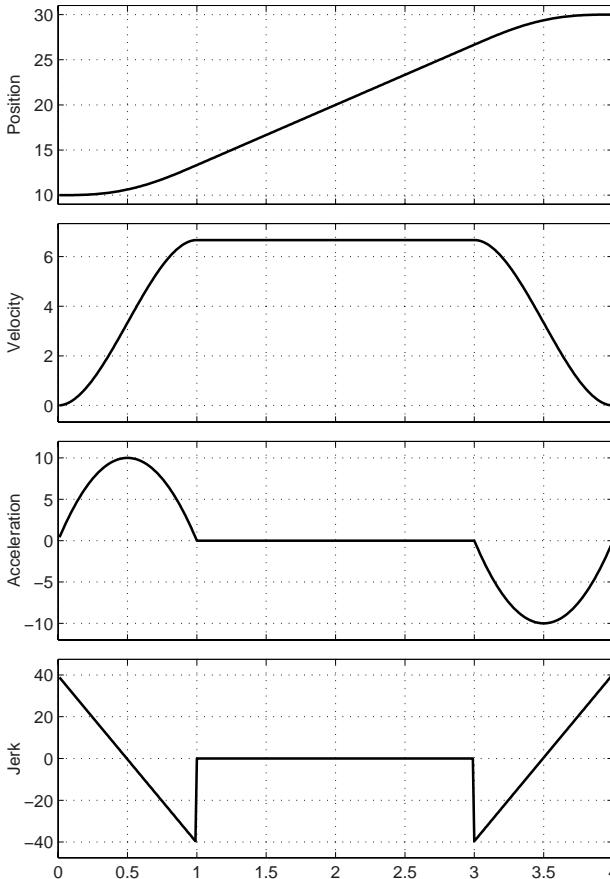
### 3.3 Linear Trajectory with Polynomial Blends

It is possible to define motions with profiles smoother than the trapezoidal velocity trajectories by defining the blends among the linear segments by means of polynomial functions with degree higher than two. Alternatively, it is possible to adopt trajectories with a *double S* velocity profile, quite common in the industrial practice, see Sec. 3.4.

In order to plan a linear trajectory with polynomial blends of degree  $n$ , the following general procedure can be adopted.



**Fig. 3.10.** Linear trajectory with polynomial blends.



**Fig. 3.11.** Linear trajectory with fifth degree polynomials blends.

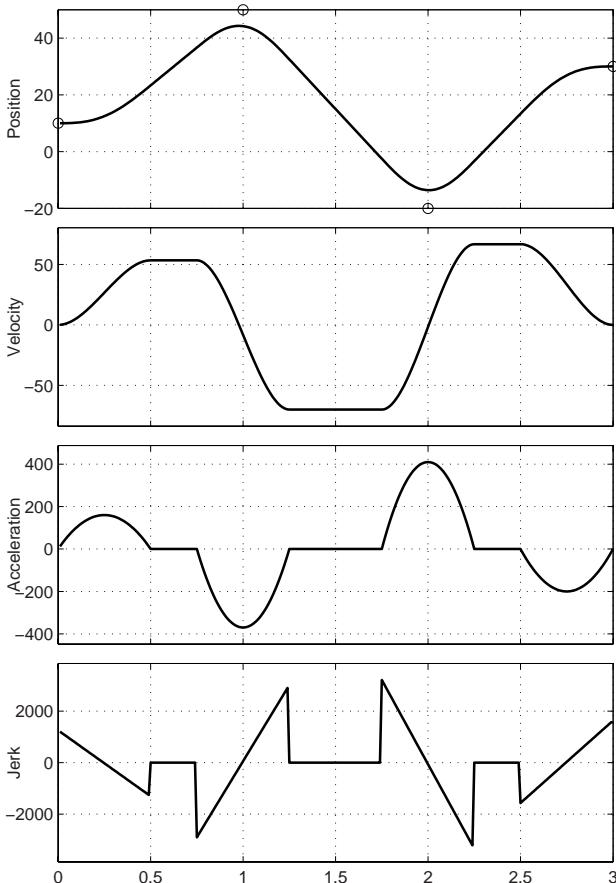
Let us define the points  $q_0$ ,  $q_1$ , the acceleration/deceleration period  $T_a = 2T_s$  and the total displacement time  $T = t_1 - t_0$ . With reference to Fig. 3.10:

1. Compute the expression  $q_r(t)$  of the line joining the points  $(t_0 + T_s, q_0)$  and  $(t_1 - T_s, q_1)$ .
2. On this line, compute the values  $q_a = q_r(t_0 + T_a)$  and  $q_b = q_r(t_1 - T_a)$ .
3. Assign the values  $\dot{q}(t_0) = \dot{q}(t_1) = 0$ ,  $\ddot{q}(t_0) = \ddot{q}(t_1) = 0$  and  $\dot{q}(t_0 + T_a) = \dot{q}(t_1 - T_a) = 0$ .
4. Compute the velocity in the segment  $(t_0 + T_a) \div (t_1 - T_a)$  as  $v_c = (q_b - q_a)/(t_1 - t_0 - 2T_a)$ .

Then, the trajectory can be computed in the two acceleration/deceleration phases by using the expressions already defined for the polynomials of degree  $n$  (for example, if  $n = 5$ , eq. (2.4)-(2.5)), and in the constant velocity segment with

$$\begin{cases} q(t) = v_c t + q_a \\ \dot{q}(t) = v_c \\ \ddot{q}(t) = 0. \end{cases}$$

See the profiles shown in Fig. 3.11. With minor modifications, this method can be applied also to trajectories through a sequence of points, and also in the case of *via-points* not exactly crossed by the trajectory. In this case, in the blend segment relative to the generic point  $q_k$  (with exclusion of the first and last ones), the acceleration and the velocity are computed so that the point  $(t, q(t))$ , at the end of the period  $T_a$ , lies on the linear segment connecting  $(t_k, q_k)$  to  $(t_{k+1}, q_{k+1})$ . See for example Fig. 3.12, in which the two via-points in  $t = 1, 2$  are not crossed by the trajectory.



**Fig. 3.12.** Linear trajectory with via-points approximated by fifth degree polynomials.

### 3.4 Trajectory with Double S Velocity Profile

A trapezoidal (or triangular) velocity motion profile presents a discontinuous acceleration. For this reason, this trajectory may generate efforts and stresses on the mechanical system that may result detrimental or generate undesired vibrational effects (see Chapter 7 for a more detailed discussion on these aspects). Therefore, a smoother motion profile must be defined, for example by adopting a continuous, linear piece-wise, acceleration profile as shown in Fig. 3.13. In this manner, the resulting velocity is composed by linear segments connected by parabolic blends. The shape of the velocity profile is the reason of the name *double S* for this trajectory, known also as *bell* trajectory or *seven segments* trajectory, because it is composed by seven different tracts with constant jerk. Since the jerk is characterized by a step profile, the stress and the vibrational effects generated on the transmission chain and on the load by this motion profile are reduced with respect to trapezoidal velocity trajectories, characterized by an impulsive jerk profile.

Let us assume that

$$j_{min} = -j_{max}, \quad a_{min} = -a_{max}, \quad v_{min} = -v_{max}$$

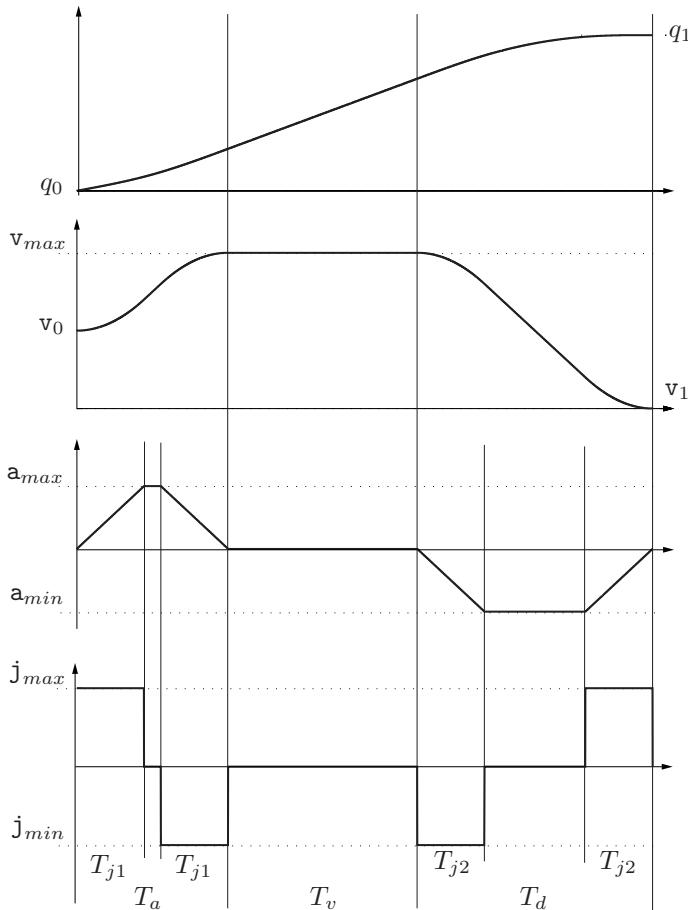
where  $j_{min}$  and  $j_{max}$  are the minimum and maximum value of the jerk, respectively. With these conditions, it is desired to plan a trajectory that, when possible, reaches the maximum (minimum) values for jerk, acceleration and velocity so that the total duration  $T$  is minimized (*minimum time* trajectory). Only the case  $q_1 > q_0$  is now considered. The case  $q_1 < q_0$  is addressed in the following Sec. 3.4.2. Moreover, for the sake of simplicity, the value  $t_0 = 0$  is assumed. The boundary conditions are:

- Generic initial and final values of velocity  $v_0, v_1$ .
- Initial and final accelerations  $a_0, a_1$  set to zero.

Three phases can be distinguished:

1. *Acceleration phase*,  $t \in [0, T_a]$ , where the acceleration has a linear profile from the initial value (zero) to the maximum and then back to zero.
2. *Maximum velocity phase*,  $t \in [T_a, T_a + T_v]$ , with a constant velocity.
3. *Deceleration phase*,  $t \in [T_a + T_v, T]$ , being  $T = T_a + T_v + T_d$ , with profiles opposite with respect to the acceleration phase.

Given the constraints on the maximum values of jerk, acceleration and velocity, and given the desired displacement  $h = q_1 - q_0$ , the trajectory is computed by using (3.30a)-(3.30g). However, it is first necessary to verify whether a trajectory can be actually performed or not. As a matter of fact, there are several cases in which a trajectory cannot be computed with the given constraints. For example, if the desired displacement  $h$  is small with respect to the difference between the initial and final velocities  $v_0$  and  $v_1$ , it might be not possible



**Fig. 3.13.** Typical profiles for position, velocity, acceleration and jerk for the double S trajectory.

to change the velocity (with the given limits on jerk and acceleration), while accomplishing the displacement  $h$ .

The limit case is represented by a single acceleration (if  $v_0 < v_1$ ) or deceleration (if  $v_0 > v_1$ ) phase. Therefore, it is first necessary to check if it is possible to perform the trajectory with a double jerk impulse (one positive and one negative) only. For this purpose, let us define

$$T_j^* = \min \left\{ \sqrt{\frac{|v_1 - v_0|}{j_{max}}}, \frac{a_{max}}{j_{max}} \right\}. \quad (3.17)$$

If  $T_j^* = a_{max}/j_{max}$ , the acceleration reaches its maximum value and a segment with zero jerk may exist.

Then, the trajectory is feasible if

$$q_1 - q_0 > \begin{cases} T_j^* (v_0 + v_1), & \text{if } T_j^* < \frac{a_{max}}{j_{max}} \\ \frac{1}{2} (v_0 + v_1) \left[ T_j^* + \frac{|v_1 - v_0|}{a_{max}} \right], & \text{if } T_j^* = \frac{a_{max}}{j_{max}}. \end{cases} \quad (3.18)$$

If this inequality holds, it is possible to compute the trajectory parameters. In this case, by defining the maximum value of the velocity during the motion as  $v_{lim} = \max(\dot{q}(t))$ , there are two possibilities:

Case 1:  $v_{lim} = v_{max}$ .

Case 2:  $v_{lim} < v_{max}$ .

In the latter case, that can be verified only after the computation of the trajectory's parameters, the maximum velocity is not reached, and there is only an acceleration and a deceleration phase (no segment with constant velocity).

In both Case 1 and Case 2, it is possible that the maximum acceleration (positive or negative) is not reached. This may happen if the displacement is small, if the maximum allowed acceleration  $a_{max}$  is high ("high dynamic" case), or if the initial (final) velocity is close enough to the maximum allowed speed. In these cases, the constant acceleration segment is not present. In particular, it is worth to notice that, because the different values of the initial and final velocities  $v_0, v_1$ , the amounts of time necessary to accelerate (from  $v_0$  to  $v_{lim}$ ) and to decelerate (from  $v_{lim}$  to  $v_1$ ) are in general different, and it may happen that the maximum acceleration  $a_{max}$  is reached only in one of these phases, while in the other one the maximum acceleration is  $a_{lim} < a_{max}$ . Let us define:

$T_{j1}$  : time-interval in which the jerk is constant ( $j_{max}$  or  $j_{min}$ ) during the acceleration phase;

$T_{j2}$  : time-interval in which the jerk is constant ( $j_{max}$  or  $j_{min}$ ) during the deceleration phase;

$T_a$  : acceleration period;

$T_v$  : constant velocity period;

$T_d$  : deceleration period;

$T$  : total duration of the trajectory ( $= T_a + T_v + T_d$ ).

#### Case 1: $v_{lim} = v_{max}$ .

In this case, it is possible to verify if the maximum acceleration ( $a_{max}$  or  $a_{min} = -a_{max}$ ) is reached by means of the following conditions:

$$\text{if } (v_{max} - v_0)j_{max} < a_{max}^2 \implies a_{max} \text{ is not reached;} \quad (3.19)$$

$$\text{if } (v_{max} - v_1)j_{max} < a_{max}^2 \implies a_{min} \text{ is not reached.} \quad (3.20)$$

Then, the time intervals of the acceleration segment can be computed if (3.19) holds as

$$T_{j1} = \sqrt{\frac{v_{max} - v_0}{j_{max}}}, \quad T_a = 2T_{j1}, \quad (3.21)$$

otherwise as

$$T_{j1} = \frac{a_{max}}{j_{max}}, \quad T_a = T_{j1} + \frac{v_{max} - v_0}{a_{max}}. \quad (3.22)$$

The time intervals of the deceleration segment can be computed if (3.20) holds as

$$T_{j2} = \sqrt{\frac{v_{max} - v_1}{j_{max}}}, \quad T_d = 2T_{j2}, \quad (3.23)$$

otherwise as

$$T_{j2} = \frac{a_{max}}{j_{max}}, \quad T_d = T_{j2} + \frac{v_{max} - v_1}{a_{max}}. \quad (3.24)$$

Finally, it is possible to determine the time duration of the constant velocity segment as

$$T_v = \frac{q_1 - q_0}{v_{max}} - \frac{T_a}{2} \left( 1 + \frac{v_0}{v_{max}} \right) - \frac{T_d}{2} \left( 1 + \frac{v_1}{v_{max}} \right). \quad (3.25)$$

If  $T_v > 0$ , then the maximum velocity is actually reached and the values obtained by (3.21)-(3.25) can be used to compute the trajectory.

The condition  $T_v < 0$  simply means that the maximum velocity  $v_{lim}$  is smaller than  $v_{max}$  and, therefore, the following Case 2 must be considered.

**Example 3.9** Fig. 3.14 reports the position, velocity, acceleration and jerk for a double S trajectory when the constant velocity phase is present. The boundary conditions are

$$q_0 = 0, \quad q_1 = 10, \quad v_0 = 1, \quad v_1 = 0,$$

while the constraints are

$$v_{max} = 5, \quad a_{max} = 10, \quad j_{max} = 30.$$

The resulting time intervals are

$$T_a = 0.7333, \quad T_v = 1.1433, \quad T_d = 0.8333, \quad T_{j1} = 0.3333, \quad T_{j2} = 0.3333.$$

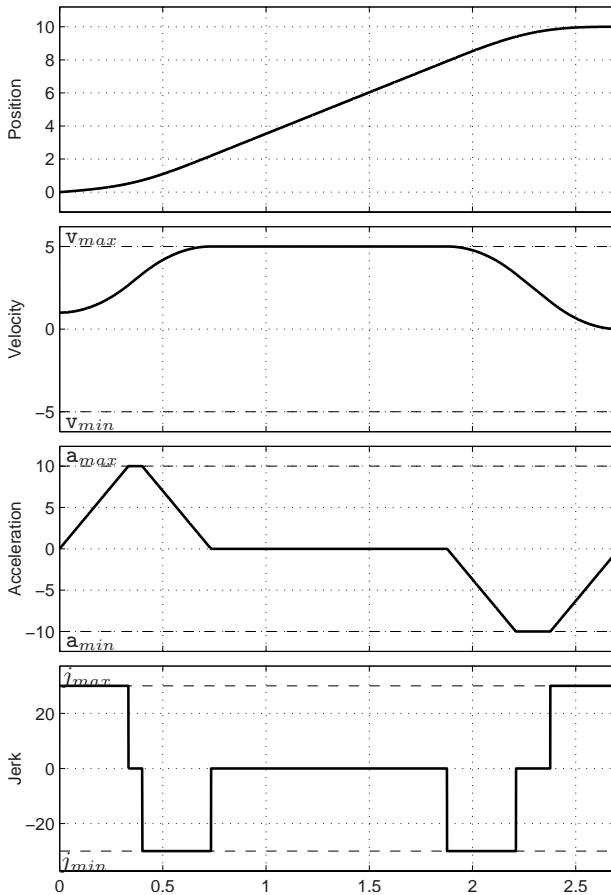
□

### Case 2: $v_{lim} < v_{max}$ .

In this case, the constant velocity segment is not present ( $T_v = 0$ ), and the duration of the acceleration and deceleration segments can be easily computed if the maximum/minimum accelerations are reached in both segments. In this case

$$T_{j1} = T_{j2} = T_j = \frac{a_{max}}{j_{max}} \quad (3.26a)$$

and



**Fig. 3.14.** Double S trajectory profiles (position, velocity, acceleration and jerk) including a constant velocity phase.

$$T_a = \frac{\frac{a_{max}^2}{j_{max}} - 2v_0 + \sqrt{\Delta}}{2a_{max}} \quad (3.26b)$$

$$T_d = \frac{\frac{a_{max}^2}{j_{max}} - 2v_1 + \sqrt{\Delta}}{2a_{max}} \quad (3.26c)$$

where

$$\Delta = \frac{a_{max}^4}{j_{max}^2} + 2(v_0^2 + v_1^2) + a_{max} \left( 4(q_1 - q_0) - 2 \frac{a_{max}}{j_{max}} (v_0 + v_1) \right). \quad (3.27)$$

**Example 3.10** Fig. 3.15 reports the position, velocity, acceleration and jerk for a double S trajectory when the constant velocity phase is not present. The

boundary conditions are

$$q_0 = 0, \quad q_1 = 10, \quad v_0 = 1, \quad v_1 = 0,$$

with constraints

$$v_{max} = 10, \quad a_{max} = 10, \quad j_{max} = 30.$$

The resulting time intervals are

$$T_a = 1.0747, \quad T_v = 0, \quad T_d = 1.1747, \quad T_{j1} = 0.3333, \quad T_{j2} = 0.3333.$$

and the maximum speed is  $v_{lim} = 8.4136$ .  $\square$

If  $T_a < 2T_j$  or  $T_d < 2T_j$ , then the maximum (minimum) acceleration is not reached in at least one of the two segments, and therefore it is not possible to use eq. (3.26a), (3.26b), (3.26c). In this case (indeed rather unusual), the determination of parameters is quite difficult, and it may be more convenient to find an approximated solution that, although not optimal, results acceptable from a computational point of view. A possible way to determine this solution is to progressively decrease the value of  $a_{max}$  (e.g. by assuming  $a_{max} = \gamma a_{max}$ , with  $0 < \gamma < 1$ ) and compute the durations of the segments by means of (3.26a), (3.26b), (3.26c), until the conditions  $T_a > 2T_j$  and  $T_d > 2T_j$  are both true.

**Example 3.11** Fig. 3.16 reports the position, velocity, acceleration and jerk of a double S trajectory when the constant velocity segment is not present. In this case, also the maximum acceleration is not reached and the above recursive algorithm is adopted. The boundary conditions are

$$q_0 = 0, \quad q_1 = 10, \quad v_0 = 7, \quad v_1 = 0,$$

while the constraints are

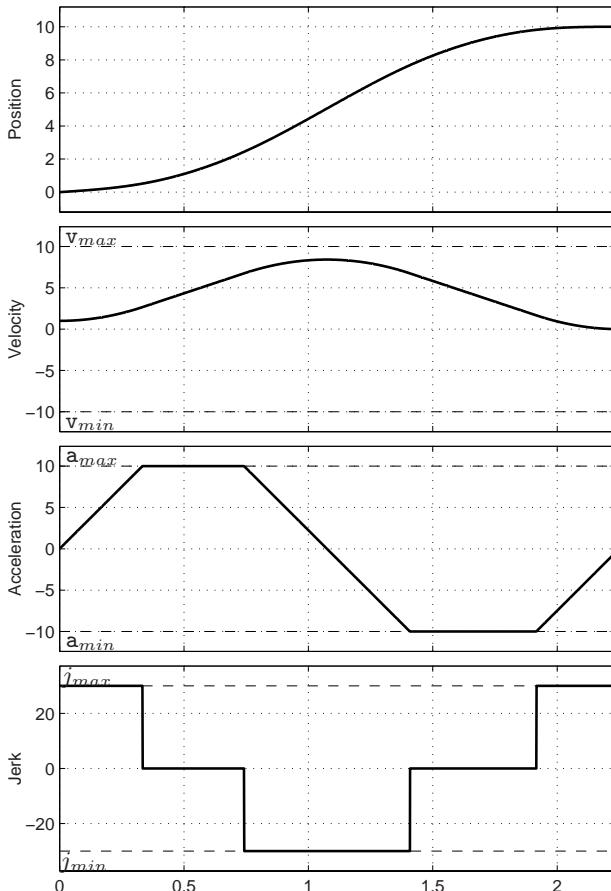
$$v_{max} = 10, \quad a_{max} = 10, \quad j_{max} = 30.$$

The time intervals defining the double S trajectory result

$$T_a = 0.4666, \quad T_v = 0, \quad T_d = 1.4718, \quad T_{j1} = 0.2321, \quad T_{j2} = 0.2321.$$

The maximum speed is  $v_{lim} = 8.6329$ , and the limit values of the acceleration during the acceleration and deceleration period are respectively  $a_{lim_a} = 6.9641$  and  $a_{lim_d} = -6.9641$ .  $\square$

During this recursive computation, it may happen that  $T_a$  or  $T_d$  becomes negative. In this case, only one of the acceleration or deceleration phase is necessary, depending on the values of the initial and final velocities. If  $T_a < 0$



**Fig. 3.15.** Double S trajectory profiles (position, velocity, acceleration and jerk) without a constant velocity phase.

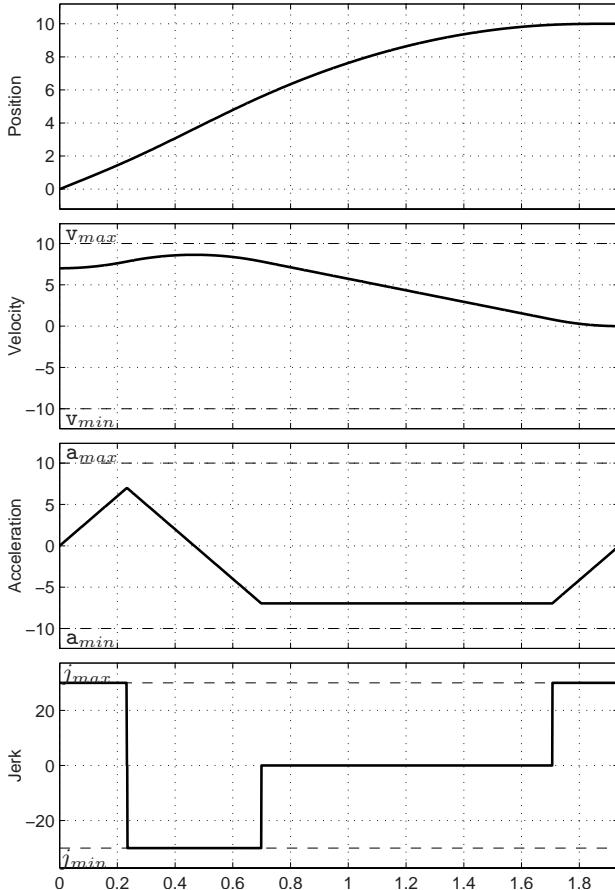
(note that in this case necessarily  $v_0 > v_1$ ), the acceleration phase is not present. Then,  $T_a$  is set to 0 and the duration of the deceleration segment can be computed according to

$$T_d = 2 \frac{q_1 - q_0}{v_1 + v_0} \quad (3.28a)$$

$$T_{j2} = \frac{j_{max}(q_1 - q_0) - \sqrt{j_{max}(j_{max}(q_1 - q_0)^2 + (v_1 + v_0)^2(v_1 - v_0))}}{j_{max}(v_1 + v_0)}. \quad (3.28b)$$

**Example 3.12** Fig. 3.17 reports position, velocity, acceleration and jerk of a double S trajectory composed only by the deceleration phase. The boundary conditions are

$$q_0 = 0, \quad q_1 = 10, \quad v_0 = 7.5, \quad v_1 = 0,$$



**Fig. 3.16.** Double S trajectory profiles (position, velocity, acceleration and jerk) without the constant velocity phase and with limit acceleration lower than the maximum value.

with constraints

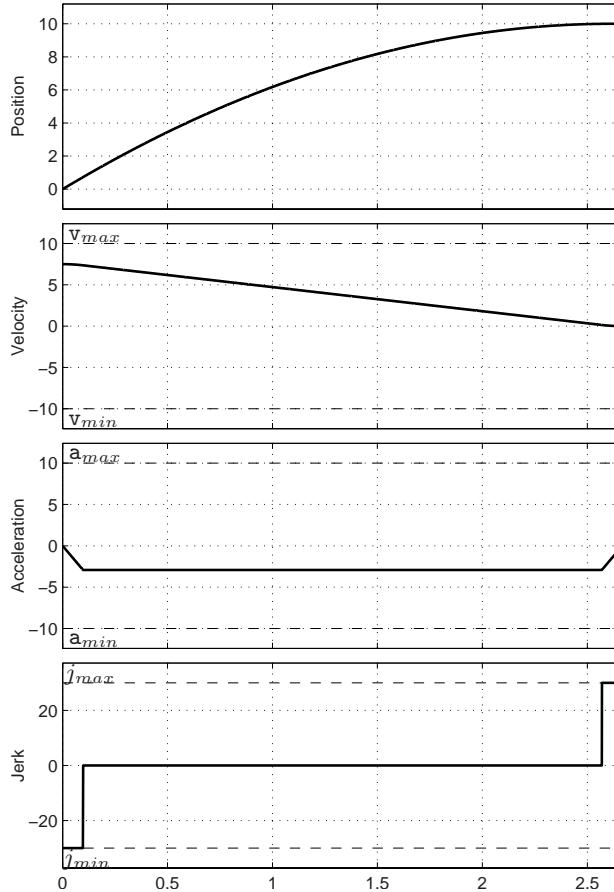
$$v_{max} = 10, \quad a_{max} = 10, \quad j_{max} = 30.$$

The resulting time intervals are

$$T_a = 0, \quad T_v = 0, \quad T_d = 2.6667, \quad T_{j1} = 0, \quad T_{j2} = 0.0973.$$

The maximum velocity is  $v_{lim} = 7.5$ , and the limit values of the acceleration during the acceleration and deceleration periods are respectively  $a_{lim_a} = 0$  and  $a_{lim_d} = -2.9190$ .  $\square$

In the dual case, i.e. when  $T_d < 0$  (this case is possible when  $v_1 > v_0$ ), the deceleration phase is not necessary ( $T_d = 0$ ) and the duration of the



**Fig. 3.17.** Double S trajectory profiles (position, velocity, acceleration and jerk) composed by only a deceleration phase.

acceleration period must be computed as

$$T_a = 2 \frac{q_1 - q_0}{v_1 + v_0} \quad (3.29a)$$

$$T_{j1} = \frac{j_{max}(q_1 - q_0) - \sqrt{j_{max}(j_{max}(q_1 - q_0)^2 - (v_1 + v_0)^2(v_1 - v_0))}}{j_{max}(v_1 + v_0)}. \quad (3.29b)$$

After the duration of each segment of the trajectory has been defined, it is possible to compute the values of the maximum/minimum accelerations ( $a_{lim_a}$  and  $a_{lim_d}$ ) and of the maximum velocity ( $v_{lim}$ ) of the trajectory:

$$a_{lim_a} = j_{max} T_{j1}, \quad a_{lim_d} = -j_{max} T_{j2}$$

$$v_{lim} = v_0 + (T_a - T_{j1})a_{lim_a} = v_1 - (T_d - T_{j2})a_{lim_d}.$$

### 3.4.1 Computation of the trajectory for $q_1 > q_0$

Once the time lengths and the other parameters have been defined, the double S trajectory is computed by means of the following equations (one for each segment defined in Fig. 3.13). The case  $t_0 = 0$  is assumed, otherwise a translation in time has to be applied, see Sec. 5.1.

#### Acceleration phase

a)  $t \in [0, T_{j1}]$

$$\begin{cases} q(t) &= q_0 + v_0 t + j_{max} \frac{t^3}{6} \\ \dot{q}(t) &= v_0 + j_{max} \frac{t^2}{2} \\ \ddot{q}(t) &= j_{max} t \\ q^{(3)}(t) &= j_{max} \end{cases} \quad (3.30a)$$

b)  $t \in [T_{j1}, T_a - T_{j1}]$

$$\begin{cases} q(t) &= q_0 + v_0 t + \frac{a_{lim_a}}{6} (3t^2 - 3T_{j1}t + T_{j1}^2) \\ \dot{q}(t) &= v_0 + a_{lim_a} \left( t - \frac{T_{j1}}{2} \right) \\ \ddot{q}(t) &= j_{max} T_{j1} = a_{lim_a} \\ q^{(3)}(t) &= 0 \end{cases} \quad (3.30b)$$

c)  $t \in [T_a - T_{j1}, T_a]$

$$\begin{cases} q(t) &= q_0 + (v_{lim} + v_0) \frac{T_a}{2} - v_{lim} (T_a - t) - j_{min} \frac{(T_a - t)^3}{6} \\ \dot{q}(t) &= v_{lim} + j_{min} \frac{(T_a - t)^2}{2} \\ \ddot{q}(t) &= -j_{min} (T_a - t) \\ q^{(3)}(t) &= j_{min} = -j_{max} \end{cases} \quad (3.30c)$$

#### Constant velocity phase

a)  $t \in [T_a, T_a + T_v]$

$$\begin{cases} q(t) = q_0 + (\mathbf{v}_{lim} + \mathbf{v}_0) \frac{T_a}{2} + \mathbf{v}_{lim}(t - T_a) \\ \dot{q}(t) = \mathbf{v}_{lim} \\ \ddot{q}(t) = 0 \\ q^{(3)}(t) = 0 \end{cases} \quad (3.30d)$$

**Deceleration phase**a)  $t \in [T - T_d, T - T_d + T_{j2}]$ 

$$\begin{cases} q(t) = q_1 - (\mathbf{v}_{lim} + \mathbf{v}_1) \frac{T_d}{2} + \mathbf{v}_{lim}(t - T + T_d) - \mathbf{j}_{max} \frac{(t - T + T_d)^3}{6} \\ \dot{q}(t) = \mathbf{v}_{lim} - \mathbf{j}_{max} \frac{(t - T + T_d)^2}{2} \\ \ddot{q}(t) = -\mathbf{j}_{max}(t - T + T_d) \\ q^{(3)}(t) = \mathbf{j}_{min} = -\mathbf{j}_{max} \end{cases} \quad (3.30e)$$

b)  $t \in [T - T_d + T_{j2}, T - T_{j2}]$ 

$$\begin{cases} q(t) = q_1 - (\mathbf{v}_{lim} + \mathbf{v}_1) \frac{T_d}{2} + \mathbf{v}_{lim}(t - T + T_d) + \\ + \frac{\mathbf{a}_{lim_d}}{6} \left( 3(t - T + T_d)^2 - 3T_{j2}(t - T + T_d) + T_{j2}^2 \right) \\ \dot{q}(t) = \mathbf{v}_{lim} + \mathbf{a}_{lim_d} \left( t - T + T_d - \frac{T_{j2}}{2} \right) \\ \ddot{q}(t) = -\mathbf{j}_{max} T_{j2} = \mathbf{a}_{lim_d} \\ q^{(3)}(t) = 0 \end{cases} \quad (3.30f)$$

c)  $t \in [T - T_{j2}, T]$ 

$$\begin{cases} q(t) = q_1 - \mathbf{v}_1(T - t) - \mathbf{j}_{max} \frac{(T - t)^3}{6} \\ \dot{q}(t) = \mathbf{v}_1 + \mathbf{j}_{max} \frac{(T - t)^2}{2} \\ \ddot{q}(t) = -\mathbf{j}_{max}(T - t) \\ q^{(3)}(t) = \mathbf{j}_{max} \end{cases} \quad (3.30g)$$

### 3.4.2 Computation of the trajectory for $q_1 < q_0$

In case  $q_1 < q_0$ , the parameters of the trajectory can be computed according to the same procedure reported above. It is necessary to consider the initial and final positions/velocities with opposite signs, and, after the computation, to invert the resulting profiles of position, velocity, acceleration, and jerk.

More generally, given any initial and final values for position and velocity ( $\hat{q}_0, \hat{q}_1, \hat{v}_0, \hat{v}_1$ ), in order to compute the trajectory it is necessary to transform these values as

$$q_0 = \sigma \hat{q}_0, \quad q_1 = \sigma \hat{q}_1, \quad v_0 = \sigma \hat{v}_0, \quad v_1 = \sigma \hat{v}_1 \quad (3.31)$$

where  $\sigma = \text{sign}(\hat{q}_0 - \hat{q}_1)$ . Similarly, also the constraints on maximum and minimum velocity, acceleration and jerk ( $\hat{v}_{max}, \hat{v}_{min}, \hat{a}_{max}, \hat{a}_{min}, \hat{j}_{max}, \hat{j}_{min}$ ) must be transformed:

$$\left\{ \begin{array}{l} v_{max} = \frac{(\sigma + 1)}{2} \hat{v}_{max} + \frac{(\sigma - 1)}{2} \hat{v}_{min} \\ v_{min} = \frac{(\sigma + 1)}{2} \hat{v}_{min} + \frac{(\sigma - 1)}{2} \hat{v}_{max} \\ a_{max} = \frac{(\sigma + 1)}{2} \hat{a}_{max} + \frac{(\sigma - 1)}{2} \hat{a}_{min} \\ a_{min} = \frac{(\sigma + 1)}{2} \hat{a}_{min} + \frac{(\sigma - 1)}{2} \hat{a}_{max} \\ j_{max} = \frac{(\sigma + 1)}{2} \hat{j}_{max} + \frac{(\sigma - 1)}{2} \hat{j}_{min} \\ j_{min} = \frac{(\sigma + 1)}{2} \hat{j}_{min} + \frac{(\sigma - 1)}{2} \hat{j}_{max}. \end{array} \right. \quad (3.32)$$

Finally, the computed profiles (i.e.  $q(t), \dot{q}(t), \ddot{q}(t), \dddot{q}(t)$ ) must be transformed again as

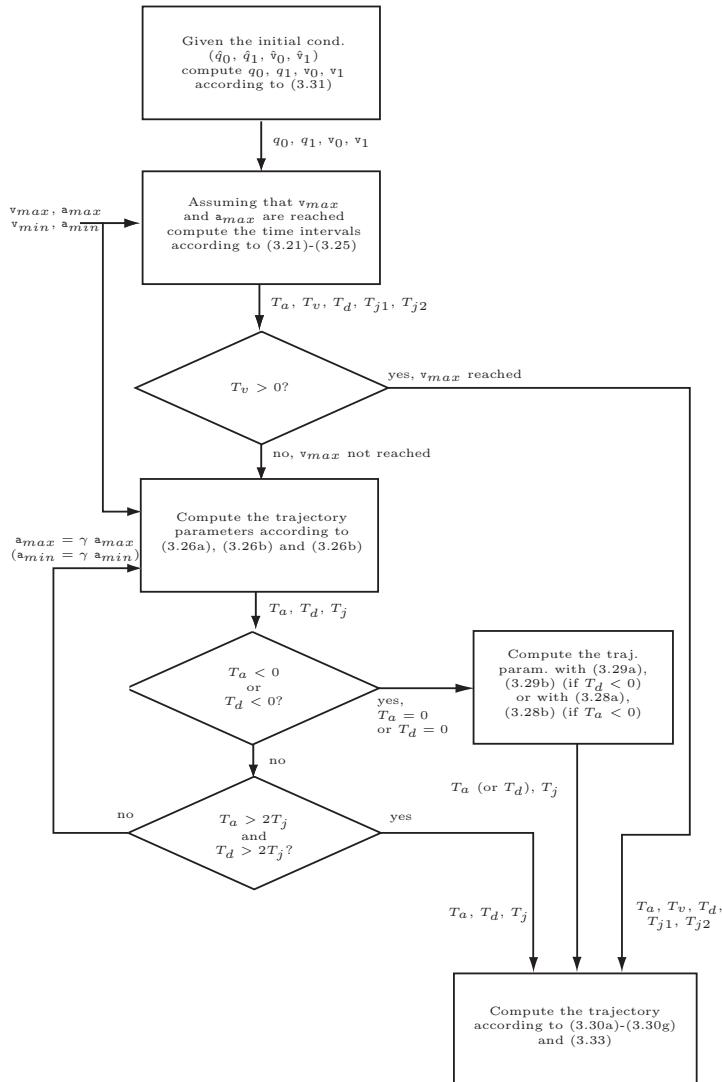
$$\left\{ \begin{array}{l} \hat{q}(t) = \sigma q(t) \\ \dot{\hat{q}}(t) = \sigma \dot{q}(t) \\ \ddot{\hat{q}}(t) = \sigma \ddot{q}(t) \\ \ddot{\hat{q}}(t) = \sigma \ddot{q}(t). \end{array} \right. \quad (3.33)$$

### Flux diagram for the double S computation

Since the synthesis of the double S trajectory is quite articulated, a scheme summarizing the algorithm to determine the trajectory in all possible conditions is shown in Fig. 3.18.

### 3.4.3 Double S with null initial and final velocities

When the initial and final velocities  $v_0$  and  $v_1$  are null, the computation of the double S trajectory is much simpler, in particular when the constraints are symmetric ( $j_{min} = -j_{max}$ ,  $a_{min} = -a_{max}$ ,  $v_{min} = -v_{max}$ ).



**Fig. 3.18.** Flux diagram for the double S trajectory computation.

As a matter of fact, in this case the acceleration and deceleration segments are symmetric, and then  $T_a = T_d$  and  $T_{j1} = T_{j2} = T_j$ . Moreover, it is always possible to find a trajectory joining the initial and the final position, which meets the constraints on velocity, acceleration and jerk.

Let us assume  $q_1 > q_0$  (otherwise consider Sec. 3.4.2). Four situations are possible:

**1.**  $v_{lim} = v_{max}$ :

**1.a.**  $a_{lim} = a_{max}$

**1.b.**  $a_{lim} < a_{max}$

**2.**  $v_{lim} < v_{max}$ :

**2.a.**  $a_{lim} = a_{max}$

**2.b.**  $a_{lim} < a_{max}$

where  $v_{lim}$  and  $a_{lim}$  are the maximum values of velocity and acceleration actually reached during the trajectory, i.e.  $v_{lim} = \max_t(\dot{q}(t))$  and  $a_{lim} = \max_t(\ddot{q}(t))$ .

**Case 1.**  $v_{lim} = v_{max}$ .

In this case, it is necessary to check if the maximum acceleration  $a_{max}$  is reached or not, and then compute  $T_j$  and  $T_a (= T_d)$

$$\begin{aligned} \text{a.} \quad \text{if } v_{max} j_{max} \geq a_{max}^2 \quad \Rightarrow \quad T_j &= \frac{a_{max}}{j_{max}} \\ &\quad T_a = T_j + \frac{v_{max}}{a_{max}} \\ \text{b.} \quad \text{if } v_{max} j_{max} < a_{max}^2 \quad \Rightarrow \quad T_j &= \sqrt{\frac{v_{max}}{j_{max}}} \\ &\quad T_a = 2T_j. \end{aligned}$$

Then, the duration of the constant velocity segment can be computed as

$$T_v = \frac{q_1 - q_0}{v_{max}} - T_a.$$

If  $T_v$  is positive, the maximum velocity is reached, otherwise it is necessary to consider Case 2 (and  $T_v = 0$ ).

**Case 2.**  $v_{lim} < v_{max}$ .

Again, two sub-cases are possible, depending whether the maximum acceleration  $a_{max}$  is reached or not. Also in this case the solution can be found in a closed form, as

$$\begin{aligned}
 \text{a.} \quad & \text{if } (q_1 - q_0) \geq 2 \frac{\mathbf{a}_{max}^3}{\mathbf{j}_{max}^2} \quad \Rightarrow \quad T_j = \frac{\mathbf{a}_{max}}{\mathbf{j}_{max}} \\
 & \quad T_a = \frac{T_j}{2} + \sqrt{\left(\frac{T_j}{2}\right)^2 + \frac{q_1 - q_0}{\mathbf{a}_{max}}} \\
 \text{b.} \quad & \text{if } (q_1 - q_0) < 2 \frac{\mathbf{a}_{max}^3}{\mathbf{j}_{max}^2} \quad \Rightarrow \quad T_j = \sqrt[3]{\frac{q_1 - q_0}{2\mathbf{j}_{max}}} \\
 & \quad T_a = 2T_j.
 \end{aligned}$$

Once  $T_j$ ,  $T_a$  (and  $T_d$ ),  $T_v$  are available, the trajectory can be evaluated according to (3.30a)-(3.30g), with

$$\begin{aligned}
 \mathbf{a}_{lim} &= \mathbf{j}_{max} T_j & = \mathbf{a}_{lim_a} = -\mathbf{a}_{lim_d} \\
 \mathbf{v}_{lim} &= (T_a - T_j) \mathbf{a}_{lim}.
 \end{aligned}$$

**Example 3.13** Fig. 3.19 shows the position, velocity, acceleration and jerk for a double S trajectory with zero initial and final velocities when the constant velocity phase is not present. In this case, also the maximum acceleration is not reached, however the trajectory parameters are computed in closed form. The boundary conditions are

$$q_0 = 0, \quad q_1 = 10, \quad \mathbf{v}_0 = 0, \quad \mathbf{v}_1 = 0,$$

while the constraints are

$$\mathbf{v}_{max} = 10, \quad \mathbf{a}_{max} = 20, \quad \mathbf{j}_{max} = 30.$$

The resulting time intervals are

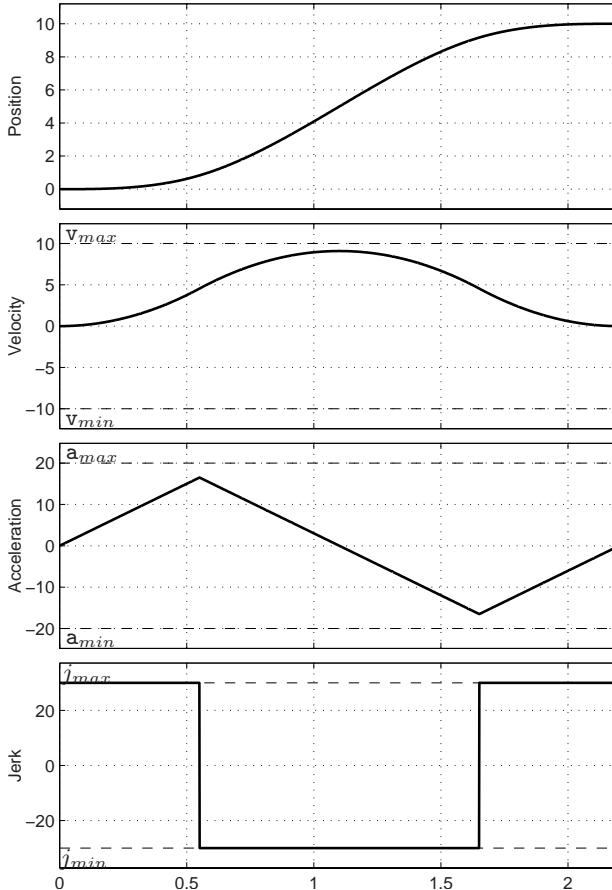
$$T_j = 0.5503, \quad T_a = 1.1006, \quad T_v = 0.$$

The maximum velocity is  $\mathbf{v}_{lim} = 8.6329$ , and the limit values of the acceleration during the acceleration and deceleration period are respectively  $\mathbf{a}_{lim_a} = 6.9641$  and  $\mathbf{a}_{lim_d} = -6.9641$ .  $\square$

### 3.4.4 On-line computation of the double S trajectory

A simplified approach for the computation of the double S profile is based on a discrete time formulation of the trajectory. This method is suitable when it is necessary to define complex trajectories, composed by several double S segments, and is appropriate for CNC machines, where the trajectory profiles are computed in discrete time.

Let us define

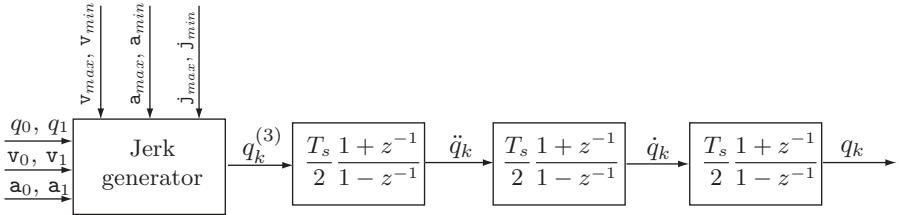


**Fig. 3.19.** Double S trajectory profiles with zero initial and final velocities, without a constant velocity phase and with limit acceleration lower than the maximum value.

$$\begin{cases} q(t = kT_s) &= q_k \\ \dot{q}(t = kT_s) &= \dot{q}_k \\ \ddot{q}(t = kT_s) &= \ddot{q}_k \\ q^{(3)}(t = kT_s) &= q_k^{(3)} \end{cases}$$

the values of position, velocity, acceleration and jerk at the  $k$ -th time instant, being  $T_s$  the sampling period. The structure of the trajectory planner is shown in Fig. 3.20. Given the initial and final values of position, velocity and acceleration and the constraints<sup>2</sup> ( $v_{max}$ ,  $v_{min}$ ,  $a_{max}$ ,  $a_{min}$ ,  $j_{max}$ ,  $j_{min}$ ), the jerk profile is computed as detailed below and then it is integrated three

<sup>2</sup> It is worth noticing that in this case the constraints can be freely chosen and it is not necessary to consider symmetric conditions, such as  $j_{min} = -j_{max}$ , etc.



**Fig. 3.20.** Block diagram of the trajectory planner for online computation of the double S trajectory.

times to obtain acceleration, velocity and position, respectively. In particular, the trapezoidal integration<sup>3</sup> is adopted, and accordingly the relations between jerk, acceleration, velocity and position are

$$\begin{aligned}\ddot{q}_k &= \ddot{q}_{k-1} + \frac{T_s}{2}(q_{k-1}^{(3)} + q_k^{(3)}) \\ \dot{q}_k &= \dot{q}_{k-1} + \frac{T_s}{2}(\ddot{q}_{k-1} + \ddot{q}_k) \\ q_k &= q_{k-1} + \frac{T_s}{2}(\dot{q}_{k-1} + \dot{q}_k).\end{aligned}\quad (3.34)$$

The basic idea of this trajectory planner is to perform, with acceleration and jerk compliant with the desired constraints, the acceleration phase and then the constant velocity segment until it is necessary to decelerate in order to reach the final position  $q_1$  with the desired values of velocity and acceleration  $v_1$  and  $a_1$ . Therefore, the computation of the trajectory is composed by two phases, see Fig. 3.21:

1. An acceleration profile is computed with the classical trapezoidal acceleration, possibly followed by a constant velocity phase ( $= v_{max}$ ).
2. During the motion, at each time instant  $kT_s$  it is checked whether it is possible to decelerate from the current velocity  $\dot{q}_k$  to the final one  $v_1$ , with the constraints on  $\ddot{q}_k$  and  $q_k^{(3)}$ , and with the goal to reach exactly  $q_1$ .

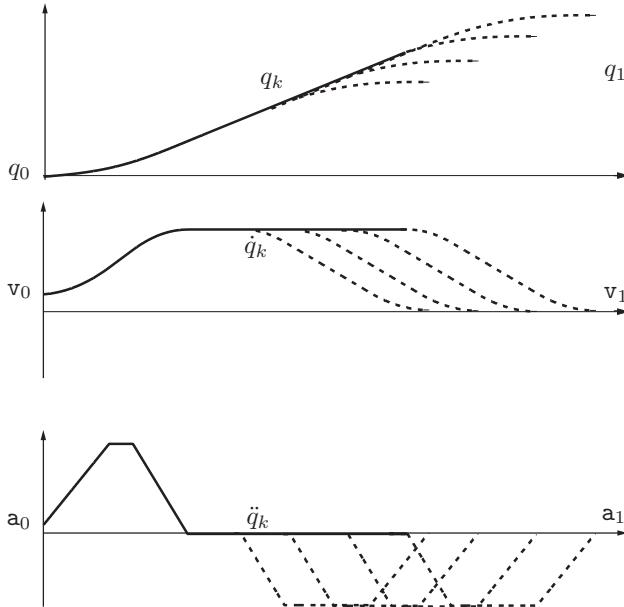
### Phase 1: Acceleration and constant velocity phase

In order to perform a double S trajectory from  $q_0$  to  $q_1$  ( $> q_0$ ), the jerk is kept at its maximum value until  $\ddot{q}_k < a_{max}$ . Then, the jerk is set to zero ( $q_k^{(3)} = 0$ ), and therefore the acceleration is constant ( $a_{max}$ ). Finally, the jerk is set to the minimum value ( $q_k^{(3)} = j_{min}$ ) in order to have a null acceleration when the maximum velocity is reached. At this point, the maximum velocity is maintained until the deceleration phase starts.

Mathematically, this can be described by

---

<sup>3</sup> The discrete time transfer function of an integrator is  $G_I(z^{-1}) = \frac{T_s}{2} \frac{1 + z^{-1}}{1 - z^{-1}}$ .



**Fig. 3.21.** Online computation of the double S trajectory.

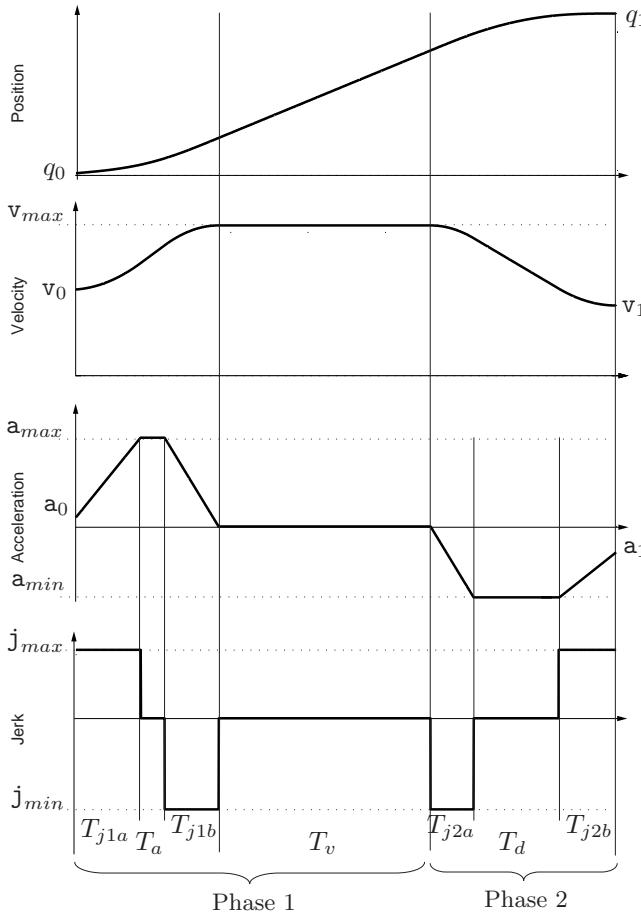
$$q_k^{(3)} = \begin{cases} j_{max}, & \text{if } \dot{q}_k - \frac{\ddot{q}_k^2}{2j_{min}} < v_{max} \text{ and } \ddot{q}_k < a_{max} \\ 0, & \text{if } \dot{q}_k - \frac{\ddot{q}_k^2}{2j_{min}} < v_{max} \text{ and } \ddot{q}_k \geq a_{max} \\ j_{min}, & \text{if } \dot{q}_k - \frac{\ddot{q}_k^2}{2j_{min}} \geq v_{max} \text{ and } \ddot{q}_k > 0 \\ 0, & \text{if } \dot{q}_k - \frac{\ddot{q}_k^2}{2j_{min}} \geq v_{max} \text{ and } \ddot{q}_k \leq 0. \end{cases} \quad (3.35)$$

The conditions  $\dot{q}_k - \frac{\ddot{q}_k^2}{2j_{min}} \geq v_{max}$ ,  $\ddot{q}_k \geq a_{max}$  and  $\ddot{q}_k \leq 0$  are considered in lieu of  $\dot{q}_k - \frac{\ddot{q}_k^2}{2j_{min}} = v_{max}$ ,  $\ddot{q}_k = a_{max}$  and  $\ddot{q}_k = 0$ , since usually small numerical errors affect the computations.

### Phase 2: Deceleration phase

At each sampling time<sup>4</sup>, one must compute the time intervals  $T_d$ ,  $T_{j2a}$  and  $T_{j2b}$  (refer to Fig. 3.22), necessary to change the acceleration and the velocity from the current values ( $\dot{q}_k$  and  $\ddot{q}_k$ ) to the final ones ( $v_1$  and  $a_1$ ) according to a trapezoidal deceleration profile, subject to constraints on maximum/minimim acceleration and jerk. From the expressions of velocity and acceleration varia-

<sup>4</sup> Being this a deceleration phase, it is expected that  $\dot{q}_k \geq v_1$ . Otherwise, the equations are not valid.



**Fig. 3.22.** Typical profiles for position, velocity, acceleration and jerk for the double S trajectory, with initial and final velocities and accelerations  $\neq 0$ .

tions, in the case that the minimum acceleration  $a_{min}$  is reached (accordingly  $T_d \geq T_{j2a} + T_{j2b}$ ), one obtains

$$\left\{ \begin{array}{l} T_{j2a} = \frac{a_{min} - \ddot{q}_k}{j_{min}} \\ T_{j2b} = \frac{a_0 - a_{min}}{j_{max}} \\ T_d = \frac{v_1 - \dot{q}_k}{a_{min}} + T_{j2a} \frac{a_{min} - \ddot{q}_k}{2a_{min}} + T_{j2b} \frac{a_{min} - a_1}{2a_{min}}. \end{array} \right. \quad (3.36)$$

Otherwise,

$$\begin{cases} T_{j2a} = -\frac{\ddot{q}_k}{j_{min}} + \frac{\sqrt{(j_{max} - j_{min})(\ddot{q}_k^2 j_{max} - j_{min}(a_1^2 + 2j_{max}(\dot{q}_k - v_1)))}}{j_{min}(j_{min} - j_{max})} \\ T_{j2b} = \frac{a_1}{j_{max}} + \frac{\sqrt{(j_{max} - j_{min})(\ddot{q}_k^2 j_{max} - j_{min}(a_1^2 + 2j_{max}(\dot{q}_k - v_1)))}}{j_{max}(j_{max} - j_{min})} \\ T_d = T_{j2a} + T_{j2b}. \end{cases} \quad (3.37)$$

Note that the periods with maximum and minimum jerk ( $T_{j2b}$  and  $T_{j2a}$ ) may be different since the initial and final accelerations of the deceleration phase,  $\ddot{q}_k$  and  $a_1$  respectively, are in general not equal (and in particular are not null).

At this point, it is necessary to compute the position displacement produced by the acceleration and velocity profiles obtained from (3.36) or (3.37)

$$h_k = \frac{1}{2}\ddot{q}_k T_d^2 + \frac{1}{6}(j_{min} T_{j2a}(3T_d^2 - 3T_d T_{j2a} + T_{j2a}^2) + j_{max} T_{j2b}^3) + T_d \dot{q}_k$$

and check if  $h_k < q_1 - q_k$ . If this condition holds, it is necessary to continue the trajectory computation according to Case 1 (iterating the calculation of the deceleration parameters with the new values of  $q_k$ ,  $\dot{q}_k$  and  $\ddot{q}_k$ ), otherwise the deceleration phase must start and the jerk is computed as

$$q_k^{(3)} = \begin{cases} j_{min}, & \text{if } (k - \bar{k}) \in \left[0, \frac{T_{j2a}}{T_s}\right] \\ 0, & \text{if } (k - \bar{k}) \in \left[\frac{T_{j2a}}{T_s}, \frac{T_d - T_{j2b}}{T_s}\right] \\ j_{max}, & \text{if } (k - \bar{k}) \in \left[\frac{T_d - T_{j2b}}{T_s}, \frac{T_d}{T_s}\right] \end{cases} \quad (3.38)$$

where  $\bar{k}$  is the time instant in which Phase 2 starts.

**Example 3.14** Fig. 3.23 reports the position, velocity, acceleration and jerk for a double S trajectory computed online. In this case, it is simple to consider non-null initial and final values of velocity and acceleration, and asymmetric constraints on the maximum values of velocity, acceleration and jerk. In particular, the boundary conditions are

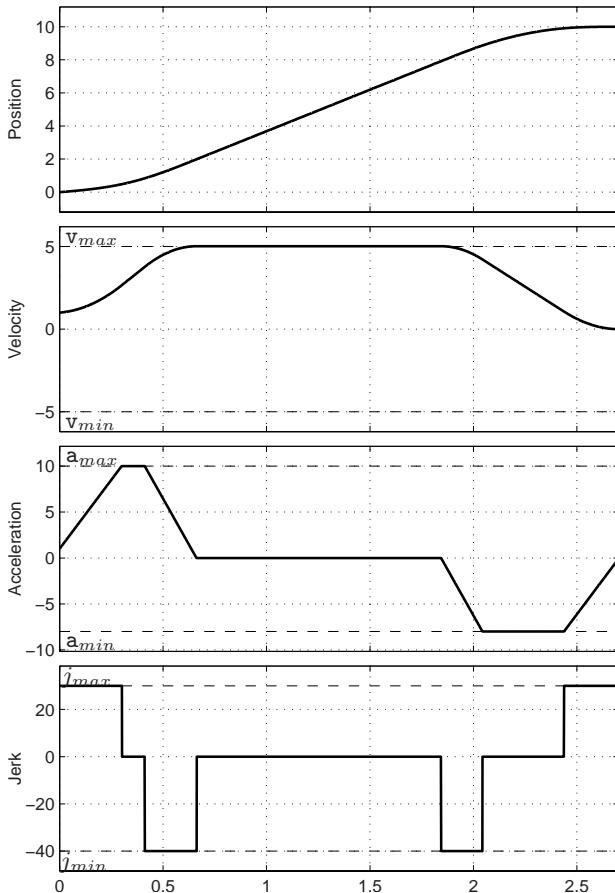
$$q_0 = 0, \quad q_1 = 10, \quad v_0 = 1, \quad v_1 = 0, \quad a_0 = 1, \quad a_1 = 0$$

while the constraints are

$$\begin{aligned} v_{max} &= 5, & a_{max} &= 10, & j_{max} &= 30, \\ v_{min} &= -5, & a_{min} &= -8, & j_{min} &= -40. \end{aligned}$$

The sampling period is  $T_s = 0.001s$ . □

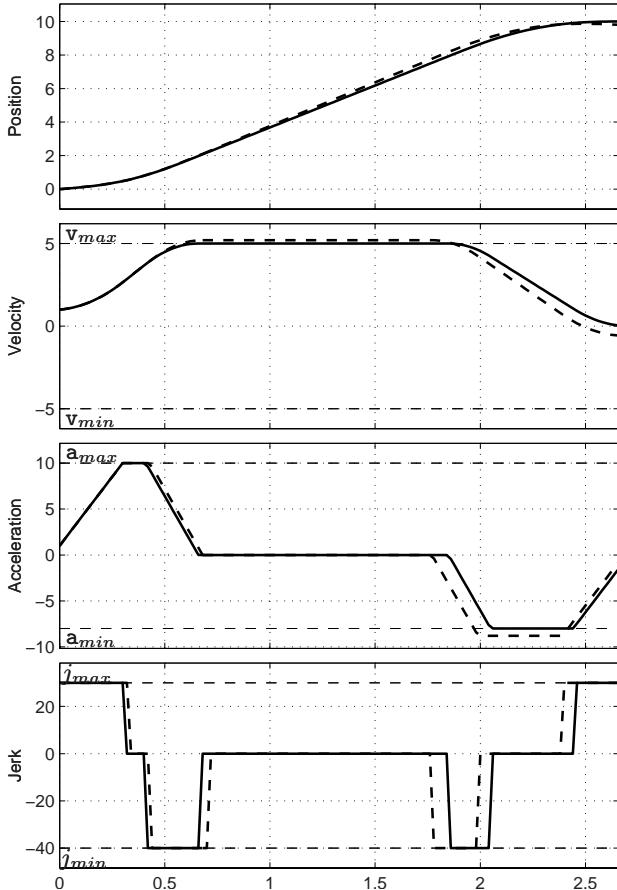
As already mentioned, this algorithm is affected by numerical errors which depend on the sampling period: the larger  $T_s$  is, the larger the errors are on



**Fig. 3.23.** Double S trajectory profiles (position, velocity, acceleration and jerk) computed online.

acceleration, velocity and position. However, if it is necessary for some reasons to have a large value for  $T_s$ , it is possible to assume, only for the computation of the trajectory, a sampling period  $N$  times smaller, e.g.  $T_s^* = \frac{T_s}{N}$ , and then under-sample the data points so obtained.

**Example 3.15** Fig. 3.24 reports the position, velocity, acceleration and jerk for a double S trajectory computed online, with  $T_s = 0.02s$ . The same values of boundary and peak conditions of the previous example are considered. The dashed lines have been obtained with  $T_s = 0.02s$ , while the solid lines have been computed with  $T_s^* = T_s/200 = 0.0001s$ , and then under-sampling the profiles by considering a point every two hundred samples. Note the errors that affect the dashed curves.  $\square$



**Fig. 3.24.** Double S trajectory profiles (position, velocity, acceleration and jerk) computed online, with  $T_s = 0.0001s$  (solid) and with  $T_s = 0.02s$  (dashed).

If  $q_1 < q_0$  it is possible to adopt the method described in Sec. 3.4.2.

**Example 3.16** Fig. 3.25 reports the position, velocity, acceleration and jerk for a double S trajectory computed online for  $q_1 < q_0$ . In particular, the boundary conditions are

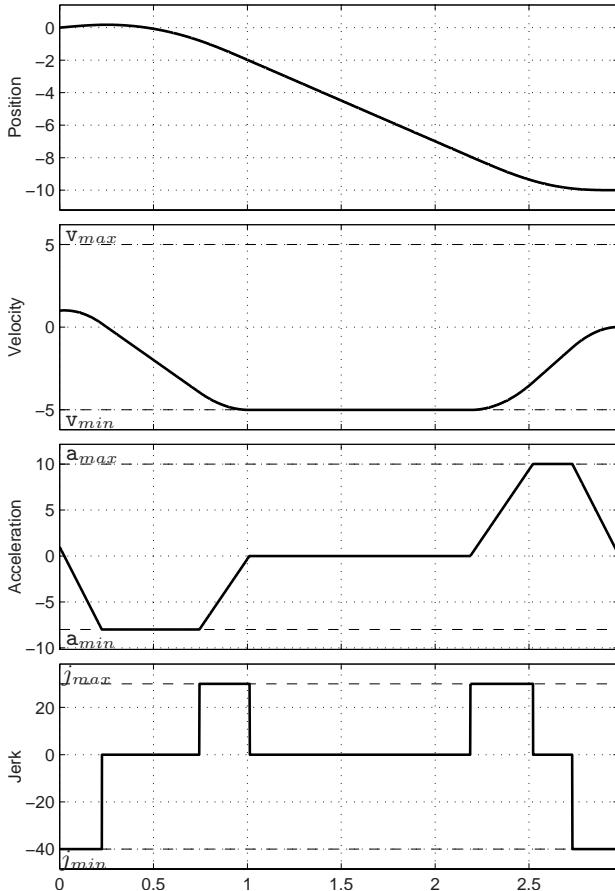
$$q_0 = 0, \quad q_1 = -10, \quad v_0 = 1, \quad v_1 = 0, \quad a_0 = 1, \quad a_1 = 0$$

with the constraints

$$\begin{aligned} v_{max} &= 5, & a_{max} &= 10, & j_{max} &= 30, \\ v_{min} &= -5, & a_{min} &= -8, & j_{min} &= -40. \end{aligned}$$

The sampling period is  $T_s = 0.001s$ .

□



**Fig. 3.25.** Double S trajectory profiles (position, velocity, acceleration and jerk) computed online with  $q_1 < q_0$ .

### 3.4.5 Displacement time of a double S trajectory

The computation of the displacement time of a double S trajectory is rather complex, because of the large number of different possible cases. For this reason, only some specific, but common, situations are considered here.

In particular, assuming  $h = q_1 - q_0 > 0$ , symmetric constraints  $v_{min} = -v_{max}$ ,  $a_{min} = -a_{max}$ ,  $j_{min} = -j_{max}$ , and that the maximum values of both acceleration and speed ( $a_{max}$  and  $v_{max}$ ) are reached, the total duration of the trajectory can be easily obtained as

$$T = \frac{h}{v_{max}} + \frac{T_a}{2} \left( 1 - \frac{v_0}{v_{max}} \right) + \frac{T_d}{2} \left( 1 - \frac{v_1}{v_{max}} \right) \quad (3.39)$$

with

$$T_a = \frac{a_{max}}{j_{max}} + \frac{v_{max} - v_0}{a_{max}}, \quad T_d = \frac{a_{max}}{j_{max}} + \frac{v_{max} - v_1}{a_{max}}.$$

If both initial and final speeds are zero, (3.39) becomes

$$T = \frac{h}{v_{max}} + \frac{v_{max}}{a_{max}} + \frac{a_{max}}{j_{max}}. \quad (3.40)$$

From eq. (3.40), it is straightforward to verify that the time length of the trajectory can be easily modified by properly scaling the values of  $v_{max}$ ,  $a_{max}$ ,  $j_{max}$ . As a matter of fact, if the new constraints

$$v'_{max} = \lambda v_{max}, \quad a'_{max} = \lambda^2 a_{max}, \quad j'_{max} = \lambda^3 j_{max}$$

are considered, the duration  $T'$  becomes

$$T' = \frac{T}{\lambda}$$

and therefore it is possible to compute the value of  $\lambda$  which leads to a desired duration  $T' = T_D$ :

$$\lambda = \frac{T}{T_D}.$$

The same considerations are valid also if the initial and final velocities are not null, but in this case it is necessary to scale also  $v_0$  and  $v_1$  (see eq. (3.39)):

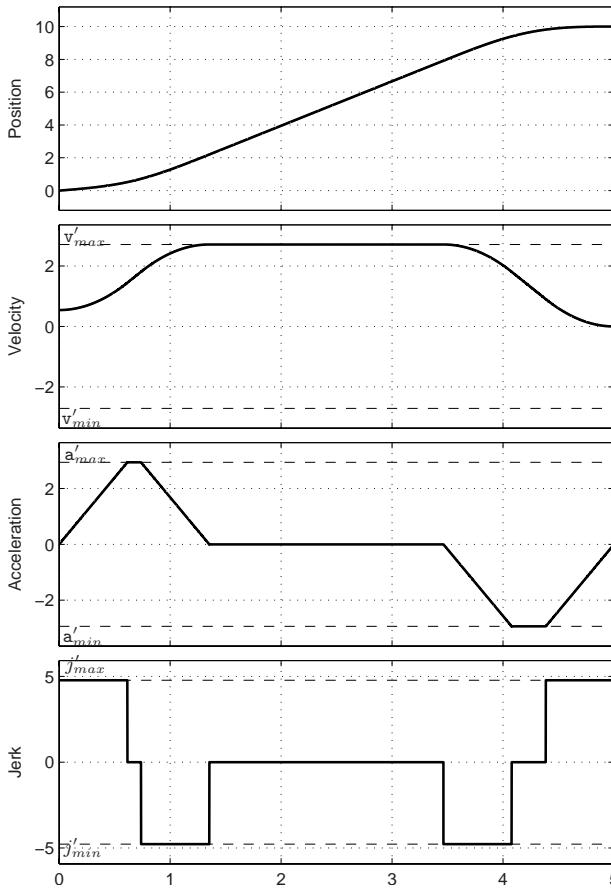
$$v'_0 = \lambda v_0, \quad v'_1 = \lambda v_1.$$

For other considerations about the scaling in time of trajectories, see Chapter 5.

**Example 3.17** In Fig. 3.26 the position, velocity, acceleration and jerk of a double S trajectory with a total duration  $T_D = 5$  s are shown. The boundary conditions and the constraints are the same of Example 3.9, which lead to the time length  $T = 2.71$  s. In order to modify the duration of the trajectory and to obtain the desired value  $T_D$ , the constraints and the initial and final velocities are scaled by  $\lambda = 0.542$ .  $\square$

### 3.4.6 Double S trajectory with assigned duration of the different phases

A general approach for planning a double S trajectory, with a given time length  $T$  and with specified durations of the acceleration and of the constant jerk phases, consists in defining the values of  $v_{max}$ ,  $a_{max}$ , and  $j_{max}$  as a function

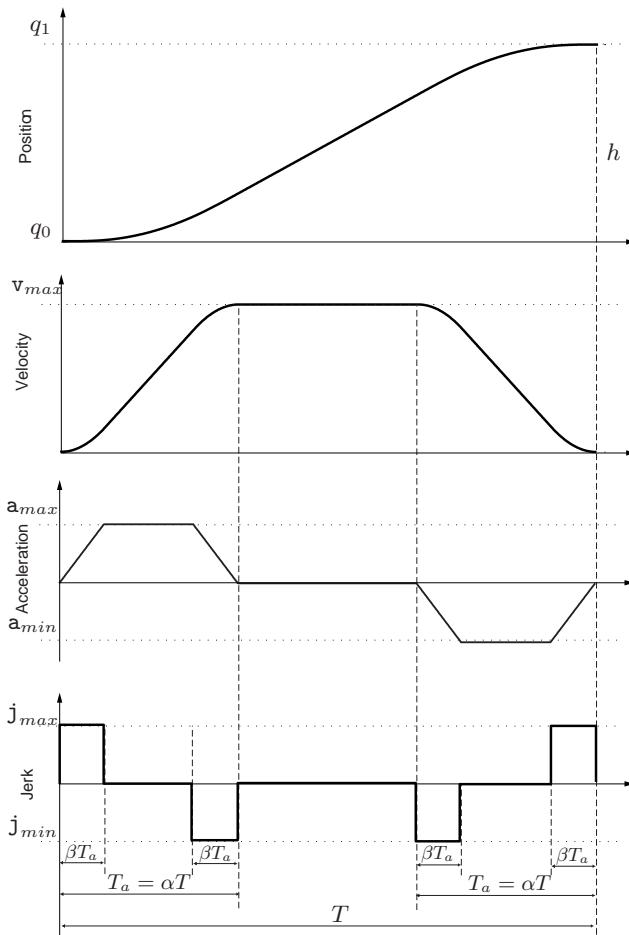


**Fig. 3.26.** Double S trajectory profiles (position, velocity, acceleration and jerk) properly scaled to impose a desired duration ( $T_D = 5$  s).

of the desired  $T$ ,  $T_a$ ,  $T_d$ ,  $T_j$ . In particular, the symmetric case with  $v_{min} = -v_{max}$ ,  $a_{min} = -a_{max}$ ,  $j_{min} = -j_{max}$  is considered, and the initial and final velocities  $v_0$ ,  $v_1$  are both assumed to be zero (therefore  $T_d = T_a$ ). Moreover, it is supposed that both the maximum speed and the maximum acceleration are reached. Therefore, with reference to equations (3.30a)-(3.30g) which define the trajectory profiles, it results

$$v_{lim} = v_{max}, \quad a_{lim_a} = a_{lim_d} = a_{max}.$$

From the expressions of the total duration, of the time length of acceleration phase and of the constant jerk segment, i.e.



**Fig. 3.27.** Double S trajectory profiles with zero initial and final velocities, and with desired durations of the single tracts.

$$\left\{ \begin{array}{l} T = \frac{h}{v_{max}} + T_a \\ T_a = \frac{v_{max}}{a_{max}} + T_j \\ T_j = \frac{a_{max}}{j_{max}} \end{array} \right. \quad (3.41)$$

it is possible to deduce the corresponding values of  $v_{max}$ ,  $a_{max}$ ,  $j_{max}$ :

$$\begin{cases} v_{max} = \frac{h}{T - T_a} \\ a_{max} = \frac{h}{(T - T_a)(T_a - T_j)} \\ j_{max} = \frac{h}{(T - T_a)(T_a - T_j)T_j}. \end{cases}$$

If one assumes that the acceleration period is a fraction of the entire trajectory duration:

$$T_a = \alpha T, \quad 0 < \alpha \leq 1/2$$

and, in a similar manner, that the time length of the constant jerk phase is a fraction of the acceleration period:

$$T_j = \beta T_a, \quad 0 < \beta \leq 1/2$$

then the values of the maximum speed, acceleration and jerk of the double S trajectory  $q(t)$  are obtained as:

$$\begin{cases} v_{max} = \frac{h}{(1 - \alpha)T} \\ a_{max} = \frac{h}{\alpha(1 - \alpha)(1 - \beta)T^2} \\ j_{max} = \frac{h}{\alpha^2 \beta (1 - \alpha)(1 - \beta)T^3}. \end{cases} \quad (3.42)$$

By substituting these values in (3.30a)-(3.30g), the trajectory with the given durations is defined.

**Example 3.18** A double S trajectory with the boundary conditions

$$q_0 = 0, \quad q_1 = 10, \quad v_0 = 0, \quad v_1 = 0,$$

is computed with the purpose of obtaining a total duration  $T = 5$ . The values

$$\alpha = 1/3, \quad \beta = 1/5$$

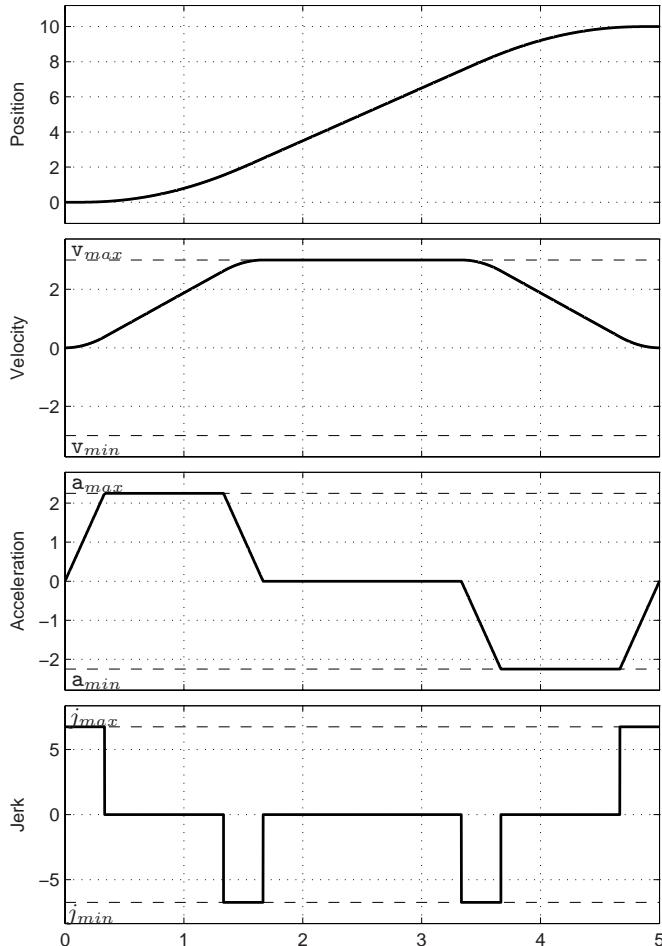
are considered, or equivalently

$$T_a = 1.6666, \quad T_j = 0.3333.$$

The resulting values of the velocity, acceleration and jerk are

$$v_{max} = 3.14, \quad a_{max} = 2.25, \quad j_{max} = 6.75.$$

They produce the trajectory reported in Fig. 3.28. □



**Fig. 3.28.** Double S trajectory profiles with prescribed duration  $T$ , and with the conditions  $T_a = T/3$  and  $T_j = T_a/5$ .

Obviously, the equations (3.41) which relate the variables of the double S trajectory ( $v_{max}$ ,  $a_{max}$ ,  $j_{max}$ ,  $h$ ,  $T$ ,  $T_a$ ,  $T_j$ ) can also be solved with respect to other sets of variables, e.g. ( $v_{max}$ ,  $a_{max}$ ,  $T_j$ ), or ( $v_{max}$ ,  $T_a$ ,  $T_j$ ), and so on, if the other terms are known. For instance, if one desires a trajectory with a total duration  $T$  and with a given maximum acceleration and jerk values  $a_{max}$ ,  $j_{max}$ , it is possible to obtain the remaining coefficients as

$$\begin{cases} v_{max} = \frac{-a_{max}^2 + a_{max} j_{max} T - \sqrt{a_{max}(-4hj_{max}^2 + a_{max}(a_{max} - j_{max}T)^2)}}{2j_{max}} \\ T_a = \frac{a_{max}^2 + a_{max} j_{max} T - \sqrt{a_{max}(-4hj_{max}^2 + a_{max}(a_{max} - j_{max}T)^2)}}{2a_{max}j_{max}} \\ T_j = \frac{a_{max}}{j_{max}}. \end{cases}$$

**Example 3.19** A double S trajectory with the boundary conditions

$$q_0 = 0, \quad q_1 = 10, \quad v_0 = 0, \quad v_1 = 0,$$

is computed with the purpose of obtaining a total duration  $T = 5$ . Moreover, the constraints

$$a_{max} = 2, \quad j_{max} = 8$$

are considered. The resulting values of the velocity, and of duration of the constant acceleration and jerk phases are

$$v_{max} = 3, \quad T_a = 1.82, \quad T_j = 0.25.$$

They define the trajectory reported in Fig. 3.29. □

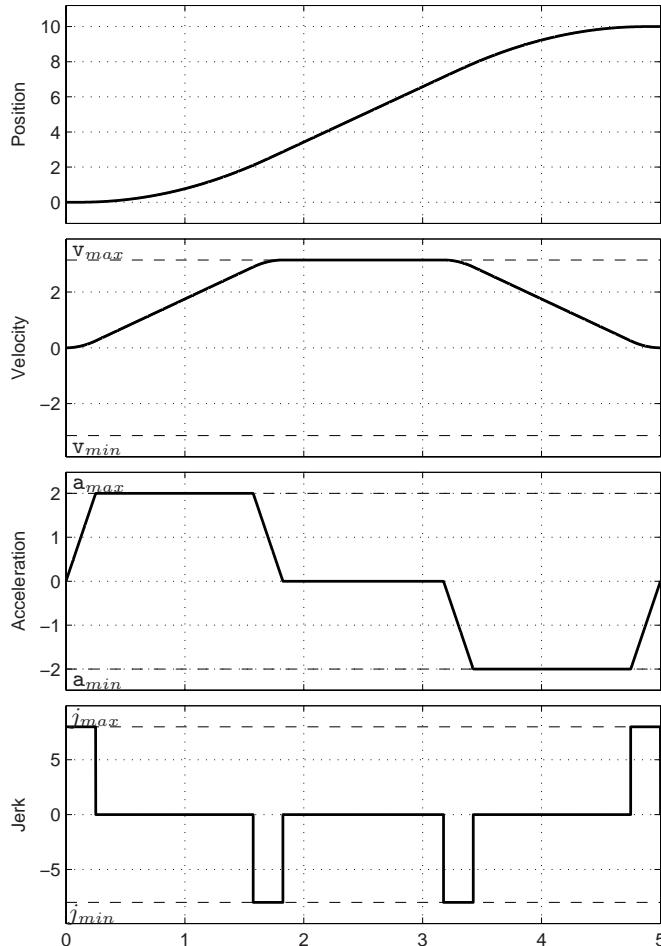
### 3.5 Fifteen Segments Trajectory

In some applications, it is necessary to adopt a trajectory with a continuous jerk. In this case, a variation of the double S trajectory can be used, characterized by a trapezoidal profile for the jerk. The overall trajectory is composed of fifteen segments (instead of the usual seven of the double S profile), in which the jerk increases or decreases linearly, or is constant. It is necessary to define the maximum rate of variation of the jerk by assuming a bound on its first derivative, called *jounce*, *ping*, or *snap*<sup>5</sup>, see Fig. 3.30.

In the standard case (with the conditions  $q_1 > q_0$  and  $s_{min} = -s_{max}$ ,  $j_{min} = -j_{max}$ ,  $a_{min} = -a_{max}$ ,  $v_{min} = -v_{max}$ ), in which the peak values of jerk, acceleration, velocity are reached, the trajectory is determined by defining the time length of the different segments, according to

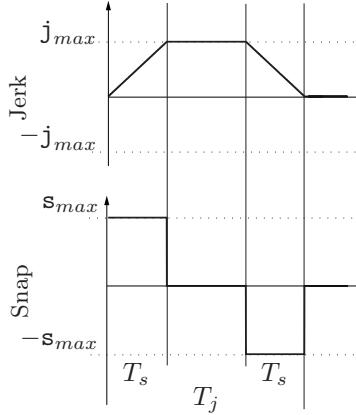
---

<sup>5</sup> There is not an official terminology yet; the term *snap* is used here.



**Fig. 3.29.** Double S trajectory profiles with prescribed duration  $T$ , and with conditions on  $a_{max}$  and  $j_{max}$ .

$$\left\{ \begin{array}{l} T_s = \frac{j_{max}}{s_{max}} \\ T_j = \frac{a_{max}}{j_{max}} + T_s \\ T_a = T_j + \frac{v_{max} - v_1}{a_{max}} \\ T_d = T_j + \frac{v_{max} - v_0}{a_{max}} \\ T_v = \frac{q_1 - q_0}{v_{max}} - \frac{T_a}{2} \left( 1 + \frac{v_0}{v_{max}} \right) - \frac{T_d}{2} \left( 1 + \frac{v_1}{v_{max}} \right) \end{array} \right. \quad (3.43)$$



**Fig. 3.30.** Continuous jerk profile in a 15-segments “double S” trajectory.

where  $s_{max}$ ,  $j_{max}$ ,  $a_{max}$ ,  $v_{max}$  are the maximum values of snap, jerk, acceleration, and velocity,  $q_0$ ,  $v_0$  and  $q_1$ ,  $v_1$  are the initial and final values of position and velocity, while the meaning of  $T_s$ ,  $T_j$ ,  $T_a$ ,  $T_d$ ,  $T_v$  is shown in Fig. 3.30 and Fig. 3.13. The expressions in eq. (3.43) are valid if the following inequalities hold:

$$T_j \geq 2T_s \Leftrightarrow a_{max} \geq \frac{j_{max}^2}{s_{max}} \quad (3.44)$$

$$T_a \geq 2T_j \Leftrightarrow (v_{max} - v_0) \geq \frac{a_{max}^2}{j_{max}} + \frac{a_{max}j_{max}}{s_{max}} \quad (3.45)$$

$$T_d \geq 2T_j \Leftrightarrow (v_{max} - v_1) \geq \frac{a_{max}^2}{j_{max}} + \frac{a_{max}j_{max}}{s_{max}} \quad (3.46)$$

$$T_v \geq 0 \Leftrightarrow \frac{q_1 - q_0}{v_{max}} - \frac{T_a}{2} \left( 1 + \frac{v_0}{v_{max}} \right) - \frac{T_d}{2} \left( 1 + \frac{v_1}{v_{max}} \right) \geq 0 \quad (3.47)$$

If eq. (3.44) does not hold, the maximum jerk is not reached; when (3.45) or (3.46) are not true, the acceleration (during the acceleration or deceleration phase) is smaller than the maximum allowed value; finally,  $T_v < 0$  means that the maximum velocity  $v_{max}$  is not reached.

If the durations  $T_s$ ,  $T_j$ ,  $T_a$ ,  $T_d$ ,  $T_v$  meet the conditions (3.44-3.47) (this means that all the tracts of the fifteen segments trajectory are present) and under the hypothesis that  $q_1 > q_0$  (otherwise the approach of Sec. 3.4.2 must be used), the trajectory can be evaluated according to the following system of equations, which defines for each tract the corresponding position, velocity,

acceleration, jerk and snap.

### Acceleration phase

1)  $t \in [0, T_s]$

$$\begin{cases} q^{(4)}(t) = s_{max} \\ q^{(3)}(t) = s_{max} t \\ \ddot{q}(t) = \frac{s_{max}}{2} t^2 \\ \dot{q}(t) = \frac{s_{max}}{6} t^3 + v_0 \\ q(t) = \frac{s_{max}}{24} t^4 + v_0 t + q_0. \end{cases} \quad (3.48a)$$

2)  $t \in [T_s, T_j - T_s]$

$$\begin{cases} q^{(4)}(t) = 0 \\ q^{(3)}(t) = j_{max} \\ \ddot{q}(t) = j_{max} t - \frac{1}{2} j_{max} T_s \\ \dot{q}(t) = \frac{j_{max}}{6} T_s^2 + \frac{1}{2} j_{max} t (t - T_s) + v_0 \\ q(t) = \frac{j_{max}}{24} (2t - T_s) (2t(t - T_s) + T_s^2) + v_0 t + q_0. \end{cases} \quad (3.48b)$$

3)  $t \in [T_j - T_s, T_j]$

$$\begin{cases} q^{(4)}(t) = -s_{max} \\ q^{(3)}(t) = -s_{max} (t - T_j) \\ \ddot{q}(t) = -\frac{s_{max}}{2} (t - T_j)^2 + a_{max} \\ \dot{q}(t) = \frac{s_{max}}{6} (7T_s^3 - 9T_s^2(t + T_s) + 3T_s(t + T_s)^2 - (t - T_j + T_s)^3) + v_0 \\ q(t) = \frac{s_{max}}{24} (-15T_s^4 + 28T_s^3(t + T_s) - 18T_s^2(t + T_s)^2 + 4T_s(t + T_s)^3 - (t - T_j + T_s)^4) + v_0 t + q_0. \end{cases} \quad (3.48c)$$

4)  $t \in [T_j, T_a - T_j]$ 

$$\left\{ \begin{array}{l} q^{(4)}(t) = 0 \\ q^{(3)}(t) = 0 \\ \ddot{q}(t) = \mathbf{a}_{max} \\ \dot{q}(t) = \frac{\mathbf{a}_{max}}{2} (2t - T_j) + v_0 \\ q(t) = \frac{\mathbf{a}_{max}}{12} (6t^2 - 6tT_j + 2T_j^2 - T_jT_s + T_s^2) + v_0 t + q_0. \end{array} \right. \quad (3.48d)$$

5)  $t \in [T_a - T_j, T_a - T_j + T_s]$ 

$$\left\{ \begin{array}{l} q^{(4)}(t) = -\mathbf{s}_{max} \\ q^{(3)}(t) = -\mathbf{s}_{max} (t - T_a + T_j) \\ \ddot{q}(t) = \mathbf{a}_{max} - \frac{\mathbf{s}_{max}}{2} (t - T_a + T_j)^2 \\ \dot{q}(t) = -\frac{\mathbf{s}_{max}}{6} (t - T_a + T_j)^3 + \frac{\mathbf{a}_{max}}{2} (2t - T_j) + v_0 \\ q(t) = -\frac{\mathbf{s}_{max}}{24} (t - T_a + T_j)^4 + \frac{\mathbf{a}_{max}}{12} (6t^2 - 6tT_j + 2T_j^2 - T_jT_s + T_s^2) + \\ \quad + v_0 t + q_0. \end{array} \right. \quad (3.48e)$$

6)  $t \in [T_a - T_j + T_s, T_a - T_s]$ 

$$\left\{ \begin{array}{l} q^{(4)}(t) = 0 \\ q^{(3)}(t) = -\mathbf{j}_{max} \\ \ddot{q}(t) = -\frac{\mathbf{j}_{max}}{2} (2t - 2T_a + T_s) \\ \dot{q}(t) = -\frac{\mathbf{j}_{max}}{6} (3(t - T_a)^2 - 6T_aT_j + 6T_j^2 + 3(t + T_a - 2T_j)T_s + T_s^2) + v_0 \\ q(t) = -\frac{\mathbf{j}_{max}}{24} (4(t - T_a)^3 - 12(2t - T_a)T_aT_j + 12(2t - T_a)T_j^2 + \\ \quad + 6(t^2 + 2t(T_a - 2T_j) - T_a(T_a - 2T_j))T_s + 4(t - T_a)T_s^2 + T_s^3) + \\ \quad + v_0 t + q_0. \end{array} \right. \quad (3.48f)$$

7)  $t \in [T_a - T_s, T_a]$ 

$$\begin{cases} q^{(4)}(t) = s_{max} \\ q^{(3)}(t) = s_{max}(t - T_a) \\ \ddot{q}(t) = \frac{s_{max}}{2}(t - T_a)^2 \\ \dot{q}(t) = \frac{s_{max}}{6}(t - T_a)^3 + a_{max}(T_a - T_j) + v_0 \\ q(t) = \frac{s_{max}}{24}(t - T_a)^4 + \frac{a_{max}}{2}(2(t - T_a)(T_a - T_j) + v_0 t + q_0. \end{cases} \quad (3.48g)$$

**Constant velocity phase**8)  $t \in [T_a, T_a + T_v]$ 

$$\begin{cases} q^{(4)}(t) = 0 \\ q^{(3)}(t) = 0 \\ \ddot{q}(t) = 0 \\ \dot{q}(t) = v_{max} \\ q(t) = \frac{(v_{max} - v_0)}{2}(2(t - T_a)) + v_0 t + q_0. \end{cases} \quad (3.48h)$$

**Deceleration phase**9)  $t \in [T_a + T_v, T_a + T_v + T_s]$ 

$$\begin{cases} q^{(4)}(t) = -s_{max} \\ q^{(3)}(t) = s_{max}((T - t) - T_d) \\ \ddot{q}(t) = -\frac{s_{max}}{2}((T - t) - T_d)^2 \\ \dot{q}(t) = \frac{s_{max}}{6}((T - t) - T_d)^3 + a_{max}(T_d - T_j) + v_1 \\ q(t) = -\frac{s_{max}}{24}((T - t) - T_d)^4 - \frac{a_{max}}{2}(2(T - t) - T_d)(T_d - T_j) - v_1(T - t) + q_1. \end{cases} \quad (3.48i)$$

$$10) \quad t \in [T_a + T_v + T_s, T_a + T_v + T_j - T_s]$$

$$\left\{ \begin{array}{l} q^{(4)}(t) = 0 \\ q^{(3)}(t) = -j_{max} \\ \ddot{q}(t) = \frac{j_{max}}{2} (2(T-t) - 2T_d + T_s) \\ \dot{q}(t) = -\frac{j_{max}}{6} \left( 3((T-t) - T_d)^2 - 6T_dT_j + 6T_j^2 + 3((T-t) + T_d - 2T_j)T_s + T_s^2 \right) + v_1 \\ q(t) = \frac{j_{max}}{24} \left( 4((T-t) - T_d)^3 - 12(2(T-t) - T_d)T_dT_j + 12(2(T-t) - T_d)T_j^2 + 6((T-t)^2 + 2(T-t)(T_d - 2T_j) - T_d(T_d - 2T_j))T_s + 4((T-t) - T_d)T_s^2 + T_s^3 \right) - v_1(T-t) + q_1. \end{array} \right. \quad (3.48j)$$

$$11) \quad t \in [T_a + T_v + T_j - T_s, T_a + T_v + T_j]$$

$$\left\{ \begin{array}{l} q^{(4)}(t) = s_{max} \\ q^{(3)}(t) = -s_{max}((T-t) - T_d + T_j) \\ \ddot{q}(t) = -a_{max} + \frac{s_{max}}{2} ((T-t) - T_d + T_j)^2 \\ \dot{q}(t) = -\frac{s_{max}}{6} ((T-t) - T_d + T_j)^3 + \frac{a_{max}}{2} (2(T-t) - T_j) + v_1 \\ q(t) = \frac{s_{max}}{24} ((T-t) - T_d + T_j)^4 - \frac{a_{max}}{12} (6(T-t)^2 - 6(T-t)T_j + 2T_j^2 - T_jT_s + T_s^2) - v_1(T-t) + q_1. \end{array} \right. \quad (3.48k)$$

$$12) \quad t \in [T_a + T_v + T_j, T - T_j]$$

$$\left\{ \begin{array}{l} q^{(4)}(t) = 0 \\ q^{(3)}(t) = 0 \\ \ddot{q}(t) = -a_{max} \\ \dot{q}(t) = \frac{a_{max}}{2} (2(T-t) - T_j) + v_1 \\ q(t) = -\frac{a_{max}}{12} (6(T-t)^2 - 6(T-t)T_j + 2T_j^2 - T_jT_s + T_s^2) - v_1(T-t) + q_1. \end{array} \right. \quad (3.48l)$$

13)  $t \in [T - T_j, T - T_j + T_s]$ 

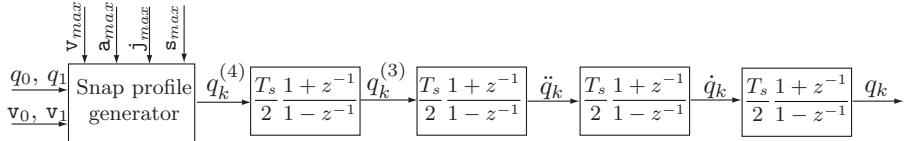
$$\left\{ \begin{array}{l} q^{(4)}(t) = \mathbf{s}_{max} \\ q^{(3)}(t) = -\mathbf{s}_{max} ((T-t) - T_j) \\ \ddot{q}(t) = \frac{\mathbf{s}_{max}}{2} ((T-t) - T_j)^2 - \mathbf{a}_{max} \\ \dot{q}(t) = \frac{\mathbf{s}_{max}}{6} \left( 7T_s^3 - 9T_s^2 ((T-t) + T_s) + 3T_s ((T-t) + T_s)^2 - ((T-t) - T_j + T_s)^3 \right) + \mathbf{v}_1 \\ q(t) = -\frac{\mathbf{s}_{max}}{24} \left( -15T_s^4 + 28T_s^3 ((T-t) + T_s) - 18T_s^2 ((T-t) + T_s)^2 + 4T_s ((T-t) + T_s)^3 - ((T-t) - T_j + T_s)^4 \right) - \mathbf{v}_1(T-t) + q_1. \end{array} \right. \quad (3.48m)$$

14)  $t \in [T - T_j + T_s, T - T_s]$ 

$$\left\{ \begin{array}{l} q^{(4)}(t) = 0 \\ q^{(3)}(t) = \mathbf{j}_{max} \\ \ddot{q}(t) = -\mathbf{j}_{max}(T-t) + \frac{\mathbf{j}_{max}}{2} T_s \\ \dot{q}(t) = \frac{\mathbf{j}_{max}}{6} T_s^2 + \frac{\mathbf{j}_{max}}{2} (T-t)((T-t) - T_s) + \mathbf{v}_1 \\ q(t) = -\frac{\mathbf{j}_{max}}{24} (2(T-t) - T_s)(2(T-t)((T-t) - T_s) + T_s^2) - \mathbf{v}_1(T-t) + q_1. \end{array} \right. \quad (3.48n)$$

15)  $t \in [T - T_s, T]$ 

$$\left\{ \begin{array}{l} q^{(4)}(t) = -\mathbf{s}_{max} \\ q^{(3)}(t) = \mathbf{s}_{max}(T-t) \\ \ddot{q}(t) = -\frac{\mathbf{s}_{max}}{2} (T-t)^2 \\ \dot{q}(t) = \frac{\mathbf{s}_{max}}{6} (T-t)^3 + \mathbf{v}_1 \\ q(t) = -\frac{\mathbf{s}_{max}}{24} (T-t)^4 - \mathbf{v}_1(T-t) + q_1. \end{array} \right. \quad (3.48o)$$



**Fig. 3.31.** Conceptual scheme for the computation of jerk, acceleration, velocity and position profiles of the fifteen segments trajectory starting from the jerk profile.

Alternatively, in order to obtain the profiles of jerk, acceleration, velocity and position one can numerically integrate the snap profiles, whose expression is particularly simple, i.e.

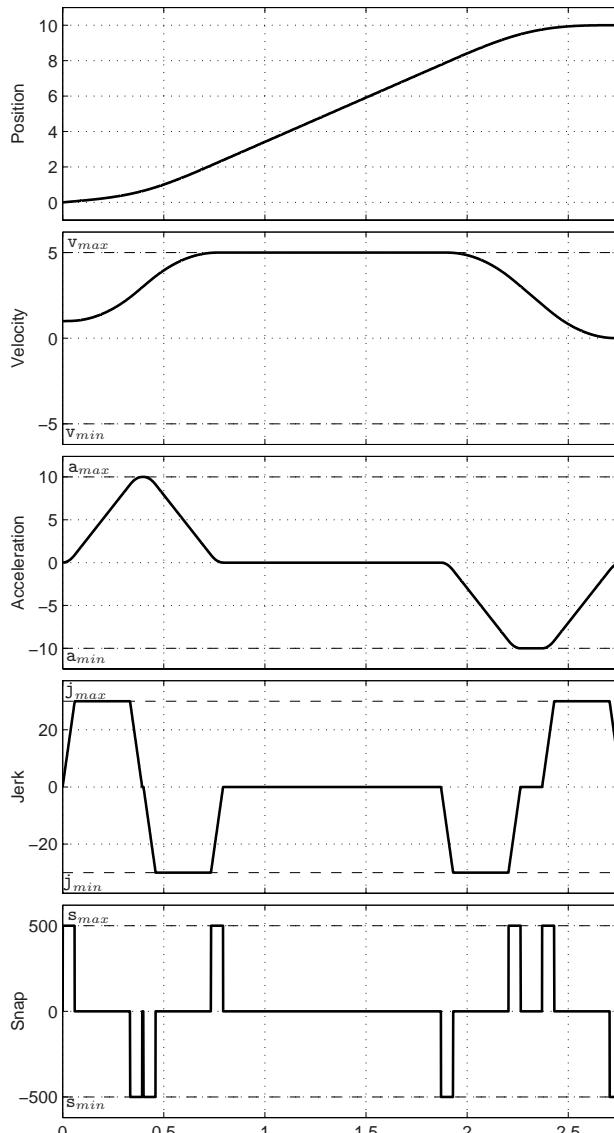
$$q^{(4)}(t) = \begin{cases} +s_{max} & \text{if } t \in [0, T_s] \\ 0 & \text{if } t \in [T_s, T_j - T_s] \\ -s_{max} & \text{if } t \in [T_j - T_s, T_j] \\ 0 & \text{if } t \in [T_j, T_a - T_j] \\ -s_{max} & \text{if } t \in [T_a - T_j, T_a - T_j + T_s] \\ 0 & \text{if } t \in [T_a - T_j + T_s, T_a - T_s] \\ +s_{max} & \text{if } t \in [T_a - T_s, T_a] \\ 0 & \text{if } t \in [T_a, T_a + T_v] \\ -s_{max} & \text{if } t \in [T_a + T_v, T_a + T_v + T_s] \\ 0 & \text{if } t \in [T_a + T_v + T_s, T_a + T_v + T_j - T_s] \\ +s_{max} & \text{if } t \in [T_a + T_v + T_j - T_s, T_a + T_v + T_j] \\ 0 & \text{if } t \in [T_a + T_v + T_j, T - T_j] \\ +s_{max} & \text{if } t \in [T - T_j, T - T_j + T_s] \\ 0 & \text{if } t \in [T - T_j + T_s, T - T_s] \\ -s_{max} & \text{if } t \in [T - T_s, T] \end{cases}$$

where  $T$  is the total duration of the trajectory. Unfortunately this approach, although conceptually quite simple (see Fig. 3.31), may generate numerical problems and acceleration/velocity/position drifts.

**Example 3.20** Fig. 3.32 reports the position, velocity, acceleration, jerk and snap profiles of a fifteen segments trajectory with the same constraints and boundary conditions as in Example 3.9, concerning a double S motion:  $v_{max} = 5$ ,  $a_{max} = 10$ ,  $j_{max} = 30$ ,  $s_{max} = 500$ ,  $q_0 = 0$ ,  $q_1 = 10$ ,  $v_0 = 1$ ,  $v_1 = 0$ . The resulting time intervals are

$$T_a = 0.7933, \quad T_v = 1.0773, \quad T_d = 0.8933, \quad T_j = 0.3933, \quad T_s = 0.0600$$

and the total duration results  $T = 2.7640$ , that is 2% more than the duration of the equivalent double S trajectory.  $\square$



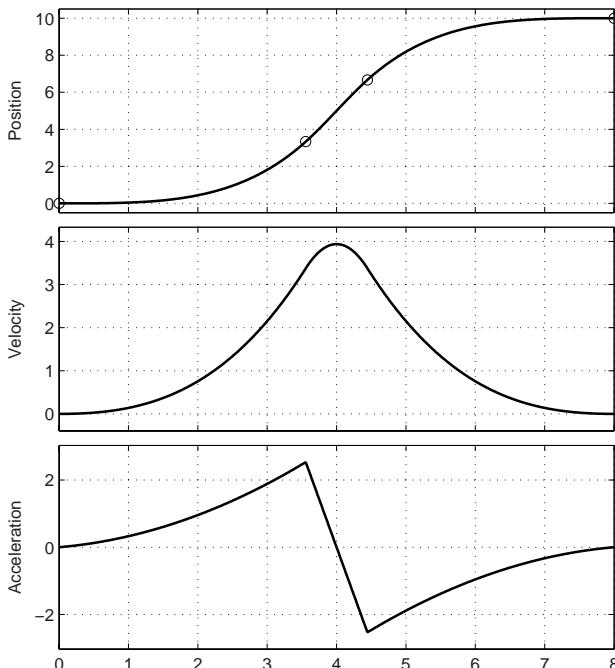
**Fig. 3.32.** Fifteen segments trajectory (position, velocity, acceleration, jerk and snap).

### 3.6 Piecewise Polynomial Trajectory

In particular applications, it may be convenient to define a trajectory as a composition of polynomial segments [17]. In these cases, in order to compute the trajectory it is necessary to define an adequate number of conditions (boundary conditions, point crossing, continuity of velocity, acceleration, ...). This type of approach has been already used in Sec. 3.2 and Sec. 3.3 for the computation of linear trajectories (first degree polynomials) with second or higher degree polynomials blends, and in Sec. 3.4 for double S trajectories.

For example, in *pick-and-place* operations by an industrial robot it may be of interest to have motions with very smooth initial and final phases. In such a case, one can use a motion profile obtained as the connection of three polynomials  $q_l(t)$ ,  $q_t(t)$ ,  $q_s(t)$  (i.e. *lift-off*, *travel*, *set-down*) with (for example):

$$\begin{aligned} q_l(t) &\implies \text{4-th degree polynomial} \\ q_t(t) &\implies \text{3-rd degree polynomial} \\ q_s(t) &\implies \text{4-th degree polynomial} \end{aligned}$$



**Fig. 3.33.** Position, velocity and acceleration profiles for a ‘4-3-4’ trajectory; the circles in the position diagram refer to the inner points  $(t_a, q_a), (t_b, q_b)$ .

This trajectory, known as *trajectory 4-3-4* (see Fig. 3.33), is computed by assigning  $5 + 4 + 5 = 14$  parameters, and therefore 14 conditions must be defined. If  $t_0, t_1$  are the initial and final instants,  $t_a, t_b$  the “switching” instants between the polynomial segments, and  $q_0, q_a, q_b, q_1$  the relative position values, the conditions necessary to compute the parameters are

$$\left. \begin{array}{l} q_l(t_0) = q_0, \\ q_t(t_b) = q_s(t_b) = q_b, \end{array} \right. \quad \left. \begin{array}{l} q_l(t_a) = q_t(t_a) = q_a \\ q_s(t_1) = q_1 \end{array} \right\} 6 \text{ crossing conditions}$$

$$\dot{q}_l(t_0) = \dot{q}_s(t_1) = 0, \quad \ddot{q}_l(t_0) = \ddot{q}_s(t_1) = 0 \quad 4 \text{ initial and final conditions}$$

$$\left. \begin{array}{l} \dot{q}_l(t_a) = \dot{q}_t(t_a), \\ \dot{q}_t(t_b) = \dot{q}_s(t_b), \end{array} \right. \quad \left. \begin{array}{l} \ddot{q}_l(t_a) = \ddot{q}_t(t_a) \\ \ddot{q}_t(t_b) = \ddot{q}_s(t_b) \end{array} \right\} 4 \text{ continuity conditions for velocity and acceleration}$$

It is convenient to express each segment of the trajectory in parametric form as a function of a normalized variable  $\tau$ :

$$q_l(t) = \tilde{q}_l(\tau) \Big|_{\tau = \frac{t-t_0}{T_l}}, \quad q_t(t) = \tilde{q}_t(\tau) \Big|_{\tau = \frac{t-t_a}{T_t}}, \quad q_s(t) = \tilde{q}_s(\tau) \Big|_{\tau = \frac{t-t_b}{T_s}}$$

where

$$T_l = t_a - t_0, \quad T_t = t_b - t_a, \quad T_s = t_1 - t_b.$$

Therefore, the trajectory is defined by

$$\left\{ \begin{array}{l} \tilde{q}_l(\tau) = a_{4l}\tau^4 + a_{3l}\tau^3 + a_{2l}\tau^2 + a_{1l}\tau + a_{0l}, \\ \tilde{q}_t(\tau) = a_{3t}\tau^3 + a_{2t}\tau^2 + a_{1t}\tau + a_{0t}, \\ \tilde{q}_s(\tau) = a_{4s}(\tau-1)^4 + a_{3s}(\tau-1)^3 + a_{2s}(\tau-1)^2 + a_{1s}(\tau-1) + a_{0s}, \end{array} \right. \quad \left. \begin{array}{l} \tau = \frac{t-t_0}{T_l} \\ \tau = \frac{t-t_a}{T_t} \\ \tau = \frac{t-t_b}{T_s} \end{array} \right.$$

From the conditions on initial and final positions, velocities and accelerations, one obtains the 7 parameters

$$\begin{aligned} a_{0l} &= q_0, & a_{1l} &= 0, & a_{2l} &= 0 \\ a_{0t} &= q_a & & & & \\ a_{0s} &= q_1, & a_{1s} &= 0, & a_{2s} &= 0. \end{aligned}$$

From the continuity conditions on position, velocity and acceleration, one obtains

$$\begin{aligned}
a_{4l} + a_{3l} &= (q_a - q_0) \\
a_{3t} + a_{2t} + a_{1t} &= (q_b - q_a) \\
-a_{4s} + a_{3s} &= (q_1 - q_b) \\
(4a_{4l} + 3a_{3l})/T_l &= a_{1t}/T_t \\
(12a_{4l} + 6a_{3l})/T_l^2 &= 2a_{2t}/T_t^2 \\
(3a_{3t} + 2a_{2t} + a_{1t})/T_t &= (-4a_{4s} + 3a_{3s})/T_s \\
(6a_{3t} + 2a_{2t})/T_t^2 &= (12a_{4s} - 6a_{3s})/T_s^2.
\end{aligned}$$

The remaining parameters can be computed from these 7 equations. Their analytical expressions are reported in Appendix A.

Note that besides the initial and final points, also the intermediate points (and the relative time instants) must be specified. Obviously, this technique can be applied also in case of non-null initial and final velocities and accelerations.

This method can be considered as an example of computation of piecewise polynomial trajectories. In case of a motion with  $m$  segments, each one defined by a polynomial function of degree  $p_k$ ,  $k = 1, \dots, m$ , a total of  $m + \sum_{k=1}^m p_k$  conditions must be assigned for the computation of the unknown parameters  $a_{jk}$ . Further considerations concerning the composition of trajectories will be given in Sec. 5.1, showing that in simple cases trajectories can be defined on the basis of elementary functions and simple geometrical operations.

### 3.7 Modified Trapezoidal Trajectory

This trajectory, based on the cycloidal trajectory, Sec. 2.2.2, and generating profiles similar to the double S, Sec. 3.4, can be considered an improvement of the trapezoidal velocity trajectory presented in Sec. 3.2 because of the continuity of the acceleration profile. In this case, see Fig. 3.34(a), the trajectory is subdivided into six parts: the second and the fifth segments are defined by second degree polynomials, while the remaining ones are expressed by cycloidal functions.

In the segment between point  $A = (t_a, q_a)$  and point  $B = (t_b, q_b)$  the acceleration has a sinusoidal profile, between point  $B$  and point  $C = (t_c, q_c)$  the acceleration is constant, while between point  $C$  and point  $D = (t_d, q_d)$  the acceleration decreases sinusoidally to zero. After point  $D$ , a deceleration phase similar to the acceleration one takes place with a specular profile.

Let us define  $T = t_1 - t_0$  and  $h = q_1 - q_0$ , considering for the sake of simplicity  $t_0 = 0$  and  $q_0 = 0$  and the temporal length of each segment as shown in Fig. 3.34(b). The trajectory between points  $A$  and  $B$  is described by a cycloidal function, see eq. (2.22),

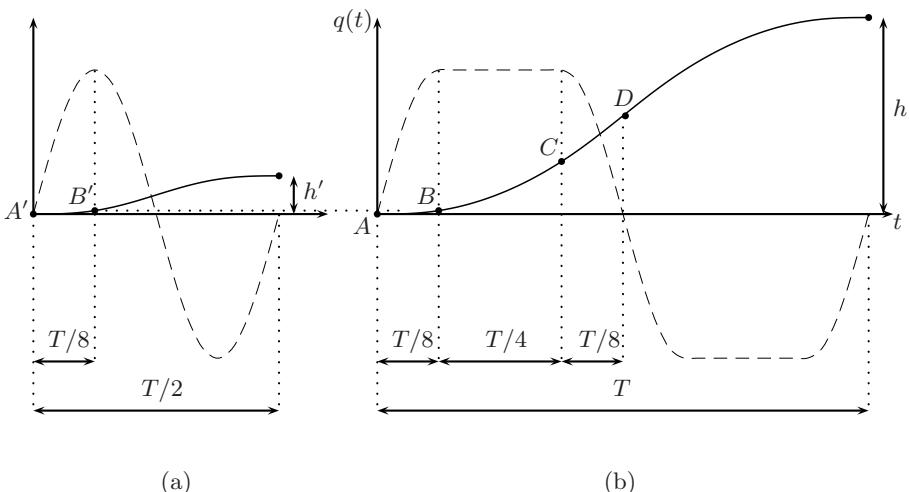
$$\begin{cases} q(t) = h' \left( \frac{2t}{T} - \frac{1}{2\pi} \sin \frac{4\pi t}{T} \right) \\ \dot{q}(t) = \frac{h'}{T} \left( 2 - 2 \cos \frac{4\pi t}{T} \right) \\ \ddot{q}(t) = \frac{8\pi h'}{T^2} \sin \frac{4\pi t}{T} \end{cases} \quad (3.49)$$

with the displacement  $h'$  defined below and duration  $T/2$ , see Fig. 3.34(a). The position  $q_b$  is reached at  $t_b = \frac{T}{8}$ . By substituting this value in (3.49), the position, velocity and acceleration in B are

$$\begin{cases} q_b = h' \left( \frac{1}{4} - \frac{1}{2\pi} \right) \\ \dot{q}_b = \frac{2h'}{T} \\ \ddot{q}_b = \frac{8\pi h'}{T^2}. \end{cases}$$

The expression of the trajectory between B and C is

$$\begin{cases} q(t) = q_b + v_c \left( t - \frac{T}{8} \right) + \frac{1}{2} a_c \left( t - \frac{T}{8} \right)^2 \\ \dot{q}(t) = v_c + a_c \left( t - \frac{T}{8} \right) \\ \ddot{q}(t) = a_c. \end{cases}$$



**Fig. 3.34.** Acceleration and position profiles for a cycloidal trajectory (a) and a modified trapezoidal trajectory (b).

For continuity reasons in point B

$$v_c = \frac{2h'}{T}, \quad a_c = \frac{8\pi h'}{T^2}.$$

Then, the expression of the trajectory between points B and C is

$$\begin{cases} q(t) = h' \left( \frac{1}{4} - \frac{1}{2\pi} \right) + \frac{2h'}{T} \left( t - \frac{T}{8} \right) + \frac{4\pi h'}{T^2} \left( t - \frac{T}{8} \right)^2 \\ \dot{q}(t) = \frac{2h'}{T} + \frac{8\pi h'}{T^2} \left( t - \frac{T}{8} \right) \\ \ddot{q}(t) = \frac{8\pi h'}{T^2}. \end{cases} \quad (3.50)$$

The position  $q_c$  is reached at  $t_c = \frac{3}{8}T$ . By using this value in (3.50) one gets

$$q_c = h' \left( \frac{3}{4} + \frac{\pi}{4} - \frac{1}{2\pi} \right).$$

The trajectory between C and D is given by

$$\begin{cases} q(t) = q_c + c_1 + c_2 \frac{t - \frac{3}{8}T}{T} + c_3 \sin \left( 4\pi \frac{t - \frac{T}{4}}{T} \right) \\ \dot{q}(t) = \frac{c_2}{T} + c_3 \frac{4\pi}{T} \cos \left( 4\pi \frac{t - \frac{T}{4}}{T} \right) \\ \ddot{q}(t) = -c_3 \frac{16\pi^2}{T^2} \sin \left( 4\pi \frac{t - \frac{T}{4}}{T} \right). \end{cases} \quad (3.51)$$

The parameters  $c_1$ ,  $c_2$ , and  $c_3$  can be computed by using the continuity conditions for position, velocity and acceleration in point C, i.e. at  $t = \frac{3}{8}T$ . From the continuity condition for the acceleration, one obtains

$$\frac{8\pi h'}{T^2} = \left[ -c_3 \frac{16\pi^2}{T^2} \sin \left( 4\pi \frac{t - \frac{T}{4}}{T} \right) \right]_{t=\frac{3}{8}T}$$

and then

$$c_3 = -\frac{h'}{2\pi}.$$

For the velocity

$$\left[ \frac{2h'}{T} + \frac{8\pi h'}{T^2} \left( t - \frac{T}{8} \right) \right]_{t=\frac{3}{8}T} = \left[ \frac{c_2}{T} - \frac{h'}{2\pi} \quad \frac{4\pi}{T} \cos \left( 4\pi \frac{t - \frac{T}{4}}{T} \right) \right]_{t=\frac{3}{8}T}$$

from which

$$c_2 = 2h'(1 + \pi).$$

Finally, by assigning the position value in C, i.e.  $q\left(\frac{3}{8}T\right) = q_c$ , one obtains

$$\left[ c_1 + 2h'(1+\pi)\frac{t - \frac{3}{8}T}{T} - \frac{h'}{2\pi} \sin\left(4\pi\frac{t - \frac{T}{4}}{T}\right) \right]_{t=\frac{3}{8}T} = 0$$

that yields

$$c_1 = \frac{h'}{2\pi}.$$

By using these values of  $c_1$ ,  $c_2$ ,  $c_3$  in eq. (3.51) one obtains

$$q(t) = h' \left[ -\frac{\pi}{2} + 2(1+\pi)\frac{t}{T} - \frac{1}{2\pi} \sin\left(4\pi\frac{t - \frac{T}{4}}{T}\right) \right].$$

In point D (at  $t = \frac{T}{2}$ ) the position is

$$q_d = h' \left(1 + \frac{\pi}{2}\right).$$

Finally, from the condition  $q_d = \frac{h}{2}$  the relation between  $h$  and  $h'$  is determined

$$h' = \frac{h}{2 + \pi}.$$

Summarizing the above results, the trajectory is defined as

$$q(t) = \begin{cases} \frac{h}{2 + \pi} \left[ \frac{2t}{T} - \frac{1}{2\pi} \sin\left(\frac{4\pi t}{T}\right) \right], & 0 \leq t < \frac{T}{8} \\ \frac{h}{2 + \pi} \left[ \frac{1}{4} - \frac{1}{2\pi} + \frac{2}{T} \left(t - \frac{T}{8}\right) + \frac{4\pi}{T^2} \left(t - \frac{T}{8}\right)^2 \right], & \frac{T}{8} \leq t < \frac{3}{8}T \\ \frac{h}{2 + \pi} \left[ -\frac{\pi}{2} + 2(1+\pi)\frac{t}{T} - \frac{1}{2\pi} \sin\left(\frac{4\pi}{T} \left(t - \frac{T}{4}\right)\right) \right], & \frac{3}{8}T \leq t \leq \frac{T}{2}. \end{cases}$$

The second part of the trajectory (from  $T/2$  to  $T$ ) can be easily deduced by exploiting its symmetry and the rules concerning translation and reflection operations reported in Sec. 5.1. In particular, see Example 5.1, the following expression is obtained

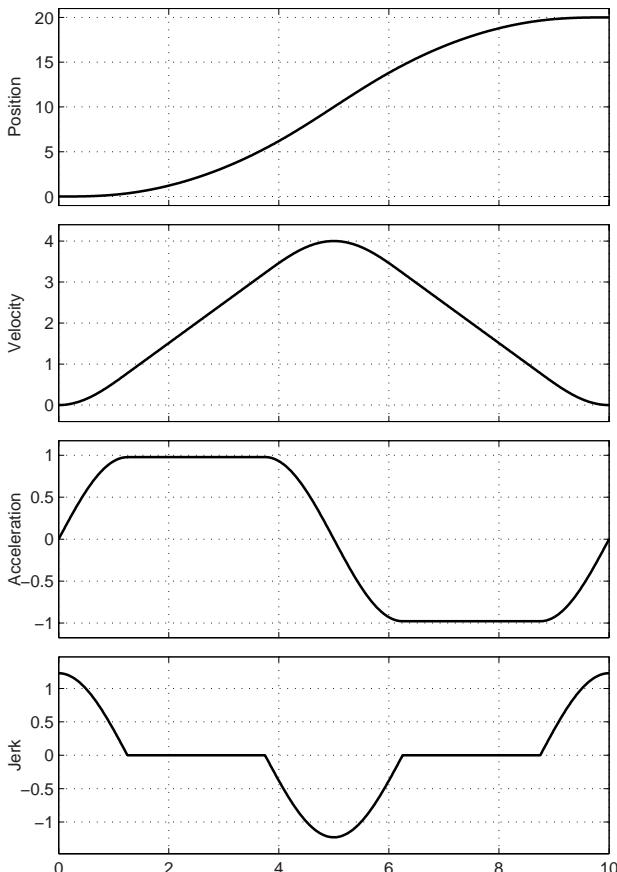
$$q(t) = \begin{cases} h + \frac{h}{2 + \pi} \left[ \frac{\pi}{2} + 2(1+\pi)\frac{t - T}{T} - \frac{1}{2\pi} \sin\left(\frac{4\pi}{T} \left(t - \frac{3T}{4}\right)\right) \right], & \frac{1}{2}T \leq t < \frac{5}{8}T \\ h + \frac{h}{2 + \pi} \left[ -\frac{1}{4} + \frac{1}{2\pi} + \frac{2}{T} \left(t - \frac{7T}{8}\right) - \frac{4\pi}{T^2} \left(t - \frac{7T}{8}\right)^2 \right], & \frac{5}{8}T \leq t < \frac{7}{8}T \\ h + \frac{h}{2 + \pi} \left[ \frac{2(t - T)}{T} - \frac{1}{2\pi} \sin\left(\frac{4\pi}{T}(t - T)\right) \right], & \frac{7}{8}T \leq t \leq T. \end{cases}$$

If  $q_0 \neq 0$ ,  $t_0 \neq 0$ , the value  $q_0$  must be added to the above equations, and  $(t - t_0)$  must be used in place of  $t$ .

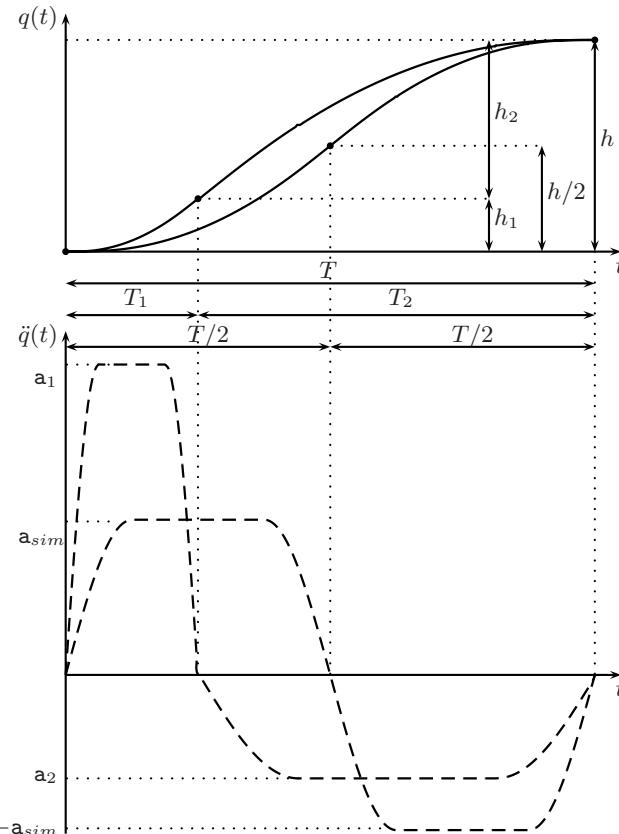
The maximum values of velocity, acceleration and jerk for the modified trapezoidal trajectory are

$$\dot{q}_{max} = 2 \frac{h}{T}, \quad \ddot{q}_{max} = 4.888 \frac{h}{T^2}, \quad q_{max}^{(3)} = 61.43 \frac{h}{T^3}.$$

**Example 3.21** Fig. 3.35 reports the profiles for position, velocity, acceleration and jerk for this trajectory when  $h = 20$ ,  $T = 10$ .  $\square$



**Fig. 3.35.** Position, velocity, acceleration and jerk of a modified trapezoidal trajectory with  $h = 20$ ,  $T = 10$ .



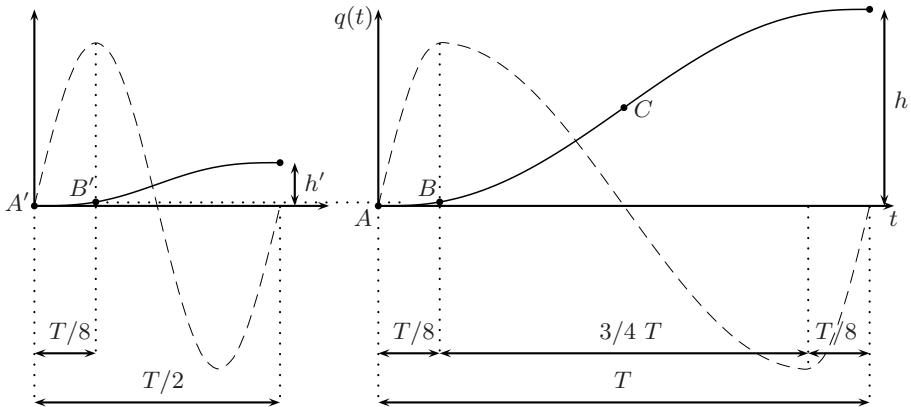
**Fig. 3.36.** Modified trapezoidal trajectory with different acceleration and deceleration periods.

It is possible to define a modified trapezoidal trajectory with different deceleration/acceleration periods, see Fig. 3.36. This is obtained by using the above method with proper boundary conditions. This type of trajectory can be very convenient when it is required to have given values for the velocity and the acceleration in specific points of the motion.

### 3.8 Modified Sinusoidal Trajectory

This trajectory is the combination of a cycloidal and of an harmonic trajectory, see Fig. 3.37 where a modified sinusoidal trajectory has been obtained from a cycloidal profile.

The first half of the trajectory is divided into two parts: between A and B (from  $t_0 = t_a = 0$  to  $t_b = \frac{T}{8}$ ) the acceleration has a sinusoidal profile (with duration of a quarter of period); then, it decreases to zero between B and C (from  $t_b = \frac{T}{8}$  to  $t_c = \frac{T}{2}$ ) with a specular sinusoidal profile.



**Fig. 3.37.** Modified sinusoidal trajectory.

The expression of the cycloidal trajectory from A to B is

$$\begin{cases} q(t) = h' \left( \frac{2t}{T} - \frac{1}{2\pi} \sin \frac{4\pi t}{T} \right) \\ \dot{q}(t) = \frac{h'}{T} \left( 2 - 2 \cos \frac{4\pi t}{T} \right) \\ \ddot{q}(t) = \frac{8\pi h'}{T^2} \sin \frac{4\pi t}{T}. \end{cases}$$

At  $t_b = \frac{T}{8}$  one obtains the following values

$$\begin{cases} q(t_b) = h' \left( \frac{1}{4} - \frac{1}{2\pi} \right) = q_b \\ \dot{q}(t_b) = \frac{2h'}{T} \\ \ddot{q}(t_b) = \frac{8\pi h'}{T^2}. \end{cases}$$

The general expression of the sinusoidal trajectory from B to C is

$$\begin{cases} q(t) = q_b + c_1 + c_2 \frac{t - \frac{T}{8}}{T} + c_3 \sin \left( \frac{4\pi t}{3T} + \frac{\pi}{3} \right) \\ \dot{q}(t) = \frac{c_2}{T} + c_3 \frac{4\pi}{3T} \cos \left( \frac{4\pi t}{3T} + \frac{\pi}{3} \right) \\ \ddot{q}(t) = -c_3 \frac{16\pi^2}{9T^2} \sin \left( \frac{4\pi t}{3T} + \frac{\pi}{3} \right) \end{cases}$$

where the coefficients  $c_1$ ,  $c_2$ , and  $c_3$  can be determined by means of the continuity conditions in point B. In particular, for the acceleration in B

$$\frac{8\pi h'}{T^2} = \left[ -c_3 \frac{16\pi^2}{9T^2} \sin\left(\frac{4\pi t}{3T} + \frac{\pi}{3}\right) \right]_{t=\frac{T}{8}}$$

from which

$$c_3 = -\frac{9h'}{2\pi}$$

and for the velocity

$$\frac{2h'}{T} = \left[ \frac{c_2}{T} - \frac{9h'}{2\pi} \cos\left(\frac{4\pi t}{3T} + \frac{\pi}{3}\right) \right]_{t=\frac{T}{8}}$$

from which

$$c_2 = 2h'.$$

Finally, at  $t = \frac{T}{8}$  the position is  $q(t) = q_b$ , and therefore

$$\left[ c_1 + 2h' \frac{t - \frac{T}{8}}{T} - \frac{9h'}{2\pi} \sin\left(\frac{4\pi t}{3T} + \frac{\pi}{3}\right) \right]_{t=\frac{T}{8}} = 0$$

then

$$c_1 = \frac{9h'}{2\pi}.$$

In conclusion, between B and C

$$\begin{aligned} q(t) &= \left( \frac{h'}{4} - \frac{h'}{2\pi} \right) + \frac{9h'}{2\pi} + 2h' \frac{t - \frac{T}{8}}{T} - \frac{9h'}{2\pi} \sin\left(\frac{4\pi t}{3T} + \frac{\pi}{3}\right) \\ &= h' \left[ \frac{4}{\pi} + 2 \frac{t}{T} - \frac{9}{2\pi} \sin\left(\frac{4\pi t}{3T} + \frac{\pi}{3}\right) \right]. \end{aligned}$$

The position at  $t = \frac{T}{2}$  is

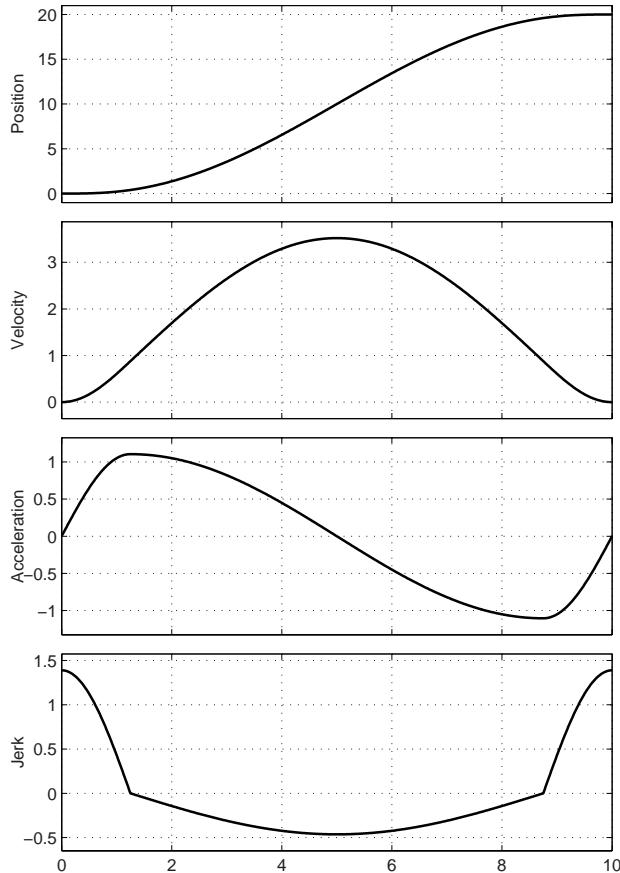
$$q_c = h' \left( 1 + \frac{4}{\pi} \right).$$

From the condition  $q_c = h/2$  one obtains

$$h' = \frac{\pi}{2(\pi + 4)} h.$$

Summarizing, the equations defining the modified sinusoidal trajectory are

$$q(t) = \begin{cases} h \left[ \frac{\pi t}{T(4+\pi)} - \frac{1}{4(4+\pi)} \sin \frac{4\pi t}{T} \right], & 0 \leq t < \frac{T}{8} \\ h \left[ \frac{2}{4+\pi} + \frac{\pi t}{T(4+\pi)} - \frac{9}{4(4+\pi)} \sin \left( \frac{4\pi t}{3T} + \frac{\pi}{3} \right) \right], & \frac{T}{8} \leq t < \frac{7}{8}T \\ h \left[ \frac{4}{4+\pi} + \frac{\pi t}{T(4+\pi)} - \frac{1}{4(4+\pi)} \sin \frac{4\pi t}{T} \right], & \frac{7}{8}T \leq t \leq T. \end{cases}$$



**Fig. 3.38.** Profiles of position, velocity, acceleration and jerk of a modified sinusoidal trajectory when  $h = 20$ ,  $T = 10$ .

The maximum values for velocity, acceleration and jerk are

$$\dot{q}_{max} = 1.76 \frac{h}{T}, \quad \ddot{q}_{max} = 5.528 \frac{h}{T^2}, \quad q_{max}^{(3)} = 69.47 \frac{h}{T^3}.$$

**Example 3.22** Fig. 3.38 shows the position, velocity, acceleration and jerk for this trajectory with  $h = 20$  and  $T = 10$ .  $\square$

### 3.9 Modified Cycloidal Trajectory

A cycloidal trajectory, defined by eq. (2.22), can be interpreted as the sum of a sinusoidal trajectory and of a constant velocity trajectory, with slope opposite

(and equal in amplitude) to the final slope of the sinusoidal profile, as shown in Fig. 3.39(a). In the figure, A and B are the initial and final points, P is the intermediate transition point, APB is the line of the constant velocity motion, M the intermediate point between A and P. The amplitude of the sinusoidal profile, in the cycloidal motion, must be added to the constant velocity trajectory in the direction perpendicular to the axis  $t$ .

A possible modification of the cycloidal trajectory is obtained by adding the sinusoidal profile in a direction perpendicular to the constant velocity profile, as shown in Fig. 3.39(b). This is the so-called “Alt modification”, after Herman Alt, a German kinematician who first proposed it, [7]. This modification allows to obtain a smoother profile, but implies an higher value of the maximum acceleration.

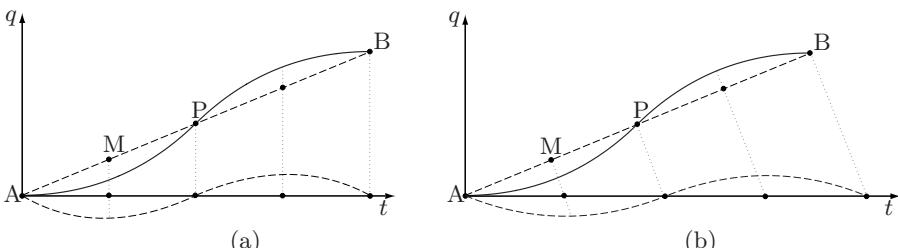
Another modification, which aims at reducing the maximum acceleration value, is shown in Fig. 3.40(a), known as the “Wildt modification” (after Paul Wildt, [7]). Point D is located at a distance equal to  $0.57\frac{T}{2}$  from  $\frac{T}{2}$ , and then connected to M. The segment DM defines the direction along which the sinusoidal profile is added to the constant velocity trajectory. In this manner, the maximum acceleration is  $5.88\frac{h}{T^2}$ , similar to the modified trapezoidal trajectory, whereas the standard cycloidal trajectory has a maximum acceleration of  $6.28\frac{h}{T^2}$ . Therefore, a reduction of 6.8% of the maximum acceleration value is achieved. In Fig. 3.41 the three acceleration profiles for the cycloidal trajectory and the two modified profiles are shown. Further details can be found in [6] and [7].

The general expression of the modified cycloidal trajectory, where the base sinusoidal motion is projected along a direction specified by a generic angle  $\gamma$  (see Fig. 3.40(b)), is

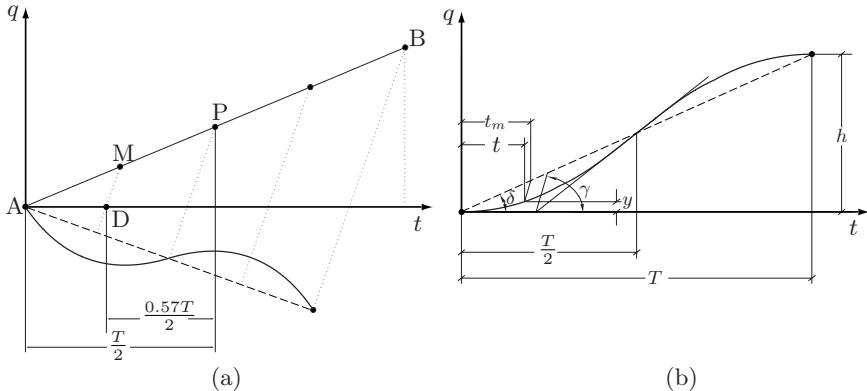
$$q(t_m) = h \left[ \frac{t_m}{T} - \frac{1}{2\pi} \sin \frac{2\pi t_m}{T} \right] \quad (3.52)$$

where  $t_m$  is defined by

$$t = t_m - \kappa \frac{T}{2\pi} \sin \frac{2\pi t_m}{T} \quad (3.53)$$



**Fig. 3.39.** Geometrical construction of the cycloidal (a) and modified cycloidal trajectory with the Alt method (b).



**Fig. 3.40.** Geometrical construction of the modified cycloidal trajectory with the Wildt method.

with

$$\kappa = \frac{\tan \delta}{\tan \gamma} \quad \text{and} \quad \tan \delta = \frac{h}{T}.$$

The angle  $\gamma$  determines the direction of the basis of the sinusoidal trajectory ( $\gamma$  is sometimes called the “distortion” angle). For example, for a pure cycloidal trajectory  $\gamma = \pi/2$  (as shown in Fig. 3.39(a)). The velocity and the acceleration of the trajectory can be computed by differentiating  $q(t)$  with respect to the time<sup>6</sup>  $t$ :

$$\begin{aligned}\dot{q}(t) &= \frac{h}{T} \frac{1 - \cos \frac{2\pi t}{T}}{1 - \kappa \cos \frac{2\pi t}{T}} \\ \ddot{q}(t) &= \frac{h}{T^2} \frac{2\pi(1 - \kappa) \sin \frac{2\pi t}{T}}{\left[1 - \kappa \cos \frac{2\pi t}{T}\right]^3}.\end{aligned}$$

The minimum value of the maximum acceleration is obtained for  $\kappa = 1 - \frac{\sqrt{3}}{2} = 0.134$ , case in which the acceleration is  $\ddot{q}_{max} = 5.88 \frac{h}{T^2}$ .

The Alt modification is obtained for  $\gamma = \pi/2 + \arctan(h/T)$ .

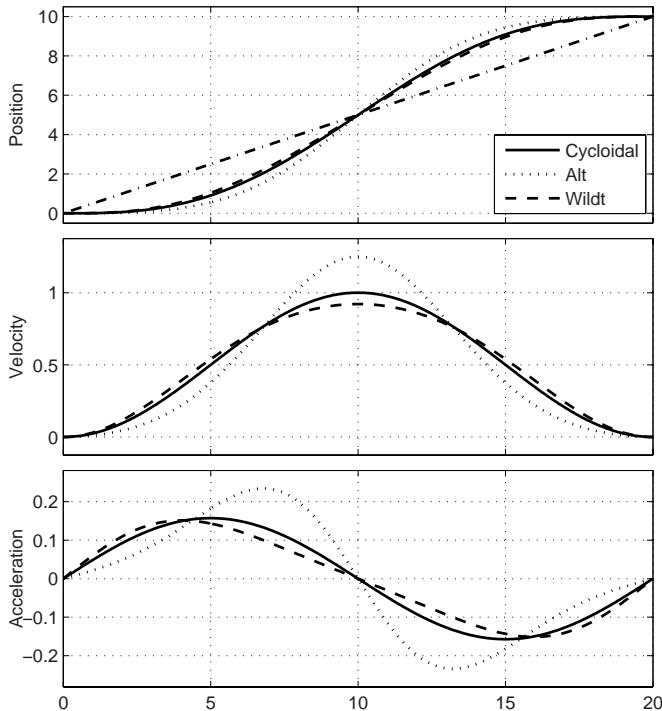
The velocity value at  $t = T/2$  (note that, in this case,  $t_m = T/2$ ) can be simply computed as

$$\dot{q}\left(\frac{T}{2}\right) = \frac{2h}{T}(1 - \kappa). \quad (3.54)$$

If  $\kappa = 0$  (i.e.  $\gamma = \pi/2$ ), the standard cycloidal motion is obtained with  $\dot{q}(T/2) = 2h/T$ , while the velocity at  $t = T/2$  is null for  $\kappa = 1$ . Note that

<sup>6</sup> The velocity and acceleration are obtained as

$$\dot{q} = \frac{dq}{dt_m} / \frac{dt}{dt_m}, \quad \ddot{q} = \frac{d\dot{q}}{dt_m} / \frac{dt}{dt_m}.$$

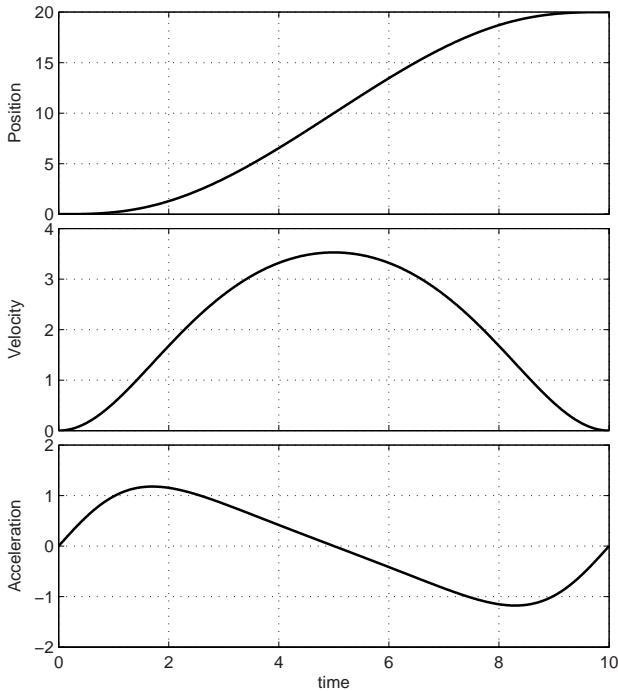


**Fig. 3.41.** Cycloidal and modified cycloidal trajectories (position, velocity, acceleration).

(3.54) can be used to compute the value of  $\kappa$ , and then of the distortion angle  $\gamma$ , necessary to obtain a specific value of the velocity at  $t = T/2$ .

**Example 3.23** Fig. 3.42 shows the profiles of position, velocity and acceleration for the modified cycloidal trajectory (Wildt method) with  $h = 20$ ,  $T = 10$  and  $\kappa = 0.134$ . In Fig. 3.43 the relation between the time  $t$  and  $t_m$  is reported.  $\square$

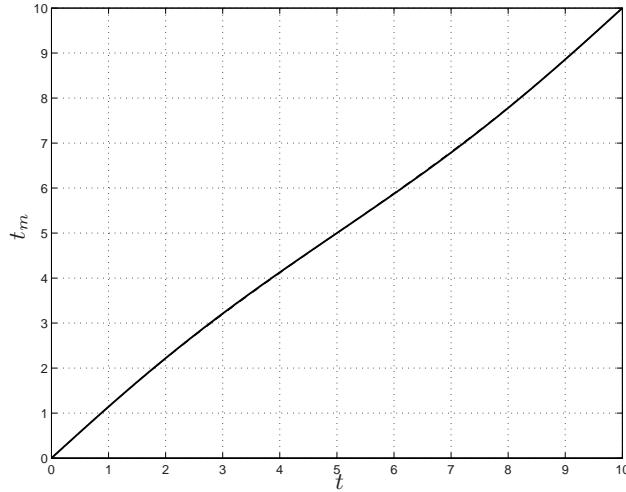
The practical implementation of the modified cycloidal expressed by (3.52) requires the inversion of (3.53) in order to find the function  $t_m(t)$ . This is obtained by solving numerically eq. (3.53) or by approximating in some way this function. For example, a simple 5-th degree polynomial  $\sigma(t)$  computed with the following boundary conditions



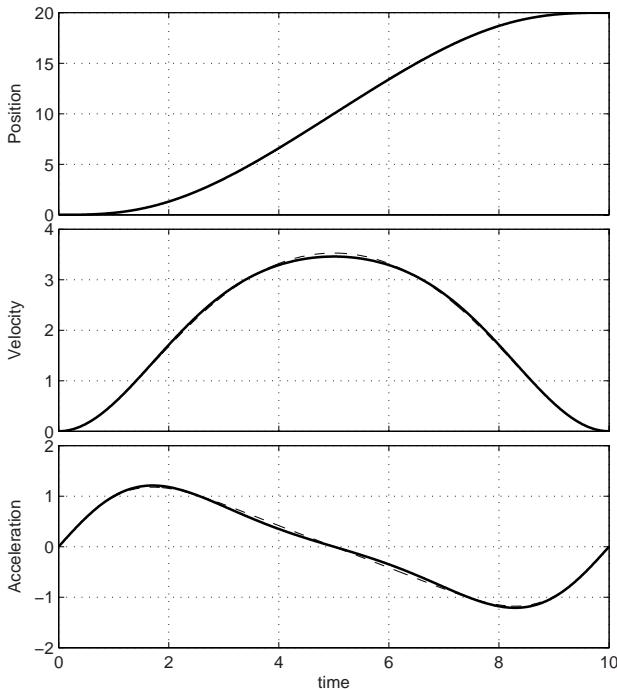
**Fig. 3.42.** Position, velocity and acceleration profiles for a modified cycloidal trajectory when  $h = 20$ ,  $T = 10$ ,  $\kappa = 0.134$ .

$$\begin{array}{ll} \sigma_0 = 0, & \sigma_1 = T, \\ v_0 = \frac{1}{1 - \kappa}, & v_1 = \frac{1}{1 - \kappa}, \\ a_0 = 0, & a_1 = 0, \end{array}$$

provides a good approximation of  $t_m(t)$  and the resulting trajectory is almost coincident with the ideal one (see Fig. 3.44, where the trajectory computed by approximating the function  $t_m(t)$  with a polynomial is compared with the theoretical one).



**Fig. 3.43.** Relation between  $t$  and  $t_m$  for the modified cycloidal trajectory of Fig. 3.42.



**Fig. 3.44.** Position, velocity and acceleration profiles for a modified cycloidal trajectory with  $h = 20$ ,  $T = 10$ ,  $\kappa = 0.134$  (solid), and same trajectory computed with a polynomial function of degree five approximating  $t_m(t)$  (dashed).

## 3.10 Constant Velocity/Acceleration Trajectories with Cycloidal or Harmonic Blends

In some applications, it may be of interest to plan trajectories by directly specifying the velocity (or acceleration) profile as a composition of constant segments connected by sinusoidal profiles. Then, the position, jerk, and so on, are simply obtained by means of integration or derivation operations.

In the following sections, some methods for planning trajectories with this approach are illustrated. In particular, constraints on the velocity or acceleration profile are considered with the goal of defining minimum-time motions.

### 3.10.1 Constraints on the velocity profile

In this case, the velocity profile is directly assigned by specifying the values of the velocity in the segments where it is constant, the duration of these segments and of the connecting blends, that are defined by sinusoidal functions. If these parameters are known, the trajectory is completely specified by computing the velocity and then the position by integration and the acceleration/jerk by derivation.

Note that the trajectory is subdivided into  $n$  segments, alternatively with sinusoidal velocity profile (odd) or with constant velocity (even), as shown in Fig. 3.45. The velocity in each segment is expressed by a function as

$$\dot{q}_k(t) = V_k \sin(\omega_k(t - t_{k-1}) + \phi_k) + K_k, \quad k = 1, \dots, n \quad (3.55)$$

where  $V_k$ ,  $\omega_k$ ,  $\phi_k$ , and  $K_k$  are parameters to be defined in order to satisfy the given constraints, and  $t_k$  are the transition instants between two consecutive segments. In a segment with constant velocity, it is necessary to define only the parameter  $K_k$ . Once the velocity profile is computed, the position is simply obtained by integration, i.e.

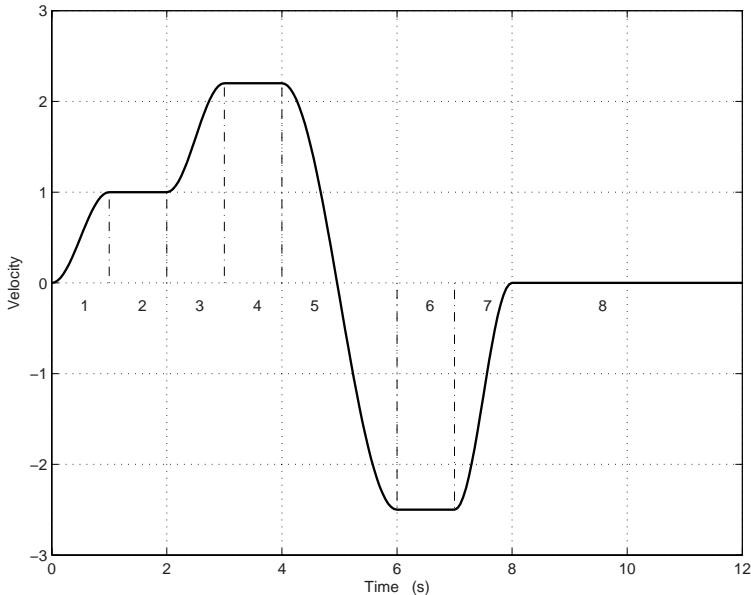
$$q(t) = q_0 + \int_{t_0}^t \dot{q}(\tau) d\tau$$

while the acceleration and jerk are obtained by derivation

$$\begin{cases} \ddot{q}_k(t) = V_k \omega_k \cos(\omega_k(t - t_{k-1}) + \phi_k) \\ q_k^{(3)}(t) = -V_k \omega_k^2 \sin(\omega_k(t - t_{k-1}) + \phi_k) \end{cases} \quad k = 1, \dots, n.$$

Note that for this trajectory the acceleration profile is continuous, composed by sinusoidal or null segments, while the jerk has a discontinuous profile.

The parameters in eq. (3.55) are defined as follows. With reference to Fig. 3.45, let  $t_0, t_1, t_2, \dots, t_n$  be the transition instants between two consecutive segments,  $v_0, v_n$  the initial and final velocity respectively, and  $v_2, v_4, v_6, \dots$  the constant velocity values in the intermediate segments. The parameters  $V_k$ ,  $K_k$ ,  $\omega_k$ ,  $\phi_k$ , are



**Fig. 3.45.** Subdivision of a velocity profile.

$$\begin{cases} V_k = \frac{v_{k+1} - v_{k-1}}{2} \\ K_k = \frac{v_{k+1} + v_{k-1}}{2} \\ \omega_k = \frac{\pi}{t_k - t_{k-1}} \\ \phi_k = \frac{3\pi}{2} \end{cases} \quad k = 1, 3, 5, \dots \quad (\text{odd segments})$$

$$\begin{cases} V_k = 0 \\ K_k = v_k \\ \omega_k = 0 \\ \phi_k = 0 \end{cases} \quad k = 2, 4, 6, \dots \quad (\text{even segments}).$$

Note that with this choice of the parameters, and in particular of the phase  $\phi_k$ , the expression of the segments composing the trajectory is

$$\begin{cases} q_k(t) = -\frac{V_k}{\omega_k} \sin(\omega_k(t - t_{k-1})) + K_k(t - t_{k-1}) + q_{k-1} \\ \dot{q}_k(t) = -V_k \cos(\omega_k(t - t_{k-1})) + K_k \\ \ddot{q}_k(t) = V_k \omega_k \sin(\omega_k(t - t_{k-1})) \\ q_k^{(3)}(t) = V_k \omega_k^2 \cos(\omega_k(t - t_{k-1})) \end{cases} \quad k = 1, \dots, n.$$

where  $q_{k-1}$  is the position at time  $t_{k-1}$ .

**Example 3.24** Fig. 3.46 and Fig. 3.47 show the jerk, acceleration, velocity and position profiles of trajectories obtained with the following conditions:

a)  $[t_0, t_1, \dots, t_8] = [0, 1, 2, 3, 4, 6, 7, 8, 12]$ ,

$$[v_0, v_2, v_4, v_6, v_8] = [0, 1, 2, -3, 0].$$

b)  $[t_0, t_1, t_2, t_3] = [0, 1, 2, 3]$ ,

$$[v_0, v_2, v_4] = [0, 1, 0].$$

c)  $[t_0, t_1, \dots, t_7] = [0, 1, 2, 3, 4, 5, 6, 7]$ ,

$$[v_0, v_2, v_4, v_6, v_8] = [0, 1, 0, -1, 0].$$

□

From the above examples, it is possible to appreciate the flexibility of these functions. In particular, the trajectory described in the last example can be considered as a modified trapezoidal velocity trajectory with a continuous acceleration profile. Finally, notice that the profiles in each segment are computed independently from each other. For this reason, some of these segments, in particular those with constant velocity, may be not present.

### 3.10.2 Constraints on the acceleration profile

An alternative way to plan this type of trajectory is to assign constraints not on the velocity but on the acceleration profile. As in the previous case, also in this manner very flexible functions are obtained. The resulting trajectory can be considered as a generalization of the modified trapezoidal function illustrated in Sec. 3.7.

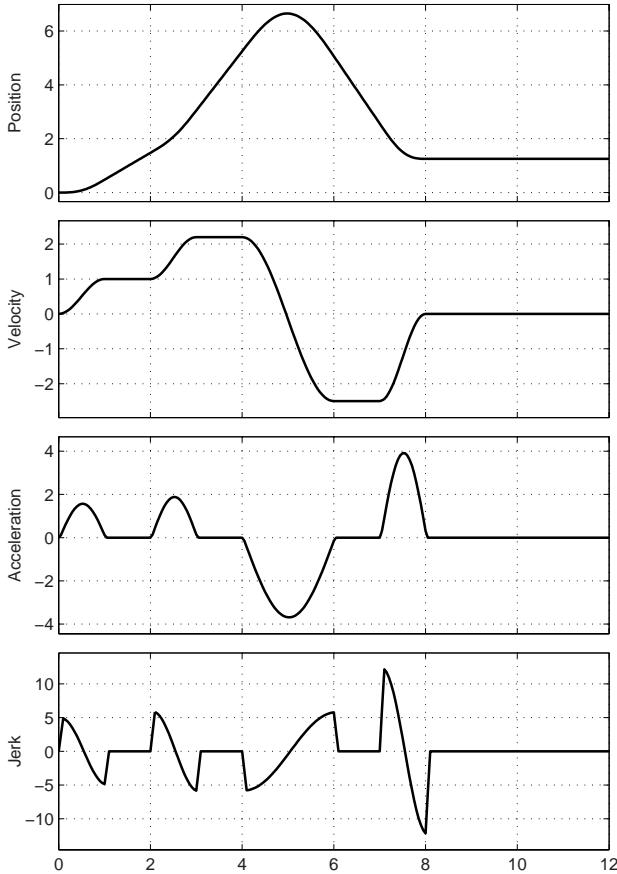
In Fig. 3.48 a generic acceleration profile composed by seven segment is shown. The odd segments ( $k = 1, 3, 5, 7$ ) are characterized by a sinusoidal profile, while the even ones ( $k = 2, 4, 6$ ) have a constant value  $a_2, a_4 (= 0)$  and  $a_6$  respectively.  $T_1, T_2, \dots, T_7$  are the time lengths of each segment,  $t_0, t_1, t_2, \dots, t_7 = t_f$  are the transition instants between two consecutive segments and  $v_0 = \dot{q}(t_0), \dots, v_7 = \dot{q}(t_7), a_0 = \ddot{q}(t_0), \dots, a_7 = \ddot{q}(t_7)$  are the corresponding velocity and acceleration values.

The following boundary conditions are considered at  $t = t_0$  and  $t = t_7$

$$q(t_0) = q_0 = 0, \quad \dot{q}(t_0) = v_0 = 0, \quad \ddot{q}(t_0) = a_0 = 0,$$

$$q(t_7) = q_7 = q_f, \quad \dot{q}(t_7) = v_7 = 0, \quad \ddot{q}(t_7) = a_7 = 0.$$

Moreover, for the sake of simplicity, a normalized expression of the trajectory is now considered, i.e.  $t_0 = 0, t_f = 1$  ( $T = 1$ ) and  $q_0 = 0, q_f = 1$  ( $h = 1$ ).



**Fig. 3.46.** Position, velocity, acceleration and jerk of Example 3.24.a.

Therefore, if a generic motion has to be computed, the consideration reported in Chapter 5 can be applied.

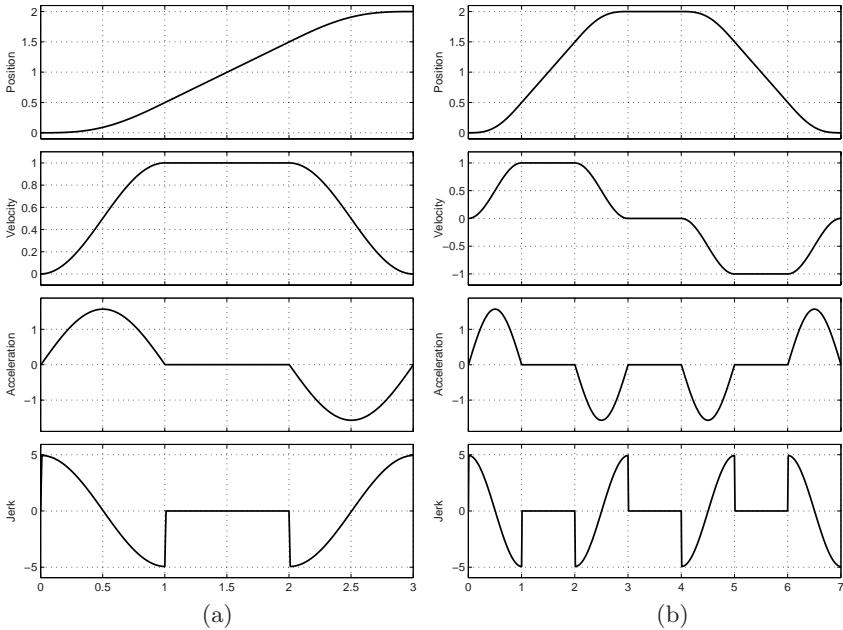
The value of the accelerations  $a_2$  and  $a_6$  are firstly computed by imposing the conditions  $q_7 = 1$ ,  $v_7 = 0$ . One obtains

$$\begin{cases} a_2 = \frac{-c_2}{c_1 c_4 - c_2 c_3} \\ a_6 = \frac{c_1}{c_1 c_4 - c_2 c_3} \end{cases}$$

where the constant  $c_k$  are

$$c_1 = \frac{2T_1}{\pi} + T_2 + \frac{2T_3}{\pi}$$

$$c_2 = -\frac{2T_5}{\pi} - T_6 - \frac{2T_7}{\pi}$$



**Fig. 3.47.** Position, velocity, acceleration and jerk of Example 3.24.b (a), and Example 3.24.c (b).

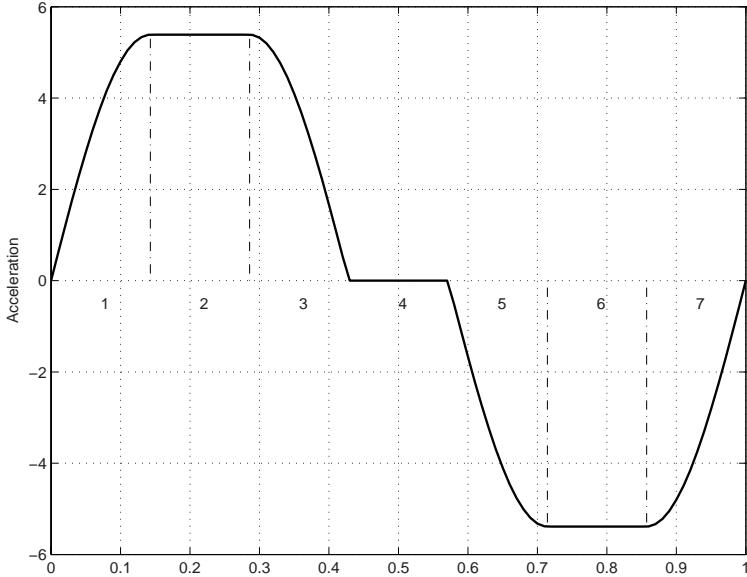
$$\begin{aligned}
 c_3 &= \frac{2T_1}{\pi} \left( \frac{\pi-2}{\pi} T_1 + \frac{T_2}{2} \right) + \left( \frac{2T_1}{\pi} + T_2 \right) \left( \frac{T_2}{2} + \frac{\pi-2}{\pi} T_3 \right) + \\
 &\quad + \left( \frac{2T_1}{\pi} + T_2 + \frac{2T_3}{\pi} \right) \left( \frac{2T_3}{\pi} + T_4 + \frac{2T_5}{\pi} \right) \\
 c_4 &= \left( \frac{2T_7}{\pi} + T_6 \right) \left( \frac{\pi-2}{\pi} T_5 + \frac{T_6}{2} \right) + \frac{2T_7}{\pi} \left( \frac{T_6}{2} + \frac{\pi-2}{\pi} T_7 \right).
 \end{aligned}$$

Once  $a_2$  and  $a_6$  have been defined, the trajectory is computed according to the following scheme:

1<sup>st</sup> segment  $t_0 \leq t \leq t_1$

$$\begin{cases} \ddot{q}(t) = a_2 \sin \frac{(t-t_0)\pi}{2T_1} \\ \dot{q}(t) = a_2 \frac{2T_1}{\pi} \left( 1 - \cos \frac{(t-t_0)\pi}{2T_1} \right) \\ q(t) = a_2 \frac{2T_1}{\pi} \left( t - \frac{2T_1}{\pi} \sin \frac{(t-t_0)\pi}{2T_1} \right) \end{cases}$$

with



**Fig. 3.48.** Generic acceleration profile with sinusoidal blends.

$$q_0 = 0, \quad v_0 = 0.$$

**2<sup>nd</sup> segment**  $t_1 \leq t \leq t_2$

$$\begin{cases} \ddot{q}(t) = a_2 \\ \dot{q}(t) = v_1 + a_2(t - t_1) \\ q(t) = q_1 + v_1(t - t_1) + a_2 \frac{(t - t_1)^2}{2} \end{cases}$$

with

$$q_1 = a_2 \frac{2T_1^2}{\pi} \left( 1 - \frac{2}{\pi} \right), \quad v_1 = a_2 \frac{2T_1}{\pi}.$$

**3<sup>rd</sup> segment**  $t_2 \leq t \leq t_3$

$$\begin{cases} \ddot{q}(t) = a_2 \cos \frac{(t - t_2)\pi}{2T_3} \\ \dot{q}(t) = v_2 + a_2 \frac{2T_3}{\pi} \sin \frac{(t - t_2)\pi}{2T_3} \\ q(t) = q_2 + v_2(t - t_2) + a_2 \left( \frac{2T_3}{\pi} \right)^2 \left( 1 - \cos \frac{(t - t_2)\pi}{2T_3} \right) \end{cases}$$

with

$$q_2 = q_1 + a_2 T_2 \left( \frac{2T_1}{\pi} + \frac{T_2}{2} \right), \quad v_2 = v_1 + a_2 T_2.$$

**4<sup>th</sup> segment**  $t_3 \leq t \leq t_4$

$$\begin{cases} \ddot{q}(t) = 0 \\ \dot{q}(t) = v_3 \\ q(t) = q_3 + v_3(t - t_3) \end{cases}$$

with

$$q_3 = q_2 + a_2 T_3 \left( \frac{2T_1}{\pi} + T_2 + \frac{4T_3}{\pi^2} \right), \quad v_3 = v_2 + a_2 \frac{2T_3}{\pi}$$

**5<sup>th</sup> segment**  $t_4 \leq t \leq t_5$

$$\begin{cases} \ddot{q}(t) = -a_6 \sin \frac{(t - t_4)\pi}{2T_5} \\ \dot{q}(t) = v_4 - a_6 \frac{2T_5}{\pi} \left( 1 - \cos \frac{(t - t_4)\pi}{2T_5} \right) \\ q(t) = q_4 + v_4(t - t_4) - a_6 \frac{2T_5}{\pi} \left( t - t_4 - \frac{2T_5}{\pi} \sin \frac{(t - t_4)\pi}{2T_5} \right) \end{cases}$$

with

$$q_4 = q_3 + a_2 T_4 \left( \frac{2T_1}{\pi} + T_2 + \frac{2T_3}{\pi} \right), \quad v_4 = v_3.$$

**6<sup>th</sup> segment**  $t_5 \leq t \leq t_6$

$$\begin{cases} \ddot{q}(t) = -a_6 \\ \dot{q}(t) = v_5 - a_6(t - t_5) \\ q(t) = q_5 + v_5(t - t_5) - a_6 \frac{(t - t_5)^2}{2} \end{cases}$$

with

$$q_5 = q_4 + a_2 T_5 \left( \frac{2T_1}{\pi} + T_2 + \frac{2T_3}{\pi} \right) - a_6 \frac{2T_5^2}{\pi} \left( 1 - \frac{2}{\pi} \right), \quad v_5 = v_4 - a_6 \frac{2T_5}{\pi}.$$

**7<sup>th</sup> segment**  $t_6 \leq t \leq t_7$

$$\begin{cases} \ddot{q}(t) = -a_6 \cos \frac{(t - t_6)\pi}{2T_7} \\ \dot{q}(t) = v_6 - a_6 \frac{2T_7}{\pi} \sin \frac{(t - t_6)\pi}{2T_7} \\ q(t) = q_6 + v_6(t - t_6) - a_6 \left( \frac{2T_7}{\pi} \right)^2 \left( 1 - \cos \frac{(t - t_6)\pi}{2T_7} \right) \end{cases}$$

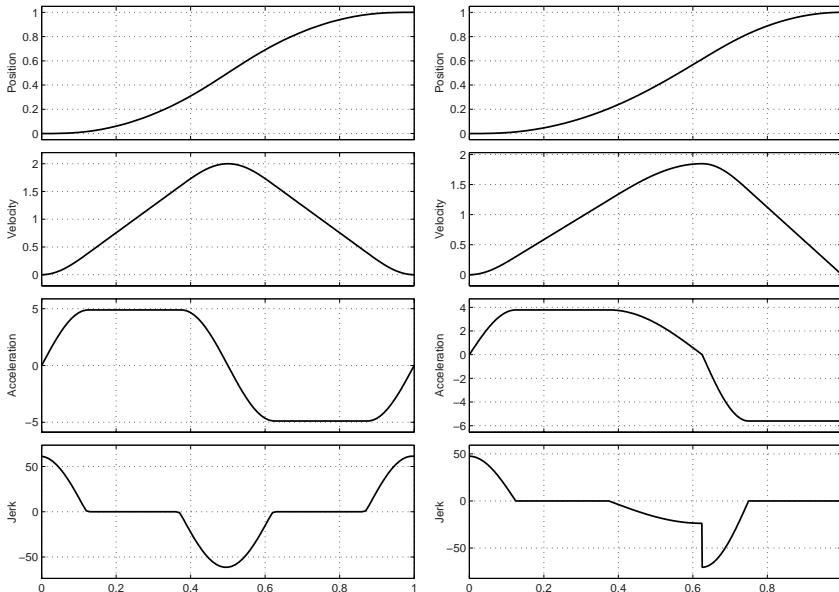
with

$$q_6 = q_5 + v_5 T_6 - a_6 \frac{T_6^2}{2}, \quad v_6 = v_5 - a_6 T_6.$$

The acceleration profile is then defined once the values of  $t_1, t_2, \dots, t_7$  are assigned. As a matter of fact, the parameters  $a_2, a_6, v_1, \dots, v_6, q_1, \dots, q_6$  depend only on the time instants  $t_k$  and may be computed in advance. Note that some of the time instants  $t_k$  may be coincident, and therefore the corresponding periods  $T_k$  are null. As a consequence, the relative segment is not present.

By properly assigning the values of  $t_1, t_2, \dots, t_7 \in [0, 1]$  one may obtain a rich set of trajectories, able to satisfy a number of requirements that can be assigned on the motion law.

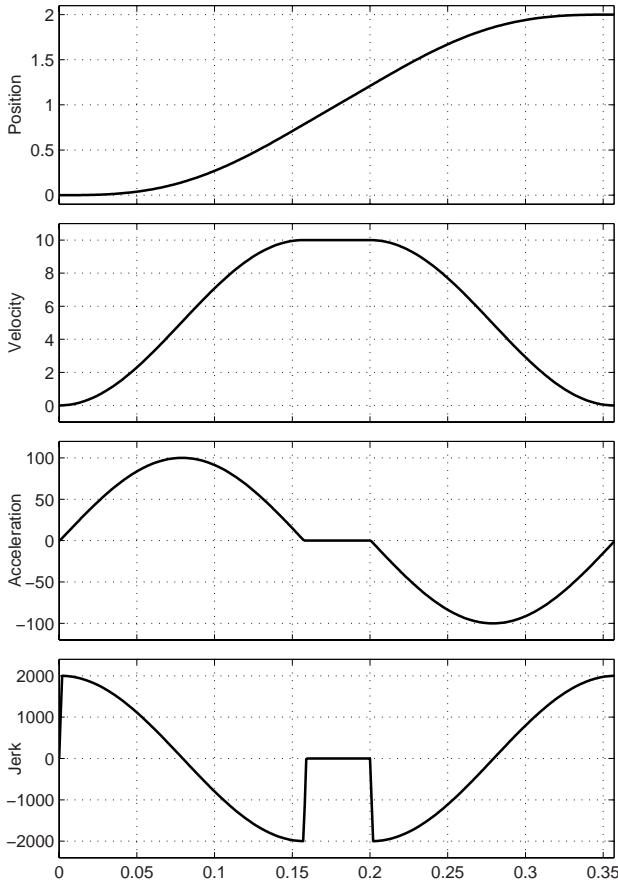
**Example 3.25** The trajectories reported in Fig. 3.49 have been obtained with the two sets of periods  $T_k$   $[0.125, 0.25, 0.125, 0, 0.125, 0.25, 0.125]$  and  $[0.125, 0.25, 0.25, 0, 0.125, 0.25, 0]$  respectively.  $\square$



**Fig. 3.49.** Trajectories obtained with different choices of the parameters  $T_k$ : position, velocity, acceleration, and jerk.

### 3.10.3 Minimum-time trajectories

A third way to compute this type of trajectories is to specify initial and final position values  $q_0$  and  $q_1$ , and then compute the trajectory in order to minimize its duration. This is equivalent to impose in each segment of the



**Fig. 3.50.** Minimum-time trajectory composed by trigonometric and polynomial segments.

trajectory the maximum value allowed for the velocity or the acceleration, similarly to what illustrated for the trapezoidal velocity trajectory in Sec. 3.2. For this purpose, we refer to the expression reported in eq. (3.55), that is

$$\dot{q}_k(t) = V_k \sin(\omega_k(t - t_{k-1}) + \phi_k) + K_k, \quad k = 1, \dots, n$$

and to the profile shown in Fig. 3.50, composed by an acceleration segment (with duration  $T_1$ ), a constant velocity tract ( $T_2$ ), and a deceleration segment (with  $T_3 = T_1$  because of symmetry).

For the sake of simplicity, only the case  $q_1 > q_0$ , with  $q_0 = 0$ , is now considered, with  $v_{min} = -v_{max}$ ,  $a_{min} = -a_{max}$ , being  $v_{max}$ ,  $a_{max}$  the velocity and acceleration limits.

One obtains

$$\left\{ \begin{array}{ll} T_1 = \frac{\pi v_{max}}{2 a_{max}}, & \text{acceleration time} \\ T_2 = \frac{h}{v_{max}} - T_1, & \text{duration of the const. vel. segment} \\ T_3 = T_1, & \text{deceleration time} \\ T = 2T_1 + T_2 = \frac{2ha_{max} + \pi v_{max}^2}{2a_{max}v_{max}}, & \text{total duration.} \end{array} \right.$$

The four parameters  $A_k$ ,  $\omega_k$ ,  $\phi_k$ ,  $K_k$  in (3.55) are

**1<sup>st</sup> segment**  $t \in [0, T_1]$

$$V_1 = \frac{v_{max}}{2}, \quad K_1 = \frac{v_{max}}{2}, \quad \omega_1 = \frac{\pi}{T_1}, \quad \phi_1 = \frac{3\pi}{2}.$$

**2<sup>nd</sup> segment**  $t \in [0, T_2]$

$$V_2 = 0, \quad K_2 = v_{max}, \quad \omega_2 = 0, \quad \phi_2 = 0.$$

**3<sup>rd</sup> segment**  $t \in [0, T_3]$

$$V_3 = \frac{v_{max}}{2}, \quad K_3 = \frac{v_{max}}{2}, \quad \omega_3 = \frac{\pi}{T_3}, \quad \phi_3 = \frac{\pi}{2}.$$

Once these parameters are known, the jerk, acceleration, velocity and position profiles can be computed as

$$\left\{ \begin{array}{ll} q_k^{(3)}(t) = -V_k \omega_k^2 \sin(\omega_k(t - t_{k-1}) + \phi_k) \\ \ddot{q}_k(t) = V_k \omega_k \cos(\omega_k(t - t_{k-1}) + \phi_k) \\ \dot{q}_k(t) = V_k \sin(\omega_k(t - t_{k-1}) + \phi_k) + K_k \\ q_k(t) = q_{k-1} - \frac{A_k}{\omega_k} \cos(\omega_k(t - t_{k-1}) + \phi_k) + K_k (t - t_{k-1}) \end{array} \right. \quad k = 1, 2, 3.$$

Notice that the constant velocity segment exists if and only if the following condition holds

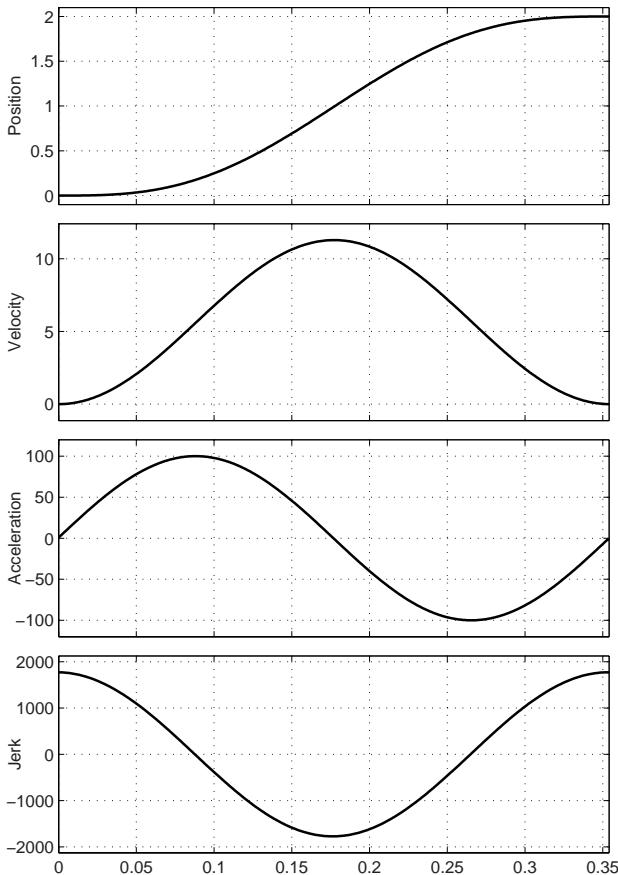
$$h \geq \frac{\pi}{4} \frac{v_{max}^2}{a_{max}}$$

otherwise

$$\left\{ \begin{array}{ll} T_1 = \sqrt{\frac{h \pi}{2 a_{max}}}, & \text{acceleration time} \\ T = 2 T_1, & \text{total time} \end{array} \right.$$

and the maximum velocity  $v_{lim}$  actually reached by the trajectory is

$$v_{lim} = \sqrt{\frac{2 h a_{max}}{\pi}}.$$

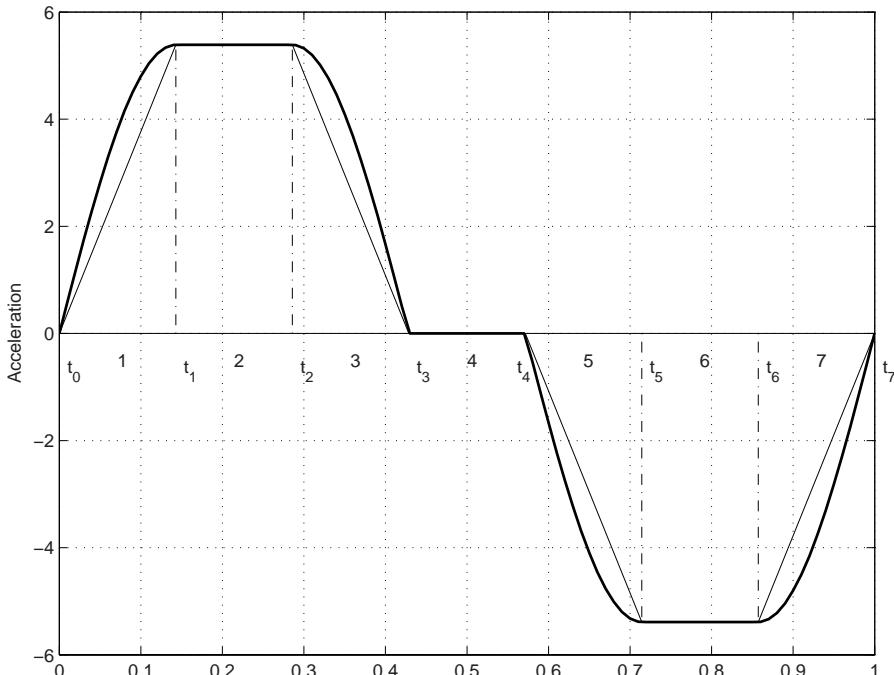


**Fig. 3.51.** Minimum-time cycloidal trajectory without constant velocity segment.

Given these two new parameters  $T_1, v_{lim}$ , the above equations can be used for the computation of the trajectory. An example of this trajectory is reported in Fig. 3.51.

### 3.11 Trajectories with Constant Acceleration and Cycloidal/Cubic Blends

The procedures illustrated above allow to define a trajectory composed by constant velocity/acceleration segments connected by trigonometric blends. A more general approach allows to define a trajectory made of constant acceleration segments connected by either trigonometric or polynomial functions. The total displacement is subdivided into seven parts, as shown in Fig. 3.52, where  $t_0, t_1, t_2, \dots, t_7 = t_f$  are the time instants of transition between two consecutive segments. Each tract is characterized by a different motion law. The even segments (2, 4, 6) have a constant acceleration, and therefore a parabolic position profile, while in the odd segments (1, 3, 5, 7) it is possible to define either trigonometric or linear acceleration profiles.



**Fig. 3.52.** Acceleration of a generic trajectory: the profiles in the odd segments may be defined either by linear or trigonometric functions.

By assuming  $T_k = t_k - t_{k-1}$ , the trajectory is defined as follows:

**1<sup>st</sup> segment** (cycloidal or cubic)  $t_0 \leq t \leq t_1$

$$\text{Cycloidal: } \begin{cases} \ddot{q}_1(t) = \mathbf{a}_1 \sin\left(\frac{\pi(t-t_0)}{2T_1}\right) \\ \dot{q}_1(t) = -\mathbf{a}_1 T_1 \frac{2}{\pi} \cos\left(\frac{\pi(t-t_0)}{2T_1}\right) + k_{11} \\ q_1(t) = -\mathbf{a}_1 \left(T_1 \frac{2}{\pi}\right)^2 \sin\left(\frac{\pi(t-t_0)}{2T_1}\right) + k_{11}(t-t_0) + k_{12} \end{cases}$$

$$\text{Cubic: } \begin{cases} \ddot{q}_1(t) = \mathbf{a}_1 \left(\frac{t-t_0}{T_1}\right) \\ \dot{q}_1(t) = \frac{\mathbf{a}_1}{2T_1}(t-t_0)^2 + k_{11} \\ q_1(t) = \frac{\mathbf{a}_1}{6T_1}(t-t_0)^3 + k_{11}(t-t_0) + k_{12} \end{cases}$$

**2<sup>nd</sup> segment** (parabolic)  $t_1 \leq t \leq t_2$

$$\text{Parabolic: } \begin{cases} \ddot{q}_2(t) = \mathbf{a}_1 \\ \dot{q}_2(t) = \mathbf{a}_1(t-t_1) + k_{21} \\ q_2(t) = \frac{\mathbf{a}_1}{2}(t-t_1)^2 + k_{21}(t-t_1) + k_{22} \end{cases}$$

**3<sup>rd</sup> segment** (cycloidal or cubic)  $t_2 \leq t \leq t_3$

$$\text{Cycloidal: } \begin{cases} \ddot{q}_3(t) = \mathbf{a}_1 \cos\left(\frac{\pi(t-t_2)}{2T_3}\right) \\ \dot{q}_3(t) = \mathbf{a}_1 T_3 \frac{2}{\pi} \sin\left(\frac{\pi(t-t_2)}{2T_3}\right) + k_{31} \\ q_3(t) = -\mathbf{a}_1 \left(T_3 \frac{2}{\pi}\right)^2 \cos\left(\frac{\pi(t-t_2)}{2T_3}\right) + k_{31}(t-t_2) + k_{32} \end{cases}$$

$$\text{Cubic: } \begin{cases} \ddot{q}_3(t) = \mathbf{a}_1 \left(1 - \frac{t-t_2}{T_3}\right) \\ \dot{q}_3(t) = -\frac{\mathbf{a}_1}{2T_3}(t-t_2)^2 + \mathbf{a}_1(t-t_2) + k_{31} \\ q_3(t) = -\frac{\mathbf{a}_1}{6T_3}(t-t_2)^3 + \frac{\mathbf{a}_1}{2}(t-t_2)^2 + k_{31}(t-t_2) + k_{32} \end{cases}$$

**4<sup>th</sup> segment** (constant velocity)  $t_3 \leq t \leq t_4$

$$\text{Constant velocity: } \begin{cases} \ddot{q}_4(t) = 0 \\ \dot{q}_4(t) = k_{41} \\ q_4(t) = k_{41}(t-t_3) + k_{42} \end{cases}$$

$5^{th}$  segment (cycloidal or cubic)  $t_4 \leq t \leq t_5$

$$\begin{aligned} \text{Cycloidal: } & \begin{cases} \ddot{q}_5(t) = \mathbf{a}_2 \sin\left(\frac{\pi(t-t_4)}{2T_5}\right) \\ \dot{q}_5(t) = -\mathbf{a}_2 T_5 \frac{2}{\pi} \cos\left(\frac{\pi(t-t_4)}{2T_5}\right) + k_{51} \\ q_5(t) = -\mathbf{a}_2 \left(T_5 \frac{2}{\pi}\right)^2 \sin\left(\frac{\pi(t-t_4)}{2T_5}\right) + k_{51}(t-t_4) + k_{52} \end{cases} \\ \text{Cubic: } & \begin{cases} \ddot{q}_5(t) = \mathbf{a}_2 \left(\frac{t-t_4}{T_5}\right) \\ \dot{q}_5(t) = \frac{\mathbf{a}_2}{2T_5} (t-t_4)^2 + k_{51} \\ q_5(t) = \frac{\mathbf{a}_2}{6T_5} (t-t_4)^3 + k_{51}(t-t_4) + k_{52} \end{cases} \end{aligned}$$

$6^{th}$  segment (parabolic)  $t_5 \leq t \leq t_6$

$$\text{Parabolic: } \begin{cases} \ddot{q}_6(t) = \mathbf{a}_2 \\ \dot{q}_6(t) = \mathbf{a}_2(t-t_5) + k_{61} \\ q_6(t) = \frac{\mathbf{a}_2}{2} (t-t_5)^2 + k_{61}(t-t_5) + k_{62} \end{cases}$$

$7^{th}$  segment (cycloidal or cubic)  $t_6 \leq t \leq t_7$

$$\begin{aligned} \text{Cycloidal: } & \begin{cases} \ddot{q}_7(t) = \mathbf{a}_2 \cos\left(\frac{\pi(t-t_6)}{2T_7}\right) \\ \dot{q}_7(t) = \mathbf{a}_2 T_7 \frac{2}{\pi} \sin\left(\frac{\pi(t-t_6)}{2T_7}\right) + k_{71} \\ q_7(t) = -\mathbf{a}_2 \left(T_7 \frac{2}{\pi}\right)^2 \cos\left(\frac{\pi(t-t_6)}{2T_7}\right) + k_{71}(t-t_6) + k_{72} \end{cases} \\ \text{Cubic: } & \begin{cases} \ddot{q}_7(t) = \mathbf{a}_2 \left(1 - \frac{t-t_6}{T_7}\right) \\ \dot{q}_7(t) = -\frac{\mathbf{a}_2}{2T_7} (t-t_6)^2 + \mathbf{a}_2(t-t_6) + k_{71} \\ q_7(t) = -\frac{\mathbf{a}_2}{6T_7} (t-t_6)^3 + \frac{\mathbf{a}_2}{2} (t-t_6)^2 + k_{71}(t-t_6) + k_{72}. \end{cases} \end{aligned}$$

The parameters  $\mathbf{a}_1$  and  $\mathbf{a}_2$  represent the maximum and minimum acceleration value, respectively with positive and negative sign, see Fig. 3.52. The parameters  $k_{ij}$  are defined by imposing the boundary conditions and the constraints on the continuity of position and velocity in the intermediate points. One obtains a linear system of sixteen equations in the sixteen unknowns  $\mathbf{a}_1$ ,  $\mathbf{a}_2$ ,  $k_{ij}$ :

$$\begin{aligned}
\dot{q}_1(t_0) &= \mathbf{v}_0, & q_1(t_0) &= q_0 \\
\dot{q}_1(t_1) &= \dot{q}_2(t_1), & q_1(t_1) &= q_2(t_1) \\
\dot{q}_2(t_2) &= \dot{q}_3(t_2), & q_2(t_2) &= q_3(t_2) \\
&\dots &&\dots \\
\dot{q}_7(t_7) &= \mathbf{v}_7, & q_7(t_7) &= q_7
\end{aligned}$$

where  $q_0$ ,  $q_7$  and  $\mathbf{v}_0$ ,  $\mathbf{v}_7$  are the initial and final positions and velocities respectively. In matrix form

$$\left[ \begin{array}{cccccccccccccc} m_{1,1} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ m_{2,1} & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ m_{3,1} & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ m_{4,1} & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{5,2} & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{6,2} & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{7,2} & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{8,2} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ m_{9,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ m_{10,1} & 0 & T_1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ m_{11,1} & 0 & 0 & T_2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ m_{12,1} & 0 & 0 & 0 & T_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & m_{13,2} & 0 & 0 & 0 & T_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & m_{14,2} & 0 & 0 & 0 & 0 & T_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & m_{15,2} & 0 & 0 & 0 & 0 & 0 & T_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & m_{16,2} & 0 & 0 & 0 & 0 & 0 & 0 & T_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] = \left[ \begin{array}{c} \mathbf{a}_1 \\ \mathbf{a}_2 \\ k_{11} \\ k_{21} \\ k_{31} \\ k_{41} \\ k_{51} \\ k_{61} \\ k_{71} \\ k_{12} \\ k_{22} \\ k_{32} \\ k_{42} \\ k_{52} \\ k_{62} \\ k_{72} \end{array} \right] = \left[ \begin{array}{c} \mathbf{v}_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ q_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ q_7 \end{array} \right]$$

or

$$\mathbf{M} \mathbf{k} = \mathbf{q}. \quad (3.56)$$

By solving this system, the unknowns  $\mathbf{a}_1, \mathbf{a}_2, k_{ij}$  which define the trajectory are obtained. Notice that the vector  $\mathbf{q}$  and the structure of the matrix  $\mathbf{M}$  are fixed and do not depend on the functions adopted for each segment. On the other hand, the coefficients  $m_{i,j}$  in the first two columns of  $\mathbf{M}$  depend on the choice of the function in the odd segments.

As a matter of fact, one obtains

### 1<sup>st</sup> segment

$$\text{Cycloidal: } m_{1,1} = -\frac{2T_1}{\pi}, \quad m_{2,1} = 0, \quad m_{9,1} = 0, \quad m_{10,1} = -\left(\frac{2T_1}{\pi}\right)^2,$$

$$\text{Cubic: } m_{1,1} = 0, \quad m_{2,1} = \frac{T_1}{2}, \quad m_{9,1} = 0, \quad m_{10,1} = \frac{T_1^2}{6}.$$

**$3^{rd}$  segment**

$$\text{Cycloidal: } m_{3,1} = T_2, \quad m_{4,1} = \frac{2T_3}{\pi}, \quad m_{11,1} = \frac{T_2^2}{2} + \left(\frac{2T_3}{\pi}\right)^2, \quad m_{12,1} = 0,$$

$$\text{Cubic: } m_{3,1} = T_2, \quad m_{4,1} = \frac{T_3}{2}, \quad m_{11,1} = \frac{T_2^2}{2}, \quad m_{12,1} = \frac{T_3^2}{3}.$$

 **$5^{th}$  segment**

$$\text{Cycloidal: } m_{5,2} = \frac{2T_5}{\pi}, \quad m_{6,2} = 0, \quad m_{13,2} = 0, \quad m_{14,2} = -\left(\frac{2T_5}{\pi}\right)^2,$$

$$\text{Cubic: } m_{5,2} = 0, \quad m_{6,2} = \frac{T_5}{2}, \quad m_{13,2} = 0, \quad m_{14,2} = \frac{T_5^2}{6}.$$

 **$7^{th}$  segment**

$$\text{Cycloidal: } m_{7,2} = T_6, \quad m_{8,2} = \frac{2T_7}{\pi}, \quad m_{15,2} = \frac{T_6^2}{2} + \left(\frac{2T_7}{\pi}\right)^2, \quad m_{16,2} = 0,$$

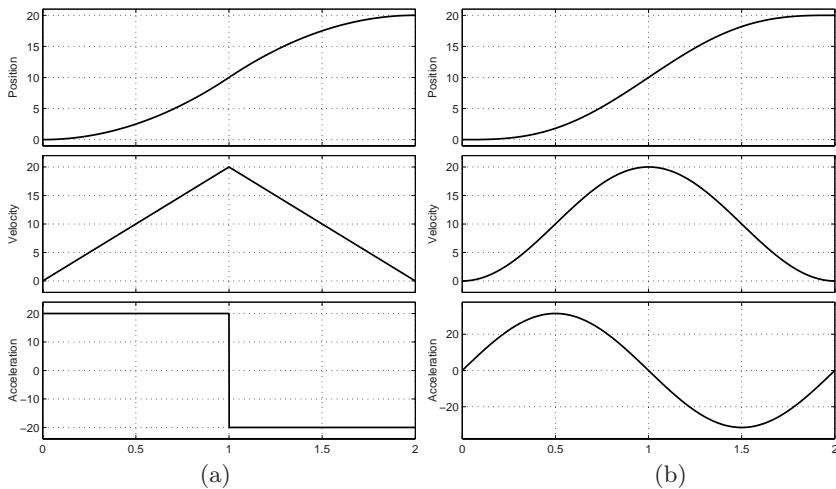
$$\text{Cubic: } m_{7,2} = T_6, \quad m_{8,2} = \frac{T_7}{2}, \quad m_{15,2} = \frac{T_6^2}{2}, \quad m_{16,2} = \frac{T_7^2}{3}.$$

Once these expressions are substituted in the above system, the parameters  $a_1$ ,  $a_2$  and  $k_{ij}$  may be found and the trajectory  $q(t)$  is completely defined. The general expressions of the parameters  $a_1$ ,  $a_2$ ,  $k_{ij}$  as functions of the variables  $m_{i,j}$ ,  $q_0$ ,  $q_7$  are reported in Appendix A.

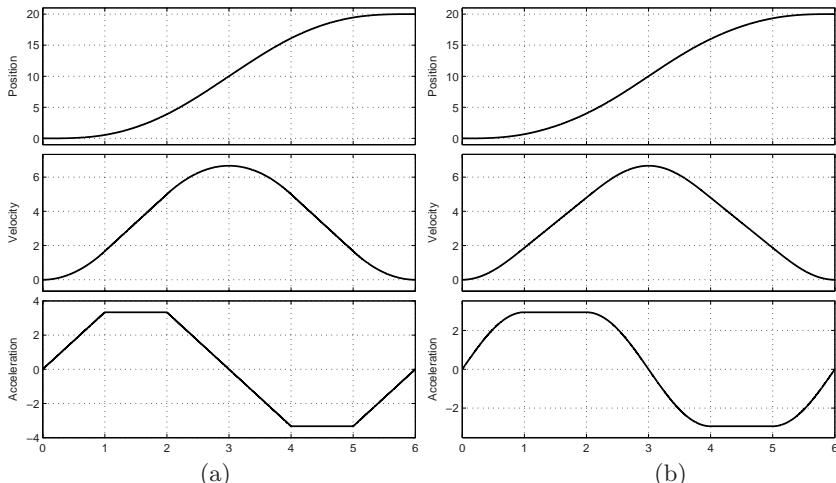
Note that several of the elementary and modified profiles described in previous sections can be obtained from the above equations. In order to compose more trajectories, it is sufficient to impose, besides the initial and final time instants, the initial and final position values and the temporal lengths of the intermediate segments, with the possibility of setting some of them to zero.

**Example 3.26** With reference to Fig. 3.52, a parabolic trajectory is obtained by setting to zero the duration of segments 1, 3, 4, 5, 7, i.e. by assigning  $t_1 = t_0$ ,  $t_5 = t_4 = t_3 = t_2$ ,  $t_7 = t_6$ ; a cycloidal trajectory is obtained if the durations of segments 2, 4, and 6 are zero (see Fig. 3.53), while a trajectory with (modified) trapezoidal acceleration profile is obtained by neglecting the fourth segment, Fig. 3.54. Obviously, the proper motion law (cycloidal or cubic) must be defined in each one of the remaining segments.  $\square$

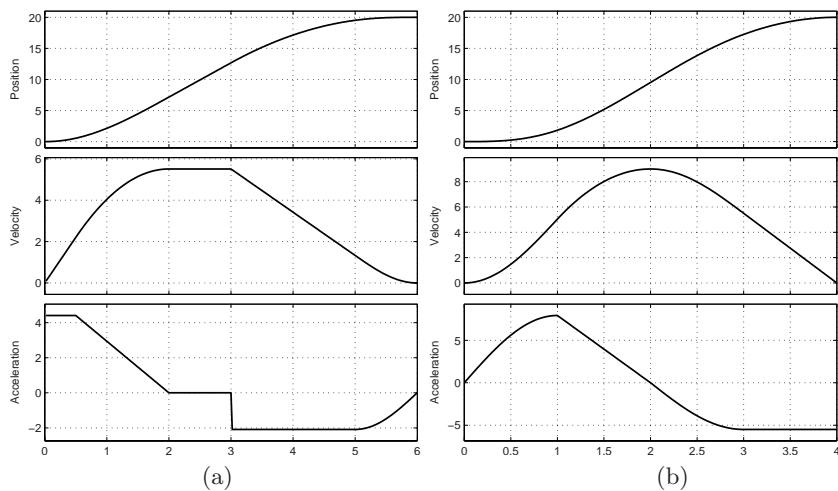
**Example 3.27** A further example is shown in Fig. 3.55, where two trajectories composed by several profiles are reported. The trajectory in Fig. 3.55(a) has a parabolic-cubic-constant-cycloidal profile, with  $T_2 = 0.5$ ,  $T_3 = 1.5$ ,  $T_4 = 1$ ,  $T_6 = 2$ ,  $T_7 = 1$  and  $v_0 = v_7 = 0$ , while the first and fifth segments are not present ( $T_1 = T_5 = 0$ ). The trajectory in Fig. 3.55(b) is a composition of cycloidal, cubic, cycloidal and parabolic profiles with  $T_2 = T_4 = T_7 = 0$ .  $\square$



**Fig. 3.53.** Trajectory with constant (a) and cycloidal (b) acceleration.



**Fig. 3.54.** Trajectory with a trapezoidal (a) and modified trapezoidal (b) acceleration profile.



**Fig. 3.55.** Composition of elementary trajectories: parabolic-cubic-constant-cycloidal (a); cycloidal-cubic-cycloidal-parabolic (b).

## Multipoint Trajectories

In this chapter, the problem of the definition of multipoint trajectories is addressed, i.e. of functions suitable for the interpolation or approximation of a set of given points  $(t_k, q_k)$ ,  $k = 0, \dots, n$ . In particular, the problem is discussed in the case of a single axis of motion: the more general problem related to 3D space is addressed in Chapter 8. Different approaches are presented: polynomial functions of proper degree, orthogonal and trigonometric polynomials, spline functions, and nonlinear filters able to generate in real time optimal trajectories satisfying given constraints on maximum velocity, acceleration, and jerk.

---

### 4.1 Interpolation by Polynomial Functions

The problem of computing a trajectory through  $n + 1$  points can be solved by means of a polynomial function of degree  $n$ :

$$q(t) = a_0 + a_1 t + \dots + a_n t^n. \quad (4.1)$$

As a matter of fact, given two points it is easy to determine the unique line joining them; similarly, it is possible to define a unique quadratic function through three points and, more generally, given the points  $(t_k, q_k)$ ,  $k = 0, \dots, n$  it exists a unique interpolating polynomial  $q(t)$  of degree  $n$ .

From a mathematical point of view, the solution of the interpolation problem of  $n + 1$  points can be obtained by solving a linear system of  $n + 1$  equations in  $n + 1$  unknowns (the coefficients  $a_k$  of the interpolating polynomial (4.1)). This method is based on the following algorithm. Given the points  $(t_k, q_k)$ ,  $k = 0, \dots, n$ , it is possible to build the vectors  $\mathbf{q}$ ,  $\mathbf{a}$ , and the so-called *Vandermonde* matrix  $\mathbf{T}$  as

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-1} \\ q_n \end{bmatrix} = \begin{bmatrix} 1 & t_0 & \cdots & t_0^n \\ 1 & t_1 & \cdots & t_1^n \\ & & \vdots & \\ 1 & t_{n-1} & \cdots & t_{n-1}^n \\ 1 & t_n & \cdots & t_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ a_n \end{bmatrix} = \mathbf{T} \mathbf{a}. \quad (4.2)$$

If  $t_{k+1} > t_k$ ,  $k = 0, \dots, n-1$ , the matrix  $\mathbf{T}$  is always invertible and, therefore, the coefficients  $a_k$  can be computed as

$$\mathbf{a} = \mathbf{T}^{-1} \mathbf{q}.$$

The advantages of using polynomial functions as (4.1) for the interpolation of  $n+1$  points are that:

1. The trajectory defined in this way crosses all the given points.
2. The interpolating function can be easily described since only  $n+1$  coefficients are needed.
3. The derivatives (of any order) of the function  $q(t)$  defined in this manner are continuous in the range  $[t_0, t_n]$ ; in particular, the  $n$ -th derivative is constant and therefore all the higher order derivatives are null.
4. The interpolating trajectory  $q(t)$  is unique.

On the other hand this method, conceptually simple, is rather inefficient from the computational point of view and may produce numerical errors for large values of  $n$ . As a matter of fact, the numerical solution of the system (4.2) suffers from an error which is approximatively equal to the error affecting the representation of the data (truncation errors, accuracy errors, etc.) multiplied by the condition number<sup>1</sup>  $\kappa$  of the Vandermonde matrix of the system. The value of  $\kappa$  is proportional to  $n$ , and therefore for relatively large numbers of

---

<sup>1</sup> The condition number of a matrix  $\mathbf{A}$  is by definition the ratio of the maximum and the minimum singular value of the matrix itself:  $\kappa = \frac{\sigma_{\max}}{\sigma_{\min}}$ , [18]. It is equivalent to the quantity

$$|\mathbf{A}| |\mathbf{A}^{-1}|$$

which provides an estimation of the accuracy of the result obtained from the matrix inversion. As a matter of fact, given a linear system expressed in matrix form as

$$\mathbf{A} \mathbf{x} = \mathbf{c}$$

where  $\mathbf{x}$  is the vector of the unknowns, it is possible to show that a perturbation  $\Delta \mathbf{A}$  produces an error  $\Delta \mathbf{x}$ , such that

$$\frac{\Delta \mathbf{x}}{\mathbf{x} + \Delta \mathbf{x}} \leq |\mathbf{A}| |\mathbf{A}^{-1}| \frac{\Delta \mathbf{A}}{\mathbf{A}} = \kappa \frac{\Delta \mathbf{A}}{\mathbf{A}}$$

that is the relative error in the solution vector is bounded by the relative error on the given matrix  $\mathbf{A}$  multiplied the condition number  $\kappa$ . As a consequence, in a computer with  $m$  decimal digits of accuracy, the quantity

$$m - \log_{10}(\kappa)$$

points the error in computing the parameters of the trajectory may result too large.

**Example 4.1** In order to compute the coefficients  $a_k$  of the polynomial of degree  $n$  interpolating the points  $(t_k, q_k)$  with

$$t_k = \frac{k}{n}, \quad k = 0, \dots, n$$

it is necessary to define the matrix  $\mathbf{T}$ . For  $n = 3$ , it results

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{3} & \frac{1}{9} & \frac{1}{27} \\ 1 & \frac{2}{3} & \frac{4}{9} & \frac{8}{27} \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad \kappa = \frac{\sigma_{\max}}{\sigma_{\min}} = \frac{2.5957}{0.02625} \simeq 99.$$

If  $n = 5$ , we have

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & \frac{1}{5} & \frac{1}{25} & \frac{1}{125} & \frac{1}{625} & \frac{1}{3125} \\ 1 & \frac{2}{5} & \frac{4}{25} & \frac{8}{125} & \frac{16}{625} & \frac{32}{3125} \\ 1 & \frac{3}{5} & \frac{9}{25} & \frac{27}{125} & \frac{81}{625} & \frac{243}{3125} \\ 1 & \frac{4}{5} & \frac{16}{25} & \frac{64}{125} & \frac{256}{625} & \frac{1024}{3125} \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad \kappa = \frac{\sigma_{\max}}{\sigma_{\min}} = \frac{3.339263}{0.0006781} \simeq 4924.$$

The condition number of  $\mathbf{T}$ , for different values of  $n$ , is reported in the following table.

$n$	2	3	4	5	10	15	20
$\kappa$	15.1	98.87	686.43	4924.37	$1.156 \cdot 10^8$	$3.122 \cdot 10^{12}$	$9.082 \cdot 10^{16}$

As shown in the table, the condition number increases with  $n$  and, as a consequence, numerical problems (inaccuracy of the solution) tied to this technique arise. For instance, in the Matlab environment (where a double precision representation of the numbers is adopted), in case  $n = 10$ , and with the vector of via-points

$$\mathbf{q} = [1, 0.943, 1.394, 2.401, 4.052, 6.507, 10.074, 15.359, 23.594, 37.231, 61]^T$$

---

indicates the number of decimal places we can expect to be correct in the result of a linear system, because of the matrix inversion. For instance, in the IEEE standard double precision representation, the numbers have about 16 decimal digits of accuracy (with 53 bits for the fraction), so if a matrix has a condition number of  $10^{10}$  only six digits will be correct in the result.

the maximum difference between the true values of the coefficients

$$\mathbf{a} = [1, -3, 24, 3, 5, 6, 7, -3, 5, 12, 4]^T$$

and the ones computed by inverting  $\mathbf{T}$  ( $\mathbf{a}' = \mathbf{T}^{-1}\mathbf{q}$ ) is

$$\Delta a_{max} = 2.887 \cdot 10^{-8} \quad (\approx \kappa \cdot 10^{-16})$$

being  $10^{-16}$  the accuracy of the Matlab number representation. If  $n = 20$ , and the vector  $\mathbf{q}$  has elements of the same order of magnitude of the previous ones, the maximum error is  $\Delta a_{max} = 25$  (note that in this case  $\kappa = 9.082 \cdot 10^{16}$ ).  $\square$

A different way for computing the coefficients of the interpolating polynomial  $q(t)$  is based on the well-known Lagrange formula:

$$\begin{aligned} q(t) &= \frac{(t - t_1)(t - t_2) \cdots (t - t_n)}{(t_0 - t_1)(t_0 - t_2) \cdots (t_0 - t_n)} q_0 + \frac{(t - t_0)(t - t_2) \cdots (t - t_n)}{(t_1 - t_0)(t_1 - t_2) \cdots (t_1 - t_n)} q_1 + \\ &\quad + \cdots + \frac{(t - t_0) \cdots (t - t_{n-1})}{(t_n - t_0)(t_n - t_1) \cdots (t_n - t_{n-1})} q_n. \end{aligned}$$

It is also possible to define recursive formulations, that allow a more efficient computation of the polynomial  $q(t)$ , such as the Neville algorithm, see [19] or [20]. Unfortunately also these techniques, although more efficient from a computational point of view, are affected by numerical problems for high values of  $n$ .

In addition to numerical problems, the use of a polynomial of degree  $n$  for interpolating  $n+1$  points has a number of other drawbacks, since:

1. The degree of the polynomial depends on the number of points and, for large values of  $n$ , the amount of calculations may be remarkable.
2. The variation of a single point  $(t_k, q_k)$  implies that all the coefficients of the polynomial must be recomputed.
3. The insertion of an additional point  $(t_{n+1}, q_{n+1})$  implies the adoption of a polynomial of higher degree  $(n+1)$  and the calculation of all the coefficients.
4. The resulting trajectories are usually characterized by pronounced ‘oscillations’, that are unacceptable in motion profiles for automatic machines.

Moreover, standard techniques for polynomial interpolation do not take into account further conditions on initial, final or intermediate velocities and accelerations. In this case, it is necessary to assume an higher order polynomial function and consider additional constraints on the polynomial coefficients. For example, in order to assign initial/final velocities and accelerations (i.e. 4 additional constraints) on a trajectory interpolating  $n+1$  given points, the polynomial must be of degree  $n+4$ , and the equation system providing the coefficients  $a_k$  becomes

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-1} \\ q_n \\ v_0 \\ a_0 \\ v_n \\ a_n \end{bmatrix} = \begin{bmatrix} 1 & t_0 & \cdots & t_0^{n+4} \\ 1 & t_1 & \cdots & t_1^{n+4} \\ & & \vdots & \\ 1 & t_{n-1} & \cdots & t_{n-1}^{n+4} \\ 1 & t_n & \cdots & t_n^{n+4} \\ 0 & 1 & 2t_0 & \cdots & (n+4)t_0^{n+3} \\ 0 & 0 & 2 & 6t_0 & \cdots & (n+4)(n+3)t_0^{n+2} \\ 0 & 1 & 2t_n & \cdots & (n+4)t_n^{n+3} \\ 0 & 0 & 2 & 6t_n & \cdots & (n+4)(n+3)t_n^{n+2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ a_n \\ a_{n+1} \\ a_{n+2} \\ a_{n+3} \\ a_{n+4} \end{bmatrix} = \mathbf{T} \mathbf{a}.$$

Obviously, also in this case the coefficients  $a_k$  can be computed as

$$\mathbf{a} = \mathbf{T}^{-1} \mathbf{q}.$$

In order to cope with the problems related to the interpolation by polynomials (bad numerical conditioning and oscillating behavior), other techniques can be adopted. In particular valid alternatives are the *orthogonal polynomials* and the *spline functions*, described in the following sections.

## 4.2 Orthogonal Polynomials

An orthogonal polynomial of degree  $m$  is defined as

$$q(t) = a_0 p_0(t) + a_1 p_1(t) + \cdots + a_m p_m(t) \quad (4.3)$$

where  $a_0, a_1, \dots, a_m$  are constant parameters and  $p_0(t), p_1(t), \dots, p_m(t)$  are polynomials of proper degree. The polynomials  $p_0(t), \dots, p_m(t)$  are called *orthogonal* because they enjoy the following properties

$$\begin{aligned} \gamma_{ji} &= \sum_{k=0}^n p_j(t_k) p_i(t_k) = 0, & \forall j, i : j \neq i \\ \gamma_{ii} &= \sum_{k=0}^n [p_i(t_k)]^2 \neq 0 \end{aligned} \quad (4.4)$$

where  $t_0, t_1, \dots, t_n$  are the instants in which the polynomials are orthogonal (i.e. satisfy these conditions). For example, if there are five points to be interpolated with a second degree orthogonal polynomial ( $m = 2$ ), one obtains

$$\begin{aligned} \sum_{k=0}^4 p_0(t_k) p_1(t_k) &= \sum_{k=0}^4 p_0(t_k) p_2(t_k) = \sum_{k=0}^4 p_1(t_k) p_2(t_k) = 0 \\ \sum_{k=0}^4 [p_0(t_k)]^2 &\neq 0, \quad \sum_{k=0}^4 [p_1(t_k)]^2 \neq 0, \quad \sum_{k=0}^4 [p_2(t_k)]^2 \neq 0. \end{aligned}$$

With orthogonal polynomials, it is possible to interpolate, or approximate with a prescribed tolerance,  $n + 1$  given points.

For the computation of the approximating polynomial, a “least squares” technique is adopted. For each point  $q_k$ , the error  $\epsilon_k$  is defined as

$$\epsilon_k = \left( q_k - \sum_{j=0}^m a_j p_j(t_k) \right), \quad k = 0, \dots, n$$

and then the total square error results

$$\mathcal{E}^2 = \sum_{k=0}^n \epsilon_k^2.$$

The parameters  $a_j$  are determined in order to minimize  $\mathcal{E}^2$ . Obviously, if  $\mathcal{E}^2 = 0$ , the interpolation by means of orthogonal polynomials is exact. Let us define

$$\begin{aligned} \delta_i &= \sum_{k=0}^n q_k p_i(t_k), & i &\in [0, \dots, m] \\ \gamma_{ji} &= \sum_{k=0}^n p_j(t_k) p_i(t_k), & j, i &\in [0, \dots, m]. \end{aligned}$$

From the minimization condition on  $\mathcal{E}^2$  one obtains

$$\frac{\partial \mathcal{E}^2}{\partial a_i} = 0, \quad i = 0, 1, \dots, m$$

that is

$$\begin{aligned} \frac{\partial}{\partial a_i} \left[ \sum_{k=0}^n \left( q_k - \sum_{j=0}^m a_j p_j(t_k) \right)^2 \right] &= \frac{\partial}{\partial a_i} \sum_{k=0}^n \left( \sum_{j=0}^m a_j p_j(t_k) \right)^2 + \\ &\quad - 2 \frac{\partial}{\partial a_i} \sum_{k=0}^n \left( q_k \sum_{j=0}^m a_j p_j(t_k) \right) \\ &= 2 \sum_{k=0}^n \sum_{j=0}^m a_j p_j(t_k) p_i(t_k) - 2 \sum_{k=0}^n q_k p_i(t_k) = 0 \end{aligned}$$

from which

$$\delta_i = \sum_{j=0}^m a_j \gamma_{ji}, \quad i = 0, 1, \dots, m$$

and

$$\begin{cases} \delta_0 = \gamma_{00} a_0 + \gamma_{01} a_1 + \dots + \gamma_{0m} a_m \\ \delta_1 = \gamma_{10} a_0 + \gamma_{11} a_1 + \dots + \gamma_{1m} a_m \\ \dots \\ \delta_m = \gamma_{m0} a_0 + \gamma_{m1} a_1 + \dots + \gamma_{mm} a_m. \end{cases}$$

This is a system of  $m + 1$  equations in the  $m + 1$  unknowns  $a_0, a_1, \dots, a_m$ . From the orthogonality conditions (4.4) one gets

$$\begin{cases} \delta_0 = \gamma_{00} a_0 \\ \delta_1 = \gamma_{11} a_1 \\ \delta_2 = \gamma_{22} a_2 \\ \dots \\ \delta_m = \gamma_{mm} a_m \end{cases}$$

that is

$$a_j = \frac{\delta_j}{\gamma_{jj}}.$$

The next problem is to define the orthogonal polynomials  $p_j(t)$  in order to satisfy additional criteria. For this purpose, several techniques have been developed, such as those proposed by G. E. Forsythe, [21], and by C. W. Clenshaw and J. G. Hayes, [22].

With the first technique, [21], the polynomials  $p_0(t), \dots, p_m(t)$  are computed by the following recursive equation

$$p_j(t) = (t - \alpha_j)p_{j-1}(t) - \beta_{j-1}p_{j-2}(t), \quad j = 1, \dots, m$$

where  $\alpha_j$  and  $\beta_{j-1}$  are proper constants, and  $j$  is the degree of the polynomial. By choosing  $p_0(t) = 1$  one obtains

$$\begin{cases} p_0(t) = 1 \\ p_1(t) = t p_0(t) - \alpha_1 p_0(t) \\ p_2(t) = t p_1(t) - \alpha_2 p_1(t) - \beta_1 p_0(t) \\ \vdots \\ p_j(t) = t p_{j-1}(t) - \alpha_j p_{j-1}(t) - \beta_{j-1} p_{j-2}(t). \end{cases}$$

The parameters  $\alpha_j$  and  $\beta_j$  must be selected in order to satisfy the orthogonality conditions:

$$\sum_{k=0}^n p_j(t_k) p_i(t_k) = 0, \quad j \neq i.$$

In particular,  $\alpha_j$  is computed by multiplying the expressions of  $p_j$  and  $p_{j-1}$  and summing them for the  $n + 1$  points:

$$\begin{aligned} \sum_{k=0}^n p_j(t_k) p_{j-1}(t_k) &= \sum_{k=0}^n t_k [p_{j-1}(t_k)]^2 - \alpha_j \sum_{k=0}^n [p_{j-1}(t_k)]^2 + \\ &\quad - \beta_{j-1} \sum_{k=0}^n p_{j-1}(t_k) p_{j-2}(t_k). \end{aligned}$$

Since the pairs  $(p_j, p_{j-1})$  and  $(p_{j-1}, p_{j-2})$  must be reciprocally orthogonal, the following equations descend

$$\alpha_j = \frac{\sum_{k=0}^n t_k [p_{j-1}(t_k)]^2}{\sum_{k=0}^n [p_{j-1}(t_k)]^2}, \quad \beta_j = \frac{\sum_{k=0}^n [p_j(t_k)]^2}{\sum_{k=0}^n [p_{j-1}(t_k)]^2}.$$

Finally, the coefficients  $a_0, a_1, \dots, a_m$  are computed as

$$a_j = \frac{\sum_{k=0}^n q_k p_j(t_k)}{\sum_{k=0}^n [p_j(t_k)]^2}.$$

**Example 4.2** Fig. 4.1 shows the behavior of a trajectory obtained with orthogonal polynomials of degree four ( $m = 4$ ) with the conditions

$$\begin{aligned} t_0 &= 0, & t_1 &= 1, & t_2 &= 3, & t_3 &= 7, & t_4 &= 8, & t_5 &= 10, \\ q_0 &= 2, & q_1 &= 3, & q_2 &= 5, & q_3 &= 6, & q_4 &= 8, & q_5 &= 9. \end{aligned}$$

In this case, the following polynomials are obtained

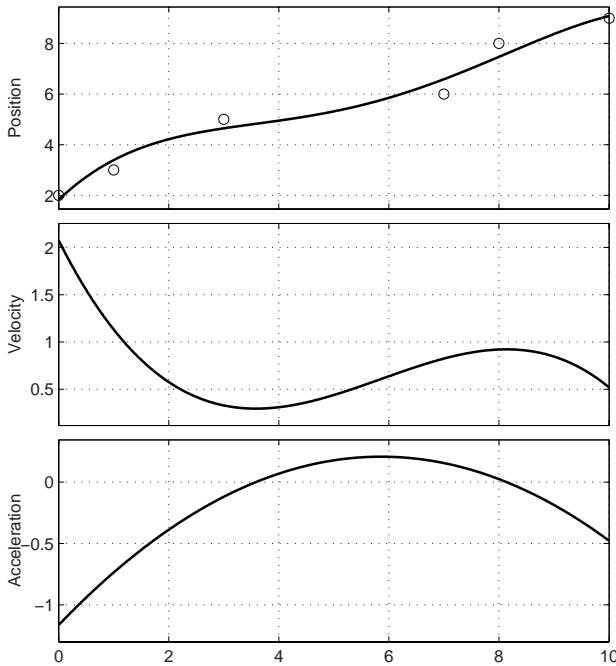
$$\begin{cases} p_0(t) = 1 \\ p_1(t) = -4.833 + t \\ p_2(t) = 9.8149 - 9.7203t + t^2 \\ p_3(t) = -28.5693 + 59.3370t - 15.3917t^2 + t^3 \\ p_4(t) = 38.9880 - 212.9051t + 121.8979t^2 - 20.0860t^3 + t^4 \end{cases}$$

with coefficients

$$a_0 = 5.5, \quad a_1 = 0.6579, \quad a_2 = -0.049, \quad a_3 = 0.0112, \quad a_4 = -0.0033.$$

Note that in this case the trajectory does not pass through the specified points.  $\square$

Often, in practice, it is necessary that the trajectory satisfies proper boundary conditions in terms of position, velocity and acceleration. In general, these conditions are not satisfied with the previous technique for the computation of the polynomials  $p_j(t)$ , as shown in Fig. 4.1, and velocity and acceleration discontinuities may be present at the initial and final points. The reason is that the given points are approximated, and the velocity and acceleration values are not specified. To solve this problem, different techniques can be



**Fig. 4.1.** Trajectory approximating 6 points with fourth degree orthogonal polynomials.

adopted, as the method illustrated below and based on a different choice of the initial polynomials, [22]. In order to interpolate a set of points with orthogonal polynomials so that the resulting trajectory crosses the initial and final points  $A = (t_a, q_a)$  and  $B = (t_b, q_b)$ , a normalized variable  $\tau$  is considered

$$\tau = \frac{t - t_a}{t_b - t_a}, \quad \tau \in [0, 1]$$

and the trajectory  $q(t)$  is transformed according to

$$\tilde{q}(\tau) = q(t) - q_a(1 - \tau) - q_b\tau.$$

In this manner, the conditions at the points  $A$  and  $B$  result

$$\text{at } t = t_a \ (\tau = 0), \quad \tilde{q}(0) = q(t_a) - q_a, \quad \rightarrow \quad q(t_a) = q_a + \tilde{q}(0)$$

$$\text{at } t = t_b \ (\tau = 1), \quad \tilde{q}(1) = q(t_b) - q_b, \quad \rightarrow \quad q(t_b) = q_b + \tilde{q}(1).$$

These conditions are satisfied by imposing  $\tilde{q} = 0$  at  $t = t_a, t_b$  ( $\tau = 0, 1$ ). For this purpose, one can incorporate the term  $\tau(1 - \tau)$  in the function  $\tilde{q}$ .

By considering the orthogonal polynomial

$$\tilde{q}(\tau) = a_0 p_0(\tau) + a_1 p_1(\tau) + \cdots + a_m p_m(\tau)$$

the simplest manner to include the term  $\tau(1 - \tau)$  is to compute  $p_j(\tau)$  as

$$\begin{cases} p_0(\tau) = \tau(1 - \tau) \\ p_1(\tau) = \tau p_0(\tau) - \alpha_1 p_0(\tau) \\ p_2(\tau) = \tau p_1(\tau) - \alpha_2 p_1(\tau) - \beta_1 p_0(\tau) \\ \dots \\ p_m(\tau) = \tau p_{m-1}(\tau) - \alpha_m p_{m-1}(\tau) - \beta_{m-1} p_{m-2}(\tau). \end{cases}$$

Similarly, if a null velocity is required in  $A$ , the transformation

$$\tilde{q}(\tau) = q(t) - q_a(1 - \tau^2) - q_b\tau^2$$

is used and the initial polynomial is chosen as  $p_0(\tau) = \tau^2(1 - \tau)$ . If a null acceleration is specified in  $A$ , one defines

$$\tilde{q}(\tau) = q(t) - q_a(1 - \tau^3) - q_b\tau^3$$

and the first polynomial  $p_0(\tau) = \tau^3(1 - \tau)$ .

**Example 4.3** Fig. 4.2 reports the position, velocity and acceleration of an orthogonal polynomials trajectory with crossing conditions for the initial and final points. The points are the same as in the previous example. In this case, the orthogonal polynomials result

$$\begin{cases} p_0(\tau) = \tau - \tau^2 \\ p_1(\tau) = -0.5364\tau + 1.5364\tau^2 - \tau^3 \\ p_2(\tau) = 0.1848\tau - 1.1724\tau^2 + 1.9875\tau^3 - \tau^4 \\ p_3(\tau) = -0.0520\tau + 0.5513\tau^2 - 1.8255\tau^3 + 2.3261\tau^4 - \tau^5 \\ p_4(\tau) = 0.0168\tau - 0.2858\tau^2 + 1.4590\tau^3 - 3.0900\tau^4 + 2.9000\tau^5 - \tau^6 \end{cases}$$

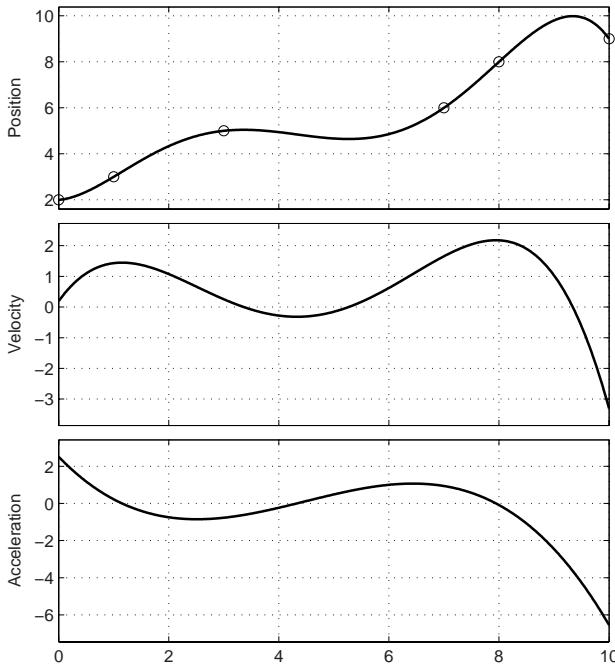
with coefficients

$$a_0 = 0.7465, \quad a_1 = -10.1223, \quad a_2 = 28.1252, \quad a_3 = 317.4603, \quad a_4 = 0.$$

□

**Example 4.4** Fig. 4.3 shows the position, velocity and acceleration of a trajectory with null initial velocity. The points are the same as in the previous examples, but in this case the orthogonal polynomials are

$$\begin{cases} p_0(\tau) = \tau^2 - \tau^3 \\ p_1(\tau) = -0.7001\tau^2 + 1.7001\tau^3 - \tau^4 \\ p_2(\tau) = 0.2546\tau^2 - 1.3464\tau^3 + 2.0918\tau^4 - \tau^5 \\ p_3(\tau) = -0.1501\tau^2 + 1.0937\tau^3 - 2.6874\tau^4 + 2.7438\tau^5 - \tau^6 \\ p_4(\tau) = 0.0168\tau^2 - 0.2858\tau^3 + 1.4590\tau^4 - 3.0900\tau^5 + 2.9000\tau^6 - \tau^7 \end{cases}$$



**Fig. 4.2.** Position, velocity and acceleration of an orthogonal polynomials trajectory ( $m = 4$ ) with the crossing condition for the initial and final points.

with coefficients

$$a_0 = 10.3710, \quad a_1 = -54.7230, \quad a_2 = 334.9531, \quad a_3 = -111.9615, \quad a_4 = 0.$$

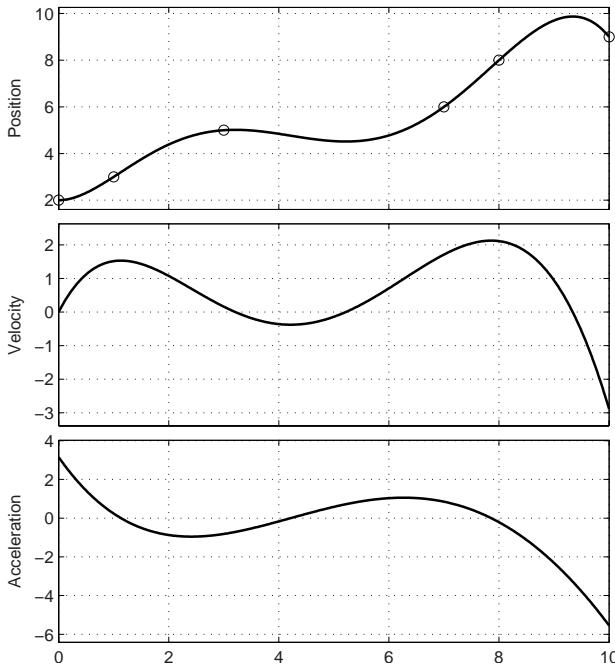
□

**Example 4.5** Fig. 4.4 shows position, velocity and acceleration of an orthogonal polynomials trajectory with null initial acceleration. The points are the same as in the previous examples. The orthogonal polynomials are

$$\left\{ \begin{array}{l} p_0(\tau) = \tau^3 - \tau^4 \\ p_1(\tau) = -0.7422\tau^3 + 1.7422\tau^4 - \tau^5 \\ p_2(\tau) = 0.3663\tau^3 - 1.6099\tau^4 + 2.2436\tau^5 - \tau^6 \\ p_3(\tau) = -0.1667\tau^3 + 1.1728\tau^4 - 2.8033\tau^5 + 2.7972\tau^6 - \tau^7 \\ p_4(\tau) = 0.0168\tau^3 - 0.2858\tau^4 + 1.4590\tau^5 - 3.0900\tau^6 + 2.9000\tau^7 - \tau^8 \end{array} \right.$$

and the coefficients

$$a_0 = 21.7, \quad a_1 = -134.8, \quad a_2 = 853.7, \quad a_3 = -9390, \quad a_4 = 0.$$



**Fig. 4.3.** Position, velocity and acceleration of an orthogonal polynomials trajectory ( $m = 4$ ) with null initial velocity.

□

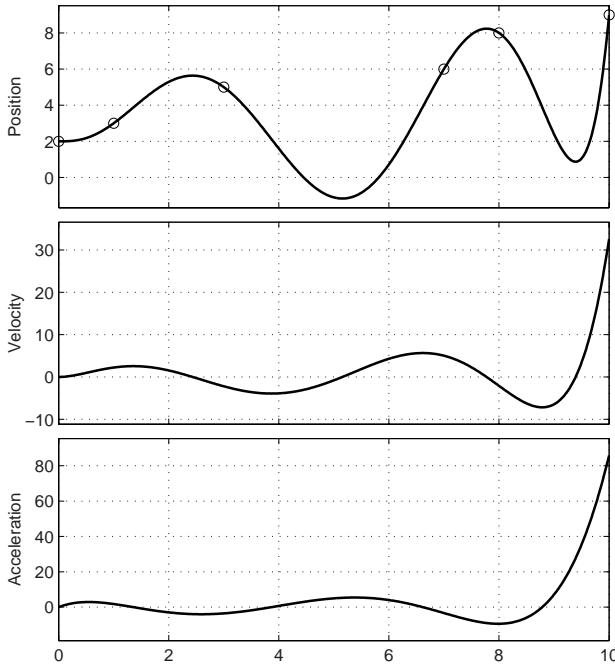
From the above example, one can notice that the coefficient  $a_4$  is always null. Therefore, in these cases the most appropriate degree for the polynomial satisfying the given conditions is three ( $m = 3$ ).

Another important class of orthogonal polynomials consists of the *Chebyshev polynomials* [23, 24], defined by

$$r_j(t) = \cos(j\phi), \quad \phi = \cos^{-1}(t), \quad -1 \leq t \leq 1$$

for  $j = 0, 1, 2, \dots, m$ . For example, if  $m = 4$ , one obtains

$$\begin{cases} r_0(t) = 1 \\ r_1(t) = t \\ r_2(t) = 2t^2 - 1 \\ r_3(t) = 4t^3 - 3t \\ r_4(t) = 8t^4 - 8t^2 + 1. \end{cases}$$



**Fig. 4.4.** Position, velocity and acceleration of an orthogonal polynomials trajectory ( $m = 4$ ) with null initial acceleration.

These polynomials have some important properties:

1. They can be computed recursively as

$$r_0(t) = 1, \quad r_1(t) = t, \quad r_{j+1}(t) = 2t r_j(t) - r_{j-1}(t), \quad j \geq 1.$$

2. They are symmetric, i.e.  $r_j(-t) = (-1)^j r_j(t)$ .
3. The coefficient of the term with the highest degree is null for  $j = 0$  and equal to  $2^{j-1}$  for  $j \geq 1$ .
4. The polynomial  $r_j(t)$  has  $j$  roots in  $[-1, 1]$  given by

$$t_k = \cos\left(\frac{\pi}{2} \frac{2k+1}{j}\right), \quad k = 0, 1, 2, \dots, j-1.$$

Moreover,  $r_j(t)$  has  $j+1$  maximum/minimum points, given by

$$t_k^m = \cos \frac{k\pi}{j}, \quad r_j(t_k^m) = (-1)^k, \quad k = 0, 1, 2, \dots, j.$$

5. The polynomials are orthogonal, i.e. given two functions  $f$  and  $g$  and defined the product as

$$(f, g) = \sum_{k=0}^m f(t_k)g(t_k)$$

where  $t_k$  are the roots of  $r_{m+1}(t)$ , then for  $0 \leq i \leq m$ ,  $0 \leq l \leq m$  one gets

$$(r_i, r_j) = \begin{cases} 0, & i \neq l \\ (m+1)/2, & i = l \neq 0 \\ m+1, & i = l = 0. \end{cases}$$

6. Among all the monic  $j$ -th degree polynomials, the polynomial  $2^{1-j}r_j$  has the smallest maximum norm in  $[-1, 1]$ , given by  $2^{1-j}$  (minimax property).

In conclusion, orthogonal polynomials provide a very flexible tool, based on the least squares approach, for the definition of a trajectory interpolating a sequence of points. On the other hand, the formulation of the polynomials obtained in this manner is not very efficient from the computational point of view. However, once the coefficients  $a_i$  and the polynomials  $p_j(t)$  have been determined, it is very simple to convert the expression (4.3) to the standard form

$$q(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_m t^m.$$

### 4.3 Trigonometric Polynomials

When the trajectory represents a periodic motion (and consequently  $q(t+T) = q(t)$ ), it may be convenient to adopt the so called trigonometric polynomials [25, 26], i.e.

$$q(t) = a_0 + \sum_{k=1}^m a_k \cos\left(k \frac{2\pi t}{T}\right) + \sum_{k=1}^m b_k \sin\left(k \frac{2\pi t}{T}\right)$$

whose coefficients are determined by imposing interpolation conditions on the via-points. Therefore, given a set of points  $q_k$ ,  $k = 0, \dots, n$  (with  $q_0 = q_n$ ) to be interpolated at time instants  $t_k$ ,  $k = 0, \dots, n$  (without loss of generality it is assumed  $t_0 = 0$ ), the degree  $m$  of the trigonometric polynomial is chosen in a such way that  $2m+1 = n$  (an even number of points is therefore required), and it is assumed  $T = t_n$ . Then, the following system of  $2m+1$  equations in the unknowns  $a_k$ ,  $b_k$  is set up

$$\begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-2} \\ q_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & c_1(t_0) & s_1(t_0) & \cdots & c_m(t_0) & s_m(t_0) \\ 1 & c_1(t_1) & s_1(t_1) & \cdots & c_m(t_1) & s_m(t_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & c_1(t_{n-2}) & s_1(t_{n-2}) & \cdots & c_m(t_{n-2}) & s_m(t_{n-2}) \\ 1 & c_1(t_{n-1}) & s_1(t_{n-1}) & \cdots & c_m(t_{n-1}) & s_m(t_{n-1}) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ a_m \\ b_m \end{bmatrix} \quad (4.5)$$

where

$$c_k(t) = \cos\left(k \frac{2\pi t}{T}\right), \quad s_k(t) = \sin\left(k \frac{2\pi t}{T}\right).$$

The coefficients of the polynomial are computed by solving (4.5). It is worth noticing that the trajectory is periodic by construction, and it is not necessary to impose the continuity of the trajectory derivatives (velocity, acceleration, jerk, etc.) at the boundaries (the so-called *cyclic conditions*). In particular the trajectory will be  $C^\infty$  continuous, that is continuous for any order of derivative.

The same trajectory can be written in a form similar to the Lagrange formula for polynomial interpolation, which does not require the inversion of the matrix in (4.5)

$$q(t) = \sum_{k=0}^n \left( q_k \prod_{j=0, j \neq k}^n \frac{\sin(\frac{\pi}{T}(t - t_j))}{\sin(\frac{\pi}{T}(t_k - t_j))} \right). \quad (4.6)$$

**Example 4.6** The trigonometric polynomial trajectory passing through the points

$$\begin{aligned} t_0 &= 0, & t_1 &= 4, & t_2 &= 6, & t_3 &= 8, & t_4 &= 9, & t_5 &= 11, & t_6 &= 15, & t_7 &= 17, & t_8 &= 19, & t_9 &= 20, \\ q_0 &= 2, & q_1 &= 3, & q_2 &= 3, & q_3 &= 2, & q_4 &= 2, & q_5 &= 2, & q_6 &= 3, & q_7 &= 4, & q_8 &= 5, & q_9 &= 2, \end{aligned}$$

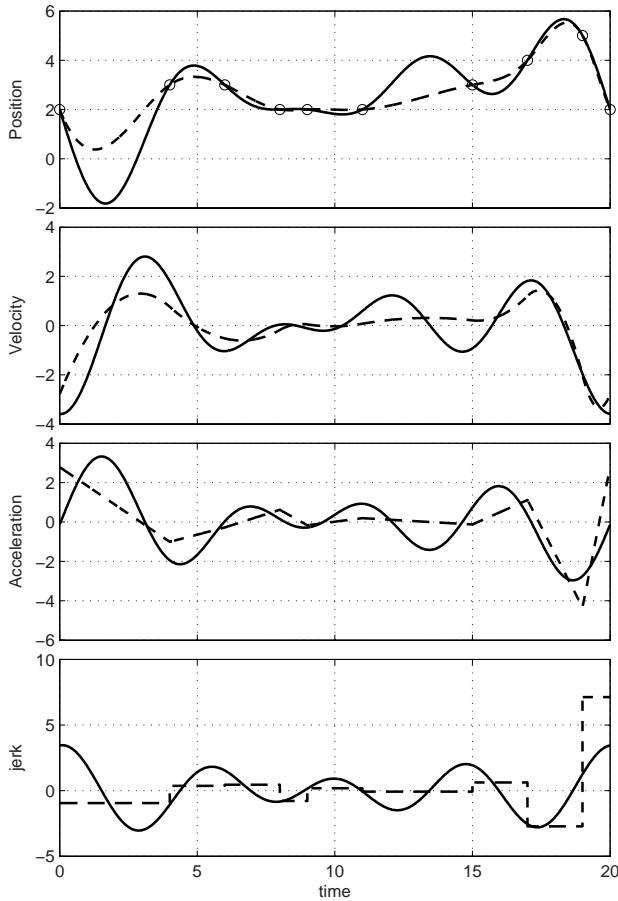
is shown in Fig. 4.5. The resulting parameters are

$$\begin{aligned} a_0 &= 2.53, & a_1 &= -0.21, & a_2 &= -0.73, & a_3 &= 0.30, & a_4 &= 0.11, \\ b_1 &= -1.08, & b_2 &= -1.03, & b_3 &= -1.47, & b_4 &= -0.96. \end{aligned}$$

□

If compared with an algebraic spline trajectory (see the following Section 4.4), the trigonometric polynomial shows more pronounced oscillations, with larger speeds and accelerations, but on the other hand all the derivatives are continuous also between the last and the first via-point (*periodic conditions*).

Similarly to algebraic splines, it is possible to define trigonometric splines, obtained by joining  $n$  trigonometric polynomial segments of proper degree, and by guaranteeing that their derivatives, up to a desired order, agree where the segments abut [27, 28, 29]. However, practical applications have demonstrated that in general algebraic splines are superior to trigonometric splines because accelerations and jerks result smaller, see Fig. 4.5, [30].

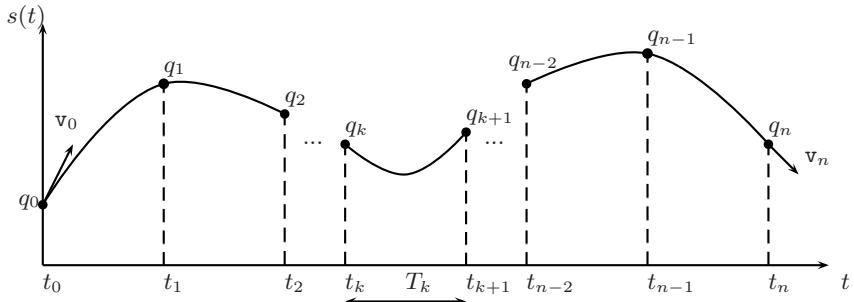


**Fig. 4.5.** Interpolation of a set of via-points by means of a trigonometric polynomial (solid) compared with a standard cubic spline (dashed).

## 4.4 Cubic Splines

When  $n + 1$  points are given, in lieu of a unique interpolating polynomial of degree  $n$  it is possible to use  $n$  polynomials of degree  $p$  (usually lower), each one defining a segment of the trajectory. The overall function  $s(t)$  defined in this manner is called *spline*<sup>2</sup> of degree  $p$ . The value of  $p$  is chosen according to the desired degree of continuity of the spline. For instance, in order to obtain the continuity of velocities and accelerations at the time instants  $t_k$ , where the transition between two consecutive segments occurs, it is possible to assume a polynomial of degree  $p = 3$  (cubic polynomial)

<sup>2</sup> *Spline*: the term has been introduced by I.J. Schoenberg, who exploited cubic splines to approximate the French curve used by designers to trace curves [31].



**Fig. 4.6.** A spline trajectory through  $n + 1$  points.

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3.$$

The overall function is given by

$$s(t) = \{q_k(t), t \in [t_k, t_{k+1}], k = 0, \dots, n - 1\},$$

$$q_k(t) = a_{k0} + a_{k1}(t - t_k) + a_{k2}(t - t_k)^2 + a_{k3}(t - t_k)^3.$$

In this way, it is necessary to compute 4 coefficients for each polynomial. Since  $n$  polynomials are necessary for the definition of a trajectory through  $n + 1$  points, the total number of coefficients to be determined is  $4n$ . In order to solve this problem, the following conditions must be considered:

- $2n$  conditions for the interpolation of the given points, since each cubic function must cross the points at its extremities.
- $n - 1$  conditions for the continuity of the velocities at the transition points.
- $n - 1$  conditions for the continuity of the accelerations at the transition points.

In this way, there are  $2n + 2(n - 1)$ , conditions and therefore the remaining degrees of freedom are  $4n - 2n - 2(n - 1) = 2$ . Then, two additional constraints must be imposed in order to compute the spline. Among the possible choices, one can assign:

1. The initial and final velocity  $\dot{s}(t_0) = v_0$ ,  $\dot{s}(t_n) = v_n$ , see Fig. 4.6.
2. The initial and final acceleration  $\ddot{s}(t_0)$ ,  $\ddot{s}(t_n)$  (these conditions are generally referred to as *natural*).
3. The conditions  $\dot{s}(t_0) = \dot{s}(t_n)$ ,  $\ddot{s}(t_0) = \ddot{s}(t_n)$ ; these conditions are usually called *cyclic* and are used when it is necessary to define a periodic spline, with period  $T = t_n - t_0$ .
4. The continuity of the jerk at time instants  $t_1, t_{n-1}$ :

$$\frac{d^3 s(t)}{dt^3} \Big|_{t=t_1^-} = \frac{d^3 s(t)}{dt^3} \Big|_{t=t_1^+}, \quad \frac{d^3 s(t)}{dt^3} \Big|_{t=t_{n-1}^-} = \frac{d^3 s(t)}{dt^3} \Big|_{t=t_{n-1}^+}.$$

In general, a spline is characterized by the following properties:

1.  $[n(p + 1)]$  parameters are sufficient for the definition of a trajectory  $s(t)$  of degree  $p$ , interpolating the given points  $(t_k, q_k)$ ,  $k = 0, \dots, n$ .
2. Given  $n + 1$  points, and given the boundary conditions, the interpolating spline  $s(t)$  of degree  $p$  is univocally determined.
3. The degree  $p$  of the polynomials used to construct the spline does not depend on the number of data points.
4. The function  $s(t)$  has continuous derivatives up to the order  $(p - 1)$ .
5. By assuming the conditions  $\ddot{s}(t_0) = \ddot{s}(t_n) = 0$ , the cubic spline is, among all the functions  $f(t)$  interpolating the given points and with continuous first and second derivatives, the function which minimizes the functional

$$J = \int_{t_0}^{t_n} \left( \frac{d^2 f(t)}{dt^2} \right)^2 dt$$

that can be interpreted as a sort of deformation energy, proportional to the curvature of  $f(t)$ .

In fact, let  $s(t)$  be a cubic spline and  $f(t) \in C^2[t_0, t_n]$  a generic function, continuous with continuous first and second derivatives over  $[t_0, t_n]$ . The function

$$E = \int_{t_0}^{t_n} \left( f^{(2)}(t) - s^{(2)}(t) \right)^2 dt$$

is clearly always positive or null, i.e.  $E \geq 0$ . Then

$$\begin{aligned} E &= \int_{t_0}^{t_n} \left( f^{(2)}(t) - s^{(2)}(t) \right)^2 dt \\ &= \int_{t_0}^{t_n} \left( f^{(2)}(t) \right)^2 dt - 2 \int_{t_0}^{t_n} f^{(2)}(t) s^{(2)}(t) dt + \int_{t_0}^{t_n} \left( s^{(2)}(t) \right)^2 dt \\ &= \int_{t_0}^{t_n} \left( f^{(2)}(t) \right)^2 dt - \int_{t_0}^{t_n} \left( s^{(2)}(t) \right)^2 dt + 2 \int_{t_0}^{t_n} s^{(2)}(t) \left( s^{(2)}(t) - f^{(2)}(t) \right) dt. \end{aligned}$$

By assigning null initial and final accelerations  $s^{(2)}(t_0) = s^{(2)}(t_n) = 0$ , and by considering the condition that the functions  $s(t)$  and  $f(t)$  must interpolate the given points, that is  $s(t_k) = f(t_k)$ ,  $k = 0, \dots, n$ , and the fact that the jerk  $s^{(3)}(t)$  of the cubic spline is piecewise constant, one obtains

$$\begin{aligned}
E &= 2 \int_{t_0}^{t_n} s^{(2)}(t) \left( s^{(2)}(t) - f^{(2)}(t) \right) dt \\
&= \left[ s^{(2)}(t) \left( s^{(1)}(t) - f^{(1)}(t) \right) \right]_{t_0}^{t_n} - \int_{t_0}^{t_n} s^{(3)}(t) \left( s^{(1)}(t) - f^{(1)}(t) \right) dt \\
&= - \sum_{k=0}^{n-1} s^{(3)}(t_k) \int_{t_k}^{t_{k+1}} \left( s^{(1)}(t) - f^{(1)}(t) \right) dt \\
&= - \sum_{k=0}^{n-1} s^{(3)}(t_k) \left[ s(t) - f(t) \right]_{t_k}^{t_{k+1}} \\
&= 0.
\end{aligned}$$

Therefore

$$E = \int_{t_0}^{t_n} \left( f^{(2)}(t) \right)^2 dt - \int_{t_0}^{t_n} \left( s^{(2)}(t) \right)^2 dt \geq 0$$

and then

$$\int_{t_0}^{t_n} \left( f^{(2)}(t) \right)^2 dt \geq \int_{t_0}^{t_n} \left( s^{(2)}(t) \right)^2 dt.$$

Therefore, the function  $f(t)$  which minimizes the functional  $J$  (and for which  $E = 0$ , i.e.  $f(t) = s(t)$ ) is the cubic spline with zero conditions on the initial and final acceleration. With these conditions, the spline is called *natural spline*.

#### 4.4.1 Computation of the coefficients for assigned initial and final velocities

For the definition of a trajectory for an automatic machine, the condition on the continuity of the velocity profile is of fundamental importance. For this reason, a typical choice for the computation of the spline is to assign the initial and final velocities  $v_0$  and  $v_n$ . Therefore, given the points  $(t_k, q_k)$ ,  $k = 0, \dots, n$  and the boundary conditions on the velocity  $v_0, v_n$ , the goal is to determine the function

$$\begin{aligned}
s(t) &= \{q_k(t), t \in [t_k, t_{k+1}], k = 0, \dots, n-1\}, \\
q_k(t) &= a_{k0} + a_{k1}(t - t_k) + a_{k2}(t - t_k)^2 + a_{k3}(t - t_k)^3
\end{aligned}$$

with the conditions

$$\begin{aligned}
q_k(t_k) &= q_k, \quad q_k(t_{k+1}) = q_{k+1}, \quad k = 0, \dots, n-1 \\
\dot{q}_k(t_{k+1}) &= \dot{q}_{k+1}(t_{k+1}) = v_{k+1}, \quad k = 0, \dots, n-2 \\
\ddot{q}_k(t_{k+1}) &= \ddot{q}_{k+1}(t_{k+1}), \quad k = 0, \dots, n-2 \\
\dot{q}_0(t_0) &= v_0, \quad \dot{q}_{n-1}(t_n) = v_n.
\end{aligned}$$

The coefficients  $a_{k,i}$  can be computed with the following algorithm.

If the velocities  $v_k$ ,  $k = 1, \dots, n - 1$ , in the intermediate points were known, for each cubic polynomial it would be possible to write

$$\begin{cases} q_k(t_k) = a_{k0} & = q_k \\ \dot{q}_k(t_k) = a_{k1} & = v_k \\ q_k(t_{k+1}) = a_{k0} + a_{k1}T_k + a_{k2}T_k^2 + a_{k3}T_k^3 = q_{k+1} \\ \dot{q}_k(t_{k+1}) = a_{k1} + 2a_{k2}T_k + 3a_{k3}T_k^2 & = v_{k+1} \end{cases} \quad (4.7)$$

being  $T_k = t_{k+1} - t_k$ . By solving this system one would obtain the following coefficients

$$\begin{cases} a_{k,0} = q_k \\ a_{k,1} = v_k \\ a_{k,2} = \frac{1}{T_k} \left[ \frac{3(q_{k+1} - q_k)}{T_k} - 2v_k - v_{k+1} \right] \\ a_{k,3} = \frac{1}{T_k^2} \left[ \frac{2(q_k - q_{k+1})}{T_k} + v_k + v_{k+1} \right]. \end{cases} \quad (4.8)$$

On the other hand, the velocities  $v_1, \dots, v_{n-1}$  in the intermediate points are not known, and therefore they must be computed. For this purpose, the continuity conditions on the acceleration in the intermediate points are considered:

$$\ddot{q}_k(t_{k+1}) = 2a_{k,2} + 6a_{k,3}T_k = 2a_{k+1,2} = \ddot{q}_{k+1}(t_{k+1}), \quad k = 0, \dots, n - 2.$$

From these conditions, by taking into account the expression of the parameters  $a_{k,2}$ ,  $a_{k,3}$ ,  $a_{k+1,2}$  and multiplying by  $(T_k T_{k+1})/2$ , after simple manipulations one gets

$$T_{k+1}v_k + 2(T_{k+1} + T_k)v_{k+1} + T_kv_{k+2} = \frac{3}{T_k T_{k+1}} [T_k^2(q_{k+2} - q_{k+1}) + T_{k+1}^2(q_{k+1} - q_k)] \quad (4.9)$$

for  $k = 0, \dots, n - 2$ .

These relations can be rewritten in matrix form as  $\mathbf{A}'\mathbf{v}' = \mathbf{c}'$ , with

$$\mathbf{A}' = \begin{bmatrix} T_1 & 2(T_0 + T_1) & T_0 & 0 & \cdots & 0 \\ 0 & T_2 & 2(T_1 + T_2) & T_1 & & \vdots \\ \vdots & & & \ddots & & \\ & & & T_{n-2} & 2(T_{n-3} + T_{n-2}) & T_{n-3} & 0 \\ 0 & \cdots & & 0 & T_{n-1} & 2(T_{n-2} + T_{n-1}) & T_{n-2} \end{bmatrix}$$

$$\mathbf{v}' = [v_0, v_1, \dots, v_{n-1}, v_n]^T, \quad \mathbf{c}' = [c_0, c_1, \dots, c_{n-3}, c_{n-2}]^T$$

where the constant terms  $c_k$  depend only on the intermediate positions and on the time duration  $T_k$  of the spline segments, which are known. Since the velocities  $v_0$  and  $v_n$  are also known, it is possible to eliminate the corresponding columns of matrix  $\mathbf{A}'$  and obtain

$$\begin{bmatrix} 2(T_0 + T_1) & T_0 & 0 & \cdots & 0 \\ T_2 & 2(T_1 + T_2) & T_1 & 0 & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & & & T_{n-2} & 2(T_{n-3} + T_{n-2}) \\ 0 & \cdots & 0 & T_{n-1} & 2(T_{n-2} + T_{n-1}) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{3}{T_0 T_1} [T_0^2(q_2 - q_1) + T_1^2(q_1 - q_0)] - T_1 v_0 \\ \frac{3}{T_1 T_2} [T_1^2(q_3 - q_2) + T_2^2(q_2 - q_1)] \\ \vdots \\ \frac{3}{T_{n-3} T_{n-2}} [T_{n-3}^2(q_{n-1} - q_{n-2}) + T_{n-2}^2(q_{n-2} - q_{n-3})] \\ \frac{3}{T_{n-2} T_{n-1}} [T_{n-2}^2(q_n - q_{n-1}) + T_{n-1}^2(q_{n-1} - q_{n-2})] - T_{n-2} v_n \end{bmatrix} \quad (4.10)$$

that is

$$\mathbf{A}(\mathbf{T}) \mathbf{v} = \mathbf{c}(\mathbf{T}, \mathbf{q}, \mathbf{v}_0, \mathbf{v}_n) \quad (4.11)$$

where  $\mathbf{T} = [T_0, T_1, \dots, T_{n-1}]^T$ ,  $\mathbf{q} = [q_0, q_1, \dots, q_n]^T$ .

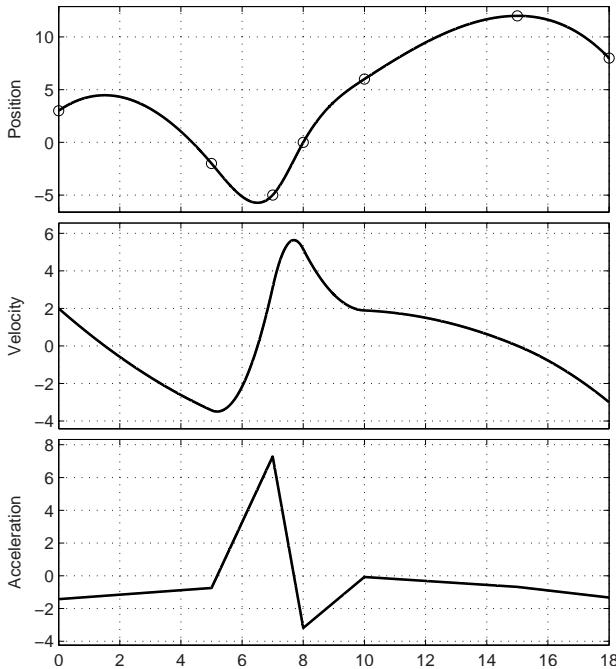
The  $(n-1) \times (n-1)$  matrix  $\mathbf{A}$  has a diagonal dominant structure, and therefore it is always invertible if  $T_k > 0$  ( $|a_{kk}| > \sum_{j \neq k} |a_{kj}|$ ). Moreover, being  $\mathbf{A}$  tridiagonal, computationally efficient techniques are available for its inversion, see Appendix A.5. Once the inverse of  $\mathbf{A}$  has been computed, the velocities  $v_1, \dots, v_{n-1}$  can be calculated from  $\mathbf{v} = \mathbf{A}^{-1} \mathbf{c}$  and therefore the problem is solved: the spline coefficients are obtained with (4.8).

**Example 4.7** Fig. 4.7 shows a spline computed according to the above algorithm. The trajectory is defined by the following points

$$\begin{aligned} t_0 &= 0, & t_1 &= 5, & t_2 &= 7, & t_3 &= 8, & t_4 &= 10, & t_5 &= 15, & t_6 &= 18, \\ q_0 &= 3, & q_1 &= -2, & q_2 &= -5, & q_3 &= 0, & q_4 &= 6, & q_5 &= 12, & q_6 &= 8, \end{aligned}$$

with initial velocity  $v_0 = 2$  and final velocity  $v_6 = -3$ . The resulting matrix  $\mathbf{A}$  and vector  $\mathbf{c}$  are

$$\mathbf{A} = \begin{bmatrix} 14 & 5 & 0 & 0 & 0 \\ 1 & 6 & 2 & 0 & 0 \\ 0 & 2 & 6 & 1 & 0 \\ 0 & 0 & 5 & 14 & 2 \\ 0 & 0 & 0 & 3 & 16 \end{bmatrix} \quad \mathbf{c} = [-32.5, 25.5, 39, 52.2, 5.8]^T.$$



**Fig. 4.7.** Spline trajectory with constraints on initial and final velocities.

The velocities in the intermediate points result

$$\mathbf{v} = [-3.43, \quad 3.10, \quad 5.10, \quad 1.88, \quad 0.008]^T.$$

The expression of the spline is therefore

$$s(t) = \begin{cases} 3 + 2t - 0.71t^2 + 0.02t^3, & \text{if } 0 \leq t < 5 \\ -2 - 3.4(t-5) - 0.37(t-5)^2 + 0.66(t-5)^3, & \text{if } 5 \leq t < 7 \\ -5 + 3.1(t-7) + 3.64(t-7)^2 - 1.74(t-7)^3, & \text{if } 7 \leq t < 8 \\ 5.15(t-8) - 1.59(t-8)^2 + 0.25(t-8)^3, & \text{if } 8 \leq t < 10 \\ 6 + 1.88(t-10) - 0.03(t-10)^2 - 0.02(t-10)^3, & \text{if } 10 \leq t < 15 \\ 12 + 0.008(t-15) - 0.34(t-15)^2 - 0.03(t-15)^3, & \text{if } 15 \leq t < 18. \end{cases}$$

□

#### 4.4.2 Periodic cubic splines

In many applications, the motion to be performed is *periodic*, i.e. initial and final positions are the same. In this case, the last two degrees of freedom to be assigned for the computation of the spline are exploited in order to impose the

continuity of initial and final velocities and accelerations. As a consequence, the method for the calculation of the coefficients is slightly different from what previously reported. As a matter of fact, in lieu of the conditions on the initial and final velocities  $v_0$  and  $v_n$  which were arbitrarily chosen, in this case one must consider

$$\begin{aligned} v_0 &= \dot{q}_0(t_0) = \dot{q}_{n-1}(t_n) = v_n \\ \ddot{q}_0(t_0) &= \ddot{q}_{n-1}(t_n). \end{aligned}$$

This last equation can be written as

$$\ddot{q}_0(t_0) = 2a_{0,2} = 2a_{n-1,2} + 6a_{n-1,3}T_{n-1} = \ddot{q}_{n-1}(t_n) \quad (4.12)$$

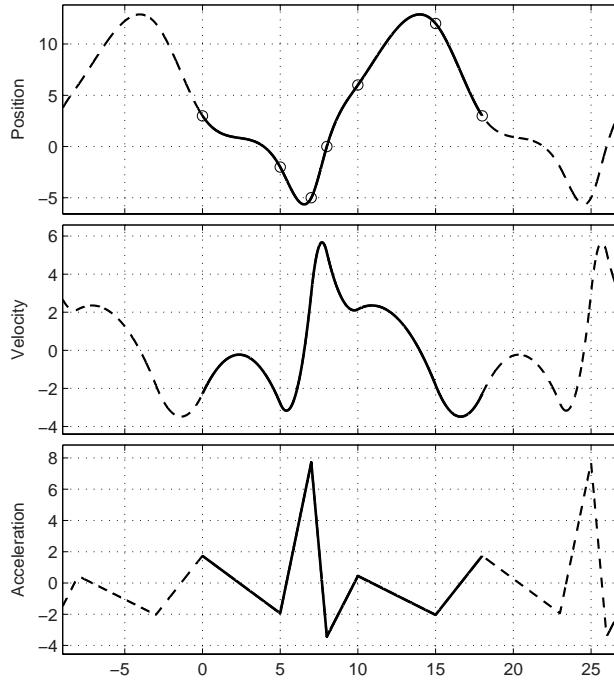
and, after the substitution of the coefficients expressions, from (4.8) one obtains

$$T_0v_{n-1} + 2(T_{n-1} + T_0)v_0 + T_{n-1}v_1 = \frac{3}{T_{n-1}T_0} [T_{n-1}^2(q_1 - q_0) + T_0^2(q_n - q_{n-1})]. \quad (4.13)$$

By adding this equation to the system (4.10), and by taking into account that, in this case, the velocity  $v_n$  is equal to  $v_0$  but unknown (and therefore in (4.10) the terms  $T_{n-2}v_n$  and  $T_1v_0$  must be moved at the left-hand side), the linear system for the computation of the velocities becomes

$$\left[ \begin{array}{ccccccccc} 2(T_{n-1} + T_0) & T_{n-1} & 0 & \cdots & 0 & T_0 & v_0 \\ T_1 & 2(T_0 + T_1) & T_0 & & 0 & 0 & v_1 \\ 0 & & \ddots & & & \vdots & \vdots \\ \vdots & & & & 0 & & v_{n-2} \\ 0 & & T_{n-2} & 2(T_{n-3} + T_{n-2}) & T_{n-3} & & v_{n-1} \\ T_{n-1} & 0 & \cdots & 0 & T_{n-1} & 2(T_{n-2} + T_{n-1}) & \end{array} \right] = \left[ \begin{array}{c} \frac{3}{T_{n-1}T_0} [T_{n-1}^2(q_1 - q_0) + T_0^2(q_n - q_{n-1})] \\ \frac{3}{T_0T_1} [T_0^2(q_2 - q_1) + T_1^2(q_1 - q_0)] \\ \frac{3}{T_1T_2} [T_1^2(q_3 - q_2) + T_2^2(q_2 - q_1)] \\ \vdots \\ \frac{3}{T_{n-1}T_{n-2}} [T_{n-3}^2(q_{n-1} - q_{n-2}) + T_{n-2}^2(q_{n-2} - q_{n-3})] \\ \frac{3}{T_{n-2}T_{n-1}} [T_{n-2}^2(q_n - q_{n-1}) + T_{n-1}^2(q_{n-1} - q_{n-2})] \end{array} \right].$$

The matrix of the system is not tridiagonal anymore. However, also in this case (the system of equations is called *cyclic*) efficient computational methods exist for the solution, see Appendix A.5. Once the velocities  $v_0, \dots, v_{n-1}$  have been obtained, the coefficients of the splines can be calculated by means of (4.8).



**Fig. 4.8.** Spline trajectory for periodic motions.

**Example 4.8** Fig. 4.8 shows a periodic spline trajectory with continuous velocity and acceleration profiles at the initial and final points, which are obviously assumed equal. The trajectory is defined by the following conditions

$$\begin{aligned} t_0 &= 0, & t_1 &= 5, & t_2 &= 7, & t_3 &= 8, & t_4 &= 10, & t_5 &= 15, & t_6 &= 18, \\ q_0 &= 3, & q_1 &= -2, & q_2 &= -5, & q_3 &= 0, & q_4 &= 6, & q_5 &= 12, & q_6 &= 3. \end{aligned}$$

The resulting matrix  $\mathbf{A}$  is

$$\mathbf{A} = \begin{bmatrix} 16 & 3 & 0 & 0 & 0 & 5 \\ 2 & 14 & 5 & 0 & 0 & 0 \\ 0 & 1 & 6 & 2 & 0 & 0 \\ 0 & 0 & 2 & 6 & 1 & 0 \\ 0 & 0 & 0 & 5 & 14 & 2 \\ 3 & 0 & 0 & 0 & 3 & 16 \end{bmatrix}$$

and

$$\mathbf{c} = [-54, -28, 25.5, 39, 52.2, -34.2]^T$$

which give the following velocities in the intermediate points (and in this case also in the first point<sup>3</sup>)

<sup>3</sup> Note that  $v_0 = -2.2823$  is the initial velocity ( $v_0 = v_6$ ), while the acceleration in the first and in the last point results  $\omega_0 = \omega_6 = 2a_{02} = 1.72$ .

$$\mathbf{v} = [-2.28, -2.78, 2.99, 5.14, 2.15, -1.8281]^T.$$

The expression of the spline is

$$s(t) = \begin{cases} 3 - 2.28t + 0.86t^2 - 0.12t^3, & \text{if } 0 \leq t < 5 \\ -2 - 2.78(t-5) - 0.96(t-5)^2 + 0.80(t-5)^3, & \text{if } 5 \leq t < 7 \\ -5 + 2.99(t-7) + 3.85(t-7)^2 - 1.85(t-7)^3, & \text{if } 7 \leq t < 8 \\ 5.14(t-8) - 1.71(t-8)^2 + 0.32(t-8)^3, & \text{if } 8 \leq t < 10 \\ 6 + 2.15(t-10) + 0.22(t-10)^2 - 0.008(t-10)^3, & \text{if } 10 \leq t < 15 \\ 12 + 1.82(t-15) - 1.02(t-15)^2 + 0.21(t-15)^3, & \text{if } 15 \leq t < 18 \end{cases}$$

□

#### 4.4.3 Cubic splines with assigned initial and final velocities: computation based on the accelerations

A different method for the definition of splines is based on the fact that the generic cubic polynomial  $q_k(t)$  of the spline can be expressed as a function of the second derivative computed at its endpoints, i.e. of the accelerations  $\ddot{q}(t_k) = \omega_k$ ,  $k = 0, \dots, n$ , instead of the velocities  $v_k$

$$q_k(t) = \frac{(t_{k+1}-t)^3}{6T_k}\omega_k + \frac{(t-t_k)^3}{6T_k}\omega_{k+1} + \left(\frac{q_{k+1}}{T_k} - \frac{T_k\omega_{k+1}}{6}\right)(t-t_k) + \left(\frac{q_k}{T_k} - \frac{T_k\omega_k}{6}\right)(t_{k+1}-t), \quad t \in [t_k, t_{k+1}]. \quad (4.14)$$

Then, the velocity and the acceleration are computed as

$$\dot{q}_k(t) = \frac{(t-t_k)^2}{2T_k}\omega_{k+1} + \frac{(t_{k+1}-t)^2}{2T_k}\omega_k + \frac{q_{k+1}-q_k}{T_k} - \frac{T_k(\omega_{k+1}-\omega_k)}{6} \quad (4.15)$$

$$\ddot{q}_k(t) = \frac{\omega_{k+1}(t-t_k) + \omega_k(t_{k+1}-t)}{T_k}. \quad (4.16)$$

In this case, it is necessary to find the accelerations  $\omega_k$  which univocally define the spline. Because of the continuity of velocities and accelerations in the intermediate points, one obtains

$$\dot{q}_{k-1}(t_k) = \dot{q}_k(t_k) \quad (4.17)$$

$$\ddot{q}_{k-1}(t_k) = \ddot{q}_k(t_k) = \omega_k. \quad (4.18)$$

By substituting (4.15) in (4.17), and using (4.18), one obtains

$$\frac{T_{k-1}}{T_k}\omega_{k-1} + \frac{2(T_k+T_{k-1})}{T_k}\omega_k + \omega_{k+1} = \frac{6}{T_k} \left( \frac{q_{k+1}-q_k}{T_k} - \frac{q_k-q_{k-1}}{T_{k-1}} \right) \quad (4.19)$$

for  $k = 1, \dots, n-1$ . From the conditions on initial and final velocities

$$\dot{s}(t_0) = \mathbf{v}_0, \quad \dot{s}(t_n) = \mathbf{v}_n$$

one deduces

$$\frac{T_0^2}{3}\omega_0 + \frac{T_0^2}{6}\omega_1 = q_1 - q_0 - T_0\mathbf{v}_0 \quad (4.20)$$

$$\frac{T_{n-1}^2}{3}\omega_n + \frac{T_{n-1}^2}{6}\omega_{n-1} = q_{n-1} - q_n + T_{n-1}\mathbf{v}_n. \quad (4.21)$$

By stacking eq. (4.19)-(4.21), one obtains the linear system

$$\mathbf{A} \boldsymbol{\omega} = \mathbf{c} \quad (4.22)$$

with the  $(n+1) \times (n+1)$  matrix  $\mathbf{A}$  (tridiagonal and symmetric)

$$\mathbf{A} = \begin{bmatrix} 2T_0 & T_0 & 0 & \dots & 0 \\ T_0 & 2(T_0 + T_1) & T_1 & & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & & & T_{n-2} & 2(T_{n-2} + T_{n-1}) & T_{n-1} \\ 0 & \dots & 0 & T_{n-1} & 2T_{n-1} & \end{bmatrix} \quad (4.23)$$

and the vector of known variables

$$\mathbf{c} = \begin{bmatrix} 6 \left( \frac{q_1 - q_0}{T_0} - \mathbf{v}_0 \right) \\ 6 \left( \frac{q_2 - q_1}{T_1} - \frac{q_1 - q_0}{T_0} \right) \\ \vdots \\ 6 \left( \frac{q_n - q_{n-1}}{T_{n-1}} - \frac{q_{n-1} - q_{n-2}}{T_{n-2}} \right) \\ 6 \left( \mathbf{v}_n - \frac{q_n - q_{n-1}}{T_{n-1}} \right) \end{bmatrix}. \quad (4.24)$$

The solution of this system is straightforward, by applying the remarks reported in the previous section and exploiting the algorithm described in Appendix A.5. The spline is finally obtained by substituting the values of the parameters  $\omega_k$  in (4.14).

Obviously, it is also possible to describe the cubic spline according to the initial definition, that is

$$s(t) = \{q_k(t), t \in [t_k, t_{k+1}], k = 0, \dots, n-1\},$$

$$q_k(t) = a_{k0} + a_{k1}(t - t_k) + a_{k2}(t - t_k)^2 + a_{k3}(t - t_k)^3$$

by computing the coefficients of the polynomials from the points  $q_k$  and the accelerations  $\omega_k$  as

$$\begin{cases} a_{k0} = q_k \\ a_{k1} = \frac{q_{k+1} - q_k}{T_k} - \frac{T_k}{6}(\omega_{k+1} + 2\omega_k) \\ a_{k2} = \frac{\omega_k}{2} \\ a_{k3} = \frac{\omega_{k+1} - \omega_k}{6T_k} \end{cases} \quad k = 0, \dots, n-1. \quad (4.25)$$

#### 4.4.4 Cubic splines with assigned initial and final velocities and accelerations

A spline is a function continuous up to the second derivative, but in general it is not possible to assign at the same time both initial and final velocities and accelerations. As a consequence, at its extremities the spline is characterized by a discontinuity on the velocities or on the accelerations. In case these discontinuities represent a problem, different approaches can be adopted:

1. A polynomial function of degree 5 can be used for the first and last tract, with the consequent drawback of allowing larger overshoot in these segments and slightly increasing the computational burden.
2. Two free<sup>4</sup> extra points are added in the first and last segment, and their value is computed by imposing the desired initial and final values of both velocity and acceleration.

The latter method is now illustrated in details.

Let us consider a vector of  $n - 1$  points to be interpolated

$$\mathbf{q} = [q_0, q_2, q_3, \dots, q_{n-3}, q_{n-2}, q_n]^T$$

at the time instants

$$\mathbf{t} = [t_0, t_2, t_3, \dots, t_{n-3}, t_{n-2}, t_n]^T$$

and the boundary conditions on the velocities  $v_0, v_n$ , and on the accelerations  $a_0, a_n$ . In order to impose the desired accelerations, two extra points  $\bar{q}_1$  and  $\bar{q}_{n-1}$  are added. Time instants  $\bar{t}_1$  and  $\bar{t}_{n-1}$  are placed between  $t_0$  and  $t_2$  and between  $t_{n-2}$  and  $t_n$  respectively. The problem is then to determine the spline through

$$\bar{\mathbf{q}} = [q_0, \bar{q}_1, q_2, q_3, \dots, q_{n-3}, q_{n-2}, \bar{q}_{n-1}, q_n]^T$$

at

$$\bar{\mathbf{t}} = [t_0, \bar{t}_1, t_2, t_3, \dots, t_{n-3}, t_{n-2}, \bar{t}_{n-1}, t_n]^T$$

with initial and final velocities  $v_0$  and  $v_n$ . This problem can be solved by means of the linear system (4.22) but, since  $\bar{q}_1$  and  $\bar{q}_{n-1}$  are unknown, it is

---

<sup>4</sup> In the sense that these points cannot be fixed a priori.

necessary to express such points in terms of known variables, namely first/last position, velocity, acceleration ( $q_0/q_n, v_0/v_n, a_0/a_n$ ) and the acceleration at these points ( $\omega_1, \omega_{n-1}$ ). In this way, it is possible to consider the constraints on the initial and final acceleration. By substituting

$$q_1 = q_0 + T_0 v_0 + \frac{T_0^2}{3} a_0 + \frac{T_0^2}{6} \omega_1 \quad (4.26)$$

$$q_{n-1} = q_n - T_{n-1} v_n + \frac{T_{n-1}^2}{3} a_n + \frac{T_{n-1}^2}{6} \omega_{n-1} \quad (4.27)$$

in (4.23) and (4.24), and by rearranging the  $n - 1$  equations, one obtains a linear system

$$\mathbf{A} \boldsymbol{\omega} = \mathbf{c} \quad (4.28)$$

with

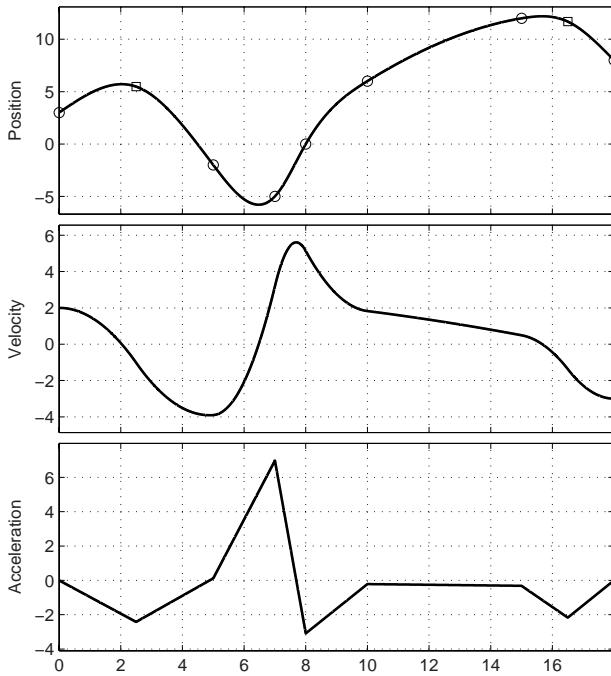
$$\mathbf{A} = \begin{bmatrix} 2T_1 + T_0 \left( 3 + \frac{T_0}{T_1} \right) & T_1 & 0 & \dots & 0 \\ T_1 - \frac{T_0^2}{T_1} & 2(T_1 + T_2) & T_2 & & \vdots \\ 0 & T_2 & 2(T_2 + T_3) & T_3 & \\ \vdots & & \vdots & & 0 \\ 0 & \dots & 0 & T_{n-2} & T_{n-2} - \frac{T_{n-1}^2}{T_{n-2}} \\ & & & T_{n-2} & 2T_{n-2} + T_{n-1} \left( 3 + \frac{T_{n-1}}{T_{n-2}} \right) \end{bmatrix}$$

and

$$\mathbf{c} = \begin{bmatrix} 6 \left( \frac{q_2 - q_0}{T_1} - v_0 \left( 1 + \frac{T_0}{T_1} \right) - a_0 \left( \frac{1}{2} + \frac{T_0}{3T_1} \right) T_0 \right) \\ 6 \left( \frac{q_3 - q_2}{T_2} - \frac{q_2 - q_0}{T_1} + v_0 \frac{T_0}{T_1} + a_0 \frac{T_0^2}{3T_1} \right) \\ 6 \left( \frac{q_4 - q_3}{T_3} - \frac{q_3 - q_2}{T_2} \right) \\ \vdots \\ 6 \left( \frac{q_{n-2} - q_{n-3}}{T_{n-3}} - \frac{q_{n-3} - q_{n-4}}{T_{n-4}} \right) \\ 6 \left( \frac{q_n - q_{n-2}}{T_{n-2}} - \frac{q_{n-2} - q_{n-3}}{T_{n-3}} - v_n \frac{T_{n-1}}{T_{n-2}} + a_n \frac{T_{n-1}^2}{3T_{n-2}} \right) \\ 6 \left( \frac{q_{n-2} - q_n}{T_{n-2}} + v_n \left( 1 + \frac{T_{n-1}}{T_{n-2}} \right) - a_n \left( \frac{1}{2} + \frac{T_{n-1}}{3T_{n-2}} \right) T_{n-1} \right) \end{bmatrix}.$$

Note that  $T_0, T_1$  and  $T_{n-2}, T_{n-1}$  are functions of  $\bar{t}_1$  and  $\bar{t}_{n-1}$  respectively, which can be arbitrarily selected in the intervals  $(t_0, t_2)$  and  $(t_{n-2}, t_n)$ , e.g.  $\bar{t}_1 = \frac{t_0 + t_2}{2}$  and  $\bar{t}_{n-1} = \frac{t_{n-2} + t_n}{2}$ .

By solving the system (4.28), it is possible to determine the accelerations in the intermediate points



**Fig. 4.9.** Spline trajectory with constraints on initial and final velocities and accelerations.

$$\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3, \dots, \omega_{n-2}, \omega_{n-1}]^T$$

which, together with the boundary values  $\mathbf{a}_0$  and  $\mathbf{a}_n$ , allow to compute the overall spline according to (4.14).

**Example 4.9** Fig. 4.9 shows a spline computed according to the above algorithm. In particular, the goal is to find a trajectory through the points

$$\begin{aligned} t_0 &= 0, & t_2 &= 5, & t_3 &= 7, & t_4 &= 8, & t_5 &= 10, & t_6 &= 15, & t_8 &= 18, \\ q_0 &= 3, & q_2 &= -2, & q_3 &= -5, & q_4 &= 0, & q_5 &= 6, & q_6 &= 12, & q_8 &= 8, \end{aligned}$$

with  $v_0 = 2$ ,  $v_8 = -3$ ,  $a_0 = 0$ ,  $a_8 = 0$ . In order to impose the conditions on the acceleration, two extra points are added in  $t_1 = 2.5$  and  $t_7 = 16.5$  (note that this choice is completely arbitrary). As a consequence, the vector of the time interval length is

$$\mathbf{T} = [2.5, 2.5, 2, 1, 2, 5, 1.5, 1.5]^T.$$

The resulting matrix  $\mathbf{A}$  is

$$\mathbf{A} = \begin{bmatrix} 3 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 5 & 2 & 0 & 0 & 0 \\ 0 & 2 & 18 & 7 & 0 & 0 \\ 0 & 0 & 7 & 18 & 2 & 0 \\ 0 & 0 & 0 & 2 & 8 & 0 \\ 0 & 0 & 0 & 0 & 2 & 12 \end{bmatrix}$$

and the known variables

$$\mathbf{c} = [-12, 12, -14.57, 8.57, 0, -24]^T.$$

The solution of the linear system (4.28) provides the following accelerations in the intermediate points

$$\boldsymbol{\omega} = [-4.50, 3.03, -1.58, 1.12, -0.28, -1.95]^T.$$

Therefore, the two extra points, computed with (4.26) and (4.27), are

$$q_1 = 5.48 \quad \text{and} \quad q_7 = 11.68.$$

The final expression of the spline is

$$s(t) = \begin{cases} s_0(t), & \text{if } 0 \leq t < 2.5 \\ s_1(t), & \text{if } 2.5 \leq t < 5 \\ s_2(t), & \text{if } 5 \leq t < 7 \\ s_3(t), & \text{if } 7 \leq t < 8 \\ s_4(t), & \text{if } 8 \leq t < 10 \\ s_5(t), & \text{if } 10 \leq t < 15 \\ s_6(t), & \text{if } 15 \leq t < 16.5 \\ s_7(t), & \text{if } 16.5 \leq t < 18 \end{cases}$$

where

$$\begin{aligned} s_0(t) &= -0.16t^3 + 3.2t + 1.2(2.5-t) \\ s_1(t) &= -0.16(5-t)^3 + 0.007(t-2.5)^3 - 0.84(t-2.5) + 3.2(5-t) \\ s_2(t) &= 0.009(7-t)^3 + 0.58(t-5)^3 - 4.82(t-5) - 1.03(7-t) \\ s_3(t) &= 1.16(8-t)^3 - 0.51(t-7)^3 + 0.51(t-7) - 6.16(8-t) \\ s_4(t) &= -0.25(10-t)^3 - 0.018(t-8)^3 + 3.07(t-8) + 1.03(10-t) \\ s_5(t) &= -0.007(15-t)^3 - 0.01(t-10)^3 + 2.66(t-10) + 1.38(15-t) \\ s_6(t) &= -0.03(16.5-t)^3 - 0.24(t-15)^3 + 8.33(t-15) + 8.07(16.5-t) \\ s_7(t) &= -0.24(18-t)^3 + 5.33(t-16.5) + 8.33(18-t). \end{aligned}$$

□

#### 4.4.5 Smoothing cubic splines

Smoothing cubic splines are defined in order to approximate, and not to interpolate, a set of given data points [32, 33, 34, 35, 36]. In particular, this kind of trajectory is adopted to find a tradeoff between two opposite goals:

- A good fit of the given via-points.
- A trajectory as smooth as possible, i.e. with curvature/acceleration as small as possible.

Given the vector of points

$$\mathbf{q} = [q_0, q_1, q_2, \dots, q_{n-2}, q_{n-1}, q_n]^T$$

and the time instants

$$\mathbf{t} = [t_0, t_1, t_2, \dots, t_{n-2}, t_{n-1}, t_n]^T$$

the coefficients of the cubic smoothing spline  $s(t)$  are computed with the purpose of minimizing

$$L := \mu \sum_{k=0}^n w_k (s(t_k) - q_k)^2 + (1 - \mu) \int_{t_0}^{t_n} \ddot{s}(t)^2 dt \quad (4.29)$$

where the parameter  $\mu \in [0, 1]$  reflects the different importance given to the two conflicting goals, while  $w_k$  are parameters which can be arbitrarily chosen in order to modify the weight of the  $k$ -th quadratic error on the global optimization problem. Note that the selection of different coefficients  $w_k$  allows to operate locally on the spline, by reducing the approximation error only in some points of interest. The integral in the second term of (4.29) can be written as

$$\int_{t_0}^{t_n} \ddot{s}(t)^2 dt = \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} \ddot{q}_k(t)^2 dt. \quad (4.30)$$

Since the spline is composed by cubic segments, the second derivative in each interval  $[t_k, t_{k+1}]$  is a linear function from the initial acceleration  $\omega_k$  to the final acceleration  $\omega_{k+1}$ , and therefore

$$\begin{aligned} \int_{t_k}^{t_{k+1}} \ddot{q}_k(t)^2 dt &= \int_{t_k}^{t_{k+1}} \left( \omega_k + \frac{(t - t_k)}{T_k} (\omega_{k+1} - \omega_k) \right)^2 dt = \\ &= \int_0^{T_k} \left( \omega_k + \frac{\tau}{T_k} (\omega_{k+1} - \omega_k) \right)^2 d\tau = \frac{1}{3} T_k (\omega_k^2 + \omega_k \omega_{k+1} + \omega_{k+1}^2) \end{aligned} \quad (4.31)$$

where it is assumed  $\tau = t - t_k$ .

Then, the criterion function (4.29) can be written as

$$L = \sum_{k=0}^n w_k (q_k - s(t_k))^2 + \lambda \sum_{k=0}^{n-1} 2T_k (\omega_k^2 + \omega_k \omega_{k+1} + \omega_{k+1}^2) \quad (4.32)$$

where  $\lambda = \frac{1 - \mu}{6\mu}$ , with  $\mu \neq 0$ , or, with a more compact notation, as

$$L = (\mathbf{q} - \mathbf{s})^T \mathbf{W} (\mathbf{q} - \mathbf{s}) + \lambda \boldsymbol{\omega}^T \mathbf{A} \boldsymbol{\omega} \quad (4.33)$$

where  $\mathbf{s}$  is the vector of the approximations  $[s(t_k)]$ ,  $\boldsymbol{\omega} = [\omega_0, \dots, \omega_n]^T$  is the vector of accelerations,  $\mathbf{W} = \text{diag}\{w_0, \dots, w_n\}$ , and  $\mathbf{A}$  is the constant matrix defined in (4.23).

Equation (4.22) provides the relation between the position of the intermediate points  $q(t_k)$  and the accelerations  $\omega_k$ . In the case of smoothing splines, the intermediate points are not the given via-points, which are only approximated, but the value of the spline itself at time instants  $t_k$ , i.e.  $s(t_k)$ . For a *clamped spline*, i.e. with null initial and final velocity  $v_0 = v_n = 0$ , the linear function (4.22) can be rewritten as

$$\mathbf{A} \boldsymbol{\omega} = \mathbf{C} \mathbf{s} \quad (4.34)$$

where

$$\mathbf{C} = \begin{bmatrix} -\frac{6}{T_0} & \frac{6}{T_0} & 0 & \cdots & 0 \\ \frac{6}{T_0} & -\left(\frac{6}{T_0} + \frac{6}{T_1}\right) & \frac{6}{T_1} & & \vdots \\ 0 & \frac{6}{T_1} & -\left(\frac{6}{T_1} + \frac{6}{T_2}\right) & \frac{6}{T_2} & \\ \vdots & & & \ddots & 0 \\ & & & & \frac{6}{T_{n-2}} & -\left(\frac{6}{T_{n-2}} + \frac{6}{T_{n-1}}\right) & \frac{6}{T_{n-1}} \\ 0 & \cdots & & & 0 & \frac{6}{T_{n-1}} & -\frac{6}{T_{n-1}} \end{bmatrix}.$$

By substituting (4.34) in (4.33), it is possible to obtain an expression of  $L$  which depends only on  $\mathbf{s}$

$$L(\mathbf{s}) = (\mathbf{q} - \mathbf{s})^T \mathbf{W} (\mathbf{q} - \mathbf{s}) + \lambda \mathbf{s}^T \mathbf{C}^T \mathbf{A}^{-1} \mathbf{C} \mathbf{s}.$$

The optimal value of the approximation  $\mathbf{s}$  minimizes the function  $L(\mathbf{s})$ . Therefore, by differentiating  $L(\mathbf{s})$  with respect to  $\mathbf{s}$  and setting the result to zero, one obtains

$$-(\mathbf{q} - \mathbf{s})^T \mathbf{W} + \lambda \mathbf{s}^T \mathbf{C}^T \mathbf{A}^{-1} \mathbf{C} = 0$$

that is

$$\mathbf{s} = (\mathbf{W} + \lambda \mathbf{C}^T \mathbf{A}^{-1} \mathbf{C})^{-1} \mathbf{W} \mathbf{q}.$$

By exploiting the matrix inversion lemma, this relation can be rewritten as

$$\mathbf{s} = \mathbf{q} - \lambda \mathbf{W}^{-1} \mathbf{C}^T (\mathbf{A} + \lambda \mathbf{C} \mathbf{W}^{-1} \mathbf{C}^T)^{-1} \mathbf{C} \mathbf{q}.$$

In this way, being  $\mathbf{W}$  diagonal, the solution of the problem requires only one matrix inversion. In particular, it is necessary to solve the following linear system

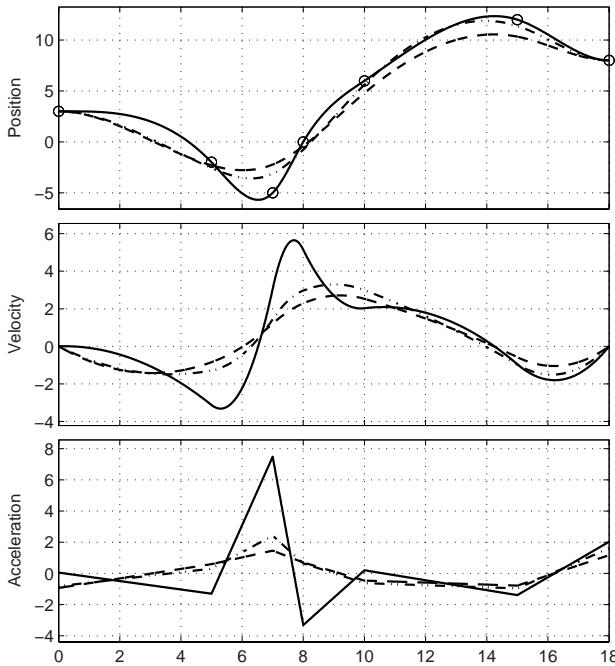
$$(\mathbf{A} + \lambda \mathbf{C} \mathbf{W}^{-1} \mathbf{C}^T) \boldsymbol{\omega} = \mathbf{C} \mathbf{q} \quad (4.35)$$

which provides the parameters<sup>5</sup>  $\boldsymbol{\omega}$ . Then, it is possible to compute

$$\mathbf{s} = \mathbf{q} - \lambda \mathbf{W}^{-1} \mathbf{C}^T \boldsymbol{\omega}. \quad (4.36)$$

---

<sup>5</sup> It can be easily shown that these parameters are the accelerations in the intermediate points, which appear in (4.31) - (4.34).



**Fig. 4.10.** Smoothing splines computed for different values of  $\mu$  (0.3, 0.6, 1).

Matrix  $(\mathbf{A} + \lambda \mathbf{C} \mathbf{W}^{-1} \mathbf{C}^T)$  in (4.35) is symmetric with five diagonal bands, and therefore the system can be solved by means of computationally efficient procedures. Moreover, it is possible to assign directly  $\mathbf{W}^{-1} = \text{diag} \left\{ \frac{1}{w_0}, \dots, \frac{1}{w_{n-1}} \right\}$  (without computing the inverse of  $\mathbf{W}$ ) in (4.35) and (4.36), in particular when it is desired that the  $k$ -th position error  $q_k - s(t_k)$  is zero. In this case, it is sufficient to impose that the element of  $\mathbf{W}^{-1}$  corresponding to this point is null.

Once the accelerations in the intermediate points, obtained from (4.35), have been computed, it is possible to define the spline trajectory from (4.14).

**Example 4.10** Fig. 4.10 shows a smoothing spline computed for different values of  $\mu$  (in particular,  $\mu = 1$  for the solid line trajectory,  $\mu = 0.6$  for the dashdot line and  $\mu = 0.3$  for the dashed line). The trajectory approximates the following points

$$\begin{aligned} t_0 &= 0, & t_1 &= 5, & t_2 &= 7, & t_3 &= 8, & t_4 &= 10, & t_5 &= 15, & t_6 &= 18 \\ q_0 &= 3, & q_1 &= -2, & q_2 &= -5, & q_3 &= 0, & q_4 &= 6, & q_5 &= 12, & q_6 &= 8 \end{aligned}$$

which are weighted by

$$\mathbf{W}^{-1} = \text{diag} \{ 0, 1, 1, 1, 1, 1, 0 \}.$$

As a consequence, the first and the last point are exactly interpolated, while the approximation on the intermediate ones depends on  $\mu$ :

- For  $\mu = 0.3$ ,  $s = [3, -2.29, -2.22, -0.41, 4.79, 10.33, 8]^T$ .
- For  $\mu = 0.6$ ,  $s = [3, -2.55, -3.11, -0.76, 5.62, 11.38, 8]^T$ .
- For  $\mu = 1$  all the data points are exactly interpolated ( $s = q$ ), but, on the other hand, the acceleration has the maximum values.

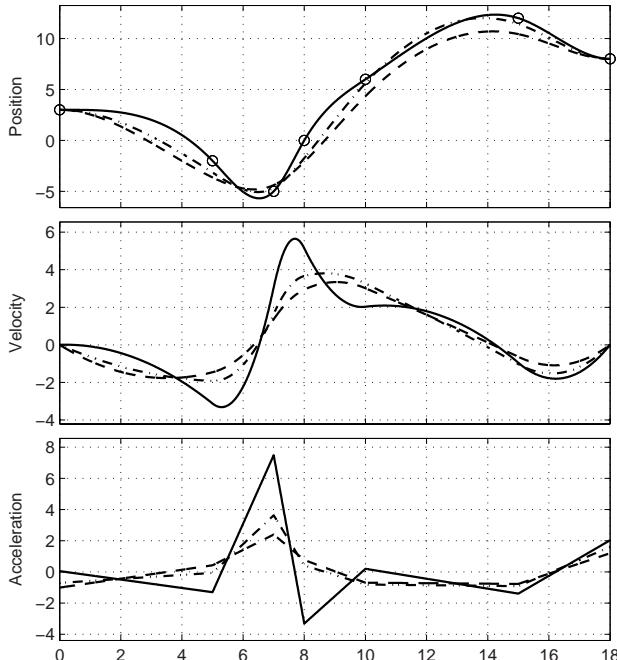
From Fig. 4.10, it is evident that (when  $\mu \neq 1$ ) approximation errors are larger for those points in which the acceleration(/curvature) is higher. For this reason, it is possible to selectively reduce these errors by changing the weights in  $\mathbf{W}$ . In particular, by assuming

$$\mathbf{W}^{-1} = \text{diag}\{0, 1, 0.1, 1, 1, 1, 0\}$$

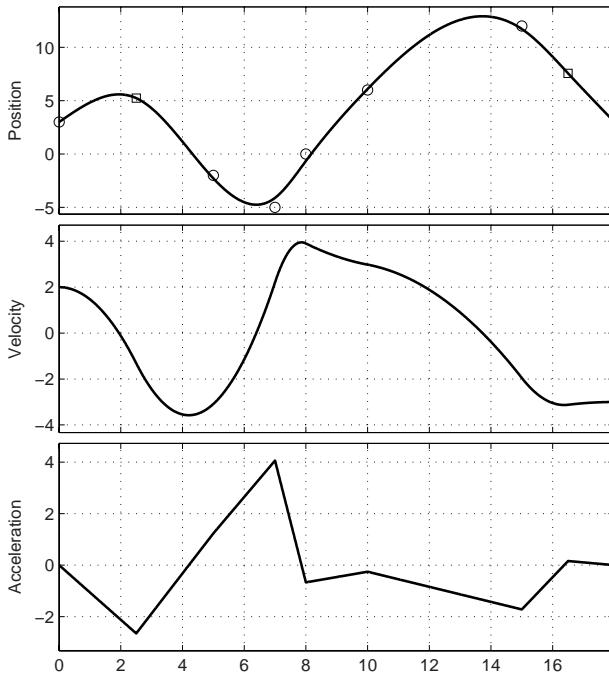
and  $\mu = 0.6$ , the approximating points are

$$s = [3, -3.14, -4.66, -1.73, 5.60, 11.42, 8]^T.$$

As shown in Fig. 4.11, the error with respect to the third via-point is considerably reduced.  $\square$



**Fig. 4.11.** Smoothing splines for different values of  $\mu$  and  $w_k$ .



**Fig. 4.12.** Smoothing splines with assigned initial and final velocity/acceleration.

It is worth noticing that the elements of  $\mathbf{s}$  are the positions of the (smoothing) spline at the time instants  $t_k$ , i.e.  $s_k = s(t_k)$ . Therefore, it is possible to regard them as new points which must be exactly interpolated with one of the techniques reported above. In this way, at the expense of a small deformation of the spline with respect to the smoothing spline with null initial and final velocities, it is possible to consider boundary conditions on the velocity and on the acceleration.

**Example 4.11** Fig. 4.12 shows a smoothing spline computed for  $\mu = 0.9$  and with initial and final velocities  $v_0 = 2$ ,  $v_n = -3$ , while the initial and final accelerations have been set to zero ( $a_0 = 0$ ,  $a_n = 0$ ). The trajectory must approximate the points

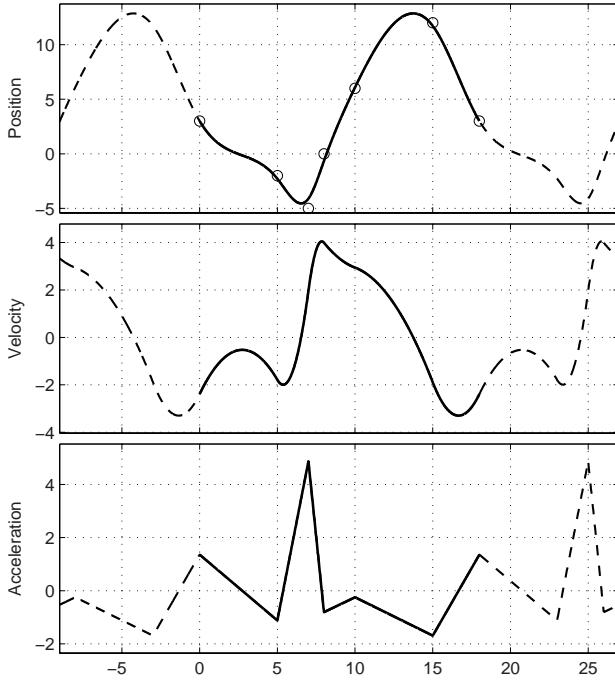
$$\begin{aligned} t_0 &= 0, & t_1 &= 5, & t_2 &= 7, & t_3 &= 8, & t_4 &= 10, & t_5 &= 15, & t_6 &= 18, \\ q_0 &= 3, & q_1 &= -2, & q_2 &= -5, & q_3 &= 0, & q_4 &= 6, & q_5 &= 12, & q_6 &= 3, \end{aligned}$$

with the weights

$$\mathbf{W}^{-1} = \text{diag} \{ 0, 1, 1, 1, 1, 1, 0 \}.$$

The approximating points are

$$\mathbf{s} = [3, -2.28, -4.10, -0.65, 6.09, 11.71, 3]^T$$



**Fig. 4.13.** Smoothing splines with periodic conditions.

which, used for computing the trajectory according to the algorithm reported in Sec. 4.4.4, provide the following accelerations on the intermediate points

$$\boldsymbol{\omega} = [0, -2.65, 1.23, 4.06, -0.66, -0.25, -1.71, 0.16, 0]^T,$$

while the two extra points are  $q_1 = 5.23$  (at  $t_1 = 2.5$ ) and  $q_{n-1} = 7.56$  (at  $t_{n-1} = 16.5$ ).  $\square$

**Example 4.12** The same points  $q_k$  and weights  $w_k$  of the previous example are considered, with the purpose of finding a smoothing trajectory subject to periodic constraints ( $\mathbf{v}_0 = \mathbf{v}_n$ ,  $\mathbf{a}_0 = \mathbf{a}_n$ ). The approximating points  $s_k$  are the same, and by means of the algorithm reported in Sec. 4.4.2 they produce the trajectory of Fig. 4.13. In this case, the coefficients  $a_{ki}$  ( $k = 0, \dots, n$ ,  $i = 0, \dots, 3$ ) of the  $n - 1$  cubic polynomials composing the spline are

$k$	$a_{k0}$	$a_{k1}$	$a_{k2}$	$a_{k3}$
0	3	-2.39	0.67	-0.08
1	-2.28	-1.78	-0.55	0.49
2	-4.10	1.96	2.43	-0.94
3	-0.65	3.99	-0.40	0.04
4	6.09	2.94	-0.12	-0.04
5	11.71	-1.89	-0.84	0.16

□

### Smoothing spline with prescribed tolerance

By recursively applying the algorithm for the computation of smoothing splines, it is possible to find the value of the coefficient  $\mu$  which guarantees that the maximum approximation error ( $\epsilon_{max} = \max_k \{q(t_k) - q_k\}$ ) is smaller than a given threshold ( $\delta$ ). In particular, by applying a binary research on  $\mu \in [0, 1]$ , few iterations are sufficient to determine the correct value. The algorithm is based on three steps, which must be iterated until  $\epsilon_{max} > \delta$  and the maximum number of iterations is not reached. The generic  $i$ -th iteration consists in:

1. Assume  $\mu(i) = \frac{L(i) + R(i)}{2}$  where  $L(i)$  and  $R(i)$  are two auxiliary variables (whose initial values are 0 and 1 respectively).
2. Compute  $s_k(i)$  with (4.35) and (4.36), and  $\epsilon_{max}(i)$ .
3. Update  $L$  and  $R$  according to the value of  $\epsilon_{max}(i)$ :

```

if   ( $\epsilon_{max}(i) > \delta$ )
     $L(i+1) = \mu$ 
     $R(i+1) = R(i)$ 
else
     $R(i+1) = \mu$ 
     $L(i+1) = L(i)$ 
end

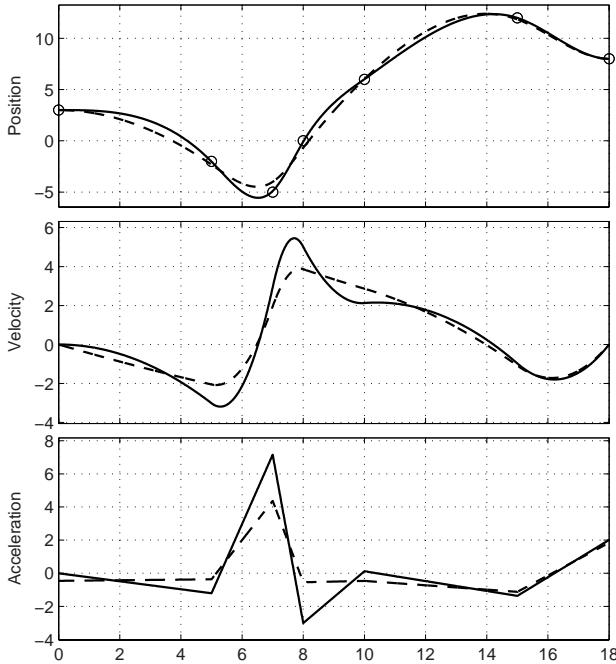
```

**Example 4.13** Fig. 4.14 shows two smoothing splines computed with the purpose of approximating the points

$$\begin{aligned} t_0 &= 0, & t_1 &= 5, & t_2 &= 7, & t_3 &= 8, & t_4 &= 10, & t_5 &= 15, & t_6 &= 18, \\ q_0 &= 3, & q_1 &= -2, & q_2 &= -5, & q_3 &= 0, & q_4 &= 6, & q_5 &= 12, & q_6 &= 3, \end{aligned}$$

with a maximum tolerance of 0.1 (solid line) and 1 (dashed line). The resulting coefficients  $\mu$  are respectively 0.9931 and 0.8792. In this case, the maximum number of iterations has been set to 20, and the matrix of the weights is

$$\mathbf{W}^{-1} = \text{diag} \{ 0, 1, 1, 1, 1, 1, 0 \}.$$



**Fig. 4.14.** Smoothing splines computed with a approximation error of 0.1 (solid) and 1 (dashed).

□

The procedure described in this section operates only on  $\mu$ , while it has been highlighted in previous examples that also the weights  $w_k$  play an important role on the approximation error. For this reason, it is important to underline that a proper choice of these coefficients is of fundamental importance for the computation of the trajectory, see [37]. On the other hand, it is not practical to define an algorithm which acts iteratively on all the weights  $w_k$  to obtain a prescribed maximum approximation error, since, especially when the set of data points to be approximated is very large, the computational burden (and duration) would be excessive.

#### 4.4.6 Choice of the time instants and optimization of cubic splines

If not imposed by the specific application, the time instants  $t_k$ , used to perform the interpolation process of the points  $q_k$ ,  $k = 0, \dots, n$ , can be chosen in several ways, with different results. In particular, according to the most common techniques available in the literature [38] and with reference to a unitary interval

$$t_0 = 0, \quad t_n = 1,$$

the distribution of the intermediate time instants can be defined as

$$t_k = t_{k-1} + \frac{d_k}{d} \quad \text{with} \quad d = \sum_{k=0}^{n-1} d_k \quad (4.37)$$

where  $d_k$  may be computed as:

1.  $d_k = \frac{1}{n-1}$ , i.e. a constant value leading to equally spaced points.
2.  $d_k = |q_{k+1} - q_k|$ , producing the cord length distribution.
3.  $d_k = |q_{k+1} - q_k|^\mu$ , where it is usually assumed  $\mu = \frac{1}{2}$ , defining the so-called centripetal distribution.

The first method usually gives the highest speeds, while the last one is particularly useful when it is desired to reduce the accelerations.

**Example 4.14** The spline trajectory interpolating the points

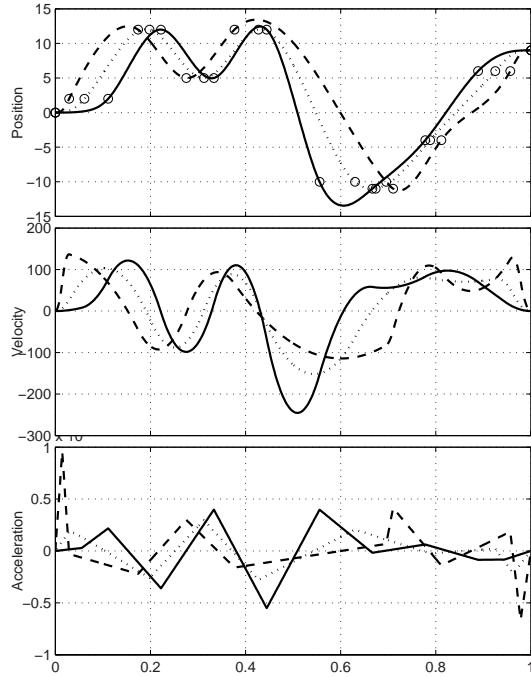
$$\begin{aligned} q_0 &= 0, & q_1 &= 2, & q_2 &= 12, & q_3 &= 5, & q_4 &= 12, \\ q_5 &= -10, & q_6 &= -11, & q_7 &= -4, & q_8 &= 6, & q_9 &= 9, \end{aligned}$$

and with null initial and final velocities and accelerations is shown in Fig. 4.15. The choice of the time instants has been made according to the above three techniques. The uniform distribution produces peaks in the velocity profile, while the cord length distribution leads to high accelerations. As a consequence, if the trajectories are scaled in time in order to meet given constraints on velocity and acceleration (in particular it is assumed  $v_{max} = 3$ ,  $a_{max} = 2$ ), the duration of the spline computed with a cord length distribution of  $t_k$  is limited by the maximum acceleration, while the other two trajectories reach the maximum speed, see Fig. 4.16. □

The total duration of a spline trajectory  $s(t)$ , interpolating the points  $(t_k, q_k)$ ,  $k = 0, \dots, n$ , is

$$T = \sum_{k=0}^{n-1} T_k = t_n - t_0$$

where  $T_k = t_{k+1} - t_k$ . Therefore, when limit values of velocity, acceleration, etc. are provided, it is possible to define an optimization problem aiming at minimizing the total time  $T$ , [39, 40, 41]. From a formal point of view, the problem can be formulated as

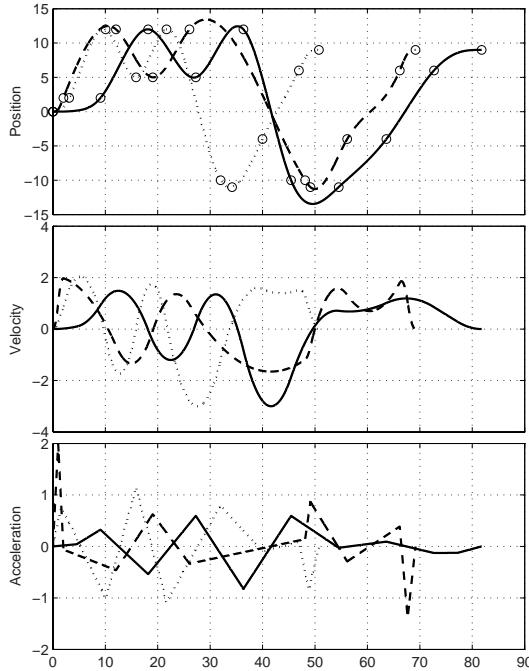


**Fig. 4.15.** Spline trajectories interpolating a set of data points, with different distributions of time instants: uniformly spaced (solid), cord length (dashed), centripetal (dotted).

$$\left\{ \begin{array}{ll} \min & T = \sum_{k=0}^{n-1} T_k \\ \text{such that} & |\dot{s}(t, T_k)| < v_{max}, \quad t \in [0, T] \\ & |\ddot{s}(t, T_k)| < a_{max}, \quad t \in [0, T]. \end{array} \right. \quad (4.38)$$

This is a nonlinear optimum problem with a linear objective function, solvable with classical techniques of operational research. Since the coefficients which determine the spline (and as a consequence the value of the velocity and of the acceleration along the trajectory) are computed as a function of the intervals  $T_k$ , the optimization problem can be solved in a iterative way, by scaling in time the segments which compose the spline [34, 42, 43]. As a matter of fact, if the time interval  $T_k$  is replaced by  $T'_k = \lambda T_k$ , then the velocity, acceleration and jerk are scaled by  $1/\lambda$ ,  $1/\lambda^2$ ,  $1/\lambda^3$  respectively. Therefore, by choosing

$$\lambda = \max\{\lambda_v, \lambda_a, \lambda_j\}$$



**Fig. 4.16.** Spline trajectories interpolating a set of data points, with different distributions of time instants, and properly scaled to be compliant with the constraints on the velocity and acceleration: uniformly spaced (solid), cord length (dashed), centripetal (dotted).

where

$$\begin{aligned}
 \lambda_v &= \max_k \{\lambda_{v,k}\}, \quad \text{with} \quad \lambda_{v,k} = \max_{t \in [t_k, t_{k+1}]} \left\{ \frac{|\dot{q}_k(t)|}{v_{max}} \right\} \\
 \lambda_a &= \max_k \{\lambda_{a,k}\}, \quad \text{with} \quad \lambda_{a,k} = \max_{t \in [t_k, t_{k+1}]} \left\{ \sqrt{\frac{|\ddot{q}_k(t)|}{a_{max}}} \right\} \\
 \lambda_j &= \max_k \{\lambda_{j,k}\}, \quad \text{with} \quad \lambda_{j,k} = \max_{t \in [t_k, t_{k+1}]} \left\{ \sqrt[3]{\frac{|q_k^{(3)}(t)|}{j_{max}}} \right\}
 \end{aligned}$$

it is possible to optimize the trajectory, in the sense that the spline will reach the maximum speed or the maximum acceleration or, if given, the maximum jerk, in at least a point of the interval  $[t_0, t_n]$  (see Example 4.14). In order to obtain a spline composed by segments individually optimized, that is executed in minimum time, it is necessary to scale in time each interval according to

$$T'_k = \lambda_k T_k \quad (4.39)$$

with

$$\lambda_k = \max\{\lambda_{v,k}, \lambda_{a,k}, \lambda_{j,k}\}.$$

In this way, in each tract the maximum speed, the maximum acceleration or the maximum jerk is obtained; on the other hand, by scaling separately each spline segment, the velocity, the acceleration and the jerk are discontinuous at the *joints* between two contiguous tracts. It is therefore necessary to recalculate the spline coefficients with the new values  $T'_k$ . This approach can be iterated until the variation between  $T_k$  and  $T'_k$  is small enough. Therefore, the optimization procedure is composed by two steps to be iterated:

- Given the spline compute the time intervals  $T'_k$  with (4.39).
- Recalculate the spline coefficients with the new  $T'_k$  according to one of the methods described in this section.

Since the limit trajectory is obtained by a local modification process, we cannot expect it to be a global solution of the optimization problem (4.38). However, the solution in general is quite satisfactory, even after few iterations [42]. An alternative method for obtaining optimal minimum time splines, based on heuristics, is reported in [17].

**Example 4.15** The goal is to plan the spline trajectory interpolating the following points

$$q_0 = 0, q_1 = 2, q_2 = 12, q_3 = 5$$

which minimizes the total time  $T$  with the constraints  $v_{max} = 3$ ,  $a_{max} = 2$ . Null initial and final velocities are assumed.

It is necessary to solve the nonlinear optimum problem defined by

$$\min \{T = T_0 + T_1 + T_2\}$$

with the constraints

$$\left\{ \begin{array}{lll} a_{01} & \leq v_{max} & (\text{init. vel. 1-st tract} \leq v_{max}) \\ a_{11} & \leq v_{max} & (\text{init. vel. 2-nd tract} \leq v_{max}) \\ a_{21} & \leq v_{max} & (\text{init. vel. 3-rd tract} \leq v_{max}) \\ a_{01} + 2a_{02}T_1 + 3a_{03}T_1^2 & \leq v_{max} & (\text{final vel. 1-st tract} \leq v_{max}) \\ a_{11} + 2a_{12}T_2 + 3a_{13}T_2^2 & \leq v_{max} & (\text{final vel. 2-nd tract} \leq v_{max}) \\ a_{21} + 2a_{22}T_3 + 3a_{23}T_3^2 & \leq v_{max} & (\text{final vel. 3-rd tract} \leq v_{max}) \\ a_{01} + 2a_{02} \left( -\frac{a_{02}}{3a_{03}} \right) + 3a_{03} \left( -\frac{a_{02}}{3a_{03}} \right)^2 & \leq v_{max} & (\text{vel. 1-st tract} \leq v_{max}) \\ a_{11} + 2a_{12} \left( -\frac{a_{12}}{3a_{13}} \right) + 3a_{13} \left( -\frac{a_{12}}{3a_{13}} \right)^2 & \leq v_{max} & (\text{vel. 2-nd tract} \leq v_{max}) \\ a_{21} + 2a_{22} \left( -\frac{a_{22}}{3a_{23}} \right) + 3a_{23} \left( -\frac{a_{22}}{3a_{23}} \right)^2 & \leq v_{max} & (\text{vel. 3-rd tract} \leq v_{max}) \\ 2a_{02} & \leq a_{max} & (\text{init. acc. 1-st tract} \leq a_{max}) \\ 2a_{12} & \leq a_{max} & (\text{init. acc. 2-nd tract} \leq a_{max}) \\ 2a_{22} & \leq a_{max} & (\text{init. acc. 3-rd tract} \leq a_{max}) \\ 2a_{02} + 6a_{03}T_1 & \leq a_{max} & (\text{final acc. 1-st tract} \leq a_{max}) \\ 2a_{12} + 6a_{13}T_2 & \leq a_{max} & (\text{final acc. 2-nd tract} \leq a_{max}) \\ 2a_{22} + 6a_{23}T_3 & \leq a_{max} & (\text{final acc. 3-rd tract} \leq a_{max}). \end{array} \right.$$

Notice that all the constraints depend on  $T_k$ , since the coefficients  $a_{k,i}$  are computed as a function of the time intervals  $T_k$ . By solving iteratively this optimization problem<sup>6</sup>, after 10 iterations one obtains the following values

$$T_0 = 1.5576, \quad T_1 = 4.4874, \quad T_2 = 4.5537 \quad \Rightarrow \quad T = 10.5987$$

while after 100 iterations

$$T_0 = 1.5551, \quad T_1 = 4.4500, \quad T_2 = 4.5767 \quad \Rightarrow \quad T = 10.5818$$

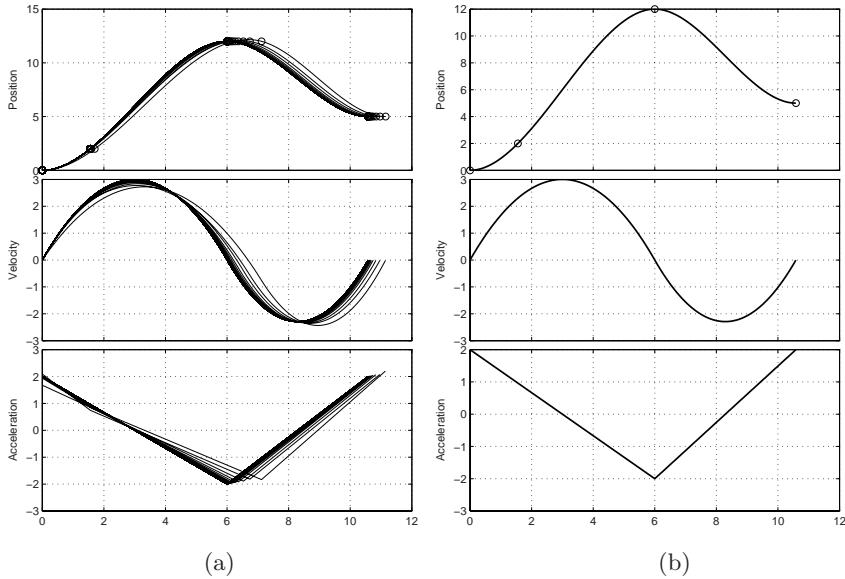
and after 1000 iterations

$$T_0 = 1.5551, \quad T_1 = 4.4500, \quad T_2 = 4.5767 \quad \Rightarrow \quad T = 10.5818.$$

Figure 4.17(a) shows the profiles of the splines obtained during the iterative process. Notice the very fast convergence rate of the procedure towards the “optimal” trajectory, reported in Fig. 4.17(b).  $\square$

---

<sup>6</sup> The first computation of the spline is performed with a cord length distribution of the time instants  $t_k$ .



**Fig. 4.17.** Optimal spline through a set of points: iterative procedure (a) and final trajectory (b).

## 4.5 B-spline Functions for Trajectories with High Degree of Continuity

In some applications, it is required to plan trajectories with continuous derivatives up to a given order  $d > 2$  (e.g. with continuous jerk, snap or even higher order derivatives)[44]. For this purpose, instead of the standard piecewise polynomial form adopted in the previous sections:

$$s(t) = \{q_k(t), t \in [t_k, t_{k+1}], k = 0, \dots, n-1\},$$

$$q_k(t) = a_{k0} + a_{k1}(t - t_k) + \dots + a_{ki}(t - t_k)^i + \dots + a_{kp}(t - t_k)^p$$

it is preferable to use spline functions expressed in the B-form (or B-splines), i.e.

$$s(u) = \sum_{j=0}^m \mathbf{p}_j B_j^p(u), \quad u_{min} \leq u \leq u_{max} \quad (4.40)$$

where  $\mathbf{p}_j$  are the control points and  $B_j^p(u)$  the basis functions of degree  $p$ , defined for the knot vector  $\mathbf{u} = [u_0, \dots, u_{n_{knot}}]$ .

Because of their clear geometrical meaning, B-splines (described in details in Appendix B) are particularly useful for the definition of multi-dimensional parametric curves. For this reason they are adopted in Chapter 8, focussed on path planning in 3D space. On the other hand, an important feature of B-splines is that, by definition, they are  $p - k$  continuous differentiable at

a knot of multiplicity  $k$ , see Sec. B.1. Therefore, if all the internal knots are distinct ( $k = 1$ ), the continuity of velocity and acceleration simply requires the adoption of a B-spline of degree three ( $p = 3$ ). If a continuous jerk is required, then it is necessary to set  $p = 4$ , while the condition  $p = 5$  guarantees also the continuity of the snap. For this reason, B-splines can be profitably used also in one-dimensional problems, since the continuity between contiguous segments within the curve is implicitly obtained. In this case, the independent variable is  $u = t$ , and (4.40) may be rewritten as

$$s(t) = \sum_{j=0}^m p_j B_j^p(t), \quad t_{min} \leq t \leq t_{max} \quad (4.41)$$

where  $p_j$  are scalar parameters.

Given the desired degree  $p$  (selected on the basis of the desired degree of continuity for the B-spline), and the points  $q_k$ ,  $k = 0, \dots, n$  to be interpolated at time instants  $t_k$ , the problem consists in finding the values of the unknown parameters  $p_j$ ,  $j = 0, \dots, m$ , which guarantee that

$$s(t_k) = q_k, \quad k = 0, \dots, n.$$

First of all, it is necessary to define the knot vector  $\mathbf{u}$ . A typical choice is

$$\mathbf{u} = [\underbrace{t_0, \dots, t_0}_{p+1}, t_1, \dots, t_{n-1}, \underbrace{t_n, \dots, t_n}_{p+1}]. \quad (4.42)$$

Therefore the total number of knots is  $n_{knot} + 1 = n + 2p + 1$ . As a consequence, because of the relationship between  $n_{knot}$ ,  $m$ , and  $p$  for a B-spline function (i.e.  $n_{knot} - p - 1 = m$ ), the number of unknown control points  $p_j$  is  $m + 1 = (n + 1) + p - 1$ . Another possibility for the choice of the knot vector is

$$\mathbf{u} = [\underbrace{t_0, \dots, t_0}_{p+1}, (t_0 + t_1)/2, \dots, (t_{k-1} + t_k)/2, \dots, (t_{n-1} + t_n)/2, \underbrace{t_n, \dots, t_n}_{p+1}]. \quad (4.43)$$

In this case, the knots are  $n_{knot} + 1 = n + 2p + 2$ , and, as a consequence, the number of control points to be determined is  $m + 1 = (n + 1) + p$ . With this choice, the interpolation of points  $q_k$  occurs at time instants  $t_k$ , as in the previous case. The difference is that now the segments composing the B-spline trajectory abut at time instants shifted with respect to the  $t_k$ . i.e. at  $t = (t_{k-1} + t_k)/2$ . The adoption of (4.42) or (4.43) is strictly related to the degree of the spline. In particular, as highlighted in the Examples 4.16 and 4.17, the fact that the degree  $p$  is odd or even strongly affects the trajectory profiles obtained with B-spline functions. If  $p$  is odd, the choice (4.42) is preferable while, if  $p$  is even, the knot vector expressed by (4.43) provides better results, [45].

In order to determine the unknown coefficients  $p_j$ ,  $j = 0, \dots, m$ , one can build a linear system by stacking the  $n + 1$  equations obtained by imposing the interpolation conditions of each point  $q_k$  at time  $t_k$ :

$$q_k = \left[ B_0^p(t_k), B_1^p(t_k), \dots, B_{m-1}^p(t_k), B_m^p(t_k) \right] \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{m-1} \\ p_m \end{bmatrix}, \quad k = 0, \dots, n.$$

In this way, a system of  $n + 1$  equations in the  $m + 1$  unknown control points  $p_j$  is obtained. However, in order to have a unique solution, more constraints have to be imposed. In particular,  $p - 1$  or  $p$  additional equations, depending on the choice of  $\mathbf{u}$ , are necessary to obtain a square system with  $m + 1$  equations and  $m + 1$  unknown variables. Typical conditions, beside the via-points interpolation, concern velocities and accelerations (and more generally higher order time derivatives of the curve) at the initial and final points:

$$\begin{aligned} s^{(1)}(t_0) &= \mathbf{v}_0, & s^{(1)}(t_n) &= \mathbf{v}_n \\ s^{(2)}(t_0) &= \mathbf{a}_0, & s^{(2)}(t_n) &= \mathbf{a}_n \\ &\vdots & &\vdots \end{aligned}$$

These constraints can be written as

$$\begin{aligned} \mathbf{v}_k &= \left[ B_0^{p(1)}(t_k), B_1^{p(1)}(t_k), \dots, B_{m-1}^{p(1)}(t_k), B_m^{p(1)}(t_k) \right] \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{m-1} \\ p_m \end{bmatrix}, \quad k = 0, n \\ \mathbf{a}_k &= \left[ B_0^{p(2)}(t_k), B_1^{p(2)}(t_k), \dots, B_{m-1}^{p(2)}(t_k), B_m^{p(2)}(t_k) \right] \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{m-1} \\ p_m \end{bmatrix}, \quad k = 0, n \end{aligned}$$

where  $B_j^{p(i)}(t_k)$  are the  $i$ -th derivatives of the basis functions  $B_j^p(t)$  computed at time instant  $t_k$ . For the calculation of  $B_j^{p(i)}(t_k)$ , see Sec. B.1.

Note that the generic equation

$$s^{(i)}(t_k) = \left[ B_0^{p(i)}(t_k), B_1^{p(i)}(t_k), \dots, B_{m-1}^{p(i)}(t_k), B_m^{p(i)}(t_k) \right] \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{m-1} \\ p_m \end{bmatrix} \quad (4.44)$$

is equivalent to

$$s^{(i)}(t_k) = \sum_{j=0}^m p_j B_j^{p(i)}(t_k).$$

Alternatively, instead of assigning boundary conditions on the velocity or on the acceleration, one may impose the continuity of the curve and of its derivatives at the initial and final time instants (the so-called periodic or cyclic conditions), i.e.

$$s^{(i)}(t_0) = s^{(i)}(t_n)$$

or, in matrix notation,

$$\left[ B_0^{p(i)}(t_0) - B_0^{p(i)}(t_n), B_1^{p(i)}(t_0) - B_1^{p(i)}(t_n), \dots, B_m^{p(i)}(t_0) - B_m^{p(i)}(t_n) \right] \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{m-1} \\ p_m \end{bmatrix} = 0. \quad (4.45)$$

The conditions in (4.44) or (4.45) can be mixed in order to obtain desired profiles.

By setting  $p = 3$ , the standard cubic splines are obtained and, depending on the choice of the two additional conditions, the different cases reported in the previous sections can be easily deduced.

If  $p = 4$  also the jerk profile is continuous. Since with the choice of the knot vector expressed by (4.43), there are four free parameters to be determined, it is possible to assign initial and final velocities and accelerations. This leads to a linear system of  $(n + 1) + 4$  equations in  $(n + 1) + 4$  unknowns (in this case  $m = n + 4$ ):

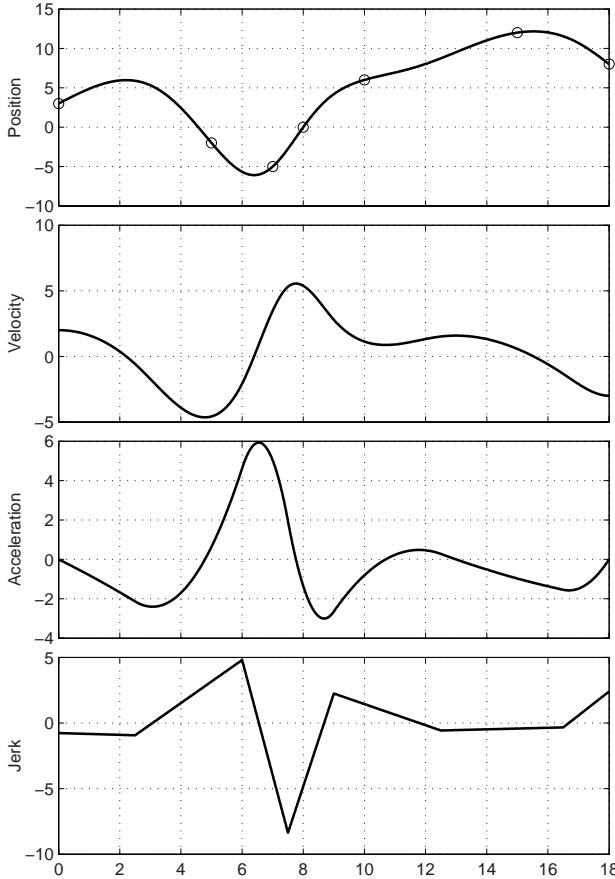
$$\mathbf{A} \mathbf{p} = \mathbf{c} \quad (4.46)$$

where

$$\mathbf{p} = [p_0, p_1, \dots, p_{m-1}, p_m]^T$$

and (with  $p = 4$ )

$$\mathbf{A} = \begin{bmatrix} B_0^p(t_0) & B_1^p(t_0) & \cdots & B_m^p(t_0) \\ B_0^{p(1)}(t_0) & B_1^{p(1)}(t_0) & \cdots & B_m^{p(1)}(t_0) \\ B_0^{p(2)}(t_0) & B_1^{p(2)}(t_0) & \cdots & B_m^{p(2)}(t_0) \\ B_0^p(t_1) & B_1^p(t_1) & \cdots & B_m^p(t_1) \\ \vdots & \vdots & & \vdots \\ B_0^p(t_{n-1}) & B_1^p(t_{n-1}) & \cdots & B_m^p(t_{n-1}) \\ B_0^{p(2)}(t_n) & B_1^{p(2)}(t_n) & \cdots & B_m^{p(2)}(t_n) \\ B_0^{p(1)}(t_n) & B_1^{p(1)}(t_n) & \cdots & B_m^{p(1)}(t_n) \\ B_0^p(t_n) & B_1^p(t_n) & \cdots & B_m^p(t_n) \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} q_0 \\ v_0 \\ a_0 \\ q_1 \\ \vdots \\ q_{n-1} \\ a_n \\ v_n \\ q_n \end{bmatrix}. \quad (4.47)$$



**Fig. 4.18.** B-spline trajectory with continuity of the jerk ( $p = 4$ ), obtained by computing the knot vector as in (4.43).

The control points  $p_j$ ,  $j = 0, \dots, m$ , are obtained by solving (4.46), and the B-spline can be evaluated for any value of  $t \in [t_0, t_n]$  according to the algorithm reported in Appendix B.

**Example 4.16** Fig. 4.18 shows a spline computed according to the above algorithm (with  $p = 4$ ). The trajectory is computed by considering the interpolation of the following points

$$\begin{aligned} t_0 &= 0, & t_1 &= 5, & t_2 &= 7, & t_3 &= 8, & t_4 &= 10, & t_5 &= 15, & t_6 &= 18, \\ q_0 &= 3, & q_1 &= -2, & q_2 &= -5, & q_3 &= 0, & q_4 &= 6, & q_5 &= 12, & q_6 &= 8, \end{aligned}$$

and with the boundary conditions

$$v_0 = 2, \quad v_6 = -3, \quad a_0 = 0, \quad a_6 = 0.$$

With the knot vector

$$\mathbf{u} = [0, 0, 0, 0, 0, 2.5, 6, 7.5, 9, 12.5, 16.5, 18, 18, 18, 18, 18]$$

the resulting matrix  $\mathbf{A}$  is

$$\mathbf{A} = \begin{bmatrix} 1.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.60 & 1.60 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.92 & -2.72 & 0.80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.00 & 0.08 & 0.43 & 0.44 & 0.03 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.00 & 0.05 & 0.58 & 0.34 & 0.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.00 & 0.34 & 0.59 & 0.05 & 0.00 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.03 & 0.50 & 0.40 & 0.05 & 0.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.00 & 0.08 & 0.39 & 0.45 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.45 & -6.78 & 5.33 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.66 & 2.66 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.00 \end{bmatrix}$$

and the known variables are

$$\mathbf{c} = [3, 2, 0, -2, -5, 0, 6, 12, 0, -3, 8]^T.$$

From (4.46) the control points defining the trajectory shown in Fig. 4.18 result

$$\mathbf{p} = [3, 4.25, 7.25, 7.39, -13.66, 7.44, 4.98, 12.59, 13.25, 9.12, 8].$$

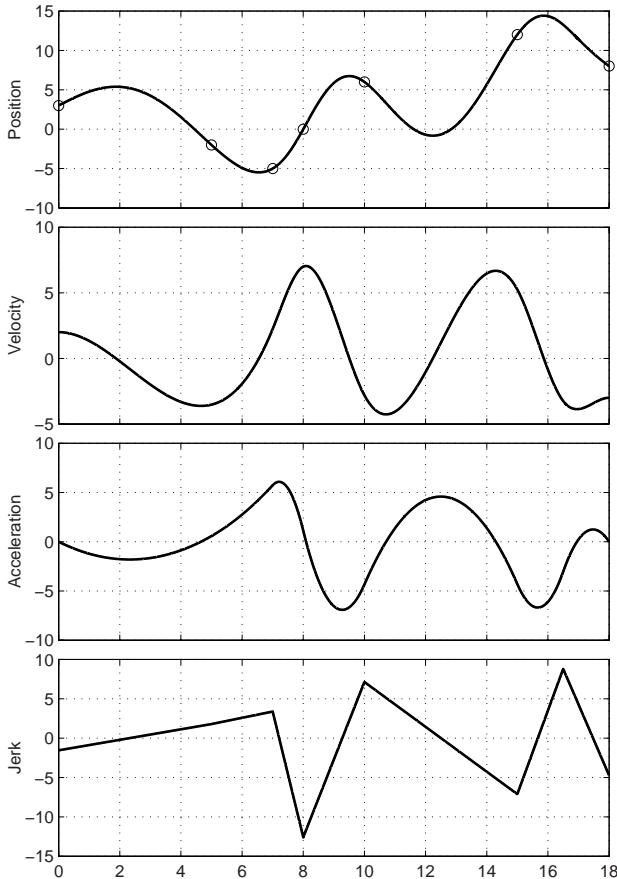
If one computes the knot vector according to (4.42), the trajectory obtained by interpolating the point  $q_k$  is still satisfactory (see Fig. 4.19), but it presents more pronounced oscillations, and as a consequence the peak values of velocity and acceleration are higher than in the previous case. Note that by adopting (4.42), the knot vector is

$$\mathbf{u} = [0, 0, 0, 0, 0, 5, 7, 8, 10, 15, 16.5, 18, 18, 18, 18] \quad (4.48)$$

where an extra-knot at 16.5 has been inserted in the standard knot vector expression, in order to impose the four additional constraints. As a matter of fact, the choice of (4.42) leads to  $p - 1$  (in this case three) free parameters which must be determined by imposing the boundary conditions. Since, in this example, the number of free parameters is smaller than the number of desired conditions, one has to increase the number of unknown control points  $p_j$ . This can be obtained by increasing the number of knots. For instance, the knot vector (4.48) is obtained by assuming

$$\mathbf{u} = [\underbrace{t_0, \dots, t_0}_5, t_1, \dots, t_{n-1}, (t_n + t_{n-1})/2, \underbrace{t_n, \dots, t_n}_5]$$

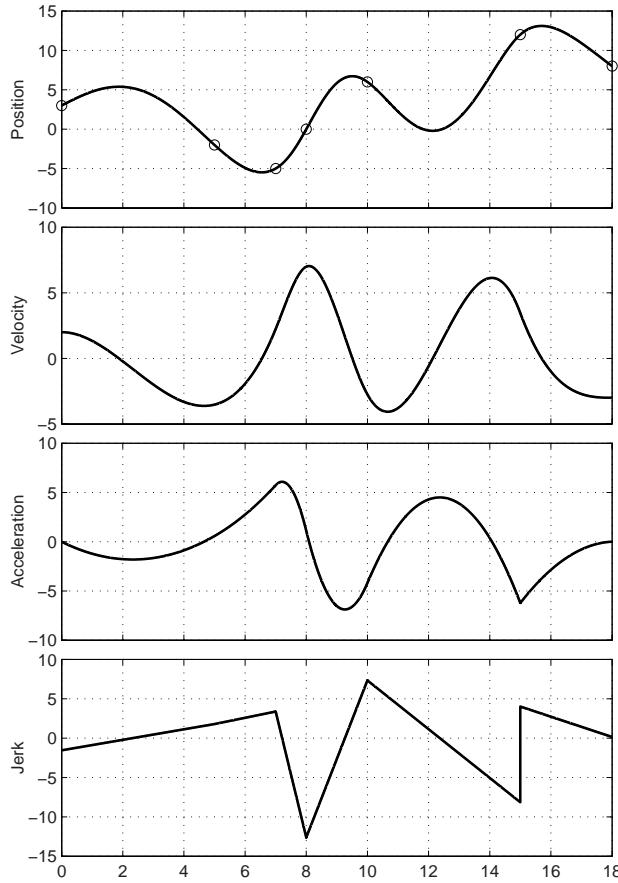
but it is worth noticing that additional knots can be added in any position of  $\mathbf{u}$ , with the only constraint of avoiding coincident knots which would produce a reduction of the continuity order of the trajectory. In Fig. 4.20 a trajectory



**Fig. 4.19.** B-spline trajectory of degree 4, obtained by computing the knot vector as in (4.42).

computed with a double knot at 15 is shown. Note the discontinuity on the jerk at  $t = 15$ . □

For  $p = 4$ , one can assume cyclic conditions for the velocity, acceleration, jerk and, since the choice (4.43) requires four additional constraints, also for the snap (which in the interior of the trajectory will be however discontinuous). Alternatively, besides the periodic conditions on the first three derivatives, one can assume a boundary condition on the velocity or acceleration. The system (4.46) is now defined by



**Fig. 4.20.** B-spline trajectory of degree 4, with a double knot at 15.

$$\mathbf{A} = \begin{bmatrix} B_0^p(t_0) & B_1^p(t_0) & \cdots & B_m^p(t_0) \\ B_0^p(t_1) & B_1^p(t_1) & \cdots & B_m^p(t_1) \\ \vdots & \vdots & & \vdots \\ B_0^p(t_{n-1}) & B_1^p(t_{n-1}) & \cdots & B_m^p(t_{n-1}) \\ B_0^p(t_n) & B_1^p(t_n) & \cdots & B_m^p(t_n) \\ B_0^{p(1)}(t_n) - B_0^{p(1)}(t_0) & B_1^{p(1)}(t_n) - B_1^{p(1)}(t_0) & \cdots & B_m^{p(1)}(t_n) - B_m^{p(1)}(t_0) \\ B_0^{p(2)}(t_n) - B_0^{p(2)}(t_0) & B_1^{p(2)}(t_n) - B_1^{p(2)}(t_0) & \cdots & B_m^{p(2)}(t_n) - B_m^{p(2)}(t_0) \\ B_0^{p(3)}(t_n) - B_0^{p(3)}(t_0) & B_1^{p(3)}(t_n) - B_1^{p(3)}(t_0) & \cdots & B_m^{p(3)}(t_n) - B_m^{p(3)}(t_0) \\ B_0^{p(4)}(t_n) - B_0^{p(3)}(t_0) & B_1^{p(4)}(t_n) - B_1^{p(3)}(t_0) & \cdots & B_m^{p(4)}(t_n) - B_m^{p(4)}(t_0) \end{bmatrix}$$

$$\mathbf{c} = [q_0, q_1, \dots, q_{n-1}, q_n, 0, 0, 0, 0]^T.$$

**Example 4.17** A B-spline trajectory of degree 4 interpolating the points

$$\begin{aligned} t_0 &= 0, & t_1 &= 5, & t_2 &= 7, & t_3 &= 8, & t_4 &= 10, & t_5 &= 15, & t_6 &= 18, \\ q_0 &= 3, & q_1 &= -2, & q_2 &= -5, & q_3 &= 0, & q_4 &= 6, & q_5 &= 12, & q_6 &= 3, \end{aligned}$$

is defined by imposing the periodic conditions

$$\begin{aligned} s^{(1)}(t_0) &= s^{(1)}(t_6) \\ s^{(2)}(t_0) &= s^{(2)}(t_6) \\ s^{(3)}(t_0) &= s^{(3)}(t_6) \\ s^{(4)}(t_0) &= s^{(4)}(t_6). \end{aligned}$$

With the use of the knot vector

$$\mathbf{u} = [0, 0, 0, 0, 2.5, 6, 7.5, 9, 12.5, 16.5, 18, 18, 18, 18, 18]$$

the resulting matrix  $\mathbf{A}$  and vector  $\mathbf{c}$  are

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.00 & 0.08 & 0.43 & 0.44 & 0.03 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.00 & 0.05 & 0.58 & 0.34 & 0.00 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.00 & 0.34 & 0.59 & 0.05 & 0.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.03 & 0.50 & 0.40 & 0.05 & 0.00 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.00 & 0.08 & 0.39 & 0.45 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1.60 & 1.60 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.66 & -2.66 & \\ 1.92 & -2.72 & 0.80 & 0 & 0 & 0 & 0 & 0 & -1.45 & 6.78 & -5.33 & \\ -1.53 & 2.44 & -1.12 & 0.21 & 0 & 0 & 0 & 0.32 & -2.79 & 9.57 & -7.11 & \\ 0.61 & -1.02 & 0.55 & -0.17 & 0.02 & 0 & -0.03 & 0.34 & -2.05 & 6.48 & -4.74 & \end{bmatrix}$$

and

$$\mathbf{c} = [3, 2, -2, -5, 0, 6, 12, 3, 0, 0, 0, 0]^T.$$

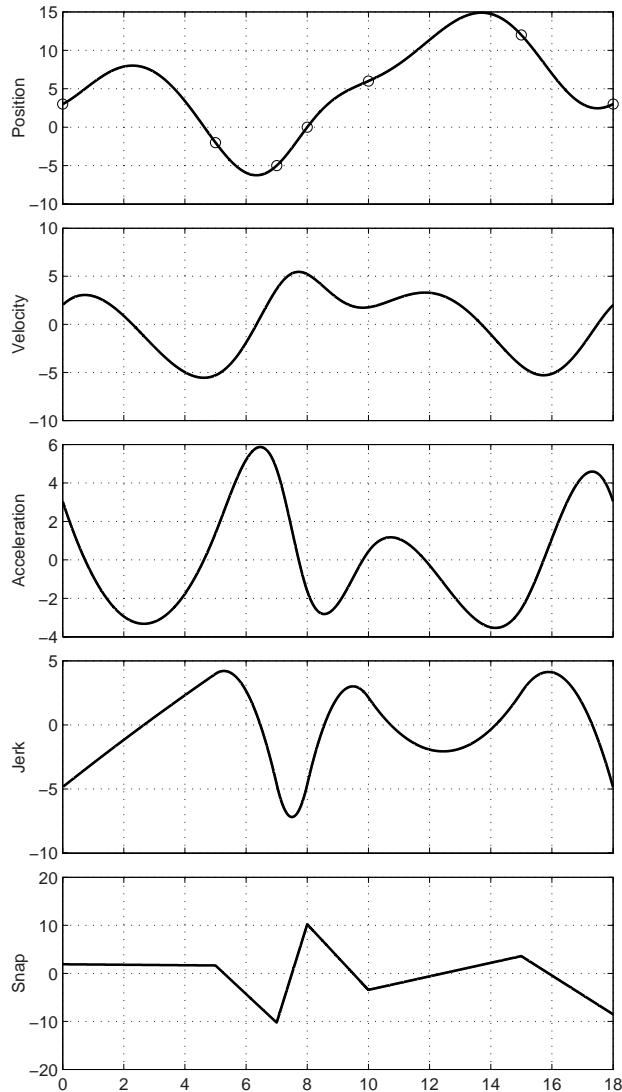
The control points obtained with these values are

$$\mathbf{p} = [3, 1.82, 1.30, 8.88, -13.96, 7.73, 3.66, 20.41, 7.53, 3.70, 3]^T.$$

The profiles of the B-spline trajectory and of its derivatives are shown in Fig. 4.21.

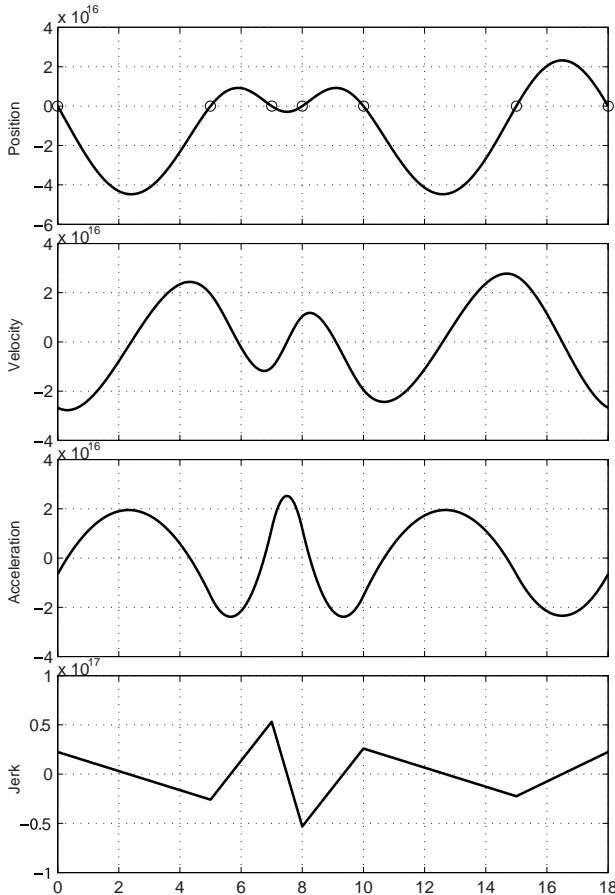
By considering the same via-points of the above example, and by adopting a knot vector of the form (4.42) with periodic conditions on the velocity, acceleration and jerk, the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.02 & 0.23 & 0.51 & 0.22 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.00 & 0.24 & 0.69 & 0.05 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0.69 & 0.24 & 0.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.22 & 0.58 & 0.18 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.03 & 0.23 & 0.49 & 0.24 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -0.80 & 0.80 & 0 & 0 & 0 & 0 & 0 & 0 & 1.33 & -1.33 & \\ 0.48 & -0.82 & 0.34 & 0 & 0 & 0 & 0 & -0.50 & 1.83 & -1.33 & \\ -0.19 & 0.42 & -0.32 & 0.08 & 0 & 0 & 0.10 & -0.55 & 1.34 & -0.88 & \end{bmatrix}$$



**Fig. 4.21.** B-spline trajectory of degree 4 with cyclic conditions.

results ill-conditioned. As a consequence, it leads to a solution which cannot be used in practice, see Fig. 4.22 where the values of velocities, accelerations and jerk are shown to range in  $\pm 10^{17}$ .  $\square$



**Fig. 4.22.** B-spline trajectory of degree 4 with cyclic conditions and with the knot vector  $\mathbf{u} = [0, 0, 0, 0, 0, 5, 7, 8, 10, 15, 18, 18, 18, 18, 18]$ .

With  $p = 5$ , the knots should be selected according to (4.42), i.e.

$$\mathbf{u} = [\underbrace{t_0, \dots, t_0}_{6}, t_1, \dots, t_{n-1}, \underbrace{t_n, \dots, t_n}_{6}].$$

It is therefore possible to impose the desired values of the initial and final velocities/accelerations (four conditions). Moreover, this value of  $p$  guarantees the continuity of the trajectory up to the fourth derivative (snap). The system (4.46) with the expressions of  $\mathbf{A}$  and  $\mathbf{c}$  reported in (4.47) remains unchanged.

**Example 4.18** The data (via-points and constraints) of Example 4.16 are used to build a B-spline of degree 5, with the knot vector constructed as in (4.42). By solving (4.46), with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.8 & -1.37 & 0.57 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0.10 & 0.40 & 0.40 & 0.07 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.00 & 0.07 & 0.54 & 0.37 & 0.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0.36 & 0.56 & 0.06 & 0.00 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.07 & 0.50 & 0.35 & 0.05 & 0.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.08 & 0.31 & 0.43 & 0.15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.83 & -3.05 & 2.22 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.66 & 1.66 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{c} = [3, 2, 0, -2, -5, 0, 6, 12, 0, -3, 8]^T,$$

the control points

$$\mathbf{p} = [35, 7.80, 9.69, -18.83, 12.01, 1.5137, 12.33, 14.60, 9.80, 8]^T$$

are obtained. The profiles of the B-spline trajectory and of its derivatives are shown in Fig. 4.23, superimposed to the ones of Example 4.16 (dashed) which are relative to a B-spline trajectory of degree 4. Note the small differences between the two position profiles.  $\square$

If  $p = 5$ , the four additional constraints can also be determined by mixing boundary conditions on the velocity, and periodic conditions on the acceleration and jerk. In this way, the matrix  $\mathbf{A}$  and the vector  $\mathbf{c}$  in (4.46) become

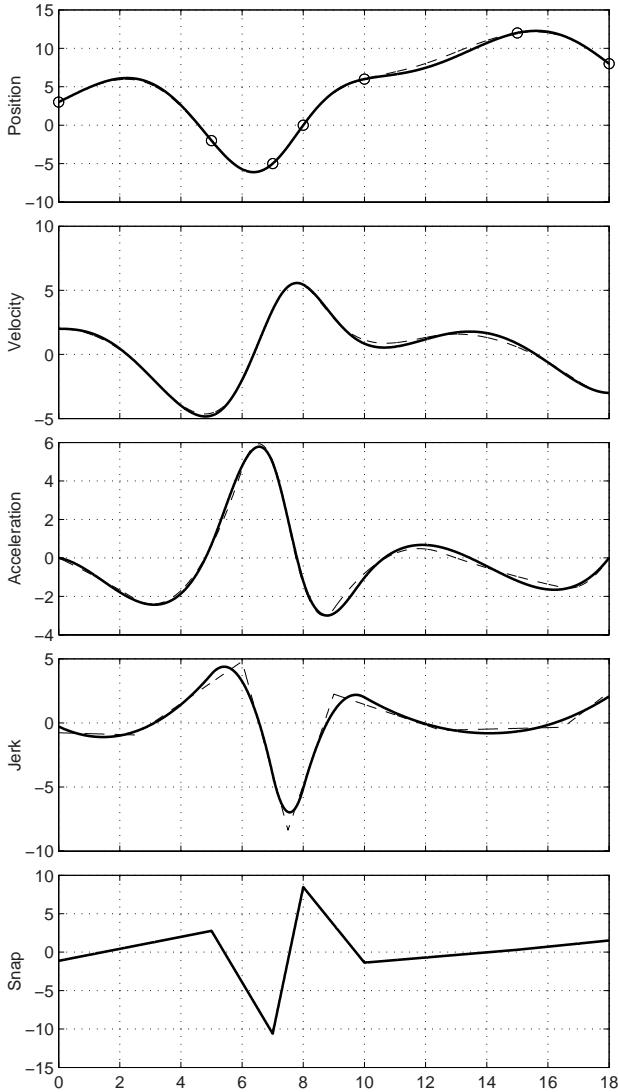
$$\mathbf{A} = \begin{bmatrix} B_0^p(t_0) & B_1^p(t_0) & \cdots & B_m^p(t_0) \\ B_0^{p(1)}(t_0) & B_1^{p(1)}(t_0) & \cdots & B_m^{p(1)}(t_0) \\ B_0^p(t_1) & B_1^p(t_1) & \cdots & B_m^p(t_1) \\ \vdots & \vdots & & \vdots \\ B_0^p(t_{n-1}) & B_1^p(t_{n-1}) & \cdots & B_m^p(t_{n-1}) \\ B_0^{p(1)}(t_n) & B_1^{p(1)}(t_n) & \cdots & B_m^{p(1)}(t_n) \\ B_0^p(t_n) & B_1^p(t_n) & \cdots & B_m^p(t_n) \\ B_0^{p(2)}(t_n) - B_0^{p(2)}(t_0) & B_1^{p(2)}(t_n) - B_1^{p(2)}(t_0) & \cdots & B_m^{p(2)}(t_n) - B_m^{p(2)}(t_0) \\ B_0^{p(3)}(t_n) - B_0^{p(3)}(t_0) & B_1^{p(3)}(t_n) - B_1^{p(3)}(t_0) & \cdots & B_m^{p(3)}(t_n) - B_m^{p(3)}(t_0) \end{bmatrix}$$

and

$$\mathbf{c} = [q_0, v_0, q_1, \dots, q_{n-1}, v_n, q_n, 0, 0]^T.$$

**Example 4.19** A B-spline trajectory of degree 5 interpolating the points

$$\begin{aligned} t_0 &= 0, & t_1 &= 5, & t_2 &= 7, & t_3 &= 8, & t_4 &= 10, & t_5 &= 15, & t_6 &= 18, \\ q_0 &= 3, & q_1 &= -2, & q_2 &= -5, & q_3 &= 0, & q_4 &= 6, & q_5 &= 12, & q_6 &= 3, \end{aligned}$$



**Fig. 4.23.** B-spline trajectory with continuity of the snap ( $p = 5$ ).

is defined by imposing the conditions

$$s^{(1)}(t_0) = 2, \quad s^{(1)}(t_6) = 2$$

and

$$\begin{aligned} s^{(2)}(t_0) &= s^{(2)}(t_6) \\ s^{(3)}(t_0) &= s^{(3)}(t_6). \end{aligned}$$

The resulting matrix  $\mathbf{A}$  and vector  $\mathbf{c}$  are

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0.10 & 0.40 & 0.40 & 0.07 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.00 & 0.07 & 0.54 & 0.37 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0.36 & 0.56 & 0.06 & 0.00 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.07 & 0.50 & 0.35 & 0.05 & 0.00 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0.08 & 0.31 & 0.43 & 0.15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.66 & 1.66 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0.80 & -1.37 & 0.57 & 0 & 0 & 0 & 0 & 0 & -0.83 & 3.05 & -2.22 \\ -0.48 & 1.06 & -0.80 & 0.21 & 0 & 0 & 0 & 0.25 & -1.39 & 3.36 & -2.22 \end{bmatrix}$$

and

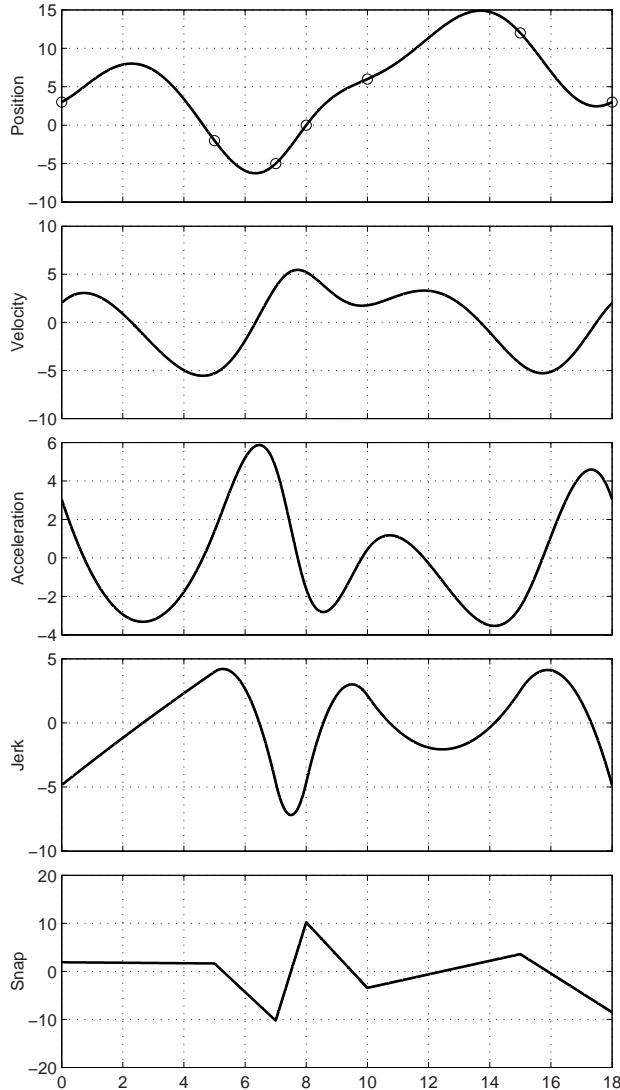
$$\mathbf{c} = [3, 2, -2, -5, 0, 6, 12, 2, 3, 0, 0, 0]^T.$$

The control points obtained with these values are

$$\mathbf{p} = [3, 5.00, 13.13, 8.34, -18.96, 12.57, -2.81, 34.44, 2.26, 1.80, 3]^T$$

The profiles of the B-spline trajectory and of its derivatives are shown in Fig. 4.24.  $\square$

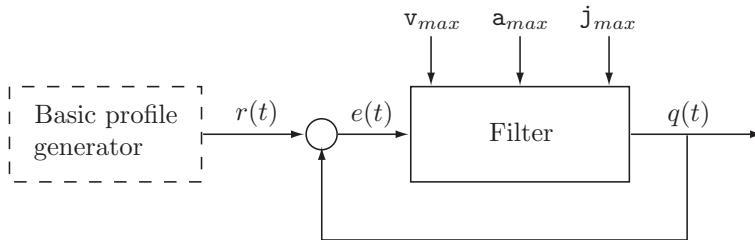
In general, to build a  $r$  times differentiable trajectory which interpolates  $n+1$  points, it is sufficient to consider a B-spline of degree  $p = r+1$ . If  $p$  is odd, it is convenient to assume a knot distribution as in (4.42) while, if  $p$  is even, a knot vector of the kind (4.43) provides better results. In the former case,  $p - 1$  additional constraints are necessary to have a unique solution (in this case the unknown control points are  $m + 1 = n + p$ ) while, in the latter one,  $p$  additional condition are needed (in this case the unknown control points are  $m + 1 = n + p + 1$ ). By adding further knots (and accordingly by increasing the number of control points  $p_j$ ), it is also possible to consider additional constraints.



**Fig. 4.24.** B-spline trajectory with continuity of the snap ( $p = 5$ ) computed with periodic conditions.

## 4.6 Nonlinear Filters for Optimal Trajectory Planning

The computation of trajectories with constraints on velocity, acceleration, and jerk is of particular interest for the control of multi-axis automatic machines. In general, this problem is solved *offline*, by defining a function satisfying the desired conditions on the interpolated points and on the maximum values of



**Fig. 4.25.** Conceptual scheme of the nonlinear filter for online optimal trajectory generation.

velocity, acceleration and jerk. Usually, if any of these conditions changes, the whole trajectory must be recomputed.

The approach presented in this section is conceptually different: a dynamic nonlinear filter is used in cascade to a trajectory generator which provides only basic motion profiles, such as steps or ramps, that, although very simple, are in general not directly applicable in industrial tasks. The goal of the filter is to process *online* these basic profiles and provide, as output, feasible trajectories satisfying the given constraints. As schematically shown in Fig. 4.25, the nonlinear filter is based on a feedback scheme, which guarantees that the output signal  $q(t)$ , subject to constraints on maximum values of its derivatives, follows as well as possible the external reference  $r(t)$ .

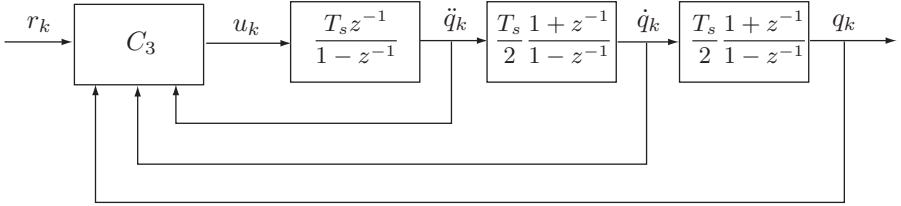
For instance, a step reference may be tracked in minimum time, in compliance with the constraints, producing a continuous motion profile, with velocity and acceleration also continuous and jerk piecewise constant (in this case the trajectory generator is called “third order filter”) or with continuous velocity and piecewise constant acceleration (in the case of a “second order filter”). The design of the filter is based on the well known nonlinear variable structure control theory, for which a wide scientific literature is available, [46, 47]. In the following, only the algorithmic results are reported, trying to simplify as much as possible the theoretical presentation.

#### 4.6.1 Online trajectory planner with velocity, acceleration and jerk constraints

This trajectory planner is able to transform in real-time any standard reference signal  $r(t)$ , received as input, in a smooth signal  $q(t)$  which satisfies the following constraints:

$$\begin{aligned} v_{min} &\leq \dot{q}(t) \leq v_{max} \\ a_{min} &\leq \ddot{q}(t) \leq a_{max} \\ -U = j_{min} &\leq q^{(3)}(t) \leq j_{max} = U. \end{aligned}$$

The input signal  $r(t)$  can be generated by a simple trajectory planner, which provides only simple motion profiles such as steps or ramps. Note that it is



**Fig. 4.26.** Third order filter for online optimal trajectory generation.

also possible to use this trajectory generator to filter a signal generated at runtime by another device, as in tracking operations, or directly provided by a human operator, e.g. by means of a teach-pendant.

Since the filter is implemented on digital controllers, for its definition it is convenient to adopt a discrete time formulation. The digital realization of the trajectory planner has been obtained by discretizing the continuous time version reported in [48]. The scheme of the trajectory generator is shown in Fig. 4.26. At each time instant  $t_k = kT_s$ ,  $k = 1, 2, \dots$ , the variable structure controller  $C_3$  receives the reference signal  $r_k$  (and its derivatives  $\dot{r}_k$  and  $\ddot{r}_k$ ) and the current values of position, velocity, and acceleration ( $q_k$ ,  $\dot{q}_k$ ,  $\ddot{q}_k$  respectively), and computes the value of the control action  $u_k$ . This control variable corresponds to the desired jerk, which must be integrated three times to obtain the position profile. The integration is performed by means of a rectangular approximation for the acceleration, i.e.

$$\ddot{q}_k = \ddot{q}_{k-1} + T_s u_{k-1}$$

while the trapezoidal approximation is used for the velocity and the position:

$$\begin{aligned}\dot{q}_k &= \dot{q}_{k-1} + \frac{T_s}{2}(\ddot{q}_k + \ddot{q}_{k-1}) \\ q_k &= q_{k-1} + \frac{T_s}{2}(\dot{q}_k + \dot{q}_{k-1}).\end{aligned}$$

The controller  $C_3$  is based on the tracking error between the reference signal  $r_k$  and the output position  $q_k$ .

Let us define the normalized error variables

$$e_k = \frac{q_k - r_k}{U}, \quad \dot{e}_k = \frac{\dot{q}_k - \dot{r}_k}{U}, \quad \ddot{e}_k = \frac{\ddot{q}_k - \ddot{r}_k}{U}$$

where  $U$  is the maximum value of the control action  $u_k$ . The constraints on maximum/minimum velocity and acceleration are translated into constraints on  $\dot{e}_k$  and  $\ddot{e}_k$ :

$$\begin{aligned}\dot{e}_{min} &= \frac{v_{min} - \dot{r}_k}{U}, & \dot{e}_{max} &= \frac{v_{max} - \dot{r}_k}{U}, \\ \ddot{e}_{min} &= \frac{a_{min} - \ddot{r}_k}{U}, & \ddot{e}_{max} &= \frac{a_{max} - \ddot{r}_k}{U}.\end{aligned}$$

It is worth noticing that these constraints are not constant, but depend on  $\dot{r}_k$ ,  $\ddot{r}_k$ , and therefore they must be re-computed at each sampling time. On the other hand, this implies that it is possible to change online the limits on velocity and acceleration  $v_{min}$ ,  $v_{max}$ ,  $a_{min}$ ,  $a_{max}$ : the controller will then change the current velocity or acceleration in order to match the new constraints.

The controller  $C_3$  in Fig. 4.26 is defined as

$$C_3 : \begin{cases} \delta = \dot{e}_k + \frac{\ddot{e}_k |\ddot{e}_k|}{2} \\ \sigma = e_k + \dot{e}_k \ddot{e}_k s_\delta - \frac{\ddot{e}_k^3}{6}(1 - 3|s_\delta|) + \frac{s_\delta}{4} \sqrt{2[\ddot{e}_k^2 + 2\dot{e}_k s_\delta]^3} \\ \nu^+ = e_k - \frac{\ddot{e}_{max}(\ddot{e}_k^2 - 2\dot{e}_k)}{4} - \frac{(\ddot{e}_k^2 - 2\dot{e}_k)^2}{8\ddot{e}_{max}} - \frac{\ddot{e}_k(3\dot{e}_k - \ddot{e}_k^2)}{3} \\ \nu^- = e_k - \frac{\ddot{e}_{min}(\ddot{e}_k^2 + 2\dot{e}_k)}{4} - \frac{(\ddot{e}_k^2 + 2\dot{e}_k)^2}{8\ddot{e}_{min}} + \frac{\ddot{e}_k(3\dot{e}_k + \ddot{e}_k^2)}{3} \\ \Sigma = \begin{cases} \nu^+ & \text{if } \ddot{e}_k \leq \ddot{e}_{max} \text{ and } \dot{e}_k \leq \frac{\ddot{e}_k^2}{2} - \ddot{e}_{max}^2 \\ \nu^- & \text{if } \ddot{e}_k \geq \ddot{e}_{min} \text{ and } \dot{e}_k \geq \ddot{e}_{min}^2 - \frac{\ddot{e}_k^2}{2} \\ \sigma & \text{otherwise} \end{cases} \\ u_c = -U \operatorname{sign}(\Sigma + (1 - |\operatorname{sign}(\Sigma)|)[\delta + (1 - |s_\delta|)\ddot{e}_k]) \\ u_k = \max \{u_v(\dot{e}_{min}), \min\{u_c, u_v(\dot{e}_{max})\}\} \end{cases} \quad (4.49)$$

where  $s_\delta = \operatorname{sign}(\delta)$  and  $\operatorname{sign}(\cdot)$  is the sign function

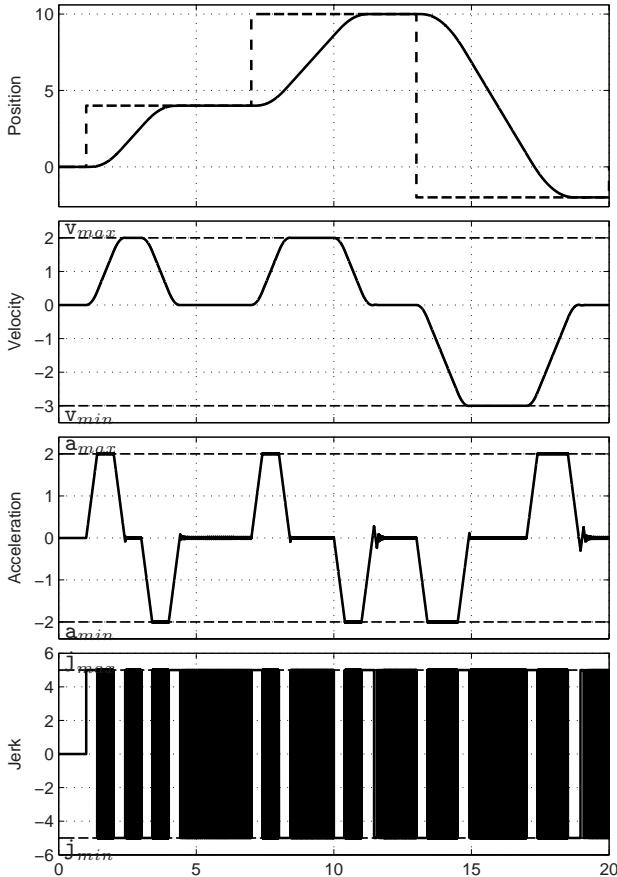
$$\operatorname{sign}(x) = \begin{cases} +1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0. \end{cases}$$

The function  $u_v(\cdot)$ , if used as control action  $u_k = u_v(v)$ , forces the “velocity”  $\dot{e}_k$  to reach the value  $\dot{e}_k = v$  in minimum time. It is defined as

$$C_v : \begin{cases} u_v(v) = \max \{u_a(\ddot{e}_{min}), \min\{u_{cv}(v), u_a(\dot{e}_{max})\}\} \\ u_{cv}(v) = -U \operatorname{sign}(\delta_v(v) + (1 - |\operatorname{sign}(\delta_v(v))|)\ddot{e}_k) \\ \delta_v(v) = \ddot{e}_k |\ddot{e}_k| + 2(\dot{e}_k - v) \\ u_a(a) = -U \operatorname{sign}(\ddot{e}_k - a). \end{cases} \quad (4.50)$$

The result is a trajectory composed by double S segments, which tracks the reference signal  $r_k$  in an optimal way, see Fig. 4.27.

Unfortunately, the discretization process has some drawbacks. In particular, the output position  $q_k$  is characterized by small overshoots with respect

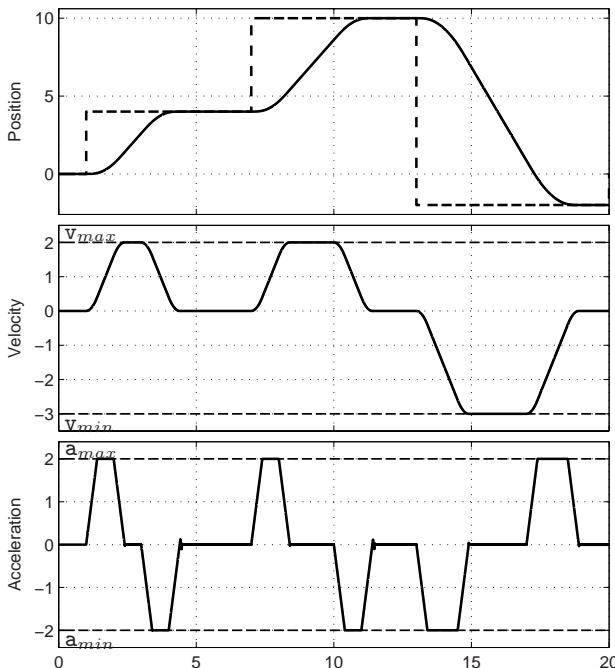


**Fig. 4.27.** Double S trajectory profiles (position, velocity, acceleration and jerk) computed by means of the third order nonlinear filter, with  $T_s = 0.001\text{s}$ .

to the desired position profile. Moreover, the filter  $C_3$  is affected by chattering on the controlled variable  $u_k$  when the value of the jerk should be zero. In general, this is not a problem, since the three integrators filter the signal  $u_k$  and produce in any case smooth profiles.

**Example 4.20** Fig. 4.27 shows position, velocity, acceleration and jerk of a trajectory computed with the third order nonlinear filter. The reference signal  $r_k$  is a sequence of three step functions: the first of magnitude 4 applied at  $t = 1$ , the second of magnitude 6 at time  $t = 7$ , and the last one of amplitude  $-12$  applied at  $t = 13$ . The constraints are

$$\begin{aligned} v_{min} &= -3, & v_{max} &= 2 \\ a_{min} &= -2, & a_{max} &= 2 \end{aligned}$$



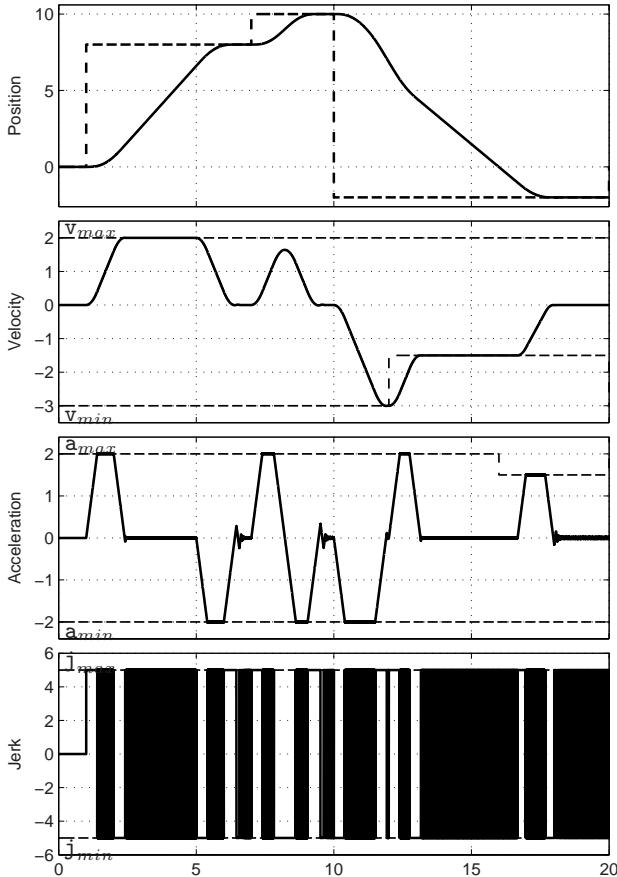
**Fig. 4.28.** Double S trajectory profiles (position, velocity, acceleration and jerk) computed by means of the third order nonlinear filter; in this case  $T_s = 0.0001s$ .

while  $U = j_{max} = 5$ . Note that the minimum and the maximum values of velocity and acceleration are asymmetric. The sampling time is  $T_s = 0.001s$ .  $\square$

From the acceleration profile of the previous example it results that the trajectory (or, more precisely, each segment of the trajectory) is not an ideal double S, but is characterized by small overshoots. The amplitude of this error strictly depends on the sampling time, as shown in Fig. 4.28, which reports a trajectory calculated with the same boundary conditions but with  $T_s = 0.0001s$ .

As already mentioned, an interesting aspect of this method for trajectory generation with respect to other techniques, as those introduced in Sections 3.2, 3.4, is the possibility to change at runtime the constraints for the motion planning.

**Example 4.21** Fig. 4.29 shows position, velocity, acceleration and jerk for a trajectory computed by means of the third order nonlinear filter, when the reference signal  $r_k$  is a piecewise constant signal and the constraints are changed during the computation of the trajectory. The limit values are

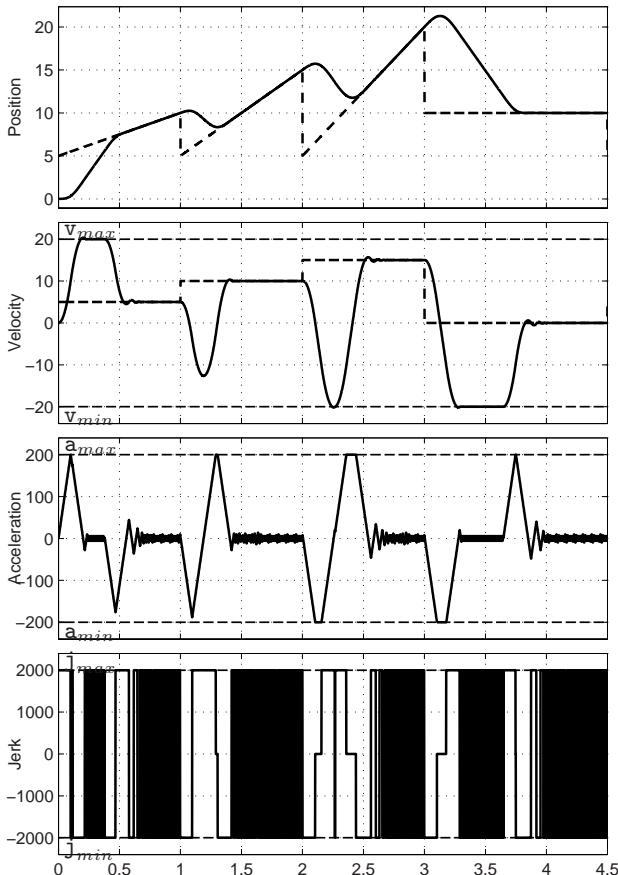


**Fig. 4.29.** Double S trajectory profiles computed by means of the third order nonlinear filter with variable constraints.

$$\begin{aligned} v_{min} &= -3 \xrightarrow{t=12} -1.5, & v_{max} &= 2 \\ a_{min} &= -2, & a_{max} &= 2 \xrightarrow{t=16} 1.5 \end{aligned}$$

and  $U = j_{max} = 5$ . It is worth noticing that when the limit velocity ( $v_{min}$ ) is modified, the filter reduce the output velocity as fast as possible in order to satisfy the new constraint.  $\square$

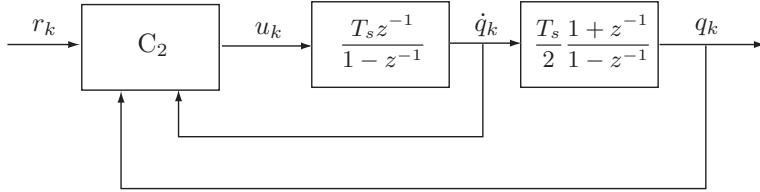
An interesting application of this method for trajectory planning is the tracking of an unknown reference signal. A typical example is given by an automatic machine, working on mechanical parts transported by a conveyor belt and disposed in a casual way. In this case, the tool must be synchronized with the object on the belt (whose velocity is constant and known). The tool must wait and “catch” the pieces and, therefore, the motion task must



**Fig. 4.30.** Output (position, velocity, acceleration and jerk) of the third order nonlinear filter with a “sawtooth” reference signal.

be planned online. This problem is quite complex with offline algorithmic techniques, but can be easily solved by using the nonlinear filter.

**Example 4.22** Fig. 4.30 shows position, velocity, acceleration and jerk for a trajectory computed by means of the third order nonlinear filter, when the reference signal  $r_k$  has a “sawtooth” profile. Note that in this case the controller tries to reduce the position “error” as quickly as possible, and then follows the input signal until a new change occurs. In order to achieve a good tracking of the desired motion, it is important to provide the filter with  $r_k$  and also  $\dot{r}_k$  (in previous examples the values  $\dot{r}_k = 0$ , and  $\ddot{r}_k = 0$  have been considered). □



**Fig. 4.31.** Second order filter for online optimal trajectory generation.

#### 4.6.2 Online trajectory planner with velocity and acceleration constraints

If a limited jerk is not required, and it is desirable to have a low computational burden, it may be preferable to adopt a so-called “second order” filter, designed to reduce the position error  $q_k - r_k$  as quickly as possible while satisfying the constraints

$$\begin{aligned} |\dot{q}_k| &\leq v_{max} \\ |u_k| = |\ddot{q}_k| &\leq a_{max} = U \end{aligned}$$

where  $\dot{q}_k$ ,  $\ddot{q}_k$  are respectively the trajectory velocity and acceleration at time  $kT_s$  (being  $T_s$  the sampling time).

Figure 4.31 shows the block-scheme of the trajectory generator, similar to the third order filter. In this case, the variable structure controller has been directly designed in discrete time [49, 50] and it is not affected by the non-idealities characterizing the implementation of the third order filter, namely chattering and overshoot with respect to the desired position.

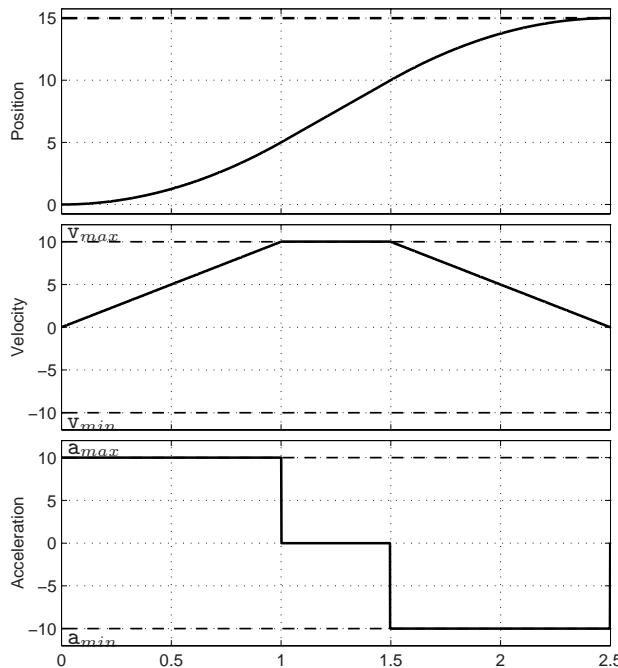
Therefore, when the reference signal  $r_k$  is a step displacement, the output motion is perfectly equivalent to the trapezoidal trajectory of Sec. 3.2 (see Fig. 4.32) with, on the other hand, all the advantages of online planning:

- Possibility to change the reference signal  $r_k$  at any instant.
- Possibility to filter time-varying reference signals.
- Possibility to change the constraints at runtime.

Once the (normalized) tracking errors for the position and velocity have been defined as

$$e_k = \frac{q_k - r_k}{U}, \quad \dot{e}_k = \frac{\dot{q}_k - \dot{r}_k}{U}$$

the control signal  $u_k = \ddot{q}_k$  is computed at each time instant  $t_k = kT_s$  as



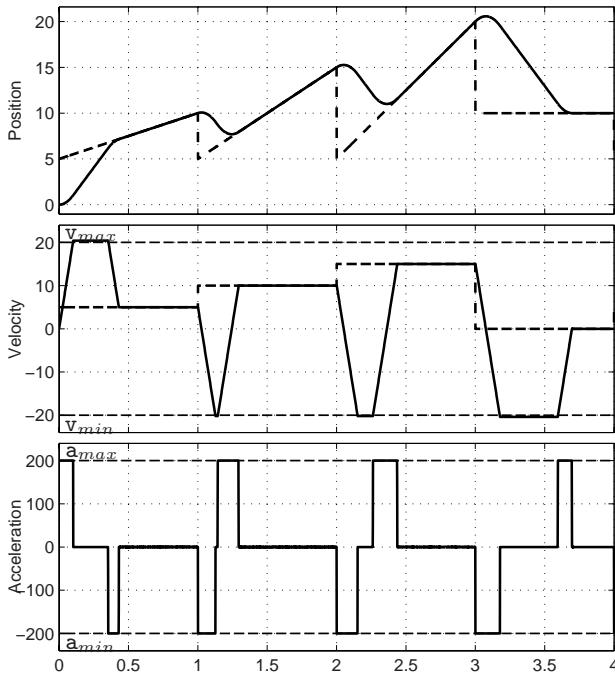
**Fig. 4.32.** Output (position, velocity and acceleration) of the second order nonlinear filter with a step reference signal.

$$C_2 : \begin{cases} z_k = \frac{1}{T_s} \left( \frac{e_k}{T_s} + \frac{\dot{e}_k}{2} \right), & \dot{z}_k = \frac{\dot{e}_k}{T_s} \\ m = \text{floor} \left( \frac{1 + \sqrt{1 + 8|z_k|}}{2} \right) \\ \sigma_k = \dot{z}_k + \frac{z_k}{m} + \frac{m-1}{2} \text{sign}(z_k) \\ u_k = -U \text{sat}(\sigma_k) \frac{1 + \text{sign}(\dot{q}_k) \text{sign}(\sigma_k) + v_{max} - T_s U}{2} \end{cases}$$

where  $\text{floor}(\cdot)$  is the function ‘integer part’,  $\text{sign}(\cdot)$  is the sign function, and  $\text{sat}(\cdot)$  is a saturation function defined by

$$\text{sat}(x) = \begin{cases} -1, & x < -1 \\ x, & -1 \leq x \leq 1 \\ +1, & x > 1. \end{cases}$$

Once the variable  $u_k$  (desired acceleration) has been computed, the velocity and the position are

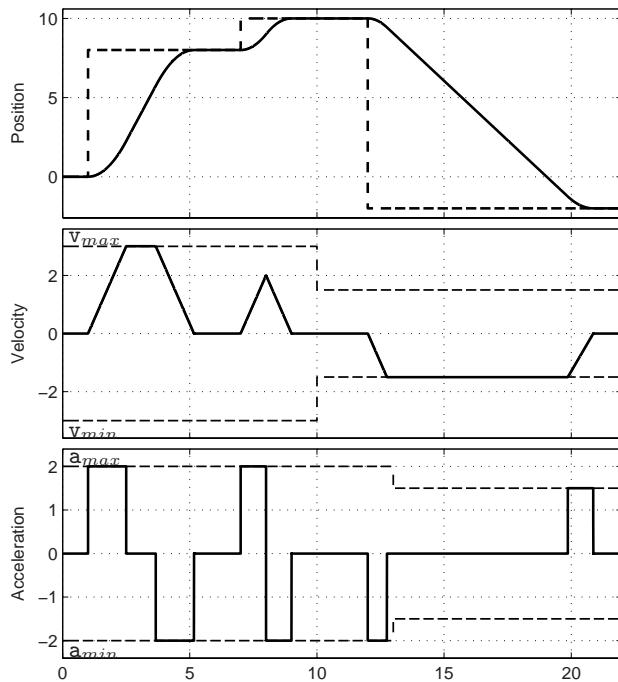


**Fig. 4.33.** Output (position, velocity and acceleration) of the second order nonlinear filter with a “sawtooth” reference signal.

$$\begin{cases} \dot{q}_k = \dot{q}_{k-1} + T_s u_{k-1} \\ q_k = q_{k-1} + \frac{T_s}{2}(\dot{q}_k + \dot{q}_{k-1}) \end{cases}$$

**Example 4.23** Fig. 4.32 shows position, velocity, and acceleration profiles of a trajectory computed by means of the second order nonlinear filter, when the reference signal  $r_k$  is a step from  $q_0 = 0$  to  $q_1 = 15$ , and the constraints are  $a_{max} = v_{max} = 10$ . Note that the trajectory has an ideal trapezoidal velocity profile.  $\square$

As in the case of the third order generator, this trajectory planner allows to track time-varying reference inputs, see Fig. 4.33, and to change at runtime the limit values of velocity and acceleration, as shown in Fig. 4.34.



**Fig. 4.34.** Trajectory computed by means of the second order nonlinear filter when the constraints on velocity and acceleration are changed at runtime.

## **Part II**

---

### **Elaboration and Analysis of Trajectories**

## Operations on Trajectories

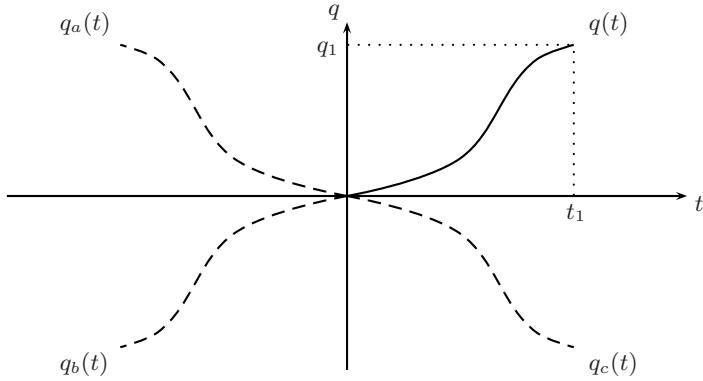
The trajectory profiles can be modified with the purpose of obtaining motions satisfying given constraints as, for example, the saturation limits of the actuation system. Typical modifications that can be applied are geometric scaling, translations in time or position, reflections. Another operation, particularly useful when it is necessary to impose on the motion profile proper constraints due to saturations of the actuation system is the *scaling in time*. There are two kinds of saturations: the *kinematic* and the *dynamic* saturation. In the former case, in order to execute the given trajectory it is necessary to define desired displacements characterized by velocities and/or accelerations achievable by the actuation system. In the latter case, typical of multi-axis machines like e.g. robotic manipulators, the actuation system may be not able to apply the torques needed to execute the given motion because of the coupled and variable dynamics which may characterize the machine.

Another typical operation, particularly useful for the synchronization of many axes of motion, is the *analytical composition* of the functions which define the trajectories. In *electronic cam* systems, the trajectory of the so-called *slave* (in general more than one) is defined with respect to the position of the *master*, and not as a function of the time. Therefore, the final motion law of each slave axis is given by the composition of the function defining the relation master-slave with the profile which describes the motion of the master, typically a “sawtooth” profile.

---

### 5.1 Geometric Modification of a Trajectory

In order to satisfy some constraints, not considered during the definition of the trajectory, some simple ‘geometric’ rules may be used. These rules are based on translations with respect to time  $t$  and/or to position variables  $q$ , on reflections about the coordinate axes, or on scaling operations. Figure 5.1



**Fig. 5.1.** Geometric construction of trajectories by reflection.

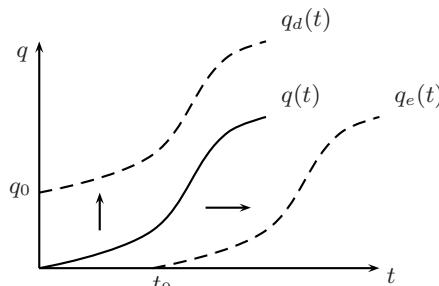
shows a generic trajectory  $q(t)$  from  $(t_0, q_0)$  to  $(t_1, q_1)$ , obtained by means of one of the methods reported in previous chapters (solid line). For the sake of simplicity it is assumed  $t_0 = 0$  and  $q_0 = 0$ . The other profiles in Fig. 5.1 are simply obtained by means of the following rules:

1.  $q_a(t) = q(-t), \quad t \in [-t_1, 0].$
2.  $q_b(t) = -q(-t), \quad t \in [-t_1, 0].$
3.  $q_c(t) = -q(t), \quad t \in [0, t_1].$

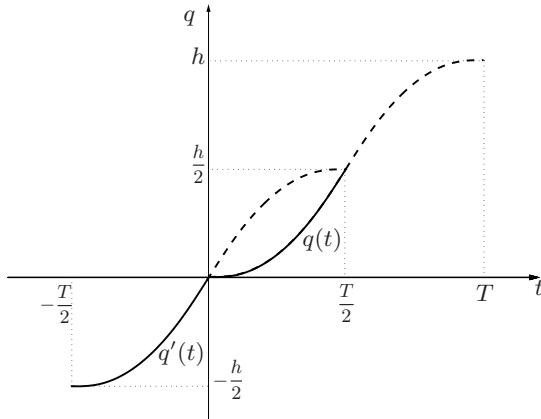
On the other hand, with reference to Fig. 5.2, the two trajectories denoted by a dashed line are obtained from  $q(t)$  by translation, i.e.

4.  $q_d(t) = q(t) + q_0, \quad t \in [0, t_1].$
5.  $q_e(t) = q(t - t_0), \quad t \in [t_0, t_0 + t_1].$

These operations can be exploited to modify the initial time instant or the initial position of a trajectory. For instance, when it is necessary to consider generic conditions on  $t_0$  and/or  $q_0$  for a trajectory whose expression has been computed with  $t_0 = 0$  and  $q_0 = 0$ , it is sufficient to consider  $t - t_0$  in lieu of  $t$  and/or add the initial position  $q_0$ .



**Fig. 5.2.** Geometric construction of trajectories by translation.



**Fig. 5.3.** Translation operations on a modified trapezoidal.

**Example 5.1** The modified trapezoidal trajectory described in Sec. 3.7 has been determined analytically for  $t \in [0, T/2]$ , being  $T$  the duration of the trajectory, and  $h$  the desired displacement:

$$q(t) = \begin{cases} \frac{h}{2+\pi} \left[ \frac{2t}{T} - \frac{1}{2\pi} \sin \left( \frac{4\pi t}{T} \right) \right], & 0 \leq t < \frac{T}{8} \\ \frac{h}{2+\pi} \left[ \frac{1}{4} - \frac{1}{2\pi} + \frac{2}{T} \left( t - \frac{T}{8} \right) + \frac{4\pi}{T^2} \left( t - \frac{T}{8} \right)^2 \right], & \frac{T}{8} \leq t < \frac{3}{8}T \\ \frac{h}{2+\pi} \left[ -\frac{\pi}{2} + 2(1+\pi) \frac{t}{T} - \frac{1}{2\pi} \sin \left( \frac{4\pi}{T} \left( t - \frac{T}{4} \right) \right) \right], & \frac{3}{8}T \leq t \leq \frac{T}{2}. \end{cases}$$

The second part of the trajectory can be obtained by exploiting its symmetry and the rules stated above. In particular, since the trajectory is symmetric with respect to the point  $(T/2, h/2)$ , it is convenient to translate  $q(t)$  so that the entire trajectory will be centered on  $(0, 0)$  (and accordingly  $t \in [-T/2, T/2]$  and  $q(t) \in [-h/2, h/2]$ ). The new trajectory obtained by translation, that is

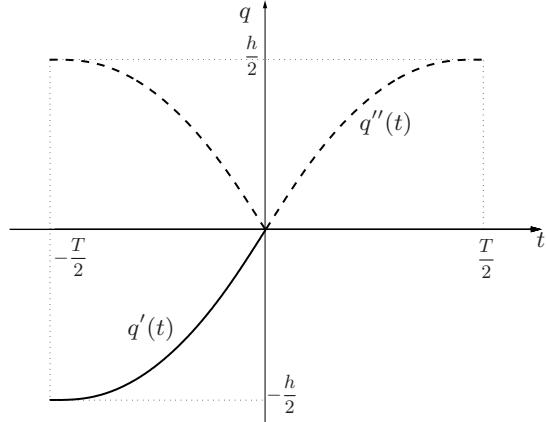
$$q'(t) = q(t - (-T/2)) - h/2, \quad (5.1)$$

is shown in Fig. 5.3. At this point, the second half of the trajectory can be easily deduced by reflecting  $q'(t)$  through the origin (by applying a reflection about the  $t$ -axis and then one about the  $q$ -axis), i.e.

$$q''(t) = -q'(-t)$$

as shown in Fig. 5.4. Now, it is necessary to translate again  $q''(t)$ , with a dual operation with respect to (5.1):

$$q'''(t) = q''(t - (T/2)) + h/2.$$



**Fig. 5.4.** Reflection operations on a segment of the modified trapezoidal profile.

Following this procedure, the expression of the modified trapezoidal trajectory for  $t \in [T/2, T]$  results

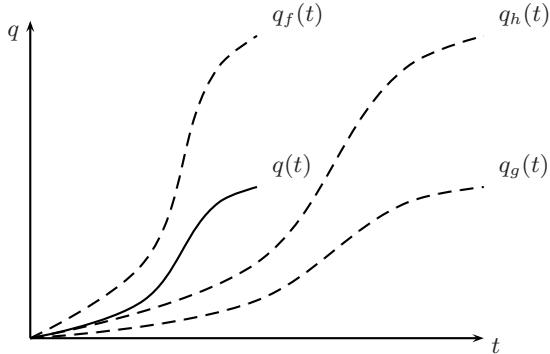
$$q(t) = \begin{cases} h + \frac{h}{2+\pi} \left[ \frac{\pi}{2} + 2(1+\pi) \frac{t-T}{T} - \frac{1}{2\pi} \sin \left( \frac{4\pi}{T} \left( t - \frac{3T}{4} \right) \right) \right], & \frac{1}{2}T \leq t < \frac{5}{8}T \\ h + \frac{h}{2+\pi} \left[ -\frac{1}{4} + \frac{1}{2\pi} + \frac{2}{T} \left( t - \frac{7T}{8} \right) - \frac{4\pi}{T^2} \left( t - \frac{7T}{8} \right)^2 \right], & \frac{5}{8}T \leq t < \frac{7}{8}T \\ h + \frac{h}{2+\pi} \left[ \frac{2(t-T)}{T} - \frac{1}{2\pi} \sin \left( \frac{4\pi}{T}(t-T) \right) \right], & \frac{7}{8}T \leq t \leq T. \end{cases}$$

□

Finally, with geometric scaling operations, it is possible to vary the displacement  $h$  or, with the techniques reported in the following section, to change the time scale, i.e. to modify the duration  $T$  of the trajectory. With reference to Fig. 5.5, notice that the profile  $q_f(t)$  is obtained from  $q(t)$  by increasing the displacement, the profile  $q_g(t)$  by increasing the duration of the trajectory, while both  $T$  and  $h$  have been modified in  $q_h(t)$ . For instance, if a normalized trajectory is considered, i.e.  $q(t) \in [0, 1]$  with  $t \in [0, 1]$ , the other trajectories are obtained as:

6.  $q_f(t) = h q(t).$
7.  $q_g(t) = q(t/T).$
8.  $q_h(t) = h q(t/T).$

**Example 5.2** In some applications, it may be computationally convenient to plan the trajectory as a composition of one or more elementary functions, for



**Fig. 5.5.** Geometric modification of trajectories by scaling operations.

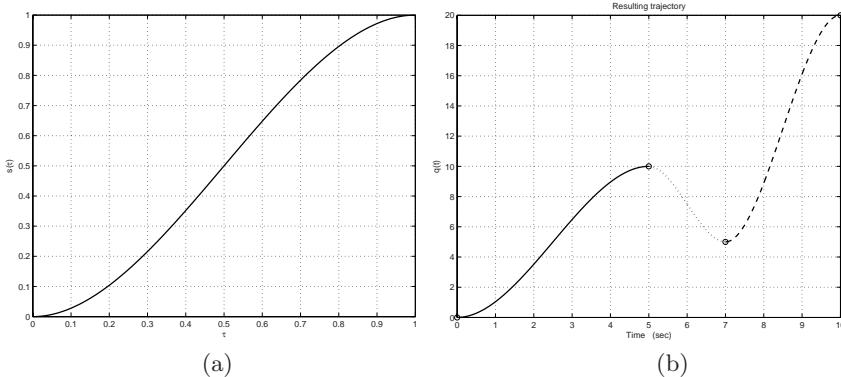
example cubic polynomial functions. For instance, consider the case in which a third degree polynomial trajectory, interpolating the points  $q_0 = 0$ ,  $q_1 = 10$ ,  $q_2 = 5$ ,  $q_3 = 20$  at  $t_0 = 0$ ,  $t_1 = 5$ ,  $t_2 = 7$ ,  $t_3 = 10$ , must be computed. For this purpose, a normalized expression of the cubic polynomial trajectory is adopted, see eq. (5.5) and Sec. 5.2.1. A unitary displacement in a unitary duration is expressed by

$$q_N(\tau) = 3\tau^2 - 2\tau^3, \quad \tau \in [0, 1].$$

The overall trajectory is obtained as composition of the three segments defined in terms of  $q_N(\tau)$  as

$$\begin{aligned} q_a(t) &= q_0 + h_1 \tilde{q}_N(t), \quad \text{with} \quad \tilde{q}_N(t) = q_N\left(\frac{t - t_0}{T_0}\right) \\ q_b(t) &= q_1 + h_2 \tilde{q}_N(t), \quad \text{with} \quad \tilde{q}_N(t) = q_N\left(\frac{t - t_1}{T_1}\right) \\ q_c(t) &= q_2 + h_3 \tilde{q}_N(t), \quad \text{with} \quad \tilde{q}_N(t) = q_N\left(\frac{t - t_2}{T_2}\right). \end{aligned}$$

where  $h_i = q_{i+1} - q_i$ ,  $T_i = t_{i+1} - t_i$ ,  $i = 0, 1, 2$ . The profiles of  $q_N(\tau)$  and of the overall trajectory  $q(t)$  are reported in Fig. 5.6.  $\square$



**Fig. 5.6.** Composition of 3-rd degree polynomial trajectories; normalized function  $q_N(\tau)$  (a), and overall trajectory obtained as union of the three segments (b).

## 5.2 Scaling in Time

In some cases, a trajectory must be modified in order to take into proper consideration the *saturation* limits of the actuation system, and to plan the desired motion so that these limits are not violated. Motion profiles requiring values of velocity, acceleration and torque outside the allowed ranges must be avoided, since these motions cannot be performed.

It is possible to distinguish two different types of saturations:

1. *Kinematic* saturation, when the velocities and/or the accelerations of the planned trajectory are larger than those achievable by the actuation system.
2. *Dynamic* saturation, that may be present when not feasible torques are requested to the actuation system; typically, this situation occurs in case of multi-axis mechanisms, such as industrial robots, because of the nonlinear and coupled dynamics.

If these effects are not considered a priori in the planning phase of the trajectory, it is necessary to verify them before the motion profile is applied to the system, taking proper actions in case the saturation limits are violated (e.g. by increasing the duration of the motion).

Given a generic trajectory

$$q = q(t)$$

it is possible to make it slower or faster, and more generally to modify the profiles of velocity, acceleration, etc., by considering a new time variable  $t'$  related to  $t$  by means of a strictly increasing function

$$t = \sigma(t').$$

In this way

$$\tilde{q}(t') = (q \circ \sigma)(t') = q(\sigma(t'))$$

and the velocity and the acceleration are

$$\dot{\tilde{q}}(t') = \frac{dq(\sigma)}{d\sigma} \frac{d\sigma(t')}{dt'} \quad (5.2a)$$

$$\ddot{\tilde{q}}(t') = \frac{dq(\sigma)}{d\sigma} \frac{d^2\sigma(t')}{dt'^2} + \frac{d^2q(\sigma)}{d\sigma^2} \left( \frac{d\sigma(t')}{dt'} \right)^2 \quad (5.2b)$$

⋮

Therefore, by properly defining the function  $\sigma$ , one can change the time-derivatives of  $\tilde{q}(t')$  according to the needs.

An example is represented by the modified cycloidal trajectory, Sec. 3.9, obtained by combining a standard cycloidal trajectory

$$q(t) = h \left[ \frac{t}{T} - \frac{1}{2\pi} \sin \left( \frac{2\pi t}{T} \right) \right]$$

with the function  $\sigma$  implicitly defined (the following equation provides  $\sigma^{-1}$ ) by

$$t' = t - k \frac{T}{2\pi} \sin \left( \frac{2\pi t}{T} \right)$$

with the purpose of reducing the maximum acceleration.

The combinations of trajectories  $q$  and functions  $\sigma$  are virtually infinite. A function of particular interest is the linear one

$$t = \sigma(t') = \lambda t' \implies t' = \frac{t}{\lambda}$$

which leads to

$$\begin{aligned} \dot{\tilde{q}}(t') &= \frac{dq(\sigma)}{d\sigma} \lambda \\ \ddot{\tilde{q}}(t') &= \frac{d^2q(\sigma)}{d\sigma^2} \lambda^2 \\ \tilde{q}^{(3)}(t') &= \frac{d^3q(\sigma)}{d\sigma^3} \lambda^3 \\ &\vdots \\ \tilde{q}^{(n)}(t') &= \frac{d^nq(\sigma)}{d\sigma^n} \lambda^n. \end{aligned} \quad (5.3)$$

For the sake of simplicity, one can rewrite these relations as

$$\begin{aligned}
\dot{\tilde{q}}(t') &= \lambda \dot{q}(t) \\
\ddot{\tilde{q}}(t') &= \lambda^2 \ddot{q}(t) \\
\ddot{q}^{(3)}(t') &= \lambda^3 q^{(3)}(t) \\
&\vdots \\
\ddot{q}^{(n)}(t') &= \lambda^n q^{(n)}(t).
\end{aligned} \tag{5.4}$$

Therefore by assuming the new time variable  $t'$ , obtained by scaling  $t$  with a constant parameter  $1/\lambda$ , all the derivatives of the reparameterized trajectory are scaled by powers of  $\lambda$ . By properly selecting  $\lambda$  it is possible to reduce, or increase, the speed, acceleration, jerk, ..., and make the trajectory compliant with given constraints. For example the choice

$$\lambda = \min \left\{ \frac{v_{max}}{|\dot{q}(t)|_{max}}, \sqrt{\frac{a_{max}}{|\ddot{q}(t)|_{max}}}, \sqrt[3]{\frac{j_{max}}{|q^{(3)}(t)|_{max}}} \right\}$$

guarantees that the maximum values of speed, acceleration and jerk are never exceeded.

### 5.2.1 Kinematic scaling

In order to define a trajectory satisfying given constraints on maximum velocity and acceleration, it is convenient to consider it in a *normalized form* and then perform geometric/time scaling operations. Any trajectory  $q(t)$ , defined for a displacement  $h = q_1 - q_0$  and a duration  $T = t_1 - t_0$ , can be written in terms of the normalized form  $q_N(\tau)$  with

$$0 \leq q_N(\tau) \leq 1, \quad 0 \leq \tau \leq 1.$$

As a matter of fact, it results

$$q(t) = q_0 + (q_1 - q_0) \tilde{q}_N(t) = q_0 + h \tilde{q}_N(t) \tag{5.5}$$

where

$$\tilde{q}_N(t) = q_N(\tau), \quad \text{with } \tau = \frac{t - t_0}{t_1 - t_0} = \frac{t - t_0}{T}.$$

Notice that in this case the time variable  $t$  is obtained by scaling  $\tau$  by a factor  $\lambda = T$ , and by translating it of  $t_0$ . From (5.5) it follows

$$\begin{aligned}
q^{(1)}(t) &= \frac{h}{T} q_N^{(1)}(\tau) \\
q^{(2)}(t) &= \frac{h}{T^2} q_N^{(2)}(\tau) \\
q^{(3)}(t) &= \frac{h}{T^3} q_N^{(3)}(\tau) \\
&\vdots \\
q^{(n)}(t) &= \frac{h}{T^n} q_N^{(n)}(\tau)
\end{aligned} \tag{5.6}$$

where

$$q_N^{(1)}(\tau) = \frac{d q_N(\tau)}{d\tau}, \quad q_N^{(2)}(\tau) = \frac{d^2 q_N(\tau)}{d\tau^2}, \quad \dots$$

It is simple to verify that the maximum values of velocity, acceleration, jerk, and so on, are obtained in correspondence of the maximum values of the functions  $q_N^{(1)}$ ,  $q_N^{(2)}$ ,  $\dots$ . Then, it is easy to compute both these values and the corresponding time instants  $\tau$  from the given parameterization  $q_N(\tau)$ .

Note that, by properly changing the time length  $T$  of the trajectory, from (5.6) it is easy to obtain motion profiles with maximum velocity/acceleration values equal to the saturation limits, i.e. *optimal* trajectories in the sense that their time duration is minimized. Moreover, it is also simple to coordinate several motion axes.

The application of these considerations to some of the motion profiles discussed in the previous chapters is now described.

### Polynomial trajectories with degree 3, 5, and 7

The parameterization of a cubic polynomial (2.1) in a normalized form is

$$q_N(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3.$$

With the boundary conditions  $q_{N_0}^{(1)} = 0$ ,  $q_{N_1}^{(1)} = 0$ , and obviously  $q_N(0) = 0$  and  $q_N(1) = 1$ , the following values are obtained

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 3, \quad a_3 = -2.$$

Therefore

$$\begin{aligned} q_N(\tau) &= 3\tau^2 - 2\tau^3 \\ q_N^{(1)}(\tau) &= 6\tau - 6\tau^2 \\ q_N^{(2)}(\tau) &= 6 - 12\tau \\ q_N^{(3)}(\tau) &= -12. \end{aligned} \tag{5.7}$$

The maximum velocity and acceleration values are

$$\begin{aligned} q_{N\max}^{(1)} &= q_N^{(1)}(0.5) = \frac{3}{2} \implies \dot{q}_{max} = \frac{3h}{2T} \\ q_{N\max}^{(2)} &= q_N^{(2)}(0) = 6 \implies \ddot{q}_{max} = \frac{6h}{T^2}. \end{aligned} \tag{5.8}$$

For polynomial functions of degree 5 the normalized form is

$$q_N(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4 + a_5\tau^5.$$

With the boundary conditions  $q_{N_0}^{(1)} = 0$ ,  $q_{N_1}^{(1)} = 0$ ,  $q_{N_0}^{(2)} = 0$ ,  $q_{N_1}^{(2)} = 0$ , the following parameters are obtained<sup>1</sup>

---

<sup>1</sup> The trajectory obtained with these conditions is also known in the literature as "3-4-5" polynomial.

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 0, \quad a_3 = 10, \quad a_4 = -15, \quad a_5 = 6.$$

Therefore

$$\begin{aligned} q_N(\tau) &= 10\tau^3 - 15\tau^4 + 6\tau^5 \\ q_N^{(1)}(\tau) &= 30\tau^2 - 60\tau^3 + 30\tau^4 \\ q_N^{(2)}(\tau) &= 60\tau - 180\tau^2 + 120\tau^3 \\ q_N^{(3)}(\tau) &= 60 - 360\tau + 360\tau^2 \end{aligned} \tag{5.9}$$

and

$$\begin{aligned} q_{N\max}^{(1)} &= q_N^{(1)}(0.5) = \frac{15}{8} \quad \Rightarrow \quad \dot{q}_{\max} = \frac{15h}{8T} \\ q_{N\max}^{(2)} &= q_N^{(2)}(0.2123) = \frac{10\sqrt{3}}{3} \quad \Rightarrow \quad \ddot{q}_{\max} = \frac{10\sqrt{3}h}{3T^2} \\ q_{N\max}^{(3)} &= q_N^{(3)}(0) = 60 \quad \Rightarrow \quad q_{\max}^{(3)} = 60 \frac{h}{T^3}. \end{aligned} \tag{5.10}$$

The normalized polynomial of degree 7 is

$$q_N(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4 + a_5\tau^5 + a_6\tau^6 + a_7\tau^7.$$

With the boundary conditions  $q_{N_0}^{(1)} = q_{N_1}^{(1)} = 0$ ,  $q_{N_0}^{(2)} = q_{N_1}^{(2)} = 0$ ,  $q_{N_0}^{(3)} = q_{N_1}^{(3)} = 0$  one obtains<sup>2</sup>

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 0, \quad a_3 = 0, \quad a_4 = 35, \quad a_5 = -84, \quad a_6 = 70, \quad a_7 = -20.$$

Then

$$\begin{aligned} q_N(\tau) &= 35\tau^4 - 84\tau^5 + 70\tau^6 - 20\tau^7 \\ q_N^{(1)}(\tau) &= 140\tau^3 - 420\tau^4 + 420\tau^5 - 140\tau^6 \\ q_N^{(2)}(\tau) &= 420\tau^2 - 1680\tau^3 + 2100\tau^4 - 840\tau^5 \\ q_N^{(3)}(\tau) &= 840\tau - 5040\tau^2 + 8400\tau^3 - 4200\tau^4 \end{aligned} \tag{5.11}$$

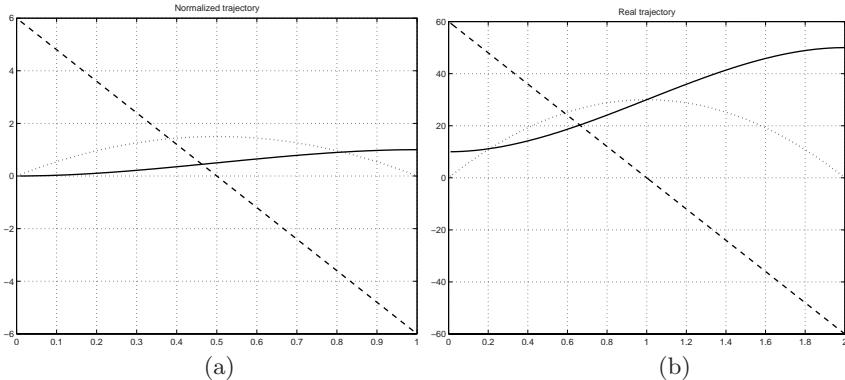
and

$$\begin{aligned} q_{N\max}^{(1)} &= q_N^{(1)}(0.5) = \frac{35}{16} \quad \Rightarrow \quad \dot{q}_{\max} = \frac{35h}{16T} \\ q_{N\max}^{(2)} &= q_N^{(2)}(0.2764) = 7.5132 \quad \Rightarrow \quad \ddot{q}_{\max} = 7.5132 \frac{h}{T^2} \\ |q_N^{(3)}|_{\max} &= |q_N^{(3)}(0.5)| = 52.5 \quad \Rightarrow \quad |q^{(3)}|_{\max} = 52.5 \frac{h}{T^3}. \end{aligned} \tag{5.12}$$

Notice that in this case the maximum magnitude of the jerk is obtained for negative values, i.e. -52.5.

---

<sup>2</sup> The trajectory obtained with these conditions is also known in the literature as "4-5-6-7" polynomial.



**Fig. 5.7.** Profiles of acceleration (dashed), velocity (dotted) and position (solid) a polynomial trajectory of degree 3: normalized form (a) and with conditions  $q_0 = 10$ ,  $q_1 = 50$ ,  $t_0 = 0$ ,  $t_1 = 2$  (b).

**Example 5.3** A polynomial function of degree 3 is used to define an optimal (minimum time) trajectory from  $q_0 = 10$  to  $q_1 = 50$  ( $h = 40$ ). The maximum velocity that the actuation system can generate is  $\dot{q}_{max} = 30$ , and the maximum acceleration is  $\ddot{q}_{max} = 80$ . From (5.7) and (5.8) one obtains

$$\dot{q}_{max} = \frac{3h}{2T} = 30 \quad \Rightarrow \quad T = \frac{3h}{2 \cdot 30} = 2$$

$$\ddot{q}_{max} = \frac{6h}{T^2} = 80 \quad \Rightarrow \quad T = \sqrt{\frac{6h}{80}} = 1.732.$$

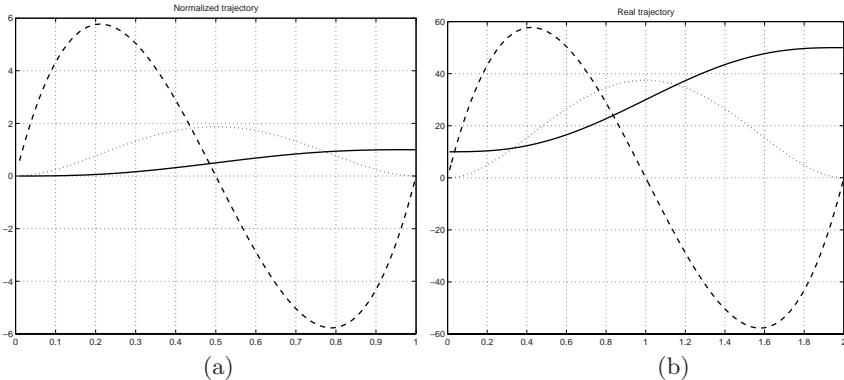
In order to meet the limit values of the actuator, it is then necessary to set  $T = 2$ , since the constraint on the maximum velocity is more restrictive than the constraint on the maximum acceleration. The motion profiles of this example are shown in Fig. 5.7. Note that the maximum velocity is achieved at  $t = 1$ . If a polynomial trajectory of degree 5 is adopted, and the actuator limits are  $\dot{q}_{max} = 37.5$  and  $\ddot{q}_{max} = 60$ , from (5.9), (5.10) one obtains

$$\dot{q}_{max} = \frac{15h}{8T} = 37.5 \quad \Rightarrow \quad T = \frac{15 \cdot 40}{8 \cdot 37.5} = 2$$

$$\ddot{q}_{max} = \frac{10\sqrt{3}h}{3T^2} = 60 \quad \Rightarrow \quad T = \sqrt{\frac{10\sqrt{3} \cdot 40}{3 \cdot 60}} = 1.962.$$

It is necessary to set  $T = 2$  since also in this case the constraint on the maximum velocity is more restrictive than the constraint on the maximum acceleration. The motion profiles are reported in Fig. 5.8.

Finally, if a polynomial of degree 7 is considered, with limits on the actuation system  $\dot{q}_{max} = 45$  and  $\ddot{q}_{max} = 50$ , from (5.11), (5.12) one obtains

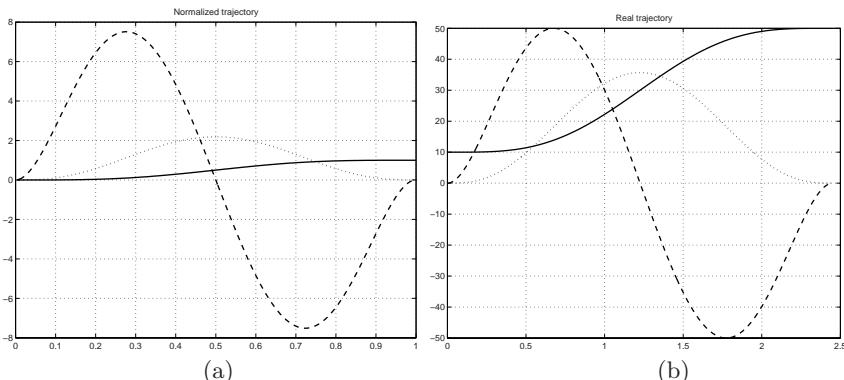


**Fig. 5.8.** Profiles of acceleration (dashed), velocity (dotted) and position (solid) of a polynomial of degree 5: normalized form (a) and with boundary conditions  $q_0 = 10$ ,  $q_1 = 50$ ,  $t_0 = 0$ ,  $t_1 = 2$  (b).

$$\begin{aligned}\dot{q}_{max} &= \frac{35h}{16T} = 45 & \Rightarrow \quad T &= \frac{35 \cdot 40}{16 \cdot 45} = 1.944 \\ \ddot{q}_{max} &= 7.5132 \frac{h}{T^2} = 50 & \Rightarrow \quad T &= \sqrt{\frac{7.5132 \cdot 40}{50}} = 2.4516.\end{aligned}$$

In this case, in order to satisfy the actuator limits it is necessary to set  $T = 2.4516$  since the constraint on the maximum acceleration is more restrictive than the constraint on the maximum velocity. The profiles of this trajectory are shown in Fig. 5.9.

Notice that if the same actuator (i.e. characterized by  $\dot{q}_{max} = 30$ ,  $\ddot{q}_{max} = 80$ ) is used for the three trajectories, the durations reported in Table 5.1 are



**Fig. 5.9.** Profiles of acceleration (dashed), velocity (dotted) and position (solid) of a polynomial of degree 7: normalized form (a) and with boundary conditions  $q_0 = 10$ ,  $q_1 = 50$ ,  $t_0 = 0$ ,  $t_1 = 2.4516$  (b).

	$T_{vel}$	$T_{acc}$
Polynomial 3	2.0000	1.7321
Polynomial 5	2.5000	1.6990
Polynomial 7	2.9167	1.9382

**Table 5.1.** Minimum duration for some polynomial trajectories with  $\dot{q}_{max} = 30$  and  $\ddot{q}_{max} = 80$ .

obtained. From this Table, it is possible to see that in any case the most restrictive constraint is due to the velocity limit and that, if the degree of the polynomial function increases, the duration  $T$  of the trajectory increases as well (the maximum velocity is fixed). This result has a general validity: given a maximum value for the velocity (or for the acceleration), the ‘smoother’ the motion profiles are, the longer the relative durations are.  $\square$

### Cycloidal motion

The cycloidal profile (2.22) has a normalized parameterization given by

$$q_N(\tau) = \tau - \frac{1}{2\pi} \sin 2\pi\tau$$

from which

$$\begin{aligned} q_N^{(1)}(\tau) &= 1 - \cos 2\pi\tau \\ q_N^{(2)}(\tau) &= 2\pi \sin 2\pi\tau \\ q_N^{(3)}(\tau) &= 4\pi^2 \cos 2\pi\tau \end{aligned}$$

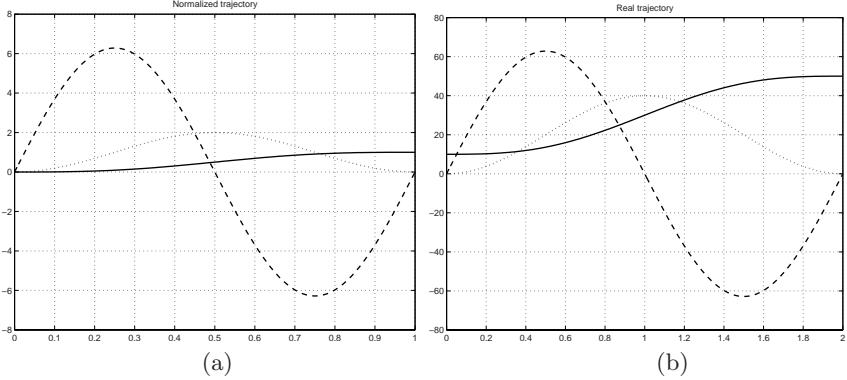
and

$$\begin{aligned} q_{N \max}^{(1)} &= q_N^{(1)}(0.5) = 2 \quad \implies \quad \dot{q}_{max} = 2\frac{h}{T} \\ q_{N \max}^{(2)} &= q_N^{(2)}(0.25) = 2\pi \quad \implies \quad \ddot{q}_{max} = 2\pi\frac{h}{T^2} \\ q_{N \max}^{(3)} &= q_N^{(3)}(0) = 4\pi^2 \quad \implies \quad q_{max}^{(3)} = 4\pi^2\frac{h}{T^3}. \end{aligned}$$

### Harmonic motion

The harmonic profile (2.20) has the following normalized parameterization

$$q_N(\tau) = \frac{1}{2}(1 - \cos \pi\tau)$$



**Fig. 5.10.** Profiles of acceleration (dashed), velocity (dotted) and position (solid) of a cycloidal motion: normalized form (a) and with boundary conditions  $q_0 = 10$ ,  $q_1 = 50$ ,  $t_0 = 0$ ,  $t_1 = 2$ , (b).

from which

$$q_N^{(1)}(\tau) = \frac{\pi}{2} \sin \pi \tau$$

$$q_N^{(2)}(\tau) = \frac{\pi^2}{2} \cos \pi \tau$$

$$q_N^{(3)}(\tau) = -\frac{\pi^3}{2} \sin \pi \tau$$

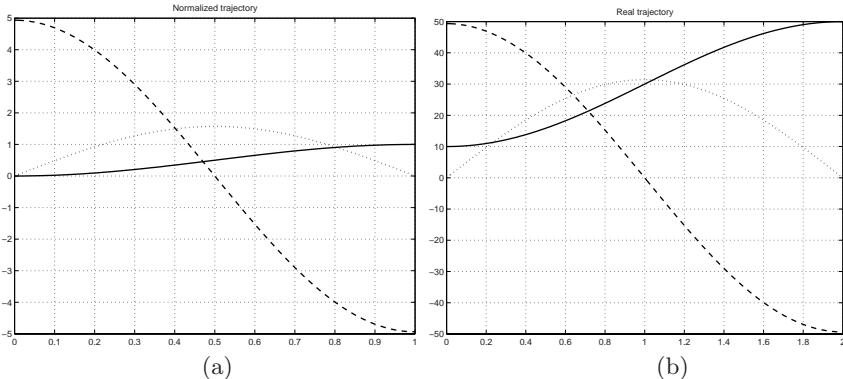
and

$$\begin{aligned} q_{N \max}^{(1)} &= q_N^{(1)}(0.5) = \frac{\pi}{2} & \Rightarrow & \dot{q}_{\max} = \frac{\pi h}{2T} \\ q_{N \max}^{(2)} &= q_N^{(2)}(0) = \frac{\pi^2}{2} & \Rightarrow & \ddot{q}_{\max} = \frac{\pi^2 h}{2T^2} \\ |q_N^{(3)}|_{\max} &= |q_N^{(3)}(0.5)| = \frac{\pi^3}{2} & \Rightarrow & |q_{\max}| = \frac{\pi^3 h}{2T^3}. \end{aligned}$$

**Example 5.4** In Fig. 5.10 and 5.11 the cycloidal and harmonic trajectories are reported. The boundary conditions are  $q_0 = 10$ ,  $q_1 = 50$ ,  $t_0 = 0$ ,  $t_1 = 2$ .  $\square$

### 5.2.2 Dynamic Scaling

When dynamic couplings or nonlinear effects are present in an automatic machine, it is possible that, during the execution of a trajectory, the torques requested to the actuation system exceed the physical limits. In particular,



**Fig. 5.11.** Profiles of acceleration (dashed), velocity (dotted) and position (solid) of an harmonic motion: normalized form (a) and with boundary conditions  $q_0 = 10$ ,  $q_1 = 50$ ,  $t_0 = 0$ ,  $t_1 = 2$ , (b).

this may happen when the inertia at the motor is not constant, but is a non-linear function of the load position. In order to avoid these problems, once the trajectory has been defined it is possible to apply a suitable scaling procedure which allows to obtain torques within the given limits of the actuation system [51]. This scaling procedure does not imply the re-computation of the whole trajectory.

An important example of mechanical system with nonlinear dynamics and coupling effects is a robotic manipulator. The dynamic model of an industrial robot with  $n$  degrees of freedom (actuated joints) with positions, velocities and accelerations described by the vectors  $\mathbf{q}(t)$ ,  $\dot{\mathbf{q}}(t)$ ,  $\ddot{\mathbf{q}}(t)$  respectively, has the following formulation

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (5.13)$$

where  $\mathbf{M}(\mathbf{q})$  is the  $(n \times n)$  inertia matrix of the robot (symmetric and positive definite),  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  is a  $(n \times n)$  matrix describing the Coriolis and centrifugal effects,  $\mathbf{g}(\mathbf{q})$  is the  $(n \times 1)$  vector of the gravitational forces acting on the system and  $\boldsymbol{\tau}$  is the  $(n \times 1)$  vector of the joint torques applied by the actuation system. For a more detailed discussion about robot manipulators and their dynamic models a wide literature is available, see for example [12, 52, 53].

Let us consider the  $i$ -th row of (5.13). For each actuator the following equation holds

$$\mathbf{m}_i^T(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \frac{1}{2}\dot{\mathbf{q}}^T(t)\mathbf{L}_i(\mathbf{q}(t))\dot{\mathbf{q}}(t) + g_i(\mathbf{q}(t)) = \tau_i(t), \quad i = 1, \dots, n \quad (5.14)$$

where  $\mathbf{m}_i(\mathbf{q})$  is the  $i$ -th column of matrix  $\mathbf{M}(\mathbf{q})$ , the term  $1/2\dot{\mathbf{q}}^T\mathbf{L}_i(\mathbf{q})\dot{\mathbf{q}}$  a proper formulation of  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ ,  $g_i(\mathbf{q})$  the gravitational force and  $\tau_i$  the torque

applied to the joint by the  $i$ -th actuator. If a trajectory  $\mathbf{q}(t)$ ,  $t \in [0, T]$  has been defined for the joints, the corresponding torques  $\tau_i(t)$  can be computed by means of (5.14), that can be rewritten as

$$\tau_i(t) = \tau_{s,i}(t) + \tau_{p,i}(t) \quad (5.15)$$

where

$$\begin{aligned} \tau_{s,i}(t) &= \mathbf{m}_i^T(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \frac{1}{2}\dot{\mathbf{q}}^T(t)\mathbf{L}_i(\mathbf{q}(t))\dot{\mathbf{q}}(t) \\ \tau_{p,i}(t) &= g_i(\mathbf{q}(t)). \end{aligned} \quad (5.16)$$

Notice that  $\tau_{s,i}(t)$  depends on positions, velocities and accelerations, while  $\tau_{p,i}(t)$  on positions only.

Let us consider a new trajectory  $\tilde{\mathbf{q}}(t')$ ,  $t' \in [0, T']$ , obtained by re-parameterizing  $\mathbf{q}(t)$  by means of a strictly increasing scalar function  $t = \sigma(t')$  such that  $0 = \sigma(0)$  and  $T = \sigma(T')$ . The torques necessary to execute this new trajectory can be computed as

$$\tilde{\tau}_i(t') = \mathbf{m}_i^T(\tilde{\mathbf{q}}(t'))\ddot{\tilde{\mathbf{q}}}(t') + \frac{1}{2}\dot{\tilde{\mathbf{q}}}^T(t')\mathbf{L}_i(\tilde{\mathbf{q}}(t'))\dot{\tilde{\mathbf{q}}}(t') + g_i(\tilde{\mathbf{q}}(t')). \quad (5.17)$$

Since

$$\tilde{\mathbf{q}}(t') = (\mathbf{q} \circ \sigma)(t')$$

the relationship between the time-derivatives of  $\mathbf{q}(t)$  and those of  $\tilde{\mathbf{q}}(t')$  (in this case with respect to the new time variable  $t'$ ) are

$$\begin{aligned} \dot{\tilde{\mathbf{q}}}(t') &= \dot{\mathbf{q}}(t)\dot{\sigma} \\ \ddot{\tilde{\mathbf{q}}}(t') &= \ddot{\mathbf{q}}(t)\dot{\sigma}^2 + \dot{\mathbf{q}}(t)\ddot{\sigma}. \end{aligned}$$

where  $\dot{\sigma} = d\sigma/dt'$  and  $\ddot{\sigma} = d^2\sigma/dt'^2$ .

If these equations are substituted in (5.17), one obtains

$$\tilde{\tau}_i(t') = \left[ \mathbf{m}_i^T(\mathbf{q}(t))\dot{\mathbf{q}}(t) \right] \ddot{\sigma} + \left[ \mathbf{m}_i^T(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \frac{1}{2}\dot{\mathbf{q}}^T(t)\mathbf{L}_i(\mathbf{q}(t))\dot{\mathbf{q}}(t) \right] \dot{\sigma}^2 + g_i(\mathbf{q}(t))$$

where  $t = \sigma(t')$ . Notice that the term  $g_i(\tilde{\mathbf{q}}(t'))$  depends on the joints position only, and thus its contribution is not influenced by the time scaling. Therefore, it is convenient to consider only the term

$$\tilde{\tau}_{s,i}(t') = \left[ \mathbf{m}_i^T(\mathbf{q}(t))\dot{\mathbf{q}}(t) \right] \ddot{\sigma} + \left[ \mathbf{m}_i^T(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \frac{1}{2}\dot{\mathbf{q}}^T(t)\mathbf{L}_i(\mathbf{q}(t))\dot{\mathbf{q}}(t) \right] \dot{\sigma}^2$$

which can be rewritten as

$$\tilde{\tau}_{s,i}(t') = \left[ \mathbf{m}_i^T(\mathbf{q}(t))\dot{\mathbf{q}}(t) \right] \ddot{\sigma} + \tau_{s,i}(t)\dot{\sigma}^2. \quad (5.18)$$

In order to understand the effects of the time scaling on  $\tau_{s,i}$  it is necessary to specify the scaling function  $\sigma$ . The simplest choice consists in the linear function

$$t = \sigma(t') = \lambda t' \quad (5.19)$$

which leads to

$$\dot{\sigma}(t') = \lambda, \quad \ddot{\sigma}(t') = 0.$$

By substituting these values in (5.18) one obtains

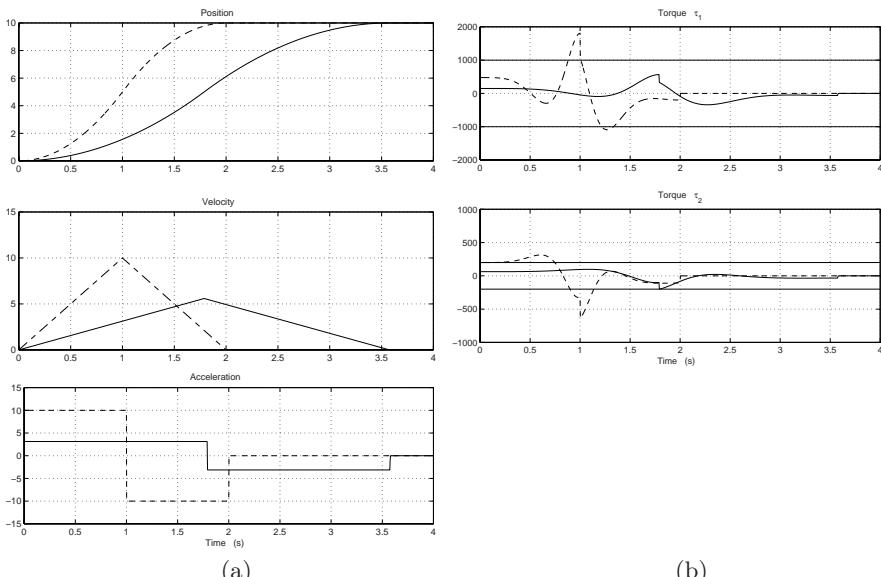
$$\tilde{\tau}_{s,i}(t') = \lambda^2 \tau_{s,i}(t), \quad i = 1, \dots, n \quad (5.20)$$

or, recalling the definition of  $\tau_{s,i}$ ,

$$\tilde{\tau}_i(t') - g_i(\tilde{\mathbf{q}}(t')) = \lambda^2 [\tau_i(t) - g_i(\mathbf{q}(t))], \quad i = 1, \dots, n.$$

Therefore, a linear time scaling by a constant  $1/\lambda$  (the new time value  $t'$  can be obtained by inverting (5.19) and therefore  $t' = t/\lambda$ ) produces a scaling of the magnitude of the torques (depending on velocity/acceleration) by a coefficient  $\lambda^2$ . If  $\lambda < 1$ , the new trajectory  $\tilde{\mathbf{q}}(t')$  has a duration  $T'(= T/\lambda)$  larger than  $\mathbf{q}(t)$ , and accordingly the torques  $\tilde{\tau}_{s,i}(t')(= \lambda^2 \tau_{s,i}(t))$  necessary to perform the motion are smaller than those required to execute the original trajectory.

**Example 5.5** Let us consider a two degrees of freedom planar manipulator. The limit values for the torques are  $\bar{\tau}_1 = 1000$ ,  $\bar{\tau}_2 = 200$ . If the trapezoidal trajectories shown in Fig. 5.12(a) are specified for both the joints, with duration  $T = 2$ , one obtains the torques profiles shown in dashed lines



**Fig. 5.12.** Position, velocity, acceleration (a) and torque (b) profiles before (solid) and after (dashed) a dynamic scaling operation (Example 5.5).

in Fig. 5.12(b). The maximum values needed for the execution of this motion are  $\tau_{1,max} = 1805.9$  and  $|\tau_2|_{max} = 639.8$ . In order to obtain a trajectory that can be physically executed, a dynamic scaling is performed. The value

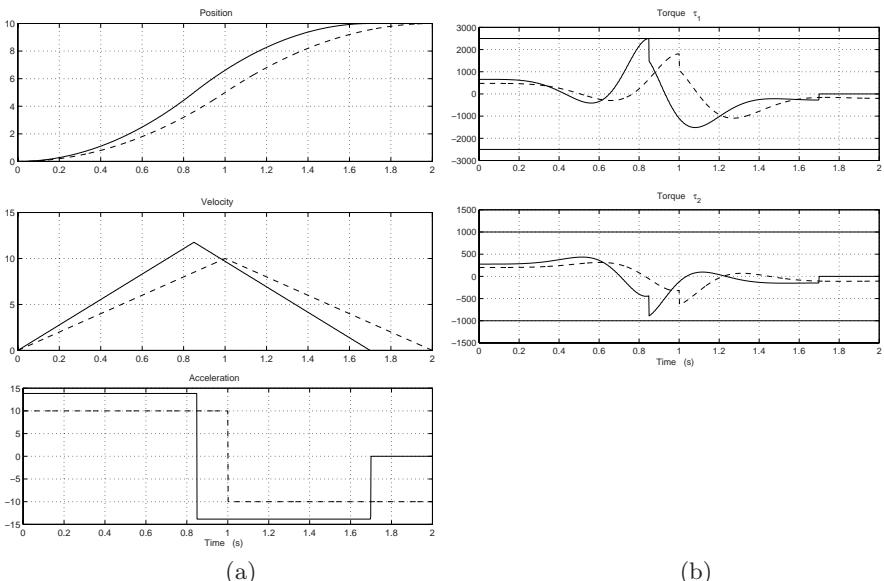
$$\lambda^2 = \min \left\{ \frac{\bar{\tau}_1}{\tau_{1,max}}, \frac{\bar{\tau}_2}{\tau_{2,max}} \right\} = \min \left\{ \frac{1000}{1805.9}, \frac{200}{639.8} \right\} = 0.3126$$

is used to scale the time:

$$t' = \frac{t}{\lambda} = \frac{t}{0.5591} \quad \longrightarrow \quad T' = \frac{T}{\lambda} = \frac{2}{0.5591} = 3.5771.$$

The new torques ( $\tau(t') = \lambda^2 \tau(t)$ ) are both feasible and  $\tau_2$  reaches the limit value at  $t' = 1.7885$ .  $\square$

**Example 5.6** Let us consider the planar robot of the previous example, with the torque limits  $\bar{\tau}_1 = 2500$ ,  $\bar{\tau}_2 = 1000$ . The maximum values required by the trajectory considered in the previous example are  $\tau_{1,max} = 1805.9$  and  $\tau_{2,max} = 639.8$ , in this case lower than the allowed limits. In this case, by using the dynamic scaling, it is possible to compute the optimal trajectory (minimum duration) by increasing the requested torques. As a matter of fact, one obtains



**Fig. 5.13.** Position, velocity, acceleration (a) and torque (b) profiles before (dashed) and after (solid) a dynamic scaling operation (Example 5.6).

$$\lambda^2 = \min \left\{ \frac{\bar{\tau}_1}{\tau_{1,max}}, \frac{\bar{\tau}_2}{\tau_{2,max}} \right\} = \min \left\{ \frac{2500}{1805.9}, \frac{1000}{639.8} \right\} = 1.3844$$

and then

$$t' = \frac{t}{\lambda} = \frac{t}{1.1766} \quad \longrightarrow \quad T' = \frac{T}{1.1766} = \frac{2}{1.1766} = 1.6999.$$

In this case, a shorter trajectory is achieved ( $T' < T$ ). Also for this motion one of the two torques, in this case the first one, saturates in a point. The profiles of this motion are reported in Fig. 5.13(a) and the corresponding torques in Fig. 5.13(b).  $\square$

If the whole trajectory is scaled in order to avoid to exceed the limits in a single point, an unnecessarily slow motion is likely obtained. Therefore, it is possible to apply a *variable scaling*, to be used only in correspondence of the segments where the torques overcome the given limit values.

A last consideration is that it is possible to show that the *minimum time motion* along a given path saturates the torque, the acceleration or the velocity of one of the actuators in at least a point of each segment.

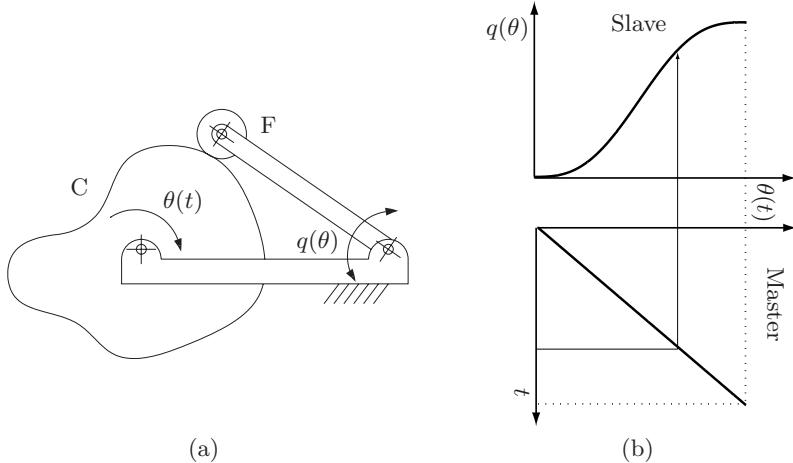
### 5.3 Synchronization of Trajectories

The composition of functions is an useful method not only for the scaling in time of trajectories, but also for their synchronization. As a matter of fact, it is possible to define the position profile of an actuator as a function of a generic variable  $\theta$  instead of the time  $t$ . For example, in master-slave applications the motion profiles of the slave axis are defined with respect to the configuration of the master, which can be either real, that is an axis of motion of the machine, or virtual, that is a simple signal in the controller, implementing the so-called *electronic cams*.

This idea derives from the *mechanical cams*, see Fig. 5.14, used in automatic machines with the goal of transferring, coordinating and changing the type of motion from a master device to one or more slave systems. With reference to Fig. 5.14 the body C, the *cam*, is supposed to rotate at a constant angular velocity, and therefore its angular position  $\theta$  is a linear function of time, while body the F, the *follower*, has an alternative motion  $q(\theta)$  defined by the profile of the cam<sup>3</sup>. In the same manner, an electronic cam is defined by providing the function  $q(\theta)$  which describes the position of the slave with respect to  $\theta$ . The trajectory of the axis of motion is therefore  $\tilde{q}(t) = (q \circ \theta)(t)$ , with the velocity and the acceleration given by

---

<sup>3</sup> The design of mechanical cams have been extensively and carefully investigated, and on this argument a wide literature is available in the mechanical field, [4, 5, 6, 7, 8, 9].



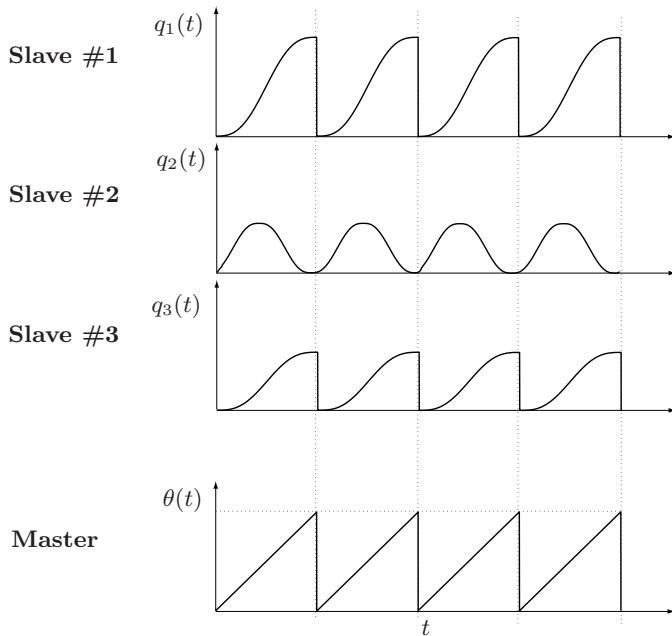
**Fig. 5.14.** (a) Mechanical cam (C) with a follower (F); (b) function defining an electronic cam.

$$\begin{aligned}\dot{q}(t) &= \frac{dq}{d\theta} \dot{\theta}(t) \\ \ddot{q}(t) &= \frac{d^2q}{d\theta^2} \dot{\theta}^2(t) + \frac{dq}{d\theta} \ddot{\theta}(t)\end{aligned}$$

where  $\dot{\theta}(t)$  and  $\ddot{\theta}(t)$  are the velocity and the acceleration of the master axis respectively. Therefore by acting on the velocity (or acceleration) of the master (whose motion law is usually a simple constant velocity trajectory, i.e.  $\theta(t) = v_c t$ , between  $\theta_0 = 0^\circ$  and  $\theta_1 = 360^\circ$ ) it is possible to change the velocities, accelerations of all the slaves “connected” to the master. In particular, when several slaves are present, like in Fig. 5.15 where  $q_k(\theta)$  denotes the relationship between the master and the  $k$ -th slave ( $k = 1, \dots, n$ ), the constant velocity  $v_c$  of the master must be selected in such a way that all the trajectories  $\tilde{q}_k(t)$  satisfy the constraints on their velocity, acceleration and jerk (which may be different for each actuator). For this purpose, it is sufficient to assume

$$v_c = \min \left\{ \frac{v_{max1}}{|\dot{q}_1(\theta)|_{max}}, \dots, \frac{v_{maxn}}{|\dot{q}_n(\theta)|_{max}}, \sqrt{\frac{a_{max1}}{|\ddot{q}_1(\theta)|_{max}}}, \dots, \sqrt{\frac{a_{maxn}}{|\ddot{q}_n(\theta)|_{max}}}, \right. \\ \left. \sqrt[3]{\frac{j_{max1}}{|q_1^{(3)}(\theta)|_{max}}}, \dots, \sqrt[3]{\frac{j_{maxn}}{|q_n^{(3)}(\theta)|_{max}}}} \right\}$$

being the relationships between the derivatives of  $q_k(\theta)$  and those of  $\tilde{q}_k(t)$  similar to (5.4):



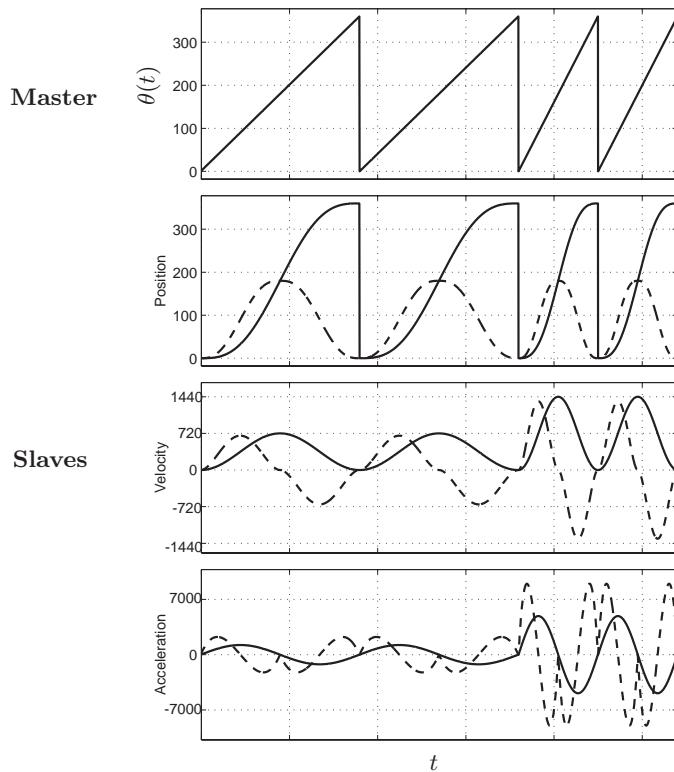
**Fig. 5.15.** Position profiles of master and slaves as a function of time in a multi-slave system.

$$\begin{aligned}
 \dot{\tilde{q}}(t) &= v_c \frac{dq(\theta)}{d\theta} \\
 \ddot{\tilde{q}}(t) &= v_c^2 \frac{d^2q(\theta)}{d\theta^2} \\
 \tilde{q}^{(3)}(t) &= v_c^3 \frac{d^3q(\theta)}{d\theta^3} \\
 &\vdots \\
 \tilde{q}^{(n)}(t) &= v_c^n \frac{d^nq(\theta)}{d\theta^n}.
 \end{aligned}$$

**Example 5.7** An example of synchronization between a master and two slaves is reported in Fig. 5.16. The angular position  $\theta \in [0, 360^\circ]$  is a linear function of the time, while the trajectory of the slave axes are respectively:

- A cycloidal profile from  $q_{c0}(\theta = 0^\circ) = 0^\circ$  to  $q_{c1}(\theta = 360^\circ) = 360^\circ$ .
- A polynomial profile of degree 5 interpolating the points  $q_{p0}(\theta = 0^\circ) = 0^\circ$ ,  $q_{p1}(\theta = 180^\circ) = 180^\circ$ ,  $q_{p2}(\theta = 360^\circ) = 0^\circ$ .

In the first two cycles the motion is “slow” (that is  $v_c = 360^\circ s^{-1}$ ), while in the last two the motion is made faster by increasing the (constant) speed of the master ( $v_c = 720^\circ s^{-1}$ ). Accordingly, in these cycles both the slave velocities are doubled, while the accelerations are four times those of the first two periods.  $\square$



**Fig. 5.16.** Synchronization of two electronic cams with different profiles: cycloidal (solid) and polynomial of degree 5 (dashed).

## Trajectories and Actuators

The main properties of the trajectories most frequently adopted in industrial applications are now considered. In particular, the following discussion highlights the connections and implications between the choice of a trajectory and the actuation system. As a matter of fact any given actuation system, with its physical limits, has a relevant impact on the selection of the motion law and, conversely, it may happen that the motor is selected on the basis of the desired trajectory to be executed.

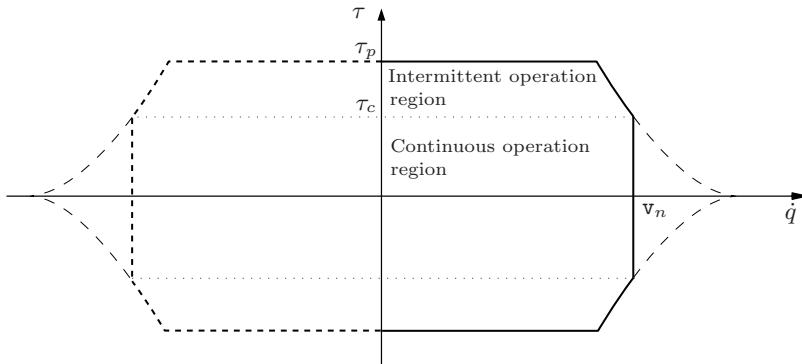
---

### 6.1 Trajectories and Electric Motors

Each trajectory, characterized by different profiles and maximum values of velocity, acceleration, and jerk, generates different effects on the actuator, on the motion transmission system, and on the mechanical load. Moreover, it should always be considered that the motion profile has also relevant implications on the tracking errors. Therefore, in the choice of a trajectory also the control aspects should be taken into account.

For these reasons, it is fundamental to choose the desired trajectory by taking into consideration the available actuator or, vice versa, to select the actuator on the basis (also) of the desired motion profile. Electric motors will be considered in the following discussion, since these are a very important class of actuators for automatic machines.

It is well known that electric motors can be synthetically described by means of their *mechanical characteristics*, usually expressed by diagrams in which the torques are reported as functions of the speed, see for example Fig. 6.1 [54]. The speed-torque curve shows some important properties of the electric motor.



**Fig. 6.1.** Generic speed-torque diagram for an electric motor.

It may be noticed that the torque generated by the motor in general is not constant over the entire speed range. Two regions can be distinguished:

- The *continuous operation region*, in which the motor can operate indefinitely.
- The *intermittent operation region*, where the motor can work only for a limited period of time, because of thermal reasons.

If the motor operates in the latter region, it may happen that the excess of generated heat could not be dissipated, and therefore a thermal protection will eventually stop the motor to avoid damages. Therefore, the duration of the task performed by the motor in this region has to be carefully considered in the definition of the trajectory. It is clear that stationary working points, i.e. with speed and torque constant for long intervals, should be in the continuous operation region. For these motions the size of the motor must be chosen to satisfy the thermal constraint. Vice versa, if “fast” and cyclic motions are involved, one can exploit also the intermittent region with a proper design of the motion profile.

Strictly related to the speed-torque diagram, the following motor parameters (reported in the data-sheets) provide important information for sizing the actuation system [54, 55]:

- *Continuous torque* ( $\tau_c$ ) (or *rated torque*): torque that the motor can produce continuously without exceeding thermal limits.
- *Peak torque* ( $\tau_p$ ): maximum torque that the motor can generate for short periods.
- *Rated speed* ( $v_n$ ): maximum value of the speed at rated torque (and at rated voltage).
- *Maximum power*: maximum amount of output power generated by the motor.

The joint analysis of these features and of the characteristics of the desired motion profile may be very useful in the design of the actuation system. With this respect, a possible procedure for the choice of the motor is here discussed, without taking into consideration other important aspects of the design, such as the selection of the inverter or the sizing of the transmission system (including the reduction ratio), since these issues are less relevant for the purpose of this discussion.

### 6.1.1 Trajectories and choice of the actuator

The selection of a suitable actuation system for a given task must consider two main aspects: the kinematic characteristics (maximum velocity, acceleration, etc.) of the motion law  $q(t)$  necessary to perform the task, and the dynamic features of the load and of the motor [56]. In particular, besides the obvious condition on the maximum allowable speed

$$\dot{q}_{max} \leq v_n$$

it is also necessary to verify whether the torque  $\tau(t)$ , necessary to perform the task, can be actually provided by the motor, i.e.

$$\max_t \{\tau(t)\} = \tau_{max} \leq \tau_p.$$

In an automatic machine, the torque of a generic actuator can be considered to be composed by two main terms:

$$\tau(t) = \tau_i(t) + \tau_{rl}(t)$$

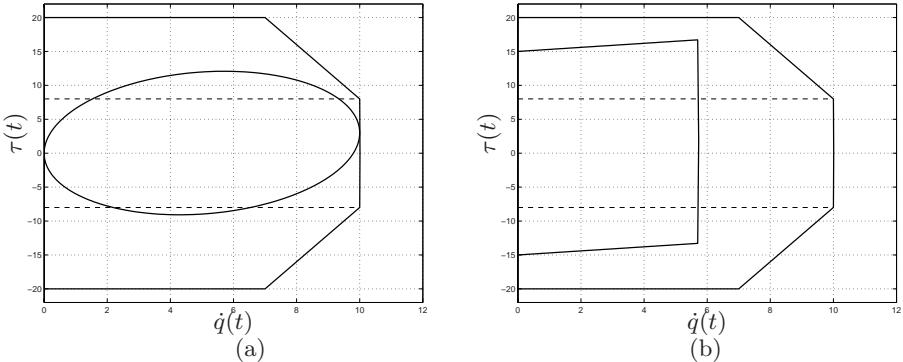
the inertial torque  $\tau_i$  required to accelerate and decelerate the load (and the rotor), and the reflected torque  $\tau_{rl}$  including all external forces, such as friction, gravity, and applied forces. In the following discussion, a reduction gear between the motor and the mechanical load is implicitly considered, and all the variables of interest, i.e. acceleration, velocity, position, inertia and friction, are computed at the motor side.

In case that only inertial and frictional forces are considered, the expression of the torque becomes

$$\tau(t) = J_t \ddot{q}(t) + B_t \dot{q}(t) \quad (6.1)$$

where  $J_t = J_m + J_l/k_r^2$  is the total moment of inertia, composed by the contribution of the motor  $J_m$  and of the load  $J_l$  reported at the motor side (and therefore divided by the square of the reduction ratio  $k_r$ ), while  $B_t$  is the damping coefficient of the overall system, whose expression is  $B_t = B_m + B_l/k_r^2$ .

A motion law can be actually performed by a given motor if the *mechanical task*, i.e. the curve described by  $(\dot{q}(t), \tau(t))$ ,  $t \in [0, T]$ , is completely included within the area defined by speed-torque characteristics of the motor itself. Vice



**Fig. 6.2.** Mechanical tasks obtained with two different motion laws: a cycloidal trajectory (a) and a trapezoidal velocity trajectory (b).

versa, once the mechanical task is known, it is possible to select the actuator, whose speed-torque characteristics contains the curve, with the purpose to exploit also the intermittent region of the motor.

When the load is mainly inertial, and the friction can be neglected,  $\tau(t) \approx J_t \ddot{q}(t)$ , and therefore the acceleration profile provides a good estimate of the torque needed to perform the planned application.

**Example 6.1** Fig. 6.2(a) and Fig. 6.2(b) refer to a motor characterized by

$$v_n = 10, \quad \tau_p = 20, \quad \tau_c = 8$$

used to actuate a system with inertia and damping coefficients

$$J_t = 1, \quad B_t = 0.3.$$

Two different motion laws are considered: in case (a) a cycloidal trajectory with  $h = 15$ ,  $T = 3$ , while in case (b) a trapezoidal velocity trajectory with the additional condition  $a_{max} = 15$ . By considering the maximum values of torques and speeds, it results that both tasks are feasible since the related diagrams lie within the area enclosed by the speed-torque curve of the motor.  $\square$

When cyclic trajectories<sup>1</sup> are considered, it is necessary to take into account also the thermal problem. As already mentioned, the speed-torque diagram of an electric motor has two main working regions: the continuous and the intermittent region. For thermic reasons, a period in the intermittent working region must be followed by a period in which the excess of heat is dissipated. Therefore, the cycle imposed by the trajectory should be designed so that the excess of thermal energy can be dissipated by the cooling system.

<sup>1</sup> In this context, “cyclic” means that the period of the trajectory is considerably smaller than the *thermal time constant* of the motor, that is the time it takes for the motor to reach 63.2 % of its rated temperature.

A simple condition, which allows to check whether a periodic motion law is feasible or not for a given motor, concerns the Root Mean Square (RMS) value of the torque  $\tau(t)$  required by the task, which can be computed as

$$\tau_{rms} = \sqrt{\frac{1}{T} \int_0^T \tau^2(t) dt}.$$

Such a torque is compared with the continuous torque  $\tau_c$  of the motor. The task is compatible with the thermal characteristics of the motor only if

$$\tau_{rms} \leq \tau_c.$$

For a generic system, like the one represented by eq. (6.1), the RMS torque is given by

$$\begin{aligned} \tau_{rms}^2 &= \frac{1}{T} \int_0^T \tau^2(t) dt \\ &= \frac{J_t^2}{T} \int_0^T \ddot{q}^2(t) dt + \frac{B_t^2}{T} \int_0^T \dot{q}^2(t) dt + 2 \frac{J_t B_t}{T} \int_0^T \dot{q}(t) \ddot{q}(t) dt \\ &= J_t^2 \ddot{q}_{rms}^2 + B_t^2 \dot{q}_{rms}^2 \end{aligned}$$

where

$$\ddot{q}_{rms} = \sqrt{\frac{1}{T} \int_0^T \ddot{q}^2(t) dt}, \quad \dot{q}_{rms} = \sqrt{\frac{1}{T} \int_0^T \dot{q}^2(t) dt}$$

are the RMS values of the acceleration and velocity respectively, and the term  $2 \frac{J_t B_t}{T} \int_0^T \dot{q}(t) \ddot{q}(t) dt$  is equal to zero if repetitive motions are considered<sup>2</sup> ( $\dot{q}(0) = \dot{q}(T)$ ).

In case  $B_t \approx 0$ , the expression of the RMS torque can be further simplified and becomes

$$\tau_{rms} = J_t \ddot{q}_{rms}.$$

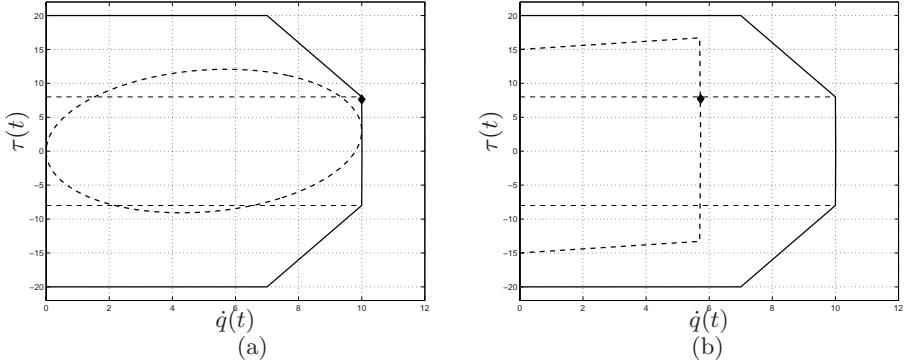
Therefore, the RMS value of the acceleration profile (multiplied by the total moment of inertia) is a good estimate of the RMS torque.

**Example 6.2** The same system (motor and load) and motion laws of Example 6.1 are considered. The RMS torques are respectively

$$\tau_{rms}^{(a)} = 7.6293, \quad \tau_{rms}^{(b)} = 7.7104$$

---

<sup>2</sup> Note that  $2 \int_0^T \dot{q}(t) \ddot{q}(t) dt = [\dot{q}(t)^2]_{t=0}^{t=T} = 0$ .



**Fig. 6.3.** Location of the point  $(\dot{q}_{max}, \tau_{rms})$  for two different motion laws: a cycloidal trajectory (a) and a trapezoidal velocity trajectory (b).

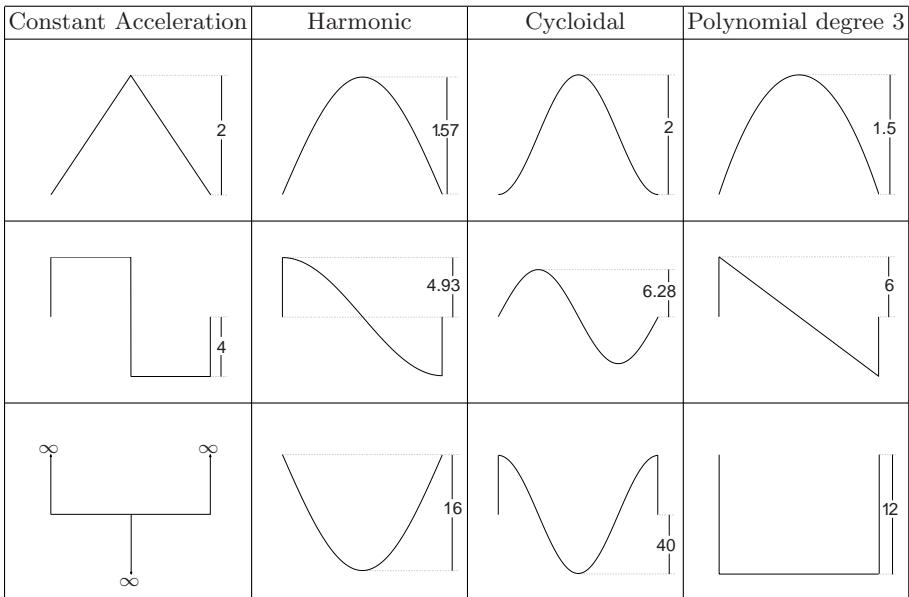
and, since  $\tau_c = 8$ , both trajectories are feasible. Note that in the two cases the point<sup>3</sup>  $(\tau_{rms}, \dot{q}_{max})$  falls within the continuous operating area of the speed-torque curve, see Fig. 6.3.  $\square$

## 6.2 Characteristics of the Motion Profiles

It is not possible to define a priori what is the “best trajectory” for *any* application. The choice is related to several considerations that depend on the type of load, on the boundary conditions (displacement length, motion duration, etc.), on the motion profile, on the available actuation system and, in general, on many other constraints. In any case, for the choice of the motion law it may be of interest to have some information about the profiles of velocity and acceleration of a specific trajectory or of a set of possible trajectories. In particular, as seen in the previous section, peak and RMS values are of great importance for sizing the actuation system, or vice versa for choosing a proper motion law for a given motor. For this purpose, it may be convenient to define some dimensionless coefficients, which do not depend on the displacement  $h$  or on the duration  $T$  of the motion law, but only on the “shape” of the trajectory. These parameters allow to quantify how the peak and RMS values of velocity and acceleration overcome the ideal mean values. If  $\dot{q}_{max} = \max_t \{|\dot{q}(t)|\}$  and  $\ddot{q}_{max} = \max_t \{|\ddot{q}(t)|\}$ , it is possible to define

$$\begin{aligned} \text{Coefficient of velocity} \quad C_v &= \frac{\dot{q}_{max}}{h/T} \quad \Rightarrow \quad \dot{q}_{max} = C_v \frac{h}{T} \\ \text{Coefficient of acceleration} \quad C_a &= \frac{\ddot{q}_{max}}{h/T^2} \quad \Rightarrow \quad \ddot{q}_{max} = C_a \frac{h}{T^2}. \end{aligned}$$

<sup>3</sup> The maximum speeds are respectively  $\dot{q}_{max}^{(a)} = 10$  and  $\dot{q}_{max}^{(b)} = 5.7295$ .



**Fig. 6.4.** From the top, velocity, acceleration and jerk profiles for the trajectories: with constant acceleration, harmonic, cycloidal, 3-rd degree polynomial.

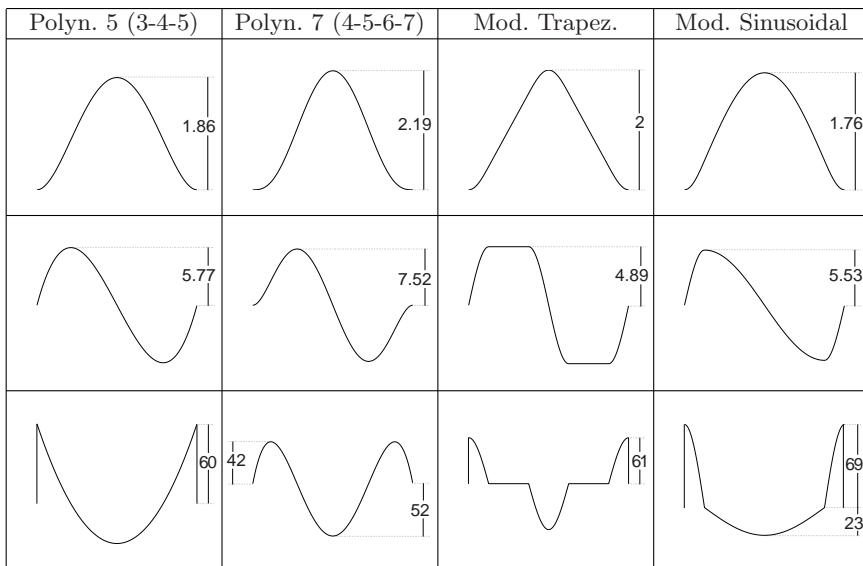
Since the maximum velocity cannot be smaller than the mean speed  $h/T$ ,  $C_v$  is certainly greater than 1, while it is possible to prove that  $C_a$  cannot be smaller than 4. In the same manner, it is possible to define a coefficient of the peak value of the jerk whose expression is  $C_j = \frac{q_{max}^{(3)}}{h/T^3}$ , where  $q_{max}^{(3)}$  is the maximum value of the jerk, although this parameter is not used in the following considerations. Obviously, because of their definitions, the coefficients  $C_v$ ,  $C_a$ ,  $C_j$  are the maximum values for velocity, acceleration, and jerk of normalized trajectories  $q_N(\tau)$ .

By taking into account the RMS values of the speed and acceleration  $\dot{q}_{rms}$  and  $\ddot{q}_{rms}$ , the corresponding coefficients are defined as

$$C_{v,rms} = \frac{\dot{q}_{rms}}{h/T}, \quad C_{a,rms} = \frac{\ddot{q}_{rms}}{h/T^2}.$$

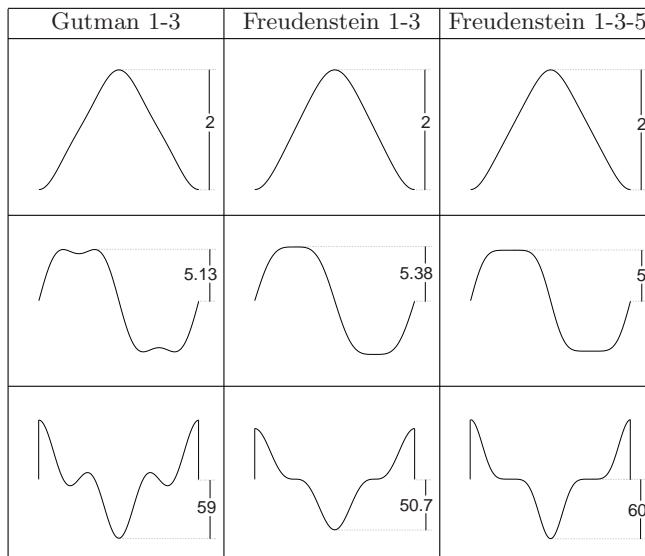
Figures 6.4-6.6 show three tables that report, in each column, the velocity, acceleration and jerk profiles of the main trajectories illustrated in previous chapters. The profiles have been computed with the conditions  $t_0 = 0$ ,  $t_1 = 1$ ,  $q_0 = 0$ ,  $q_1 = 1$  (therefore,  $h = 1$  and  $T = 1$ ), which imply  $C_v = \dot{q}_{max}$  and  $C_a = \ddot{q}_{max}$ .

The coefficient  $C_v$  and  $C_a$  of all the trajectories considered are collected in Tab. 6.1, where the per cent variation with respect to the minimum theoretical



**Fig. 6.5.** From the top, velocity, acceleration and jerk profiles for the trajectories: 5-th, 7-th degree polynomial, modified trapezoidal, modified sinusoidal.

values, respectively 1 and 4, are also reported. The coefficients for polynomial trajectories of degree higher than 7 are reported in Tab. 2.2.



**Fig. 6.6.** From the top, velocity, acceleration and jerk profiles for the trajectories: Gutman 1-3, Freudenstein 1-3, Freudenstein 1-3-5.

From the table, it is clear that trajectories with smoother profiles present higher peak values of velocity and acceleration. These values are of interest since the dynamic forces applied to a mechanism by the motor are proportional to its acceleration, see eq. (6.1). In general, it is desired to maintain these forces as low as possible, and therefore it is convenient to choose functions with low acceleration and continuous profiles. Moreover, since the kinetic energy is proportional to the velocity, it may result convenient to use trajectories characterized also by low velocity values.

Since in all those applications in which inertial loads are predominant the RMS torque required to the motor is proportional to the RMS value of the acceleration, it may be of interest to compare the coefficients  $C_{a,rms}$  relative to the different trajectories. Table 6.2 summarizes peak and RMS coefficients of the motion laws already considered in Tab. 6.1.

In order to avoid overloading the actuators and the application of excessive stress on the mechanical components, it is necessary to minimize the above coefficients. In general, a compromise is necessary, since by minimizing one of the coefficients of speed or acceleration, the other ones increase.

By considering the RMS values, the “best” trajectory is the 3-rd degree polynomial, for which  $C_{a,rms} = 3.4131$ , while the worst one is the 7-th degree polynomial, characterized by  $C_{a,rms} = 5.0452$ .

On the other hand, by considering the maximum acceleration(/torque) values, the trajectory with the minimum coefficient  $C_a$  is the constant acceleration (i.e. the triangular, the limit case of the trapezoidal profile) with  $C_a = 4$ , while the 7-th degree polynomial is characterized by the maximum value  $C_a = 7.5107$  (+87.77% with respect to the minimum possible value). The cubic polynomial trajectory has a quite high acceleration,  $C_a = 6$ , i.e. 50% more than the constant acceleration profile.

Moreover, the constant acceleration (triangular) trajectory and the modified trapezoidal trajectory (with cycloidal blends) have the same speed coefficient  $C_v$  (and accordingly the same maximum speed), while both the peak and RMS values of the acceleration are larger for the modified trapezoidal. Similar considerations hold for the double S trajectory, that presents only a different blend profile for the acceleration.

Besides being a valid tool for comparing trajectories, the speed and acceleration coefficients are a good starting point for sizing the actuation system or, vice versa, for choosing the most appropriate trajectory for a given task, especially when high performances are required. If the load can be modelled as an inertia, the linear relation between the acceleration and the torque imposed to it allows to easily transform the torque constraints into acceleration constraints and vice versa: a multiplication by the moment of inertia  $J_t$  is sufficient. Therefore, if the motor is already available and it is necessary to select the most appropriate trajectory, one can transform the maximum ratings of the motor (peak torque  $\tau_p$  and continuative torque  $\tau_c$ ) on constraints on the motion law (maximum acceleration  $a_{max}$  and maximum value of the

Trajectory	$C_v$	$\Delta C_v \%$	$C_a$	$\Delta C_a \%$
Constant acceleration	2	100.00	4	0.00
Harmonic	1.5708	57.08	4.9348	23.37
Cycloidal	2	100.00	6.2832	57.08
Polynomial: degree 3	1.5	50.00	6	50.00
Polynomial: 3-4-5	1.875	87.5	5.7733	44.33
Polynomial: 4-5-6-7	2.1875	118.75	7.5107	87.77
Modified Trapezoidal	2	100.00	4.8881	22.20
Modified Sinusoidal	1.7593	75.93	5.5279	38.20
Gutman 1-3	2	100.00	5.1296	28.24
Freudenstein 1-3	2	100.00	5.3856	34.64
Freudenstein 1-3-5	2	100.00	5.0603	26.51

**Table 6.1.** Coefficients of the maximum values for speeds and accelerations of some of the trajectories introduced in previous chapters, and percent variations with respect to the minimum theoretical values.

Trajectory	$C_v$	$C_a$	$C_{v,rms}$	$C_{a,rms}$
Constant acceleration	2	4	1.1547	4
Harmonic	1.5708	4.9348	1.1107	3.4544
Cycloidal	2	6.2832	1.2247	4.4428
Polynomial: degree 3	1.5	6	1.0954	3.4131
Polynomial: 3-4-5	1.875	5.7733	1.1952	4.1402
Polynomial: 4-5-6-7	2.1875	7.5107	1.2774	5.0452
Modified Trapezoidal	2	4.8881	1.2245	4.3163
Modified Sinusoidal	1.7593	5.5279	1.1689	3.9667
Gutman 1-3	2	5.1296	1.2006	4.2475
Freudenstein 1-3	2	5.3856	1.2106	4.3104
Freudenstein 1-3-5	2	5.0603	1.2028	4.2516

**Table 6.2.** Coefficients of the maximum and RMS values for speeds and accelerations of some of the trajectories introduced in previous chapters.

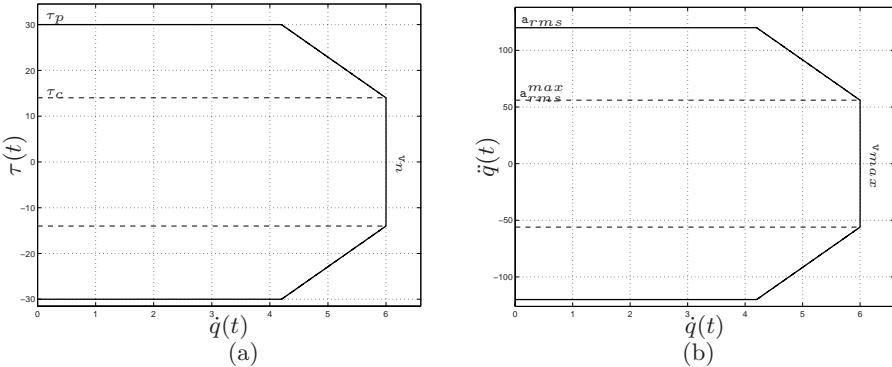
RMS acceleration  $a_{max}^{rms}$ :

$$a_{max} = \frac{\tau_p}{J_t}, \quad a_{max}^{rms} = \frac{\tau_c}{J_t}$$

besides the obvious condition on the velocity  $v_{max} = v_n$ .

On the contrary if the task is well defined, and requires a prescribed trajectory ( $C_v$ ,  $C_a$ ,  $C_{v,rms}$  and  $C_{a,rms}$  are therefore already settled), with given values of the displacement  $h$  and of the duration  $T$ , from the RMS acceleration and maximum acceleration it is possible to easily obtain the minimum value of the peak torque and continuative torque that the motor must provide to perform it.

More generally, (in case of an inertial load) from the mechanical characteristics of the motor, it is possible to deduce a curve in a speed-acceleration diagram in which the trajectory must be contained, and vice versa (see Fig. 6.7).



**Fig. 6.7.** Speed-torque and speed-acceleration diagrams for a given mechanical/actuation system.

**Example 6.3** Given a mechanical system, characterized by the total inertia  $J_t = 0.25$ , a negligible damping coefficient, i.e.  $B_t \approx 0$ , and driven by an actuator with the speed-torque diagram plotted in Fig. 6.7(a), the goal is to find the fastest harmonic trajectory from  $q_0 = 0$  to  $q_1 = 0.6$  ( $h = 0.6$ ). From the figure, the maximum ratings of the motor can be easily deduced:

$$\tau_p = 30, \quad \tau_c = 14, \quad v_n = 6$$

and the diagram of Fig. 6.7(b) is obtained, with the corresponding constraints on the trajectory:

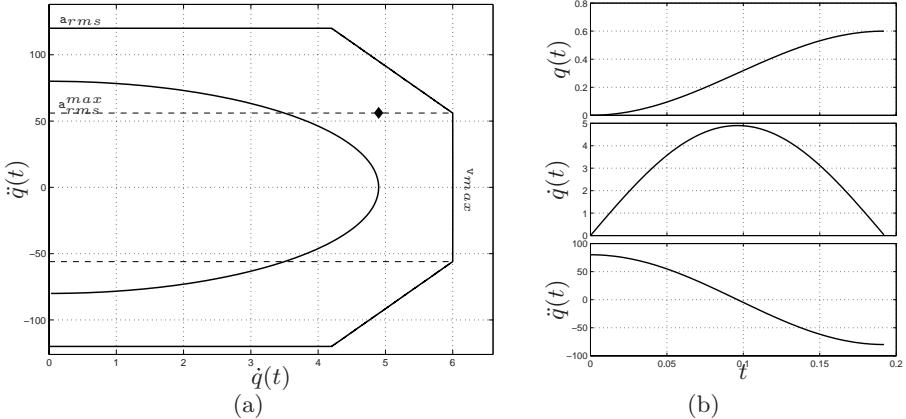
$$a_{max} = 120, \quad a_{rms}^{rms} = 56, \quad v_{max} = 6.$$

From the values of the coefficients  $C_v$ ,  $C_a$  and  $C_{a,rms}$  for the harmonic trajectory, and the constraints on the maximum velocity, acceleration and RMS acceleration, one can compute the value of  $T$ . As a matter of fact, by considering the definition of the velocity and acceleration coefficients, it follows that

$$\begin{aligned} \dot{q}_{max} &= C_v \frac{h}{T} \quad \Rightarrow \quad T \geq C_v \frac{h}{v_{max}} = T_{min,1} \\ \ddot{q}_{max} &= C_a \frac{h}{T^2} \quad \Rightarrow \quad T \geq \sqrt{C_a \frac{h}{a_{max}}} = T_{min,2} \\ \ddot{q}_{a,max} &= C_{a,rms} \frac{h}{T^2} \quad \Rightarrow \quad T \geq \sqrt{C_{a,rms} \frac{h}{a_{rms}^{rms}}} = T_{min,3}. \end{aligned} \quad (6.2)$$

The numerical values of this example lead to

$$T_{min,1} = 0.1571, \quad T_{min,2} = 0.1571, \quad T_{min,3} = 0.1924.$$



**Fig. 6.8.** Task obtained with a harmonic trajectory represented in a speed-acceleration diagram (a) and single plots of position, velocity, acceleration (b).

The most limiting constraint is relative to the RMS acceleration, which implies the highest duration of the trajectory. Therefore, the length of the motion is

$$T_{min} = \max\{T_{min,1}, T_{min,2}, T_{min,3}\} = 0.1924.$$

In Fig. 6.8 the task is reported in the speed-acceleration diagram, where the point  $(\dot{q}_{max}, \ddot{q}_{rms})$  is also explicitly represented.

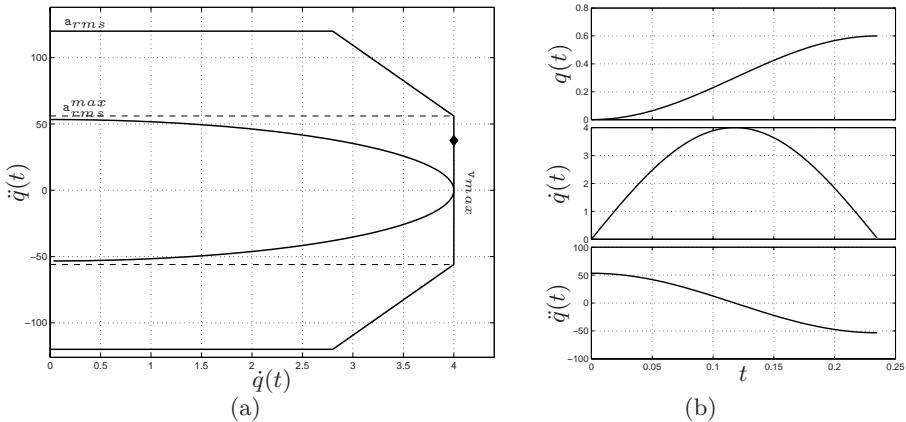
If  $v_{max} = 4$ , the durations obtained from (6.2) are

$$T_{min,1} = 0.2356, \quad T_{min,2} = 0.1571, \quad T_{min,3} = 0.1924$$

and the minimum duration of the trajectory is bounded by the maximum speed, see Fig. 6.9.  $\square$

### 6.2.1 Comparison between trapezoidal and double S trajectories

Because of their wide use in the industrial practice, it is of interest to compare the performances achievable with a trapezoidal and a double S trajectory, see Sec. 3.2 and Sec. 3.4. In order to define the trajectories, besides the conditions on the total displacement  $h = q_1 - q_0$  and on the duration  $T = t_1 - t_0$  (initial and final velocities and accelerations are assumed to be null), it is necessary to consider one more constraint for the trapezoidal motion law (e.g. the value of the maximum acceleration  $a_{max}$  or the duration  $T_a$  of the acceleration/deceleration phases supposed of the same length) and two additional constraints for the double S trajectory. In particular, for the trapezoidal motion law the duration of the acceleration and deceleration phases (supposed equal) are defined as



**Fig. 6.9.** Task obtained with a harmonic trajectory represented in a speed-acceleration diagram (a) and single plots of position, velocity, acceleration (b).

$$T_a = \alpha T, \quad \alpha \leq 1/2.$$

By substituting this value in

$$\begin{cases} T = \frac{h}{\dot{q}_{max}} + T_a \\ T_a = \frac{\dot{q}_{max}}{\ddot{q}_{max}} \end{cases}$$

it is possible to calculate the maximum values of the speed and acceleration and the related coefficients as a function of the free parameter  $\alpha$ :

$$\begin{cases} \dot{q}_{max} = \frac{h}{(1-\alpha)T} \\ \ddot{q}_{max} = \frac{h}{\alpha(1-\alpha)T^2} \end{cases} \Rightarrow \begin{cases} C_v^{tr} = \frac{1}{(1-\alpha)} \\ C_a^{tr} = \frac{1}{\alpha(1-\alpha)} \end{cases}$$

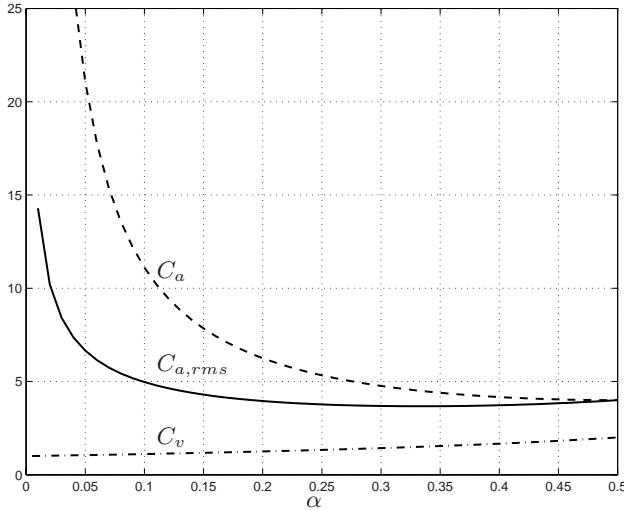
Then, one can obtain the coefficient of the RMS acceleration by substituting the expressions of  $T_a$  and  $\ddot{q}_{max}$  in the definition<sup>4</sup>

---

<sup>4</sup> For the trapezoidal trajectory, characterized by constant acceleration segments, the RMS value of the acceleration can be computed with the formula

$$\ddot{q}_{rms}^{tr} = \sqrt{\frac{\sum_i T_i a_i^2}{T}}$$

where  $T_i$  is the duration of the  $i$ -th segment with constant acceleration  $a_i$ .



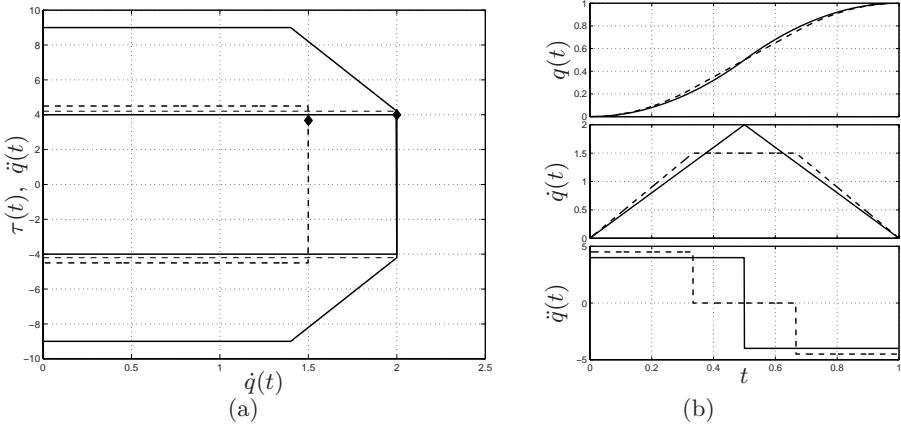
**Fig. 6.10.** Coefficients  $C_v$ ,  $C_a$  and  $C_{a,rms}$  of the trapezoidal trajectory as a function of the parameter  $\alpha$  ( $T_a = \alpha T$ ).

$\alpha$	$C_v$	$C_a$	$C_{v,rms}$	$C_{a,rms}$
1/2	2	4	1.15	4
1/3	1.5	4.5	1.12	3.67
1/4	1.33	5.33	1.09	3.77
1/5	1.25	6.25	1.07	3.95

**Table 6.3.** Coefficients of the maximum and RMS values for speeds and accelerations of trapezoidal trajectories obtained for different values of the parameter  $\alpha$ .

$$\begin{aligned}
 \ddot{q}_{rms} &= \sqrt{\frac{1}{T} \int_0^T \ddot{q}(t)^2 dt} \\
 &= \sqrt{\frac{1}{T} (2T_a \ddot{q}_{max}^2)} \\
 &= \sqrt{2\alpha} \frac{h}{\alpha(1-\alpha)T^2} \quad \Rightarrow \quad C_{a,rms}^{tr} = \sqrt{2\alpha} \frac{1}{\alpha(1-\alpha)}.
 \end{aligned}$$

In fig. 6.10, the coefficients  $C_v$ ,  $C_a$  and  $C_{a,rms}$  of the trapezoidal trajectory are plotted as a function of the parameter  $\alpha \in [0, 1/2]$ . Note that if  $\alpha$  increases also  $C_v$  (and therefore the maximum speed) grows, while  $C_a$  decreases. Conversely,  $C_{a,rms}$  is not a monotonic function of  $\alpha$ , but it has a minimum at  $\alpha = 1/3$ , see Tab. 6.3. For this reason, the trapezoidal trajectory obtained with three equal-length segments is often adopted in the industrial practice, also considering that, for  $\alpha = 1/3$ , the coefficients  $C_v$  and  $C_a$  are quite small.



**Fig. 6.11.** Tasks obtained with a trapezoidal trajectory for different values of  $\alpha$  (a) and plots of position, velocity, acceleration (b).

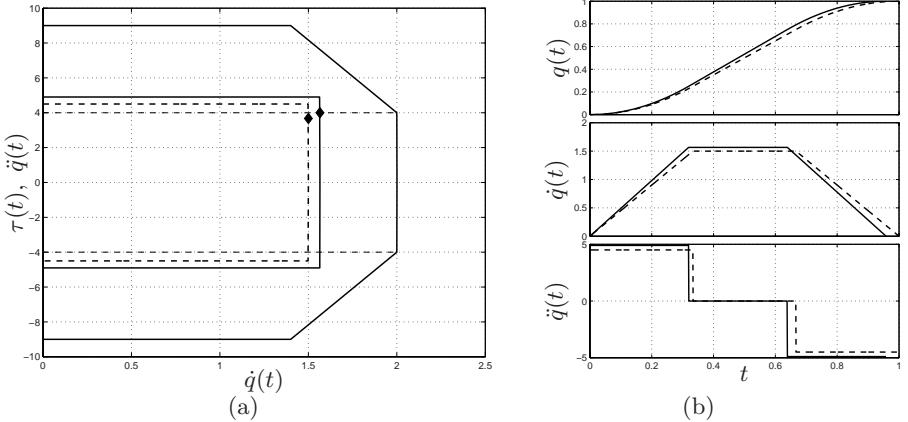
**Example 6.4** Figure 6.11 shows a task with the conditions  $h = 1$ ,  $T = 1$ , performed by means of trapezoidal trajectories computed for different values of  $\alpha$ . In particular, the values  $\alpha = 1/2$  (solid) and  $\alpha = 1/3$  (dashed) have been considered. For the sake of simplicity it is assumed that  $J_t = 1$ . In this manner, by neglecting the friction, the torque is equal to the acceleration.

From Fig. 6.11(a), where the classical speed-torque diagram of a motor (in this case equal to the diagram speed-acceleration) is reported, it results that the triangular trajectory (obtained for  $\alpha = 1/2$ ) is a limit case of feasible trajectory since not only the maximum speed  $\dot{q}_{max}$  is equal to  $v_{max} = 2$ , but also the RMS acceleration  $\ddot{q}_{rms}$  is equal to its limit value ( $a_{rms}^{max} = \tau_c/J_t = 4$ ). On the contrary, the trajectory obtained for  $\alpha = 1/3$  presents speed and acceleration values lower than the relative limits. In this case it is possible to reduce the duration  $T$  of the trajectory, which remains still feasible with the constraints imposed by the motor. Since the most limiting factor is clearly the RMS value of the acceleration (note the point  $(\dot{q}_{max}, \ddot{q}_{rms})$ , reported in the diagram of Fig. 6.11(a)), the minimum trajectory duration can be computed according to

$$T_{min} = \sqrt{C_{a,rms} \frac{h}{a_{max}^{rms}}}$$

which, with the numerical values of this example, is  $T_{min} = 0.9582$ . In Fig. 6.12 the trajectories obtained for  $T = 1$  (dashed) and  $T = 0.9582$  (solid) are reported.  $\square$

For the double S trajectory the two additional constraints here considered are the duration of the acceleration phase, supposed to be a fraction of the overall period



**Fig. 6.12.** Tasks obtained with a trapezoidal trajectory with  $\alpha = 1/3$  for different values of the duration  $T$  (a) and plots of position, velocity, acceleration (b).

$$T_a = \alpha T, \quad \alpha \leq 1/2$$

and the time length of the constant jerk phase

$$T_j = \beta T_a, \quad \beta \leq 1/2$$

that in this case is assumed to be a fraction of the acceleration period. In this way, from the expressions of the total duration, of the time length of acceleration phase and of the constant jerk segment

$$\begin{cases} T = \frac{h}{\dot{q}_{max}} + T_a \\ T_a = \frac{\dot{q}_{max}}{\ddot{q}_{max}} + T_j \\ T_j = \frac{\ddot{q}_{max}}{q_{max}^{(3)}} \end{cases}$$

one obtains the values of maximum speed, acceleration and jerk of the double S trajectory  $q(t)$  as a function of  $T$ ,  $h$ ,  $\alpha$ ,  $\beta$ :

$$\begin{cases} \dot{q}_{max} = \frac{h}{(1-\alpha)T} \\ \ddot{q}_{max} = \frac{h}{\alpha(1-\alpha)(1-\beta)T^2} \\ q_{max}^{(3)} = \frac{h}{\alpha^2\beta(1-\alpha)(1-\beta)T^3} \end{cases} \Rightarrow \begin{aligned} C_v^{ss} &= \frac{1}{(1-\alpha)} \\ C_a^{ss} &= \frac{1}{\alpha(1-\alpha)(1-\beta)} \\ C_j^{ss} &= \frac{1}{\alpha^2\beta(1-\alpha)(1-\beta)}. \end{aligned}$$

Moreover, it is possible to compute the RMS value of the acceleration of the double S trajectory as a function of  $h$ ,  $T$ ,  $\alpha$ ,  $\beta$ , by substituting the expressions of  $T_a$ ,  $T_j$  and  $\ddot{q}_{max}$  in its definition:

$\alpha \backslash \beta$	1/2	1/3	1/4	1/5
1/2	2.0000	2.0000	2.0000	2.0000
1/3	1.5000	1.5000	1.5000	1.5000
1/4	1.3333	1.3333	1.3333	1.3333
1/5	1.2500	1.2500	1.2500	1.2500

**Table 6.4.** Coefficient  $C_v$  of the double S trajectory, for some values of the parameters  $\alpha$  and  $\beta$ .

$\alpha \backslash \beta$	1/2	1/3	1/4	1/5
1/2	8.0000	6.0000	5.3333	5.0000
1/3	8.9820	6.7500	6.0000	5.6250
1/4	10.6667	8.0000	7.1111	6.6667
1/5	12.5000	9.3750	8.3333	7.8125

**Table 6.5.** Coefficient  $C_a$  of the double S trajectory, for some values of the parameters  $\alpha$  and  $\beta$ .

$$\begin{aligned} \ddot{q}_{rms} &= \sqrt{\frac{1}{T} \int_0^T \ddot{q}(t)^2 dt} \\ &= \sqrt{\frac{1}{T} \left( 2(T_a - 2T_j) \ddot{q}_{max}^2 + \frac{4}{3} T_j \ddot{q}_{max}^2 \right)} \\ &= \frac{1}{(1-\alpha)(1-\beta)} \sqrt{\frac{6-8\beta}{3\alpha}} \frac{h}{T^2} \Rightarrow C_{a,rms}^{ss} = \frac{1}{(1-\alpha)(1-\beta)} \sqrt{\frac{6-8\beta}{3\alpha}}. \end{aligned}$$

The numerical values of the coefficients  $C_v$ ,  $C_a$ ,  $C_{v,rms}$ ,  $C_{a,rms}$  for  $\alpha$  and  $\beta$  equal to 1/2, 1/3, 1/4, 1/5 are reported in Tables 6.4-6.7.

**Example 6.5** The same system (motor and load) of Example 6.4 is considered, but in this case the task is performed by means of double S trajectories. In particular, two different motion laws obtained with  $\alpha = 1/2$ ,  $\beta = 1/4$  (dashed) and  $\alpha = 1/3$ ,  $\beta = 1/4$  (solid) are taken into account.

While both trajectories are compliant with the constraints on the maximum speed and acceleration, the RMS acceleration in the case  $\alpha = 1/2$  overcomes the maximum allowed value. As a matter of fact, as can be easily deduced from Tab. 6.7 by considering  $h = 1$  and  $T = 1$ , the RMS acceleration is  $\ddot{q}_{rms} = 4.3547$  for  $\alpha = 1/2$  and  $\ddot{q}_{rms} = 4$  for  $\alpha = 1/3$ . Therefore, if one intends to adopt a trajectory without constant velocity segment ( $\alpha = 1/2$ ) it is necessary to increase its duration, whose minimum value is, for a unitary displacement,  $T = 1.0434$ . In this way, the performance with respect to the trapezoidal trajectory (characterized by  $\ddot{q}_{rms} = 4$  for  $T = 1$ ) is reduced of

$\begin{array}{c} \beta \\ \alpha \end{array}$	1/2	1/3	1/4	1/5
1/2	1.2383	1.2270	1.2141	1.2045
1/3	1.1511	1.1466	1.1414	1.1376
1/4	1.1089	1.1061	1.1029	1.1006
1/5	1.0849	1.0829	1.0807	1.0790

**Table 6.6.** Coefficient  $C_{v,rms}$  of the double S trajectory, for some values of the parameters  $\alpha$  and  $\beta$ .

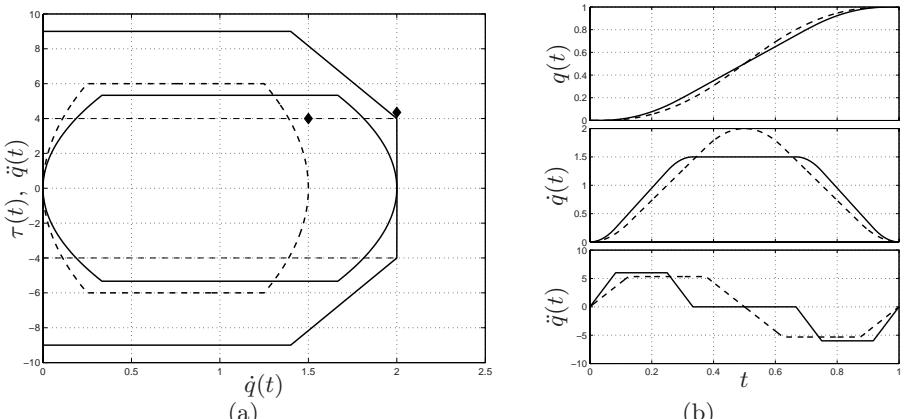
$\begin{array}{c} \beta \\ \alpha \end{array}$	1/2	1/3	1/4	1/5
1/2	4.6188	4.4721	4.3547	4.2818
1/3	4.2426	4.1079	4.0000	3.9330
1/4	4.3547	4.2163	4.1056	4.0369
1/5	4.5645	4.4194	4.3034	4.2314

**Table 6.7.** Coefficient  $C_{a,rms}$  of the double S trajectory, for some values of the parameters  $\alpha$  and  $\beta$ .

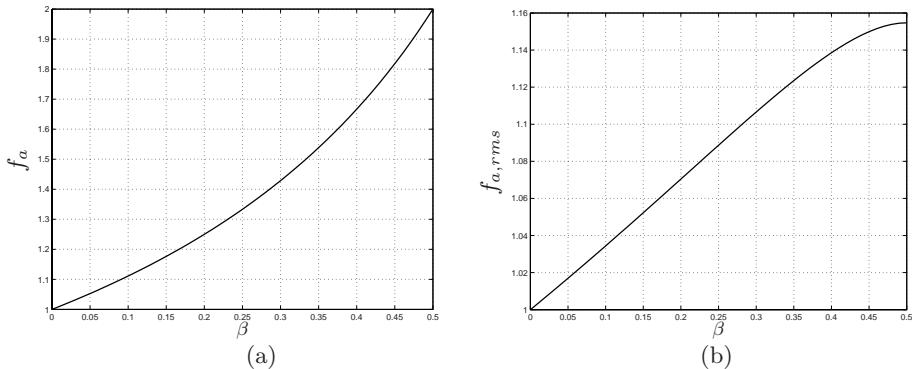
about 4%.

In the case  $\alpha = 1/3$ , one can reduce  $\beta$  in order to decrease  $C_{a,rms}$  and accordingly the acceleration  $\ddot{q}_{rms}$  (see Tab. 6.7).  $\square$

It could be of interest to compare the values of the velocity and acceleration coefficients obtained for trapezoidal and double S trajectories. This is possible since the meaning of  $T_a$ , and therefore of  $\alpha = T/T_a$ , is the same for both the



**Fig. 6.13.** Tasks obtained with a double S trajectory for two different values of  $\alpha$  ( $\beta = 1/4$  in both cases) (a) and plots of position, velocity, acceleration (b).



**Fig. 6.14.** Functions of  $\beta$  relating the coefficients  $C_a$  (a) and  $C_{a,rms}$  (b) of the trapezoidal and double S trajectories.

trajectories. In particular, it is evident that<sup>5</sup>

$$C_v^{ss} = C_v^{tr}$$

while the relationships between the coefficients  $C_a$  and  $C_{a,rms}$  of the two trajectories are more complex and related to the  $\beta$  factor of the double S. By inspection, it can be easily observed that

$$C_a^{ss} = f_a(\beta) \quad C_a^{tr} \quad \text{with} \quad f_a(\beta) = \frac{1}{1 - \beta}$$

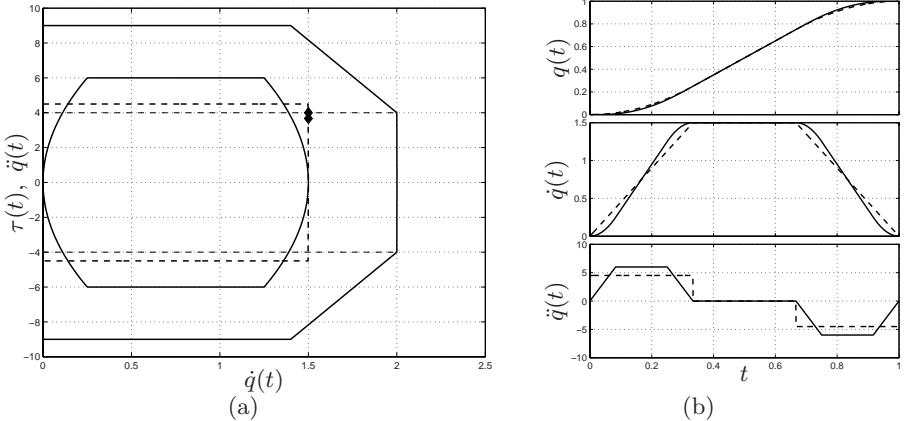
and

$$C_{a,rms}^{ss} = f_{a,rms}(\beta) \quad C_{a,rms}^{tr} \quad \text{with} \quad f_{a,rms}(\beta) = \frac{1}{1 - \beta} \sqrt{\frac{3 - 4\beta}{3}}.$$

The functions  $f_a(\beta)$  and  $f_{a,rms}(\beta)$  are plotted in Fig. 6.14; while the function  $f_a(\beta)$  doubles for  $\beta$  varying from 0 to  $1/2$ ,  $f_{a,rms}(\beta)$  ranges from 1 to 1.1547. As a consequence, the RMS value of the acceleration of the double S trajectory differs from  $\ddot{q}_{rms}$  of a trapezoidal motion law, computed with the same  $\alpha$ , at most of 15.47%. For  $\beta = 0$  (that is  $T_j = 0$ ) the two functions  $f_a(\beta)$  and  $f_{a,rms}(\beta)$  are both equal to 1, since in this case the double S degenerates in the trapezoidal trajectory (characterized by an impulsive jerk of infinite amplitude).

**Example 6.6** The trapezoidal and double S trajectories, considered in Examples 6.4 and 6.5 (for  $\alpha = 1/3$ ), are reported in Fig. 6.15. The double S trajectory has a smoother profile, with obvious advantages in terms of harmonic contents (see Ch. 7). On the other hand, the trapezoidal trajectory shows lower maximum (peak and RMS) values of acceleration, but also the

<sup>5</sup> The superscript *ss* stands for double S, while *tr* for trapezoidal.



**Fig. 6.15.** Comparison between trapezoidal (dashed) and double S (solid) trajectories for  $\alpha = 1/3$ : speed-acceleration diagram (a) and plots of position, velocity, acceleration (b).

double S is fully compatible with the given constraints and, moreover, it allows a better exploitation of the intermittent working region of the motor. In both cases, the most limiting factor of the performance is  $\ddot{q}_{rms}$  which is 4 for the normalized double S trajectory and 3.67 for the trapezoidal one. In terms of minimum time duration of the two trajectories, with the constraints considered in this example, the trapezoidal is characterized by  $T_{min} = 0.9582$ , while for the double S the minimum value is  $T = 1$ . Therefore, the difference is less than 5%.

□

## Dynamic Analysis of Trajectories

*Vibrations* are undesired phenomena often present in automatic machines. They are basically due to the presence of structural elasticity in the mechanical system, and may be generated during the normal working cycle of the machine for several causes. In particular, among other reasons, vibrations may be produced if trajectories with a discontinuous acceleration profile are imposed to the actuation system. As a matter of fact, a discontinuity in the acceleration profile implies a rapid variation (discontinuity) of the inertial forces applied to the mechanical structure. Relevant discontinuities of these forces, when applied to an elastic system, generate vibrations. Since every mechanism is characterized by some degree of elasticity, this type of phenomenon must always be considered in the design of a trajectory, that therefore should have a smooth acceleration profile or, more in general, a limited bandwidth.

---

### 7.1 Models for Analysis of Vibrations

In order to analyze vibrational phenomena, it is necessary to use models that take into account the elastic, inertial and dissipative properties of the mechanical devices composing an automatic machine. The complexity level of the model is usually chosen as a compromise between the desired precision and the computational burden. The simplest criterion, quite often adopted in practice, is to describe the mechanical devices, that are intrinsically distributed parameter systems, as lumped parameter systems, i.e. as pure rigid masses (without elasticity) and pure elastic elements (without mass). Moreover, energy dissipative elements are introduced in order to consider frictional phenomena among moving parts. The numerical values of the elements that describe inertia, elasticity and dissipative effects have to be determined by energetic considerations, i.e. trying to maintain the equivalence of the kinetic

and elastic energy of the model with the energy of the corresponding parts of the mechanism under study. The description of these phenomena will be either linear or nonlinear.

Some models for the analysis of the vibrational effects in automatic machines are now described, as examples of the mathematical tools that should be adopted for this type of study. The order of presentation starts from simple to more complex models, able to better describe the dynamics of the real mechanical systems, with the drawback of an increased computational complexity. For notational ease, all the models are described considering linear motions and forces. The discussion is obviously valid also in case of rotational movements and torques.

### 7.1.1 Linear model with one degree of freedom

Let us consider the lumped parameters model composed by a mass  $m$ , a spring with elastic coefficient  $k$  and a dumper  $d$  that takes into consideration the friction and then dissipates energy, see Fig. 7.1. Let  $x$  be the position of the mass  $m$ , and  $y$  the input position of the mechanism, i.e. the position of the actuation system. The dynamics of the system is described by the differential equation

$$m \ddot{x} + d \dot{x} + k x = d \dot{y} + k y. \quad (7.1)$$

If the difference  $z = x - y$  due to the elastic element is considered, one obtains

$$m \ddot{z} + d \dot{z} + k z = -m \ddot{y}$$

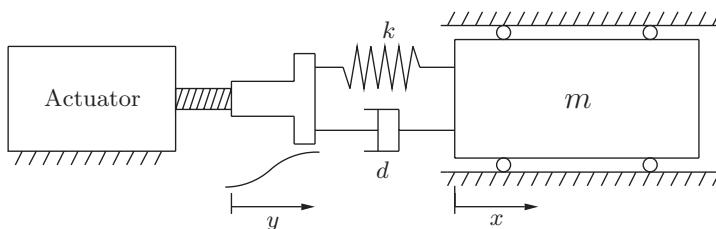
or

$$\ddot{z} + 2\delta\omega_n\dot{z} + \omega_n^2 z = -\ddot{y} \quad (7.2)$$

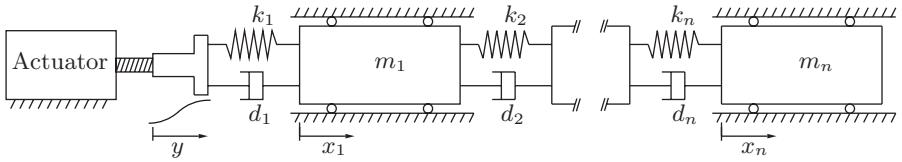
where

$$\omega_n = \sqrt{\frac{k}{m}}, \quad \delta = \frac{d}{2m\omega_n}$$

are the *natural frequency* and the *dumping coefficient* respectively of the considered dynamic model. The second order differential equation (7.2), with



**Fig. 7.1.** Linear model with one degree of freedom.



**Fig. 7.2.** Linear model with  $n$  degrees of freedom.

proper initial conditions  $z(0) = z_0$ ,  $\dot{z}(0) = \dot{z}_0$ , represents a one degree of freedom model of the mechanism under study. In the state space, with  $x_1 = z$  and  $x_2 = \dot{z}$ , the corresponding model is given by

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\omega_n^2 x_1 - 2\delta\omega_n x_2 - \ddot{y}. \end{cases}$$

In matrix form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

with

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\delta\omega_n \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad u = \ddot{y}$$

and initial conditions  $x_1(0) = x_{10} = z(0)$  and  $x_2(0) = x_{20} = \dot{z}(0)$ .

### 7.1.2 Linear model with $n$ degrees of freedom

A more complex model, closer to a distributed parameters model and able to describe in a more detailed way the physical behavior of a mechanical system, takes into consideration  $n$  lumped parameters, as shown in Fig. 7.2.

Let  $x_1, x_2, \dots, x_n$  be the positions of the masses  $m_1, m_2, \dots, m_n$ , and  $y$  the input position. The dynamics of the system is now described by the following system of differential equations

$$\begin{cases} m_1\ddot{x}_1 + d_1(\dot{x}_1 - \dot{y}) + k_1(x_1 - y) - d_2(\dot{x}_2 - \dot{x}_1) - k_2(x_2 - x_1) = 0 \\ m_2\ddot{x}_2 + d_2(\dot{x}_2 - \dot{x}_1) + k_2(x_2 - x_1) - d_3(\dot{x}_3 - \dot{x}_2) - k_3(x_3 - x_2) = 0 \\ \vdots \\ m_i\ddot{x}_i + d_i(\dot{x}_i - \dot{x}_{i-1}) + k_i(x_i - x_{i-1}) - d_{i+1}(\dot{x}_{i+1} - \dot{x}_i) - k_{i+1}(x_{i+1} - x_i) = 0 \\ \vdots \\ m_n\ddot{x}_n + d_n(\dot{x}_n - \dot{x}_{n-1}) + k_n(x_n - x_{n-1}) = 0. \end{cases}$$

By considering

$$\begin{cases} z_1 = x_1 - y \\ z_2 = x_2 - x_1 \\ \vdots \\ z_n = x_n - x_{n-1} \end{cases}$$

or, in a more compact form,

$$x_i = \sum_{j=1}^i z_j + y, \quad i = 1, \dots, n$$

the equations of the system may be written as

$$\begin{cases} m_1 \ddot{z}_1 + d_1 \dot{z}_1 + k_1 z_1 - d_2 \dot{z}_2 - k_2 z_2 & = -m_1 \ddot{y} \\ m_2 (\ddot{z}_1 + \ddot{z}_2) + d_2 \dot{z}_2 + k_2 z_2 - d_3 \dot{z}_3 - k_3 z_3 & = -m_2 \ddot{y} \\ \vdots \\ m_{n-1} \sum_{j=1}^{n-1} \ddot{z}_j + d_{n-1} \dot{z}_{n-1} + k_{n-1} z_{n-1} - d_n \dot{z}_n - k_n z_n & = -m_{n-1} \ddot{y} \\ m_n \sum_{j=1}^n \ddot{z}_j + d_n \dot{z}_n + k_n z_n & = -m_n \ddot{y} \end{cases}$$

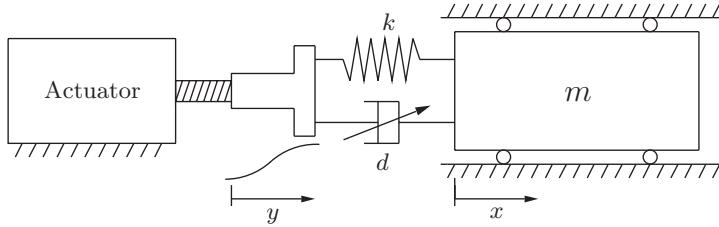
or, in matrix form, as

$$\mathbf{M}\ddot{\mathbf{z}} + \mathbf{D}\dot{\mathbf{z}} + \mathbf{K}\mathbf{z} = -\mathbf{m} \ddot{y} \quad (7.3)$$

where

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 & 0 & \dots & 0 & 0 \\ m_2 & m_2 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n-1} & m_{n-1} & m_{n-1} & \dots & m_{n-1} & 0 \\ m_n & m_n & m_n & \dots & m_n & m_n \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} d_1 & -d_2 & 0 & 0 & \dots & 0 & 0 \\ 0 & d_2 & -d_3 & 0 & \dots & 0 & 0 \\ 0 & 0 & d_3 & -d_4 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & d_{n-1} & d_n \\ 0 & 0 & 0 & 0 & \dots & 0 & d_n \end{bmatrix}$$



**Fig. 7.3.** One degree of freedom model with passive nonlinear parameters.

$$\mathbf{K} = \begin{bmatrix} k_1 & -k_2 & 0 & 0 & \dots & 0 & 0 \\ 0 & k_2 & -k_3 & 0 & \dots & 0 & 0 \\ 0 & 0 & k_3 & -k_4 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & k_{n-1} & k_n \\ 0 & 0 & 0 & 0 & \dots & 0 & k_n \end{bmatrix}$$

$$\mathbf{m} = [m_1, m_2, m_3, m_4, \dots, m_{n-1}, m_n]^T.$$

Since matrix  $\mathbf{M}$  is always invertible (its eigenvalues are  $m_1, m_2, \dots, m_n > 0$ ), from (7.3) one obtains

$$\ddot{\mathbf{z}} = -\mathbf{M}^{-1}\mathbf{D}\dot{\mathbf{z}} - \mathbf{M}^{-1}\mathbf{K}\mathbf{z} - \mathbf{M}^{-1}\mathbf{m}\ddot{y}.$$

By defining the state vector as

$$\mathbf{x} = [z_1, z_2, \dots, z_n, \dot{z}_1, \dot{z}_2, \dots, \dot{z}_n]^T$$

the model in state space is finally obtained:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (7.4)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^{-1}\mathbf{m} \end{bmatrix}, \quad u = \ddot{y}.$$

### 7.1.3 Nonlinear model with one degree of freedom

Usually, a mechanical system is characterized by nonlinear effects that may be considered by properly defining the ‘passive’ parameters of the model. Several nonlinear effects may be taken into account, such as Coulomb or viscous friction, nonlinear damping (e.g. proportional to  $\dot{x}^2$ ), backslash, and so on. Let us consider the system of Fig. 7.3 where, in particular, the damping element takes into consideration different frictional phenomena. The parameters  $d$ ,  $\alpha$  and  $\beta$  represent the viscous, the quadratic and the Coulomb damping respectively. The dynamics of the system is described now by

$$m\ddot{z} + d\dot{z} + kz = -m\dot{y} - \alpha|\dot{z}|\dot{z} - \beta\frac{\dot{z}}{|\dot{z}|}.$$

where  $z = x - y$ . This equation may also be written as

$$\ddot{z} + 2\delta\omega_n\dot{z} + \omega_n^2 z = -\omega_n^2 \zeta$$

with

$$\zeta = \frac{\ddot{y}}{\omega_n^2} + \frac{\alpha}{m\omega_n^2}|\dot{z}|\dot{z} + \frac{\beta}{m\omega_n^2}\frac{\dot{z}}{|\dot{z}|}.$$

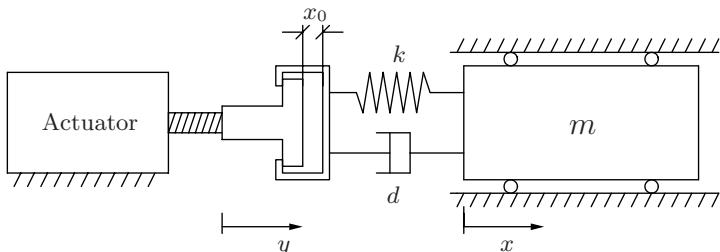
If a backslash effect has to be considered, it is necessary to define an additional nonlinear term, such as the one represented in Fig. 7.4. In this case, with a proper choice of the origin of the reference frames, the ‘contact’ between motor and load occurs only if  $y - x > x_0$ . The equations describing the dynamics are now

$$\begin{cases} m\ddot{x} + kx = 0, & y - x < x_0 \\ m\ddot{x} + d\dot{x} + kx = ky + d\dot{y}, & y - x > x_0. \end{cases}$$

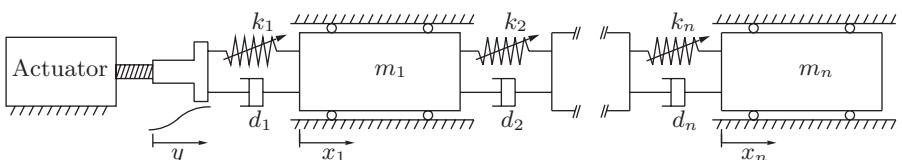
#### 7.1.4 Nonlinear model with $n$ degrees of freedom

A  $n$ -dimensional model with nonlinear passive parameters is shown in Fig. 7.5. Let  $x_1, x_2, \dots, x_n$  be the positions of the masses  $m_1, m_2, \dots, m_n$ , and  $y$  the input motion.

By using the state variables  $z_1 = x_1 - y, z_2 = x_2 - x_1, \dots, z_n = x_n - x_{n-1}$ , the system equations are



**Fig. 7.4.** A one degree of freedom model with backslash.



**Fig. 7.5.** Model with  $n$  degrees of freedom with passive nonlinear parameters.

$$m_i \ddot{z}_i + \gamma_i(\dot{z}_1, \dot{z}_2, \dots, \dot{z}_n) + \eta_i(z_1, z_2, \dots, z_n) = f_i(t), \quad i = 1, \dots, n$$

$$f_i(t) = \begin{cases} -m_1 \ddot{y}(t) + \varphi_1(t), & i = 1 \\ \varphi_i(t), & i \neq 1 \end{cases}$$

where  $\varphi_i$  is the external force acting on the  $i$ -th mass  $m_i$  (if any), and the functions  $\gamma_i(\dot{z}_1, \dot{z}_2, \dots, \dot{z}_n)$  and  $\eta_i(z_1, z_2, \dots, z_n)$  represent the  $i$ -th nonlinear damping force and the  $i$ -th nonlinear elastic force respectively. Obviously,  $\gamma_i(\cdot)$  and  $\eta_i(\cdot)$  may have any expression. The dynamics of the system can be therefore written as

$$\ddot{z} + 2\delta_i \omega_{n_i} \dot{z}_i + \omega_{n_i}^2 z = -\omega_{n_i}^2 \zeta_i \quad i = 1, \dots, n$$

with

$$\zeta_i = \frac{1}{\omega_{n_i}^2} \left( \frac{\gamma_i + \eta_i - f_i}{m_i} - 2\delta_i \omega_{n_i} \dot{z}_i - \omega_{n_i}^2 z_i \right)$$

and  $\omega_{n_i} = \sqrt{k_i/m_i}$ ,  $\delta_i = d_i/(2m_i\omega_{n_i})$ , where  $k_i$  and  $d_i$  are the linear stiffness and damping coefficients connected with  $\eta_i$  and  $\gamma_i$ , [6].

## 7.2 Analysis of the Trajectories in the Time Domain

In order to show how the different motion profiles may generate different behaviors and responses in mechanical systems with elastic elements, and how different models of the same system may lead to different results, a simulation study of a mechanical device described with two different models is now reported and discussed. The two mass/spring/dumper models are characterized respectively by one and five degrees of freedom.

In the following discussion, ideal actuation and control systems are implicitly considered, and therefore the trajectory profile is supposed to be the input signals of the two models, i.e.

$$y(t) = q(t)$$

while the position  $x$  (or  $x_5$ ) of the mass  $m$  (or  $m_5$ ) is the output. In order to perform a direct comparison of the results, in all the simulations the desired displacement and the duration of the motion have the same values. In particular, the reference trajectory has been obtained with the following conditions

$$q_0 = 0, \quad q_1 = 15, \quad t_0 = 0, \quad t_1 = 30.$$

The parameters of the mechanical systems have been assumed as:

1. One degree of freedom system

$$m = 1, \quad d = 2, \quad k = 100 \quad \Rightarrow \quad \omega_n = 10, \quad \delta = 0.1.$$

2. Five degrees of freedom system

$$m_1 = m_2 = m_3 = m_4 = m_5 = 0.2$$

$$d_1 = d_2 = d_3 = d_4 = d_5 = 10$$

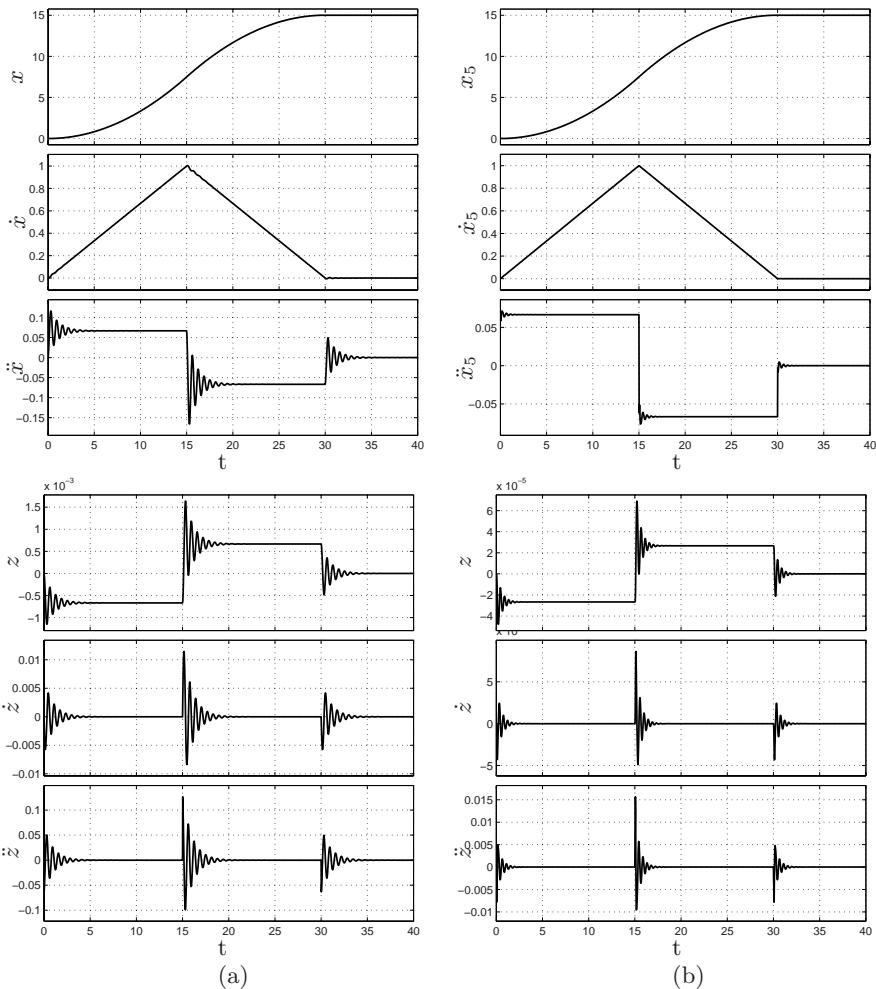
$$k_1 = k_2 = k_3 = k_4 = k_5 = 500.$$

Notice that with these choices the total mass of the system with five degrees of freedom is the same as in the one degree of freedom case ( $m = \sum_{i=1}^5 m_i$ ). Similarly, the total stiffness and the dissipation coefficient are the same for both systems.

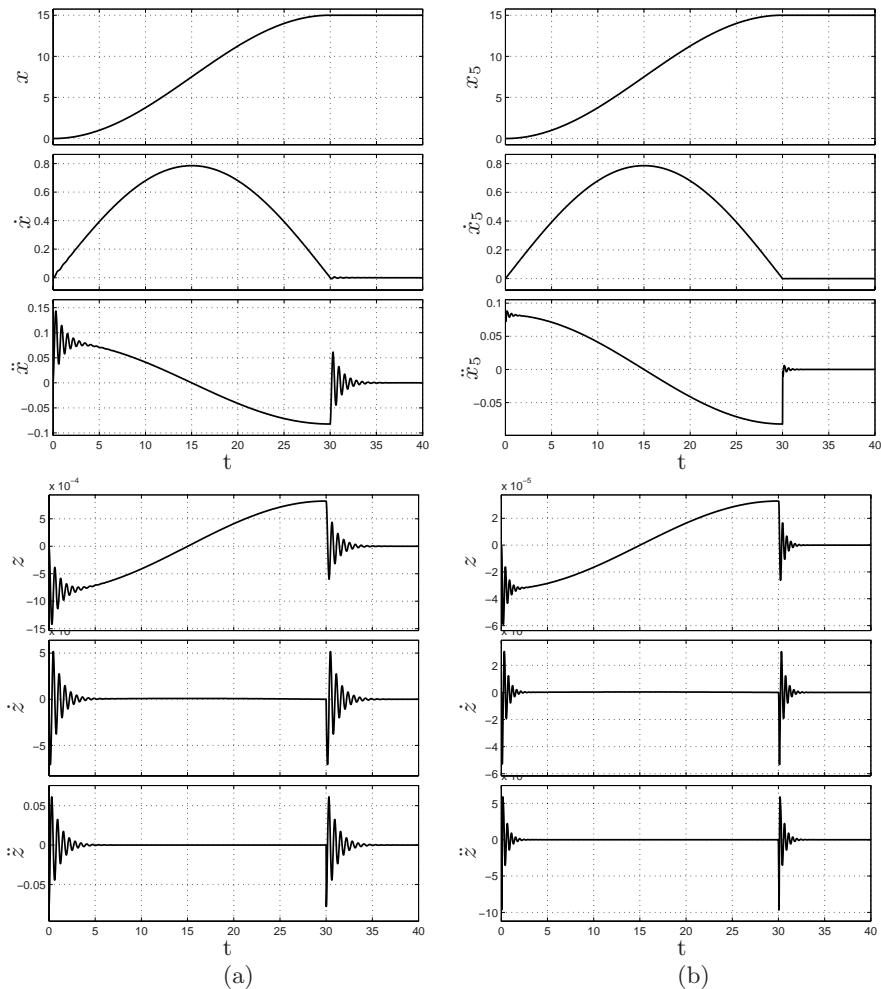
For each one of the considered trajectories, the profiles of position, velocity, and acceleration of the mass  $m$  for the one degree of freedom model (left diagrams) and of the mass  $m_5$  for the five degrees of freedom model (right diagrams) are reported in the following figures. The order of presentation is summarized in Table 7.1.

Figure n.	Trajectory
7.6	Constant acceleration
7.7	Harmonic
7.8	Cycloidal
7.9	Elliptic
7.10	Polynomial degree 3
7.11	Polynomial degree 5
7.12	Linear with circular blends
7.13	Trapezoidal
7.14	Modified trapezoidal
7.15	Gutman 1-3
7.16	Freudenstein 1-3
7.17	Freudenstein 1-3-5

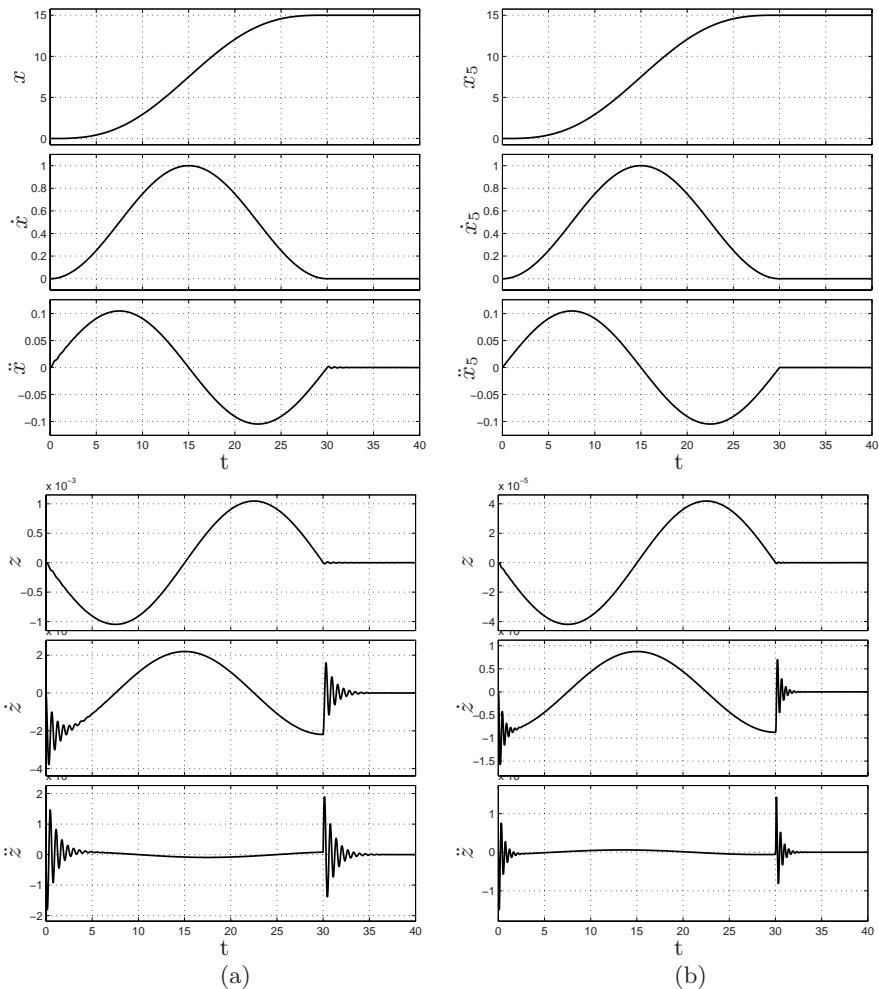
**Table 7.1.** Simulation of elastic systems.



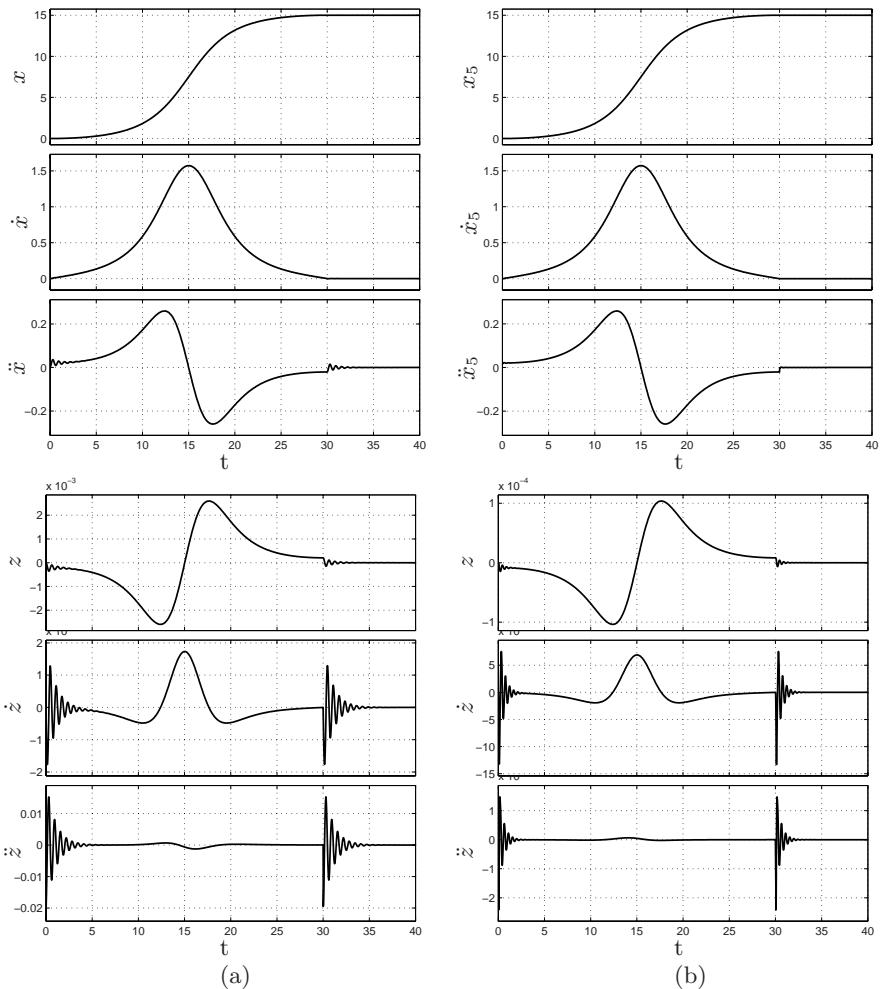
**Fig. 7.6.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a constant acceleration trajectory.



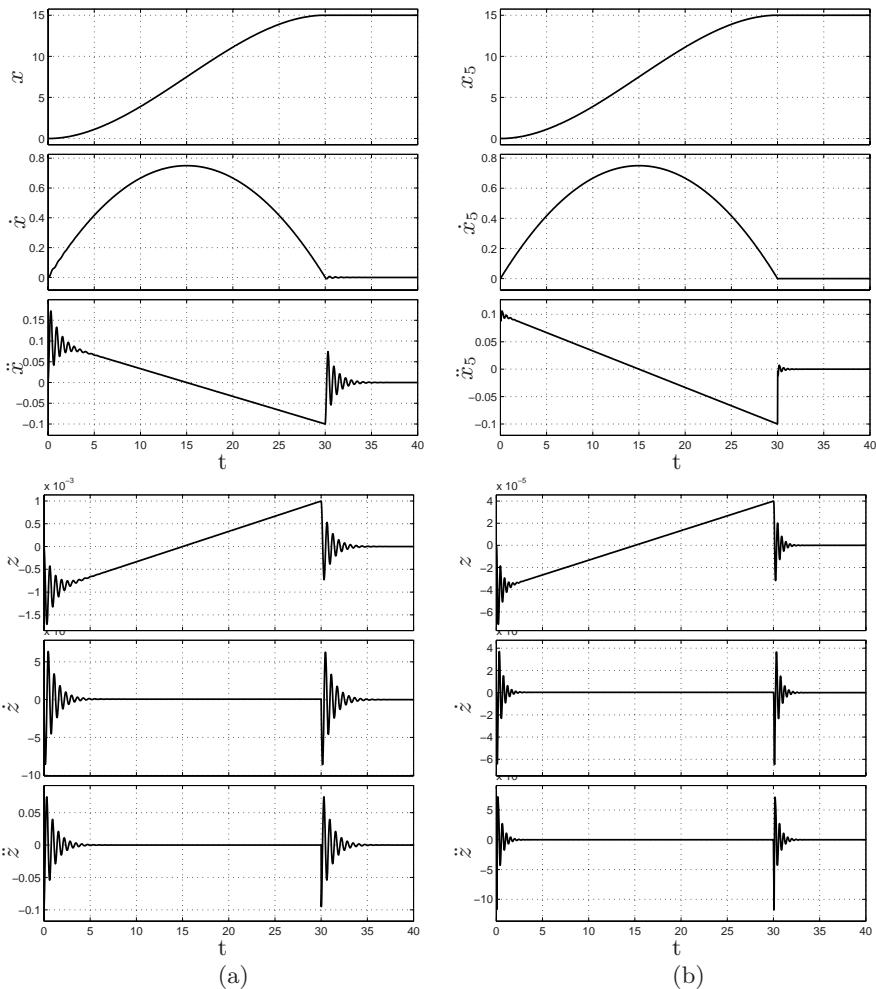
**Fig. 7.7.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a harmonic trajectory.



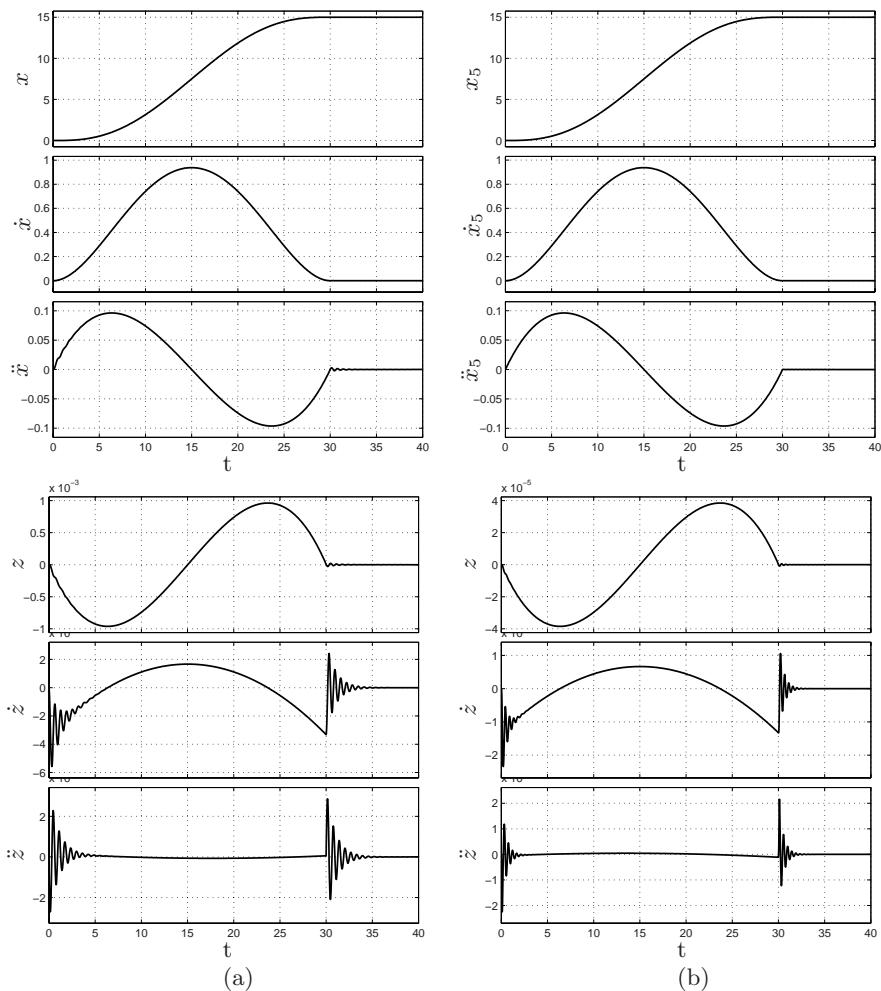
**Fig. 7.8.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a cycloidal trajectory.



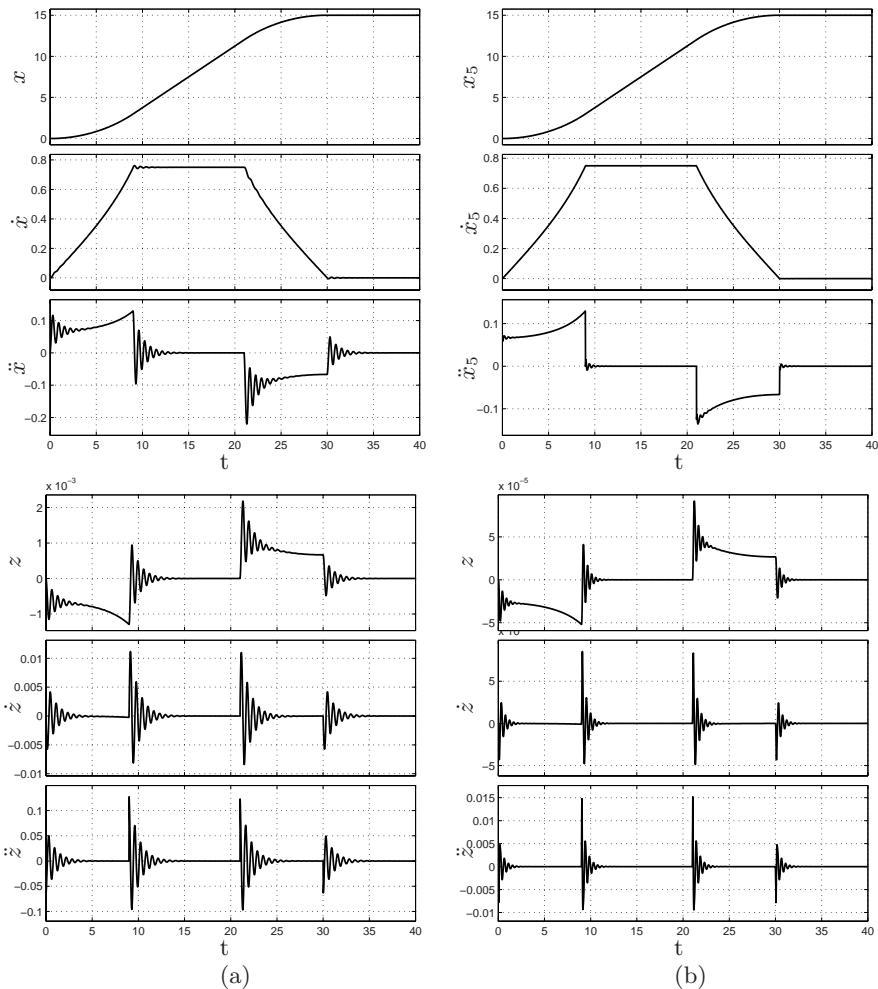
**Fig. 7.9.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is an elliptic trajectory ( $n = 2$ ).



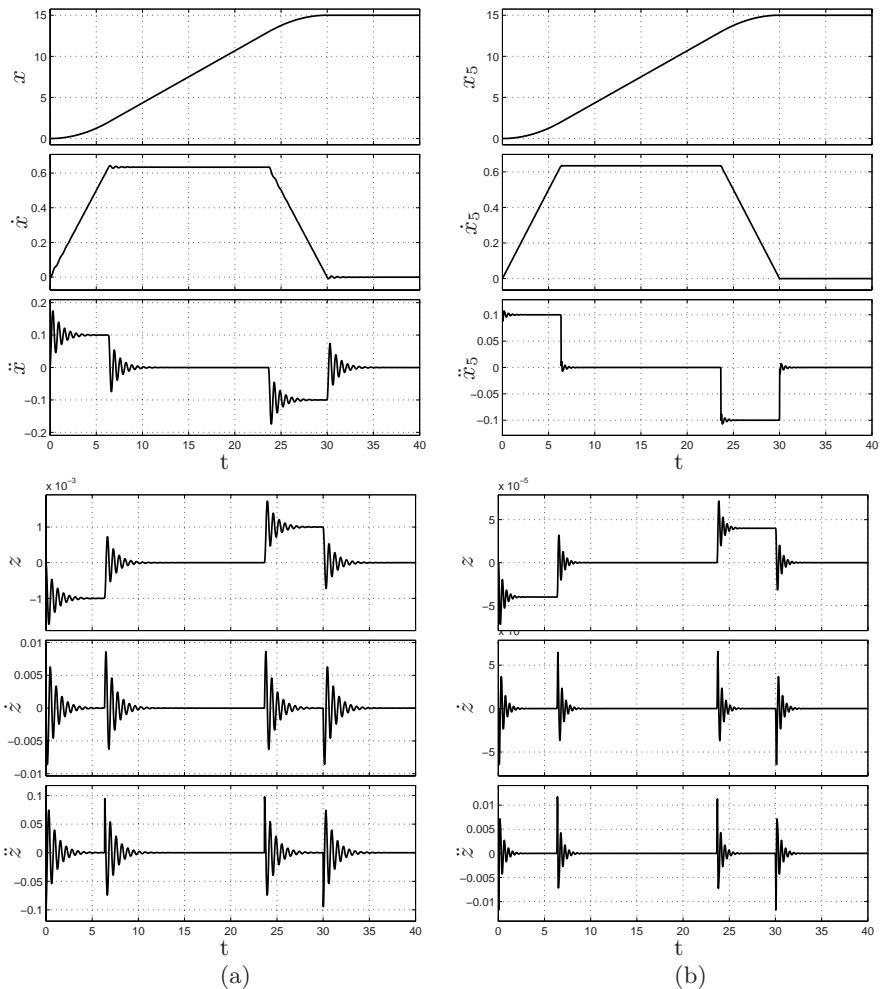
**Fig. 7.10.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a polynomial trajectory of degree three.



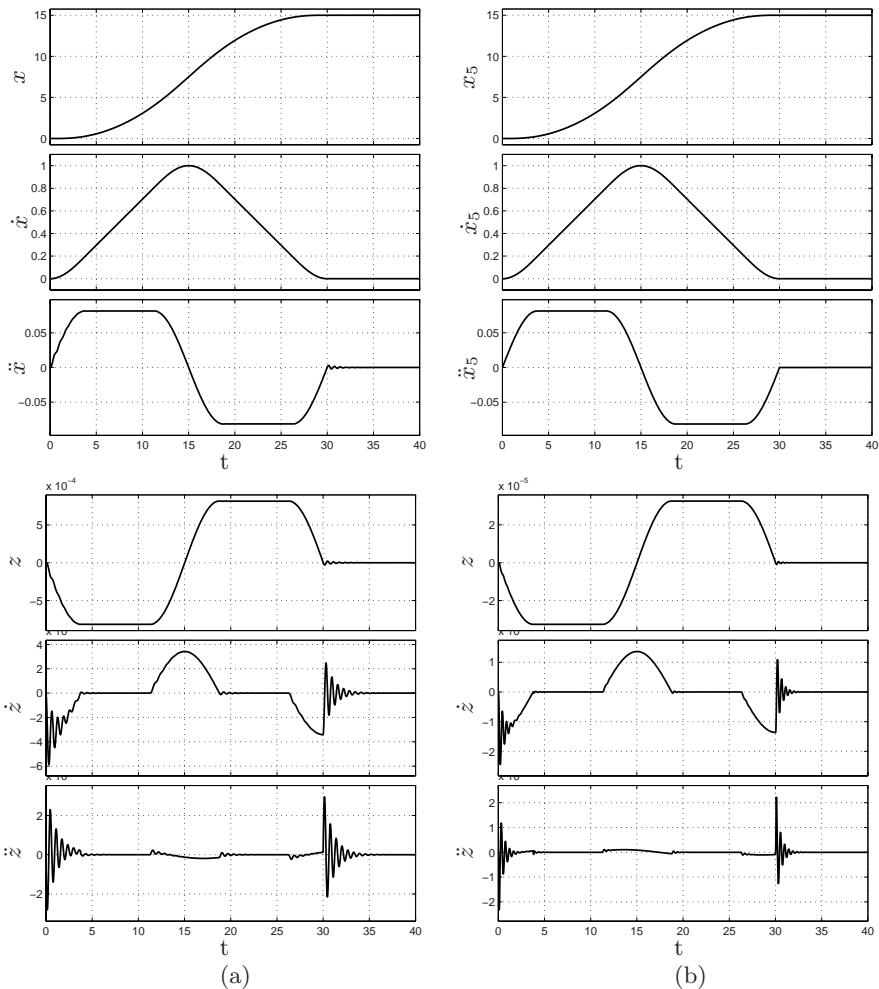
**Fig. 7.11.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a polynomial trajectory of degree five.



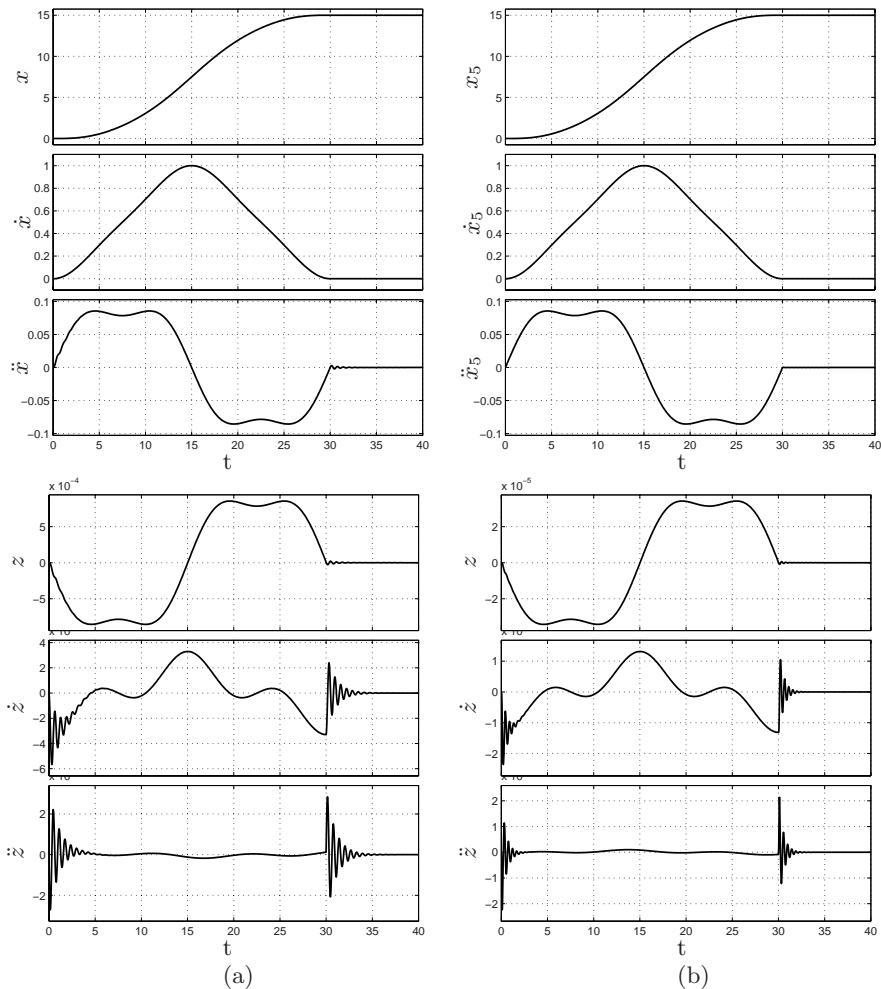
**Fig. 7.12.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a linear trajectory with circular blends.



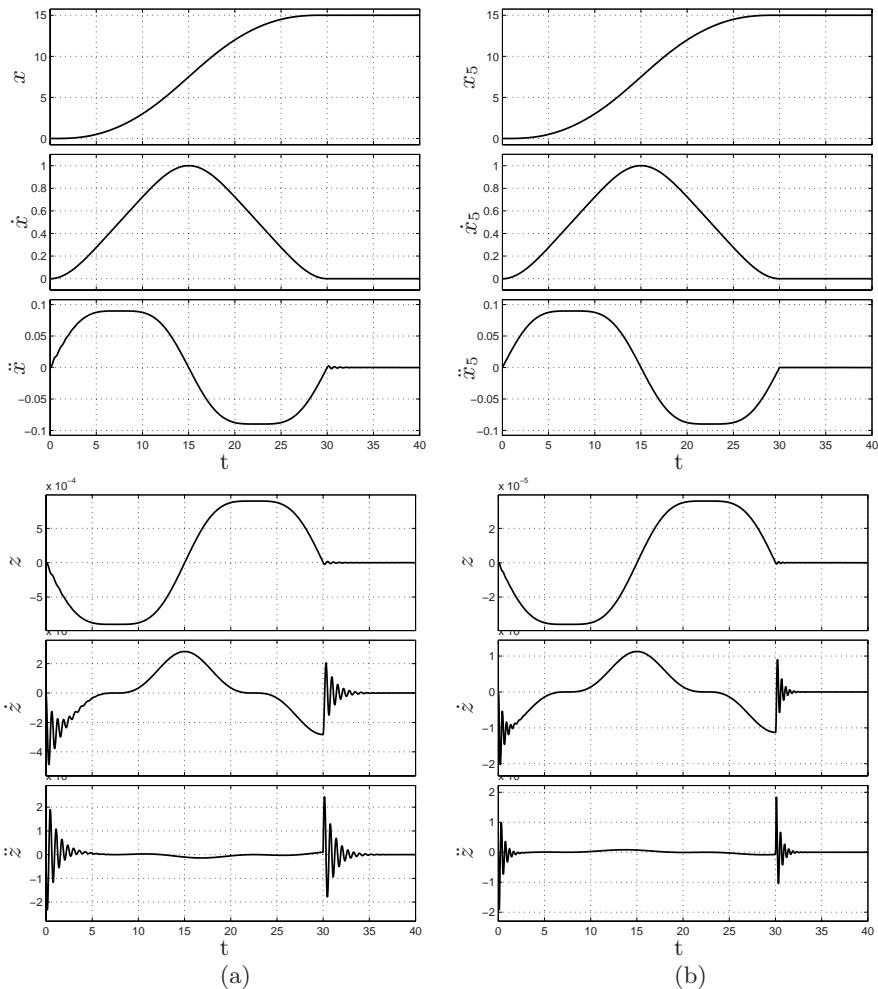
**Fig. 7.13.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a trapezoidal velocity trajectory.



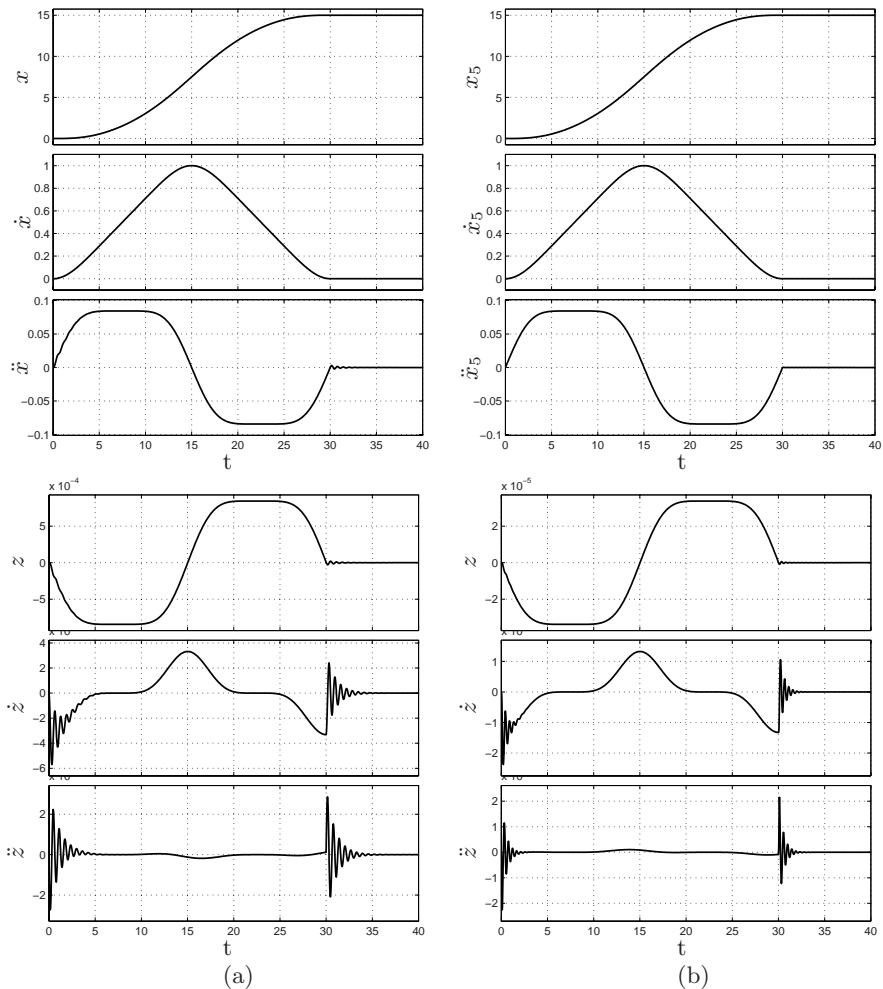
**Fig. 7.14.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a modified trapezoidal velocity trajectory.



**Fig. 7.15.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a Gutman 1-3 trajectory.



**Fig. 7.16.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a Freudenstein 1-3 trajectory.



**Fig. 7.17.** Position, velocity, acceleration and errors ( $z = x - q$ ) of mass  $m$  of the one degree of freedom system (a) and of the mass  $m_5$  of the system with five degrees of freedom (b). In this latter case  $z = x_5 - q$ . The input  $q(t)$  is a Freudenstein 1-3-5 trajectory.

### Some considerations

From the previous figures, it is possible to see that for all the trajectories the oscillations are more evident in the acceleration profiles than in the velocity and position ones. This is quite obvious, since the velocity and the position are obtained by integration of the acceleration and velocity signals respectively. Since in the frequency domain an integrator may be considered as a low-pass filter, the high-frequency oscillations are eliminated.

Another result that can be noticed in all the simulations is that oscillations are dumped more quickly and are located at higher frequencies in the case of the five degrees of freedom system.

Moreover, the simulations show another fact already mentioned and that will be discussed with more details in the following section: the trajectories with discontinuous acceleration profiles (constant acceleration, harmonic, elliptic, polynomial of degree three, linear with circular blends, linear with parabolic bends) generate larger oscillations with respect to trajectories with continuous acceleration profile (cycloidal, polynomial of degree five, modified trapezoidal, modified cycloidal, Gutman 1-3, Freudenstein 1-3, Freudenstein 1-3-5). The amplitude of the oscillations is proportional to the acceleration discontinuity.

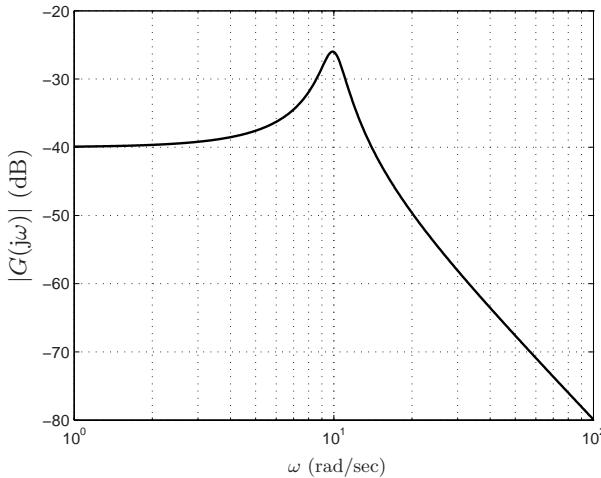
## 7.3 Analysis of the Trajectories in the Frequency Domain

A different approach for studying the trajectories from a dynamical point of view is based on the analysis of their frequency properties using the *Fourier Transform*.

A generic real function  $x(t)$  can be expressed as the “summation” of an infinite number of sinusoidal terms, each of them with frequency  $\omega$ , amplitude  $V(\omega)$  and phase  $\varphi(\omega)$  (see Appendix D for more details):

$$x(t) = \int_0^{+\infty} V(\omega) \cos[\omega t + \varphi(\omega)] d\omega.$$

By means of a frequency analysis of the acceleration profiles of the trajectories, it is possible to evaluate the stress that, in the ideal case, is applied to the mechanical structure by the actuation system during the motion. From this point of view, it is convenient to keep the harmonic content, i.e. the function  $V(\omega)$ , of the acceleration small at high frequencies. As a matter of fact, a negligible harmonic content at high frequencies implies a smoother motion profile in the time domain. If “fast” variations, or discontinuities in the limit case, are present in the acceleration profile, then  $V(\omega)$  presents relevant terms at high frequencies. In this case, the resonant frequencies of the mechanical



**Fig. 7.18.** Magnitude of the frequency response of the system (7.2).

structure could be excited and then vibrations generated. As a matter of fact the mechanical system acts like a filter which amplifies or reduces the magnitude of the different harmonics according to the value of its frequency response. In Fig. 7.18 the Bode diagram (magnitude) of the one-dof linear system represented by (7.2), with the parameters as in Sec. 7.2 (i.e.  $m = 1$ ,  $d = 2$ ,  $k = 100$ ), is reported<sup>1</sup>. In this simple case, in order to avoid vibrations (that is high values of the “error”  $z$ ) the maximum frequency of the adopted trajectory must be smaller than the resonance frequency  $\omega_r \approx 10$  (at which the magnitude of the frequency response has a peak).

The main trajectories presented in previous chapters, are now analyzed and compared with respect to their harmonic content (in particular, the acceleration profiles are considered). For some of them, the Fourier transform is presented in a closed form, while for others the spectra have been computed numerically (i.e. by means of the discrete Fourier transform). In any case, all the spectra shown in the figures of this section are obtained under the conditions of unitary displacement ( $h = q_1 - q_0 = 1$ ) and unitary duration ( $T = t_1 - t_0 = 1$ ).

---

<sup>1</sup> The transfer function of the system (7.2), providing the dynamic relation between the acceleration  $\ddot{y}(t)$  and the “error”  $z(t) = x(t) - y(t)$ , is

$$G(s) = \frac{Z(s)}{A(s)} = \frac{-1}{s^2 + d/m s + k/m}$$

being  $A(s)$  the Laplace transform of the acceleration and  $Z(s)$  the Laplace transform of  $z(t)$ .

### 7.3.1 Frequency spectrum of some elementary trajectories

#### Constant velocity trajectory

The acceleration profile of the constant velocity trajectory is represented by two impulses  $\delta(t)$  at the start and end points, being the velocity profile ( $\dot{q}(t) = h/T$ ,  $t \in [0, T]$ ) discontinuous:

$$\ddot{q}(t) = \frac{h}{T} \delta(t + T/2) - \frac{h}{T} \delta(t - T/2).$$

By transforming each term of  $\ddot{q}(t)$  (note that the transform of the impulse  $\delta(t)$  is the unitary constant) the Fourier transform can be easily deduced:

$$\begin{aligned} A(\omega) &= \frac{h}{T} e^{j\omega T/2} - \frac{h}{T} e^{-j\omega T/2} \\ &= 2j \frac{h}{T} \sin\left(\omega \frac{T}{2}\right). \end{aligned}$$

Therefore the amplitude  $V(\omega) = \frac{|A(\omega)|}{\pi}$  for the constant velocity trajectory is

$$V(\omega) = \frac{2h}{\pi T} \left| \sin\left(\omega \frac{T}{2}\right) \right|.$$

In order to compare different kinds of trajectories with the same values of  $T$  and  $h$ , it is convenient to express  $V$  as a function of a dimensionless variable  $\Omega$  defined as

$$\Omega = \frac{\omega}{\omega_0} \quad \text{with} \quad \omega_0 = \frac{2\pi}{T}. \quad (7.5)$$

Therefore, the function describing the harmonic content of the acceleration profile of a given trajectory is

$$V'(\Omega) = V(\omega) \Big|_{\omega = \omega_0 \Omega}. \quad (7.6)$$

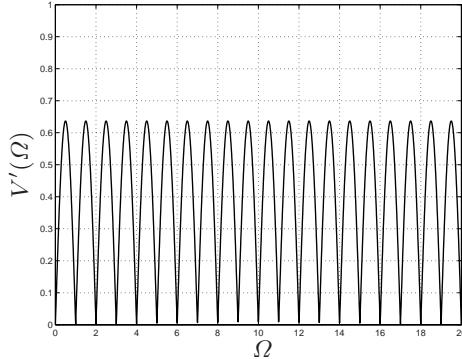
For the constant velocity trajectory,  $V'(\Omega)$  assumes the expressions

$$V'(\Omega) = \frac{h}{T} \frac{2}{\pi} |\sin(\pi \Omega)|.$$

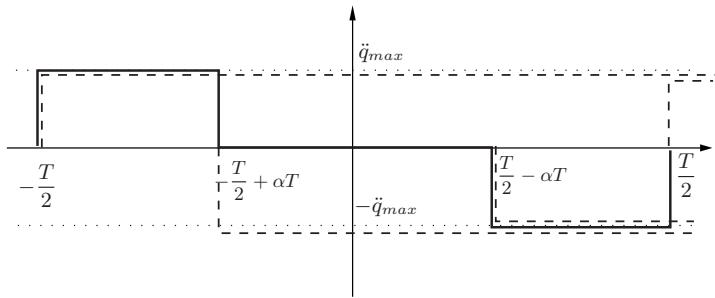
The spectrum  $V'(\Omega)$  is reported in Fig. 7.19. Note that the maximum amplitude of the lobes composing the spectrum remains equal for increasing values of the frequency  $\Omega$ .

#### Trapezoidal velocity trajectory

As in Sec. 6.2.1, the trapezoidal trajectory is defined by assuming that the acceleration period is a fraction  $\alpha$  of the total time-length:  $T_a = \alpha T$ . Accordingly, the maximum (constant) acceleration is  $\ddot{q}_{max} = \frac{h}{\alpha(1-\alpha)T^2}$ . In this



**Fig. 7.19.** Harmonic content of the constant velocity trajectory.



**Fig. 7.20.** Decomposition of the acceleration profile of the trapezoidal trajectory into elementary functions (step functions).

case, the acceleration can be represented as a sequence of steps of amplitude  $\ddot{q}_{max}$ , see Fig. 7.20:

$$\ddot{q}(t) = \ddot{q}_{max} \left( h(t + T/2) - h(t + T/2 - \alpha T) - h(t - T/2 + \alpha T) + h(t - T/2) \right)$$

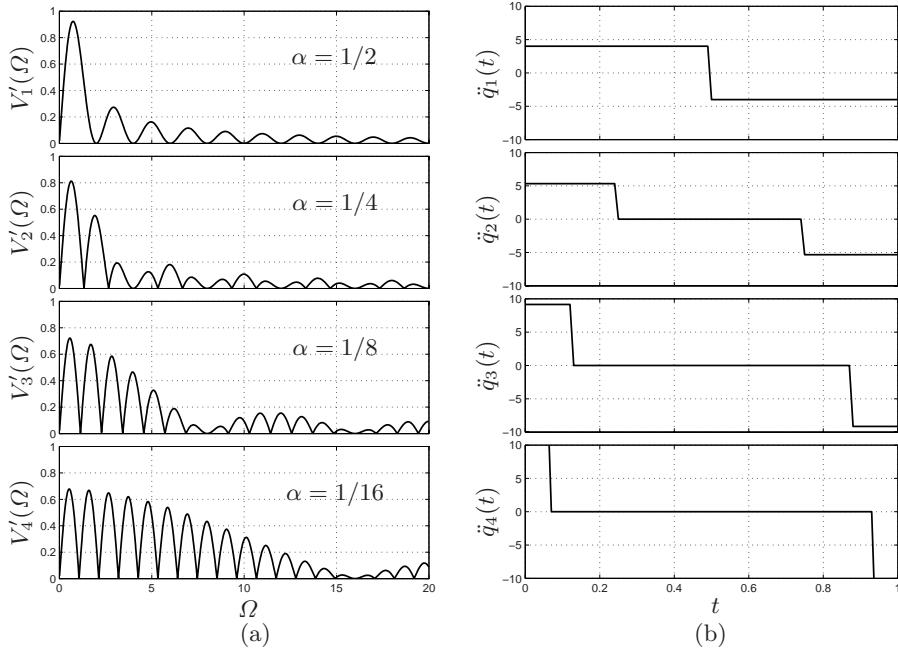
where  $h(t)$  is the Heaviside step function. After transforming each term of  $\ddot{q}(t)$ , and after some algebraic manipulations exploiting the linearity of the Fourier transform, one obtains

$$A(\omega) = 4j \frac{h}{(1-\alpha)\alpha T^2 \omega} \sin\left((1-\alpha)\omega \frac{T}{2}\right) \sin\left(\alpha\omega \frac{T}{2}\right).$$

The magnitude spectrum is therefore

$$V'(\Omega) = \frac{h}{T} \frac{2}{(1-\alpha)\alpha\pi^2\Omega} \left| \sin\left((1-\alpha)\pi\Omega\right) \sin\left(\alpha\pi\Omega\right) \right|$$

where  $\Omega$  is defined in (7.5).

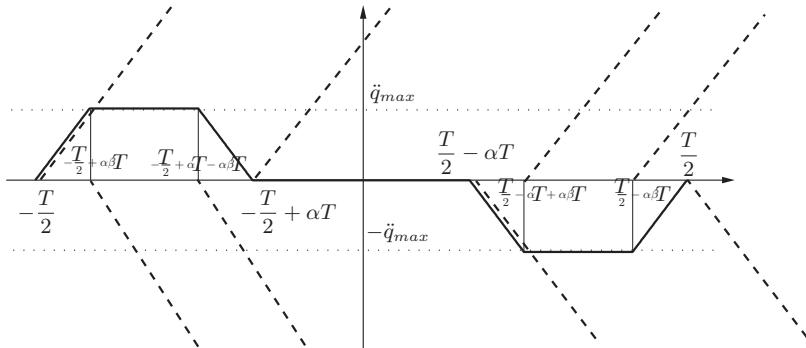


**Fig. 7.21.** Harmonic content (a) and related acceleration profiles (b) of trapezoidal trajectories for different values of the parameter  $\alpha$ .

The spectra of some trapezoidal trajectories, obtained for different values of  $\alpha$ , are reported in Fig. 7.21. If the value of  $\alpha$  decreases, and accordingly the maximum acceleration value grows, the bandwidth of the amplitude spectrum  $V'(\Omega)$  widens. The limit case, for  $\alpha \rightarrow 0$ , is the spectrum of the constant velocity trajectory<sup>2</sup> (characterized by an impulsive acceleration).

<sup>2</sup> It can be easily shown that the acceleration spectrum of the trapezoidal velocity trajectory converges to the spectrum of the constant velocity as  $\alpha$  goes to zero:

$$\begin{aligned} \lim_{\alpha \rightarrow 0} V'_{tr}(\Omega, \alpha) &= \lim_{\alpha \rightarrow 0} \left\{ \frac{h}{T} \frac{2}{(1-\alpha)\pi} \left| \sin((1-\alpha)\pi\Omega) \frac{\sin(\alpha\pi\Omega)}{\alpha\pi\Omega} \right| \right\} = \\ &= \frac{h}{T} \frac{2}{\pi} \left| \sin(\pi\Omega) \right| = V'_{cv}(\Omega). \end{aligned}$$



**Fig. 7.22.** Decomposition of the acceleration profile of the double S trajectory into elementary functions (ramp functions).

### Double S trajectory

The acceleration profile of the double S trajectory is expressed in terms of the parameters  $\alpha$  and  $\beta$ , which define the ratios between the total duration  $T$  of the trajectory, the acceleration period  $T_a$  and the duration of the constant jerk phase  $T_j$ :

$$\alpha = \frac{T_a}{T}, \quad \beta = \frac{T_j}{T_a}.$$

This leads to the following expression

$$\ddot{q}(t) = q_{max}^{(3)} \left( r(t + T/2) - r(t + T/2 - \alpha\beta T) - r(t + T/2 - \alpha T + \alpha\beta T) + r(t + T/2 - \alpha T) - r(t - T/2 + \alpha T) + r(t - T/2 + \alpha T - \alpha\beta T) + r(t - T/2 + \alpha\beta T) - r(t - T/2) \right)$$

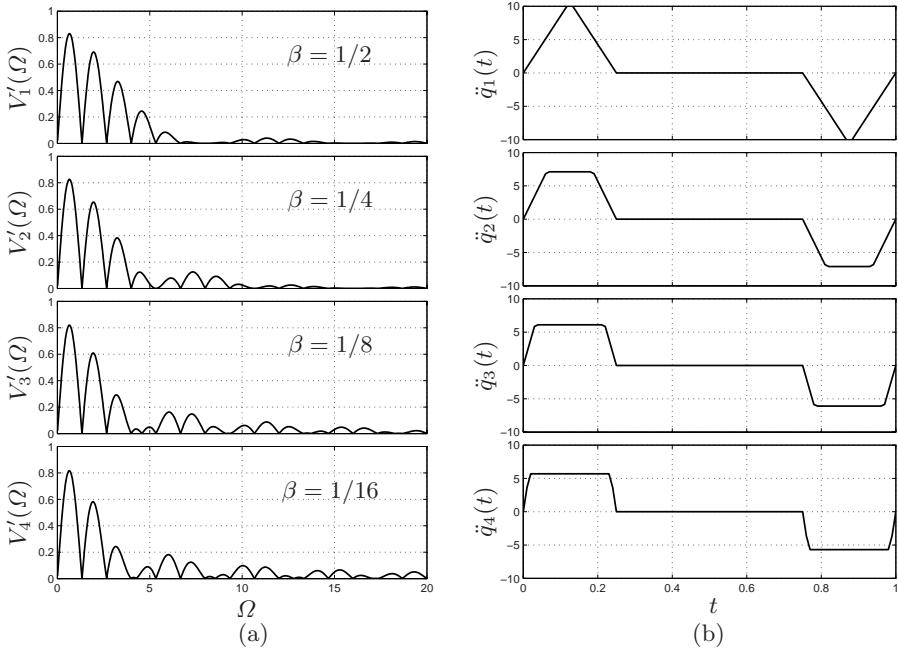
where  $r(t)$  is the ramp function defined by  $r(t) = t h(t)$  (see Fig. 7.22), and  $q_{max}^{(3)}$  is the peak value of the jerk, defined in (3.42). By means of a Fourier transform of each term of  $\ddot{q}(t)$ , after some algebraic manipulations, one obtains

$$A(\omega) = 8j \frac{h}{\alpha^2 \beta (1-\alpha)(1-\beta) T^3 \omega^2} \sin\left((1-\alpha)\omega \frac{T}{2}\right) \sin\left(\alpha(1-\beta)\omega \frac{T}{2}\right) \sin\left(\alpha\beta\omega \frac{T}{2}\right).$$

The magnitude spectrum is therefore

$$V'(\Omega) = \frac{h}{T} \frac{2}{\alpha^2 \beta (1-\alpha)(1-\beta) \pi^3 \Omega^2} \left| \sin\left((1-\alpha)\pi\Omega\right) \sin\left(\alpha(1-\beta)\pi\Omega\right) \sin\left(\alpha\beta\pi\Omega\right) \right|.$$

The shape of  $V'(\Omega)$  depends on  $\alpha$  in a manner similar to the one of the trapezoidal trajectory (harmonic components at high frequencies have larger magnitudes for small values of  $\alpha$ ), but in this case it is also possible to act on  $\beta$  in order to properly reshape the spectrum.



**Fig. 7.23.** Harmonic content (a) and related acceleration profiles (b) of double S trajectories for different values of the parameter  $\beta$  ( $\alpha$  is constant and equal to 1/4).

As reported in Fig. 7.23, by increasing  $\beta$  up to its maximum value (1/2) the harmonic components at high frequencies are reduced. Conversely for  $\beta \approx 0$ , the spectrum is the same of the trapezoidal trajectory<sup>3</sup>.

For many trajectories, the simplest way to compute the Fourier transform of the acceleration profile is to apply the definition (D.1) to its expression. In particular, it is convenient to adopt the normalized form of the trajectory  $q_N(\tau)$ , whose second derivative ( $\ddot{q}_N(\tau)$ ,  $\tau \in [0, 1]$ ) is related to the acceleration of the corresponding trajectory  $q(t)$ , with  $t \in [-\frac{T}{2}, \frac{T}{2}]$  and displacement  $h$ , through

$$\ddot{q}(t) = \frac{h}{T^2} \ddot{q}_N\left(\frac{t}{T} - \frac{1}{2}\right), \quad \text{with} \quad t \in \left[-\frac{T}{2}, \frac{T}{2}\right].$$

Therefore, the Fourier transform  $A(\omega)$  of  $\ddot{q}(t)$  can be obtained from the transform  $A_N(\omega)$  of  $\ddot{q}_N(\tau)$  by means of

<sup>3</sup> More formally

$$\begin{aligned} \lim_{\beta \rightarrow 0} V'_{ss}(\Omega, \alpha, \beta) &= \frac{h}{T} \frac{2}{\alpha(1-\alpha)(1-\beta)\pi^2\Omega} \left| \sin((1-\alpha)\pi\Omega) \sin(\alpha(1-\beta)\pi\Omega) \frac{\sin(\alpha\beta\pi\Omega)}{\alpha\beta\pi\Omega} \right| \\ &= \frac{h}{T} \frac{2}{(1-\alpha)\alpha\pi^2\Omega} \left| \sin((1-\alpha)\pi\Omega) \sin(\alpha\pi\Omega) \right| = V'_{tr}(\Omega, \alpha). \end{aligned}$$

$$A(\omega) = \frac{h}{T} A_N(\omega T) e^{j\frac{\omega}{2}}.$$

In particular, it is possible to relate the magnitude spectrum of  $\ddot{q}(t)$  to the one of  $\ddot{q}_N(\tau)$ :

$$V'(\Omega) = \frac{h}{T} V'_N(\Omega), \quad \Omega = \frac{\omega}{2\pi/T} \quad (7.7)$$

where the spectrum of the normalized trajectory is

$$V'_N(\Omega) = \frac{|A_N(\omega)|}{\pi} \Big|_{\omega = 2\pi\Omega}. \quad (7.8)$$

### Polynomial trajectories

The acceleration profile of the polynomial trajectory of degree 3 has the normalized expression

$$\ddot{q}_N(\tau) = 6 - 12\tau$$

whose Fourier transform is

$$A_N(\omega) = \frac{12j}{\omega^2} e^{-j\frac{\omega}{2}} (2 - j\omega) \sin\left(\omega \frac{T}{2}\right)$$

which provides the magnitude spectrum

$$V'(\Omega) = \frac{h}{T} V'_N(\Omega) = \frac{h}{T} \frac{6}{\pi^3 \Omega^2} \sqrt{1 + \pi^2 \Omega^2} |\sin(\pi\Omega)|.$$

In the case of the 5-th degree polynomial trajectory, the acceleration has the normalized form

$$\ddot{q}_N(\tau) = 60\tau - 180\tau^2 + 120\tau^3$$

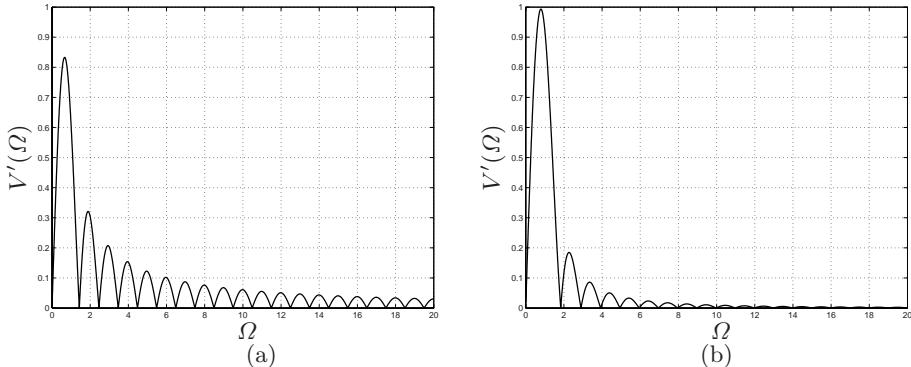
leading to the Fourier transform

$$A_N(\omega) = \frac{120j}{\omega^4} e^{-j\frac{\omega}{2}} (12 - \omega^2 + j6\omega) \sin\left(\omega \frac{T}{2}\right)$$

and to the magnitude spectrum

$$V'(\Omega) = \frac{h}{T} V'_N(\Omega) = \frac{h}{T} \frac{30}{\pi^5 \Omega^4} \sqrt{9 + 4\pi^2 \Omega^2 + \pi^4 \Omega^4} |\sin(\pi\Omega)|.$$

The harmonic contents of the polynomials of degree 3 and degree 5 are compared in Fig. 7.24. Note that, although the main lobe of the 5-th degree polynomial has a peak value larger than the one of the 3-rd degree polynomial,



**Fig. 7.24.** Harmonic content of the 3-rd degree (a) and 5-th degree (b) polynomial trajectories.

the side lobes are considerably smaller. Therefore the polynomial of order 5, with its continuous acceleration, is preferable when high frequencies must not be excited.

### Harmonic trajectory

The expression of the acceleration profile of harmonic trajectory in normal form is

$$\ddot{q}_N(\tau) = \frac{\pi^2}{2} \cos \pi \tau$$

and, accordingly, the Fourier transform results

$$A_N(\omega) = \frac{j\pi^2 \omega}{\pi^2 - \omega^2} e^{-j\frac{\omega}{2}} \cos\left(\frac{\omega}{2}\right).$$

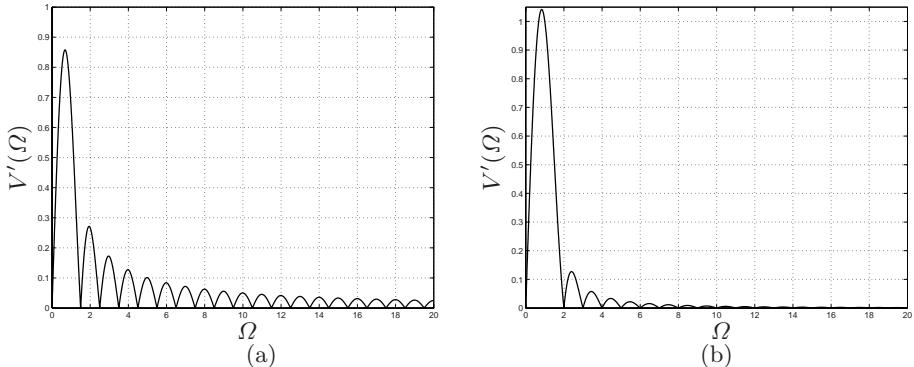
Therefore, the magnitude spectrum, depicted in Fig. 7.25(a), is

$$V'(\Omega) = \frac{h}{T} V'_N(\Omega) = \frac{h}{T} \frac{2\Omega}{1 - 4\Omega^2} |\cos(\pi\Omega)|.$$

### Cycloidal trajectory

The acceleration of the normalized cycloidal trajectory is

$$\ddot{q}_N(\tau) = 2\pi \sin 2\pi \tau$$



**Fig. 7.25.** Magnitude spectra of the harmonic (a) and cycloidal (b) trajectories.

which leads to the Fourier transform

$$A_N(\omega) = \frac{8j\pi^2}{(4\pi^2 - \omega^2)} e^{-j\frac{\omega}{2}} \sin\left(\frac{\omega}{2}\right)$$

and to the amplitude spectrum

$$V'(\Omega) = \frac{h}{T} V'_N(\Omega) = \frac{h}{T} \frac{2}{\pi(1 - \Omega^2)} |\sin(\pi\Omega)|.$$

In Fig. 7.25 the spectra of cycloidal and of harmonic trajectories are compared. Note that, despite the similar expressions of the two motion profiles (both based on trigonometric functions), the spectra are considerably different. In particular, the cycloidal trajectory is more suitable for those applications in which accelerations with high frequency components are not allowed.

### 7.3.2 Numerical computation of the frequency spectrum of generic trajectories

In many occasions, the Fourier transform of a given motion profile cannot be obtained in a closed form, mainly because of complexity reasons (e.g. when multi-segments trajectories are considered). In these cases numerical methods can help in analyzing the harmonic content of the trajectory. As a matter of fact, the Fourier transform of a generic function  $x(t)$  at the discrete frequency values  $k\Delta\omega$ , with  $\Delta\omega = \frac{2\pi}{T}$ , where  $T$  is the duration, is related to the discrete Fourier transform  $X_k$  of the sequence obtained by sampling the signal  $x(t)$  with the time period  $T_s$  by means of

$$X(k\Delta\omega) = T_s X_k. \quad (7.9)$$

For more details see Sec. D.4. Since the frequency period  $\Delta\omega$  is inversely proportional to  $T$ , it is possible to change such a period by modifying  $T$ . In particular one can augment the trajectory duration by adding to the sequence  $x_n = x(nT_s)$  an arbitrary number of zeros (this technique is known as *zero padding*). As a consequence the DFT can be compute for an arbitrary  $\Delta\omega$  (which de facto does not depend on the trajectory duration).

The magnitude spectrum can be then obtained

$$V(k\Delta\omega) = \frac{|X(k\Delta\omega)|}{\pi} = \frac{|T_s X_k|}{\pi}.$$

As for continuous transforms, it is convenient to express  $V$  as a function of a dimensionless variable

$$\Omega_k = k\Delta\Omega = k \frac{\Delta\omega}{\omega_0}.$$

Moreover, since the relations (7.7) and (7.8) are still valid<sup>4</sup>, it is convenient to consider unitary displacements in a unitary duration. This allows to easily compare different kinds of trajectories. For those motion laws which require several via-points (orthogonal polynomials, splines), the following data have been assumed

$$\begin{aligned} t_0 &= 0, & t_1 &= 0.3, & t_2 &= 0.6, & t_3 &= 1, \\ q_0 &= 0, & q_1 &= 0.2, & q_2 &= 0.7, & q_3 &= 1. \end{aligned}$$

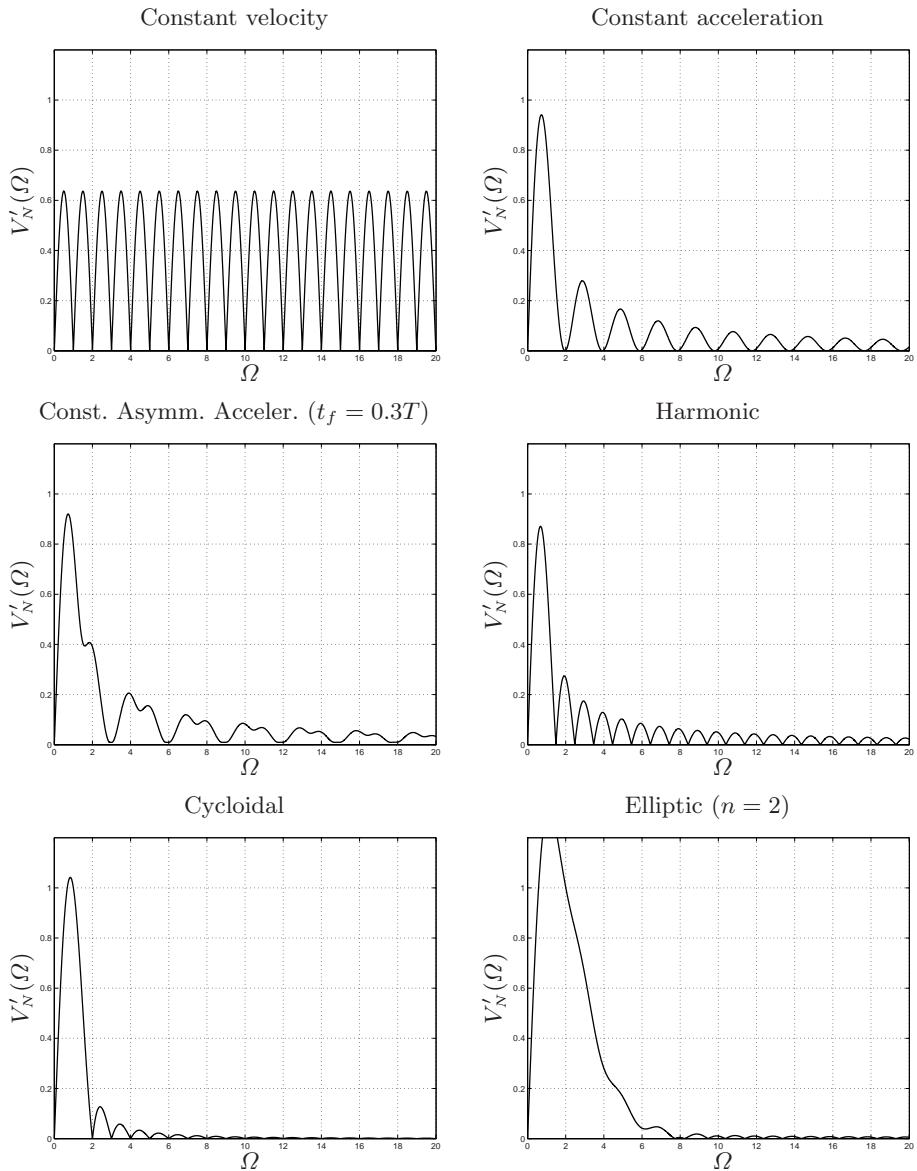
In Fig. 7.26-7.28, the spectra of the acceleration profiles of many of the trajectories illustrated in the previous chapters are shown.

---

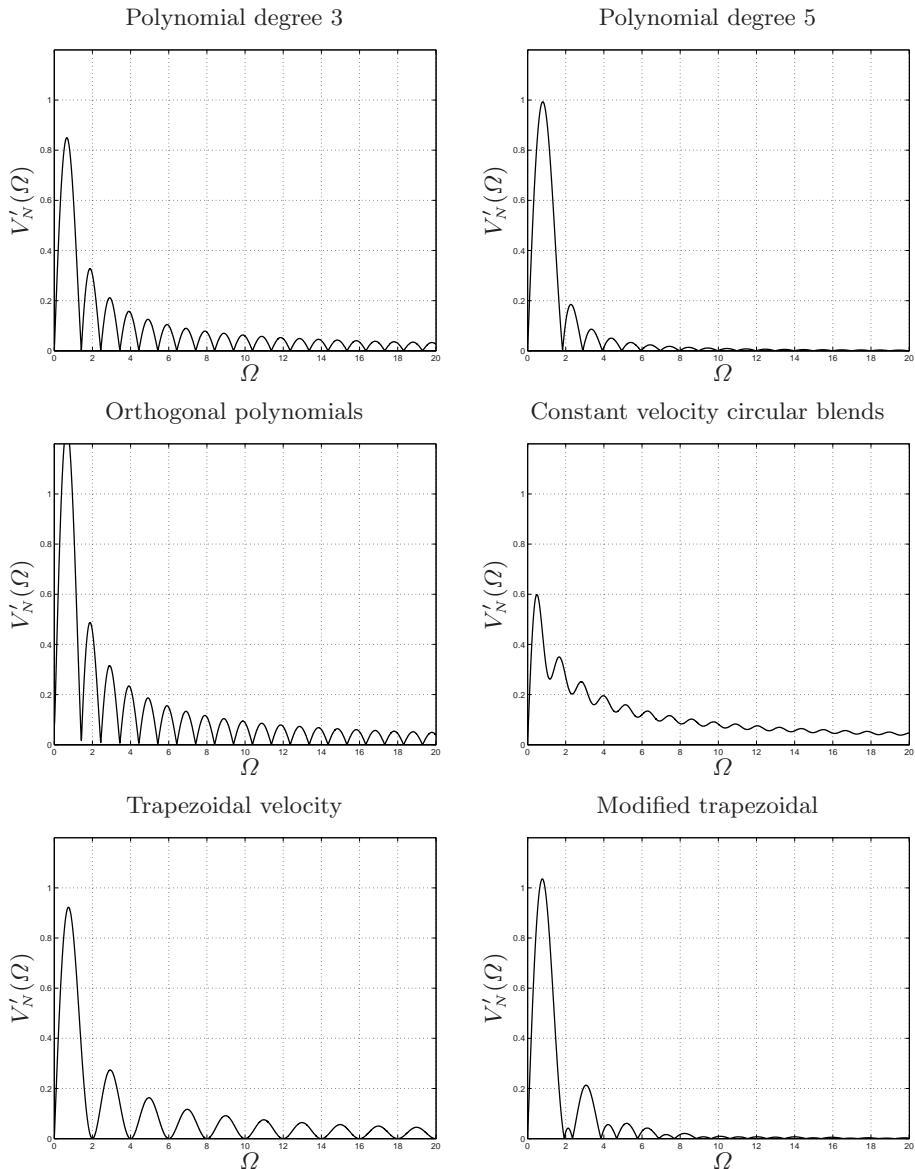
<sup>4</sup> In this case, the magnitude spectrum of the acceleration of a given trajectory  $\ddot{q}(t)$ , sampled at time instants  $nT_s$ , is related to the one of the corresponding normalized trajectory by

$$V'(k\Delta\Omega) = \frac{h}{T} V'_N(k\Delta\Omega) = \frac{h}{T} \frac{|A_N(k\Delta\omega)|}{\pi} \Big|_{\Delta\omega = 2\pi\Delta\Omega} = \frac{h}{T} \frac{|T_s A_{N_k}|}{\pi}$$

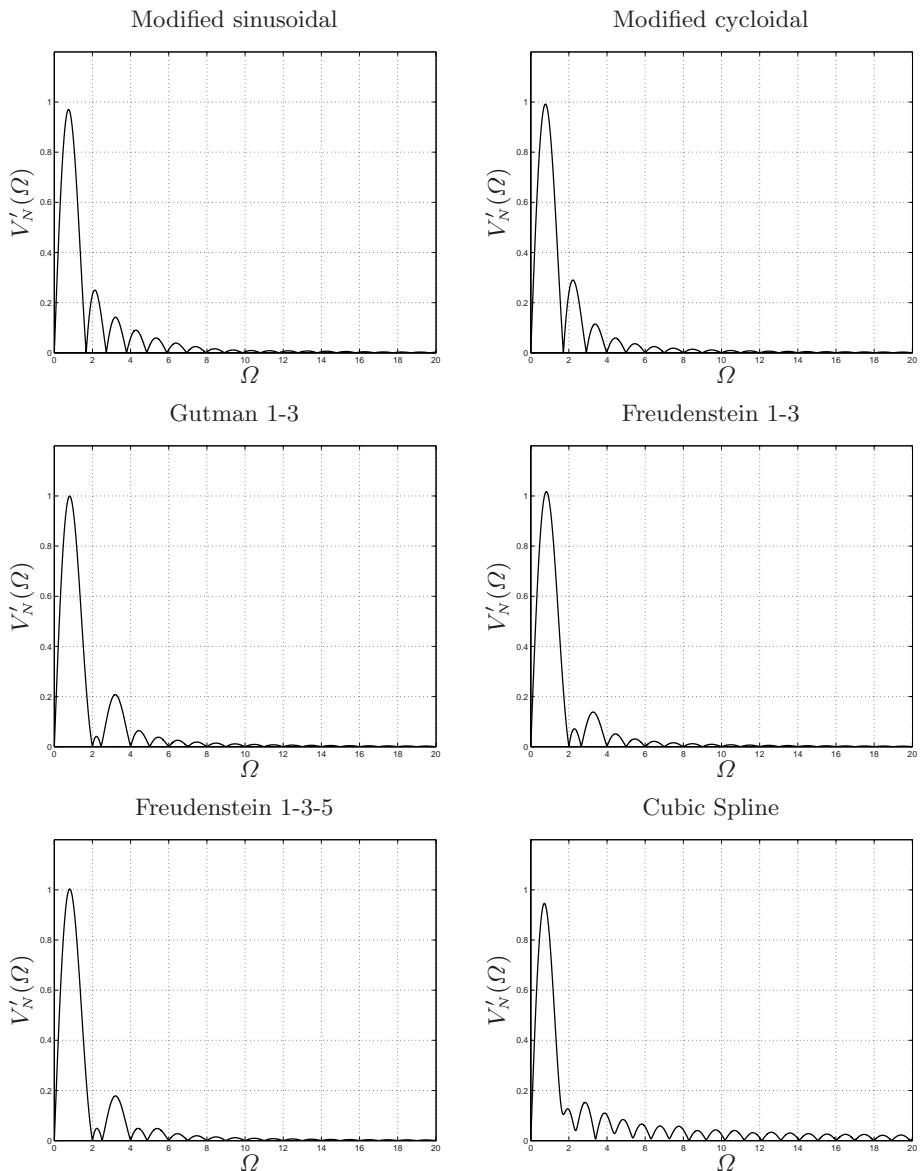
where  $A_{N_k}$  is the DFT of the sequence  $\ddot{q}_{N_n} = \ddot{q}_N(nT_s)$ .



**Fig. 7.26.** Frequency analysis of the acceleration profiles for the trajectories: constant velocity, constant acceleration, asymmetric constant acceleration, harmonic, cycloidal and elliptic.



**Fig. 7.27.** Frequency analysis of the acceleration profiles for the trajectories: polynomials of degree 3 and 5, orthogonal polynomials, constant velocity (linear) with circular blends, trapezoidal, modified trapezoidal.



**Fig. 7.28.** Frequency analysis of the acceleration profiles for the trajectories: modified sinusoidal, modified cycloidal, Gutman 1-3, Freudenstein 1-3 and 1-3-5, cubic spline.

From these figures, it is possible to appreciate that the cycloidal trajectory has the lowest harmonic content, see Fig. 7.26. A low harmonic content characterizes also for the 5-th degree polynomial, (Fig. 7.27), the modified cycloidal, and the Gutman 1-3, Freudenstein 1-3, and Freudenstein 1-3-5 (Fig. 7.28) trajectories. Among the functions considered in the analysis, besides the constant velocity trajectory, never used for practical applications, the worst profile is the one based on the orthogonal polynomials. The remaining profiles have intermediate features. Notice that at low frequencies all the profiles present a peak: the maximum value of these peaks is obtained for the elliptic trajectory.

### 7.3.3 Harmonic content of periodic trajectories

In typical industrial applications it is required that the actuators repeat continuously the same trajectory. Therefore, the motion law (and accordingly the related acceleration) consists of the periodic repetition of a basic function of duration  $T$ , i.e. (with reference with the acceleration profile)

$$\ddot{\tilde{q}}(t) = \sum_{k=-\infty}^{\infty} \ddot{q}(t - kT).$$

Since  $\ddot{\tilde{q}}(t)$  is a periodic function, its harmonic content may be analyzed by means of the Fourier series expansion, which decomposes  $\ddot{\tilde{q}}(t)$  in an infinite number of harmonic functions at frequencies multiple of  $\omega_0 = 2\pi/T$ , and with amplitude  $v_k$  and phase  $\varphi_k$ :

$$\ddot{\tilde{q}}(t) = v_0 + \sum_{k=1}^{\infty} v_k \cos(k\omega_0 t + \varphi_k), \quad \omega_0 = \frac{2\pi}{T}.$$

For more details see Sec. D.2. Because of the relation between  $v_k$  and the Fourier transform of  $\ddot{q}(t)$  (see Sec. D.2), it is possible to obtain the amplitude spectrum of the periodic function from the amplitude spectrum  $V(\omega)$  of the corresponding aperiodic motion profile:

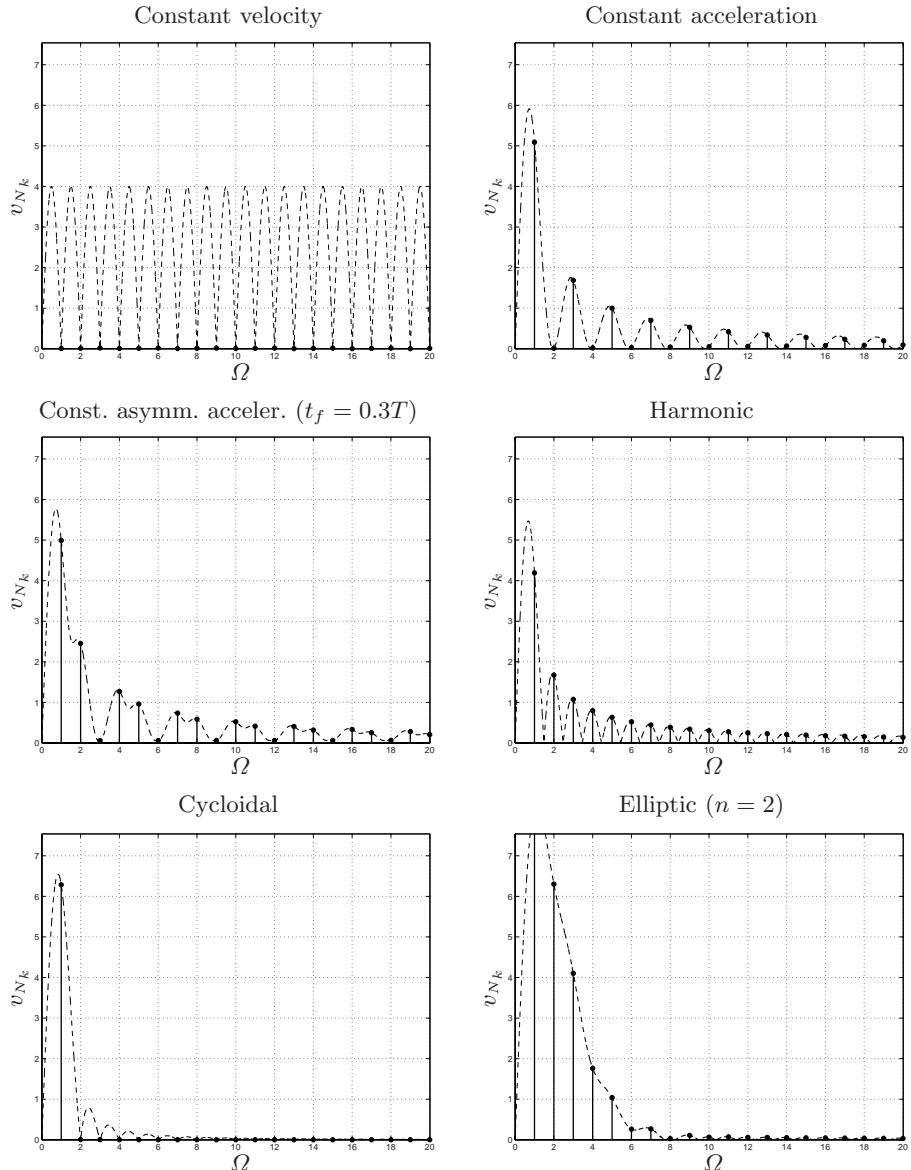
$$v_k = \frac{2\pi}{T} V(k\omega_0), \quad \omega_0 = \frac{2\pi}{T}, \quad k = 1, 2, \dots$$

Moreover, by considering the dimensionless variable  $\Omega = \omega/\omega_0$ , it possible to write (see (7.6))

$$v_k = \frac{2\pi}{T} V'(k), \quad k = 1, 2, \dots$$

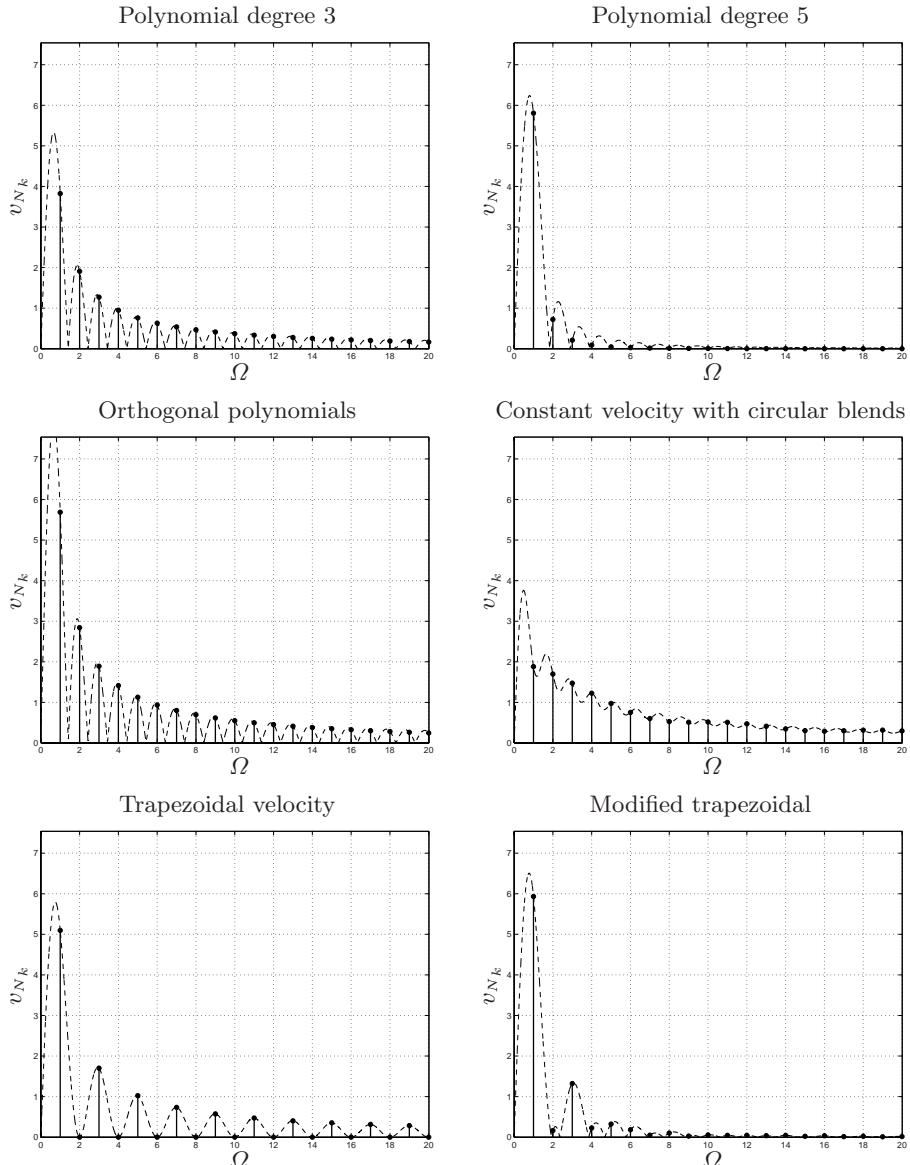
and, by taking into account the normalized trajectory (with coefficients  $v_{N_k}$  and the corresponding magnitude spectrum  $V'_N(\Omega)$ ),

$$v_k = \frac{h}{T} v_{N_k}, \quad \text{with} \quad v_{N_k} = \frac{2\pi}{T} V'_N(k).$$



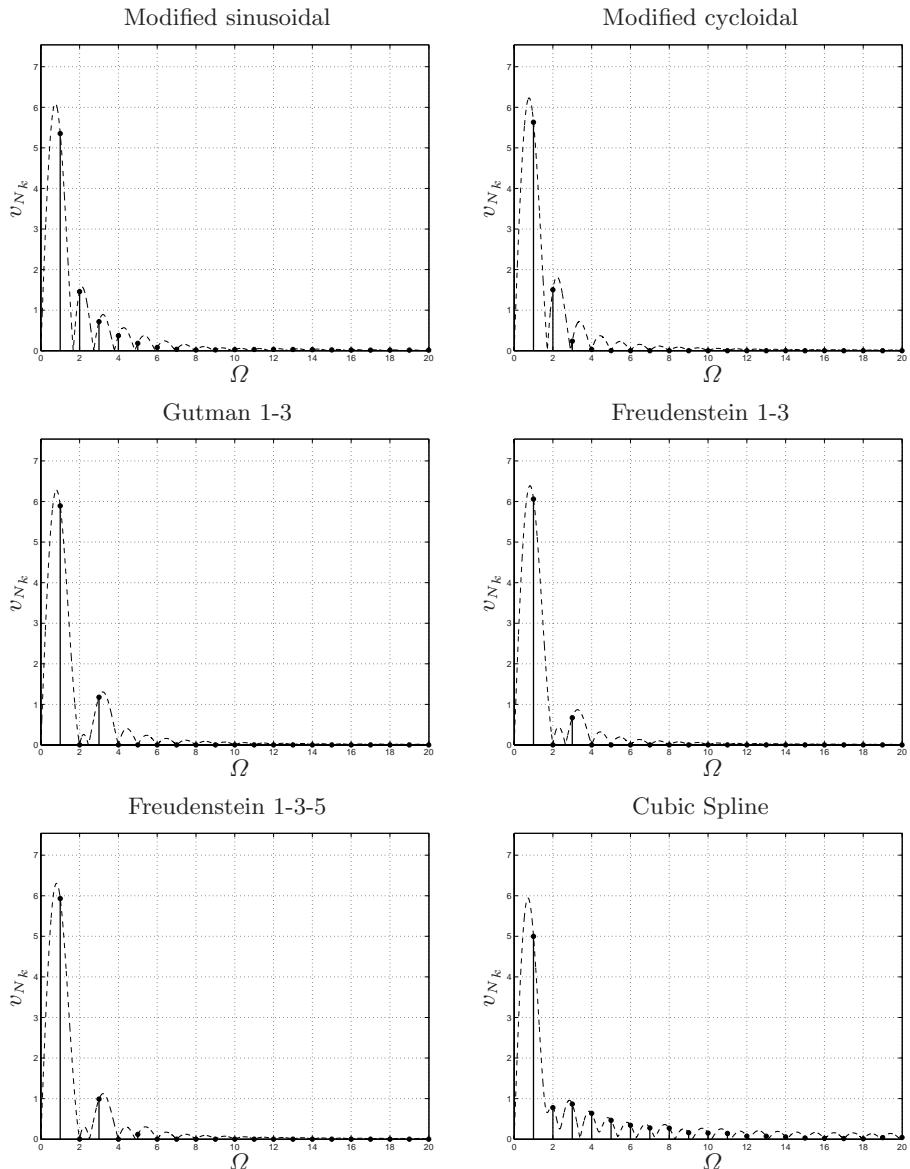
**Fig. 7.29.** Frequency analysis of the acceleration profiles of the periodic repetition of trajectories: constant velocity, constant acceleration, asymmetric constant acceleration, harmonic, cycloidal and elliptic.

From the spectra reported in the previous subsection one can immediately obtain the corresponding coefficients of the Fourier series expansion. In Fig. 7.29-7.31 such coefficients are shown for the same trajectories considered in the previous subsection.



**Fig. 7.30.** Frequency analysis of the acceleration profiles of the periodic repetition of trajectories: polynomials with degree 3 and 5, orthogonal polynomials, constant velocity with circular blends, trapezoidal, modified trapezoidal.

Note that in this case the best motion law is represented by the constant velocity trajectory which has a null harmonic content for any frequency value; this is quite obvious, since the periodic repetition of the constant velocity tra-



**Fig. 7.31.** Frequency analysis of the acceleration profiles of the periodic repetition of trajectories: modified sinusoidal, modified cycloidal, Gutman 1-3, Freudenstein 1-3 and 1-3-5, cubic spline.

jectory does not require any variation of the speed, and therefore the acceleration is always null ( $\ddot{q}(t) = 0, \forall t$ ). The cycloidal trajectory is characterized by only one component at the frequency

$$\Omega = 1 \xrightarrow{\omega = \frac{2\pi}{T}\Omega} \omega = \frac{2\pi}{T}.$$

The trajectories based on the Fourier series expansion have two or more non-null harmonic components, e.g. in the case of the Freudenstein and Gutman 1-3 trajectories only the components at  $\Omega = 1$  and  $\Omega = 3$  have a not null magnitude, while for the Freudenstein 1-3-5 also the component at  $\Omega = 5$  is not null.

All the other trajectories have larger harmonic contents, and a direct comparison can help in selecting a specific motion law among many different possibilities.

### 7.3.4 Scaling and frequency properties of a trajectory

The previous acceleration spectra  $V'_N(\Omega)$ , both of aperiodic and periodic trajectories, have been obtained with the conditions  $h = 1$ ,  $T = 1$  and are functions of the dimensionless variable  $\Omega$ . By inverting (7.6), the acceleration spectra of a generic trajectory can be easily deduced:

$$V(\omega) = V'(\Omega) \Big|_{\Omega = \omega T / 2\pi} = \frac{h}{T} V'_N(\Omega) \Big|_{\Omega = \omega T / 2\pi}.$$

Therefore, the magnitude spectrum  $V$  of the acceleration of a trajectory with a non-unitary displacement can be obtained by scaling by  $h$  the spectrum  $V'_N$  of the normalized trajectory. The duration of the trajectory has a twofold effect on this spectrum: firstly the magnitude of  $V'_N$  is scaled by  $1/T$ , and, secondly, the frequencies are scaled by  $T$  (note that  $\Omega$  is proportional to  $\omega T$ ).

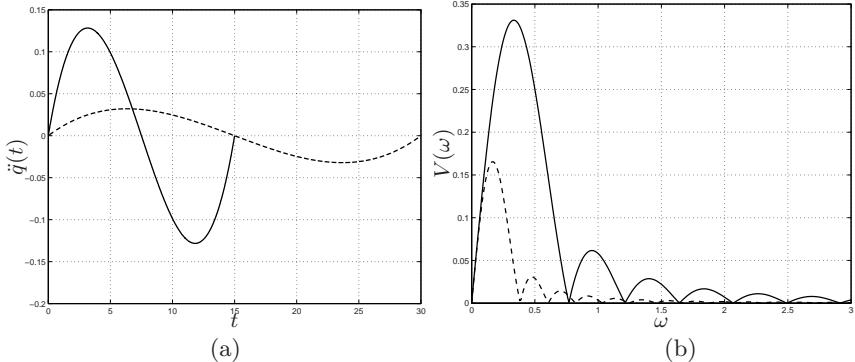
Given a generic motion law  $q(t)$  of duration  $T$  and with an acceleration spectrum  $V(\omega)$ , the trajectory obtained by time scaling  $q(t)$

$$q'(t') = q(t) \Big|_{t = \lambda t'}, \quad t' \in [0, T'], \quad \text{with } T' = \frac{T}{\lambda}$$

is characterized by a spectrum  $V_\lambda(\omega)$  of the acceleration profile related to  $V(\omega)$  by

$$V_\lambda(\omega) = \lambda V(\omega/\lambda).$$

Therefore, if we consider  $\lambda < 1$  (i.e. a slower motion), not only the frequency range but also the amplitudes, i.e.  $V_\lambda$ , are reduced. In principle, it is then possible to compute the value of  $\lambda$  in order to make in practice negligible, above an assigned limit frequency  $\bar{\omega}$ , the spectral content of the acceleration profile.



**Fig. 7.32.** (a) Acceleration profiles of a polynomial trajectory of degree 5 (solid) and of the same function scaled in time, with  $\lambda = 0.5$  (dashed). (b) Frequency content of the same signals.

**Example 7.1** Let us consider a polynomial trajectory of degree 5, defined with the conditions

$$q_0 = 10, \quad q_1 = 15, \quad t_0 = 0, \quad t_1 = 15, \quad T = t_1 - t_0 = 15.$$

If the value  $\lambda = 0.5$  is considered, then the scaled trajectory has a time length  $T' = T/\lambda = 30$ . The two acceleration profiles are shown in Fig. 7.32(a), while their frequency contents are reported in Fig. 7.32(b).

□

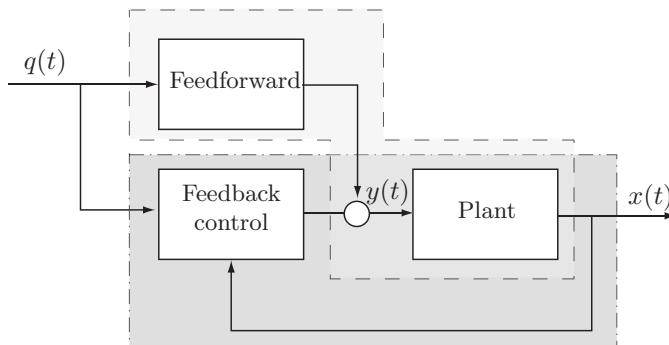
## 7.4 Frequency Modifications of Trajectories

The analyses reported in the previous sections show that a trajectory can generate undesired vibrations when applied to mechanical systems characterized by structural elasticities. For this reason, a trajectory should be selected by comparing its spectral content with the frequency response characteristics of the mechanical system. This is particularly important when high speed machines are considered. In this case, the suppression of residual vibrations<sup>5</sup> is a critical problem. This can be solved by means of different methods, based on the optimization of both the trajectory profiles and the control modalities:

<sup>5</sup> The amplitude of the residual vibrations can be defined as

$$\max_{t \geq t_1} |x(t) - q(t_1)|$$

where  $x(t)$  is the actual position of the plant at time  $t \geq t_1$ ,  $q(t_1) = q_1$  the desired position at the final time  $t_1$ .



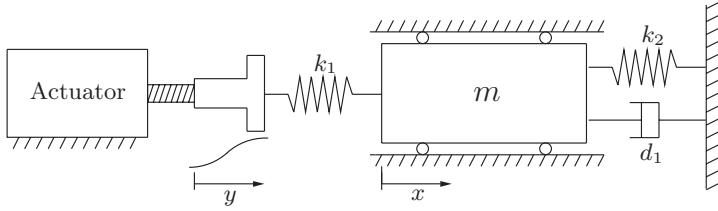
**Fig. 7.33.** Standard two dof controller.

- *Trajectory smoothing or shaping:* this approach consists in considering trajectories with high order continuity (therefore, with continuous jerk, snap, etc.), trying where possible to reduce the maximum values of velocity and acceleration. Moreover, it is possible to smooth or shape the motion profiles by means of proper filters (low-pass filters, notch filters, input shapers, and so on) with the purpose of minimizing the energy introduced at the system resonance frequencies [57].
- *Control optimization:* since in modern motion systems a combination of feedback and feedforward control action is normally used in a two degrees of freedom (two dof) controller structure, see Fig. 7.33, it is necessary to consider two main aspects: on one side the feedback control must guarantee the (robust) stability of the system and the disturbance rejection while, on the other hand, the feedforward action can be used to improve the tracking performance.

In the next sections, the effects on the system response of different types of filtering actions applied to the input trajectories are considered. The feedforward control is also taken into account, since it can be assimilated to a dynamic filter applied to the trajectory.

#### 7.4.1 Polydyne and splinedyne functions

The first attempts of shaping an input motion profile by taking into account the dynamic model of the system were made in the field of mechanical cam design [58, 59]. In 1953, the term *polydyne* was introduced as a contraction of *polynomial* and *dynamic*, [60]. It describes a design technique for mechanical cams, originally proposed in 1948 [61], that takes into account the dynamics of the cam/follower system in order to define a cam profile able to “compensate” for the dynamic vibrations of the follower train, at least at one particular cam speed. Given, for example, the simple mechanical chain shown in Fig. 7.34, it is possible to write the differential equation



**Fig. 7.34.** Scheme of the mechanical system considered for the design of a polydyne function.

$$m\ddot{x}(t) + d_1\dot{x}(t) + (k_1 + k_2)x(t) = k_1y(t). \quad (7.10)$$

As in Sec. 7.3, in the sequel it is assumed that an ideal actuation system is present, and therefore  $y(t) = q(t)$ . The basic idea of this approach is to use (7.10) for the definition of the function  $q(t) (= y(t))$  in order to obtain the desired profile  $x(t)$  or, in equivalent terms, to “invert” the dynamic model (7.10) in order to define the proper trajectory  $q(t)$  for the actuator, given the desired motion profile  $x(t)$  of the mass  $m$  (this type of approach will be considered also in the following Sec. 7.4.3, on the basis of considerations deriving from the automatic control field).

As a matter of fact, from (7.10) it is possible to compute position, velocity and acceleration of the actuator as

$$q(t) = \frac{1}{k_1} (m\ddot{x}(t) + d_1\dot{x}(t) + (k_1 + k_2)x(t)) \quad (7.11)$$

$$\dot{q}(t) = \frac{1}{k_1} (mx^{(3)}(t) + d_1\ddot{x}(t) + (k_1 + k_2)\dot{x}(t)) \quad (7.12)$$

$$\ddot{q}(t) = \frac{1}{k_1} (mx^{(4)}(t) + d_1x^{(3)}(t) + (k_1 + k_2)\ddot{x}(t)). \quad (7.13)$$

If the desired profile  $x(t)$  is known, from these equations it is possible to compute the motion  $q(t)$  so that undesired dynamic effects of the mechanical system are compensated for. Note that, in order to have smooth acceleration profiles for  $q(t)$ , it is necessary that the fourth derivative of  $x(t)$ , i.e. the snap, is smooth as well. This means that any desired motion profile  $x(t)$  should be continuous at least up to the fourth derivative.

Moreover, since in classical machines the displacement  $q(t)$  is normally obtained through the angular motion  $\theta(t)$  of a cam, and therefore  $q = q(\theta)$ , usually also the position  $x(t)$ , the velocity  $dx/dt$  and the acceleration  $d^2x/dt^2$  are defined in terms of the angular displacement  $\theta(t)$  of the cam, rotating at a constant velocity  $\dot{\theta}(t) = v_{rpm}$  expressed in rpm (revolutions per minute:  $1 \text{ rpm} = 2\pi \text{ min}^{-1} = 360^\circ \text{ min}^{-1} = 6^\circ \text{ sec}^{-1}$ ), from which  $\theta(t) = 6^\circ v_{rpm} t$ , with  $\theta(t)$  expressed in degrees. Therefore,

$$\frac{dq(t)}{dt} = \frac{dq(\theta)}{d\theta} \dot{\theta}(t) = \frac{dq(\theta)}{d\theta} 6v_{rpm}$$

$$\frac{d^2q(t)}{dt^2} = \frac{d^2q(\theta)}{d\theta^2} \dot{\theta}^2(t) = \frac{d^2q(\theta)}{d\theta^2} 36v_{rpm}^2$$

and similarly for  $x(t)$ . In conclusion, the acceleration profile  $d^2q/d\theta^2$  can be written as

$$\frac{d^2q}{d\theta^2} = 36v_{rpm}^2 \frac{m}{k_1} \frac{d^4x}{d\theta^4} + 6v_{rpm} \frac{d_1}{m} \frac{d^3x}{d\theta^3} + \frac{(k_1 + k_2)}{k_1} \frac{d^2x}{d\theta^2}$$

Therefore, the motion  $q(t)$  depends also on the desired angular speed, and a mechanical cam optimized for a given value of  $v_{rpm}$  has lower performances when used with other speeds.

In the literature, polynomial functions have been widely used to define proper desired profiles for  $x(t)$ , and a number of polydyne functions have been proposed, with specific properties aiming at satisfying different criteria. Although the minimum degree of a polynomial function compliant with boundary conditions up to the snap (the fourth derivative) is 9, polynomials of much higher degree have been utilized. In general, their derivation is not simple, and the interested reader may refer to the many publications available on this subject, see e.g. [58]-[62].

Polynomials in normalized form, i.e. with unitary displacement and duration, are usually adopted, whose coefficients can be easily modified in order to take into account a specific duration  $T$  and a desired displacement  $h$  (the coefficients of normalized polynomials up to degree 21 are reported in Sec. 2.1.7).

In [61], Dudley proposed a general polydyne profile on the basis of polynomials of the form

$$x_N(\tau) = 1 + a_2(1 - \tau)^2 + a_p(1 - \tau)^p + a_q(1 - \tau)^{(p+2)} + a_r(1 - \tau)^{p+4} \quad (7.14)$$

$$x_N \in [0, 1], \quad \tau \in [0, 1]$$

with  $p \geq 4$  and

$$a_2 = \frac{-6p^2 - 24p}{6p^2 - 8p - 8}, \quad a_p = \frac{p^3 + 7p^2 + 14p + 8}{6p^2 - 8p - 8},$$

$$a_q = \frac{-2p^3 - 4p^2 + 16p}{6p^2 - 8p - 8}, \quad a_r = \frac{p^3 - 3p^2 + 2p}{6p^2 - 8p - 8}.$$

In this way, polynomial functions with exponents 2-4-6-8, 2-6-8-10, ..., may be defined. In particular, the polynomial 2-10-12-14

$$x_N(\tau) = \frac{1}{64} \left( 64 - 105(1 - \tau)^2 + 231(1 - \tau)^{10} - 280(1 - \tau)^{12} + 90(1 - \tau)^{14} \right)$$

was considered among those offering the best performances, [61]. This polynomial may be rewritten as

$$x_N(\tau) = \frac{1}{64} \left( 90\tau^{14} - 1260\tau^{13} + 7910\tau^{12} - 29400\tau^{11} + 71841\tau^{10} - 120890\tau^9 + 142065\tau^8 - 114840\tau^7 + 60060\tau^6 - 16632\tau^5 + 1120\tau^3 \right).$$

Note that eq. (7.14) defines the polynomial function from 0 to 1, for  $\tau = [0, 1]$ . The return motion from 1 to 0, if of interest, is simply obtained as

$$x_N(\tau) = 1 + a_2\tau^2 + a_p\tau^p + a_q\tau^{(p+2)} + a_r\tau^{p+4}.$$

The profiles of the function  $x_N(\tau)$  for a rise-return motion are shown in Fig. 7.35. Note that velocity and acceleration have null values at the initial and final points. Moreover, acceleration, jerk and snap present constant values when the displacement is maximum.

Another example is the Peisekah polydyne function [63], that is based on the polynomial

$$x_N(\tau) = a_5\tau^5 + a_6\tau^6 + a_7\tau^7 + a_8\tau^8 + a_9\tau^9 + a_{10}\tau^{10} + a_{11}\tau^{11} \quad (7.15)$$

where

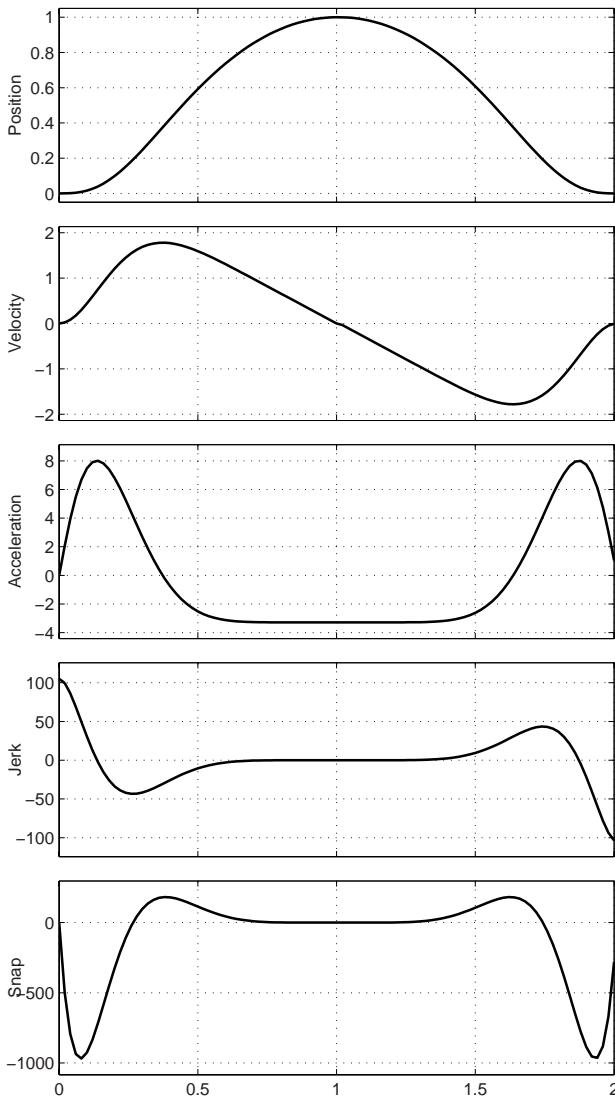
$$\begin{aligned} a_5 &= 336, & a_6 &= -1890, & a_7 &= 4740, \\ a_8 &= -6615, & a_9 &= 5320, & a_{10} &= -2310, & a_{11} &= 420. \end{aligned}$$

The degree of the polynomial and the values of the coefficients have been determined in order to satisfy proper boundary conditions (null values up to the snap), to obtain a symmetric behavior of the function, and to minimize the value of the peak acceleration (in this case the maximum value for the acceleration results 7.91, while the maximum acceleration for a standard normalized polynomial of degree 11 is 11.2666, see Tab. 2.2), [63]. The profiles of position, velocity, acceleration, jerk and snap of this polynomial function are shown in Fig. 7.36.

From the normalized polynomial  $x_N(\tau)$ , defining the shape and the characteristics of the desired trajectory, it is straightforward to determine the polynomial  $x(t)$  with the desired duration  $T = t_1 - t_0$  and displacement  $h = q_1 - q_0$ . As reported also in Sec. 2.1.7,  $x(t)$  and its  $k$ -th derivatives are polynomials of the form

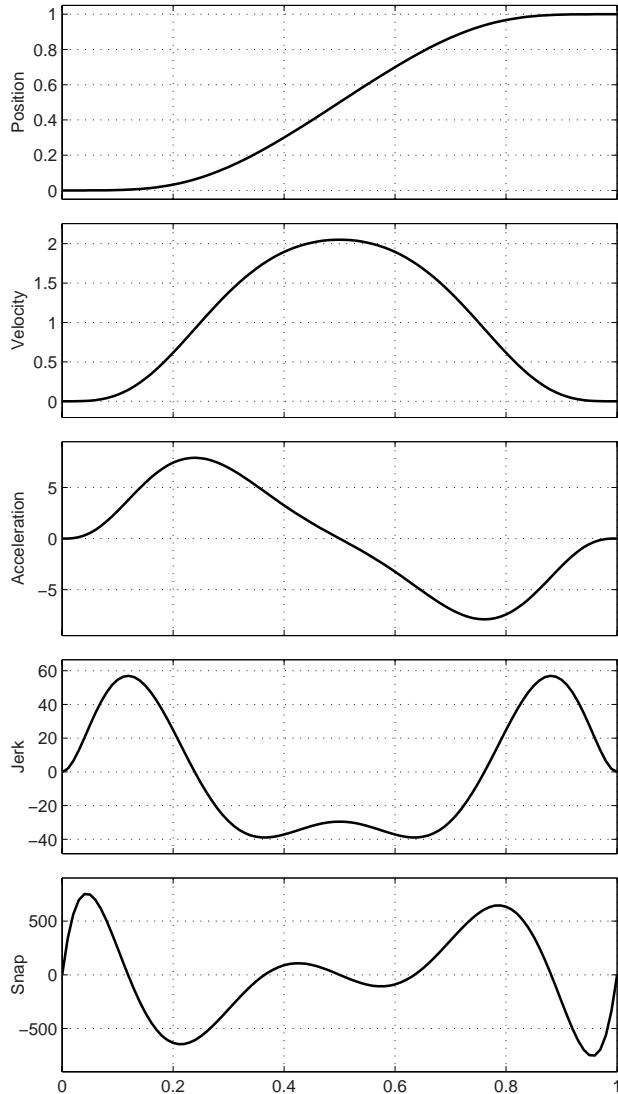
$$x^{(k)}(t) = \sum_{i=0}^{n-k} b_{i,k} (t - t_0)^i, \quad t \in [t_0, t_1] \quad (7.16)$$

whose coefficients are related to the coefficients  $a_i$  of the normalized polynomial function  $x_N(\tau)$  by



**Fig. 7.35.** Position, velocity, acceleration, jerk and snap for the Dudley polynomial function.

$$\begin{aligned}
 \text{position: } & x(t) = \sum_{i=0}^n b_{i,0}(t-t_0)^i \rightarrow b_{i,0} = \begin{cases} q_0 + h a_0, & i = 0 \\ \frac{h}{T^i} a_i, & i > 0 \end{cases} \\
 \text{velocity: } & \dot{x}(t) = \sum_{i=0}^{n-1} b_{i,1}(t-t_0)^i \rightarrow b_{i,1} = (i+1) \frac{h}{T^{i+1}} a_{i+1} \\
 \text{acceleration: } & \ddot{x}(t) = \sum_{i=0}^{n-2} b_{i,2}(t-t_0)^i \rightarrow b_{i,2} = (i+1)(i+2) \frac{h}{T^{i+2}} a_{i+2}.
 \end{aligned} \tag{7.17}$$



**Fig. 7.36.** Position, velocity, acceleration, jerk and snap for the Peisekah polynomial function.

Once a proper polynomial function  $x(t)$  and its derivatives up to the proper order have been defined, it is possible to compute the polydyne function  $q(t)$  that takes into account the dynamics of the mechanical system, as for example in eq. (7.11). Note that, as in the case of mechanical cams design based on polydyne functions, which is optimized for a specific velocity  $v_{rpm}$ , in the trajectory planning the duration  $T$  of the desired motion  $x(t)$  must be fixed in

advance, since it is not possible to scale in time a polydyne function  $q(t)$ , with the technique discussed in Sec. 5.1, without deteriorating its performances. As a matter of fact, if the duration  $T$  is changed, it is necessary firstly to recompute the coefficients of  $x(t)$ ,  $\dot{x}(t)$ ,  $\ddot{x}(t), \dots$ , and then the overall polydyne function  $q(t)$ .

A more straightforward method to obtain the polydyne function  $q(t)$  is the following. Let us consider a generic dynamic model expressed in the Laplace transform domain as

$$\frac{X(s)}{Q(s)} = G(s) = \frac{k}{s^2 + 2\delta\omega_n s + \omega_n^2}$$

equivalent, with null initial conditions, to the second order differential equation

$$\ddot{x}(t) + 2\delta\omega_n \dot{x}(t) + \omega_n^2 x(t) = k q(t). \quad (7.18)$$

If the desired displacement of the mass is expressed by means of a polynomial function  $x(t)$ , from (7.18) and by taking into account the relationships (7.17) (for the sake of simplicity  $t_0 = 0$  is assumed) one obtains

$$\begin{aligned} q(t) &= \frac{1}{k} \left[ \sum_{i=0}^n (i+1)(i+2) \frac{h}{T^{i+2}} a_{i+2} t^i + 2\delta\omega_n \sum_{i=0}^n (i+1) \frac{h}{T^{i+1}} a_{i+1} t^i + \right. \\ &\quad \left. + \omega_n^2 \left( q_0 + \sum_{i=0}^n a_i \frac{h}{T^i} t^i \right) \right] \\ &= \frac{\omega_n^2}{k} q_0 + \frac{h}{k} \sum_{i=0}^n \left( (i+1)(i+2) \frac{a_{i+2}}{T^{i+2}} + 2\delta\omega_n (i+1) \frac{a_{i+1}}{T^{i+1}} + \omega_n^2 \frac{a_i}{T^i} \right) t^i \\ &= \frac{\omega_n^2}{k} q_0 + \sum_{i=0}^n b_i t^i \end{aligned} \quad (7.19)$$

where the coefficients  $b_i$  are

$$b_i = \frac{h}{k} \left( \omega_n^2 \frac{a_i}{T^i} + 2\delta\omega_n (i+1) \frac{a_{i+1}}{T^{i+1}} + (i+2)(i+1) \frac{a_{i+2}}{T^{i+2}} \right) \quad (7.20)$$

with  $a_{n+1} = a_{n+2} = 0$ .

With (7.19) and (7.20) it is possible to compute directly the polydyne function  $q(t)$  from the coefficients  $a_i$  of the normalized polynomial  $x_N(\tau)$ , the desired duration  $T$  and displacement  $h$ , and obviously the relevant parameters of the dynamic model.

**Example 7.2** Let us consider the mechanism represented in Fig. 7.34 with

$$m = 5 \text{ kg}, \quad k_1 = 5000 \text{ N/m}, \quad k_2 = 1000 \text{ N/m}, \quad d_1 = 25 \text{ Ns/m}.$$

The desired displacement is a periodic motion from  $x_0 = 0$  to  $x_1 = 50$  ( $h = 50$ ) and back in a period of time  $T = T_r + T_d = 1$ , with  $T_r = T_d = 0.5$  for the rise ( $T_r$ ) and for the return ( $T_d$ ) part. If the Peisekah polynomial is chosen, the polydyne function  $q(t)$  is computed by using (7.15) and its derivatives in (7.17) and then in (7.11) or, equivalently, from eq. (7.19) and (7.20).

From the parameters of the system, it results

$$\omega_n = \sqrt{\frac{k_1 + k_2}{m}} = 34.641, \quad \delta = \frac{d_1}{2m\omega_n} = 0.0722, \quad k = \frac{k_1}{m} = 1000.$$

Note that the static gain of the systems in  $G(0) = k/\omega_n^2 = 0.8333$ . From eq. (7.19) and (7.20) one obtains

$$\begin{aligned} q_{rise}(t) = & \frac{h}{k} \left[ \frac{a_{11}}{T_r^{11}} \omega_n^2 t^{11} + \left( \frac{a_{10}}{T_r^{10}} \omega_n^2 + 22 \frac{a_{11}}{T_r^{11}} \delta \omega_n \right) t^{10} \right. \\ & + \left( \frac{a_9}{T_r^9} \omega_n^2 + 20 \frac{a_{10}}{T_r^{10}} \delta \omega_n + 110 \frac{a_{11}}{T_r^{11}} \right) t^9 \\ & + \left( \frac{a_8}{T_r^8} \omega_n^2 + 18 \frac{a_9}{T_r^9} \delta \omega_n + 90 \frac{a_{10}}{T_r^{10}} \right) t^8 \\ & + \left( \frac{a_7}{T_r^7} \omega_n^2 + 16 \frac{a_8}{T_r^8} \delta \omega_n + 72 \frac{a_9}{T_r^9} \right) t^7 \\ & + \left( \frac{a_6}{T_r^6} \omega_n^2 + 14 \frac{a_7}{T_r^7} \delta \omega_n + 56 \frac{a_8}{T_r^8} \right) t^6 \\ & + \left( \frac{a_5}{T_r^5} \omega_n^2 + 12 \frac{a_6}{T_r^6} \delta \omega_n + 42 \frac{a_7}{T_r^7} \right) t^5 \\ & \left. + \left( 10 \frac{a_5}{T_r^5} \delta \omega_n + 30 \frac{a_6}{T_r^6} \right) t^4 + 20 \frac{a_5}{T_r^5} t^3 \right] \end{aligned}$$

and, after the substitutions,

$$\begin{aligned} q_{rise}(t) = & 51609600 t^{11} - 139560960 t^{10} + 162247680 t^9 \\ & - 106122240 t^8 + 42822144 t^7 - 10937472 t^6 \\ & + 1737792 t^5 - 168000 t^4 + 10752 t^3 \end{aligned}$$

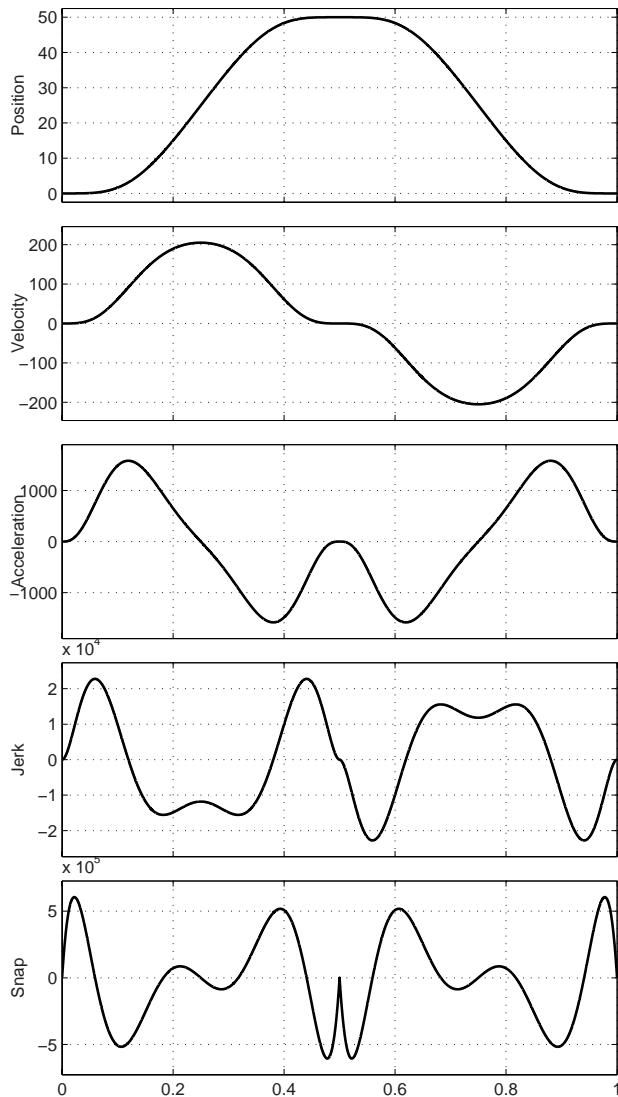
for  $t \in [0, T_r]$ . Because of symmetry ( $T_r = T_d$ ), the computation of  $q_{return}(t)$  in the interval  $[T_r, T]$ , can be simply obtained as

$$q_{return}(t) = q_{rise}(T_r) - q_{rise}(t - T_r), \quad t \in [T_r, T].$$

The desired motion  $x(t)$  for the mechanical system of Fig. 7.34, obtained from (7.17) as

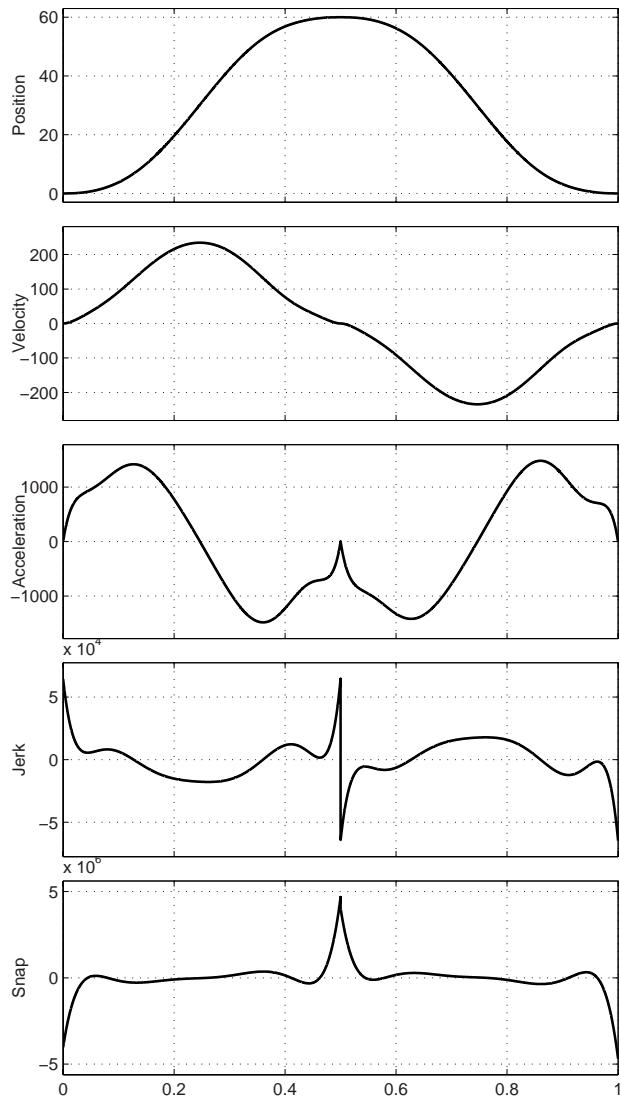
$$x(t) = h x_N(\tau), \quad \text{with } \tau = \frac{t}{T_r}, \quad t \in [0, T_r]$$

for the rise (and similarly for the return) period is reported in Fig. 7.37. The overall function  $q(t)$  for the rise and return periods and its derivatives are shown in Fig. 7.38.



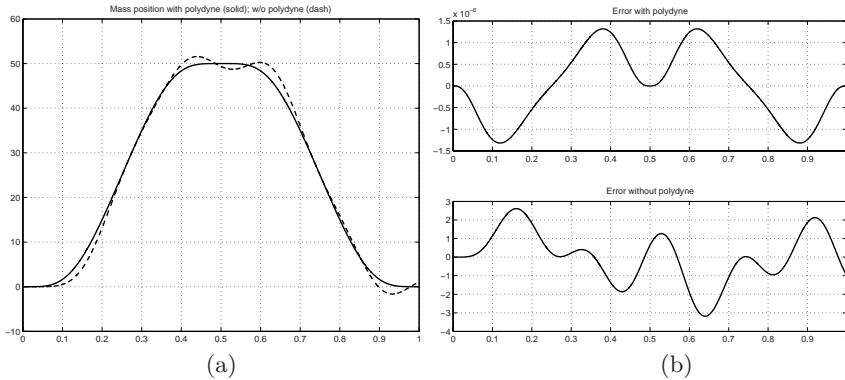
**Fig. 7.37.** Desired motion profile  $x(t)$  for the mass of Fig. 7.34.

In Fig. 7.39, a comparison is reported between the motion of the mass when  $q(t)$  is computed as described above, taking into consideration the dynamics of the mechanism, and when the input profile  $q'(t)$  is the (smooth) profile defined by the polynomial function of degree 11 in (7.15), that for the rise period  $[0, T_r]$  is

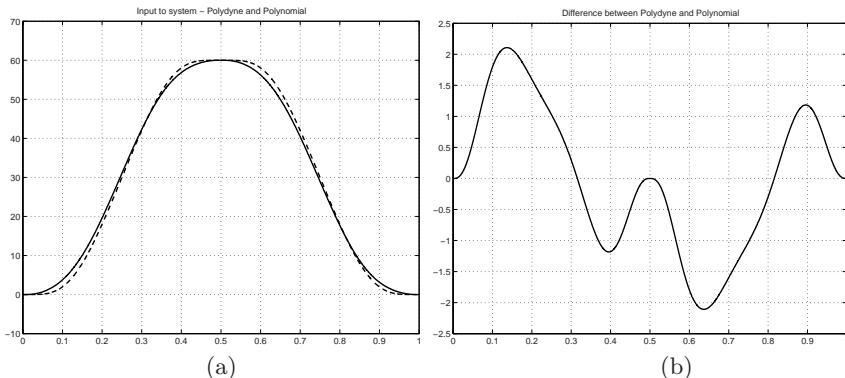


**Fig. 7.38.** Position, velocity, acceleration, jerk and snap for the Peisekah polydyne trajectory  $q(t)$ .

$$\begin{aligned}
 q'_{rise}(t) &= \frac{\omega_n^2}{k} x(t) \\
 &= 51609600 t^{11} - 141926400 t^{10} + 163430400 t^9 \\
 &\quad - 101606400 t^8 + 36403200 t^7 - 7257600 t^6 + 645120 t^5.
 \end{aligned}$$



**Fig. 7.39.** Position of the mass of Fig. 7.34 (a) and tracking errors due to the elasticity (b) with and without using the polydyne; note the different amplitudes of the diagrams in (b).

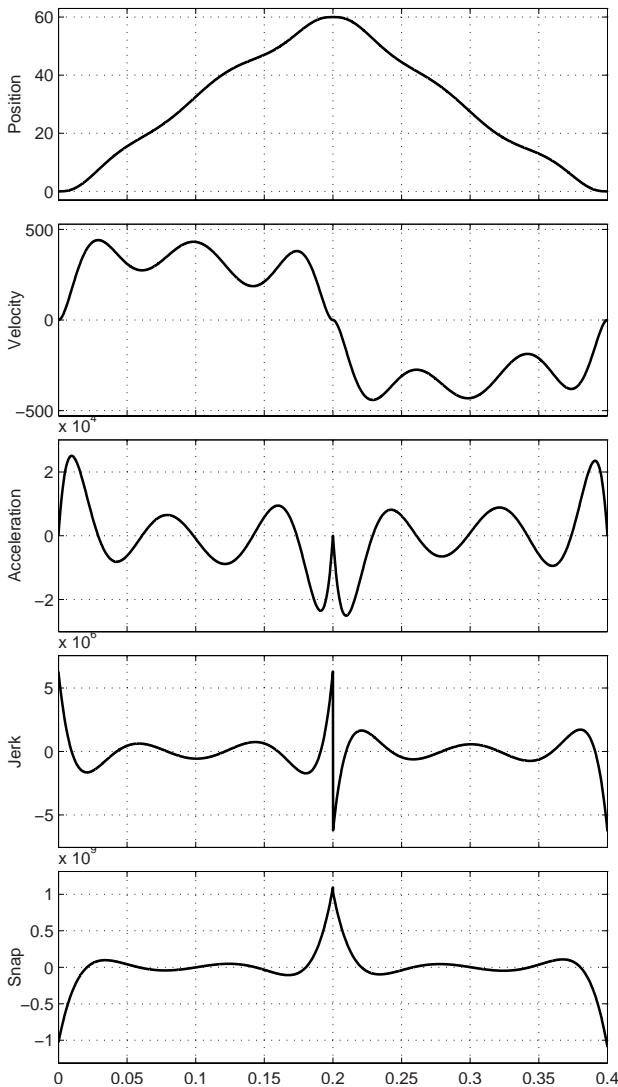


**Fig. 7.40.** Functions  $q(t)$  (solid) and  $q'(t)$  (dashed) given as input to the system (a), and their difference (b).

The two input profiles  $q(t)$  and  $q'(t)$  are compared in Fig. 7.40. Note that, although also this latter function is very smooth, and with the same degree of the polydyne, quite relevant errors are present in the motion of the mass due to the non-compensated elasticity of the system. □

**Example 7.3** The position profiles of Fig. 7.40 are quite similar: as a matter of fact the compensating actions in the polydyne profile  $q(t)$  become more evident only when the speed of the desired motion increases.

Let us consider the same rise-return motion from  $x_0 = 0$  to  $x_1 = 50$  ( $h = 50$ ) as in the previous example. Assume now  $T = T_r + T_d = 0.4$ , with  $T_r = T_d = 0.2$ . The increased velocity and acceleration components are now evident in the

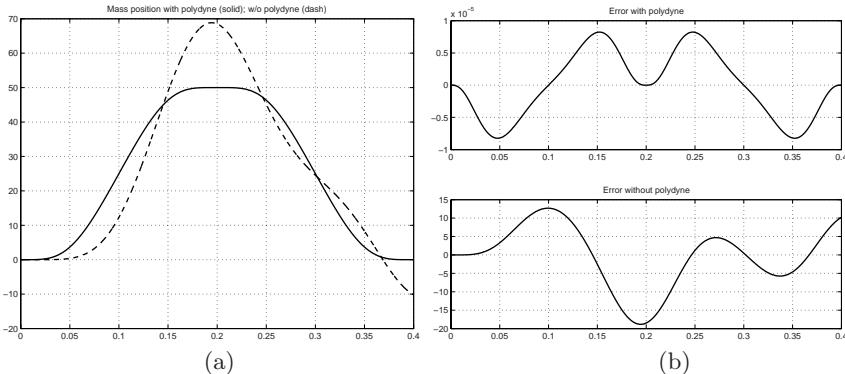


**Fig. 7.41.** Position, velocity, acceleration, jerk and snap for the Peisekah polydyne trajectory ( $T_r = T_d = 0.2$ ).

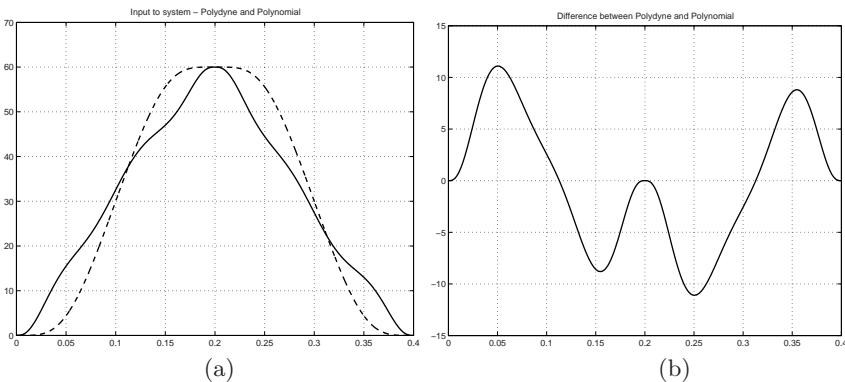
position profile  $q(t)$  shown in Fig. 7.41 (compare with Fig. 7.38).

In Fig. 7.42 and Fig. 7.43 the comparison between  $q(t)$  and  $q'(t)$  is reported. In this case, the differences originated by the two profiles are very relevant.

□



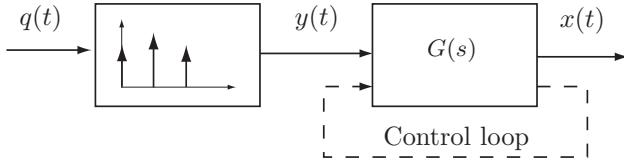
**Fig. 7.42.** Position of the mass of Fig. 7.34 (a) and tracking errors due to the elasticity (b) with and without using the polydyne; note the different amplitudes of the diagrams in (b).



**Fig. 7.43.** Functions  $q(t)$  (solid) and  $q'(t)$  (dashed) given as input to the system (a), and their difference (b).

Polydyne functions have been used extensively and in several important applications, like for example for automotive valve cams. However, this type of functions has some drawbacks, related to the fact that the mechanical cams built according to this technique are optimized only for a given speed of rotation, and to the fact that polynomial functions of high degree must be used (polynomials up to degree 30, 40 and even more have been proposed), with the consequences that high values for the maximum accelerations and undesired excursions, typical of high-degree polynomials, are obtained. For these reasons, spline functions, and in particular B-splines, are currently adopted for their superior properties with respect to simple polynomial functions. The term *splinedyne* has then been proposed, [58].

Moreover, it has to be mentioned the fact that the problem of defining proper parameters and functions for the design of mechanical cams has been faced



**Fig. 7.44.** Block diagram of a system with a shaper.

also with approaches based on the optimal control theory, [64], trying to minimize a cost function based on the derivatives of the displacement and on the forces applied to the follower. However, this type of approach has been generalized, for automatic machines with electronic cams, in the context of feedforward control as illustrated in the following sections.

#### 7.4.2 Input filtering and shaping

The simplest way to reduce the energy of a trajectory at frequencies close to the resonance of a given system is to properly filter the profile. Low-pass filters can be designed with standard techniques, i.e. Butterworth, Chebyshev, Elliptic and so on, with a pass band that anticipates the lowest resonance frequency of the system. In this manner, the spectral components of the trajectory above the cut-off frequency of the filter are reduced below a desired level. Similarly, it is possible to apply to the trajectory notch filters centered around the resonance frequency with the purpose of reducing the components of the trajectory in a more selective way. For the design of (analog and digital) filters see e.g. [65, 66, 67].

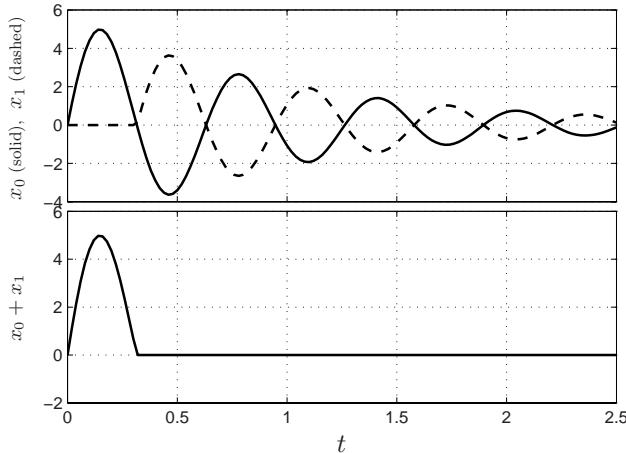
A significant drawback of this approach is the fact that this kind of filters produces distortions on the trajectory and introduces time delays. Moreover, as highlighted in [68], generic low-pass and notch filters are not able to completely cancel output vibrations.

A different and more effective method for the reduction of vibrations is the so-called *input shaping*. It consists in convolving a sequence of impulses, which form the *input shaper*, with the desired trajectory and applying the signal obtained in this way to the (controlled) system, see Fig. 7.44. The main idea of input shaping, based on the knowledge of a model of the system, supposed stable, consists in generating an input  $y(t)$  able to cancel out the vibrations induced on the plant.

For example, let us consider a second order system with a natural frequency  $\omega_n$  and a damping ratio  $\delta$  expressed by

$$G(s) = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2}. \quad (7.21)$$

If the input  $y(t)$  to this system is a Dirac impulse, the output  $x(t)$  is the damped sinusoidal function



**Fig. 7.45.** Response to the Dirac impulse of the second order system (7.21) and vibration cancellation using two impulses.

$$x(t) = \frac{e^{-\delta\omega_n t}}{\omega_n \sqrt{1 - \delta^2}} \sin(\omega_n \sqrt{1 - \delta^2} t)$$

with period  $T_0 = 2\pi/(\omega_n \sqrt{1 - \delta^2})$ , see Fig. 7.45.

In order to eliminate the output oscillations after the first lobe, the simplest approach consist in applying a second impulse delayed by  $T_0/2$ , which produces an oscillation out of phase with respect to the first one, see Fig. 7.45.

The two impulses, which form the shaper, can be convolved with any realizable trajectory to generate an input signal with the same vibration-cancelling properties of the input shaper [69]. The amplitudes of the two impulses must be computed in such a way that the response to the second impulse cancels the vibrations due to the first one, so that the residual vibrations are null.

In the case of the second order system (7.21) the residual vibration, resulting from a sequence of  $n + 1$  impulses, has the expression [70, 71]

$$Z(\omega_n, \delta) = e^{-\delta\omega_n t_n} \sqrt{\left( \sum_{i=0}^n s_i e^{\delta\omega_n t_i} \cos(\omega_n \sqrt{1 - \delta^2} t_i) \right)^2 + \left( \sum_{i=0}^n s_i e^{\delta\omega_n t_i} \sin(\omega_n \sqrt{1 - \delta^2} t_i) \right)^2}$$

where  $s_i$  is the magnitude of the  $i$ -th impulse applied at the time instant  $t_i$ .

In case  $n = 1$ , the *Zero Vibration (ZV) shaper* is obtained [69] by setting

$$Z(\omega_n, \delta) = 0.$$

It is composed by two impulses, characterized by

time instants	$t_0 = 0,$	$t_1 = T_0/2$
amplitudes	$s_0 = \frac{1}{1 + \kappa},$	$s_1 = \frac{\kappa}{1 + \kappa}$

where the two constants  $T_0$  (period vibration) and  $\kappa$  have the expressions

$$T_0 = \frac{2\pi}{\omega_n \sqrt{1 - \delta^2}}, \quad \kappa = e^{-\frac{\delta\pi}{\sqrt{1 - \delta^2}}}.$$

The success of this technique is strictly related to the knowledge of the system's parameters: if these are not perfectly known, the vibration cancellation may result imperfect. To make the input shaping technique more robust with respect to modelling errors, additional equations must be added to the problem formulation, and accordingly the number of impulses increases. This assures that the vibrations will remain limited even in presence of modelling errors.

Typical additional conditions concern the derivatives of  $Z(\omega_n, \delta)$ , which are set equal to zero. If  $\frac{\partial Z(\omega_n, \delta)}{\partial \omega_n} = 0$  the *Zero Vibration, Zero Derivative (ZVD) shaper*, composed by three impulses, is obtained:

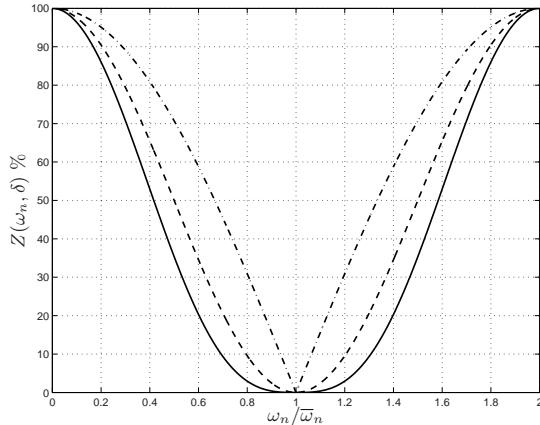
time instants	$t_0 = 0,$	$t_1 = T_0/2,$	$t_2 = T_0$
amplitudes	$s_0 = \frac{1}{1 + 2\kappa + \kappa^2},$	$s_1 = \frac{2\kappa}{1 + 2\kappa + \kappa^2},$	$s_2 = \frac{\kappa^2}{1 + 2\kappa + \kappa^2}.$

The further condition  $\frac{\partial^2 Z(\omega_n, \delta)}{\partial \omega_n^2} = 0$  leads to the *ZVDD shaper*:

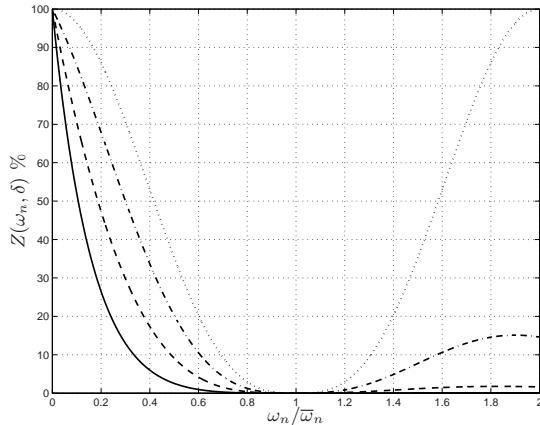
time instants	$\begin{cases} t_0 = 0, & t_1 = T_0/2, \\ t_2 = T_0, & t_3 = 3T_0/2, \end{cases}$
amplitudes	$\begin{cases} s_0 = \frac{1}{1 + 3\kappa + 3\kappa^2 + \kappa^3}, & s_1 = \frac{3\kappa}{1 + 3\kappa + 3\kappa^2 + \kappa^3}, \\ s_2 = \frac{3\kappa^2}{1 + 3\kappa + 3\kappa^2 + \kappa^3}, & s_3 = \frac{\kappa^3}{1 + 3\kappa + 3\kappa^2 + \kappa^3}. \end{cases}$

In Fig. 7.46 the sensitivity curve of the ZV, ZVD and ZVDD shapers are compared for undamped systems ( $\delta = 0$ ). The three plots are null for  $\omega_n/\bar{\omega}_n = 1$ , that is for the nominal value  $\bar{\omega}_n$  of the natural frequency. The main difference among them consists in the width of the notch around  $\omega_n = \bar{\omega}_n$ . As a consequence, the ZVDD, with the largest notch, is the most robust and, also in case of noticeable modelling errors it produces low amplitude vibrations. When systems with a non-null damping coefficient  $\delta$  are considered, the sensitivity curve is deformed but it is still null for  $\omega_n = \bar{\omega}_n$ , see Fig. 7.47. In particular, for  $\delta > 0$ , the values of the curve at frequencies higher than  $\bar{\omega}_n$  are considerably reduced. This means that for high values of the damping coefficient the robustness of the filter with respect to an overestimation of the natural frequency is improved.

After the design of the shaper, i.e. of the parameters  $s_i, t_i$  made according to the above discussion, the input signal  $y(t)$  resulting from the convolution of the shaper with the desired trajectory  $q(t)$  is computed as



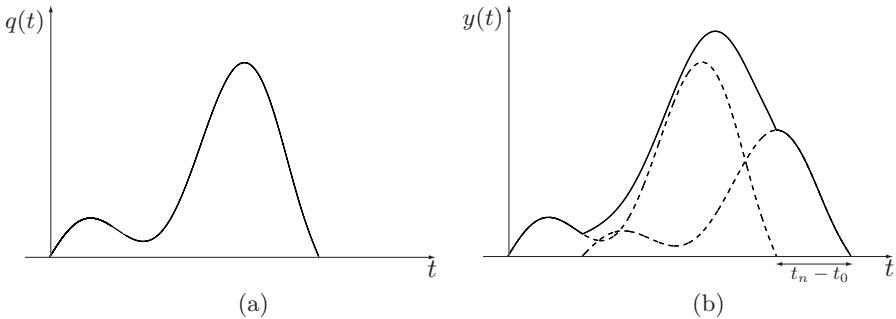
**Fig. 7.46.** Sensitivity curves of ZV (dashdot) ZVD (dashed) and ZVDD (solid) shapers, obtained for  $\delta = 0$ .



**Fig. 7.47.** Sensitivity curves of a ZVDD shaper for different value of  $\delta$ :  $\delta = 0$  (dot),  $\delta = 0.2$  (dashdot),  $\delta = 0.4$  (dashed),  $\delta = 0.6$  (solid).

$$y(t) = \sum_{i=0}^n s_i q(t - t_i).$$

As a consequence, the total duration of the motion is extended by a time period equal to  $t_n - t_0$  due to the impulse sequence, see Fig. 7.48 where a generic trajectory is filtered by a ZV shaper. Therefore, the robustness of the system is increased if more impulses are used, although a larger delay is injected into the system response.



**Fig. 7.48.** Generic trajectory  $q(t)$  (a), and input function  $y(t)$  obtained by convoluting  $q(t)$  with a ZV shaper (b).

**Example 7.4** Let us consider the second order system described in Sec. 7.1.1, with transfer function

$$G(s) = \frac{d s + k}{m s^2 + d s + k} \quad (7.22)$$

and with the parameters

$$m = 1 \text{ kg}, \quad d = 2 \text{ Ns/m}, \quad k = 100 \text{ N/m}.$$

In this case, the damping coefficient and the natural frequency are

$$\delta = 0.1, \quad \omega_n = 10.$$

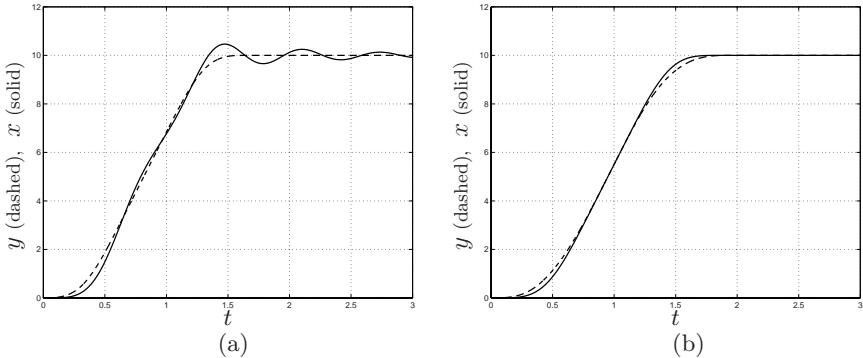
The amplitude and the application time instant of each impulse composing the ZV, ZVD and ZVDD input shapers are, respectively,

$$\begin{aligned} \text{ZV} \quad & \left\{ \begin{array}{ll} t_0 = 0, & t_1 = 0.3157 \\ s_0 = 0.5783, & s_1 = 0.4217 \end{array} \right. \\ \text{ZVD} \quad & \left\{ \begin{array}{lll} t_0 = 0, & t_1 = 0.3157, & t_2 = 0.6315 \\ s_0 = 0.3344, & s_1 = 0.4877, & s_2 = 0.1778 \end{array} \right. \\ \text{ZVDD} \quad & \left\{ \begin{array}{llll} t_0 = 0, & t_1 = 0.3157, & t_2 = 0.6315, & t_3 = 0.9472 \\ s_0 = 0.1934, & s_1 = 0.4231, & s_2 = 0.3085, & s_3 = 0.0750. \end{array} \right. \end{aligned}$$

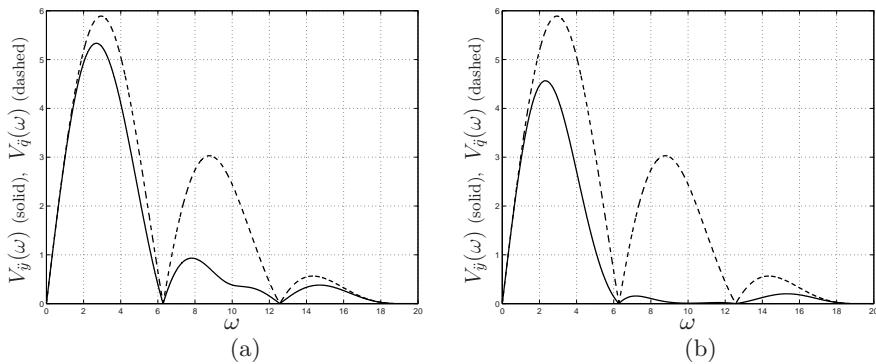
Assume the double S trajectory  $q(t)$  defined by

$$q_0 = 0, \quad q_1 = 10, \quad v_{max} = 10, \quad a_{max} = 50, \quad j_{max} = 100.$$

The application of the ZV shaper to the trajectory  $q(t)$  suppresses all the vibrations that otherwise would be superimposed to the desired output of the system. The outputs of  $G(s)$  without and with the input shaper are shown in Fig. 7.49. The shaper acts by reducing the magnitude of the spectral compo-



**Fig. 7.49.** Input (dashed) and output (solid) of  $G(s)$  without (a) and with (b) ZV shaper.



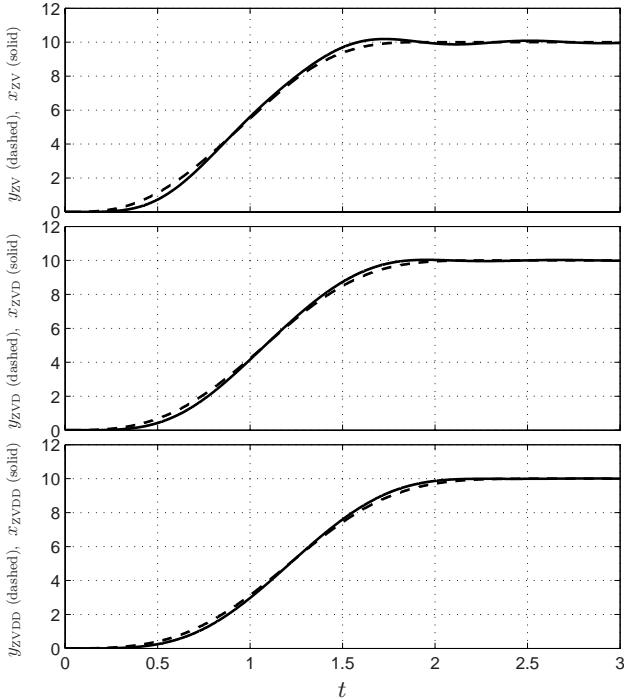
**Fig. 7.50.** Spectrum of  $\dot{y}(t)$  (solid) obtained by shaping the trajectory  $q(t)$ , compared with the spectrum of  $\ddot{q}(t)$  (dashed): ZV shaper (a), ZVDD shaper (b).

nents of  $q(t)$  located around the resonance frequency of the mechanical system that, in this case, is approximatively equal to  $\omega_n$ . In Fig. 7.50(a) the spectrum of the second derivative of

$$y(t) = s_0 q(t) + s_1 q(t - t_1)$$

is compared with the spectrum of  $\dot{q}(t)$ , while in Fig. 7.50(b) the acceleration spectrum obtained by adopting a ZVDD shaper is considered. Note that in this latter case, the components at high frequencies (around  $\omega_n = 10$ ) are reduced in a more pronounced way with respect to the case of the ZV shaper. This reflects the major robustness of the ZVDD shaper when estimation errors of the natural frequency are present, but also the larger delay injected by the ZVDD shaper into the system.

For example, as shown in Fig. 7.51, when a ZV shaper defined for the nominal parameters of  $G(s)$  is applied to the real system characterized by different values of  $\delta$  and  $\omega_n$  (in Fig. 7.51 the values  $\omega_n = 8$ ,  $\delta = 0.125$  are



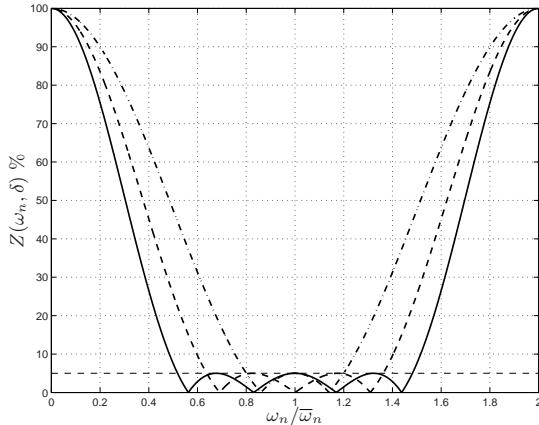
**Fig. 7.51.** Output  $x(t)$  of the system (7.22) (solid) with the parameters values  $m = 1 \text{ kg}$ ,  $d = 2 \text{ Ns/m}$ ,  $k = 64 \text{ N/m}$  when the input trajectory  $q(t)$  is modified with a ZV, a ZVD or a ZVDD shaper.

considered) some oscillations appear in the system response. Conversely, the output obtained by imposing the input  $y(t)$  computed from the trajectory  $q(t)$  by means of the ZVDD shaper is not affected by residual vibrations.  $\square$

An alternative approach for increasing the insensitivity of the shaper with respect to modelling errors is based on Extra-Insensitive (EI) constraints [72]. Instead of forcing the residual vibration  $Z(\omega_n, \delta)$  to zero at the modelling frequency  $\bar{\omega}_n$ , the residual vibration is limited to a given level  $\bar{z}$ . The width of the notch in the sensitivity curve is maximized by forcing the vibration to zero at two frequencies (one-hump EI), one lower than the nominal frequency, and the other higher (see Fig. 7.52) obtaining a shaper composed by three impulses that, in case of undumped systems ( $\delta = 0$ ), are characterized by

$$\begin{array}{lll} \text{time instants} & t_0 = 0, & t_1 = T_0/2, \quad t_2 = T_0 \\ \text{amplitudes} & s_0 = \frac{1 + \bar{z}}{4}, \quad s_1 = \frac{1 - \bar{z}}{2}, \quad s_2 = \frac{1 + \bar{z}}{4}. \end{array}$$

Obviously, it is possible to add more constraints to reduce the sensitivity of the



**Fig. 7.52.** Sensitivity curves of one-hump (dashdot), two-hump (dashed) and three-hump (solid) EI shapers, obtained for  $\delta = 0$ .

shaper, obtaining in this way multi-hump shapers. For instance, a two-hump shaper can be obtained by considering four impulses<sup>6</sup>, whose amplitudes and time instants for  $\bar{z} = 5\%$  are

$$\begin{aligned} \text{time instants} & \left\{ \begin{array}{l} t_0 = 0 \\ t_1 = (0.4989 + 0.1627 \delta - 0.5426 \delta^2 + 6.1618 \delta^3)T_0 \\ t_2 = (0.9974 + 0.1838 \delta - 1.5827 \delta^2 + 8.1712 \delta^3)T_0 \\ t_3 = (1.4992 - 0.0929 \delta - 0.2833 \delta^2 + 1.8571 \delta^3)T_0 \end{array} \right. \\ \text{amplitudes} & \left\{ \begin{array}{l} s_0 = 0.1605 + 0.7669 \delta + 2.2656 \delta^2 - 1.2275 \delta^3 \\ s_1 = 0.3391 + 0.4508 \delta - 2.5808 \delta^2 + 1.7365 \delta^3 \\ s_2 = 0.3408 - 0.6153 \delta - 0.6876 \delta^2 + 0.4226 \delta^3 \\ s_3 = 0.1599 - 0.6024 \delta + 1.0028 \delta^2 - 0.9314 \delta^3. \end{array} \right. \end{aligned}$$

A three-hump shaper composed by the five impulses defined (for  $\bar{z} = 5\%$ ) by

<sup>6</sup> The values of the impulses parameters are expressed in a polynomial form as a function of the damping coefficient  $\delta$ .

$$\begin{aligned}
 \text{time instants} & \quad \left\{ \begin{array}{l} t_0 = 0 \\ t_1 = (0.4997 + 0.2383 \delta + 0.4455 \delta^2 + 12.4720 \delta^3) T_0 \\ t_2 = (0.9984 + 0.2980 \delta - 2.3646 \delta^2 + 23.3999 \delta^3) T_0 \\ t_3 = (1.4987 + 0.1030 \delta - 2.0139 \delta^2 + 17.0320 \delta^3) T_0 \\ t_4 = (1.9960 - 0.2823 \delta + 0.6153 \delta^2 + 5.4045 \delta^3) T_0 \end{array} \right. \\
 \text{amplitudes} & \quad \left\{ \begin{array}{l} s_0 = 0.1127 + 0.7663 \delta + 3.2916, \delta^2 - 1.4438, \delta^3 \\ s_1 = 0.2369 + 0.6116 \delta - 2.5785 \delta^2 + 4.8522 \delta^3 \\ s_2 = 0.3000 - 0.1906 \delta - 2.1456 \delta^2 + 0.1374 \delta^3 \\ s_3 = 0.2377 - 0.7329 \delta + 0.4688 \delta^2 - 2.0865 \delta^3 \\ s_4 = 0.1124 - 0.4543 \delta + 0.9638 \delta^2 - 1.4600 \delta^3 \end{array} \right.
 \end{aligned}$$

produces the sensitivity curve represented by a solid line in Fig. 7.52.

**Example 7.5** For the design of an EI input shaper, the system of Example 7.4 is considered again, i.e.

$$G(s) = \frac{2s + 1}{s^2 + 2s + 100}$$

along with the double S trajectory  $q(t)$  defined by

$$q_0 = 0, \quad q_1 = 10, \quad v_{max} = 10, \quad a_{max} = 50, \quad j_{max} = 100.$$

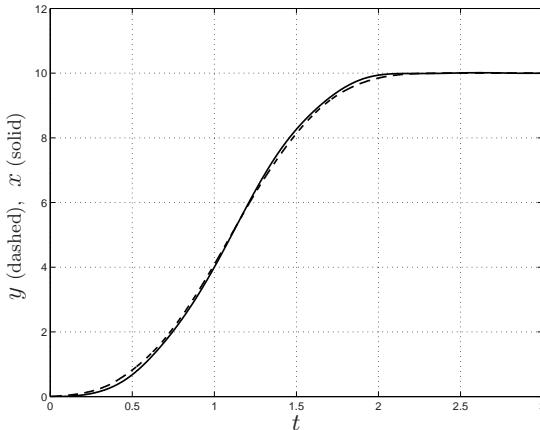
In particular, a three-hump shaper with a vibration limit  $\bar{z} = 5\%$  is taken into account. It is defined by four impulses:

$$\begin{aligned}
 \text{time instants} \quad t_0 &= 0, \quad t_1 = 0.3622, \quad t_2 = 0.6829, \quad t_3 = 0.9382 \\
 \text{amplitudes} \quad s_0 &= 0.3222, \quad s_1 = 0.4225, \quad s_2 = 0.2156, \quad s_3 = 0.0400.
 \end{aligned}$$

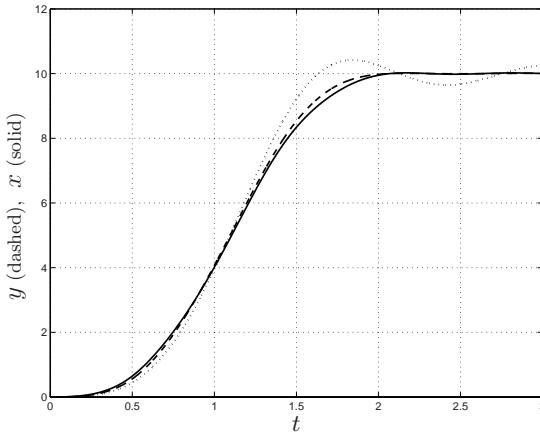
The response of the system  $G(s)$  to the trajectory  $q(t)$  filtered by the shaper is shown in Fig. 7.53 (compare with Fig. 7.49(a) where the output obtained without any filter is reported). If the model is affected by errors on the natural frequency or on the damping ratio, the approach based on the three-hump EI shaper shows an high robustness and keeps the vibrations at a very low level. For instance, in Fig. 7.54 the responses obtained with values of  $\omega_n$  different from the nominal one ( $\omega_n = 10$ ) are compared. In particular, the values  $\omega_n = 9$  (solid),  $\omega_n = 7$  (dashed),  $\omega_n = 5$  (dotted) are assumed, and despite a 50% variation the performances of the shaper remain acceptable.

□

If the system  $G(s)$  is characterized by multiple modes it is convenient to employ different impulse sequences, which increase the robustness of the filter.



**Fig. 7.53.** Response of the system  $G(s) = \frac{2s+1}{s^2+2s+100}$  to a double S trajectory with a three-hump EI input shaper.



**Fig. 7.54.** Output of three mechanical systems, which differ from the model  $G(s)$  used for the computation of a three-hump EI shaper in the value of the natural frequency. In particular, the values  $\omega_n = 9$  (solid),  $\omega_n = 7$  (dashed),  $\omega_n = 5$  (dotted) are assumed.

**Example 7.6** When the double S trajectory  $q(t)$  defined by

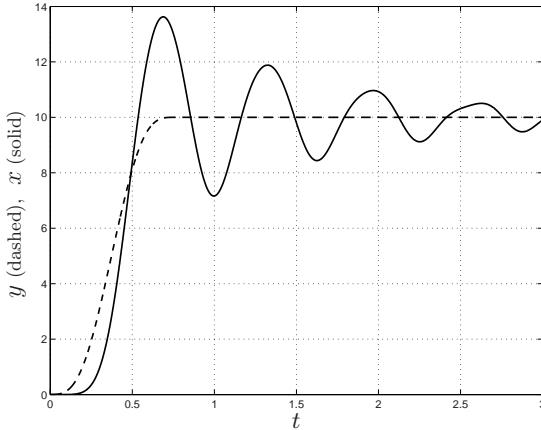
$$q_0 = 0, \quad q_1 = 10, \quad v_{max} = 40, \quad a_{max} = 120, \quad j_{max} = 800$$

is applied to the system

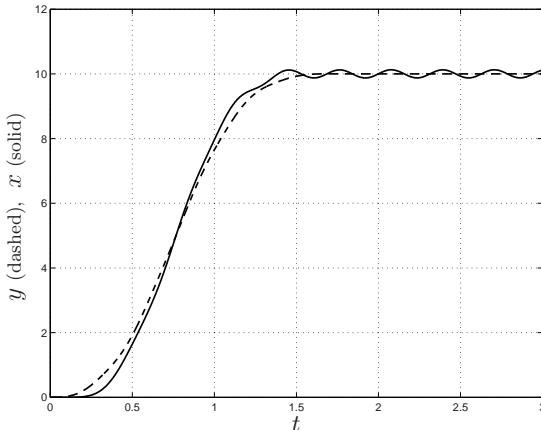
$$G(s) = 800 \frac{s + 50}{(s^2 + 2s + 100)(s^2 + 400)} \quad (7.23)$$

the oscillating output of Fig. 7.55 is obtained.

Since the system is characterized by two second-order modes defined by the



**Fig. 7.55.** Output of the system  $G(s) = 800 \frac{s+50}{(s^2+2s+100)(s^2+400)}$  when a double S trajectory is applied, without input shaper.



**Fig. 7.56.** Response of the system  $G(s) = 800 \frac{s+50}{(s^2+2s+100)(s^2+400)}$  to a double S trajectory with a ZVDD input shaper based on the first mode of the system.

pairs<sup>7</sup> ( $\omega_{n1} = 10$ ,  $\delta_1 = 0.1$ ) and ( $\omega_{n2} = 20$ ,  $\delta_2 = 0$ ), the use of an input shaper, whose design is only based on the first mode, it is not sufficient to cancel the vibrations which affect the output of  $G(s)$ . The response of  $G(s)$  obtained with the input filtered by a ZVDD shaper is shown in Fig. 7.56. In this case, one can add a second input shaper, as depicted in Fig. 7.57, obtaining the response of Fig. 7.58. A further delay is added to the system but the residual vibrations are completely suppressed. □

---

<sup>7</sup> Note that the first mode is the same of the system considered in Example 7.4.

The shapers considered in this section have an interesting interpretation if they are analyzed in the digital domain, i.e. by means of the Z-transform, [73]. For instance, in the case of the ZVD shaper of the Example 7.4, by assuming a sampling period  $T_s$  equal to  $T_0/2$ , the corresponding transfer function is

$$G_{\text{ZVD}}(z) = s_0 + s_1 z^{-1} + s_2 z^{-2} = \frac{0.3344z^2 + 0.4877z + 0.1778}{z^2}.$$

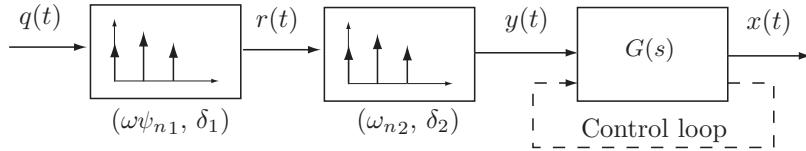
The zeros of the shaper ( $z_1 = -0.7292$ ,  $z_2 = -0.7292$ ) cancels the poles of the system, described by the discrete-time transfer function

$$G(z) = \frac{1.729z + 1.261}{z^2 + 1.458z + 0.5318}$$

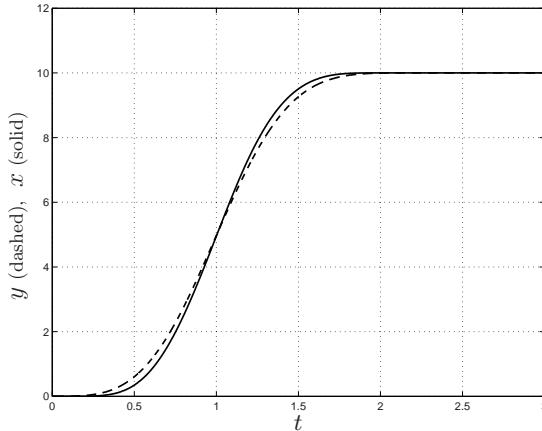
obtained by discretizing  $G(s)$  with  $T_s$ . More generally, given a system which must follow a prescribed trajectory without oscillations, and described by the discrete transfer function  $G(z)$ , it is possible to design a *Digital Shaping Filter (DSF)*, by assuming

$$G_{\text{DSF}} = \frac{C}{z^r} (z - p_1)^{n_1} (z - p_1^*)^{n_1} \cdots (z - p_m)^{n_m} (z - p_m^*)^{n_m}$$

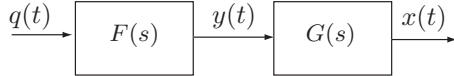
where the pairs  $(p_i, p_i^*)$  represent the (complex conjugate) poles of  $G(z)$ ,  $C$  is a constant and  $r = 2(n_1 + n_2 + \dots + n_m)$  is the total number of poles of the



**Fig. 7.57.** Block diagram of a system with a cascade of two shapers.



**Fig. 7.58.** Response of the system  $G(s) = 800 \frac{s+50}{(s^2+2s+100)(s^2+400)}$  to a double S trajectory filtered by a cascade of two ZVDD shapers.



**Fig. 7.59.** Cascade connection of the feedforward controller and of the plant.

system, [74]. Therefore, the design of a DSF is based on pole-zero cancellations according to a procedure not dissimilar from the feedforward control based on model inversion described in the next section.

#### 7.4.3 Feedforward based on the inversion of the plant dynamics

If the main mechanical properties of the machine are known and if the actuator's bandwidth is sufficiently large, more general approaches for the compensation of undesired effects, such as oscillations, can be considered. These approaches are based on an *inversion* of the dynamic model of the system.

Let us consider the simple case of a SISO (Single Input Single Output) linear<sup>8</sup> system, modelled with the transfer function  $G(s)$ . The cascade connection of the feedforward controller  $F(s)$  and of  $G(s)$  shown in Fig. 7.59 leads to the input/output transfer function

$$\frac{X(s)}{Q(s)} = F(s) G(s)$$

where  $Q(s)$  and  $X(s)$  are the Laplace transforms of the input trajectory  $q(t)$  and of the system response  $x(t)$  respectively. In order to have  $x(t) = q(t)$ ,  $\forall t$ , it is then necessary that

$$F(s) = G^{-1}(s).$$

This means that the dynamic behavior of the feedforward controller must be equal to the inverse of the dynamics of the plant.

The same results can be obtained also in presence of a feedback control  $C(s)$ , see Fig. 7.60. As a matter of fact, the relation between the input  $q(t)$  and the variable error  $e(t) = q(t) - x(t)$  is, in terms of Laplace transforms,

$$\frac{E(s)}{Q(s)} = \frac{1 - F(s)G(s)}{1 + C(s)G(s)} = S(s) \left(1 - F(s)G(s)\right)$$

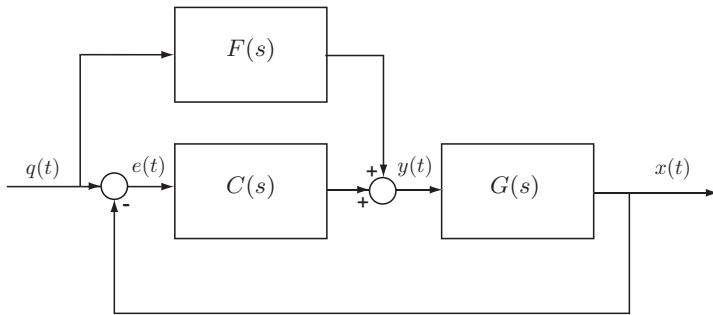
where  $S(s) = \left(1 + C(s)G(s)\right)^{-1}$  is the sensitivity function of the controlled system. Therefore, also in this case the conditions

$$F(s) = G^{-1}(s)$$

guarantees that the tracking error is null for  $t \geq 0$ .

---

<sup>8</sup> For the sake of simplicity a linear model is considered, but similar results are also valid for nonlinear systems, see e.g. [75, 76].



**Fig. 7.60.** Feedback/feedforward control of a SISO linear system.

Despite the solution of the perfect tracking problem is conceptually simple, its practical implementation is made difficult by uncertainties, delays and/or non-minimum phase dynamics.

Given a generic linear system

$$G(s) = K_1 \frac{b(s)}{a(s)} = K_1 \frac{s^m + b_{m-1}s^{m-1} + \dots + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_0}, \quad K_1 \neq 0 \quad (7.24)$$

with relative degree  $\rho = n - m$ , where it is assumed that the polynomials  $a(s)$  and  $b(s)$  are coprime (no pole-zero cancellations occur), it is convenient to express its inverse as

$$G^{-1}(s) = \frac{1}{K_1} \frac{a(s)}{b(s)} = c_\rho s^\rho + c_{\rho-1} s^{\rho-1} + \dots + c_0 + G_0(s) \quad (7.25)$$

where  $G_0(s)$  is a strictly proper rational transfer function representing the zero dynamics of the system (7.24), [77]. By using the fraction expansion,  $G_0(s)$  can be rewritten as

$$G_0(s) = G_0^-(s) + G_0^+(s) = \frac{d(s)}{b^-(s)} + \frac{e(s)}{b^+(s)}$$

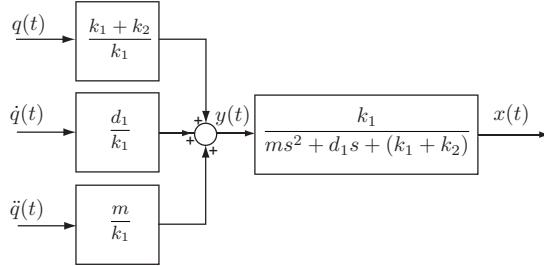
where  $b^-(s)$  and  $b^+(s)$  are the monic polynomials containing the roots of  $b(s)$  with negative and positive real parts, respectively.

The profile  $y(t)$ , which guarantees that  $x(t) = q(t)$ , can be computed by means of the inverse Laplace transform applied to

$$Y(s) = G^{-1}(s)Q(s).$$

By considering (7.25), it is possible to show that a bounded and continuous function which solves the problem of the dynamic inversion is

$$y(t) = c_\rho q^{(\rho)}(t) + \dots + c_1 q^{(1)}(t) + c_0 q(t) + \int_0^t \gamma_0^-(t-\tau) q(\tau) d\tau - \int_t^{+\infty} \gamma_0^+(t-\tau) q(\tau) d\tau \quad (7.26)$$



**Fig. 7.61.** Feedforward action for the tracking of the desired motion profile  $q(t)$ , with the mechanical system in Fig. 7.34.

where  $\gamma_0^-(t)$  and  $\gamma_0^+(t)$  are the inverse Laplace transforms of  $G_0^-(s)$  and  $G_0^+(s)$  respectively, [78]. The continuity of  $y(t)$  requires that the target trajectory profile  $q(t)$  belongs to the class of  $C^\rho$  bounded functions.

If the mechanical system has no zeros, the computation of the feedforward action is particularly simple, being the integral terms in (7.26) not present. As a matter of fact, in this case the signal  $y(t)$  which guarantees a perfect tracking is a linear combination of  $q(t)$  and of its first  $\rho$  derivatives, as already discussed in Sec. 7.4.1 with the polydyne approach.

**Example 7.7** Consider the linear and time-invariant mechanical system of Fig. 7.34, whose transfer function is

$$G(s) = \frac{k_1}{ms^2 + d_1s + (k_1 + k_2)}.$$

In this case, the feedforward filter is

$$F(s) = G^{-1}(s) = \frac{ms^2 + d_1s + (k_1 + k_2)}{k_1}$$

and, if the desired motion  $q(t)$  is known along with its derivative up to the second order, the equation (7.26) provides the expression of the desired input signal:

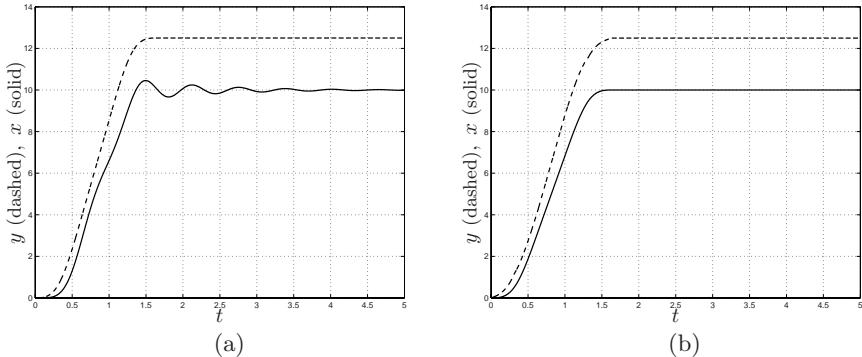
$$y(t) = \frac{m}{k_1} \ddot{q}(t) + \frac{d_1}{k_1} \dot{q}(t) + \frac{k_1 + k_2}{k_1} q(t). \quad (7.27)$$

Note that the “ $s$ ” operator in the Laplace domain is equivalent to the derivative action in the time domain.

From (7.27), it results that a continuous input function  $y(t)$  can be obtained if the trajectory  $q(t)$  is continuous at least up to its second derivative (note that the relative degree of the system is  $\rho = 2$ ).

By applying the profile  $y(t)$  to the system  $G(s)$ , as shown in Fig. 7.61, the output results equal to  $q(t)$ . Fig. 7.62(a) shows the output of  $G(s)$  with the parameters

$$m = 1 \text{ kg}, \quad d_1 = 2 \text{ Ns/m}, \quad k_1 = 80 \text{ N/m}, \quad k_2 = 20 \text{ N/m}$$



**Fig. 7.62.** Input and output signals of the system  $G(s) = \frac{80}{s^2+2s+100}$  without (a) and with (b) model inversion;  $q(t)$  is a double S trajectory.

when a double S profile<sup>9</sup>  $q(t)$  is directly applied to the input (multiplied by the gain  $(k_1 + k_2)/k_1$ ). In Fig. 7.62(b) the output of the same system is considered, with the input  $y(t)$  obtained according to (7.27). Note that in this case  $x(t) = q(t)$  without any oscillation.  $\square$

The presence of “stable” zeros (i.e. with negative real part) makes the dynamic inversion more complex since, besides the linear combination of  $q(t)$  and its derivatives, it is necessary to consider the convolution between  $q(t)$  and  $\gamma_0^-(t)$ :

$$\int_0^t \gamma_0^-(t-\tau)q(\tau)d\tau. \quad (7.28)$$

This term represents a post-action, whose duration  $T_p$  can be computed with arbitrarily precision as  $T_p = t_p - t_0$ , being  $t_0$  the instant at which the trajectory is applied and

$$t_p := \min \left\{ \tau \in \mathbb{R} : \left| y(t) - \frac{1}{G(0)} \right| \leq \epsilon_p, \forall t \in [\tau, \infty) \right\}$$

with  $\epsilon_p$  an arbitrary small parameter, [77].

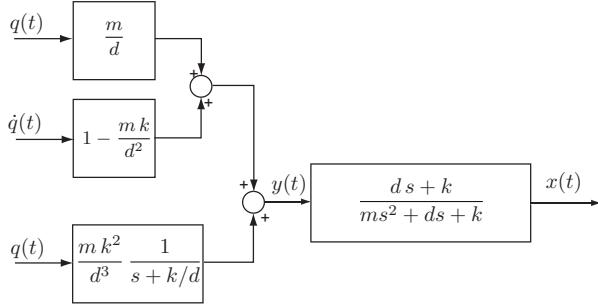
Once the trajectory  $q(t)$  has been fixed, the contribution (7.28) can be estimated by computing the integral in a closed form or in a numerical way, or, equivalently, by considering the output of the stable linear filter  $G_0^-(s)$  with the profile  $q(t)$  as input.

<sup>9</sup> The double S trajectory considered in all the examples of this section is obtained with the conditions

$$q_0 = 0, \quad q_1 = 10,$$

$$v_{max} = 10, \quad a_{max} = 50, \quad j_{max} = 100.$$

The total duration of the trajectory is  $T = 1.585$  s.



**Fig. 7.63.** Feedforward action for the tracking of the desired motion profile  $q(t)$ , with the mechanical system in Fig. 7.1.

**Example 7.8** Given the one-dof elastic system already considered in Sec. 7.1.1 and described by the transfer function

$$G(s) = \frac{ds + k}{ms^2 + ds + k}$$

where the meaning of the parameters  $m$ ,  $d$ ,  $k$  is explained in Fig. 7.1, the feedforward filter is

$$F(s) = G^{-1}(s) = \left[ \frac{m}{d}s + \left( 1 - \frac{mk}{d^2} \right) \right] + G_0^-(s)$$

with

$$G_0^-(s) = \frac{mk^2}{d^3} \frac{1}{s + k/d}.$$

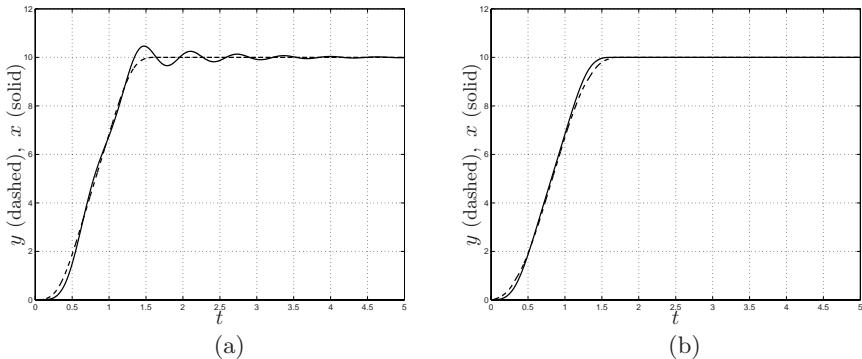
Therefore, the signal  $y(t)$  which guarantees a perfect tracking of the trajectory  $q(t)$  is composed by two terms. The former is represented by a linear combination of  $q(t)$  and  $\dot{q}(t)$ , while the latter is given by the trajectory  $q(t)$  filtered by  $G_0^-(s)$  (see Fig. 7.63):

$$y(t) = \left[ \frac{m}{d}\dot{q}(t) + \left( 1 - \frac{mk}{d^2} \right) q(t) \right] + \int_0^t \frac{mk^2}{d^3} e^{-\frac{k}{d}(t-\tau)} q(\tau) d\tau. \quad (7.29)$$

Differently from Example 7.7, in this case a continuous input function  $y(t)$  can be obtained if the trajectory  $q(t)$  is continuous up to the first derivative. As a matter of fact, the relative degree  $\rho$  of the system is equal to one. With the values of the parameters

$$m = 1 \text{ Kg}, \quad d = 2 \text{ Ns/m}, \quad k = 100 \text{ N/m}$$

the response of  $G(s)$  to the input  $y(t) = q(t)$  is shown in Fig. 7.64(a), while the output of  $G(s)$ , when the  $y(t)$  is computed as in (7.29), is reported in Fig. 7.64(b).  $\square$



**Fig. 7.64.** Input and output signals of the system  $G(s) = \frac{2s+100}{s^2+2s+100}$  without (a) and with (b) model inversion.

If the zeros of the transfer function are characterized by a positive real part, the model inversion is more complicated since, besides the post-action due to the presence of stable zeros, if any, it is necessary to consider a pre-action, whose contribution begins at

$$t_a := \min \{ \tau \in \mathbb{R} : |y(t)| \leq \epsilon_a, \forall t \in (-\infty, \tau] \}$$

where, as in the case of the post-action,  $\epsilon_a$  is an arbitrary small parameter. The analytical expression of the pre-action is

$$-\int_t^{+\infty} \gamma_0^+(t-\tau) q(\tau) d\tau$$

and can be estimated by computing (or approximating, [78]) the integral. Note that, in this case the integral form of the pre-action is not equivalent to the output of the filter  $G_0^+(s)$ , since this one is unstable.

**Example 7.9** Consider the linear SISO system [77]

$$G(s) = 4 \frac{(1-s)(s+1)}{(s+2)(s^2+2s+2)}$$

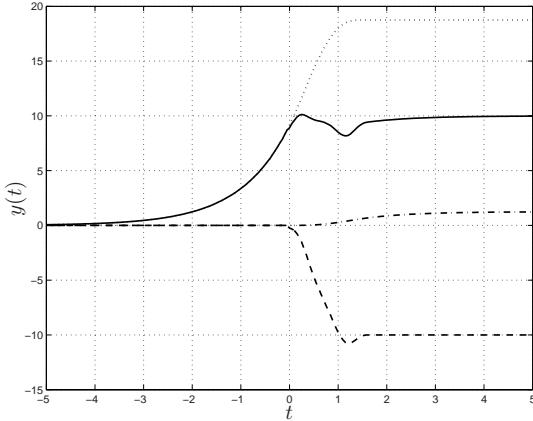
which leads to the feedforward filter

$$F(s) = G^{-1}(s) = \left[ -\frac{1}{4}s - 1 \right] + G_0^-(s) + G_0^+(s)$$

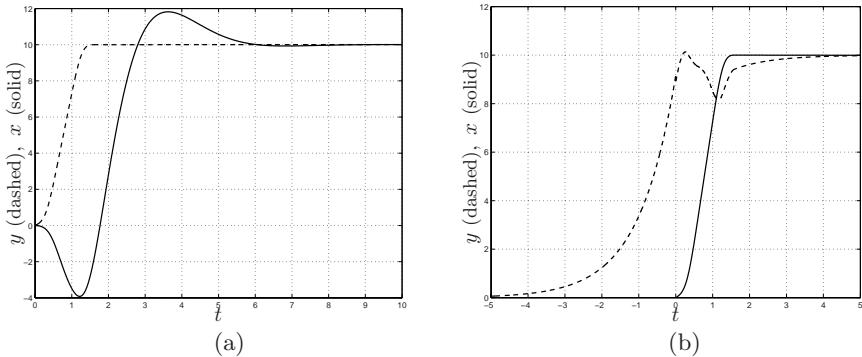
where

$$G_0^-(s) = \frac{\frac{1}{8}}{s+1}, \quad G_0^+(s) = \frac{-\frac{15}{8}}{s-1}.$$

As shown in Fig. 7.65, the feedforward control  $y(t)$  which guarantees a perfect



**Fig. 7.65.** Feedforward signal based on model inversion (solid), decomposed in its components: pre-action (dotted), post-action (dashdot), linear combination of  $q$  and  $\dot{q}$  (dashed).



**Fig. 7.66.** Input and output signals of the system  $G(s) = 4 \frac{(1-s)(s+1)}{(s+2)(s^2+2s+2)}$  without (a) and with (b) model inversion.

tracking of the trajectory  $q(t)$  is composed by three terms. The first is represented by a linear combination of  $q(t)$  and  $\dot{q}(t)$ , while the others are the pre- and post- actions due to the zero dynamics of the system:

$$y(t) = \left[ -\frac{1}{4} \dot{q}(t) - q(t) \right] + \frac{15}{8} \int_t^{+\infty} e^{(t-\tau)} q(\tau) d\tau + \frac{1}{8} \int_0^t e^{-(t-\tau)} q(\tau) d\tau.$$

In Fig. 7.66 the output of the system  $G(s)$  is shown, when the double S trajectory is applied to the input Fig. 7.66(a), and when the feedforward filter is adopted Fig. 7.66(b). Note that, in this latter case, the input  $y(t)$  starts before the application of the trajectory  $q(t)$  and ends after the trajectory has completed.  $\square$

Ideally, the approach based on feedforward compensation, derived from the automatic control field, would allow to compensate for many non-idealities of the mechanical system. Unfortunately, it is often limited in applications since it is necessary to know the exact values of the parameters of the model and, moreover, the undesired effects are usually at high frequencies where it is not possible to apply any proper corrective actions by means of the actuation system.

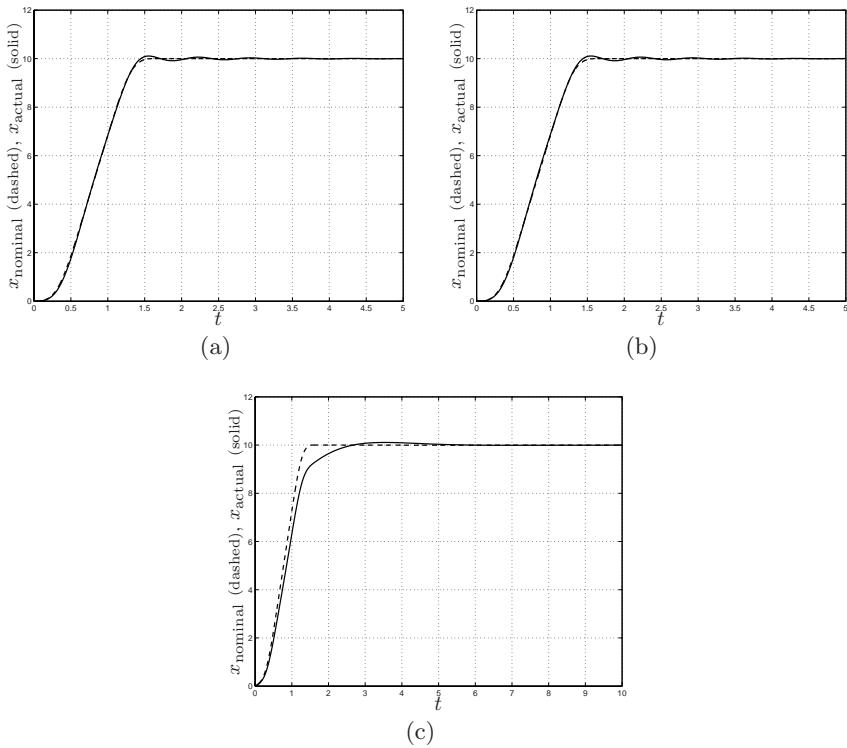
**Example 7.10** The three cases considered in the previous examples of this section are now analyzed by supposing that the parameters of the system  $G(s)$  differ from their nominal values. In particular, instead of the models  $G(s)$  new models  $G_a(s)$  are assumed, whose expressions<sup>10</sup> are respectively

$$\begin{aligned} \text{a) } G(s) &= \frac{80}{s^2 + 2s + 100} & \Rightarrow G_a(s) &= \frac{64.8}{s^2 + 1.8s + 81} \\ \text{b) } G(s) &= \frac{2s + 100}{s^2 + 2s + 100} & \Rightarrow G_a(s) &= \frac{2s + 81}{s^2 + 1.8s + 81} \\ \text{c) } G(s) &= 4 \frac{(1-s)(s+1)}{(s+2)(s^2 + 2s + 2)} & \Rightarrow G_a(s) &= 3.24 \frac{(1-s)(s+1)}{(s+2)(s^2 + 1.8s + 1.62)}. \end{aligned}$$

In Fig. 7.67 the responses obtained with the nominal models (dashed) are compared with those provided by  $G_a(s)$ , with the same inputs  $y(t)$  (solid). While in the first two cases the outputs of  $G_a(s)$  and of  $G(s)$  remain quite similar, in the last case the responses are considerably different.  $\square$

---

<sup>10</sup> In the three cases, a variation of  $-10\%$  on the natural frequency of the first mode is assumed.



**Fig. 7.67.** Response of the systems considered in the Examples 7.7, 7.8 and 7.9, when the actual system is different from the nominal model used to compute the feedforward action  $y(t)$ .

## **Part III**

---

### **Trajectories in the Operational Space**

## Multidimensional Trajectories and Geometric Path Planning

In this Chapter, the problem of trajectory planning in 3D is addressed. This problem is of relevance in operations with multi-degrees of freedom machines, like for example robots or special sanding/milling machines. Planning motions in the 3D space is more complex than in the single axis case, since in general two aspects must be defined: the *geometry* of the trajectory (e.g. a straight line, a circle, and so on), and the *motion law* to be adopted while following the geometric path. Moreover, also the problem of the orientation (of the tool) has to be considered. Therefore, there are at least seven variables to be specified for a 3D trajectory: three for the position, three for the orientation and one for the motion law.

Besides these aspects, in order to execute a desired motion in the space by means of a multi-degree-of-freedom machine, it is also necessary to consider the fact that a (inverse) kinematic model of the machine must be used in order to transform the given trajectory from the 3D space, where it is specified, to the space of actuation (usually called the *joint space*) where the motion of the actuators takes place. In general, this kinematic transformation may be quite complex and depends on the particular machine or robot at hand.

---

### 8.1 Introduction

The definition of a trajectory in the Cartesian (3D) space implies the determination of a geometric path to be tracked with a prescribed motion law, that in turns can be defined by means of functions similar to those reported in the previous chapters. For this purpose, it is convenient to consider a parametric representation of a curve in the space

$$\mathbf{p} = \mathbf{p}(u), \quad u \in [u_{min}, u_{max}] \quad (8.1)$$

where  $\mathbf{p}(\cdot)$  is a  $(3 \times 1)$  a continuous vectorial function, which describes the curve when the independent variable  $u$  ranges over some interval of the domain space.

In many cases, the definition of a trajectory in the task space of a robot or of a multi-axis automatic machine requires also to assign the orientation of the tool in each point of the curve. This can be achieved by specifying the configuration of the frame linked to the end effector (*the tool frame*) with respect to the *base (world) frame*. Therefore, in the general case the parametric description of the trajectory (8.1) is a six dimensional function<sup>1</sup> providing, for each value of the variable  $u$ , both the position and the orientation of the tool:

$$\mathbf{p} = [x, y, z, \alpha, \beta, \gamma]^T.$$

Therefore, the planning of a trajectory in the workspace consists in defining:

1. The function  $\mathbf{p}(u)$ , which interpolates a set of desired points/configurations.
2. The motion law  $u = u(t)$  describing how the tool should move along the path.

In case the multi-dimensional problem can be decomposed in its components, the 3D trajectory planning can be considered as a set of scalar problems, and the techniques reported in the previous chapters can be adopted for its solution. In this case, each function  $p_i(\cdot)$  depends directly on the time  $t$ , and the synchronization among the different components is performed by imposing interpolation conditions at the same time instants.

In this chapter, the multi-dimensional problem is approached by considering the computation of the 3D geometric path to be tracked with a prescribed motion law.

Once the trajectory in the task space has been defined, it is necessary to translate it in the joint/motor space by means of the inverse kinematic model of the system. The reader interested to this topics should refer to the specialized literature, see for example [12] and the many other excellent textbooks [79, 80, 81].

Before discussing the techniques that can be used to define the function  $\mathbf{p}(u)$ , some preliminary considerations and definitions are necessary. Quite often, the trajectories for the position and for the orientation are defined separately, since it can be desirable to track a well defined path in the work space, e.g. a straight line, with the orientation of the tool specified only at the endpoints of the motion. In fact, with the exception of some applications (e.g. welding, painting, etc.), not always a strict relation between position and orientation exists.

Although the two problems can be treated separately, they result conceptually

---

<sup>1</sup> In case that a minimal representation of the orientation is assumed, e.g. Euler or Roll-Pitch-Yaw angles, only three parameters are necessary, see Appendix C.

similar, i.e. given a set of via-points<sup>2</sup>  $\mathbf{q}_k = [x_k, y_k, z_k]^T$  (position) or  $\mathbf{q}_k = [\alpha_k, \beta_k, \gamma_k]^T$  (orientation) it is necessary to find a parametric curve which passes through o near them. In simple cases, such a function can be directly provided in an analytical manner by means of circular/straight line motion primitives. More frequently, it must be constructed by adopting more complex approaches that guarantee also a desired smoothness (continuity of the curve and its derivatives up to a desired order). In this case, classical approaches are based on B-spline functions, Bézier curves or Nurbs [82], which are *piecewise* polynomial functions defined by

$$\mathbf{p}(u) = \sum_{j=0}^m \mathbf{p}_j B_j(u) \quad (8.2)$$

where  $\mathbf{p}_j$  are the so called *control points*, i.e. constant coefficients which determine the shape of the curve by weighting the basis functions  $B_j(u)$ , properly defined according to the type of curve used. The definition and the most significant properties of B-spline, Bézier and Nurbs curves are summarized in Appendix B.

### 8.1.1 Continuity of the geometric path and continuity of the trajectory

As shown in Fig. 8.1, a geometric path is usually composed by a number of segments, i.e.

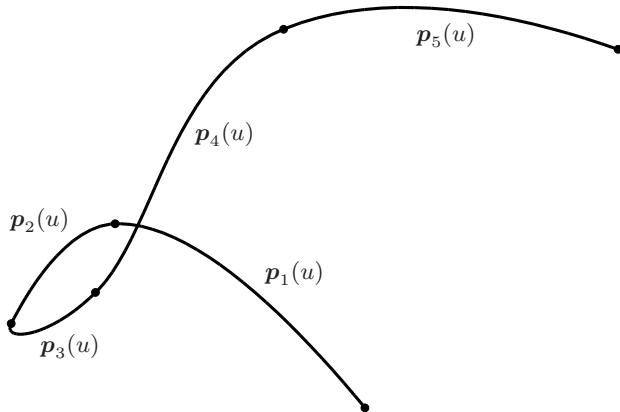
$$\mathbf{p}(u) = \mathbf{p}_k(u), \quad k = 0, \dots, n - 1.$$

It is therefore necessary to guarantee the smoothness of the curve by imposing continuity constraints at the *joints*, i.e. the locations where the segments<sup>3</sup> composing the curve abut, see Fig. 8.2. For this purpose, it is necessary to specify the meaning of *continuity*, since different notions exist about this property for motions in 3D [83, 84]. In particular, two types of continuity are of interest for a trajectory in the Cartesian space: the *geometric* and the *parametric* continuity. As a matter of fact, the use of a parametric curve for trajectory planning generally requires, besides the obvious continuity of the geometric path, that also the velocity and acceleration vectors, which are respectively the first and the second parametric derivative vectors  $\left(\frac{d\mathbf{p}}{du}, \frac{d^2\mathbf{p}}{du^2}\right)$  of the curve, are continuous.

On the other hand, if proper care is not taken, it may happen that the curve, although geometrically smooth, is discontinuous in speed or in acceleration.

<sup>2</sup> A *via-point* is a point (configuration) in the 3D space used to plan the trajectory. The via-points may be either interpolated or approximated, see the following sections.

<sup>3</sup> For the sake of simplicity it is usually assumed that each segment is defined in the interval  $u \in [0, 1]$ .



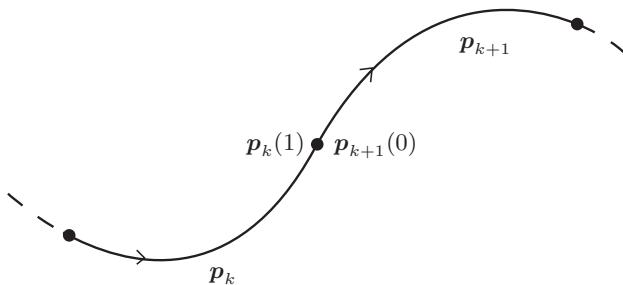
**Fig. 8.1.** Piecewise polynomial curve characterized by  $G^1$  continuity.

In this case the curve is said *geometrically continuous* but *parametrically discontinuous*. Let us consider the example of the (apparently continuous) path of Fig. 8.1. Its first derivative is shown in Fig. 8.4(a), and it is clear that with the adopted parameterization the derivative is discontinuous although the geometric path appears to be smooth.

Two infinitely differentiable segments meeting at a common point, i.e.  $\mathbf{p}_k(1) = \mathbf{p}_{k+1}(0)$ <sup>4</sup> like in Fig. 8.2, are said to meet with *n-order parametric continuity*, denoted by  $C^n$ , if the first  $n$  parametric derivatives match at the common point, that is if

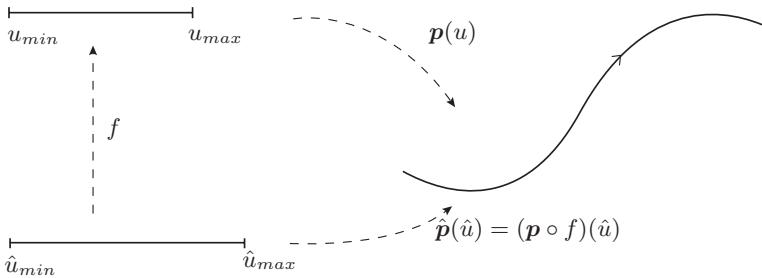
$$\mathbf{p}_k^{(i)}(1) = \mathbf{p}_{k+1}^{(i)}(0), \quad k = 1, \dots, n.$$

Unfortunately, derivative vectors are not intrinsic properties of a curve [83], and their value changes by substituting the parameterization  $u$  with an equiva-



**Fig. 8.2.** Two parametric curves meeting at a common point, the *joint*.

<sup>4</sup> It is assumed  $u \in [0, 1], \forall k$ .



**Fig. 8.3.** Reparameterization of a curve  $p(u)$ .

lent<sup>5</sup> one  $\hat{u}$ . Therefore two curves meeting at a point with a  $C^n$  continuity may lose this property if the parametrization of one of them is changed. Conversely, the unit tangent  $\left(\frac{dp}{du} / \left|\frac{dp}{du}\right|\right)$  and the unit curvature  $\left(\frac{d^2p}{du^2} / \left|\frac{d^2p}{du^2}\right|\right)$  vectors are intrinsic properties of the curve, and they lead to the notion of geometric continuity. Two parametric curves meet with a first order geometric continuity, denoted by  $G^1$ , if and only if they have a common unit tangent vector. In this case the tangent direction at the joint is preserved, but the continuity of the velocity vector is not guaranteed, since the tangent vectors may have different magnitude.

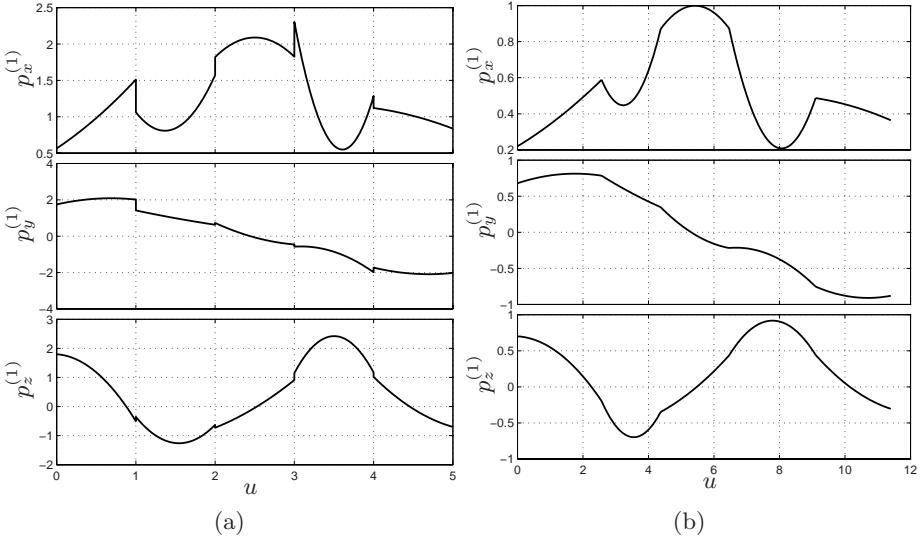
Two curves meet with  $G^2$  continuity if and only if they have common unit tangent and curvature vectors.

More generally, it is possible to state that two segments  $p_k(u)$  and  $p_{k+1}(u)$  meet with  $n$ -order geometric continuity ( $G^n$  continuity) if and only if there exists a parameterization  $\hat{u}$  equivalent to  $u$  such that  $\hat{p}_k(\hat{u})$  and  $p_{k+1}(\hat{u})$  meet with  $C^n$  continuity at the joint. This means that, given a  $G^n$  curve composed by several segments, it is possible to find a parameterization which makes the curve  $C^n$  continuous, guaranteeing in this way the continuity of speed, acceleration, and so on.

**Example 8.1** The trajectory of Fig. 8.1, composed by five polynomial segments (Bézier curves), is  $G^1$  continuous. From a visual point of view, the path appears to be smooth, but Fig. 8.4(a) reveals that the first derivative is discontinuous at the joints. An equivalent parameterization is therefore used to make the curve  $C^1$  continuous, as shown in Fig. 8.4(b).  $\square$

<sup>5</sup> Two parameterizations (see Fig. 8.3) are said to be equivalent if there exists a regular  $C^n$  function  $f : [\hat{u}_{min}, \hat{u}_{max}] \mapsto [u_{min}, u_{max}]$  such that:

- 1)  $\hat{p}(\hat{u}) = p(f(\hat{u})) = p(u)$ .
- 2)  $f([\hat{u}_{min}, \hat{u}_{max}]) = [u_{min}, u_{max}]$ .
- 3)  $f^{(1)} > 0$ .

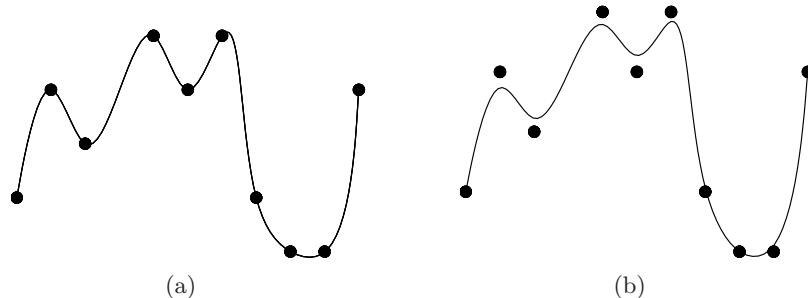


**Fig. 8.4.** Components along the  $x$ ,  $y$  and  $z$  directions of the tangent vector  $\mathbf{p}^{(1)}$  of the 3D trajectory shown in Fig. 8.1, with two different parameterizations.

### 8.1.2 Global and local interpolation/approximation

As mentioned above, a trajectory in the 3D space is usually constructed with the purpose of fitting a given set of data points. To this goal, different criteria can be adopted according to the needs imposed by the specific application. In particular, two types of fitting can be distinguished: *interpolation* and *approximation* (see Fig. 8.5)[85, 86]. If the points are interpolated, the curve passes exactly through them for some values of the independent variable. If they are approximated, the curve does not pass exactly through the given points, but in a neighborhood within a prescribed tolerance. This case is usually adopted when the trajectory must fit a large number of points, but the free parameters which characterize the curve are not sufficient to obtain an exact interpolation. In other applications approximation is preferable to interpolation, e.g. when the goal is to construct a curve reproducing the “shape” of the data, avoiding fast oscillations between contiguous points (reducing in this way the curvature/acceleration along the trajectory); this is the case of *smoothing B-splines*.

The interpolating/approximating curve can be determined by global or local procedures. With a global algorithm the parameters which define the trajectory, e.g. the control points in eq. (8.2), are computed by solving an optimization problem. This is based on the whole set of data points and usually allows to minimize some quantity, e.g. the curvature of the overall path. With a global algorithm, if some points are modified, the shape of the entire curve



**Fig. 8.5.** Interpolation (a) and approximation (b) of a set of data points.

is perturbed. Conversely, a local procedure is based only on local data (tangent vectors, curvature vectors, etc.) for each pair of points. These algorithms are generally computationally less expensive than global methods, but the achievement of the desired level of continuity at the joints is a quite tricky job. On the other hand, local methods allow to deal with corners, straight line segments, and other particularities in a simpler way, and the modification of a point involves only the two adjacent segments.

## 8.2 Orientation of the Tool

The orientation  $\mathcal{R}$  of the end effector<sup>6</sup> may be expressed by means of a rotation matrix composed by three orthogonal unit vectors (called in robotics *normal*, *slide*, and *approach*) defining the orientation of the tool with respect to the base world frame:

$$\mathbf{R} = [\mathbf{n}, \mathbf{s}, \mathbf{a}].$$

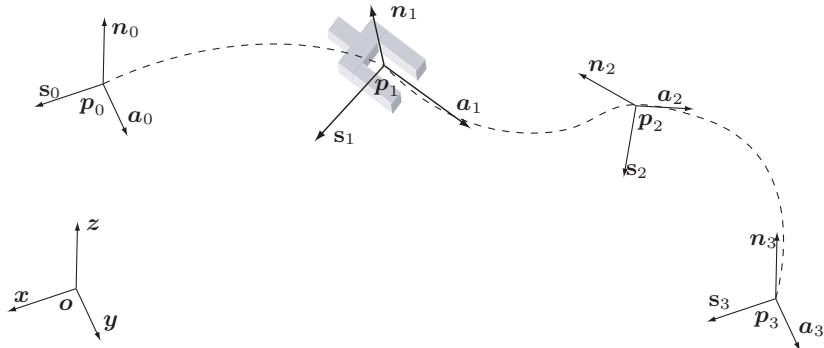
Therefore, in order to specify the orientation of the end effector in different points of the trajectory, a desired rotation matrix  $\mathbf{R}_k$  must be specified at each point  $\mathbf{p}_k$  and a proper interpolating technique must be used among them, see Fig. 8.6.

### 8.2.1 Case of independent position and orientation

In defining a trajectory, it is in general not convenient to adopt rotation matrices for the interpolation of the orientation among given poses. As a matter of fact, by interpolating the three unit vectors  $\mathbf{n}, \mathbf{s}, \mathbf{a}$ , from the initial value corresponding to the rotation matrix  $\mathbf{R}_0$  to the final one  $\mathbf{R}_1$ , it is not possible to guarantee that the orthonormality conditions<sup>7</sup> hold in every instant.

<sup>6</sup> An end effector is the (generic) tool carried by a robotic arm or the operating tool of a multi-axis machine.

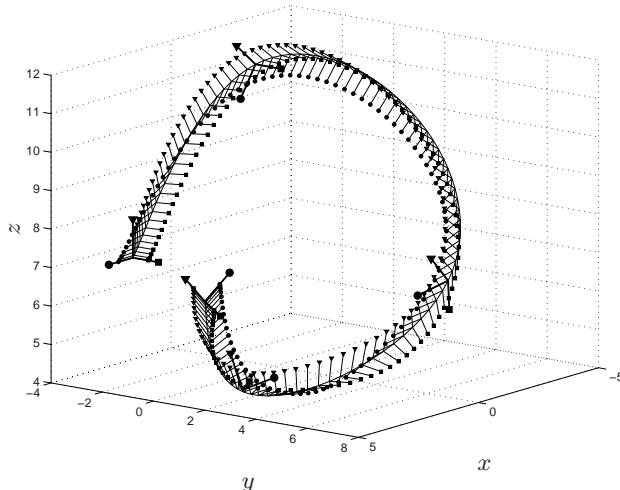
<sup>7</sup> See Appendix C.



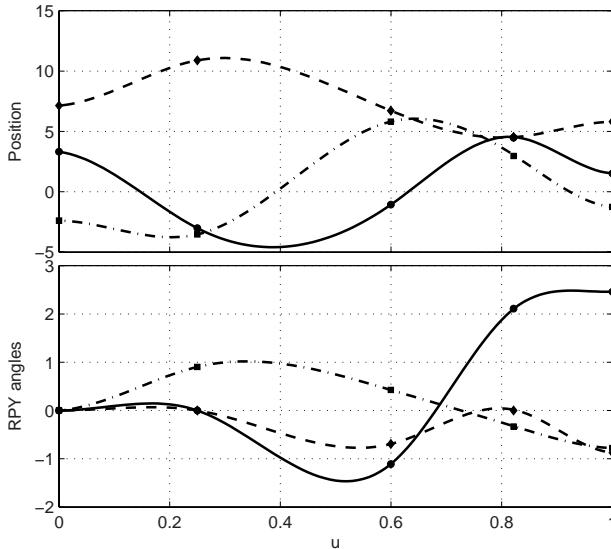
**Fig. 8.6.** Change of the orientation.

For this reason, the planning of a change of the orientation is often based on a set of three angles  $\phi = (\varphi, \vartheta, \psi)$ , e.g. the Euler or Roll-Pitch-Yaw angles, Appendix C, which are varied according to some laws (usually, a polynomial function of the time) from the initial value  $\phi_0$  to the final one  $\phi_1$ . In this way, a continuous variation of the orientation (and a continuous angular velocity) is obtained.

**Example 8.2** In Fig. 8.7 a trajectory with a displacement both in terms of position and orientation is shown. In particular, the orientation in each via-point is provided by means of Roll-Pitch-Yaw angles with respect to the world frame (in the figure the corresponding frames, computed according to



**Fig. 8.7.** Multipoint trajectory for position and orientation based on Roll-Pitch-Yaw angles representation.



**Fig. 8.8.** Trajectory for position and orientation in 3D space based on RPY angles representation: profiles of the different components (see Fig. 8.7).

eq. (C.13), are reported). The six-dimensional via-points are

$$\begin{bmatrix} q_x \\ q_y \\ q_z \\ \psi \\ \theta \\ \varphi \end{bmatrix} = \begin{bmatrix} 3.31 & -3.01 & -1.07 & 4.48 & 1.52 \\ -2.38 & -3.53 & 5.81 & 2.97 & -1.25 \\ 7.14 & 10.89 & 6.72 & 4.54 & 5.81 \\ 0 & 0 & -1.11 & 2.11 & 2.46 \\ 0 & 0.90 & 0.42 & -0.33 & -0.77 \\ 0 & 0 & -0.69 & 0 & -0.87 \end{bmatrix}.$$

By means of a suitable interpolation algorithm<sup>8</sup>, discussed in Sec. 8.4.2, a  $C^2$  curve passing through all the points is found. This guarantees that the angles change smoothly, and it is possible to demonstrate that both the corresponding angular velocities and angular accelerations are continuous. The components of the trajectory (position and orientation) are shown in Fig. 8.8.  $\square$

An alternative approach, that in the Cartesian space assumes a more clear meaning, is based on the so-called *angle-axis representation*. Given two frames with coincident origins but different orientations, it is always possible to find a vector  $\mathbf{w}$  such that the latter frame can be obtained from the former one by means of a rotation of an angle  $\vartheta$  about  $\mathbf{w}$ . Given the initial frame  $\mathcal{F}_0$

<sup>8</sup> Note that all the interpolation and approximation algorithms reported in the next sections are in general suitable for  $n$ -dimensional problems.

(represented in the base frame by the rotation matrix  $\mathbf{R}_0$ ) and the final frame  $\mathcal{F}_1$  (with the associated matrix  $\mathbf{R}_1$ ), the rotation matrix describing the transformation is

$${}^0\mathbf{R}_1 = \mathbf{R}_0^T \mathbf{R}_1 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (8.3)$$

i.e.  $\mathbf{R}_1 = \mathbf{R}_0 {}^0\mathbf{R}_1$ . Matrix  ${}^0\mathbf{R}_1$  can be expressed as the rotation matrix about the fixed axis

$$\mathbf{w} = \frac{1}{2 \sin \theta_t} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (8.4)$$

of the angle

$$\theta_t = \cos^{-1} \left( \frac{r_{11} + r_{22} + r_{33} - 1}{2} \right). \quad (8.5)$$

The rotational motion of the tool from  $\mathbf{R}_0$  to  $\mathbf{R}_1$  can be described as

$$\mathbf{R}(t) = \mathbf{R}_0 \mathbf{R}_t(\theta(t))$$

where  $\mathbf{R}_t(\theta(t))$  is a time depending matrix such that  $\mathbf{R}_t(0) = \mathbf{I}_3$ , the  $(3 \times 3)$  identity matrix, and  $\mathbf{R}_t(\theta_t) = {}^0\mathbf{R}_1$ . The expression of matrix  $\mathbf{R}_t(\theta)$  is

$$\mathbf{R}_t(\theta) = \begin{bmatrix} w_x^2(1 - c_\theta) + c_\theta & w_x w_y(1 - c_\theta) - w_z s_\theta & w_x w_z(1 - c_\theta) + w_y s_\theta \\ w_x w_y(1 - c_\theta) + w_z s_\theta & w_y^2(1 - c_\theta) + c_\theta & w_y w_z(1 - c_\theta) - w_x s_\theta \\ w_x w_z(1 - c_\theta) - w_y s_\theta & w_y w_z(1 - c_\theta) + w_x s_\theta & w_z^2(1 - c_\theta) + c_\theta \end{bmatrix}$$

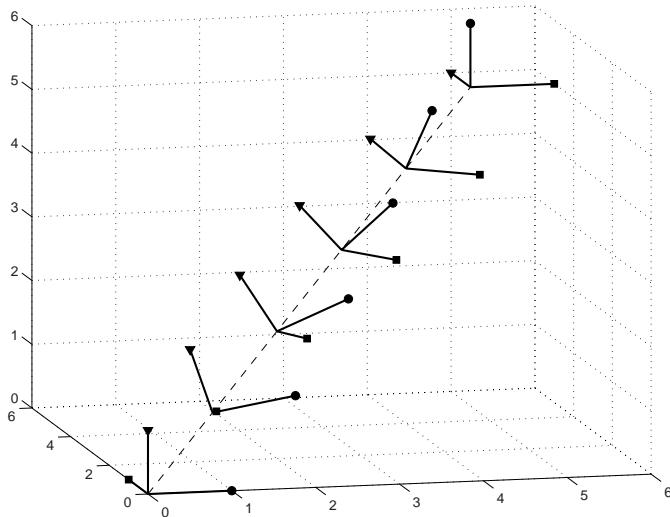
where  $c_\theta = \cos(\theta)$ ,  $s_\theta = \sin(\theta)$ . Therefore, the orientation of the tool in the transition from  $\mathbf{R}_0$  to  $\mathbf{R}_1$  depends on the parameter  $\theta$ . At this point it is only necessary to assign the motion law  $\theta(t)$ .

**Example 8.3** A change of orientation along a straight line trajectory from  $\mathbf{p}_0 = [0, 0, 0]^T$  to  $\mathbf{p}_1 = [5, 5, 5]^T$  is shown in Fig. 8.9. In particular the matrices associated to the initial and final frames are

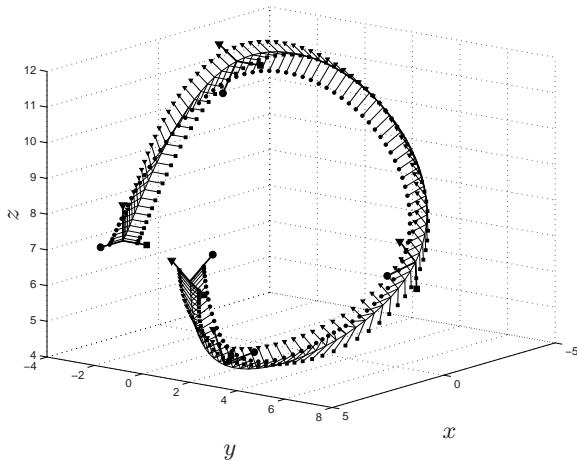
$$\mathbf{R}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

The transition from  $\mathbf{R}_0$  to  $\mathbf{R}_1$  can be expressed as a rotation of the angle  $\theta_t = 2.09 \text{ rad}$  about the vector  $\mathbf{w} = [-0.57, -0.57, -0.57]^T$ . In the figure, the tool frame is represented for different values of  $\theta$ , which is varied from 0 to  $\theta_t$  in a linear manner.

□



**Fig. 8.9.** Frame change obtained with an axis-angle representation.

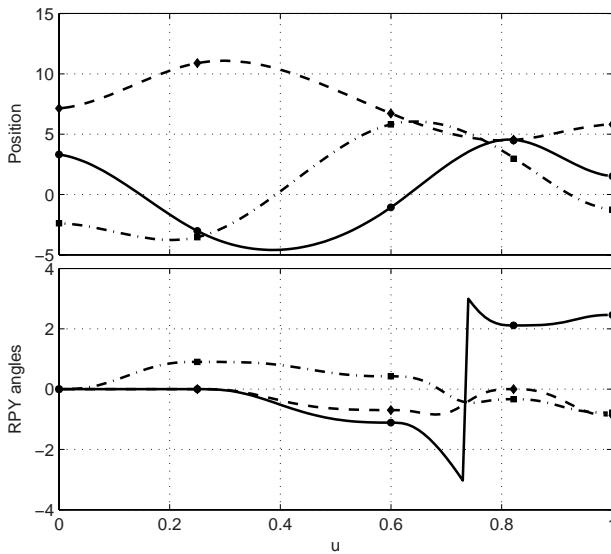


**Fig. 8.10.** Trajectory for position and orientation based on axis-angle representation.

**Example 8.4** A trajectory computed by means of the axis-angle representation is shown in Fig. 8.10. The orientation in each via-point is provided by means of a rotation matrix  $\mathbf{R}_k$  which defines the configuration of the local frame with respect to the world frame. The data points are the same as in Example 8.2. Therefore

$$\begin{aligned}
 \mathbf{q}_0 &= \begin{bmatrix} 3.31 \\ -2.38 \\ 7.14 \end{bmatrix}, & \mathbf{R}_0 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 \mathbf{q}_1 &= \begin{bmatrix} -3.01 \\ -3.53 \\ 10.89 \end{bmatrix}, & \mathbf{R}_1 &= \begin{bmatrix} 0.61 & 0 & 0.78 \\ 0.00 & 1 & 0.00 \\ -0.78 & 0 & 0.61 \end{bmatrix} \\
 \mathbf{q}_2 &= \begin{bmatrix} -1.07 \\ 5.81 \\ 6.72 \end{bmatrix}, & \mathbf{R}_2 &= \begin{bmatrix} 0.40 & 0.57 & 0.71 \\ -0.81 & 0.57 & 0.00 \\ -0.41 & -0.58 & 0.70 \end{bmatrix} \\
 \mathbf{q}_3 &= \begin{bmatrix} 4.48 \\ 2.97 \\ 4.54 \end{bmatrix}, & \mathbf{R}_3 &= \begin{bmatrix} -0.48 & -0.85 & 0.16 \\ 0.81 & -0.51 & -0.28 \\ -0.32 & 0.00 & 0.94 \end{bmatrix} \\
 \mathbf{q}_4 &= \begin{bmatrix} 1.52 \\ -1.25 \\ 5.81 \end{bmatrix}, & \mathbf{R}_4 &= \begin{bmatrix} -0.55 & -0.82 & -0.13 \\ 0.44 & -0.16 & -0.87 \\ 0.70 & -0.54 & 0.45 \end{bmatrix}.
 \end{aligned}$$

In this case the points describing the position are interpolated by means of a cubic B-spline, while the orientation is changed from the initial configuration to the final one by considering a pair of points at a time. Therefore the overall trajectory for the orientation is composed by four segments, defined by the



**Fig. 8.11.** Trajectory for position and orientation in 3D space based on axis-angle representation: profiles of the different components.

axes<sup>9</sup>

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} 0 & 0.05 & 0.05 & -0.77 \\ 1 & -1.10 & 0.32 & -0.62 \\ 0 & -0.99 & -0.94 & 0.10 \end{bmatrix}$$

and by the angles

$$[\theta_t] = [0.90, \ 0.95, \ 3.10, \ 0.85].$$

In each segment the axis is kept constant, while the angle varies from 0 to  $\theta_{tk}$  by means of a 5-th degree polynomial function of  $u$  (the same independent variable of the B-spline for the position trajectory) which allows to set initial (for  $u = \bar{u}_k$ , being  $\bar{u}_k$  the time instant in which the point  $q_k$  is crossed) and final (for  $u = \bar{u}_{k+1}$ ) velocities and accelerations equal to zero. The components of the trajectory for the position and the orientation are shown in Fig. 8.11. The RPY angles corresponding to the local frame at each point of the trajectory are used. Note in particular that the profile of an angle is discontinuous since the RPY angles are defined in the range  $[-\pi, \pi]$ .  $\square$

### 8.2.2 Case of position and orientation coupled

In many tasks, positioning and orientation problems are coupled at the Cartesian-coordinate level. In this case, a technique can be applied that allows to specify the orientation of the end effector on the basis of the orientation of the path at a given point. As a matter of fact, if the parametric form of the (regular) curve to be tracked, eq. (8.1), is expressed in terms of the curvilinear coordinate  $s$  (which measures the arc length<sup>10</sup>)

$$\Gamma : \quad \mathbf{p} = \mathbf{p}(s), \quad s \in [0, l] \quad (8.6)$$

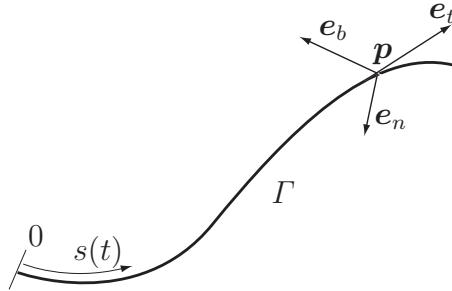
it is possible to define a coordinate frame directly tied to the curve, *the Frenet frame*, represented by three unit vectors:

- The *tangent unit vector*  $\mathbf{e}_t$ , lying on the line tangent to the curve and is oriented according to the positive direction induced on the curve by  $s$ .
- The *normal unit vector*  $\mathbf{e}_n$ , lying on the line passing through the point  $\mathbf{p}$ , and orthogonal to  $\mathbf{e}_t$ ; the orientation of  $\mathbf{e}_n$  is such that in a neighborhood of  $\mathbf{p}$  the curve is completely on the side of  $\mathbf{e}_n$  with respect to the plane passing through  $\mathbf{e}_t$  and normal to  $\mathbf{e}_n$ .
- The *binormal unit vector*  $\mathbf{e}_b$ , defined in a such way that the three vectors  $(\mathbf{e}_t, \mathbf{e}_n, \mathbf{e}_b)$  form a right handed frame.

---

<sup>9</sup> For each pair of frames  $(\mathbf{R}_k, \mathbf{R}_{k+1})$  the axis and the rotation angle are computed by means of (8.4) and (8.5).

<sup>10</sup> Therefore the parameter  $l$  in (8.6) represents the total length of the curve.



**Fig. 8.12.** Definition of a Frenet frame on a parametric curve.

The value of the Frenet vectors at a generic point  $\mathbf{p}$  can be deduced from the expression of the curve  $\Gamma$  by means of simple relations:

$$\mathbf{e}_t = \frac{d\mathbf{p}}{ds}, \quad \mathbf{e}_n = \frac{1}{\left| \frac{d^2\mathbf{p}}{ds^2} \right|} \frac{d^2\mathbf{p}}{ds^2}, \quad \mathbf{e}_b = \mathbf{e}_t \times \mathbf{e}_n.$$

Note that if the curve is characterized by the arc-length parameterization  $s$  and not by a generic parameter  $u$ , the tangent vector  $\mathbf{e}_t$  has unit length. In those applications in which the tool must have a fixed orientation with respect to the motion direction, e.g. in arc welding, the Frenet vectors implicitly define such an orientation. It is therefore sufficient to define the position trajectory function to obtain in each point the orientation of the tool.

**Example 8.5** A helicoidal trajectory is shown in Fig. 8.13, with the associated Frenet frames. The trajectory is described by the parametric form

$$\mathbf{p} = \begin{bmatrix} r \cos(u) \\ r \sin(u) \\ d u \end{bmatrix} \quad (8.7)$$

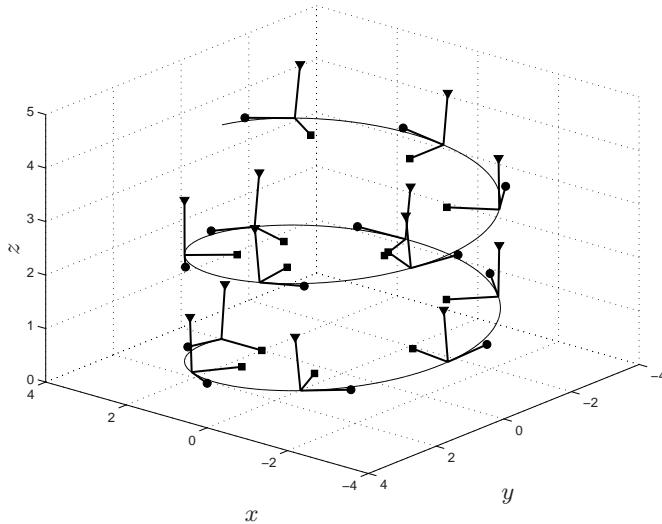
with  $u \in [0, 4\pi]$ , which leads to the frame<sup>11</sup>

$$\mathbf{R}_F = [\mathbf{e}_t, \mathbf{e}_n, \mathbf{e}_b] = \begin{bmatrix} -c \sin(u) & -\cos(u) & l \sin(u) \\ c \cos(u) & -\sin(u) & -l \cos(u) \\ l & 0 & c \end{bmatrix} \quad (8.8)$$

where  $c = \frac{r}{\sqrt{r^2 + d^2}}$  and  $l = \frac{d}{\sqrt{r^2 + d^2}}$ . □

---

<sup>11</sup> Since the parameter  $u$  is not the curvilinear coordinate, the associated (time varying) Frenet frames are computed as  $\mathbf{e}_t = \frac{d\mathbf{p}/du}{|d\mathbf{p}/du|}$ ,  $\mathbf{e}_n = \frac{d\mathbf{e}_t/du}{|d\mathbf{e}_t/du|}$ ,  $\mathbf{e}_b = \mathbf{e}_t \times \mathbf{e}_n$ .



**Fig. 8.13.** Helicoidal trajectory and associated Frenet frames.

**Example 8.6** Figure 8.14 shows a circular trajectory (for the sake of simplicity a planar case is considered) with the tool frame oriented in a fixed manner with respect to the Frenet frame. By considering the parametric description of the curve, expressed by (8.10) with  $\mathbf{o}' = \mathbf{0}$  and  $\mathbf{R} = \mathbf{I}_3$ , the Frenet frame is

$$\mathbf{R}_F(u) = \mathbf{R} \begin{bmatrix} -\sin(u) & -\cos(u) & 0 \\ \cos(u) & -\sin(u) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

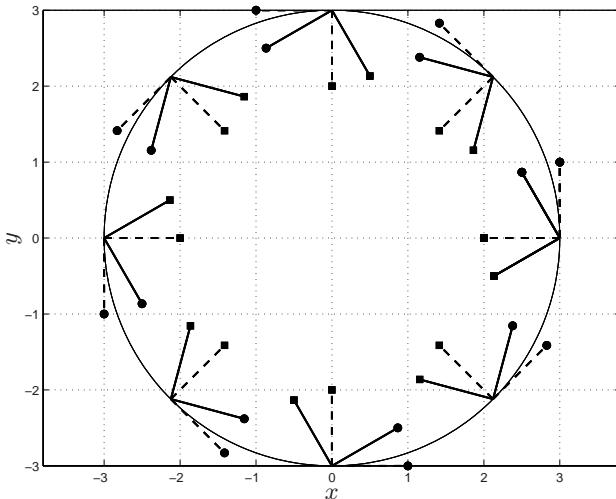
The desired tool frame differs from the Frenet frame by a rotation of  $\alpha = 30^\circ$  about the  $\mathbf{z}_F = \mathbf{e}_b$  axis. Such a rotation can be represented by the constant matrix

$$\mathbf{R}_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.866 & -0.500 & 0 \\ 0.500 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Therefore, the matrix representing the orientation in each point of the position trajectory is obtained simply by pre-multiplying the Frenet matrix by  $\mathbf{R}_\alpha$

$$\mathbf{R}_T(u) = \mathbf{R}_\alpha \mathbf{R}_F(u).$$

The tool frames computed for  $u = k\pi/4$ ,  $k = 0, \dots, 8$  are shown in Fig. 8.14.  $\square$



**Fig. 8.14.** A circular trajectory with the associated Frenet frames (dashed) and the desired tool frames (solid).

### 8.3 Definition of the Geometric Path Through Motion Primitives

A simple way to define the geometric path in the three-dimensional space is to use a sequence of basic motion primitives, such as straight lines, circles, and so on, or, more generally, a sequence of parametric functions.

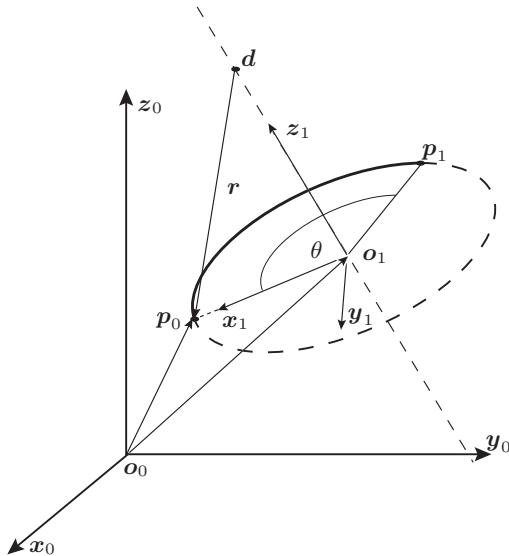
Obviously, the simplest manner to interpolate a given sequence of points is to use *straight lines*. In this case, the parametric form of a segment joining two points  $\mathbf{p}_0$  and  $\mathbf{p}_1$  is

$$\mathbf{p}(u) = \mathbf{p}_0 + (\mathbf{p}_1 - \mathbf{p}_0) u, \quad \text{with } 0 \leq u \leq 1 \quad (8.9)$$

Although the trajectory composed by a set of linear segments is continuous, it is characterized by discontinuous derivatives at the intermediate points. Then, undesired discontinuities in the velocity and acceleration profiles of the motion are present and, as discussed in more details in Sec. 8.11, this technique is often accompanied by the use of *blending functions* which guarantee a smooth transition between two consecutive segments.

Another typical motion primitive is the *circular arc*, starting from a given point  $\mathbf{p}_0$  and with the center located on a desired axis, univocally determined by a unitary vector  $\mathbf{z}_1$  and a generic point  $\mathbf{d}$ , see Fig. 8.15. Given these data, it is possible to compute the circular path. Firstly, it is necessary to determine the exact location of the center, provided that the trajectory does not degenerate in a point<sup>12</sup>. Let us define  $\mathbf{r} = \mathbf{p}_0 - \mathbf{d}$ , then the position of the

<sup>12</sup> In this case the point  $\mathbf{p}_0$  is located on the axis through the center of the circle.



**Fig. 8.15.** Example of motion primitive: circular arc.

center is

$$\mathbf{o}_1 = \mathbf{d} + (\mathbf{r}^T \mathbf{z}_1) \mathbf{z}_1$$

while the radius is

$$\rho = |\mathbf{p}_0 - \mathbf{o}_1|.$$

The parametric representation of the circular arc in the frame  $\mathcal{F}_1$  (defined by  $\mathbf{o}_1 - \mathbf{x}_1 \mathbf{y}_1 \mathbf{z}_1$ , see Fig. 8.15) is simply

$$\mathbf{p}_1(u) = \begin{bmatrix} \rho \cos(u) \\ \rho \sin(u) \\ 0 \end{bmatrix}, \quad \text{with } 0 \leq u \leq \theta$$

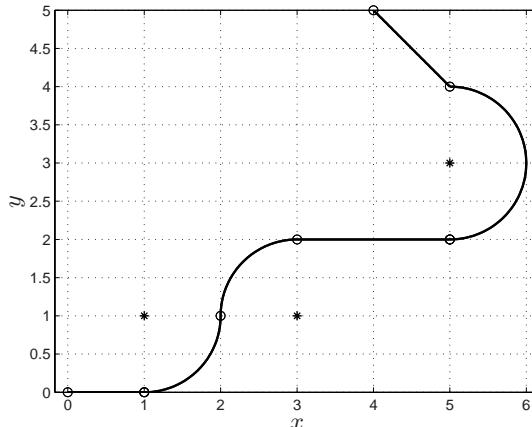
that, expressed in the base frame  $\mathcal{F}_0$ , has the form

$$\mathbf{p}(u) = \mathbf{o}_1 + \mathbf{R} \mathbf{p}_1(u) \quad (8.10)$$

where  $\mathbf{R}$  is the rotation matrix of the frame  $\mathcal{F}_1$  with respect to the frame  $\mathcal{F}_0$ , given by

$$\mathbf{R} = [\mathbf{x}_1 \ \mathbf{y}_1 \ \mathbf{z}_1].$$

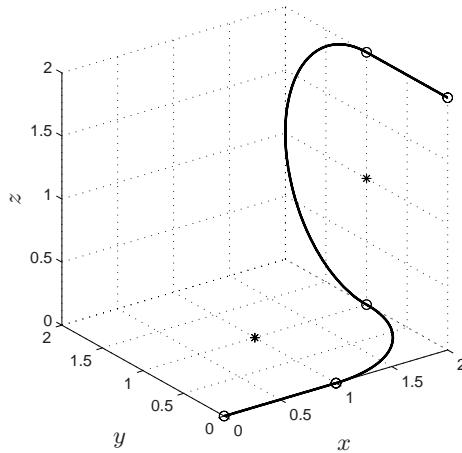
**Example 8.7** A two-dimensional trajectory composed by the following motion primitives



**Fig. 8.16.** Complex trajectory in the two-dimensional space obtained by composing linear and circular motion primitives.

	$p_0$	$p_1$	$z_1$	$d$	$\theta$
line	$[0 \ 0 \ 0]^T$	$[1 \ 0 \ 0]^T$			
circle	$[1 \ 0 \ 0]^T$		$[0 \ 0 \ 1]^T$	$[1 \ 1 \ 0]^T$	$\pi/2$
circle	$[2 \ 1 \ 0]^T$		$[0 \ 0 \ 1]^T$	$[3 \ 1 \ 0]^T$	$-\pi/2$
line	$[3 \ 2 \ 0]^T$	$[5 \ 2 \ 0]^T$			
circle	$[5 \ 3 \ 0]^T$		$[0 \ 0 \ 1]^T$	$[5 \ 3 \ 0]^T$	$\pi$
line	$[5 \ 4 \ 0]^T$	$[4 \ 5 \ 0]^T$			

is represented in Fig. 8.16. In this case, it is necessary to guarantee the continuity of the curve by properly choosing the initial and final points, the centers of the circular arcs, and so on. However, in general  $C^1$  continuity of the curve is not guaranteed (see Fig. 8.16), while  $C^2$  continuity can never be obtained by mixing linear and circular segments. Moreover, the construction of a complex curve in the 3D space, see Fig. 8.17, is a quite difficult problem.  $\square$



**Fig. 8.17.** Complex trajectory in the three-dimensional space obtained by composing linear and circular motion primitives.

## 8.4 Global Interpolation

The easiest way to interpolate a set of points  $\mathbf{q}_k$ ,  $k = 0, \dots, n$ , is to use  $p$  degree B-spline curves  $\mathbf{s}(u)$ , see Appendix B. The first step consists of the choice of the parameters  $\bar{u}_k$  (which represent a sort of “time instants”) for each point  $\mathbf{q}_k$ , and of a suitable knot vector  $\mathbf{u} = [u_0, \dots, u_{n_{knot}}]$ , where  $n_{knot}$  is the number of knots, see Appendix B. Then, it is possible to set up a system of  $m + 1$  linear equations in the unknowns  $\mathbf{p}_j$  obtained by imposing that the curve crosses each point  $\mathbf{q}_k$  at  $\bar{u}_k$ :

$$\mathbf{q}_k = \mathbf{s}(\bar{u}_k) = \sum_{j=0}^m \mathbf{p}_j B_j^p(\bar{u}_k) \quad (8.11)$$

or in a matrix form

$$\mathbf{q}_k^T = \left[ B_0^p(\bar{u}_k), B_1^p(\bar{u}_k), \dots, B_{m-1}^p(\bar{u}_k), B_m^p(\bar{u}_k) \right] \begin{bmatrix} \mathbf{p}_0^T \\ \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_{m-1}^T \\ \mathbf{p}_m^T \end{bmatrix}.$$

### 8.4.1 Definition of the set $\{\bar{u}_k\}$

In case there are no constraints due to the particular application at hand (in this case the  $\bar{u}_k$  are provided together with the point  $\mathbf{q}_k$ ), the parameters  $\bar{u}_k$  in (8.11) can be assumed within the range  $[0, 1]$ , therefore

$$\bar{u}_0 = 0, \quad \bar{u}_n = 1.$$

The most common choices for  $\bar{u}_k$ ,  $k = 1, \dots, n - 1$ , are:

- Equally spaced:

$$\bar{u}_k = \frac{k}{n}. \quad (8.12)$$

- Cord length distribution:

$$\bar{u}_k = \bar{u}_{k-1} + \frac{|\mathbf{q}_k - \mathbf{q}_{k-1}|}{d} \quad (8.13)$$

where

$$d = \sum_{k=1}^n |\mathbf{q}_k - \mathbf{q}_{k-1}|.$$

- Centripetal distribution:

$$\bar{u}_k = \bar{u}_{k-1} + \frac{|\mathbf{q}_k - \mathbf{q}_{k-1}|^\mu}{d} \quad (8.14)$$

where

$$d = \sum_{k=1}^n |\mathbf{q}_k - \mathbf{q}_{k-1}|^\mu.$$

This method, in which usually  $\mu = 1/2$ , provides good results when the data points take sharp turns.

#### 8.4.2 Cubic B-spline interpolation

The interpolation problem is quite often solved by assuming  $p = 3$ , which produces the traditional  $C^2$  cubic spline, already described in Sec. 4.4. The parameters  $\bar{u}_k$  are used to determine the knot vector  $\mathbf{u}$  as

$$\begin{aligned} u_0 &= u_1 = u_2 = \bar{u}_0 & u_{n+4} &= u_{n+5} = u_{n+6} = \bar{u}_n \\ u_{j+3} &= \bar{u}_j & j &= 0, \dots, n. \end{aligned} \quad (8.15)$$

With this choice the interpolation occurs at the knots. Since the number of (unknown) control points,  $m + 1$ , and the number of the knots,  $n_{knot} + 1$ , are related by  $n_{knot} = m + 4$  and, as can be easily deduced from (8.15),  $n_{knot} = n + 6$ , the unknown variables  $\mathbf{p}_j$  are  $n + 3$ . Therefore, in order to find a unique solution, it is necessary to impose two additional constraints (besides the  $n + 1$  points to be interpolated). In the following, these two constraints are assumed to be the first derivatives at the endpoints, respectively  $\mathbf{t}_0$  and  $\mathbf{t}_n$ . As a consequence, the first two and the last two equations of the  $(n + 3) \times (n + 3)$  linear system are

$$\begin{aligned} \mathbf{p}_0 &= \mathbf{q}_0 \\ -\mathbf{p}_0 + \mathbf{p}_1 &= \frac{u_4}{3} \mathbf{t}_0 \end{aligned}$$

and

$$\begin{aligned}-\mathbf{p}_{n+1} + \mathbf{p}_{n+2} &= \frac{1 - u_{n+3}}{3} \mathbf{t}_n \\ \mathbf{p}_{n+2} &= \mathbf{q}_n.\end{aligned}$$

These four equations can be directly solved, obtaining

$$\begin{cases} \mathbf{p}_0 = \mathbf{q}_0 \\ \mathbf{p}_1 = \mathbf{q}_0 + \frac{u_4}{3} \mathbf{t}_1 \\ \mathbf{p}_{n+2} = \mathbf{q}_n \\ \mathbf{p}_{n+1} = \mathbf{q}_n - \frac{1 - u_{n+3}}{3} \mathbf{t}_n. \end{cases} \quad (8.16)$$

The remaining  $n - 1$  control points are computed by imposing

$$\mathbf{s}(\bar{u}_k) = \mathbf{q}_k, \quad k = 1, \dots, n - 1.$$

By recalling the fact that for a cubic spline in an interior knot only three basis functions are not null, see Appendix B, the  $n - 1$  equations have the expression

$$\mathbf{q}_k = B_k^3(\bar{u}_k) \mathbf{p}_k + B_{k+1}^3(\bar{u}_k) \mathbf{p}_{k+1} + B_{k+2}^3(\bar{u}_k) \mathbf{p}_{k+2}$$

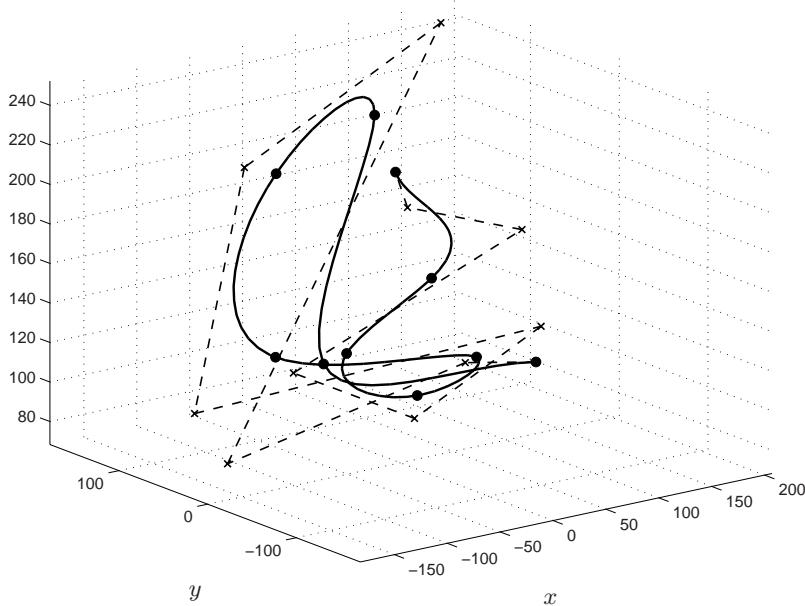
and therefore they form a tridiagonal system

$$\mathbf{B} \mathbf{P} = \mathbf{R} \quad (8.17)$$

where

$$\mathbf{B} = \begin{bmatrix} B_2^3(\bar{u}_1) & B_3^3(\bar{u}_1) & 0 & \cdots & 0 \\ B_2^3(\bar{u}_2) & B_3^3(\bar{u}_2) & B_4^3(\bar{u}_2) & & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & & B_{n-2}^3(\bar{u}_{n-2}) & B_{n-1}^3(\bar{u}_{n-2}) & B_n^3(\bar{u}_{n-2}) \\ 0 & \cdots & 0 & B_{n-1}^3(\bar{u}_{n-1}) & B_n^3(\bar{u}_{n-1}) \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_2^T \\ \mathbf{p}_3^T \\ \mathbf{p}_4^T \\ \vdots \\ \mathbf{p}_{n-1}^T \\ \mathbf{p}_n^T \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{q}_1^T - B_1^3(\bar{u}_1) \mathbf{p}_1^T \\ \mathbf{q}_2^T \\ \mathbf{q}_3^T \\ \vdots \\ \mathbf{q}_{n-2}^T \\ \mathbf{q}_{n-1}^T - B_{n+1}^3(\bar{u}_{n-1}) \mathbf{p}_{n+1}^T \end{bmatrix}.$$



**Fig. 8.18.** Three dimensional global interpolation by a cubic B-spline (solid) and related control polygon (dashed).

The control points  $\mathbf{p}_j$ ,  $j = 2, \dots, n$  are obtained by solving (8.17):

$$\mathbf{P} = \mathbf{B}^{-1} \mathbf{R}$$

see Appendix A.5 .

Once the control points  $\mathbf{p}_j$ ,  $j = 0, \dots, n + 2$  are known, and given the knot vector  $\mathbf{u}$ , the B-spline is completely defined and it is possible to compute  $s(u)$  for any value of the independent variable  $u$  according to the algorithm reported in B.1.3. Another possibility is to convert the B-spline in a standard piecewise polynomial form, as discussed in Appendix B.1.5.

**Example 8.8** The interpolation of a set of points by means of a cubic B-spline is reported in Fig. 8.18. In particular, the goal is to find a trajectory passing through the points

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} 83 & -64 & 42 & -98 & -13 & 140 & 43 & -65 & -45 & 71 \\ -54 & 10 & 79 & 23 & 125 & 81 & 32 & -17 & -89 & 90 \\ 119 & 124 & 226 & 222 & 102 & 92 & 92 & 134 & 182 & 192 \end{bmatrix}$$

with the initial and final derivatives<sup>13</sup>

---

<sup>13</sup> When the derivatives are not imposed by the particular application, it is possible to assume  $t_0 = \frac{q_1 - q_0}{\bar{u}_1 - \bar{u}_0}$  and  $t_n = \frac{q_n - q_{n-1}}{\bar{u}_n - \bar{u}_{n-1}}$ .

$$\mathbf{t}_0 = \begin{bmatrix} -1236 \\ 538 \\ 42 \end{bmatrix}, \quad \mathbf{t}_9 = \begin{bmatrix} 732 \\ 1130 \\ 63 \end{bmatrix}.$$

The parameters  $\bar{u}_k$  are assumed according to a cord length distribution, therefore the knots are

$$\mathbf{u} = [0, 0, 0, 0, 0.11, 0.23, 0.35, 0.48, 0.60, 0.68, 0.77, 0.84, 1, 1, 1, 1].$$

The control points of the spline, computed by (8.16) and (8.17), are

$$\mathbf{P} = \begin{bmatrix} 83 & 34 & -168 & 146 & -182 & -45 & 207 & 31 & -89 & -29 & 32 & 71 \\ -54 & -32 & -5 & 128 & -41 & 177 & 88 & 21 & 14 & -172 & 30 & 90 \\ 119 & 120 & 88 & 252 & 245 & 68 & 98 & 83 & 121 & 218 & 188 & 192 \end{bmatrix}^T.$$

Given the control points  $\mathbf{P}$ , the curve can be evaluated, for  $0 \leq u \leq 1$ , according to the algorithm reported in Sec. B.1.3.  $\square$

**Example 8.9** Fig. 8.19 reports a case in which the interpolation by means of B-splines is used to approximate the geometric path obtained with linear and circular motion primitives (see Fig. 8.16). In particular, the geometric path is sampled (with different levels of quantization) and then the points are interpolated. Obviously, the approximation error depends on the number of points used to describe the original trajectory. Anyway, it is worth noticing that the interpolation by means of cubic B-splines guarantees the  $C^2$  continuity of the geometric path. On the other hand, because of the continuity of the curve derivatives, it is not possible to obtain a path with sharp corners. In this case, it is necessary to split the trajectory in different tracts, each one described by a different B-spline curve.  $\square$

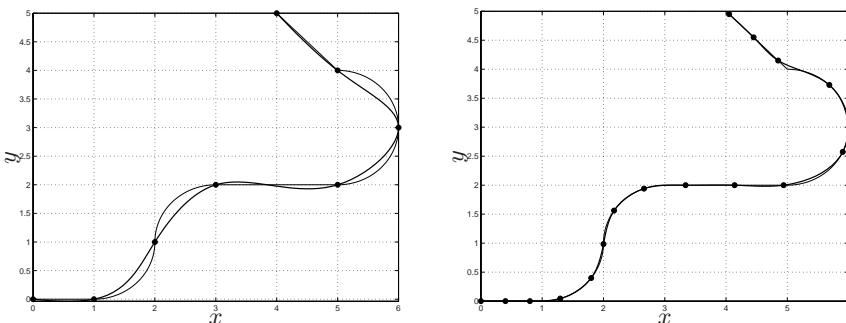


Fig. 8.19. B-spline approximation of motion primitives.

## 8.5 Global Approximation

In many cases the need of reducing the number of data points, for example for computational reasons, requires the adoption of simpler curves (with respect to those obtained with interpolation techniques) at the expense of lower precision. Such trajectories do not exactly cross the via-points but approximate them within a prescribed tolerance  $\delta$ . Also in this case the most straightforward solution is based on the use of B-splines. Nevertheless, while in an interpolation problem the number of control points (and of knots) is determined once the points and the spline order are given, approximation requires the choice of a proper number of control points according to the desired accuracy.

Given the points  $\mathbf{q}_0, \dots, \mathbf{q}_n$  to be approximated, the degree  $p \geq 1$  ( $p = 3$  is normally considered in order to guarantee the  $C^2$  continuity of the curve), and provided that  $n > m > p$ , the approximating trajectory is a B-spline

$$\mathbf{s}(u) = \sum_{j=0}^m \mathbf{p}_j B_j^p(u), \quad u_{min} \leq u \leq u_{max} \quad (8.18)$$

satisfying the following two conditions:

1. The end points are exactly interpolated, i.e.  $\mathbf{q}_0 = \mathbf{s}(0)$  and  $\mathbf{q}_n = \mathbf{s}(1)$ .
2. The internal points  $\mathbf{q}_k$  are approximated in the least square sense, i.e. by minimizing the functional

$$\sum_{k=1}^{n-1} w_k |\mathbf{q}_k - \mathbf{s}(\bar{u}_k)|^2 \quad (8.19)$$

with respect to the  $m + 1$  variables  $\mathbf{p}_j$ ; the coefficients  $w_k$  can be freely chosen to weight the error at different points.

By stacking the  $n + 1$  equations

$$\mathbf{s}(\bar{u}_k) = \sum_{j=0}^m \mathbf{p}_j B_j^p(\bar{u}_k) = \mathbf{q}_k, \quad k = 0, \dots, n$$

the approximation problem can be written in a matrix form as

$$\mathbf{B} \mathbf{P} = \mathbf{R} \quad (8.20)$$

that represents a linear system of  $n + 1$  equations in the  $m + 1$  unknowns  $\mathbf{p}_j$ . The first and last control points are determined by assigning  $\mathbf{p}_0 = \mathbf{q}_0$  and  $\mathbf{p}_m = \mathbf{q}_n$ , while the remaining control points are the solution of (8.20), where  $\mathbf{B}$ ,  $\mathbf{P}$ , and  $\mathbf{R}$  are  $(n - 1) \times (m - 1)$ ,  $(m - 1) \times 3$ , and  $(n - 1) \times 3$  matrices<sup>14</sup> respectively, defined by

---

<sup>14</sup> The 3D case is considered; more generally,  $\mathbf{P}$  and  $\mathbf{R}$  have dimensions  $(m - 1) \times d$  and  $(n - 1) \times d$  respectively, where  $d$  is the dimension of the space of the points to be interpolated, and accordingly of the control points defining the spline.

$$\mathbf{B} = \begin{bmatrix} B_1^p(\bar{u}_1) & B_2^p(\bar{u}_1) & \dots & B_{m-2}^p(\bar{u}_1) & B_{m-1}^p(\bar{u}_1) \\ B_1^p(\bar{u}_2) & \ddots & & & B_{m-1}^p(\bar{u}_2) \\ \vdots & & & & \vdots \\ B_1^p(\bar{u}_{n-2}) & & \dots & & B_{m-1}^p(\bar{u}_{n-2}) \\ B_1^p(\bar{u}_{n-1}) & B_2^p(\bar{u}_{n-1}) & \dots & B_{m-2}^p(\bar{u}_{n-1}) & B_{m-1}^p(\bar{u}_{n-1}) \end{bmatrix} \quad (8.21)$$

and

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_{m-2}^T \\ \mathbf{p}_{m-1}^T \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{q}_1^T - B_0^p(\bar{u}_1)\mathbf{q}_0^T - B_m^p(\bar{u}_1)\mathbf{q}_n^T \\ \mathbf{q}_2^T - B_0^p(\bar{u}_2)\mathbf{q}_0^T - B_m^p(\bar{u}_2)\mathbf{q}_n^T \\ \vdots \\ \mathbf{q}_{n-2}^T - B_0^p(\bar{u}_{n-2})\mathbf{q}_0^T - B_m^p(\bar{u}_{n-2})\mathbf{q}_n^T \\ \mathbf{q}_{n-1}^T - B_0^p(\bar{u}_{n-1})\mathbf{q}_0^T - B_m^p(\bar{u}_{n-1})\mathbf{q}_n^T \end{bmatrix}.$$

Since  $n > m$ , the problem (8.20) is over-constrained and an optimal solution in the least square sense can be found by using a standard technique based on a weighted pseudo-inverse matrix, which minimizes the expression

$$\text{tr} \left( (\mathbf{R}^T - \mathbf{P}^T \mathbf{B}^T) \mathbf{W} (\mathbf{R} - \mathbf{B} \mathbf{P}) \right) \quad (8.22)$$

where  $\text{tr}(\mathbf{X})$  is the trace<sup>15</sup> of the matrix  $\mathbf{X}$ , and  $\mathbf{W} = \text{diag}\{w_1, \dots, w_{n-1}\}$  the matrix of the weights. Therefore, the internal control points  $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_{m-1}]^T$

---

<sup>15</sup> The criterium (8.22) is nothing but the minimization of

$$|\mathbf{V}(\mathbf{R} - \mathbf{B} \mathbf{P})|_F^2 \quad (8.23)$$

where  $\mathbf{V}^T \mathbf{V} = \mathbf{W}$ , and  $|\mathbf{X}|_F$  is the *Frobenius norm* of the  $m \times n$  matrix  $\mathbf{X}$ , defined as

$$|\mathbf{X}|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |x_{i,j}|^2}.$$

A noteworthy property, from which the equivalence of (8.22), (8.23) and (8.19) descends, is that

$$|\mathbf{X}|_F^2 = \text{tr}(\mathbf{X} \mathbf{X}^T).$$

The results proposed in this and in the following sections can be obtained by collecting the points composing  $\mathbf{P}$  and  $\mathbf{R}$  in a different way; in particular, in the 3D case some authors consider the  $(3(m-1) \times 1)$  vector

$$\mathbf{p}_v = [p_{x,1}, \dots, p_{x,m-1}, p_{y,1}, p_{y,2}, \dots, p_{y,m-1}, p_{x,1}, p_{y,2}, \dots, p_{y,m-1}]^T$$

and the  $(3(n-1) \times 1)$  vector

$$\mathbf{r}_v = [r_{x,1}, \dots, r_{x,n-1}, r_{y,1}, r_{y,2}, \dots, r_{y,n-1}, r_{x,1}, r_{y,2}, \dots, r_{y,n-1}]^T$$

are

$$\mathbf{P} = \mathbf{B}^\dagger \mathbf{R} \quad (8.24)$$

where

$$\mathbf{B}^\dagger = (\mathbf{B}^T \mathbf{W} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{W}. \quad (8.25)$$

### 8.5.1 Knots choice

In order to set up the system (8.20), it is necessary to fix both the value of the “time instants”  $\bar{u}_k$ ,  $k = 0, \dots, n$ , at which the approximation occurs and the value of the knots which define the B-spline. While the computation of  $\bar{u}_k$  is straightforward and can be performed according to the techniques reported in Sec. 8.4.1, the choice of the knots  $u_k$  deserves a particular attention, in order to guarantee that every knot span contain at least one  $\bar{u}_k$ . In this case, the matrix  $(\mathbf{B}^T \mathbf{W} \mathbf{B})$  in eq. (8.25) is positive definite and well-conditioned.

Given  $n + 1$  points to be approximated by a  $p$  degree B-spline with  $m + 1$  control points, define

$$d = \frac{n + 1}{m - p + 1} \quad (> 0).$$

Then the  $m - p$  internal knots can be computed, for  $j = 1, \dots, m - p$ , as

$$\begin{cases} i = \text{floor}(jd) \\ \alpha = jd - i \\ u_{j+p} = (1 - \alpha)\bar{u}_{i-1} + \alpha\bar{u}_i \end{cases}$$

being  $\text{floor}(x)$  the function which gives the largest integer less than or equal to  $x$ . The first and the last  $p + 1$  knots are

$$u_0 = \dots = u_p = \bar{u}_0, \quad u_{m+1} = \dots = u_{m+p+1} = \bar{u}_n.$$

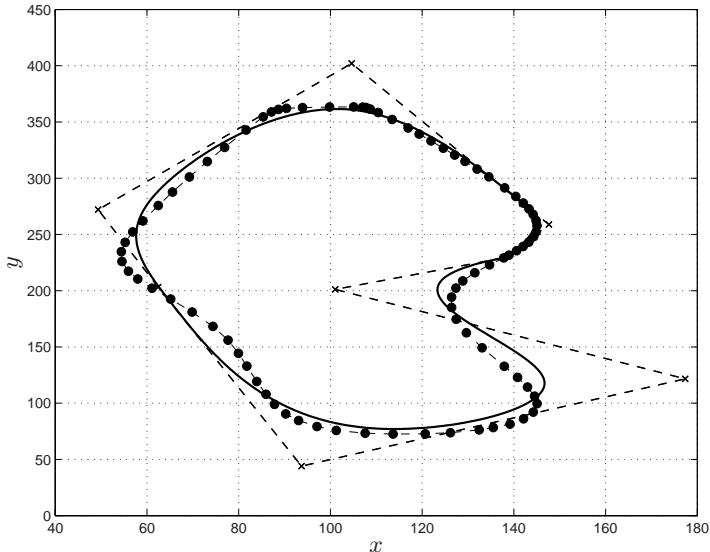
**Example 8.10** The approximation of a set of 84 points is reported in Fig. 8.20 (for the sake of simplicity the points are considered disposed on the  $x - y$  plane). In particular a cubic B-spline ( $p = 3$ ) is adopted, with 10

---

related by the  $(3(m - 1) \times 3(n - 1))$  matrix

$$\mathbf{B}_3 = \begin{bmatrix} \mathbf{B} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} \end{bmatrix}$$

where  $\mathbf{B}$  is defined by (8.21). In this case, the solution  $\mathbf{p}_v = \mathbf{B}_3^\dagger \mathbf{r}_v$ , which minimizes  $|\mathbf{V}_3(\mathbf{r}_v - \mathbf{B}_3 \mathbf{p}_v)|_2^2$  (where the standard Euclidean norm is considered), coincides with (8.24). In the next sections, in order to make the notation more simple, the matrices  $\mathbf{P}$  and  $\mathbf{R}$  are preferred to the vectors  $\mathbf{p}_v$  and  $\mathbf{r}_v$ , even if this implies the use of the Frobenius norm in lieu of the usual Euclidean norm. Note that the Frobenius norm is very useful for numerical linear algebra, being often easier to compute than induced norms.



**Fig. 8.20.** Global approximation by means of a B-spline (solid curve) defined by 10 control points, represented by the x-marks on the control polygon (dashed).

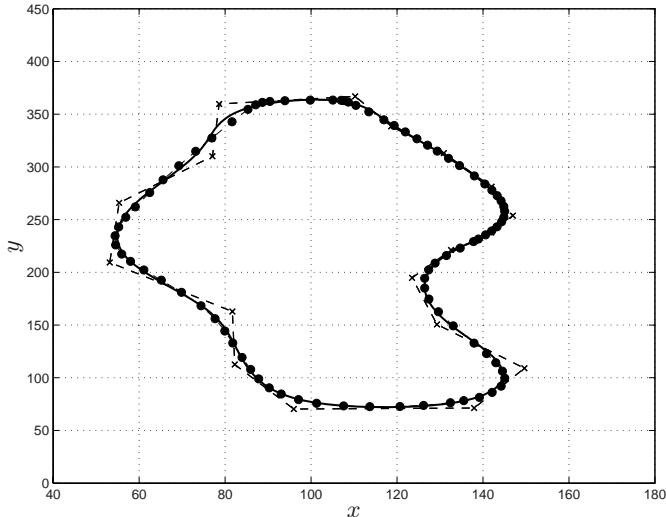
control points. The parameters  $\bar{u}_k$  are assumed to be proportional to the cord length, and the knots, computed as described in this section, are

$$\mathbf{u} = [0, 0, 0, 0, 0.16, 0.28, 0.46, 0.65, 0.78, 0.90, 1, 1, 1, 1].$$

The control points of the spline obtained by means of (8.24) are

$$\mathbf{P} = \begin{bmatrix} 137 & 101 & 177 & 93 & 62 & 49 & 104 & 141 & 147 & 138 \\ 229 & 201 & 121 & 44 & 203 & 272 & 402 & 277 & 258 & 231 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

In this case the spline, composed by few segments, provides a good approximation of the data points. Nevertheless, in order to reduce the error between the curve and the via-points one can increase the number of control points, which define the spline, searching for a good trade-off between approximation error and complexity of the trajectory. In Fig. 8.21 a B-spline defined by 20 control points is shown, while Tab. 8.1 reports the approximation errors for different values of the parameter  $m$ .  $\square$



**Fig. 8.21.** Global approximation by means of a B-spline (solid) defined by 20 control points, represented by the x-marks on the control polygon (dashed).

$m$	$\sum_k  \mathbf{q}_k - \mathbf{s}(\bar{u}_k) ^2$	$\bar{\varepsilon}$	$\varepsilon_{max}$
10	3092	4.85	12.97
20	51.22	0.57	3.25
30	7.67	0.19	1.16
40	1.40	0.07	0.55

**Table 8.1.** Total square error, mean error and maximum error of the approximating B-spline with respect to the given data points for different numbers of the control points ( $m + 1$ ).

## 8.6 A Mixed Interpolation/Approximation Technique

In some applications, it may be required that the trajectory crosses exactly a set of data points, while other points are simply approximated. In these cases, it is possible to adopt an approach that mixes the two techniques reported in the previous sections by defining a constrained minimization problem.

Let  $\mathbf{q}_{i,k}$ ,  $k = 0, \dots, n_i$  be the points to be interpolated and  $\mathbf{q}_{a,k}$ ,  $k = 0, \dots, n_a$  the points to be approximated. Then, it is possible to arrange two systems of equations, which express the fact that the  $p$  degree B-spline<sup>16</sup>, assumed to define the trajectory, crosses these points. The system composed by the constrained equations and the system of the unconstrained equations are respectively

<sup>16</sup> Let  $m + 1$  be the number of control points of the trajectory  $\mathbf{s}(u)$ . The condition  $m \geq n_i$  must be satisfied.

$$\mathbf{B}_i \mathbf{P} = \mathbf{R}_i \quad (8.26)$$

$$\mathbf{B}_a \mathbf{P} = \mathbf{R}_a \quad (8.27)$$

with

$$\mathbf{B}_i = \begin{bmatrix} B_0^p(\bar{u}_{i,0}) & \dots & B_m^p(\bar{u}_{i,0}) \\ \vdots & \ddots & \vdots \\ B_0^p(\bar{u}_{i,n_i}) & \dots & B_m^p(\bar{u}_{i,n_i}) \end{bmatrix}, \quad \mathbf{R}_i = \begin{bmatrix} \mathbf{q}_{i,0}^T \\ \vdots \\ \mathbf{q}_{i,n_i}^T \end{bmatrix}$$

and

$$\mathbf{B}_a = \begin{bmatrix} B_0^p(\bar{u}_{a,0}) & \dots & B_m^p(\bar{u}_{a,0}) \\ \vdots & \ddots & \vdots \\ B_0^p(\bar{u}_{a,n_a}) & \dots & B_m^p(\bar{u}_{a,n_a}) \end{bmatrix}, \quad \mathbf{R}_a = \begin{bmatrix} \mathbf{q}_{a,0}^T \\ \vdots \\ \mathbf{q}_{a,n_a}^T \end{bmatrix}.$$

In order to set up the two systems (8.26) and (8.27), it is necessary to define the time instants  $\bar{u}_i$  and  $\bar{u}_a$  in which the interpolation/approximation occurs. This can be obtained with the same techniques reported in Sec. 8.4.1. At this point the  $m + p + 1$  knots, necessary for the definition of the B-spline, can be computed according to the method of Sec. 8.5.1 applied to the parameters  $\bar{u}$ , obtained by merging and sorting in ascending order  $\bar{u}_i$  and  $\bar{u}_a$ . The goal is to minimize the function  $\mathbf{R}_a - \mathbf{B}_a \mathbf{P}$  subject to (8.26) that, by using Lagrange multipliers, is equivalent to minimize

$$\text{tr}\left((\mathbf{R}_a^T - \mathbf{P}^T \mathbf{B}_a^T) \mathbf{W} (\mathbf{R}_a - \mathbf{B}_a \mathbf{P}) + \boldsymbol{\lambda}^T (\mathbf{B}_i \mathbf{P} - \mathbf{R}_i)\right) \quad (8.28)$$

where  $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_0, \dots, \boldsymbol{\lambda}_{n_i}]^T$  is the  $(n_i + 1) \times 3$  matrix of Lagrange multipliers. By differentiating (8.28) with respect to the unknowns  $\mathbf{P}$  and  $\boldsymbol{\lambda}$  and by setting the result to zero, after some algebraic manipulations [38] one obtains

$$\mathbf{B}_a^T \mathbf{W} \mathbf{B}_a \mathbf{P} + \mathbf{B}_i^T \boldsymbol{\lambda} = \mathbf{B}_a^T \mathbf{W} \mathbf{R}_a \quad (8.29)$$

$$\mathbf{B}_i \mathbf{P} = \mathbf{R}_i. \quad (8.30)$$

In case  $\mathbf{B}_a^T \mathbf{W} \mathbf{B}_a$  and  $\mathbf{B}_i (\mathbf{B}_a^T \mathbf{W} \mathbf{B}_a)^{-1} \mathbf{B}_i^T$  are both invertible, the solution of (8.29) and (8.30) is given by

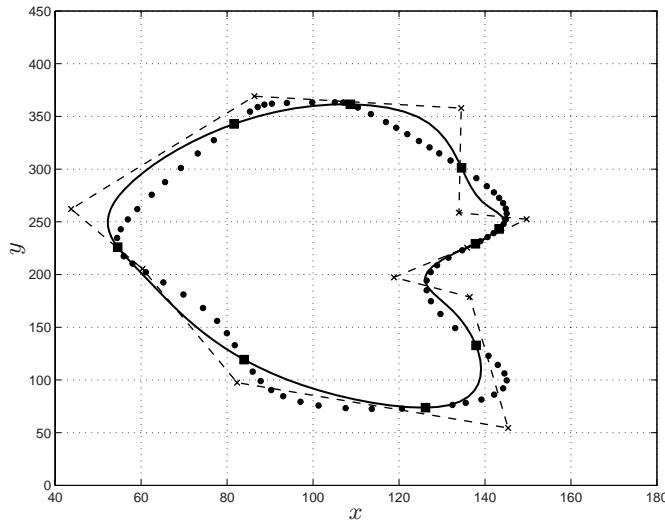
$$\boldsymbol{\lambda} = \left( \mathbf{B}_i (\mathbf{B}_a^T \mathbf{W} \mathbf{B}_a)^{-1} \mathbf{B}_i^T \right)^{-1} \left( \mathbf{B}_i (\mathbf{B}_a^T \mathbf{W} \mathbf{B}_a)^{-1} \mathbf{B}_a^T \mathbf{W} \mathbf{R}_a - \mathbf{R}_i \right)$$

$$\mathbf{P} = (\mathbf{B}_a^T \mathbf{W} \mathbf{B}_a)^{-1} \mathbf{B}_a^T \mathbf{W} \mathbf{R}_a - (\mathbf{B}_a^T \mathbf{W} \mathbf{B}_a)^{-1} \mathbf{B}_i^T \boldsymbol{\lambda}.$$

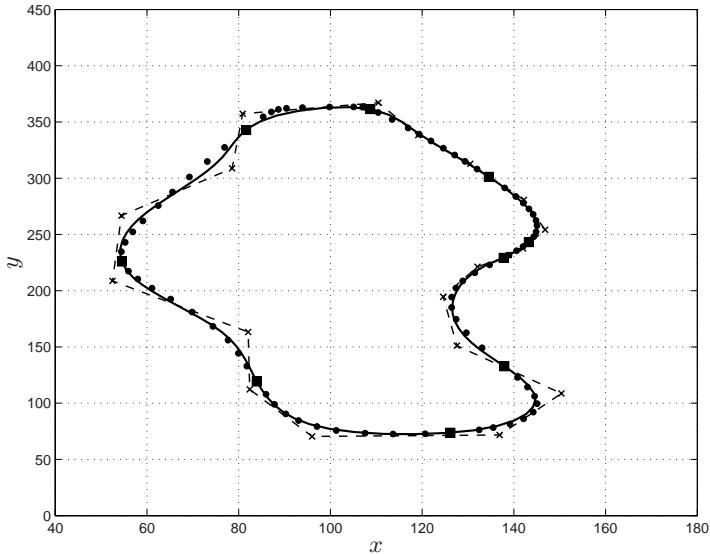
**Example 8.11** The approximation of the set of points used in the previous example is reported in Fig. 8.22. In this case, it is also required that the points (denoted in the figure by square marks)

$$\begin{bmatrix} q_{i,x} \\ q_{i,y} \\ q_{i,z} \end{bmatrix} = \begin{bmatrix} 137 & 137 & 126 & 83 & 54 & 81 & 108 & 134 & 143 \\ 229 & 132 & 73 & 119 & 226 & 342 & 361 & 301 & 243 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

are exactly interpolated by the trajectory. In particular a cubic B-spline ( $p = 3$ ) is adopted, with 12 control points. In order to reduce the approximation error, a spline with 20 control points is also considered; the result is shown in Fig. 8.23.  $\square$



**Fig. 8.22.** Global interpolation/approximation by means of a B-spline defined by 12 control points (x-marks).



**Fig. 8.23.** Global interpolation/approximation by means of a B-spline defined by 20 control points (x-marks).

## 8.7 Smoothing Cubic B-splines

Given a set of points  $\mathbf{q}_k$ ,  $k = 0, \dots, n$  to be approximated within a prescribed tolerance at time instants  $\bar{u}_k$ , it is possible to adopt the so-called smoothing cubic splines (in particular in the B-form) i.e. cubic ( $p = 3$ ) B-spline

$$\mathbf{s}(u) = \sum_{j=0}^m \mathbf{p}_j B_j^p(u), \quad u_{min} \leq u \leq u_{max}$$

whose control points are computed as a tradeoff between two apposite goals:

- To find a good fit of the given via-points.
- To obtain a trajectory as smooth as possible (with curvature/acceleration as small as possible).

The choice of the control points can be made by minimizing the functional

$$L := \mu \sum_{k=0}^n w_k |\mathbf{s}(\bar{u}_k) - \mathbf{q}_k|^2 + (1 - \mu) \int_{\bar{u}_0}^{\bar{u}_n} \left| \frac{d^2 \mathbf{s}(u)}{du^2} \right|^2 du \quad (8.31)$$

where the parameter  $\mu \in [0, 1]$  reflects the different importance given to the conflicting goals, and  $w_k$  are parameters which can be arbitrarily chosen to modify the weights of different quadratic errors on the global estimation (the selection of different coefficients  $w_k$  allows to operate locally on the spline by

reducing the approximation error in some points of interest).

In order to define the trajectory, it is necessary to choose the knot vector. A common choice is to assume

$$u_0 = \dots = u_2 = \bar{u}_0, \quad u_{n+4} = \dots = u_{n+6} = \bar{u}_n \\ u_{j+3} = \bar{u}_j, \quad j = 0, \dots, n$$

and, as a consequence, the number  $m + 1$  of control points to be determined is  $n + 3$  (note that it is possible to choose a smaller number of knots and control points).

The integral in the second term of (8.31) can be written as

$$\int_{\bar{u}_0}^{\bar{u}_n} \left| \frac{d^2 \mathbf{s}(u)}{du^2} \right|^2 du = \int_u \sum_{j=1}^{n-1} |\mathbf{r}_{j-1} B_{j-1}^1(u) + \mathbf{r}_j B_j^1(u)|^2 du \quad (8.32)$$

where  $B_j^1$  are the first degree B-spline basis defined on the knot vector

$$\mathbf{u}_r = [u_2, u_3, \dots, u_{n+3}, u_{n+4}]$$

and  $\mathbf{r}_j$  are the control points of the spline defining the ‘acceleration’ profile  $d^2 \mathbf{s}(u)/du^2$ . The control points  $\mathbf{r}_j$  are related to  $\mathbf{p}_j$  by

$$\mathbf{r}_j = \frac{6}{u_{j+4} - u_{j+2}} \left[ \frac{1}{u_{j+4} - u_{j+1}} \mathbf{p}_j - \left( \frac{1}{u_{j+4} - u_{j+1}} + \frac{1}{u_{j+5} - u_{j+2}} \right) \mathbf{p}_{j+1} + \frac{1}{u_{j+5} - u_{j+2}} \mathbf{p}_{j+2} \right].$$

Note that  $\mathbf{u}_r$  and the expression of  $\mathbf{r}_j$  are both based on the knots of  $\mathbf{s}(u)$ . Equation (8.32) can be further simplified as

$$\begin{aligned} \int_{\bar{u}_1}^{\bar{u}_n} \left| \frac{d^2 \mathbf{s}(u)}{du^2} \right|^2 du &= \frac{1}{3} \sum_{j=1}^n (u_{j+3} - u_{j+2}) (|\mathbf{r}_j|^2 + \mathbf{r}_j \mathbf{r}_{j-1} + |\mathbf{r}_{j-1}|^2) \\ &= \frac{1}{6} \text{tr}(\mathbf{R}^T \mathbf{A} \mathbf{R}) = \frac{1}{6} \text{tr}(\mathbf{P}^T \mathbf{C}^T \mathbf{A} \mathbf{C} \mathbf{P}) \end{aligned} \quad (8.33)$$

being

$$\mathbf{R} = [\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_n]^T, \quad \mathbf{P} = [\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_m]^T$$

and

$$\mathbf{A} = \begin{bmatrix} 2u_{4,3} & u_{4,3} & 0 & \cdots & 0 \\ u_{4,3} & 2(u_{4,3} + u_{5,4}) & u_{5,4} & 0 & \vdots \\ 0 & u_{5,4} & & & 0 \\ \vdots & 0 & \ddots & & \\ 0 & & u_{n+2,n+1} & 2(u_{n+2,n+1} + u_{n+3,n+2}) & u_{n+3,n+2} \\ & & \cdots & 0 & 2u_{n+3,n+2} \end{bmatrix} \quad (8.34)$$

$$\mathbf{C} = \begin{bmatrix} c_{0,1} & c_{0,2} & c_{0,3} & 0 & \cdots & 0 \\ 0 & c_{1,1} & c_{1,2} & c_{1,3} & 0 & \vdots \\ \vdots & & & & \ddots & 0 \\ 0 & \cdots & & 0 & c_{n,1} & c_{n,2} & c_{n,3} \end{bmatrix} \quad (8.35)$$

where

$$\begin{cases} c_{k,1} = \frac{6}{u_{k+4,k+2} u_{k+4,k+1}} \\ c_{k,2} = -\frac{6}{u_{k+4,k+2}} \left( \frac{1}{u_{k+4,k+1}} + \frac{1}{u_{k+5,k+2}} \right) \\ c_{k,3} = \frac{6}{u_{k+4,k+2} u_{k+5,k+2}} \end{cases} \quad (8.36)$$

and  $u_{i,j} = u_i - u_j$ .

Finally, the expression of  $L$  becomes

$$L = \text{tr}((\mathbf{Q} - \mathbf{B}\mathbf{P})^T \mathbf{W} (\mathbf{Q} - \mathbf{B}\mathbf{P})) + \lambda \text{tr}(\mathbf{P}^T \mathbf{C}^T \mathbf{A}\mathbf{C}\mathbf{P}) \quad (8.37)$$

with  $\mathbf{Q} = [\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_n]^T$ ,  $\lambda = \frac{1-\mu}{6\mu}$ ,  $\mathbf{W} = \text{diag}\{w_k\}$ ,  $k = 0, \dots, n$ , and

$$\mathbf{B} = \begin{bmatrix} B_0^3(\bar{u}_0) & \dots & B_m^3(\bar{u}_0) \\ \vdots & \ddots & \vdots \\ B_0^3(\bar{u}_n) & \dots & B_m^3(\bar{u}_n) \end{bmatrix}.$$

The control points defining the B-spline are obtained by minimizing (8.37) with respect to  $\mathbf{P}$ .

### 8.7.1 Smoothing B-splines with assigned start/end points and directions

In this section, some additional conditions are considered:

- a) The first and the last point  $\mathbf{q}_0, \mathbf{q}_n$  are exactly interpolated by the trajectory.
- b) The tangent vectors  $\mathbf{t}_0, \mathbf{t}_n$  at  $\mathbf{q}_0$  and  $\mathbf{q}_n$  are assigned.

As in Sec. 8.4.2, the conditions a) and b) imply that

$$\begin{aligned} \mathbf{p}_0 &= \mathbf{q}_0 \\ -\mathbf{p}_0 + \mathbf{p}_1 &= \frac{u_4}{3} \mathbf{d}_1 \end{aligned}$$

and

$$\begin{aligned}-\mathbf{p}_{n+1} + \mathbf{p}_{n+2} &= \frac{1 - u_{n+3}}{3} \mathbf{d}_n \\ \mathbf{p}_{n+2} &= \mathbf{q}_n.\end{aligned}$$

These equations can be directly solved, while the remaining  $n - 1$  unknowns

$$\bar{\mathbf{P}} = [\mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_{n-1}, \mathbf{p}_n]^T$$

are obtained by minimizing with respect to  $\bar{\mathbf{P}}$  the functional

$$L(\bar{\mathbf{P}}) = \text{tr}((\bar{\mathbf{Q}} - \bar{\mathbf{B}} \bar{\mathbf{P}})^T \bar{\mathbf{W}} (\bar{\mathbf{Q}} - \bar{\mathbf{B}} \bar{\mathbf{P}})) + \lambda \text{tr}((\bar{\mathbf{C}} \bar{\mathbf{P}} + \hat{\mathbf{P}})^T \mathbf{A} (\bar{\mathbf{C}} \bar{\mathbf{P}}) + \hat{\mathbf{P}}) \quad (8.38)$$

where the matrix  $\mathbf{A}$  has the expression (8.34),  $\bar{\mathbf{W}} = \text{diag}\{w_k\}$ ,  $k = 1, \dots, n-1$ ,

$$\bar{\mathbf{Q}} = \begin{bmatrix} \mathbf{q}_1^T - B_1^3(\bar{u}_1) \mathbf{p}_1^T \\ \mathbf{q}_2^T \\ \mathbf{q}_3^T \\ \vdots \\ \mathbf{q}_{n-2}^T \\ \mathbf{q}_{n-1}^T - B_{n+1}^3(\bar{u}_{n-1}) \mathbf{p}_{n+1}^T \end{bmatrix}$$

$$\bar{\mathbf{B}} = \begin{bmatrix} B_2^3(\bar{u}_1) & B_3^3(\bar{u}_1) & 0 & & \dots & 0 \\ B_2^3(\bar{u}_2) & B_3^3(\bar{u}_2) & B_4^3(\bar{u}_2) & 0 & & \vdots \\ 0 & B_3^3(\bar{u}_3) & \ddots & & & 0 \\ \vdots & 0 & & 0 & B_{n-2}^3(\bar{u}_{n-2}) & B_{n-1}^3(\bar{u}_{n-2}) & B_n^3(\bar{u}_{n-2}) \\ 0 & & \dots & & 0 & B_{n-1}^3(\bar{u}_{n-1}) & B_n^3(\bar{u}_{n-1}) \end{bmatrix}$$

$$\bar{\mathbf{C}} = \begin{bmatrix} c_{0,3} & 0 & \dots & & 0 \\ c_{1,2} & c_{1,3} & 0 & & \\ c_{2,1} & c_{2,2} & c_{2,3} & 0 & \\ 0 & c_{3,1} & c_{3,2} & c_{3,3} & 0 \\ \vdots & 0 & & \ddots & \\ 0 & & & 0 & c_{n-1,1} & c_{n-1,2} \\ & & & \dots & 0 & c_{n,3} \end{bmatrix}$$

$$\hat{\mathbf{P}} = \begin{bmatrix} c_{0,1} \mathbf{p}_0^T + c_{0,2} \mathbf{p}_1^T \\ c_{1,1} \mathbf{p}_1^T \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ c_{n-1,3} \mathbf{p}_{n+1}^T \\ c_{n,2} \mathbf{p}_{n+1}^T + c_{n,3} \mathbf{p}_{n+2}^T \end{bmatrix}$$

with the coefficients  $c_{k,i}$  defined by (8.36) and  $\mathbf{0} = [0, 0, \dots, 0]$  (of the same length of  $\mathbf{p}_k^T$ ).

By differentiating (8.38) with respect to  $\bar{\mathbf{P}}$  and setting the result to zero, the following equation is obtained

$$-(\bar{\mathbf{Q}} - \bar{\mathbf{B}} \bar{\mathbf{P}})^T \bar{\mathbf{W}} \bar{\mathbf{B}} + \lambda (\bar{\mathbf{C}} \bar{\mathbf{P}} + \hat{\mathbf{P}})^T \mathbf{A} \bar{\mathbf{C}} = 0$$

and the value which minimizes (8.31) is

$$\bar{\mathbf{P}} = \left( \bar{\mathbf{B}}^T \bar{\mathbf{W}} \bar{\mathbf{B}} + \lambda \bar{\mathbf{C}}^T \mathbf{A} \bar{\mathbf{C}} \right)^{-1} \left( \bar{\mathbf{B}}^T \bar{\mathbf{W}}^T \bar{\mathbf{Q}} - \lambda \bar{\mathbf{C}}^T \mathbf{A} \hat{\mathbf{P}} \right).$$

The spline is therefore defined by the control points

$$\mathbf{P} = [\mathbf{p}_0, \mathbf{p}_1, \bar{\mathbf{P}}^T, \mathbf{p}_{n+1}, \mathbf{p}_{n+2}]^T.$$

**Example 8.12** The interpolation of the via-points

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 & 4 & 5 & 6 \\ 0 & 2 & 3 & 3 & 2 & 0 \\ 0 & 1 & 0 & 0 & 2 & 2 \end{bmatrix}$$

by means of a cubic smoothing B-spline is reported in Fig. 8.24, with different values of the parameter  $\lambda$ , and  $w_k = 1, \forall k$ . Initial and final points are exactly interpolated; moreover, the initial and final tangent vectors are

$$\mathbf{t}_0 = [4.43, 8.87, 4.43]^T, \quad \mathbf{t}_5 = [4.85, -9.71, 0]^T.$$

The control points for  $\lambda = 10^{-4}$  are

$$\mathbf{P} = \begin{bmatrix} 0 & 0.33 & 0.82 & 2.13 & 4.07 & 5.03 & 5.66 & 6 \\ 0 & 0.66 & 1.92 & 3.11 & 3.17 & 2.15 & 0.66 & 0 \\ 0 & 0.33 & 0.90 & 0.05 & 0.03 & 1.94 & 2.00 & 2 \end{bmatrix}^T$$

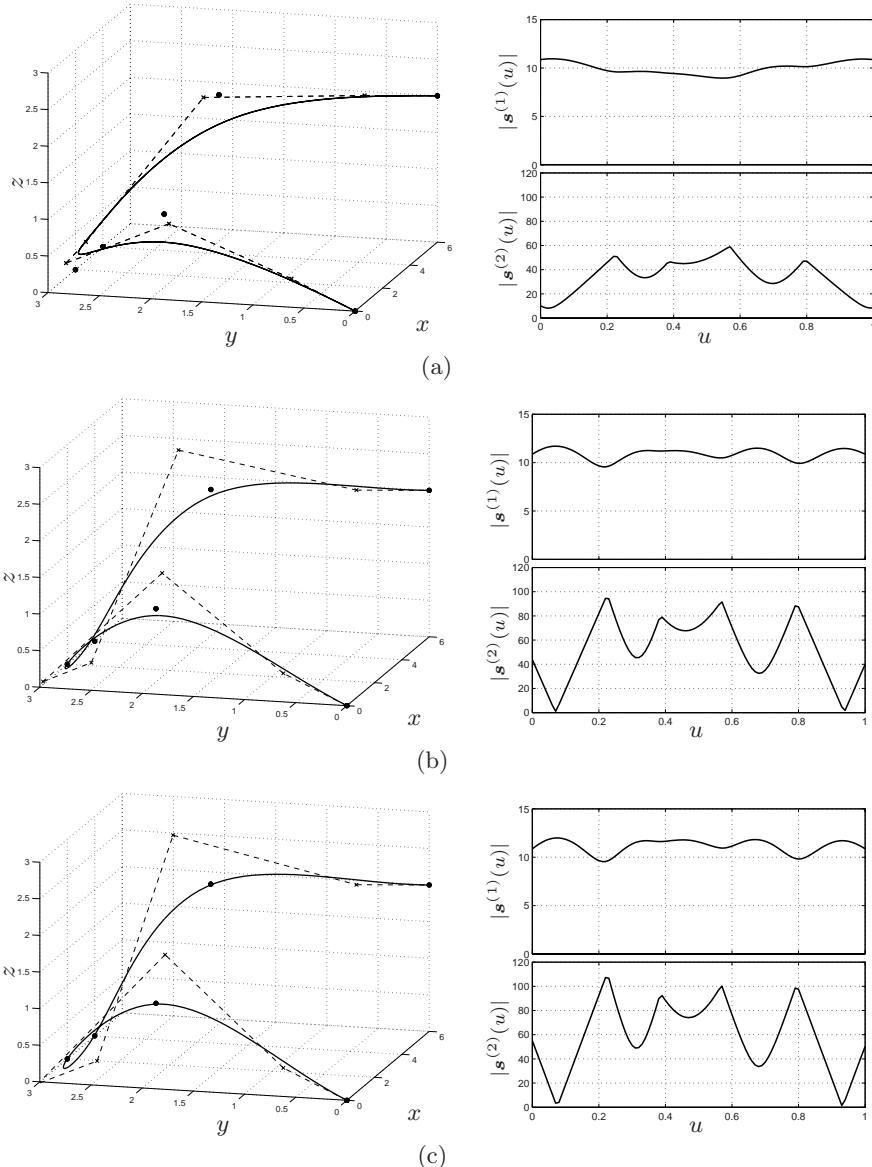
for  $\lambda = 10^{-5}$

$$\mathbf{P} = \begin{bmatrix} 0 & 0.33 & 0.82 & 1.90 & 4.44 & 4.87 & 5.66 & 6 \\ 0 & 0.66 & 1.91 & 3.22 & 3.09 & 2.29 & 0.66 & 0 \\ 0 & 0.33 & 1.52 & -0.24 & -0.36 & 2.53 & 2.00 & 2 \end{bmatrix}^T$$

and for  $\lambda = 10^{-6}$

$$\mathbf{P} = \begin{bmatrix} 0 & 0.33 & 0.87 & 1.81 & 4.56 & 4.80 & 5.66 & 6 \\ 0 & 0.66 & 1.89 & 3.26 & 3.05 & 2.34 & 0.66 & 0 \\ 0 & 0.33 & 1.69 & -0.33 & -0.43 & 2.67 & 2.00 & 2 \end{bmatrix}^T.$$

By assuming  $\lambda$  smaller and smaller, the error with respect to the via-points decreases (for  $\lambda = 0$ , exact interpolation is obtained). Therefore, given a desired tolerance  $\delta$ , it is possible to compute the value of  $\lambda$  which guarantees an error smaller than  $\delta$ . Moreover, it is possible to act on the weights  $w_k$  in order to selectively reduce the distance between the trajectory and some of the via-points, see [37]. On the other hand, when  $\lambda$  decreases, the value of the magnitude of the second derivative  $s^{(2)}(u)$  increases, see Fig. 8.24.  $\square$



**Fig. 8.24.** Interpolation by means of a smoothing B-spline (geometric path on the right, and velocity and acceleration on the left) for different values of  $\lambda$ : a)  $\lambda = 10^{-4}$ , b)  $\lambda = 10^{-5}$ , c)  $\lambda = 10^{-6}$ .

## 8.8 B-spline Functions for Trajectories with High Degree of Continuity

The techniques reported in the previous sections, for the interpolation or the approximation of data points in a multi-dimensional space by means of

B-spline curves, are mainly based on cubic functions. However, it is worth noticing that the use of B-splines with a degree  $p > 3$  can be necessary when it is required to plan trajectories with continuous derivatives up to an order  $r > 2$ , e.g. with continuous jerk, snap or even higher order derivatives.

The interpolation problem is considered here, although also all the other techniques discussed in the previous sections, i.e. approximating B-splines, smoothing B-splines, and so on, can be extended to the case  $p > 3$ .

Given a set of points  $\mathbf{q}_k$ ,  $k = 0, \dots, n$ , the control points  $\mathbf{p}_j$ ,  $j = 0, \dots, m$ , defining the B-spline function  $\mathbf{s}(u)$  can be determined by imposing

$$\mathbf{s}(\bar{u}_k) = \mathbf{q}_k, \quad k = 0, \dots, n$$

where  $\bar{u}_k$  are proper values of the independent variable  $u$ , which can be provided together with the points  $\mathbf{q}_k$  or computed with one of the methods reported in Sec. 8.4.1.

Once the degree  $p$  of the B-spline curve has been defined on the basis of the required degree of continuity<sup>17</sup>, it is necessary to build a proper knot vector  $\mathbf{u}$ . The standard choice is

$$\mathbf{u} = [\underbrace{\bar{u}_0, \dots, \bar{u}_0}_{p+1}, \bar{u}_1, \dots, \bar{u}_{n-1}, \underbrace{\bar{u}_n, \dots, \bar{u}_n}_{p+1}]. \quad (8.39)$$

In this case, the total number of knots is  $n_{knot} + 1 = n + 2p + 1$ . As a consequence, because of the relationship between  $n_{knot}$ ,  $m$ , and  $p$ , for a B-spline function (i.e.  $n_{knot} - p - 1 = m$ ), the number of unknown control points is  $m + 1 = (n + 1) + p - 1$ . An alternative choice of the knot vector is

$$\mathbf{u} = [\underbrace{\bar{u}_0, \dots, \bar{u}_0}_{p+1}, (\bar{u}_0 + \bar{u}_1)/2, \dots, (\bar{u}_{k-1} + \bar{u}_k)/2, \dots, (\bar{u}_{n-1} + \bar{u}_n)/2, \underbrace{\bar{u}_n, \dots, \bar{u}_n}_{p+1}]. \quad (8.40)$$

In this case, the knots are  $n_{knots} + 1 = n + 2p + 2$ , and, as a consequence, the number of control points to be determined is  $m + 1 = (n + 1) + p$ . The adoption of (8.39) or (8.40) is strictly related to the degree of the B-spline. In particular, as highlighted in Sec. 4.5, the fact that the degree  $p$  is odd or even strongly affects the trajectory profiles obtained with B-spline functions. If  $p$  is odd, the choice (8.39) is preferable while, if  $p$  is even, the knot vector expressed by (8.40) provides better results, [45].

---

<sup>17</sup> In order to obtain a curve  $r$  times differentiable, a value  $p > r$  must be chosen. As a matter of fact, a B-spline curve is, by definition,  $p-k$  continuously differentiable at a knot of multiplicity  $k$ , see Sec. B.1. Therefore, if all the internal knots are distinct ( $k = 1$ ), the continuity of velocity and acceleration simply requires the adoption of a B-spline of degree three ( $p = 3$ ). If a continuous jerk is required, it is necessary to set  $p = 4$ , while the condition  $p = 5$  guarantees also the continuity of the snap.

In order to determine the unknown coefficients  $\mathbf{p}_j$ ,  $j = 0, \dots, m$ , one can build a linear system by stacking the  $n + 1$  equations obtained by imposing the interpolation conditions of each point  $\mathbf{q}_k$  at  $\bar{u}_k$ :

$$\mathbf{q}_k^T = \left[ B_0^p(\bar{u}_k), B_1^p(\bar{u}_k), \dots, B_{m-1}^p(\bar{u}_k), B_m^p(\bar{u}_k) \right] \begin{bmatrix} \mathbf{p}_0^T \\ \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_{m-1}^T \\ \mathbf{p}_m^T \end{bmatrix}, \quad k = 0, \dots, n.$$

In this way, a system of  $n + 1$  equations in the  $m + 1$  unknown control points  $\mathbf{p}_j$  is obtained. In order to have a unique solution, more constraints must be imposed. In particular,  $p - 1$  or  $p$  additional equations, depending on the choice of  $\mathbf{u}$ , are necessary to obtain a square system with  $m + 1$  equations and  $m + 1$  unknown variables. Typical additional conditions concern the value of the trajectory derivatives at the initial and final points:

$$\begin{aligned} \mathbf{s}^{(1)}(\bar{u}_0) &= \mathbf{t}_0, & \mathbf{s}^{(1)}(\bar{u}_n) &= \mathbf{t}_n \\ \mathbf{s}^{(2)}(\bar{u}_0) &= \mathbf{n}_0, & \mathbf{s}^{(2)}(\bar{u}_n) &= \mathbf{n}_n \\ &\vdots & &\vdots \end{aligned}$$

where  $\mathbf{t}_k$ ,  $\mathbf{n}_k$  are respectively the tangent and curvature vectors at  $\bar{u}_k$ . These constraints can be written as

$$\begin{aligned} \mathbf{t}_k^T &= \left[ B_0^{p(1)}(\bar{u}_k), B_1^{p(1)}(\bar{u}_k), \dots, B_{m-1}^{p(1)}(\bar{u}_k), B_m^{p(1)}(\bar{u}_k) \right] \begin{bmatrix} \mathbf{p}_0^T \\ \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_{m-1}^T \\ \mathbf{p}_m^T \end{bmatrix}, \quad k = 0, n \\ \mathbf{n}_k^T &= \left[ B_0^{p(2)}(\bar{u}_k), B_1^{p(2)}(\bar{u}_k), \dots, B_{m-1}^{p(2)}(\bar{u}_k), B_m^{p(2)}(\bar{u}_k) \right] \begin{bmatrix} \mathbf{p}_0^T \\ \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_{m-1}^T \\ \mathbf{p}_m^T \end{bmatrix}, \quad k = 0, n \end{aligned}$$

where  $B_j^{p(i)}(\bar{u}_k)$  are the  $i$ -th derivatives of the basis functions  $B_j^p(u)$  computed at  $u = \bar{u}_k$ . For the calculation of  $B_j^{p(i)}(\bar{u}_k)$ , see Sec. B.1 in Appendix B.

Note that the generic equation

$$\left( \mathbf{s}^{(i)}(\bar{u}_k) \right)^T = \left[ B_0^{p(i)}(\bar{u}_k), B_1^{p(i)}(\bar{u}_k), \dots, B_{m-1}^{p(i)}(\bar{u}_k), B_m^{p(i)}(\bar{u}_k) \right] \begin{bmatrix} \mathbf{p}_0^T \\ \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_{m-1}^T \\ \mathbf{p}_m^T \end{bmatrix} \quad (8.41)$$

is equivalent to

$$\mathbf{s}^{(i)}(\bar{u}_k) = \sum_{j=0}^m \mathbf{p}_j B_j^{p(i)}(\bar{u}_k).$$

Alternatively, instead of assigning boundary conditions on the derivatives of the B-spline function, one may impose the continuity of the curve and of its derivatives at the initial and final time instants (the so-called periodic or cyclic condition), i.e.

$$\mathbf{s}^{(i)}(\bar{u}_0) = \mathbf{s}^{(i)}(\bar{u}_n)$$

or, in a matrix notation,

$$\left[ B_0^{p(i)}(\bar{u}_0) - B_0^{p(i)}(\bar{u}_n), B_1^{p(i)}(\bar{u}_0) - B_1^{p(i)}(\bar{u}_n), \dots, B_m^{p(i)}(\bar{u}_0) - B_m^{p(i)}(\bar{u}_n) \right] \begin{bmatrix} \mathbf{p}_0^T \\ \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_{m-1}^T \\ \mathbf{p}_m^T \end{bmatrix} = \mathbf{0}^T \quad (8.42)$$

where  $\mathbf{0}^T = [0, 0, \dots, 0]$  has the same dimensions of  $\mathbf{p}_j^T$ .

The conditions (8.41) or (8.42) can be mixed in order to obtain desired profiles.

If  $p = 4$ , the trajectory is differentiable up to the third derivative and, since with the choice of the knot vector expressed by (8.40) there are four free parameters to be determined, it is possible to assign initial and final values of the tangent and curvature vectors.

This leads to a linear system of  $(n+1)+4$  equations in  $(n+1)+4$  unknowns (in this case  $m = n + 4$ )

$$\mathbf{B} \mathbf{P} = \mathbf{R} \quad (8.43)$$

where

$$\mathbf{P} = [\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{m-1}, \mathbf{p}_m]^T$$

and (with  $p = 4$ )

$$\mathbf{B} = \begin{bmatrix} B_0^p(\bar{u}_0) & B_1^p(\bar{u}_0) & \cdots & B_m^p(\bar{u}_0) \\ B_0^{p(1)}(\bar{u}_0) & B_1^{p(1)}(\bar{u}_0) & \cdots & B_m^{p(1)}(\bar{u}_0) \\ B_0^{p(2)}(\bar{u}_0) & B_1^{p(2)}(\bar{u}_0) & \cdots & B_m^{p(2)}(\bar{u}_0) \\ B_0^p(\bar{u}_1) & B_1^p(\bar{u}_1) & \cdots & B_m^p(\bar{u}_1) \\ \vdots & \vdots & & \vdots \\ B_0^p(\bar{u}_{n-1}) & B_1^p(\bar{u}_{n-1}) & \cdots & B_m^p(\bar{u}_{n-1}) \\ B_0^{p(2)}(\bar{u}_n) & B_1^{p(2)}(\bar{u}_n) & \cdots & B_m^{p(2)}(\bar{u}_n) \\ B_0^{p(1)}(\bar{u}_n) & B_1^{p(1)}(\bar{u}_n) & \cdots & B_m^{p(1)}(\bar{u}_n) \\ B_0^p(\bar{u}_n) & B_1^p(\bar{u}_n) & \cdots & B_m^p(\bar{u}_n) \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{q}_0^T \\ \mathbf{t}_0^T \\ \mathbf{n}_0^T \\ \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_{n-1}^T \\ \mathbf{n}_n^T \\ \mathbf{t}_n^T \\ \mathbf{q}_n^T \end{bmatrix}. \quad (8.44)$$

The control points  $\mathbf{p}_j$ ,  $j = 0, \dots, m$ , are obtained by solving (8.43), and the B-spline can be evaluated for any value of  $u \in [\bar{u}_0, \bar{u}_n]$  according to the algorithm reported in Appendix B.

**Example 8.13** A B-spline of degree 4 is considered for the interpolation of the points

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} 3 & -2 & -5 & 0 & 6 & 12 & 8 \\ -1 & 0 & 2 & 4 & -9 & 7 & 3 \\ 0 & 0 & 0 & -2 & -1 & 3 & 0 \end{bmatrix}$$

with the further conditions on the initial and final tangent and curvature vectors

$$\mathbf{t}_0 = \begin{bmatrix} -30 \\ 10 \\ 0 \end{bmatrix}, \quad \mathbf{t}_6 = \begin{bmatrix} -20 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{n}_0 = \begin{bmatrix} -200 \\ 10 \\ 0 \end{bmatrix}, \quad \mathbf{n}_6 = \begin{bmatrix} 0 \\ 300 \\ 0 \end{bmatrix}.$$

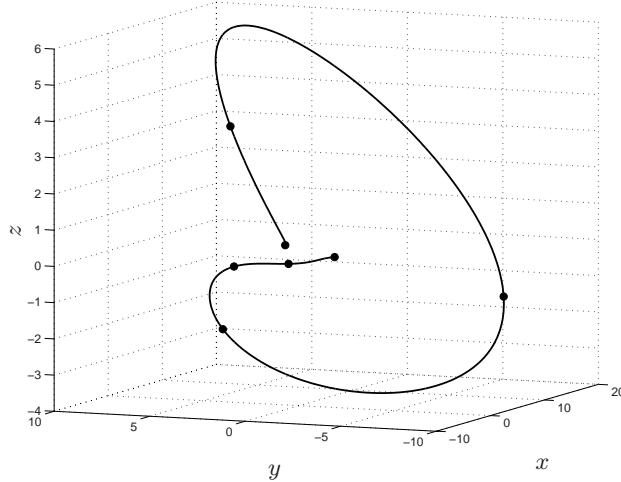
The parameters  $\bar{u}_k$  are assumed with a cord-length distribution within the range  $[0, 1]$ , i.e.

$$\bar{\mathbf{u}} = [0, 0.09, 0.16, 0.27, 0.54, 0.87, 1].$$

From these  $\bar{u}_k$  the knot vector is constructed according to (8.40), i.e.

$$\mathbf{u} = [0, 0, 0, 0, 0, 0.04, 0.13, 0.21, 0.40, 0.71, 0.93, 1, 1, 1, 1, 1]$$

and the matrices  $\mathbf{B}$  and  $\mathbf{R}$  result



**Fig. 8.25.** B-spline trajectory of degree 4.

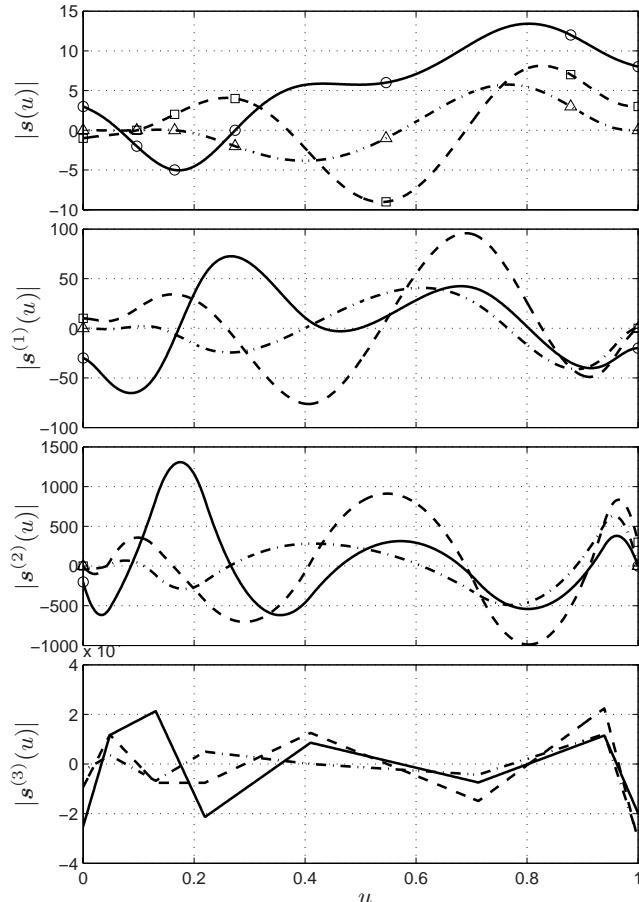
$$B = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -82.7 & 82.7 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 5137 & -7035 & 1897 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.01 & 0.29 & 0.57 & 0.13 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.01 & 0.43 & 0.50 & 0.05 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.04 & 0.54 & 0.38 & 0.03 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.01 & 0.29 & 0.50 & 0.19 & 0.01 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.02 & 0.24 & 0.59 & 0.14 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 687 & -3945 & 3258 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.00 & -0.00 & -0.00 & -0.00 & -65.9 & 65.9 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \end{bmatrix}$$

and

$$R = \begin{bmatrix} 3.00 & -1.00 & 0.00 \\ -30.00 & 10.00 & 0.00 \\ -200.00 & 0.00 & 0.00 \\ -2.00 & 0.00 & 0.00 \\ -5.00 & 2.00 & 0.00 \\ 0.00 & 4.00 & -2.00 \\ 6.00 & -9.00 & -1.00 \\ 12.00 & 7.00 & 3.00 \\ 0.00 & 300.00 & 0.00 \\ -20.00 & 0.00 & 0.00 \\ 8.00 & 3.00 & 0.00 \end{bmatrix}$$

By solving (8.43) the control points, defining the trajectory shown in Fig. 8.25, are obtained:

$$P = \begin{bmatrix} 3.00 & 2.64 & 1.55 & -2.21 & -9.57 & 14.12 & -4.12 & 21.18 & 9.74 & 8.30 & 8.00 \\ -1.00 & -0.88 & -0.55 & -0.54 & 3.58 & 8.27 & -31.18 & 21.69 & 3.44 & 3.00 & 3.00 \\ 0.00 & 0.00 & 0.00 & -0.16 & 0.78 & -6.01 & -3.33 & 12.61 & -0.00 & 0.00 & -0.00 \end{bmatrix}^T$$



**Fig. 8.26.** Components of a B-spline trajectory of degree 4 and of its derivatives (component x - solid, component y - dashed, component z - dashdot).

In Fig. 8.26 the components of the B-spline curve and of its derivatives are reported. Note the continuity of the first three derivatives.  $\square$

For  $p = 4$ , one can assign cyclic conditions for the derivatives up to the fourth order, since the choice (8.40) requires four additional constraints. However, since in any case the fourth derivative will be discontinuous in the interior of the trajectory, it is more useful to impose other conditions in place of  $s^{(4)}(\bar{u}_0) = s^{(4)}(\bar{u}_n)$ . For example, besides the periodic conditions on the first three derivatives, one can assume a boundary condition on the tangent or curvature vector. The system (8.43) is now defined by

$$\mathbf{B} = \begin{bmatrix} B_0^p(\bar{u}_0) & B_1^p(\bar{u}_0) & \cdots & B_m^p(\bar{u}_0) \\ B_0^p(\bar{u}_1) & B_1^p(\bar{u}_1) & \cdots & B_m^p(\bar{u}_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_0^p(\bar{u}_{n-1}) & B_1^p(\bar{u}_{n-1}) & \cdots & B_m^p(\bar{u}_{n-1}) \\ B_0^p(\bar{u}_n) & B_1^p(\bar{u}_n) & \cdots & B_m^p(\bar{u}_n) \\ B_0^{p(1)}(\bar{u}_n) - B_0^{p(1)}(\bar{u}_0) & B_1^{p(1)}(\bar{u}_n) - B_1^{p(1)}(\bar{u}_0) & \cdots & B_m^{p(1)}(\bar{u}_n) - B_m^{p(1)}(\bar{u}_0) \\ B_0^{p(2)}(\bar{u}_n) - B_0^{p(2)}(\bar{u}_0) & B_1^{p(2)}(\bar{u}_n) - B_1^{p(2)}(\bar{u}_0) & \cdots & B_m^{p(2)}(\bar{u}_n) - B_m^{p(2)}(\bar{u}_0) \\ B_0^{p(3)}(\bar{u}_n) - B_0^{p(3)}(\bar{u}_0) & B_1^{p(3)}(\bar{u}_n) - B_1^{p(3)}(\bar{u}_0) & \cdots & B_m^{p(3)}(\bar{u}_n) - B_m^{p(3)}(\bar{u}_0) \\ B_0^{p(4)}(\bar{u}_n) - B_0^{p(4)}(\bar{u}_0) & B_1^{p(4)}(\bar{u}_n) - B_1^{p(4)}(\bar{u}_0) & \cdots & B_m^{p(4)}(\bar{u}_n) - B_m^{p(4)}(\bar{u}_0) \end{bmatrix}$$

$$\mathbf{R} = [\mathbf{q}_0^T, \mathbf{q}_1^T, \dots, \mathbf{q}_{n-1}^T, \mathbf{q}_n^T, \mathbf{0}^T, \mathbf{0}^T, \mathbf{0}^T]^T.$$

**Example 8.14** A B-spline trajectory of degree 4 interpolating the points

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} 3 & -2 & -5 & 0 & 6 & 12 & 3 \\ -1 & 0 & 2 & 4 & -9 & 7 & -1 \\ 0 & 0 & 0 & -2 & -1 & 3 & 0 \end{bmatrix}$$

is defined by imposing the periodic conditions

$$\begin{aligned} \mathbf{s}^{(1)}(\bar{u}_0) &= \mathbf{s}^{(1)}(\bar{u}_6) \\ \mathbf{s}^{(2)}(\bar{u}_0) &= \mathbf{s}^{(2)}(\bar{u}_6) \\ \mathbf{s}^{(3)}(\bar{u}_0) &= \mathbf{s}^{(3)}(\bar{u}_6) \\ \mathbf{s}^{(4)}(\bar{u}_0) &= \mathbf{s}^{(4)}(\bar{u}_6). \end{aligned}$$

Note that the first and the last points are coincident. By assuming the same knot vector of the previous example<sup>18</sup>

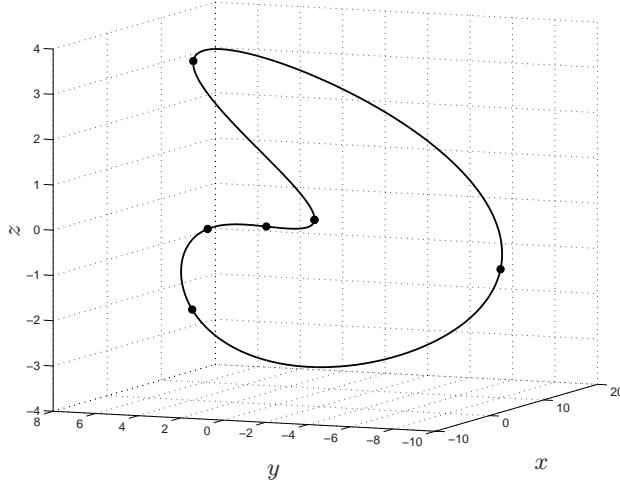
$$\mathbf{u} = [0, 0, 0, 0, 0, 0.04, 0.13, 0.21, 0.40, 0.71, 0.93, 1, 1, 1, 1]$$

the matrices  $\mathbf{B}$  and  $\mathbf{R}$  result

$$\mathbf{B} = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.01 & 0.29 & 0.57 & 0.13 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.01 & 0.43 & 0.50 & 0.05 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.04 & 0.54 & 0.38 & 0.03 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.01 & 0.32 & 0.50 & 0.16 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.09 & 0.42 & 0.45 & 0.04 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ -92.19 & 92.19 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 37.88 & -37.88 & \\ 6374 & -8729 & 2354 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -315 & 1391 & -1076 \\ -293845 & 442487 & -172546 & 23904 & 0.00 & 0.00 & 0.00 & 997 & -8716 & 28103 & -20384 \\ 6772579 & -10539889 & 4643117 & -940762 & 64953 & 0.00 & -1242 & 15035 & -91745 & 270998 & -193045 \end{bmatrix}$$

and

<sup>18</sup> With other choices of the knot vector, e.g. computed by means of (8.40), an ill-conditioned matrix  $\mathbf{B}$  can be obtained.



**Fig. 8.27.** B-spline curve of degree 4 computed with periodic conditions.

$$\mathbf{R} = \begin{bmatrix} 3.00 & -1.00 & 0.00 \\ -2.00 & 0.00 & 0.00 \\ -5.00 & 2.00 & 0.00 \\ 0.00 & 4.00 & -2.00 \\ 6.00 & -9.00 & -1.00 \\ 12.00 & 7.00 & 3.00 \\ 3.00 & -1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 \end{bmatrix}.$$

The control points obtained with these values are

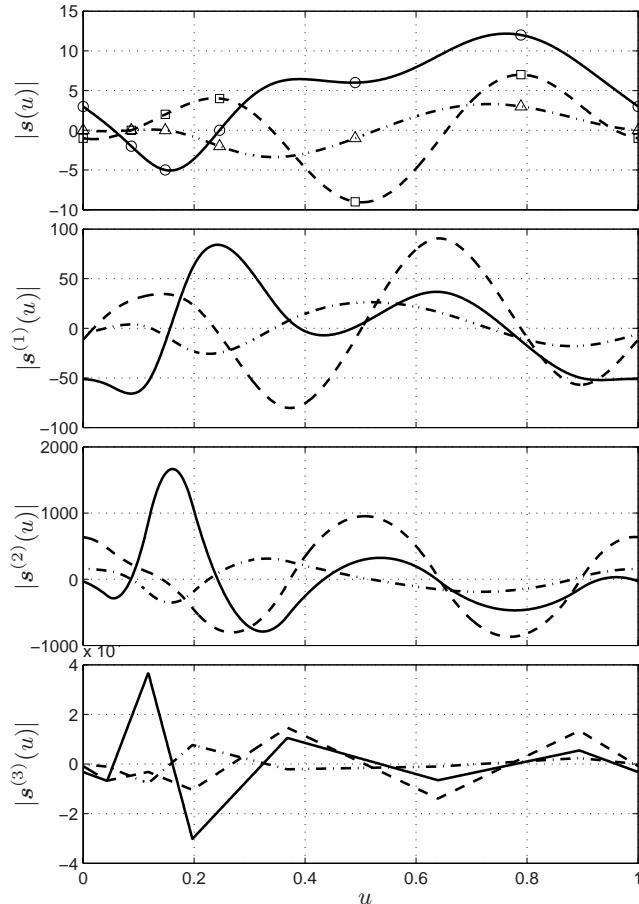
$$\mathbf{P} = \begin{bmatrix} 3.00 & 2.45 & 0.93 & -1.80 & -9.93 & 14.40 & -3.13 & 19.40 & 8.86 & 4.35 & 3.00 \\ -1.00 & -1.13 & -1.20 & -0.13 & 3.23 & 8.36 & -29.71 & 20.64 & 2.37 & -0.69 & -1.00 \\ 0.00 & -0.06 & -0.17 & -0.06 & 0.71 & -6.20 & 0.13 & 5.92 & 1.17 & 0.16 & -0.00 \end{bmatrix}^T.$$

The B-spline curve and its components are shown in Fig. 8.27 and Fig. 8.28 respectively. □

With  $p = 5$ , the knots should be selected according to (8.39), i.e.

$$\mathbf{u} = [\underbrace{\bar{u}_0, \dots, \bar{u}_0}_{6}, \bar{u}_1, \dots, \bar{u}_{n-1}, \underbrace{\bar{u}_n, \dots, \bar{u}_n}_{6}].$$

It is therefore possible to impose the desired values of the initial and final tangent and curvature vectors (four conditions). Moreover, this value of  $p$



**Fig. 8.28.** Components of a B-spline trajectory of degree 4 computed with periodic conditions (component x - solid, component y - dashed, component z - dashdot).

guarantees the continuity of the trajectory up to the fourth derivative (snap). The system (8.43) with the expressions of  $\mathbf{B}$  and  $\mathbf{R}$  reported in (8.44) remains unchanged (with obvious differences on the B-basis functions, that are computed for a different degree  $p$  and for a different knot vector  $\mathbf{u}$ ).

**Example 8.15** The data (via-points and constraints) of Example 8.13 are used to build a B-spline of degree 5, with the knot vector constructed as in (8.39), that is

$$\mathbf{u} = [0, 0, 0, 0, 0, 0, 0.09, 0.16, 0.27, 0.54, 0.87, 1, 1, 1, 1, 1].$$

By solving (8.43), with

$$\mathbf{B} = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -51.7 & 51.7 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 2140 & -3394 & 1254 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.03 & 0.37 & 0.50 & 0.10 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.04 & 0.50 & 0.41 & 0.05 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.11 & 0.54 & 0.32 & 0.04 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.04 & 0.30 & 0.43 & 0.21 & 0.02 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.02 & 0.17 & 0.51 & 0.29 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 362 & -1720 & 1357 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.00 & -0.00 & -0.00 & -0.00 & -41.1 & 41.1 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix}$$

and

$$\mathbf{R} = \begin{bmatrix} 3.00 & -1.00 & 0.00 \\ -30.00 & 10.00 & 0.00 \\ -200.00 & 0.00 & 0.00 \\ -2.00 & 0.00 & 0.00 \\ -5.00 & 2.00 & 0.00 \\ 0.00 & 4.00 & -2.00 \\ 6.00 & -9.00 & -1.00 \\ 12.00 & 7.00 & 3.00 \\ 0.00 & 300.00 & 0.00 \\ -20.00 & 0.00 & 0.00 \\ 8.00 & 3.00 & 0.00 \end{bmatrix}$$

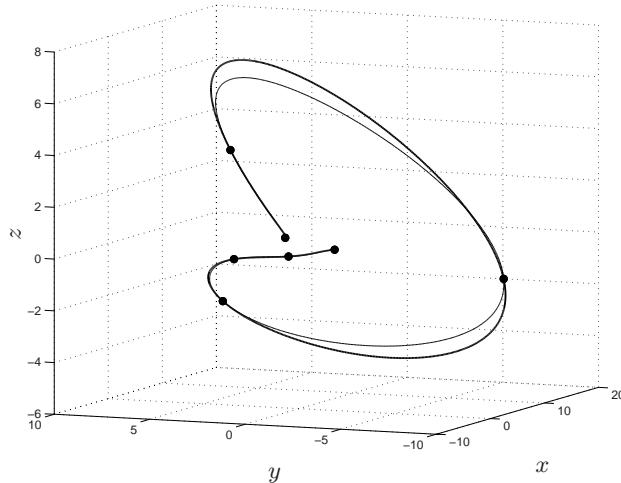
the control points

$$\mathbf{P} = \begin{bmatrix} 3.00 & 2.42 & 1.27 & -2.85 & -11.90 & 22.36 & -13.53 & 25.87 & 10.30 & 8.49 & 8.00 \\ -1.00 & -0.81 & -0.48 & -0.57 & 4.37 & 10.06 & -42.66 & 29.01 & 3.83 & 3.00 & 3.00 \\ 0.00 & 0.00 & 0.00 & -0.20 & 1.29 & -7.75 & -5.71 & 17.91 & -0.00 & 0.00 & -0.00 \end{bmatrix}^T$$

are obtained. The geometric path described by the B-spline trajectory is reported in Fig. 8.29 superimposed to the one of degree 4. The components of the trajectory and of its derivatives are shown in Fig. 8.30.

□

In general, to build a trajectory  $r$  times differentiable which interpolates  $n+1$  points, it is sufficient to consider a B-spline of degree  $p = r+1$ . If  $p$  is odd, it is convenient to assume a knot distribution as in (8.39) while, if  $p$  is even, a knot vector of the kind (8.40) provides better results. In the former case,  $p-1$  additional constraints are necessary to have a unique solution (in this case the unknown control points are  $m+1 = n+p$ ) while, in the latter one,  $p$  additional condition are needed (in this case the unknown control points are  $m+1 = n+p+1$ ). By adding further knots (and accordingly by increasing the number of control points  $\mathbf{p}_j$ ), it is also possible to consider additional constraints. For instance, one could desire to plan a trajectory interpolating a set of points with given tangent directions. Therefore, besides the points  $\mathbf{q}_k$ ,



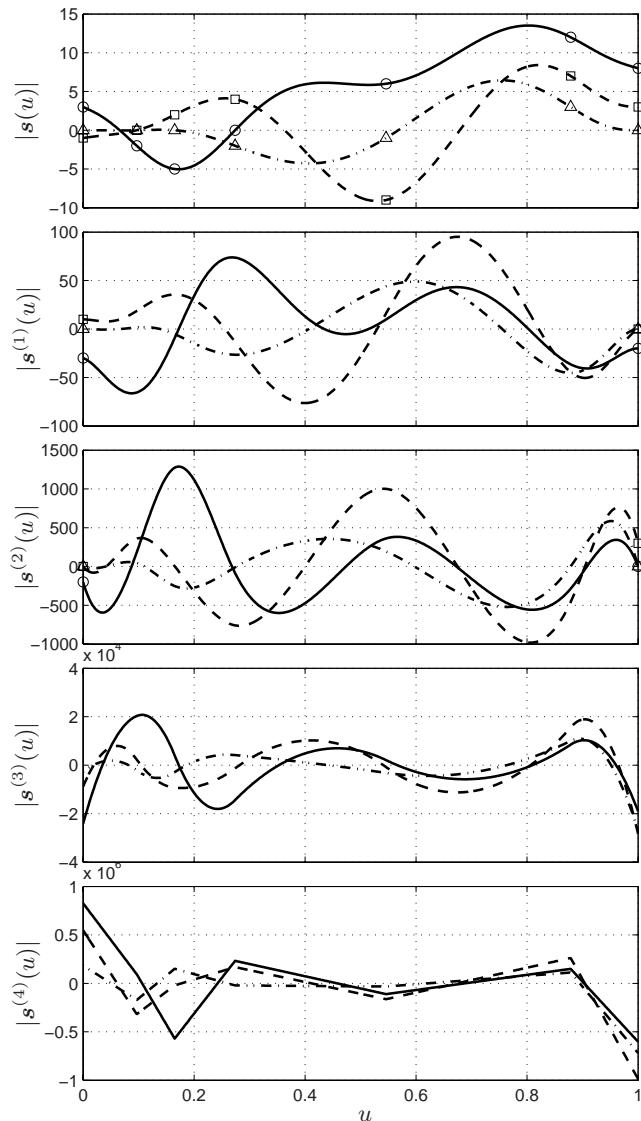
**Fig. 8.29.** B-spline curve of degree 5 superimposed to the B-spline of degree 4 (thin line).

$k = 0, \dots, n$ , (to be interpolated at  $\bar{u}_k$ ) it is necessary to provide also the tangent vectors  $\mathbf{t}_k$ . In this case, the knot vector can be defined as

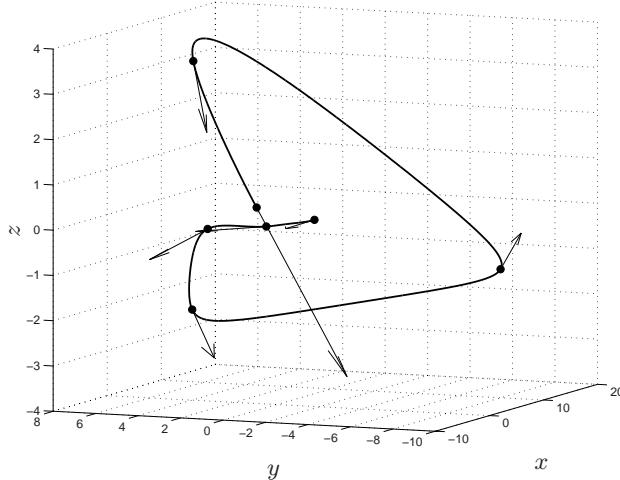
$$\mathbf{u} = [\underbrace{\bar{u}_0, \dots, \bar{u}_0}_{p+1}, (\bar{u}_0 + \bar{u}_1)/2, \bar{u}_1, \dots, \bar{u}_{k-1}, (\bar{u}_{k-1} + \bar{u}_k)/2, \bar{u}_k, \dots, \bar{u}_{n-1}, (\bar{u}_{n-1} + \bar{u}_n)/2, \underbrace{\bar{u}_n, \dots, \bar{u}_n}_{p+1}]. \quad (8.45)$$

Therefore the number of knots is  $n_{knot} + 1 = 2(n + p) + 1$ , and the number of unknown control points is  $m + 1 = 2n + p$ . Accordingly, it is possible to impose the  $n + 1$  interpolation conditions, the  $n + 1$  desired tangent vectors, and  $p - 2$  additional constraints (e.g. in the case  $p = 4$  the curvature vectors  $\mathbf{n}_0$  and  $\mathbf{n}_n$  at the endpoints) in order to univocally determine the control points  $\mathbf{p}_j$ .

For  $p = 4$  the matrices of the system (8.43) are



**Fig. 8.30.** Components of a B-spline trajectory of degree 5 (component x - solid, component y - dashed, component z - dashdot).



**Fig. 8.31.** B-spline curve of degree 4 crossing the via-points with a given direction.

$$\mathbf{B} = \begin{bmatrix} B_0^p(\bar{u}_0) & B_1^p(\bar{u}_0) & \cdots & B_m^p(\bar{u}_0) \\ B_0^{p(1)}(\bar{u}_0) & B_1^{p(1)}(\bar{u}_0) & \cdots & B_m^{p(1)}(\bar{u}_0) \\ B_0^{p(2)}(\bar{u}_0) & B_1^{p(2)}(\bar{u}_0) & \cdots & B_m^{p(2)}(\bar{u}_0) \\ B_0^p(\bar{u}_1) & B_1^p(\bar{u}_1) & \cdots & B_m^p(\bar{u}_1) \\ B_0^{p(1)}(\bar{u}_1) & B_1^{p(1)}(\bar{u}_1) & \cdots & B_m^{p(1)}(\bar{u}_1) \\ B_0^p(\bar{u}_2) & B_1^p(\bar{u}_2) & \cdots & B_m^p(\bar{u}_2) \\ \vdots & \vdots & & \vdots \\ B_0^p(\bar{u}_{n-1}) & B_1^p(\bar{u}_{n-1}) & \cdots & B_m^p(\bar{u}_{n-1}) \\ B_0^{p(1)}(\bar{u}_{n-1}) & B_1^{p(1)}(\bar{u}_{n-1}) & \cdots & B_m^{p(1)}(\bar{u}_{n-1}) \\ B_0^{p(2)}(\bar{u}_n) & B_1^{p(2)}(\bar{u}_n) & \cdots & B_m^{p(2)}(\bar{u}_n) \\ B_0^{p(1)}(\bar{u}_n) & B_1^{p(1)}(\bar{u}_n) & \cdots & B_m^{p(1)}(\bar{u}_n) \\ B_0^p(\bar{u}_n) & B_1^p(\bar{u}_n) & \cdots & B_m^p(\bar{u}_n) \end{bmatrix}, \mathbf{R} = \begin{bmatrix} \mathbf{q}_0^T \\ \mathbf{t}_0^T \\ \mathbf{n}_0^T \\ \mathbf{q}_1^T \\ \mathbf{t}_1^T \\ \mathbf{n}_1^T \\ \vdots \\ \mathbf{q}_{n-1}^T \\ \mathbf{t}_{n-1}^T \\ \mathbf{n}_{n-1}^T \\ \mathbf{q}_n^T \end{bmatrix}. \quad (8.46)$$

**Example 8.16** Fig. 8.31 shows a B-spline trajectory interpolating the via-points of Example 8.13 with given directions. Therefore, the trajectory is built by considering not only the via-points  $\mathbf{q}_k$ , but also the tangent vectors at each point, which are computed according to the algorithm reported in Sec. 8.10.1:

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} -56.32 & -47.14 & -9.26 & 39.10 & 20.25 & -19.32 & -46.59 \\ -0.74 & 21.43 & 25.06 & -0.54 & -4.65 & -11.29 & -54.62 \\ 0.00 & 0.00 & -7.08 & -12.07 & 7.43 & -14.90 & -34.54 \end{bmatrix}.$$

Moreover, two additional constraints on the curvature vectors at the initial and final points are considered:

$$\mathbf{n}_0 = \begin{bmatrix} 300 \\ 200 \\ 0 \end{bmatrix}, \quad \mathbf{n}_6 = \begin{bmatrix} 0 \\ -200 \\ 0 \end{bmatrix}.$$

The knot vector is constructed as in (8.45), that is

$$\mathbf{u} = [0, 0, 0, 0, 0, 0.04, 0.09, 0.13, 0.16, 0.21, 0.27, 0.40, 0.54, 0.71, 0.87, 0.9393, 1, 1, 1, 1]$$

and the matrices of the system (8.43) are

$$\mathbf{B} = [\mathbf{B}_1 \ \mathbf{B}_2]$$

with

$$\mathbf{B}_1 = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -82.7 & 82.7 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 5137.00 & -770.006 & 2569.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.03 & 0.35 & 0.55 & 0.07 & 0.00 & 0.00 \\ 0.00 & 0.00 & -3.31 & -12.43 & 10.07 & 5.67 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.09 & 0.58 & 0.33 & 0.01 \\ 0.00 & 0.00 & 0.00 & 0.00 & -6.37 & -7.97 & 13.02 & 1.32 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.19 & 0.60 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -5.68 & -0.15 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}$$

$$\mathbf{B}_2 = \begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.20 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 5.45 & 0.39 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.07 & 0.50 & 0.39 & 0.03 & 0.00 & 0.00 & 0.00 & 0.00 \\ -1.69 & -2.68 & 3.38 & 0.99 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.18 & 0.57 & 0.24 & 0.00 & 0.00 \\ 0.00 & 0.00 & -0.31 & -4.40 & -1.17 & 5.89 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1629.00 & -4887.00 & 3258.00 \\ 0.00 & 0.00 & 0.00 & -0.00 & -0.00 & -0.00 & -65.90 & 65.90 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix}$$

and

$$\mathbf{R} = \begin{bmatrix} 3.00 & -1.00 & 0.00 \\ -56.32 & -0.74 & 0.00 \\ 300.00 & 200.00 & 0.00 \\ -2.00 & 0.00 & 0.00 \\ -47.14 & 21.43 & 0.00 \\ -5.00 & 2.00 & 0.00 \\ -9.26 & 25.06 & -7.08 \\ 0.00 & 4.00 & -2.00 \\ 39.10 & -0.54 & -12.07 \\ 6.00 & -9.00 & -1.00 \\ 20.25 & -4.65 & 7.43 \\ 12.00 & 7.00 & 3.00 \\ -19.32 & -11.29 & -14.90 \\ 0.00 & -200 & 0.00 \\ -46.59 & -54.62 & -34.54 \\ 8.00 & 3.00 & 0.00 \end{bmatrix}.$$

By inverting (8.43) one obtains

$$\mathbf{P} = \begin{bmatrix} 3.00 & 2.32 & 1.08 & -0.90 & -2.43 & -5.34 & -5.30 & 1.22 & 1.30 & 5.47 & 6.82 \\ -1.00 & -1.01 & -0.95 & -0.51 & 0.14 & 1.86 & 2.65 & 4.71 & 3.56 & -10.72 & -10.55 \\ 0.00 & 0.00 & 0.00 & 0.04 & -0.05 & 0.18 & -0.21 & -2.44 & -2.43 & -1.02 & -1.21 \\ 14.20 & 12.16 & 10.12 & 8.71 & 8.00 \\ 8.52 & 7.37 & 5.36 & 3.83 & 3.00 \\ 4.74 & 3.10 & 1.57 & 0.52 & -0.00 \end{bmatrix}^T.$$

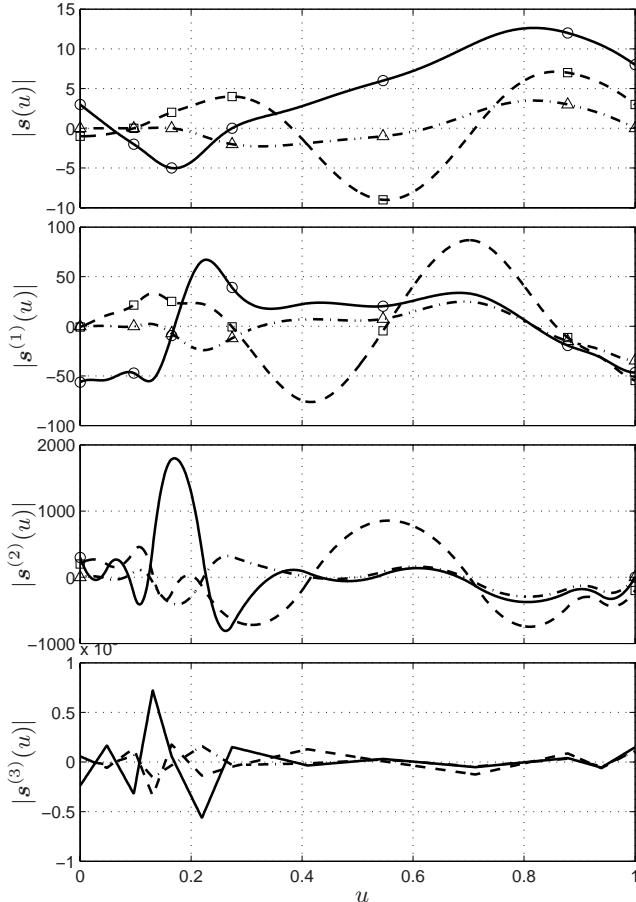
The components of the trajectory and its derivatives are shown in Fig. 8.32. Note that not only the trajectory interpolates the via-points, but also the profile of the first derivative crosses  $(\bar{u}_k, t_k)$ ,  $k = 0, \dots, n$ .  $\square$

## 8.9 Use of Nurbs for Trajectory Generation

Non-Uniform Rational B-Splines (Nurbs) are standard curves in CAD/CAM systems [87] since they offer an exact representation of a wide variety of common curves such as circles, parabolas, ellipses, lines, and hyperbolas, [88, 38]. For this reason they are adopted for trajectories description in many CNC (Computer Numerical Control) systems.

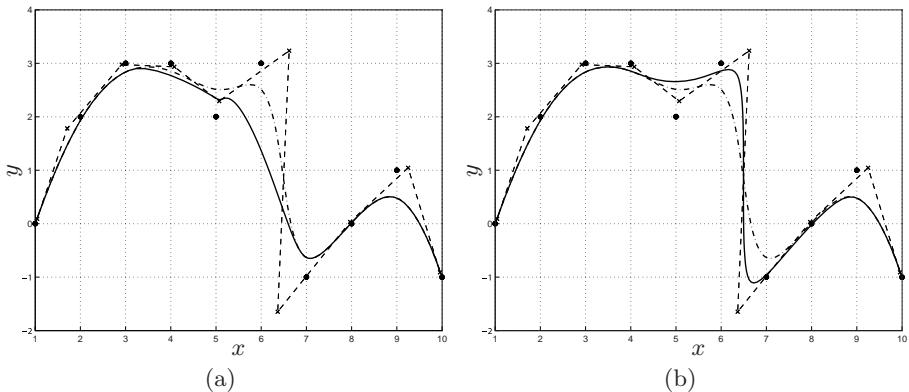
They are generalizations of non-rational B-splines, and their expression is

$$\mathbf{n}(u) = \frac{\sum_{j=0}^m \mathbf{p}_j w_j \mathbf{B}_j^p(u)}{\sum_{j=0}^m w_j \mathbf{B}_j^p(u)}, \quad u_{min} \leq u \leq u_{max} \quad (8.47)$$



**Fig. 8.32.** Components of a B-spline trajectory of degree 4 with constraints on positions and tangent vectors (component x - solid, component y - dashed, component z - dashdot).

where  $B_j^p(u)$  are standard B-spline basis functions,  $p_j$  the control points and  $w_j$  the weights associated to each control point (for a more detailed description on Nurbs curves and their properties see Sec. B.2). The techniques reported in previous sections based on B-spline functions can be adopted with minor modifications in the case of Nurbs (only the basis functions change) but de facto almost all the interpolation and approximation methods used in the literature are based on B-spline representations since the weights  $w_j$  are generally kept constant. As a matter of fact, by definition, Nurbs curves are standard B-splines when  $w_j = 1$ ,  $j = 0, \dots, m$ . Afterward the weights can be modified in order to iteratively change the shape of the trajectory obtained as a B-spline function.



**Fig. 8.33.** Interpolation by means of smoothing B-spline (dashdot) converted in a Nurbs ( $w_j = 1$ ), and effects of the modification of some weights.

**Example 8.17** In Fig. 8.33 the interpolation of the via-points

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 0 & 2 & 3 & 3 & 2 & 3 & -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

by means of a cubic smoothing B-spline is reported, with  $\lambda = 10^{-5}$ , and the weights  $w_k$  in (8.31) equal to one. The spline is then converted in a Nurbs by assigning to each of the 12 control points a unitary weight. By changing the weights  $w_j$ ,  $j = 0, \dots, 11$ , it is possible to modify the shape of the curve, which for increasing values of a specific weight moves towards the corresponding control point. In the figure, two different cases are shown. In Fig. 8.33(a), it is assumed  $w_5 = 10$ , while in Fig. 8.33(b)  $w_6$  and  $w_7$  are both equal to five. □

As mentioned above, Nurbs are widespread in CAD/CAM environments since they allow to exactly represent conic curves, such as circles, parabolas, ellipses, etc. There exist a number of techniques to construct the Nurbs curves in order to represent the desired figures, but these method are out of the purposes of this book. Interested readers should refer to the literature dedicated to this topic, see e.g. [38].

## 8.10 Local Interpolation with Bézier Curves

When the trajectory has to be constructed in an interactive manner (and the set of the via-points is not completely known in advance) the use of *Bézier curves* provides some important advantages, see Appendix B.3. Bézier curves can also be used when the goal is to find a trajectory based only on local data,

i.e. a pair of via-points, with prescribed tangent and curvature vectors [89]. Differently from the global methods reported in previous sections, the techniques based on Bézier curves need not only the knowledge of the points to be interpolated, but also of the derivatives<sup>19</sup> at these points in order to guarantee the desired level of smoothness. Often, these vectors are imposed by the task to be performed, but sometimes it is necessary to compute the tangent directions and possibly also the curvature directions from the via-points distribution.

### 8.10.1 Computation of the tangent and curvature vectors

Several methods exist for the computation of the tangent vectors for a given set of points, see [38, 90]. Given a sequence of points  $\mathbf{q}_k$ , to be interpolated at the “time instants”  $\bar{u}_k$ ,  $k = 0, \dots, n$ , the simplest way to define the tangent and curvature directions at a point  $\mathbf{q}_k$  is

$$\mathbf{t}_k = \frac{\mathbf{q}_{k+1} - \mathbf{q}_{k-1}}{\bar{u}_{k+1} - \bar{u}_{k-1}}, \quad \mathbf{n}_k = \frac{\frac{\mathbf{q}_{k+1} - \mathbf{q}_k}{\bar{u}_{k+1} - \bar{u}_k} - \frac{\mathbf{q}_k - \mathbf{q}_{k-1}}{\bar{u}_k - \bar{u}_{k-1}}}{\bar{u}_{k+1} - \bar{u}_{k-1}}. \quad (8.48)$$

Some authors [91, 90] adopt the tangent vectors and the curvature vectors of a B-spline curve, computed on the set of points in order to construct a trajectory approximating the B-spline but with additional features (e.g. an arc-length parameterization). Among the many techniques for the computation of tangent vectors, a method frequently adopted to estimate the derivative at each point is

$$\mathbf{t}_k = (1 - \alpha_k) \boldsymbol{\delta}_k + \alpha_k \boldsymbol{\delta}_{k+1}, \quad \text{for } k = 1, \dots, n-1 \quad (8.49)$$

where

$$\begin{aligned} \boldsymbol{\delta}_k &= \frac{\Delta \mathbf{q}_k}{\Delta \bar{u}_k}, & \text{for } k = 1, \dots, n \\ \alpha_k &= \frac{\Delta \bar{u}_k}{\Delta \bar{u}_k + \Delta \bar{u}_{k+1}}, & \text{for } k = 1, \dots, n-1 \end{aligned}$$

being

$$\Delta \mathbf{q}_k = \mathbf{q}_k - \mathbf{q}_{k-1}, \quad \Delta \bar{u}_k = \bar{u}_k - \bar{u}_{k-1}.$$

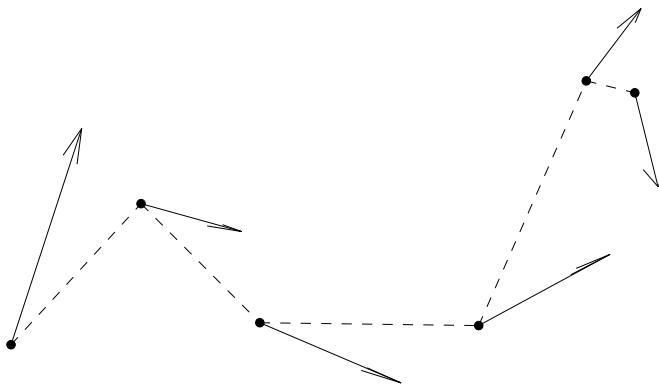
Equation (8.49) can be used only for the internal points, while the computation of the tangent vectors at endpoints requires special methods. In particular, they can be defined as

$$\mathbf{t}_0 = 2\boldsymbol{\delta}_1 - \mathbf{t}_1, \quad \mathbf{t}_n = 2\boldsymbol{\delta}_n - \mathbf{t}_{n-1}.$$

The tangent vectors for a set of points in the three-dimensional space are shown in Fig. 8.34.

---

<sup>19</sup> In general, the first and second derivatives are used.



**Fig. 8.34.** Computation of the tangent vectors for local interpolation.

### 8.10.2 Cubic Bézier curves interpolation

The generation of a  $G^1$  trajectory<sup>20</sup> requires the use of Bézier curves at least of degree 3, i.e.

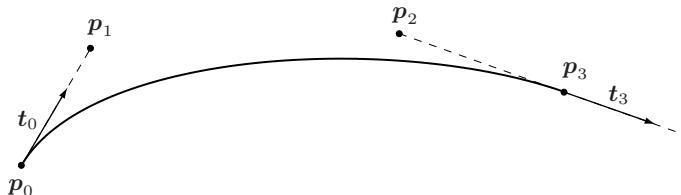
$$\mathbf{b}(u) = (1-u)^3 \mathbf{p}_0 + 3u(1-u)^2 \mathbf{p}_1 + 3u^2(1-u) \mathbf{p}_2 + u^3 \mathbf{p}_3, \quad u \in [0, 1].$$

While the control points  $\mathbf{p}_0$  and  $\mathbf{p}_3$  are generally given, since they are coincident with the two via-points to be joined, the remaining control points can be determined by imposing prescribed tangent directions, expressed by the unit vectors  $\mathbf{t}_0$  and  $\mathbf{t}_3$  at  $\mathbf{p}_0$  and  $\mathbf{p}_3$  respectively, see Fig. 8.35. In particular,  $\mathbf{p}_1$  and  $\mathbf{p}_2$  are chosen so that the first derivatives in the central point and at the endpoints of the Bézier curve have the same magnitude:

$$|\mathbf{b}^{(1)}(0)| = |\mathbf{b}^{(1)}(1)| = |\mathbf{b}^{(1)}(1/2)| = \alpha. \quad (8.50)$$

In this way, the velocities at the endpoints and at the midpoint are equal. From (B.24) it follows that

$$\mathbf{p}_1 = \mathbf{p}_0 + \frac{1}{3}\alpha \mathbf{t}_0, \quad \mathbf{p}_2 = \mathbf{p}_3 - \frac{1}{3}\alpha \mathbf{t}_3 \quad (8.51)$$



**Fig. 8.35.** Bézier curve of degree 3 and its control points.

---

<sup>20</sup> In many applications, the continuity of the curvature function is not required.

where  $\alpha$  is the positive solution of the equation

$$a\alpha^2 + b\alpha + c = 0 \quad (8.52)$$

with

$$a = 16 - |\mathbf{t}_0 + \mathbf{t}_3|^2, \quad b = 12(\mathbf{p}_3 - \mathbf{p}_0)^T \cdot (\mathbf{t}_0 + \mathbf{t}_3), \quad c = -36|\mathbf{p}_3 - \mathbf{p}_0|^2$$

which is obtained from (8.50) (for more details, see [38]).

The computation of a trajectory composed by Bézier curve segments interpolating the points  $\mathbf{q}_k$ ,  $k = 0, \dots, n$ , consists of the following steps:

- If not given, compute the tangent vectors  $\mathbf{t}_k$  according to (8.48) or (8.49).
- For each pair  $\mathbf{q}_k, \mathbf{q}_{k+1}$ , the Bézier curve  $\mathbf{b}_k(u)$ ,  $u \in [0, 1]$ , can be determined by assuming

$$\begin{aligned} \mathbf{p}_{0,k} &= \mathbf{q}_k, & \mathbf{p}_{3,k} &= \mathbf{q}_{k+1} \\ \mathbf{t}_{0,k} &= \mathbf{t}_k, & \mathbf{t}_{3,k} &= \mathbf{t}_{k+1} \end{aligned}$$

and by solving (8.52), which provides the position of the internal control points  $\mathbf{p}_{1,k}, \mathbf{p}_{2,k}$  with (8.51).

This procedure leads to the definition of a trajectory composed by  $n$  Bézier segments (if  $n + 1$  are the via-points), each one defined in the interval  $[0, 1]$ , i.e.

$$\mathbf{p} = \mathbf{b}_k(u_k), \quad u_k \in [0, 1], \quad k = 0, \dots, n - 1 \quad (8.53)$$

where the variable  $u_k$  denotes the independent variable of the  $k$ -th Bézier. In order to describe the entire geometric path as a function of the unique variable  $u \in [u_{min}, u_{max}]$  it is necessary to properly “time shift” each tract. Being the “duration” of each Bézier curve unitary, by assuming

$$u_k = u - k, \quad u \in [k, k + 1], \quad k = 0, \dots, n - 1$$

it is possible to define all the segments composing the curve in terms of  $u$ . Therefore

$$\mathbf{p}(u) = \mathbf{b}_k(u - k), \quad u \in [0, n], \quad k = 0, \dots, n - 1.$$

Since the tangent vectors of two adjacent Bézier have the same direction but different magnitude, the curve  $\mathbf{p}(u)$  constructed in this way will be  $G^1$  continuous. In order to obtain a  $C^1$  continuous trajectory, it is necessary to reparameterize it by scaling (and shifting) each Bézier curve. For this purpose, with reference to (8.53), it is sufficient to assume for the  $k$ -th tract

$$u_k = \frac{\hat{u} - \hat{u}_{0,k}}{\lambda_k}, \quad \text{with } \lambda_k = 3|\mathbf{p}_{1,k} - \mathbf{p}_{0,k}| \text{ and } \hat{u} \in [\hat{u}_{0,k}, \hat{u}_{0,k+1}] \quad (8.54)$$

where the initial time instant<sup>21</sup> is

$$\hat{u}_{0,k} = \begin{cases} 0, & \text{if } k = 0 \\ \hat{u}_{0,k-1} + \lambda_k, & \text{if } k > 0. \end{cases}$$

In this manner, the tangent vectors at the endpoints of all segments composing the trajectory have unit length, and therefore the speed is continuous. This new parameterization  $\hat{p}(\hat{u})$  is a good approximation of a *uniform parameterization* (that is a parameterization characterized by a constant speed over the entire parameter range, see Chapter 9). Moreover, from the control points of Bézier curves, it is possible to construct a  $C^1$  continuous cubic spline interpolating the  $q_k$ , which is defined by the control points

$$\mathbf{P} = [q_0, p_{1,0}, p_{2,0}, p_{1,1}, p_{2,1}, \dots, p_{1,n-2}, p_{2,n-2}, p_{1,n-1}, p_{2,n-1}, q_n]$$

and by the knots

$$\mathbf{u} = [u_0, u_0, u_0, u_0, u_1, u_1, \dots, u_{n-1}, u_{n-1}, u_n, u_n, u_n, u_n]$$

computed in a recursive way as

$$u_k = u_{k+1} + 3|\mathbf{p}_{1,k} - \mathbf{p}_{0,k}|, \quad \text{for } k = 1, \dots, n$$

with  $u_0 = 0$ . It is worth noticing that, differently from the splines adopted in previous sections, the knots are not defined within the range  $[0, 1]$ , but a simple uniform scaling<sup>22</sup> ( $\hat{\mathbf{u}} = \mathbf{u}/u_n$ ) can lead to this type of parameterization.

**Example 8.18** The local interpolation of the points

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 & 4 & 5 & 6 \\ 0 & 2 & 3 & 3 & 2 & 0 \\ 0 & 1 & 0 & 0 & 2 & 2 \end{bmatrix}$$

by means of cubic Bézier curves is reported in Fig. 8.36. The tangent vectors (with unit length), computed by assuming a cord-length distribution of the “time instants”

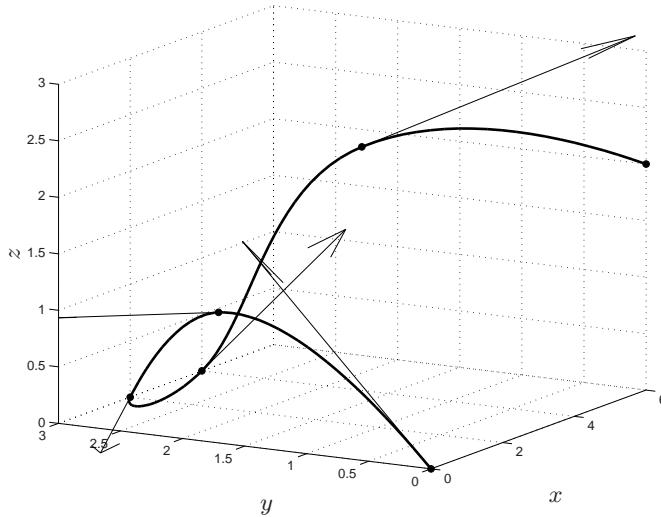
$$[\bar{u}_k] = [0, 0.22, 0.38, 0.56, 0.79, 1]$$

---

<sup>21</sup> Note that with this parameterization the duration of each Bézier segment is  $\hat{u}_{0,k+1} - \hat{u}_{0,k} = \lambda_k$ .

<sup>22</sup> The “scaling in time” of a B-spline curve can be performed by multiplying each knot span by a constant value  $\lambda$ , that is (in case  $u_0 = 0$ ) by assuming  $\hat{u}_k = \lambda u_k$ . This leads to

$$\hat{s}^{(1)}(\hat{u}_k) = \frac{1}{\lambda} s^{(1)}(u_k), \quad \hat{s}^{(2)}(\hat{u}_k) = \frac{1}{\lambda^2} s^{(2)}(u_k), \quad \hat{s}^{(3)}(\hat{u}_k) = \frac{1}{\lambda^3} s^{(3)}(u_k).$$



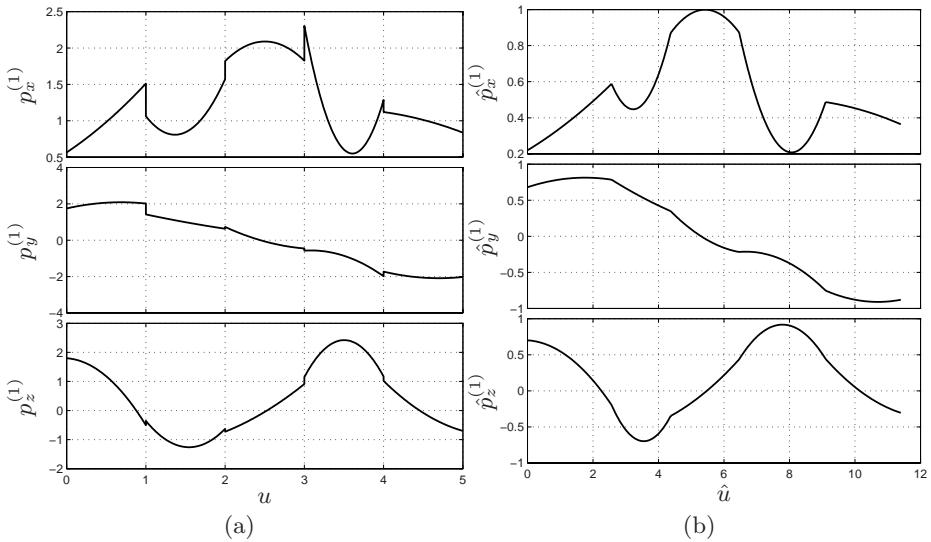
**Fig. 8.36.** Bézier curves of degree 3 interpolating a set of via-points.

are

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} 0.21 & 0.58 & 0.87 & 0.87 & 0.48 & 0.36 \\ 0.67 & 0.78 & 0.34 & -0.21 & -0.75 & -0.88 \\ 0.70 & -0.19 & -0.34 & 0.43 & 0.44 & -0.30 \end{bmatrix}$$

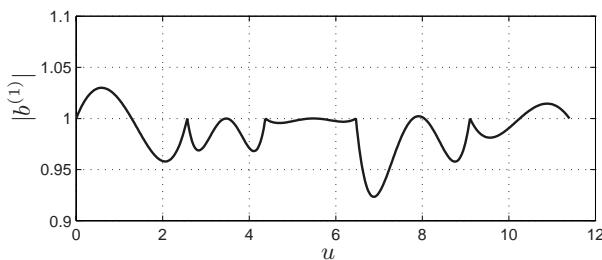
and the  $G^1$  continuous trajectory composed by Bézier segments is defined by the following control points

$$\begin{aligned} [\mathbf{p}_{0,k} \quad \mathbf{p}_{1,k} \quad \mathbf{p}_{2,k} \quad \mathbf{p}_{3,k}] = \\ \begin{bmatrix} 0 & 0.18 & 0.49 & 1 \\ 0 & 0.58 & 1.32 & 2 \\ 0 & 0.59 & 1.26 & 1 \\ 1 & 1.35 & 1.47 & 2 \\ 2 & 2.47 & 2.79 & 3 \\ 1 & 0.88 & 0.20 & 0 \\ 2 & 2.60 & 3.39 & 4 \\ 3 & 3.24 & 3.15 & 3 \\ 0 & -0.24 & -0.30 & 0 \\ 4 & 4.76 & 4.57 & 5 \\ 3 & 2.80 & 2.66 & 2 \\ 0 & 0.38 & 1.61 & 2 \\ 5 & 5.37 & 5.72 & 6 \\ 2 & 1.42 & 0.67 & 0 \\ 2 & 2.33 & 2.23 & 2 \end{bmatrix}. \end{aligned}$$



**Fig. 8.37.** Components of the tangent vector  $\mathbf{p}^{(1)}(u)$  of the trajectory composed by Bézier segments, without (a) and with (b) “scaling in time”.

The Cartesian components of the tangent vector for the entire trajectory are reported in Fig. 8.37(a); at the transition points the magnitude of this vector shows a discontinuity and therefore the speed is discontinuous. In Fig. 8.37(b) the tangent vector of the trajectory reparameterized according to (8.54) is reported. In this case, the magnitude of the tangent vector remains unchanged during the transition between two adjacent Bézier curves (and it is equal to one), and the speed is continuous over all the trajectory, which is therefore  $C^1$  continuous. As shown in Fig. 8.38 the speed is not only continuous but also rather constant (uniform parameterization).



**Fig. 8.38.** Magnitude of the speed of a trajectory composed by 3-rd degree Bézier curves properly reparameterized.

The same results can be obtained by considering the cubic B-spline defined by the control points

$$\mathbf{P} = \begin{bmatrix} 0.00 & 0.18 & 0.49 & 1.35 & 1.47 & 2.69 & 3.39 & 4.76 & 4.57 & 5.37 & 5.72 \\ 0.00 & 0.58 & 1.32 & 2.47 & 2.79 & 3.24 & 3.15 & 2.80 & 2.66 & 1.42 & 0.67 \\ 0.00 & 0.59 & 1.16 & 0.88 & 0.20 & -0.24 & -0.30 & 0.38 & 1.61 & 2.33 & 2.23 \end{bmatrix}^T$$

and by the knots

$$\mathbf{u} = [0, 0, 0, 0, 2.5, 2.5, 4.3, 4.3, 6.4, 6.4, 9.1, 9.1, 11.3, 11.3, 11.3, 11.3].$$

□

### 8.10.3 Quintic Bézier curves interpolation

In order to build a  $G^2$  (or  $C^2$ ) continuous trajectory, it is necessary to consider Bézier curves of degree  $m > 3$ . In general, 5-th degree Bézier curves, with six free parameters, may be adopted:

$$\mathbf{b}_k(u) = \sum_{j=0}^5 B_j^5(u) \mathbf{p}_j, \quad 0 \leq u \leq 1. \quad (8.55)$$

In this case, for the computation of the trajectory both tangent and curvature vectors  $\mathbf{t}_k, \mathbf{n}_k$  at each point  $\mathbf{q}_k$  are necessary. In the literature this kind of trajectory is often used to approximate the behavior of a cubic B-spline [91] with the purpose of obtaining a uniform parameterization (the advantages are highlighted in Chapter 9). Therefore, tangent and curvature vectors can be computed by interpolating the given points with a B-spline curve  $\mathbf{s}(u)$  and by considering its derivatives,  $\mathbf{s}^{(1)}(u), \mathbf{s}^{(2)}(u)$ , at  $\bar{u}_k$ . Then for each pair of via-points  $(\mathbf{q}_k, \mathbf{q}_{k+1})$  the control points which define the  $k$ -th Bézier curve  $\mathbf{b}_k(u)$  are computed, by imposing the conditions on initial and final tangent vectors (respectively  $\mathbf{t}_{0,k} = \mathbf{t}_k$  and  $\mathbf{t}_{5,k} = \mathbf{t}_{k+1}$ ) and curvature vectors ( $\mathbf{n}_{0,k} = \mathbf{n}_k$  and  $\mathbf{n}_{5,k} = \mathbf{n}_{k+1}$ ) assumed of unit length. Therefore,

$$\left\{ \begin{array}{lcl} \mathbf{b}_k(0) & = \mathbf{p}_{0,k} & = \mathbf{q}_k \\ \mathbf{b}_k(1) & = \mathbf{p}_{5,k} & = \mathbf{q}_{k+1} \\ \mathbf{b}_k^{(1)}(0) = 5(\mathbf{p}_{1,k} - \mathbf{p}_{0,k}) & & = \alpha_k \mathbf{t}_k \\ \mathbf{b}_k^{(1)}(1) = 5(\mathbf{p}_{5,k} - \mathbf{p}_{4,k}) & & = \alpha_k \mathbf{t}_{k+1} \\ \mathbf{b}_k^{(2)}(0) = 20(\mathbf{p}_{0,k} - 2\mathbf{p}_{1,k} + \mathbf{p}_{2,k}) & = \beta_k \mathbf{n}_k \\ \mathbf{b}_k^{(2)}(1) = 20(\mathbf{p}_{5,k} - 2\mathbf{p}_{4,k} + \mathbf{p}_{3,k}) & = \beta_k \mathbf{n}_{k+1} \end{array} \right.$$

where  $\alpha_k$  and  $\beta_k$  (respectively the magnitude of tangent and curvature vectors at the endpoints) are parameters which can be freely chosen. In particular,

like in the case of cubic Bézier curves, the value  $\alpha_k$  is computed by assuming that  $|\mathbf{b}_k^{(1)}|$  is equal at the endpoints and at the midpoint (this condition allows to maintain the magnitude of the first derivative almost constant along the entire curve), while it is assumed that  $\beta_k = \bar{\beta}\alpha_k^2$ . In this way, when the Bézier curves are reparameterized, by imposing that the first derivatives at the endpoints have unit magnitude, also the second derivatives of all the Bézier tracts at these points have equal magnitude ( $= \bar{\beta}$ ), and the overall trajectory is therefore  $C^2$  continuous. The conditions on  $\alpha_k$  and  $\beta_k$  lead to the quartic equation

$$a\alpha_k^4 + b\alpha_k^3 + c\alpha_k^2 + d\alpha_k + e = 0 \quad (8.56)$$

where

$$\begin{cases} a = \bar{\beta}^2 |\mathbf{n}_{k+1} - \mathbf{n}_k|^2 \\ b = -28\bar{\beta}(\mathbf{t}_k + \mathbf{t}_{k+1})^T \cdot (\mathbf{n}_{k+1} - \mathbf{n}_k) \\ c = 196|\mathbf{t}_k + \mathbf{t}_{k+1}|^2 + 120\bar{\beta}(\mathbf{q}_{k+1} - \mathbf{q}_k)^T \cdot (\mathbf{n}_{k+1} - \mathbf{n}_k) - 1024 \\ d = -1680(\mathbf{q}_{k+1} - \mathbf{q}_k)^T \cdot (\mathbf{t}_k + \mathbf{t}_{k+1}) \\ e = 3600|\mathbf{q}_{k+1} - \mathbf{q}_k|^2 \end{cases}$$

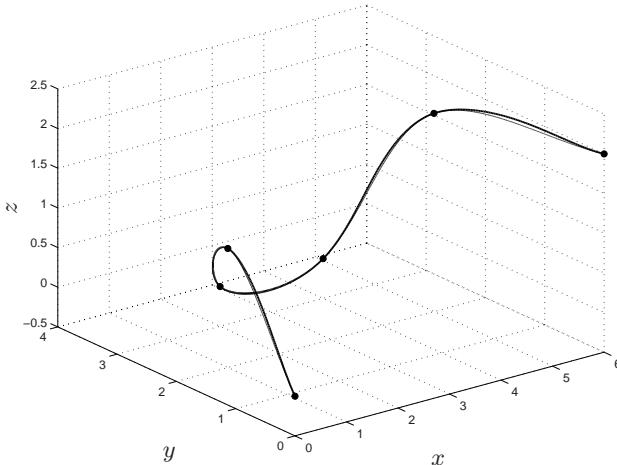
with the free parameter  $\bar{\beta}$  that can be chosen in different ways, e.g. with the purpose of minimizing the mean value of  $|\mathbf{p}^{(2)}(u)|$  over the entire trajectory  $\mathbf{p}(u)$ . The curve is completely determined by adopting the smallest positive solution of (8.56), which can be computed either in a closed form or in a numerical way. Finally, the control points of the  $k$ -th Bézier tract are

$$\begin{cases} \mathbf{p}_{0,k} = \mathbf{q}_k \\ \mathbf{p}_{1,k} = \mathbf{p}_{0,k} + \frac{\alpha_k}{5}\mathbf{t}_k \\ \mathbf{p}_{2,k} = 2\mathbf{p}_{1,k} - \mathbf{p}_{0,k} + \frac{\bar{\beta}\alpha_k^2}{20}\mathbf{n}_k \\ \mathbf{p}_{5,k} = \mathbf{q}_{k+1} \\ \mathbf{p}_{4,k} = \mathbf{p}_{5,k} - \frac{\alpha_k}{5}\mathbf{t}_{k+1} \\ \mathbf{p}_{3,k} = 2\mathbf{p}_{4,k} - \mathbf{p}_{5,k} + \frac{\bar{\beta}\alpha_k^2}{20}\mathbf{n}_{k+1} \end{cases} \quad (8.57)$$

and the trajectory can be evaluated as reported in Sec. B.3. In particular, each segment can be transformed in a standard polynomial form, i.e.

$$\mathbf{b}_k(u) = \mathbf{a}_{0,k} + \mathbf{a}_{1,k}u + \mathbf{a}_{2,k}u^2 + \mathbf{a}_{3,k}u^3 + \mathbf{a}_{4,k}u^4 + \mathbf{a}_{5,k}u^5, \quad 0 \leq u \leq 1$$

by assuming



**Fig. 8.39.** Interpolation of a set of points by means of a Bézier curve of degree 5; the interpolating cubic spline is also shown with a thin line, see also Fig. 8.36.

$$\left\{ \begin{array}{l} \mathbf{a}_{0,k} = \mathbf{p}_{0,k} \\ \mathbf{a}_{1,k} = -5\mathbf{p}_{0,k} + 5\mathbf{p}_{1,k} \\ \mathbf{a}_{2,k} = 10\mathbf{p}_{0,k} - 20\mathbf{p}_{1,k} + 10\mathbf{p}_{2,k} \\ \mathbf{a}_{3,k} = -10\mathbf{p}_{0,k} + 30\mathbf{p}_{1,k} - 30\mathbf{p}_{2,k} + 10\mathbf{p}_{3,k} \\ \mathbf{a}_{4,k} = 5\mathbf{p}_{0,k} - 20\mathbf{p}_{1,k} + 30\mathbf{p}_{2,k} - 20\mathbf{p}_{3,k} + 5\mathbf{p}_{4,k} \\ \mathbf{a}_{5,k} = -\mathbf{p}_{0,k} + 5\mathbf{p}_{1,k} - 10\mathbf{p}_{2,k} + 10\mathbf{p}_{3,k} - 5\mathbf{p}_{4,k} + \mathbf{p}_{5,k}. \end{array} \right.$$

**Example 8.19** The points of Example 8.18, interpolated by means of a quintic Bézier curve, are shown in Fig. 8.39. The curve is obtained by computing the B-spline which crosses all points and calculating tangent and curvature vectors. The B-spline trajectory  $\mathbf{s}(u)$  obtained as explained in Sec. 8.4 is defined by the knots (assumed with a cord-length distribution)

$$\mathbf{u} = [0, 0, 0, 0, 0.22, 0.38, 0.56, 0.79, 1, 1, 1]$$

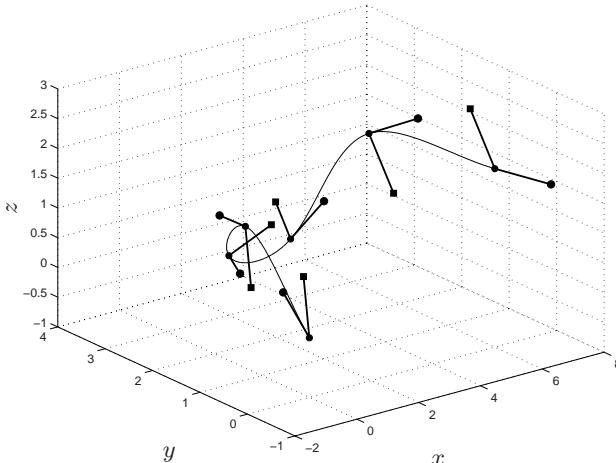
and by the control points

$$\mathbf{P} = \begin{bmatrix} 0 & 0.33 & 0.88 & 1.79 & 4.57 & 4.79 & 5.66 & 6 \\ 0 & 0.66 & 1.89 & 3.26 & 3.04 & 2.34 & 0.66 & 0 \\ 0 & 0.33 & 1.71 & -0.34 & -0.44 & 2.68 & 2.00 & 2 \end{bmatrix}.$$

By computing the derivatives of  $\mathbf{s}(u)$ , at

$$[\bar{u}_k] = [0, 0.22, 0.38, 0.56, 0.79, 1]$$

the following tangent and curvature vectors (normalized to unit length) are found



**Fig. 8.40.** Tangent and curvature vectors deduced by interpolating the given via-points with a B-spline curve.

$$\begin{aligned} \mathbf{t}_1 &= [0.40, 0.81, 0.40]^T, & \mathbf{n}_1 &= [-0.02, 0.10, 0.99]^T \\ \mathbf{t}_2 &= [0.47, 0.85, -0.19]^T, & \mathbf{n}_2 &= [0.02, -0.10, -0.99]^T \\ \mathbf{t}_3 &= [0.80, 0.28, -0.51]^T, & \mathbf{n}_3 &= [0.59, -0.50, 0.62]^T \\ \mathbf{t}_4 &= [0.77, -0.19, 0.59]^T, & \mathbf{n}_4 &= [-0.64, -0.10, 0.73]^T \\ \mathbf{t}_5 &= [0.37, -0.78, 0.48]^T, & \mathbf{n}_5 &= [0.22, -0.37, -0.90]^T \\ \mathbf{t}_6 &= [0.44, -0.89, 0]^T, & \mathbf{n}_6 &= [-0.22, 0.37, 0.90]^T. \end{aligned}$$

The vectors at each via-point are shown in Fig. 8.40. For each pairs of points  $(\mathbf{q}_k, \mathbf{q}_{k+1})$ , the value of  $\alpha_k$  is computed according to (8.56) and the Bézier tracts  $\mathbf{b}_k(u)$ ,  $u \in [0, 1]$  are completely determined. In order to approximate an arc-length parameterization (characterized by unit length tangent vectors) the trajectory must be reparameterized as in (8.54) with a little difference on  $\lambda_k$ , whose value is in this case  $\lambda_k = 5|\mathbf{p}_{1,k} - \mathbf{p}_{0,k}|$ . Therefore, for each segment, it is assumed that

$$u = \frac{\hat{u} - \hat{u}_k}{\lambda_k}, \quad \text{with } \hat{u} \in [\hat{u}_k, \hat{u}_{k+1}] \quad (8.58)$$

$$\hat{u}_k = \begin{cases} 0, & \text{if } k = 0 \\ \hat{u}_{k-1} + \lambda_k, & \text{if } k > 0 \end{cases}$$

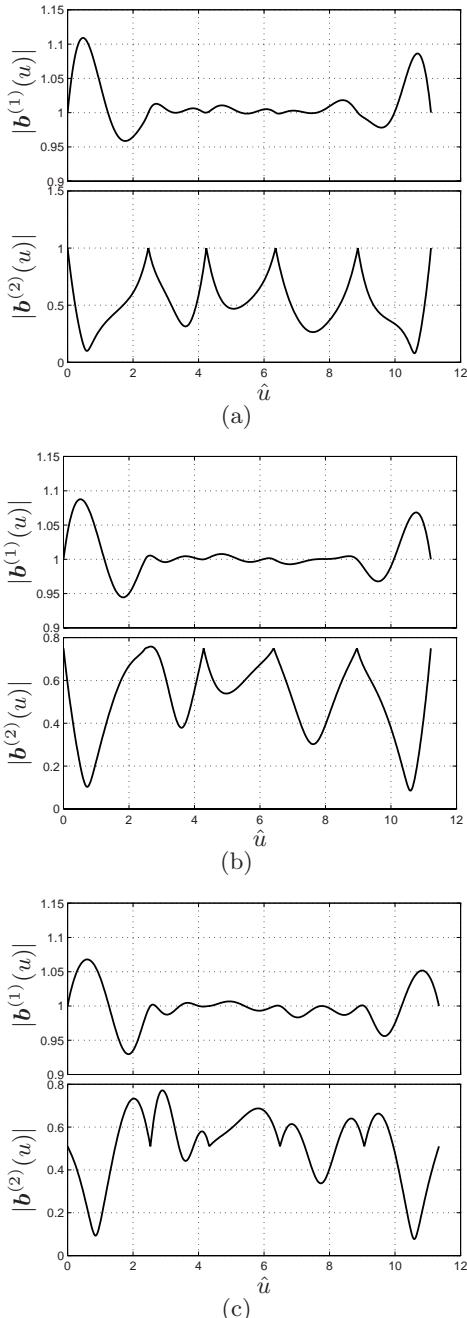
Note that, for the sake of notation simplicity, in (8.58) the subscripts  $k$  in  $\hat{u}$  and 0 in  $\hat{u}_k$  have been omitted. Therefore,  $\hat{u}$  and  $\hat{u}_k$  denotes respectively the independent variable and the initial value of  $\hat{u}$  for each Bézier segment.

The magnitude of the tangent and curvature vectors for different values of  $\bar{\beta}$  is reported in Fig. 8.41. Note that the speed has almost a constant unitary value, while the value of the acceleration at the joint between different Bézier tracts is obviously equal to  $\bar{\beta}$ . By considering  $\bar{\beta} = 0.51$ , the acceleration shows

less pronounced oscillation (with this particular value,  $\bar{\beta}$  is equal to the mean value of  $|\mathbf{b}^{(2)}(u)|$  over the entire trajectory), and also the deviations of  $|\mathbf{b}^{(1)}(u)|$  around 1 are smaller. With the value of  $\bar{\beta} = 0.51$ , the trajectory composed by quintic Bézier segments is defined by the following control points (computed with (8.57))

$$\begin{bmatrix} \mathbf{p}_{0,k} & \mathbf{p}_{1,k} & \mathbf{p}_{2,k} & \mathbf{p}_{3,k} & \mathbf{p}_{4,k} & \mathbf{p}_{5,k} \end{bmatrix} = \begin{bmatrix} 0 & 0.20 & 0.40 & 0.51 & 0.75 & 1 \\ 0 & 0.41 & 0.84 & 1.11 & 1.56 & 2 \\ 0 & 0.20 & 0.57 & 1.03 & 1.10 & 1 \\ 1 & 1.17 & 1.34 & 1.47 & 1.71 & 2 \\ 2 & 2.30 & 2.60 & 2.75 & 2.89 & 3 \\ 1 & 0.92 & 0.77 & 0.42 & 0.18 & 0 \\ 2 & 2.34 & 2.76 & 3.24 & 3.66 & 4 \\ 3 & 3.12 & 3.18 & 3.15 & 3.08 & 3 \\ 0 & -0.22 & -0.37 & -0.42 & -0.25 & 0 \\ 4 & 4.39 & 4.68 & 4.65 & 4.80 & 5 \\ 3 & 2.89 & 2.77 & 2.74 & 2.40 & 2 \\ 0 & 0.30 & 0.74 & 1.34 & 1.74 & 2 \\ 5 & 5.17 & 5.37 & 5.56 & 5.79 & 6 \\ 2 & 1.64 & 1.23 & 0.86 & 0.40 & 0 \\ 2 & 2.22 & 2.32 & 2.12 & 2.00 & 2 \end{bmatrix}.$$

□



**Fig. 8.41.** Velocity and acceleration of the trajectory composed by quintic Bézier curves for different values of  $\bar{\beta}$ : a)  $\bar{\beta} = 1$ , b)  $\bar{\beta} = 0.75$ , c)  $\bar{\beta} = 0.51$ .

## 8.11 Linear Interpolation with Polynomial Blends

The simplest approach for the definition of a trajectory which approximates a set of via-points with a prescribed tolerance  $\delta$  is based on piecewise linear segments with polynomial blends, which guarantee a smooth transition (i.e. continuity of position, speed and acceleration) between adjoining tracts [92, 93]. In particular the use of Bézier curves simplifies this application.

Given the via-points  $\mathbf{q}_k$ ,  $k = 0, \dots, n$ , the design of the trajectory is performed according to the following steps:

- For each point  $\mathbf{q}_k$  (with the only exception of the first and last ones, for which  $\mathbf{q}_0'' = \mathbf{q}_0$ ,  $\mathbf{q}_n' = \mathbf{q}_n$ ), two additional points  $\mathbf{q}'_k$  and  $\mathbf{q}''_k$  are obtained by finding the intersections between the lines  $\overline{\mathbf{q}_{k-1} \mathbf{q}_k}$ ,  $\overline{\mathbf{q}_k \mathbf{q}_{k+1}}$  and a ball of radius  $\delta$  (possibly different for each point) centered on  $\mathbf{q}_k$ , see Fig. 8.42.
- A straight line is used to join each pair  $(\mathbf{q}''_k, \mathbf{q}'_{k+1})$ .
- A Bézier curve of 4-th (or 5-th) degree is adopted to interpolate the pair  $(\mathbf{q}'_k, \mathbf{q}''_k)$ .

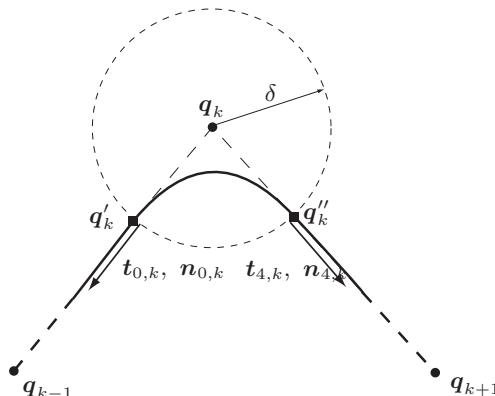
This procedure allows the construction of a  $G^2$  continuous trajectory, since the use Bézier curves guarantees that tangent vectors and curvature vectors of two contiguous segments (linear segment-Bézier, or Bézier-linear segment) at the transition have the same directions.

The overall trajectory is therefore represented by a sequence of linear segments  $\mathbf{l}_k(u)$  and the Bézier blends  $\mathbf{b}_k(u)$ , each one defined for  $u \in [0, 1]$ :

$$\mathbf{p} = \{\mathbf{l}_0(u), \mathbf{b}_1(u), \mathbf{l}_1(u), \dots, \mathbf{b}_k(u), \mathbf{l}_k(u), \mathbf{b}_{k+1}(u), \dots, \mathbf{l}_{n-2}(u), \mathbf{b}_{n-1}(u), \mathbf{l}_{n-1}(u)\}.$$

The generic  $i$ -th segment of the trajectory is denoted with  $\mathbf{p}_i(u)$ ,  $i = 0, \dots, 2n - 2$ .

While the computation of linear segments is straightforward once the end-points are known:



**Fig. 8.42.** Particular of a linear trajectory with a polynomial blend.

$$\mathbf{l}_k(u) = \mathbf{q}_k'' + (\mathbf{q}'_{k+1} - \mathbf{q}_k'')u, \quad 0 \leq u \leq 1, \quad k = 0, \dots, n-1,$$

the definition of Bézier curves of degree 4 requires also the knowledge of tangent and curvature vectors at the endpoints, in order to define the five control points  $\mathbf{p}_{j,k}$ :

$$\mathbf{b}_k(u) = \sum_{j=0}^4 B_j^4(u) \mathbf{p}_{j,k}, \quad 1 \leq u \leq 1, \quad k = 1, \dots, n-1.$$

Since a Bézier curve joins two linear tracts, the tangent vector  $\mathbf{t}$  and the curvature vector  $\mathbf{n}$  at a transition point have the same direction, and at the points  $\mathbf{q}'_k$  and  $\mathbf{q}''_k$ , they are oriented along  $\overrightarrow{\mathbf{q}'_k \mathbf{q}_k}$  and  $\overrightarrow{\mathbf{q}_k \mathbf{q}''_k}$  respectively. Therefore

$$\mathbf{t}_{0,k} = \mathbf{n}_{0,k} = \frac{\mathbf{q}_k - \mathbf{q}'_k}{\delta}, \quad \mathbf{t}_{4,k} = \mathbf{n}_{4,k} = \frac{\mathbf{q}''_k - \mathbf{q}_k}{\delta}.$$

Note that because of the definition of  $\mathbf{q}'_k$  and  $\mathbf{q}''_k$ , tangent and curvature vectors have both unit length. To define the  $k$ -th Bézier curve it is necessary to impose the interpolation of the two endpoints

$$\mathbf{p}_{0,k} = \mathbf{q}'_k, \quad \mathbf{p}_{4,k} = \mathbf{q}''_k \quad (8.59)$$

and that the first and second derivatives at the endpoints are respectively oriented along the tangent and curvature vectors

$$\begin{cases} \mathbf{b}_k^{(1)}(0) = 4(\mathbf{p}_{1,k} - \mathbf{p}_{0,k}) = \alpha_k \mathbf{t}_{0,k} \\ \mathbf{b}_k^{(1)}(1) = 4(\mathbf{p}_{4,k} - \mathbf{p}_{3,k}) = \alpha_k \mathbf{t}_{4,k} \\ \mathbf{b}_k^{(2)}(0) = 12(\mathbf{p}_{0,k} - 2\mathbf{p}_{1,k} + \mathbf{p}_{2,k}) = \beta_{0,k} \mathbf{n}_{0,k} \\ \mathbf{b}_k^{(2)}(1) = 12(\mathbf{p}_{4,k} - 2\mathbf{p}_{3,k} + \mathbf{p}_{2,k}) = \beta_{4,k} \mathbf{n}_{4,k} \end{cases} \quad (8.60)$$

Like in the previous section, the value of  $\alpha_k$  is determined by assuming that the velocities are equal at the endpoints and in the middle point

$$|\mathbf{b}_k^{(1)}(0)| = |\mathbf{b}_k^{(1)}(1)| = |\mathbf{b}_k^{(1)}(1/2)| = \alpha_k. \quad (8.61)$$

In the case of a 4-th degree Bézier, the expression of the tangent vector for  $u = 1/2$  is

$$\mathbf{b}_k^{(1)}(1/2) = \frac{1}{2}(\mathbf{p}_{4,k} + 2\mathbf{p}_{3,k} - 2\mathbf{p}_{1,k} - \mathbf{p}_{0,k}). \quad (8.62)$$

Note that the point  $\mathbf{p}_{2,k}$  can be freely changed without modifying the tangent directions of the curve at the initial and final points and also the tangent direction in the middle point (i.e. for  $u = 1/2$ ). From (8.61), (8.62) and (8.60) it follows

$$a\alpha_k^2 + b\alpha_k + c = 0 \quad (8.63)$$

with

$$\begin{cases} a = 4 - \frac{1}{4} |\mathbf{t}_{4,k} + \mathbf{t}_{0,k}|^2 \\ b = 3 (\mathbf{p}_{4,k} - \mathbf{p}_{0,k})^T \cdot (\mathbf{t}_{0,k} + \mathbf{t}_{4,k}) \\ c = -9 |\mathbf{p}_{4,k} - \mathbf{p}_{0,k}|^2 \end{cases}$$

which provides the value of  $\alpha_k$  (the largest solution must be considered). In this way from the first two equations of (8.60) the control points  $\mathbf{p}_{1,k}$  and  $\mathbf{p}_{3,k}$  are obtained.

The remaining control point  $\mathbf{p}_{2,k}$  can be computed by considering the last two equations of (8.60) which can be rewritten as

$$\mathbf{b}_k^{(2)}(0) = 12(\mathbf{p}_{0,k} - \mathbf{p}_{1,k}) - 12(\mathbf{p}_{1,k} - \mathbf{p}_{2,k}) = \beta_{0,k} \mathbf{n}_{0,k} \quad (8.64a)$$

$$\mathbf{b}_k^{(2)}(1) = 12(\mathbf{p}_{4,k} - \mathbf{p}_{3,k}) - 12(\mathbf{p}_{3,k} - \mathbf{p}_{2,k}) = \beta_{4,k} \mathbf{n}_{4,k}. \quad (8.64b)$$

By considering that  $\mathbf{n}_{0,k} = \mathbf{t}_{0,k}$  and  $\mathbf{n}_{4,k} = \mathbf{t}_{4,k}$ , (8.64a)-(8.64b) yield

$$\mathbf{p}_{2,k} = \mathbf{p}_{0,k} + \frac{1}{12} (\beta_{0,k} + 6\alpha_k) \mathbf{t}_{0,k} \quad (8.65a)$$

$$\mathbf{p}_{2,k} = \mathbf{p}_{4,k} + \frac{1}{12} (\beta_{4,k} - 6\alpha_k) \mathbf{t}_{4,k}. \quad (8.65b)$$

Both conditions (8.65a) and (8.65b) hold if  $\mathbf{p}_{2,k}$  is located in the point where the line with direction  $\mathbf{t}_{0,k}$  and passing through  $\mathbf{p}_{0,k}$  intersects the line crossing  $\mathbf{p}_{4,k}$  and oriented as  $\mathbf{t}_{4,k}$ , that is

$$\mathbf{p}_{2,k} = \mathbf{q}_k.$$

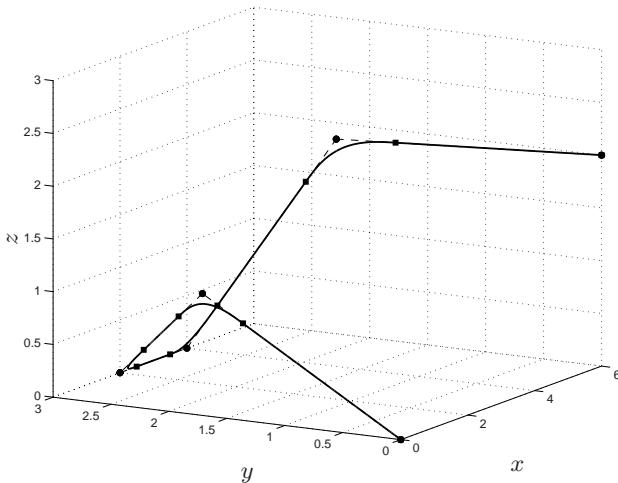
Therefore, the  $k$ -th Bézier curve is completely defined by

$$\begin{cases} \mathbf{p}_{0,k} = \mathbf{q}'_k \\ \mathbf{p}_{1,k} = \mathbf{q}'_k + \frac{1}{4} \alpha_k \mathbf{t}_{0,k} \\ \mathbf{p}_{2,k} = \mathbf{q}_k \\ \mathbf{p}_{3,k} = \mathbf{q}''_k - \frac{1}{4} \alpha_k \mathbf{t}_{4,k} \\ \mathbf{p}_{4,k} = \mathbf{q}''_k \end{cases}$$

where  $\alpha_k$  is computed using (8.63). The trajectory can be evaluated by using the definition (8.11). Otherwise, the transformation in a standard polynomial form, i.e.

$$\mathbf{b}_k(u) = \mathbf{a}_{0,k} + \mathbf{a}_{1,k} u + \mathbf{a}_{2,k} u^2 + \mathbf{a}_{3,k} u^3 + \mathbf{a}_{4,k} u^4, \quad 0 \leq u \leq 1,$$

is straightforward by assuming



**Fig. 8.43.** Linear interpolation with polynomial blends obtained as Bézier curves of degree 4 (see also Fig. 8.36 and Fig. 8.39).

$$\begin{cases} \mathbf{a}_{0,k} = \mathbf{p}_{0,k} \\ \mathbf{a}_{1,k} = -4\mathbf{p}_{0,k} + 4\mathbf{p}_{1,k} \\ \mathbf{a}_{2,k} = 6\mathbf{p}_{0,k} - 12\mathbf{p}_{1,k} + 6\mathbf{p}_{2,k} \\ \mathbf{a}_{3,k} = -4\mathbf{p}_{0,k} + 12\mathbf{p}_{1,k} - 12\mathbf{p}_{2,k} + 4\mathbf{p}_{3,k} \\ \mathbf{a}_{4,k} = 5\mathbf{p}_{0,k} - 4\mathbf{p}_{1,k} + 6\mathbf{p}_{2,k} - 4\mathbf{p}_{3,k} + \mathbf{p}_{4,k}. \end{cases}$$

**Example 8.20** The interpolation of the same points of Example 8.18 and Example 8.19 by means of linear segments with polynomial blends is shown in Fig. 8.43. By considering  $\delta = 0.5$ , the endpoints for linear segments and for Bézier curves are obtained from the points

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 & 4 & 5 & 6 \\ 0 & 2 & 3 & 3 & 2 & 0 \\ 0 & 1 & 0 & 0 & 2 & 2 \end{bmatrix}.$$

The five linear tracts join the following points

$$\begin{aligned} \mathbf{q}_0'' &= [0 \quad 0 \quad 0]^T, & \mathbf{q}_1' &= [0.79 \quad 1.59 \quad 0.79]^T \\ \mathbf{q}_1'' &= [1.28 \quad 2.28 \quad 0.71]^T, & \mathbf{q}_2' &= [1.71 \quad 2.71 \quad 0.28]^T \\ \mathbf{q}_2'' &= [2.5 \quad 3 \quad 0]^T, & \mathbf{q}_3' &= [3.5 \quad 3 \quad 0]^T \\ \mathbf{q}_3'' &= [4.20 \quad 2.79 \quad 0.40]^T, & \mathbf{q}_4' &= [4.79 \quad 2.20 \quad 1.59]^T \\ \mathbf{q}_4'' &= [5.22 \quad 1.55 \quad 2]^T, & \mathbf{q}_5' &= [6 \quad 0 \quad 2]^T \end{aligned}$$

while the Bézier curves have the following endpoints

$$\begin{aligned} \mathbf{q}'_1 &= [0.79 \ 1.59 \ 0.79]^T, & \mathbf{q}''_1 &= [1.28 \ 2.28 \ 0.71]^T \\ \mathbf{q}'_2 &= [1.71 \ 2.71 \ 0.28]^T, & \mathbf{q}''_2 &= [2.5 \ 3 \ 0]^T \\ \mathbf{q}'_3 &= [3.5 \ 3 \ 0]^T, & \mathbf{q}''_3 &= [4.20 \ 2.79 \ 0.40]^T \\ \mathbf{q}'_4 &= [4.79 \ 2.20 \ 1.59]^T, & \mathbf{q}''_4 &= [5.22 \ 1.55 \ 2]^T \end{aligned}$$

with tangent/curvature vectors

$$\begin{aligned} \mathbf{t}_{0,1} = \mathbf{n}_{0,1} &= [0.40 \ 0.81 \ 0.40]^T, & \mathbf{t}_{4,1} = \mathbf{n}_{4,1} &= [0.57 \ 0.57 \ -0.57]^T \\ \mathbf{t}_{0,2} = \mathbf{n}_{0,2} &= [0.57 \ 0.57 \ -0.57]^T, & \mathbf{t}_{4,2} = \mathbf{n}_{4,2} &= [1 \ 0 \ 0]^T \\ \mathbf{t}_{0,3} = \mathbf{n}_{0,3} &= [1 \ 0 \ 0]^T, & \mathbf{t}_{4,3} = \mathbf{n}_{4,3} &= [0.40 \ -0.40 \ 0.81]^T \\ \mathbf{t}_{0,4} = \mathbf{n}_{0,4} &= [0.40 \ -0.40 \ 0.81]^T, & \mathbf{t}_{4,4} = \mathbf{n}_{4,4} &= [0.44 \ -0.89 \ 0]^T. \end{aligned}$$

The trajectory composed by linear segments and Bézier curves properly reparametrized<sup>23</sup> is  $C^1$  continuous, as shown in Fig. 8.44(b), but the second derivative of the trajectory is discontinuous at the transition points (therefore, the trajectory is only  $G^2$  continuous). Since the curvature vectors of a Bézier tract at its endpoints are oriented along the directions of adjacent lines, it is possible to guarantee the continuity of the acceleration by assuming a proper motion law along the geometric path so defined (in particular the acceleration at the beginning and at the end of each linear tract must be not null).

□

It is possible to define a  $C^2$  continuous trajectory composed by linear segments with polynomial blends by assuming 5-th degree (or quintic) Bézier curves. As in Sec. 8.10.3, the six control points of each Bézier curve are obtained by imposing initial and final points, tangent vectors and curvature vectors:

$$\left\{ \begin{array}{ll} \mathbf{b}_k(0) = \mathbf{p}_{0,k} & = \mathbf{q}'_k \\ \mathbf{b}_k(1) = \mathbf{p}_{5,k} & = \mathbf{q}''_k \\ \mathbf{b}_k^{(1)}(0) = 5(\mathbf{p}_{1,k} - \mathbf{p}_{0,k}) & = \alpha_k \mathbf{t}_{0,k} \\ \mathbf{b}_k^{(1)}(1) = 5(\mathbf{p}_{5,k} - \mathbf{p}_{4,k}) & = \alpha_k \mathbf{t}_{5,k} \\ \mathbf{b}_k^{(2)}(0) = 20(\mathbf{p}_{0,k} - 2\mathbf{p}_{1,k} + \mathbf{p}_{2,k}) = \beta_{0,k} \mathbf{n}_{0,k} & \\ \mathbf{b}_k^{(2)}(1) = 20(\mathbf{p}_{5,k} - 2\mathbf{p}_{4,k} + \mathbf{p}_{3,k}) = \beta_{5,k} \mathbf{n}_{5,k}. & \end{array} \right.$$

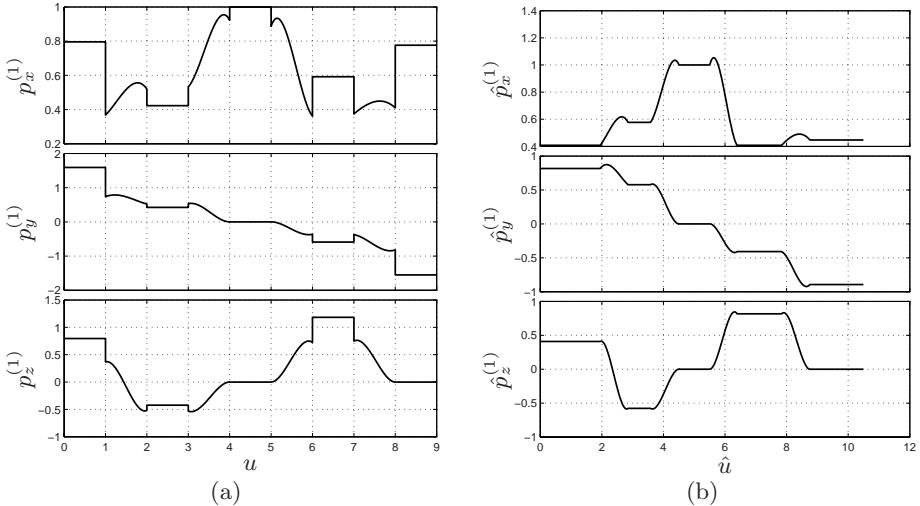
---

<sup>23</sup> Each tract  $\mathbf{p}_i(u)$  of the trajectory is reparameterized according to

$$u = \frac{\hat{u} - \hat{u}_i}{\lambda_i}, \quad \text{with } \hat{u} \in [\hat{u}_i, \hat{u}_{i+1}]$$

where  $\lambda_i = 4|\mathbf{p}_{1,i} - \mathbf{p}_{0,i}|$  for Bézier curves and  $\lambda_i = |\mathbf{p}_{1,i} - \mathbf{p}_{0,i}|$  for the linear tracts joining  $\mathbf{p}_{0,i}$  with  $\mathbf{p}_{1,i}$  and

$$\hat{u}_i = \begin{cases} 0, & \text{if } i = 0 \\ \hat{u}_{i-1} + \lambda_i, & \text{if } i > 0. \end{cases}$$



**Fig. 8.44.** Components of the tangent vectors  $\mathbf{p}^{(1)}(u)$  and  $\hat{\mathbf{p}}^{(1)}(\hat{u})$  of the trajectory composed by linear and 4-th degree Bézier segments, without (a) and with (b) “scaling in time”.

In particular, it is assumed that the second derivative of  $\mathbf{b}_k^{(2)}(u)$  for  $u = 0$  and  $u = 1$  has zero magnitude (therefore  $\beta_{0,k} = 0$  and  $\beta_{5,k} = 0$ ), while the value  $\alpha_k$  is computed by assuming that  $|\mathbf{b}_k^{(1)}(u)|$  is equal at the endpoints and at the midpoint. This last condition leads to the equation

$$a\alpha_k^2 + b\alpha_k + c = 0 \quad (8.66)$$

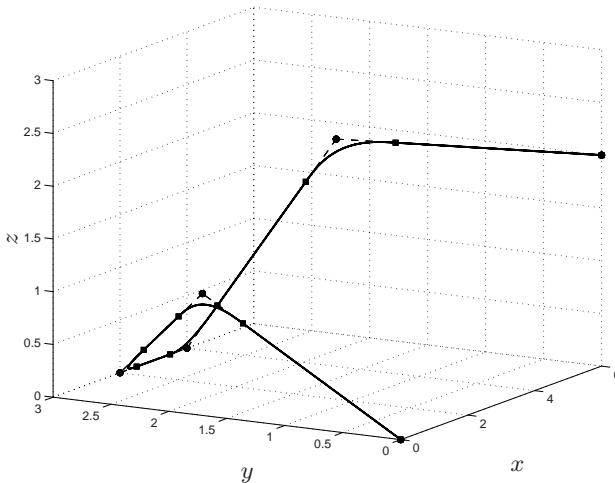
where

$$\begin{cases} a = 256 - 49 |\mathbf{t}_{0,k} + \mathbf{t}_{5,k}|^2 \\ b = 420 (\mathbf{p}_{5,k} - \mathbf{p}_{0,k})^T (\mathbf{t}_{0,k} + \mathbf{t}_{5,k}) \\ c = -900 |\mathbf{p}_{5,k} - \mathbf{p}_{0,k}|^2 = 0 \end{cases}$$

which provides<sup>24</sup> the value of  $\alpha_k$  (the largest of the two possible solutions). Finally, the control points which define the  $k$ -th Bézier segment are

<sup>24</sup> Note that all the terms in (8.66) are known since

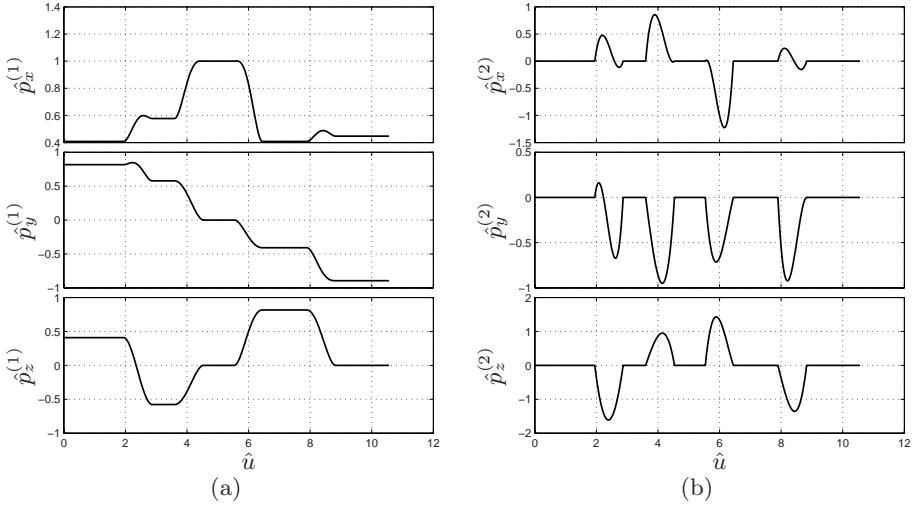
$$\begin{aligned} \mathbf{p}_{0,k} &= \mathbf{q}'_k, & \mathbf{p}_{5,k} &= \mathbf{q}''_k, \\ \mathbf{t}_{0,k} &= \mathbf{n}_{0,k} = \frac{\mathbf{q}_k - \mathbf{q}'_k}{\delta}, & \mathbf{t}_{5,k} &= \mathbf{n}_{5,k} = \frac{\mathbf{q}''_k - \mathbf{q}_k}{\delta}. \end{aligned}$$



**Fig. 8.45.** Linear interpolation with polynomial blends obtained as Bézier curves of degree 5.

$$\left\{ \begin{array}{l} \mathbf{p}_{0,k} = \mathbf{q}'_k \\ \mathbf{p}_{1,k} = \mathbf{p}_{0,k} + \frac{\alpha_k}{5} t_{0,k} \\ \mathbf{p}_{2,k} = 2\mathbf{p}_{1,k} - \mathbf{p}_{0,k} \\ \mathbf{p}_{5,k} = \mathbf{q}''_k \\ \mathbf{p}_{4,k} = \mathbf{p}_{5,k} - \frac{\alpha_k}{5} t_{5,k} \\ \mathbf{p}_{3,k} = 2\mathbf{p}_{4,k} - \mathbf{p}_{5,k}. \end{array} \right.$$

**Example 8.21** The interpolation of the same points of the previous examples by means of linear segments and 5-th degree Bézier tracts is shown in Fig. 8.45. Once the endpoints and the tangent vectors have been defined (the same as in the previous examples), the computation of Bézier curves is straightforward by means of (8.57) while the procedure for the determination of linear tracts does not change. The first and the second derivatives of the trajectory, properly



**Fig. 8.46.** Components of the velocity  $\hat{\mathbf{p}}^{(1)}(\hat{u})$  (a) and of the acceleration  $\hat{\mathbf{p}}^{(2)}(\hat{u})$  (b) of the trajectory composed by linear and 5-th degree Bézier segments.

reparameterized<sup>25</sup>, are shown in Fig. 8.46. In this case both the speed and the acceleration are continuous (in particular, the acceleration at the beginning and at the end of each Bézier segment is null), and the trajectory is  $C^2$  continuous, see Fig. 8.46. Moreover, the parameterization obtained in this way is an optimal approximation of an arc-length parametrization (with unit-length tangent vector), as shown in Fig. 8.47.

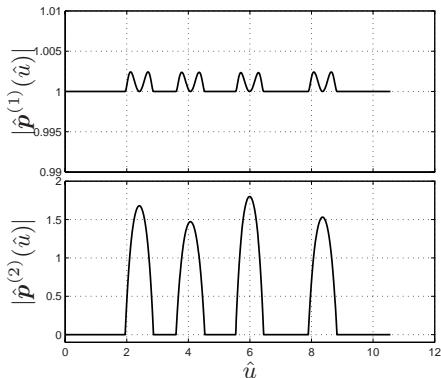
□

<sup>25</sup> Each Bézier tract of the trajectory is reparameterized according to

$$u = \frac{\hat{u} - \hat{u}_k}{\lambda_k}, \quad \text{with } \hat{u} \in [\hat{u}_k, \hat{u}_{k+1}]$$

where  $\lambda_k = 5|\mathbf{p}_{1,k} - \mathbf{p}_{0,k}|$  and

$$\hat{u}_k = \begin{cases} 0, & \text{if } k = 0 \\ \hat{u}_{k-1} + \lambda_k, & \text{if } k > 0. \end{cases}$$



**Fig. 8.47.** Magnitude of the tangent and curvature vectors of the trajectory composed by straight line segments and 5-th degree Bézier curves.

---

## From Geometric Paths to Trajectories

In this chapter the problem of composing the geometric path with the motion law is considered. The goal is to define parametric functions of time so that given constraints on velocities, accelerations, . . . , are satisfied. A particular case of interest is the “constant velocity” motion, used in several industrial tasks.

---

### 9.1 Introduction

Given a geometric path represented by a parametric curve

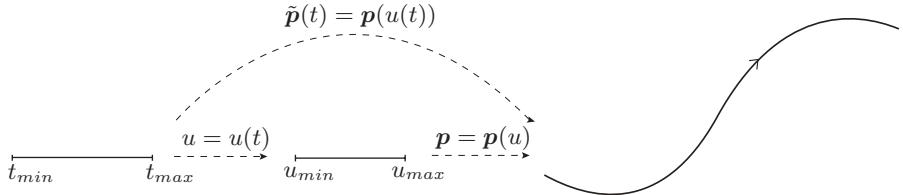
$$\mathbf{p} = \mathbf{p}(u)$$

the trajectory is completely defined only when the motion law

$$u = u(t)$$

is provided. In many cases, the function  $u(t)$  is a simple proportional law, e.g.  $u = \lambda t$ , but are common situations in which a particular motion law is adopted to guarantee that the trajectory is compliant with velocity and acceleration constraints. It is worth noticing that in this case the motion law is nothing but a reparameterization of the curve, which modifies the velocity and acceleration vectors, see Fig. 9.1. As a matter of fact, by applying the chain rule to compute the derivatives of the trajectory  $\tilde{\mathbf{p}}(t) = (\mathbf{p} \circ u)(t)$ :

$$\begin{aligned}\dot{\tilde{\mathbf{p}}}(t) &= \frac{d\mathbf{p}}{du} \dot{u}(t) \\ \ddot{\tilde{\mathbf{p}}}(t) &= \frac{d\mathbf{p}}{du} \ddot{u}(t) + \frac{d^2\mathbf{p}}{du^2} \dot{u}^2(t) \\ &\vdots\end{aligned}\tag{9.1}$$



**Fig. 9.1.** Composition of a generic 3D path  $\mathbf{p}(u)$  and of a motion law  $u(t)$ .

it is evident that the velocity vector of  $\tilde{\mathbf{p}}(t)$  is equal to the first derivative of the parametric curve  $\mathbf{p}(u)$  modulated by the velocity of the motion law, while the acceleration depends on both the acceleration and the (square) speed of  $u(t)$ . In particular by comparing (9.1) with (8.2.2), it comes out that the velocity is always oriented along the direction tangent to the curve, while the acceleration is composed by two components, oriented along the directions tangent (*tangential acceleration*) and normal (*centripetal acceleration*) to the curve.

## 9.2 Constant Scaling

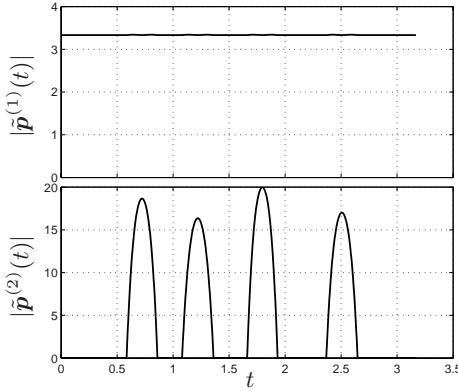
When a proportional law is used to describe the relation between the time and the variable  $u$  ( $u = \lambda t$ ), the  $k$ -th derivative of the parametric curve is simply scaled by a factor  $\lambda^k$ . Therefore

$$\begin{aligned}\tilde{\mathbf{p}}^{(1)}(t) &= \frac{d\mathbf{p}}{du} \lambda \\ \tilde{\mathbf{p}}^{(2)}(t) &= \frac{d^2\mathbf{p}}{du^2} \lambda^2 \\ \tilde{\mathbf{p}}^{(3)}(t) &= \frac{d^3\mathbf{p}}{du^3} \lambda^3 \\ &\vdots\end{aligned}$$

These relations are used when the trajectory must satisfy particular constraints on velocity ( $v_{max}$ ), acceleration ( $a_{max}$ ), jerk ( $j_{max}$ ), etc.. In this case, it is sufficient to assume

$$\lambda = \min \left\{ \frac{v_{max}}{|\mathbf{p}^{(1)}(u)|_{max}}, \sqrt{\frac{a_{max}}{|\mathbf{p}^{(2)}(u)|_{max}}}, \sqrt[3]{\frac{j_{max}}{|\mathbf{p}^{(3)}(u)|_{max}}}, \dots \right\} \quad (9.2)$$

to assure that the trajectory is compliant with all the constraints. On the other hand, it is worth observing that a constant scaling cannot guarantee a smooth starting/ending (with initial and final velocities and accelerations equal to zero). In this case a continuous motion law, like those introduced in previous chapters, should be taken into account.



**Fig. 9.2.** Magnitude of velocity and acceleration of the trajectory composed by straight line segments and 5-th degree Bézier curves.

**Example 9.1** The geometric path obtained in Example 8.21 is considered. In particular, the parametric trajectory composed by linear segments and 5-th degree Bézier curves (after the reparameterization which guarantees the continuity of the overall curve) is assumed. The goal is to make the trajectory as fast as possible with the constraints

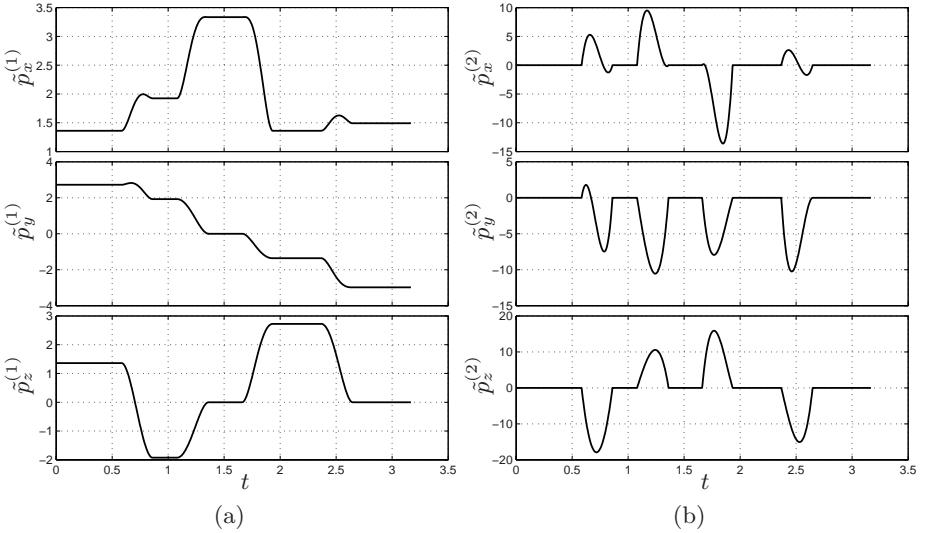
$$v_{max} = 5, \quad a_{max} = 20.$$

By applying (9.2) where  $|\mathbf{p}^{(1)}(u)|_{max} = 1.02$  and  $|\mathbf{p}^{(1)}(u)|_{max} = 1.8$  (see Fig. 8.47) the value  $\lambda = 3.33$  is obtained. It is therefore sufficient for each segment of the trajectory to consider

$$\hat{u} = \lambda t, \quad t \in \left[ \frac{\hat{u}_k}{\lambda}, \frac{\hat{u}_{k+1}}{\lambda} \right]$$

being  $\hat{u}_k$  and  $\hat{u}_{k+1}$  the limit values for the independent variable  $\hat{u}$  of the  $k$ -th tract. Note that, differently from the case of B-spline or Nurbs curves obtained by globally interpolating/approximating a set of data points (and therefore characterized by only one variable  $u \in [u_{min}, u_{max}]$ ), in this case each segment depends on the corresponding variable  $\hat{u}$ , which can be reparameterized separately. In this case, this opportunity is not exploited (by considering only a constant scaling  $\lambda$  for the overall trajectory), and the profiles of the magnitude of the velocity and acceleration are those reported in Fig. 9.2 (while the components are shown in Fig. 9.3). Note that the velocity, although rather constant, does not reach the maximum allowed value since the value of  $\lambda$  is determined by the maximum acceleration. Moreover, the velocity is discontinuous at the start and end points of the trajectory.

□



**Fig. 9.3.** Components of the velocity  $\tilde{\mathbf{p}}^{(1)}(t)$  (a) and of the acceleration  $\tilde{\mathbf{p}}^{(2)}(t)$  (b) of the trajectory composed by linear and 5-th degree Bézier segments, after a constant scaling.

### 9.3 Generic Motion Law

When a generic motion law is assumed to describe the relation between  $t$  and  $u$ , the derivatives of the parametric curve are modified according to (9.1). Therefore, it is not simple to find a motion law which satisfies prescribed values of acceleration or jerk, since the derivatives of  $u(t)$  are mixed in the expression of  $\tilde{\mathbf{p}}^{(2)}(t)$ ,  $\tilde{\mathbf{p}}^{(3)}(t)$ , etc. A special case is represented by linear trajectories, i.e.

$$\mathbf{p}(u) = \mathbf{p}_0 + (\mathbf{p}_1 - \mathbf{p}_0) u, \quad \text{with } 0 \leq u \leq 1$$

characterized by  $\mathbf{p}^{(1)}(u) = \mathbf{p}_1 - \mathbf{p}_0 = \text{const.}$ ,  $\mathbf{p}^{(2)}(u) = \mathbf{p}^{(3)}(u) = \dots = \mathbf{p}^{(m)}(u) = \mathbf{0}$ . As a consequence

$$\tilde{\mathbf{p}}^{(1)}(t) = (\mathbf{p}_1 - \mathbf{p}_0) u^{(1)}(t) \tag{9.3a}$$

$$\tilde{\mathbf{p}}^{(2)}(t) = (\mathbf{p}_1 - \mathbf{p}_0) u^{(3)}(t) \tag{9.3b}$$

$$\tilde{\mathbf{p}}^{(3)}(t) = (\mathbf{p}_1 - \mathbf{p}_0) u^{(3)}(t) \tag{9.3c}$$

and the constraints on  $|\tilde{\mathbf{p}}^{(i)}(t)|$  can be easily translated to constraints on  $u^{(i)}(t)$  by inverting (9.3a), (9.3b), etc. Therefore,

$$\begin{aligned}|u^{(1)}(t)| &\leq \frac{v_{max}}{|\mathbf{p}_1 - \mathbf{p}_0|} \\ |u^{(2)}(t)| &\leq \frac{a_{max}}{|\mathbf{p}_1 - \mathbf{p}_0|} \\ |u^{(3)}(t)| &\leq \frac{j_{max}}{|\mathbf{p}_1 - \mathbf{p}_0|}.\end{aligned}$$

Moreover, also the constraints on a single axis (e.g. the  $x$  axis) can be easily translated to limits on  $u^{(i)}(t)$ , i.e.

$$\begin{aligned}|u^{(1)}(t)| &\leq \frac{v_{xmax}}{|(\mathbf{p}_1 - \mathbf{p}_0)_x|} \\ |u^{(2)}(t)| &\leq \frac{a_{xmax}}{|(\mathbf{p}_1 - \mathbf{p}_0)_x|} \\ |u^{(3)}(t)| &\leq \frac{j_{xmax}}{|(\mathbf{p}_1 - \mathbf{p}_0)_x|}\end{aligned}$$

where the subscript  $x$  denotes the component of a vector along the  $x$  axis.

**Example 9.2** The geometric curve of Example 8.21 is considered again. Differently from the uniform scaling of the trajectory, performed in Example 9.1, the segments composing the curve (linear segments and 5-th degree Bézier curves) are now considered separately. For this reason, it is necessary to address the problem of maintaining the continuity of the trajectory (and its derivatives) at the joints. Moreover, each segment is considered before the reparameterization which leads to unit tangent vectors at the endpoints of each tract (therefore for each segment  $u \in [0, 1]$ ). The goal is to make the trajectory as fast as possible with the constraints

$$v_{max} = 5, \quad a_{max} = 20.$$

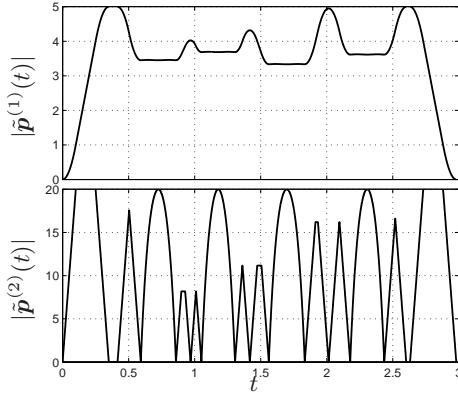
In particular, a constant scaling  $u = \lambda_k t$ , is used for Bézier segments while a double S trajectory is adopted for  $u(t)$  in each linear tract. The reparameterization is based on two subsequent steps:

1. The value of  $\lambda_k$  for each Bézier segments is found by means of (9.2).
2. A double S trajectory  $u(t)$ , is computed for each linear tract with the following constraints<sup>1</sup>:

$$\begin{array}{lll} u_0 &= 0, & u_1 &= 1 \\ \dot{u}_0 &= \frac{|\mathbf{t}_{k-1}| \lambda_{k-1}}{l_k}, & \dot{u}_1 &= \frac{|\mathbf{t}_{k+1}| \lambda_{k+1}}{l_k} \\ \ddot{u}_0 &= 0, & \ddot{u}_1 &= 0 \\ \dot{u}_{max} &= \frac{v_{max}}{l_k}, & \ddot{u}_{max} &= \frac{a_{max}}{l_k} \end{array}$$

---

<sup>1</sup> For the generation of double S motion law for straight line segments the further condition on the jerk  $j_{max} = 200$  is considered; this leads to  $\ddot{u}_{max} = \frac{j_{max}}{l_k}$ .

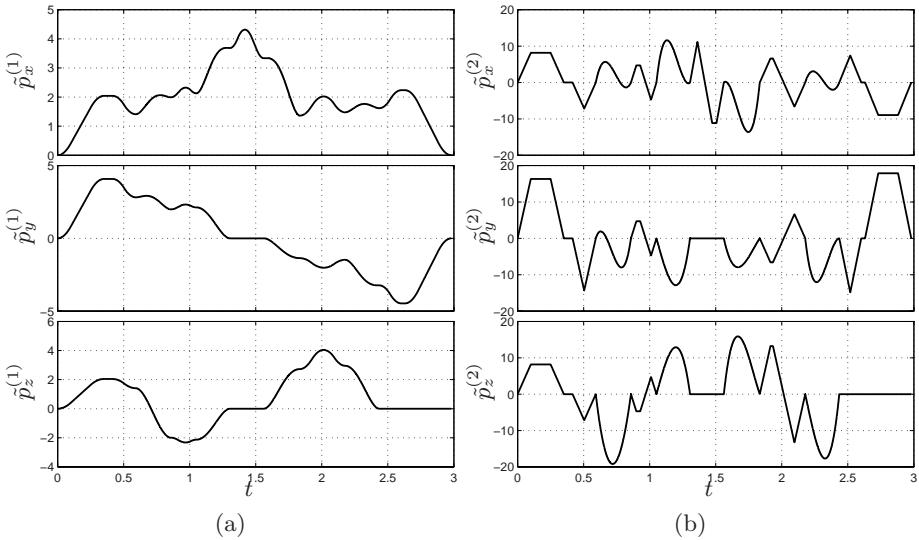


**Fig. 9.4.** Magnitude of velocity and acceleration of the trajectory composed by straight line segments and 5-th degree Bézier curves.

where  $l_k$  is the length of the linear segment ( $= |q'_{k+1} - q''_k|$ ),  $|\mathbf{t}_{k\mp 1}|$  is the norm of the tangent vector of the  $(k \mp 1)$ -th Bézier segment, at the end and at the start points respectively,  $\lambda_{k\mp 1}$  the scaling factor of the reparameterization of the  $(k \mp 1)$ -th Bézier segment.

The conditions on initial and final values of the velocity of the reparameterization  $u(t)$  guarantee the continuity of the overall trajectory. Moreover, for the first and the last linear segment the conditions  $\dot{u}_0 = 0$ ,  $\dot{u}_1 = 0$  are considered. The resulting velocity and acceleration are reported in Fig. 9.4 (magnitude) and Fig. 9.5 (components). Note that the time duration of the trajectory is considerably shorter than the one produced by a constant scaling (Example 9.1), although in this case the initial and final velocities are null. This is due to the fact that each segment is optimized, i.e. it reaches the maximum allowed value of velocity or acceleration (compare Fig. 9.4 with Fig. 9.2).

□



**Fig. 9.5.** Components of the velocity  $\tilde{\mathbf{p}}^{(1)}(t)$  (a) and of the acceleration  $\tilde{\mathbf{p}}^{(2)}(t)$  (b) of the trajectory composed by linear and 5-th degree Bézier segments.

## 9.4 Constant Feed Rate

A special problem in trajectory generation for multi-axis cutting/milling machines, which can be solved by means of a correct reparameterization, consists in obtaining a *constant feed rate*. The goal is to plan a motion of the tool in the workspace with a constant speed [94]. In this case, the function  $u(t)$  must guarantee that

$$|\dot{\tilde{\mathbf{p}}}(t)| = v_c \quad (\text{constant}) \quad (9.6)$$

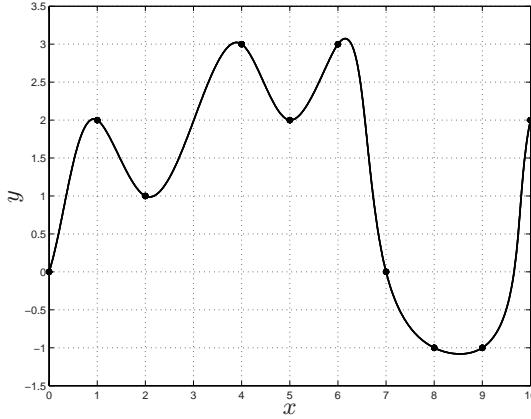
where  $\tilde{\mathbf{p}}(t) = (\mathbf{p} \circ u)(t)$ . Since nowadays automatic machines are controlled by computers, it is not necessary to obtain analytically the function  $u(t)$ : its values  $u(t_k) = u_k$  can be numerically computed in real-time at each sampling time  $t_k = kT_s$ , [95]. The calculation of  $u_k$ ,  $k = 0, 1, \dots$ , is based on the following Taylor expansion of  $u(t)$

$$u_{k+1} = u_k + T_s \dot{u}_k + \frac{T_s^2}{2} \ddot{u}_k + \mathcal{O}\left(\frac{T_s^n}{n!} u_k^{(n)}\right), \quad k = 0, 1, \dots \quad (9.7)$$

From (9.6) and (9.1) the following condition is obtained

$$\dot{u}(t) = \frac{v_c}{\left| \frac{d\mathbf{p}}{du} \right|} \quad (9.8a)$$

while, by differentiating (9.6) with respect to time, after some calculations one obtains



**Fig. 9.6.** Geometric path defined with a cubic B-spline  $s(u)$  ( $0 \leq u \leq 1$ ).

$$\ddot{u}(t) = -v_c^2 \frac{\frac{d\mathbf{p}^T}{du} \cdot \frac{d^2\mathbf{p}}{du^2}}{\left| \frac{d\mathbf{p}}{du} \right|^4}. \quad (9.8b)$$

Therefore, if the first order approximation of  $u(t)$  is considered, the value of the variable  $u$  at time  $(k+1)T_s$  can be determined as

$$u_{k+1} = u_k + \frac{v_c T_s}{\left| \frac{d\mathbf{p}}{du} \right|_{u_k}} \quad (9.9a)$$

while, by considering the second order approximation,

$$u_{k+1} = u_k + \frac{v_c T_s}{\left| \frac{d\mathbf{p}}{du} \right|_{u_k}} - \frac{(v_c T_s)^2}{2} \left[ \frac{\frac{d\mathbf{p}^T}{du} \cdot \frac{d^2\mathbf{p}}{du^2}}{\left| \frac{d\mathbf{p}}{du} \right|^4} \right]_{u_k}. \quad (9.9b)$$

**Example 9.3** Figure 9.6 shows a cubic B-spline interpolating the set of 10 via-points

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 0 & 2 & 1 & 3 & 2 & 3 & 0 & -1 & -1 & 2 \end{bmatrix}$$

with initial and final derivatives

$$\mathbf{t}_0 = \begin{bmatrix} 8.07 \\ 16.14 \end{bmatrix}, \quad \mathbf{t}_9 = \begin{bmatrix} 5.70 \\ 17.11 \end{bmatrix}.$$

The B-spline is completely determined by the knot vector

$$\mathbf{u} = [0, 0, 0, 0, 0.12, 0.20, 0.35, 0.43, 0.51, 0.69, 0.76, 0.82, 1, 1, 1, 1]$$

and by the control points

$$\mathbf{P} = \begin{bmatrix} 0 & 0.33 & 0.69 & 2.38 & 3.67 & 4.96 & 6.60 & 6.49 & 7.76 & 9.96 & 9.66 & 10 \\ 0 & 0.66 & 3.12 & -0.46 & 4.53 & 1.18 & 4.54 & 0.17 & -1.08 & -1.15 & 1.00 & 2 \end{bmatrix}^T.$$

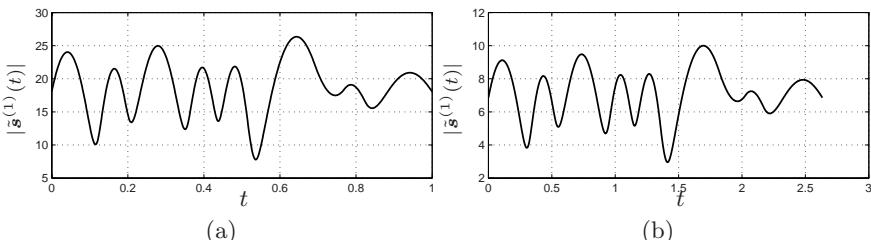
If  $u = t$ , the trajectory is characterized by an extremely variable feed rate, as shown in Fig. 9.7(a). With a constant scaling in time  $u = \lambda t$  it is possible to modify  $|\tilde{\mathbf{s}}^{(1)}(u)|$  with the purpose of obtaining desired mean or maximum values of the speed, but its shape remain unchanged. For instance, in order find the “optimal” trajectory through the given set of data points for which  $|\tilde{\mathbf{s}}^{(1)}(t)| \leq v_{max} = 10$ , it is sufficient to chose  $\lambda = \frac{v_{max}}{|\tilde{\mathbf{s}}^{(1)}(u)|_{max}} = 0.37$ . The new velocity profile is reported in Fig. 9.7(b). By applying (9.9a) with  $v_c = 10$  (and  $T_s = 0.01s$ ), the motion is executed with a speed oscillating around the desired value within a range smaller than 1.5%, see Fig. 9.8 where the relation between  $u$  and  $t$  is also reported. By adopting a second order approximation, given by (9.9b), the variations of the speed around  $v_c = 10$  are further reduced, at the expense of a higher computational complexity, see Fig. 9.9.  $\square$

It is worth noticing that, from a computational point of view, the calculation of (9.9a) and (9.9b) is rather heavy, involving at each sampling time the computation of the magnitude of the first and second derivatives of  $\mathbf{p}(u)$ . It is therefore preferable the construction of a parametric curve characterized by a tangent vector with unitary magnitude over the entire curve, i.e.

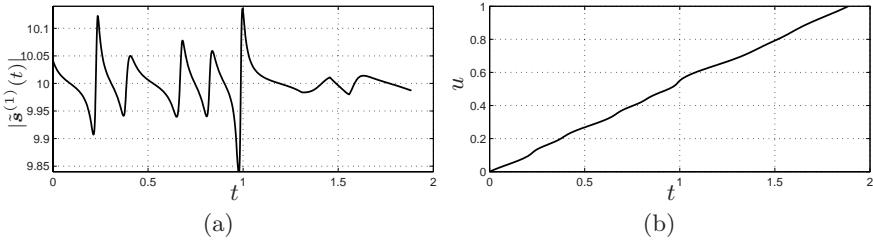
$$\left| \frac{d\mathbf{p}}{du}(u) \right| = 1, \quad \forall u \in [\mathbf{u}_{min}, \mathbf{u}_{max}].$$

This is possible by adopting an arc-length parameterization, obtaining in this way a so-called *uniform parameterization*. In this case, the reparameterization of the curve which produces a constant feed rate equal to  $v_c$  is straightforward:

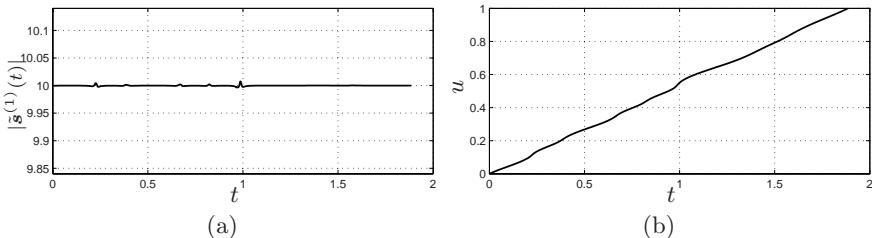
$$u = v_c t \tag{9.10}$$



**Fig. 9.7.** Feed rate of the trajectory defined with a cubic B-spline, with  $u = t$  (a), and with  $u = \lambda t$  (b).



**Fig. 9.8.** Feed rate of the trajectory defined with a cubic B-spline, with  $u$  computed by means of (9.9a) (a), and relation between  $t$  and  $u$  (b).



**Fig. 9.9.** Feed rate of the trajectory defined with a cubic B-spline, with  $u$  computed by means of (9.9b) (a), and relation between  $t$  and  $u$  (b).

or, for a sampled system,

$$u_{k+1} = u_k + v_c T_s$$

where  $T_s$  is the sampling time. Moreover, under the condition that the tangent vector has unitary magnitude over all the geometric curve, it is possible to handle in a simple way particular conditions, such as the start/end phases of the trajectory, and to impose desired profile of the feed rate.

## 9.5 Generic Feed Rate Profile

Equations (9.8a) and (9.8b) are special instances of a more general case in which  $|\dot{\tilde{p}}(t)|$  (and possibly also  $|\ddot{\tilde{p}}(t)|$ ) is not constant, but is a function of time i.e.  $|\dot{\tilde{p}}(t)| = v(t)$  ( $|\ddot{\tilde{p}}(t)| = a(t)$ ) [96]. In this case, the parameterization of the curve, which guarantees that a desired profile of  $v(t)$  (and of  $a(t)$ ) is obtained, can be again computed in real-time by considering the Taylor expansion of  $u(t)$  expressed by (9.7), where  $\dot{u}$  and  $\ddot{u}$  are

$$\dot{u}(t) = \frac{\mathbf{v}(t)}{\left| \frac{d\mathbf{p}}{du} \right|} \quad (9.11a)$$

$$\ddot{u}(t) = \frac{\mathbf{a}(t)}{\left| \frac{d\mathbf{p}}{du} \right|} - \frac{\mathbf{v}(t)^2 \left( \frac{d\mathbf{p}^T}{du} \cdot \frac{d^2\mathbf{p}}{du^2} \right)}{\left| \frac{d\mathbf{p}}{du} \right|^4}. \quad (9.11b)$$

Finally, for discrete time systems with sampling period  $T_s$ , it is possible to compute the new value of  $u_k = u(kT_s)$  as

$$u_{k+1} = u_k + \frac{\mathbf{v}_k T_s}{\left| \frac{d\mathbf{p}}{du} \right|_{u_k}} + \frac{T_s^2}{2} \left( \frac{\mathbf{a}_k}{\left| \frac{d\mathbf{p}}{du} \right|_{u_k}} - \mathbf{v}_k^2 \left[ \frac{\left( \frac{d\mathbf{p}^T}{du} \cdot \frac{d^2\mathbf{p}}{du^2} \right)}{\left| \frac{d\mathbf{p}}{du} \right|^4} \right]_{u_k} \right) \quad (9.12)$$

where  $\mathbf{v}_k = \mathbf{v}(kT_s)$  and  $\mathbf{a}_k = \mathbf{a}(kT_s)$  are the samples of the desired profiles of velocity and acceleration at time  $kT_s$ . Note that the motion law  $q(t)$ , with velocity  $\mathbf{v}(t)$  and acceleration  $\mathbf{a}(t)$ , can be computed with one of the techniques reported in previous sections by assuming a displacement equal to the curve length. As a matter of fact, by integrating  $\mathbf{v}(t) = |\dot{\mathbf{p}}(t)|$  with respect to the time, one obtains

$$q(t) = \int_{t_{min}}^t \mathbf{v}(\tau) d\tau = \int_{u_{min}}^{u(t)} \left| \frac{d\mathbf{p}(u)}{du} \right| du$$

and in particular<sup>2</sup>  $q(t_f) = \int_{u_{min}}^{u_{max}} \left| \frac{d\mathbf{p}(u)}{du} \right| du$ , that is the length of the curve<sup>3</sup>.

When the geometric path is described by an arc-length parameterization, the displacement induced by the motion law is simply equal to  $u_{max} - u_{min}$ , and (9.12) becomes

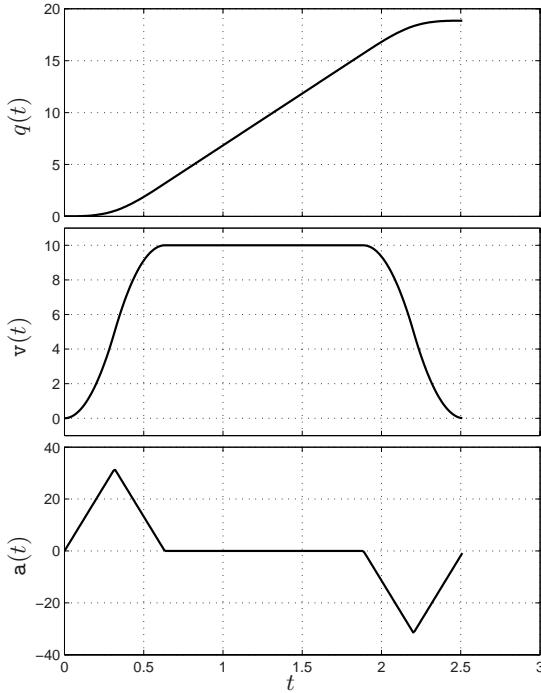
$$u_{k+1} = u_k + \mathbf{v}_k T_s + \frac{T_s^2}{2} \left( \mathbf{a}_k - \mathbf{v}_k^2 \left( \frac{d\mathbf{p}^T}{du} \cdot \frac{d^2\mathbf{p}}{du^2} \right)_{u_k} \right)$$

**Example 9.4** The same geometric path of the previous example must be tracked with a motion law which guarantees that the feed rate has a “bell” profile. In this way initial and final velocities and accelerations are null. This is obtained by considering for the feed rate a double S trajectory from  $q_0 = 0$  to  $q_1 = l$  where  $l$  is the length of the curve, obtained by numerically estimating [19] the integral

---

<sup>2</sup> It is assumed  $q(t_f) = 0$ .

<sup>3</sup> The case  $u_{min} \leq u \leq u_{max}$  is considered.



**Fig. 9.10.** Trajectory of the desired feed rate.

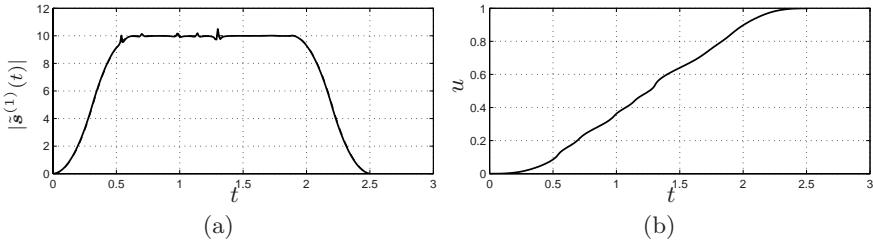
$$\int_0^1 \left| \frac{d\mathbf{s}(u)}{du} \right| du.$$

Moreover, the constraints

$$v_{max} = 10, \quad a_{max} = 50, \quad j_{max} = 100$$

on the maximum value of the feed rate and of its derivatives are assumed. The desired profile of the feed rate  $v(t)$  is reported in Fig. 9.10.

Equation (9.12) is used to compute at each sampling time the value of  $u_{k+1}$  on the basis of tangent and curvature vectors of the spline at  $u_k$ , and of the velocity and acceleration value at time  $kT_s$  (with  $T_s = 0.001s$ ). Note that the position profile  $q(t)$  used to define the desired feed rate  $v(t)$  (and possibly also  $a(t)$ ) is not employed, but only  $\dot{q}(t) = v(t)$  and  $\ddot{q}(t) = a(t)$  are exploited. The resulting feed rate is reported in Fig. 9.11, as well as the profile of the independent variable  $u$  used to evaluate the curve  $\mathbf{s}(u)$ . The approach based on the choice of a desired feed rate, imposed by properly computing the value of  $u_k$  at each sampling time, is quite different from the adoption of a desired motion law for  $u(t)$ , as presented in Sec. 9.3. By assuming a desired profile of the variable  $u(t)$ , it is possible to consider constraints on the maximum feed rate, which remains rather oscillating. For instance,  $u(t)$  can be computed ac-



**Fig. 9.11.** Feed rate of the trajectory defined with a cubic B-spline, with  $u$  computed by means of (9.12) (a), and relation between  $t$  and  $u$  (b).

cording to a double S trajectory from  $u_{min} = 0$  to  $u_{max} = 1$ , with a maximum speed

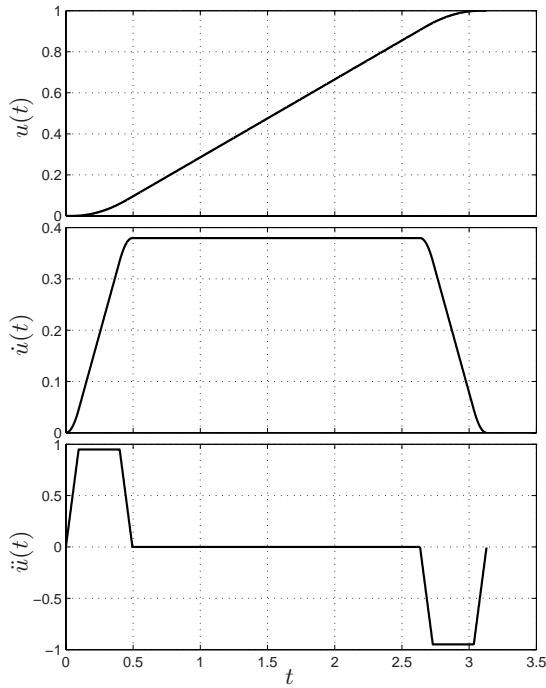
$$\dot{u}_{max} = \frac{v_{max}}{|p^{(1)}(u)|_{max}}.$$

This condition guarantees that the feed rate is not greater than  $v_{max}$ . More difficult is the choice of  $\ddot{u}_{max}$  and  $\ddot{u}_{max}$  from the constraints on the feed rate  $a_{max}$ ,  $j_{max}$ , since the expressions of the acceleration of  $\ddot{p}(t)$ , given by (9.1), and of higher order derivatives, lead to overconservative constraints on  $u(t)$ . In the example reported in Fig. 9.12 the values

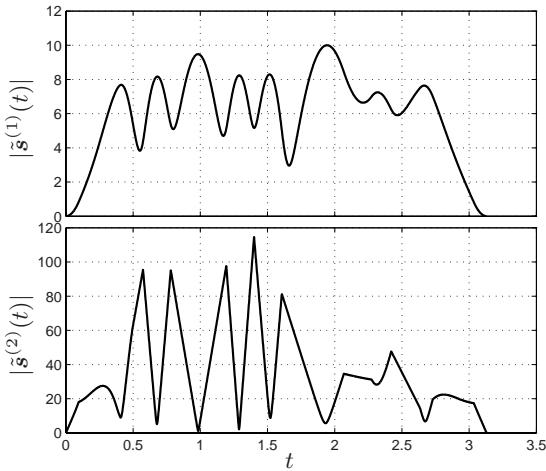
$$\dot{u}_{max} = 0.37, \quad \ddot{u}_{max} = 0.94, \quad \dddot{u}_{max} = 10$$

are used. Figure 9.13 shows the consequent profile of the feed rate and of its first derivative.

□



**Fig. 9.12.** Motion profile of the variable  $u(t)$ .



**Fig. 9.13.** Feed rate induced by a double S trajectory for  $u(t)$ .

## 9.6 Integration of Geometric Path and Motion Law for Complex 3D Tasks

In this section the main issues related to the design of complex trajectories, concerning both position and orientation in the 3D space, are shown by means of some examples, discussed step by step. In particular, given a set of via points, four cases are analyzed:

1. Approximation by means of a linear trajectory with polynomial blends.
2. Interpolation by a B-spline trajectory.
3. Approximation by a smoothing B-spline trajectory.
4. Approximation by a B-spline of a trajectory based on motion primitives.

### 9.6.1 Linear trajectory with polynomial blends

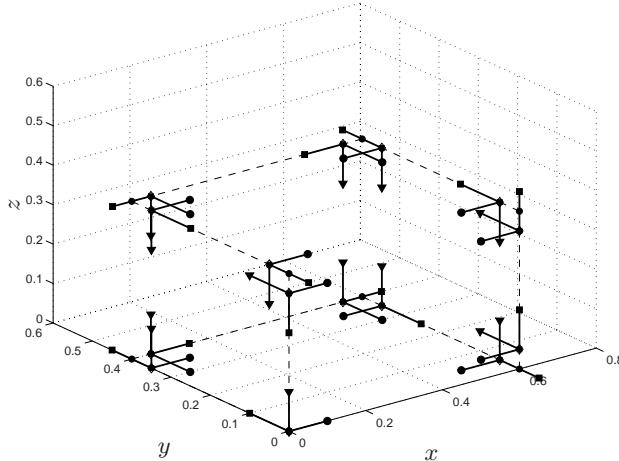
Let us consider the case of a trajectory composed by straight lines properly joined in order to avoid discontinuities of the speed and acceleration, see Sec. 8.11. As a case study, we assume a motion composed by eight linear tracts defined by the via-points

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} 0.0 & 0.0 & 0.6 & 0.6 & 0.6 & 0.6 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.4 & 0.4 & 0.0 & 0.0 & 0.4 & 0.4 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.4 & 0.4 & 0.4 & 0.4 & 0.0 \end{bmatrix}.$$

Moreover, it is desired to keep constant the orientation in each one of the eight tracts. In particular, the rotation matrices defining the desired orientation with respect to the base frame are

$$\begin{aligned} \mathbf{R}_0 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{R}_1 &= \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{R}_2 &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{R}_3 &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 2 & 0 \end{bmatrix} \\ \mathbf{R}_4 &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, & \mathbf{R}_5 &= \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \\ \mathbf{R}_6 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, & \mathbf{R}_7 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}. \end{aligned}$$

The via-points with the related local frames are shown in Fig. 9.14. Since the goal is to design a trajectory composed by linear segments with polynomial



**Fig. 9.14.** Via-points and related local frames considered for linear interpolation with 5-th degree Bézier blends.

blends (in particular 5-th degree Bézier curves are adopted), besides the via-points<sup>4</sup>, also the endpoints of the linear tracts are reported. They are obtained by intersecting the lines joining the points  $[q_k, q_{k+1}]$  with balls of radius  $\delta$  and centered in  $q_k$  and  $q_{k+1}$  respectively. In this example, it is assumed  $\delta = 0.5$  for all the via-points. Then, the sequence of points, interpolated alternatively by means of straight lines and Bézier curves, is

$$\mathbf{Q}' = [q''_0, q'_1, q''_1, \dots, q''_{k-1}, q'_k, q''_k, \dots, q''_{n-1}, q'_n] = \\ \begin{bmatrix} 0 & 0 & 0.05 & 0.55 & 0.60 & 0.60 & 0.60 & 0.60 & 0.60 & 0.55 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0.35 & 0.40 & 0.40 & 0.35 & 0.05 & 0 & 0 & 0.05 & 0.35 & 0.40 & 0.40 & 0.35 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.05 & 0.35 & 0.40 & 0.40 & 0.40 & 0.40 & 0.40 & 0.40 & 0.35 & 0 \end{bmatrix}.$$

As a first step, the trajectory for the position is computed by means of the algorithm reported in Sec. 8.11. Therefore, 5-th degree Bézier curves are inserted between adjacent straight lines. The control points defining the trajectory segments<sup>5</sup> are

<sup>4</sup> Note that with this approach the via-points are approximated and not interpolated.

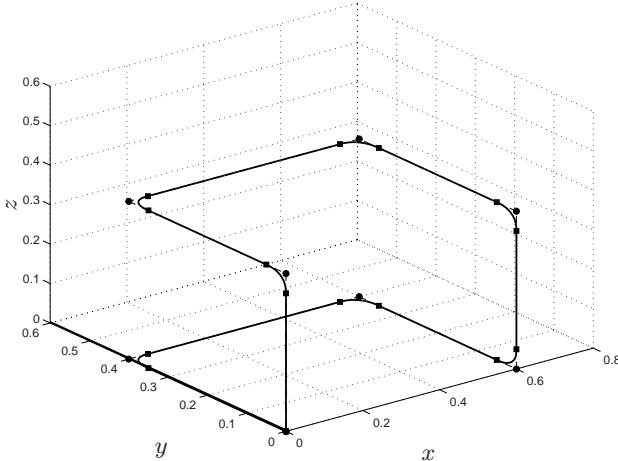
<sup>5</sup> Note that a straight line tract can be seen as 1-st degree Bézier curve expressed by

$$\mathbf{p}(u) = \mathbf{p}_0(1-u) + \mathbf{p}_1 u, \quad u \in [0, 1]$$

where  $\mathbf{p}_0$  and  $\mathbf{p}_1$  are the two control points, corresponding to the initial and final point of the segment itself.

$$\begin{aligned}
\mathbf{p}_{0,0} &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, & \mathbf{p}_{1,0} &= \begin{bmatrix} 0 \\ 0.35 \\ 0 \end{bmatrix} \\
\mathbf{p}_{0,1} &= \begin{bmatrix} 0 \\ 0.35 \\ 0 \end{bmatrix}, & \mathbf{p}_{1,1} &= \begin{bmatrix} 0 \\ 0.36 \\ 0 \end{bmatrix}, & \mathbf{p}_{2,1} &= \begin{bmatrix} 0 \\ 0.38 \\ 0 \end{bmatrix}, & \mathbf{p}_{3,1} &= \begin{bmatrix} 0.01 \\ 0.40 \\ 0 \end{bmatrix}, & \mathbf{p}_{4,1} &= \begin{bmatrix} 0.03 \\ 0.40 \\ 0 \end{bmatrix}, & \mathbf{p}_{5,1} &= \begin{bmatrix} 0.05 \\ 0.40 \\ 0 \end{bmatrix} \\
\mathbf{p}_{0,2} &= \begin{bmatrix} 0.05 \\ 0.40 \\ 0 \end{bmatrix}, & \mathbf{p}_{1,2} &= \begin{bmatrix} 0.55 \\ 0.40 \\ 0 \end{bmatrix} \\
\mathbf{p}_{0,3} &= \begin{bmatrix} 0.55 \\ 0.40 \\ 0 \end{bmatrix}, & \mathbf{p}_{1,3} &= \begin{bmatrix} 0.56 \\ 0.40 \\ 0 \end{bmatrix}, & \mathbf{p}_{2,3} &= \begin{bmatrix} 0.58 \\ 0.40 \\ 0 \end{bmatrix}, & \mathbf{p}_{3,3} &= \begin{bmatrix} 0.60 \\ 0.38 \\ 0 \end{bmatrix}, & \mathbf{p}_{4,3} &= \begin{bmatrix} 0.60 \\ 0.36 \\ 0 \end{bmatrix}, & \mathbf{p}_{5,3} &= \begin{bmatrix} 0.60 \\ 0.35 \\ 0 \end{bmatrix} \\
\mathbf{p}_{0,4} &= \begin{bmatrix} 0.60 \\ 0.35 \\ 0 \end{bmatrix}, & \mathbf{p}_{1,4} &= \begin{bmatrix} 0.60 \\ 0.05 \\ 0 \end{bmatrix} \\
\mathbf{p}_{0,5} &= \begin{bmatrix} 0.60 \\ 0.05 \\ 0 \end{bmatrix}, & \mathbf{p}_{1,5} &= \begin{bmatrix} 0.60 \\ 0.03 \\ 0 \end{bmatrix}, & \mathbf{p}_{2,5} &= \begin{bmatrix} 0.60 \\ 0.01 \\ 0 \end{bmatrix}, & \mathbf{p}_{3,5} &= \begin{bmatrix} 0.60 \\ 0 \\ 0.01 \end{bmatrix}, & \mathbf{p}_{4,5} &= \begin{bmatrix} 0.60 \\ 0 \\ 0.03 \end{bmatrix}, & \mathbf{p}_{5,5} &= \begin{bmatrix} 0.60 \\ 0 \\ 0.05 \end{bmatrix} \\
\mathbf{p}_{0,6} &= \begin{bmatrix} 0.60 \\ 0 \\ 0.05 \end{bmatrix}, & \mathbf{p}_{1,6} &= \begin{bmatrix} 0.60 \\ 0 \\ 0.35 \end{bmatrix} \\
\mathbf{p}_{0,7} &= \begin{bmatrix} 0.60 \\ 0 \\ 0.35 \end{bmatrix}, & \mathbf{p}_{1,1} &= \begin{bmatrix} 0.60 \\ 0 \\ 0.36 \end{bmatrix}, & \mathbf{p}_{2,7} &= \begin{bmatrix} 0.60 \\ 0 \\ 0.38 \end{bmatrix}, & \mathbf{p}_{3,7} &= \begin{bmatrix} 0.60 \\ 0.01 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{4,7} &= \begin{bmatrix} 0.60 \\ 0.03 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{5,7} &= \begin{bmatrix} 0.60 \\ 0.05 \\ 0.40 \end{bmatrix} \\
\mathbf{p}_{0,8} &= \begin{bmatrix} 0.60 \\ 0.05 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{1,8} &= \begin{bmatrix} 0.60 \\ 0.35 \\ 0.40 \end{bmatrix} \\
\mathbf{p}_{0,9} &= \begin{bmatrix} 0.60 \\ 0.35 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{1,9} &= \begin{bmatrix} 0.60 \\ 0.36 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{2,9} &= \begin{bmatrix} 0.60 \\ 0.38 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{3,9} &= \begin{bmatrix} 0.58 \\ 0.40 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{4,9} &= \begin{bmatrix} 0.56 \\ 0.40 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{5,9} &= \begin{bmatrix} 0.55 \\ 0.40 \\ 0.40 \end{bmatrix} \\
\mathbf{p}_{0,10} &= \begin{bmatrix} 0.55 \\ 0.40 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{1,10} &= \begin{bmatrix} 0.05 \\ 0.40 \\ 0.40 \end{bmatrix} \\
\mathbf{p}_{0,11} &= \begin{bmatrix} 0.05 \\ 0.40 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{1,11} &= \begin{bmatrix} 0.03 \\ 0.40 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{2,11} &= \begin{bmatrix} 0.01 \\ 0.40 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{3,11} &= \begin{bmatrix} 0 \\ 0.38 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{4,11} &= \begin{bmatrix} 0 \\ 0.36 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{5,11} &= \begin{bmatrix} 0 \\ 0.35 \\ 0.40 \end{bmatrix} \\
\mathbf{p}_{0,12} &= \begin{bmatrix} 0 \\ 0.35 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{1,12} &= \begin{bmatrix} 0 \\ 0.05 \\ 0.40 \end{bmatrix} \\
\mathbf{p}_{0,13} &= \begin{bmatrix} 0 \\ 0.05 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{1,13} &= \begin{bmatrix} 0 \\ 0.03 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{2,13} &= \begin{bmatrix} 0 \\ 0.01 \\ 0.40 \end{bmatrix}, & \mathbf{p}_{3,13} &= \begin{bmatrix} 0 \\ 0 \\ 0.38 \end{bmatrix}, & \mathbf{p}_{4,13} &= \begin{bmatrix} 0 \\ 0 \\ 0.36 \end{bmatrix}, & \mathbf{p}_{5,13} &= \begin{bmatrix} 0 \\ 0 \\ 0.35 \end{bmatrix} \\
\mathbf{p}_{0,14} &= \begin{bmatrix} 0 \\ 0 \\ 0.35 \end{bmatrix}, & \mathbf{p}_{1,14} &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.
\end{aligned}$$

Each tract  $\mathbf{p}_i(u)$ ,  $i = 0, \dots, 14$  is computed for values of the independent variable  $u$  ranging in  $[0, 1]$ . Therefore the duration of the trajectory in terms of  $u$  is 15.



**Fig. 9.15.** Position trajectory obtained with linear and 5-th degree Bézier segments.

Although the geometric path, reported in Fig. 9.15, is geometrically continuous, velocity and accelerations are discontinuous, see Fig. 9.16(a) where the first derivative (properly shifted with respect to  $u$ ) of each Bézier tracts is shown. For this reason, it is necessary to reparameterize the trajectory. For each segment composing the overall curve, it is assumed

$$u = \frac{\hat{u} - \hat{u}_i}{\lambda_i}, \quad \text{with } \hat{u} \in [\hat{u}_i, \hat{u}_{i+1}], \quad i = 0, \dots, 14 \quad (9.13)$$

where the constant parameters  $\lambda_i$  and  $\hat{u}_i$  are

$$\lambda_i = \begin{cases} 5|\mathbf{p}_{1,i} - \mathbf{p}_{0,i}|, & \text{for 5-th degree Bézier segments} \\ |\mathbf{p}_{1,i} - \mathbf{p}_{0,i}|, & \text{for linear segments} \end{cases}$$

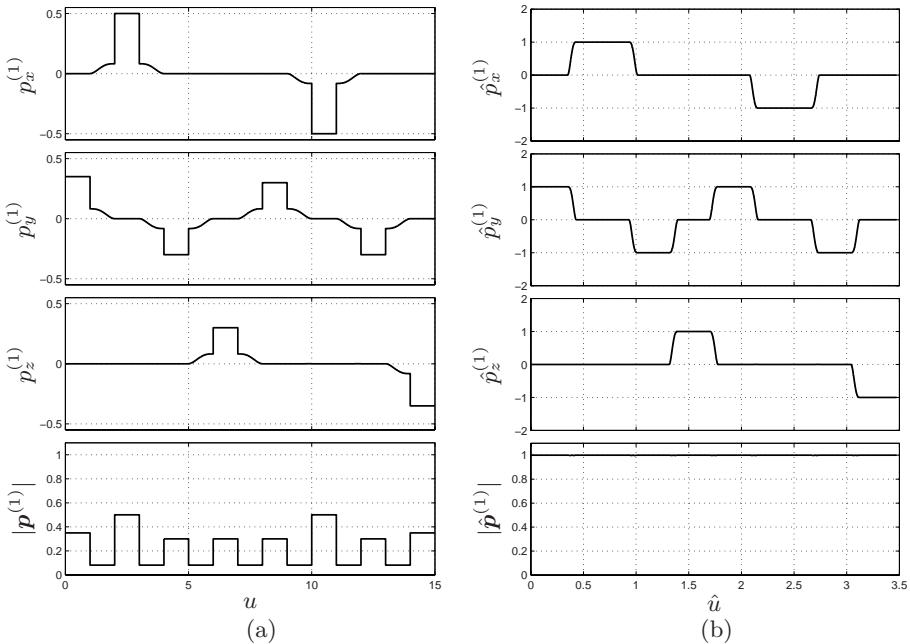
and

$$\hat{u}_i = \begin{cases} 0, & \text{if } i = 0 \\ \hat{u}_{i-1} + \lambda_{i-1}, & \text{if } i > 0. \end{cases} \quad (9.14)$$

Note that the “duration” of each tract, in terms of the variable  $\hat{u}$ , is  $\lambda_i$ . With this reparameterization of each segment, the trajectory  $\hat{\mathbf{p}}(\hat{u})$  is continuous in velocity (and acceleration) and, moreover, is defined as a function of a unique independent variable  $u \in [0, \hat{u}_{15}]$ , instead of a set of functions, each one defined for  $\hat{u} \in [0, \lambda_i]$ .

The derivative of  $\hat{\mathbf{p}}(\hat{u})$  with respect to  $\hat{u}$  is shown in Fig. 9.16(b). It is worth noticing that the magnitude of the “velocity” is practically equal to one.

Since the terms  $\hat{u}_i$ ,  $i = 0, \dots, 15$ , computed with (9.14), determine initial and final values of the segments which compose the trajectory and are defined



**Fig. 9.16.** First derivative (components and magnitude) of a trajectory composed by linear and 5-th degree Bézier segments, before (a) and after (b) the reparametrization.

by assigning the points  $\mathbf{Q}'$  to be interpolated, they are a sort of “time instants” in which these points are crossed. In this example, the vector of such terms is

$$\hat{\mathbf{u}} = [0, 0.35, 0.43, 0.93, 1.01, 1.31, 1.39, 1.69, 1.77, 2.07, 2.15, 2.65, 2.74, 3.04, 3.12, 3.47].$$

The duration of the overall trajectory in terms of  $\hat{u}$  is  $\Delta\hat{u} = \hat{u}_{15} - \hat{u}_0 = 3.47$ . In order to associate a trajectory for the orientation to the position trajectory obtained so far, it is therefore sufficient to link to each element of  $\mathbf{Q}'$  and  $\hat{\mathbf{u}}$  the frame defining the desired orientation, see Fig. 9.14. In this example, the sequence of desired frame is

$$\mathbf{R} = [\mathbf{R}_0, \mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_3, \mathbf{R}_4, \mathbf{R}_4, \mathbf{R}_5, \mathbf{R}_5, \mathbf{R}_6, \mathbf{R}_6, \mathbf{R}_7, \mathbf{R}_7]$$

where  $\mathbf{R}_k$ ,  $k = 0, \dots, 7$  are the rotation matrices which define the desired orientation over the linear paths. For each pairs of elements of  $\mathbf{R}$ , denoted with  $\mathbf{R}_i$ ,  $i = 0, \dots, 15$ , the transformation between them in terms of axis/angle notation is found as described in Sec. 8.2.1. In particular, in the linear tracts the orientation is maintained constant (in this case, for  $i$  even,  $\mathbf{R}_i = \mathbf{R}_{i+1}$ ), while the angle variation and the axis value in the Bézier blends ( $i$  odd) are

$$\begin{aligned}
\theta_{t,1} &= \frac{\pi}{2}, & \mathbf{w}_1 &= [0, 0, -1]^T \\
\theta_{t,3} &= \frac{\pi}{2}, & \mathbf{w}_3 &= [0, 0, -1]^T \\
\theta_{t,5} &= \frac{\pi}{2}, & \mathbf{w}_5 &= [1, 0, 0]^T \\
\theta_{t,7} &= \frac{\pi}{2}, & \mathbf{w}_7 &= [1, 0, 0]^T \\
\theta_{t,9} &= \frac{\pi}{2}, & \mathbf{w}_9 &= [0, 0, -1]^T \\
\theta_{t,11} &= \frac{\pi}{2}, & \mathbf{w}_{11} &= [0, 0, -1]^T \\
\theta_{t,13} &= \frac{\pi}{2}, & \mathbf{w}_{13} &= [1, 0, 0]^T .
\end{aligned}$$

In order to impose the variation of the angle  $\theta_{t,i}$  about the axis  $\mathbf{w}_i$  in the interval  $[\hat{u}_i, \hat{u}_{i+1}]$ , the trajectory  $\theta_i(\hat{u})$  has been computed by assuming a 5-th degree polynomial with the following boundary conditions

$$\begin{aligned}
\hat{u}_{0,i} &= \hat{u}_i, & \hat{u}_{1,i} &= \hat{u}_{i+1} \\
\theta_{0,i} &= 0, & \theta_{1,i} &= \theta_{t,i} \\
\dot{\theta}_{0,i} &= 0, & \dot{\theta}_{1,i} &= 0 \\
\ddot{\theta}_{0,i} &= 0, & \ddot{\theta}_{1,i} &= 0 .
\end{aligned}$$

The trajectory for the orientation is described by

$$\mathbf{R}(\hat{u}) = \begin{cases} \mathbf{R}_i, & \hat{u} \in [\hat{u}_i, \hat{u}_{i+1}], \quad i \text{ even} \\ \mathbf{R}_i \mathbf{R}_{t,i}(\theta_i(\hat{u})), & \hat{u} \in [\hat{u}_i, \hat{u}_{i+1}], \quad i \text{ odd} \end{cases} \quad (9.15)$$

where  $\mathbf{R}_{t,i}(\theta_i)$  is the matrix which defines the transition from  $\mathbf{R}_i$  to  $\mathbf{R}_{i+1}$  as a function of the variable  $\theta_i$ , see Sec. 8.2. The complete trajectory, for position and orientation, is plotted in Fig. 9.17 as a function of  $\hat{u}$ .

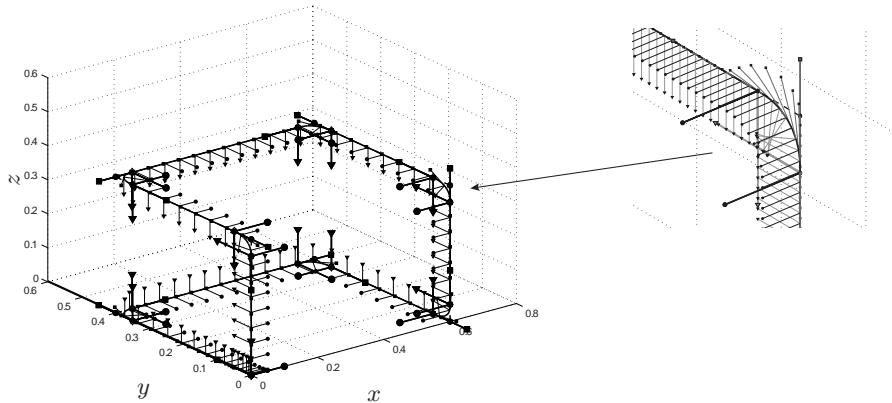
Finally, it is necessary to associate to the geometric path  $(\hat{\mathbf{p}}(\hat{u}), \mathbf{R}(\hat{u}))$  a particular motion law, by imposing the desired  $\hat{u}(t)$ . For the definition of the motion law, only the trajectory  $\hat{\mathbf{p}}(\hat{u})$  for the position is taken into account. In this example, we assume a motion at a constant feed rate but with smooth starting and ending phases, i.e. with null initial and final velocities/accelerations. Therefore, by exploiting the technique reported in Sec. 9.5, it is assumed a desired feed rate  $v(t)$  characterized by the velocity profile of a double S trajectory  $q^{ss}(t)$ , with a total displacement<sup>6</sup>  $h = 3.47$  and with the constraints (which lead to an overall duration of 2 seconds)

$$v_{max} = 1.82, \quad a_{max} = 27.41, \quad j_{max} = 822.54.$$

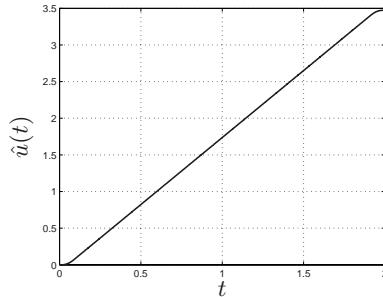
By imposing

---

<sup>6</sup> The displacement is equal to the length of the curve  $\hat{\mathbf{p}}(\hat{u})$ ,  $\hat{u} \in [0, 3.47]$ , given by the sum the length of each segment, computed by numerically integrating  $\left| \frac{d\hat{\mathbf{p}}_i(\hat{u})}{d\hat{u}} \right|$  over the interval  $[\hat{u}_i, \hat{u}_{i+1}]$ .



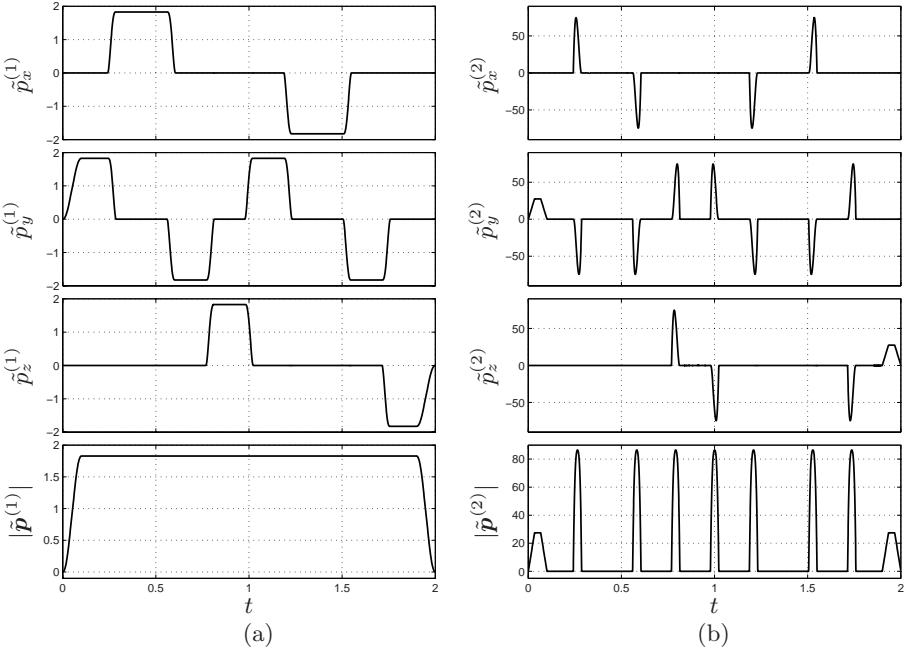
**Fig. 9.17.** Changes in the orientation superimposed to the position trajectory composed by linear and 5-th degree Bézier segments, with a particular of a Bézier blend.



**Fig. 9.18.** Motion law associated to the geometric path composed by linear and 5-th degree Bézier segments.

$$\mathbf{v}_k = \dot{\mathbf{q}}^{ss}(kT_s), \quad \mathbf{a}_k = \ddot{\mathbf{q}}^{ss}(kT_s)$$

in eq. (9.12), used for the computation of  $\hat{u}(t)$  at the  $k$ -th time instant, the profiles  $\hat{u}(t)$  of Fig. 9.18 and the trajectory velocities/accelerations of Fig. 9.19 are obtained. The function  $|\tilde{\mathbf{p}}^{(1)}(t)|$  is the velocity of a double S motion law with the considered constraints, while  $|\tilde{\mathbf{p}}^{(2)}(t)|$  includes both the acceleration components given by the initial acceleration and final deceleration (characterized by a trapezoidal shape) and the acceleration contributions due to the variation of the tangent vector in the Bézier blends, which are proportional to  $\dot{\mathbf{u}}^2(t)$ . Note that this latter contribution has a magnitude considerably larger than the former. If there exists a constraint on the maximum magnitude of the acceleration, one may scale in time the motion law  $\hat{u}(t)$ , according to the methods reported in Sec. 5.2.



**Fig. 9.19.** Velocity and acceleration (components and magnitude) of a trajectory composed by linear and 5-th degree Bézier segments, with a double S feed rate.

For instance, with the motion law  $\hat{u}(t)$  of Fig. 9.18 the acceleration magnitude is 85.76. If the acceleration must be limited by the maximum value  $|\tilde{\mathbf{p}}^{(2)}|_{max,d} = 40$ , i.e.

$$|\tilde{\mathbf{p}}^{(2)}(t)| \leq |\tilde{\mathbf{p}}^{(2)}|_{max,d} = 40$$

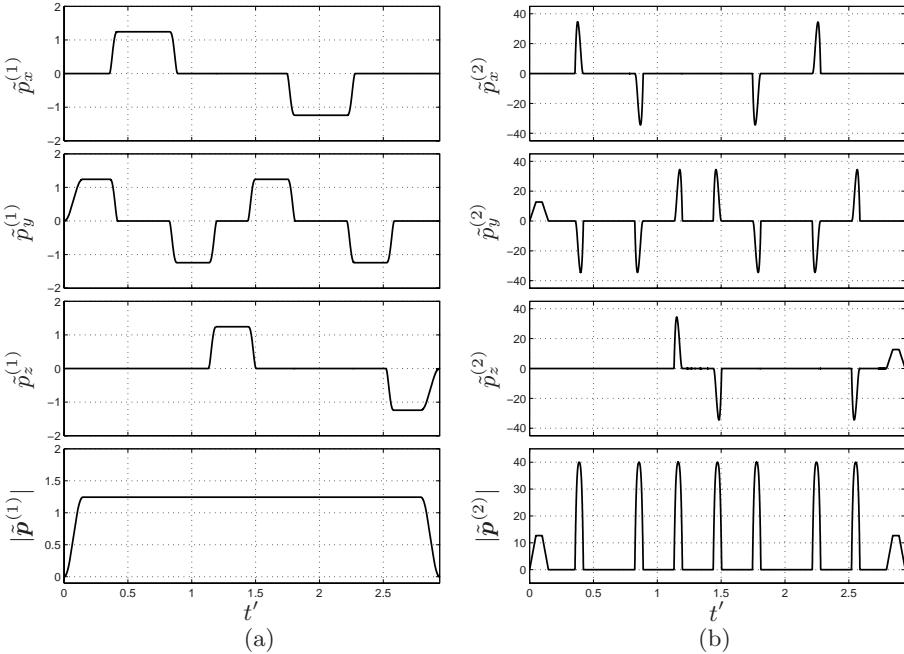
it is then necessary to assume

$$t' = \frac{t}{\lambda}, \quad \text{with} \quad \lambda = \sqrt{\frac{|\tilde{\mathbf{p}}^{(2)}|_{max,d}}{|\tilde{\mathbf{p}}^{(2)}|_{max}}} = 0.68.$$

The new velocity and acceleration profiles, re-scaled in time, are reported in Fig. 9.20. The duration of the trajectory is now  $T' = 2.92$ .

If, as in this case, the trajectory is characterized by a uniform parameterization  $|d\hat{\mathbf{p}}(\hat{u})/d\hat{u}| \approx 1, \forall \hat{u}$ , a similar result can be obtained by reducing the maximum feed rate value. Therefore, in order to obtain a trajectory compliant with the new constraint on the maximum acceleration, it is sufficient to consider a double S profile with the constraint

$$v'_{max} = v_{max} \sqrt{\frac{|\tilde{\mathbf{p}}^{(2)}|_{max,d}}{|\tilde{\mathbf{p}}^{(2)}|_{max}}} = 1.82 \sqrt{\frac{40}{85.74}} = 1.24$$



**Fig. 9.20.** Velocity and acceleration (components and magnitude) of a trajectory composed by linear and 5-th degree Bézier segments, properly scaled in time in order to have a maximum acceleration of 40.

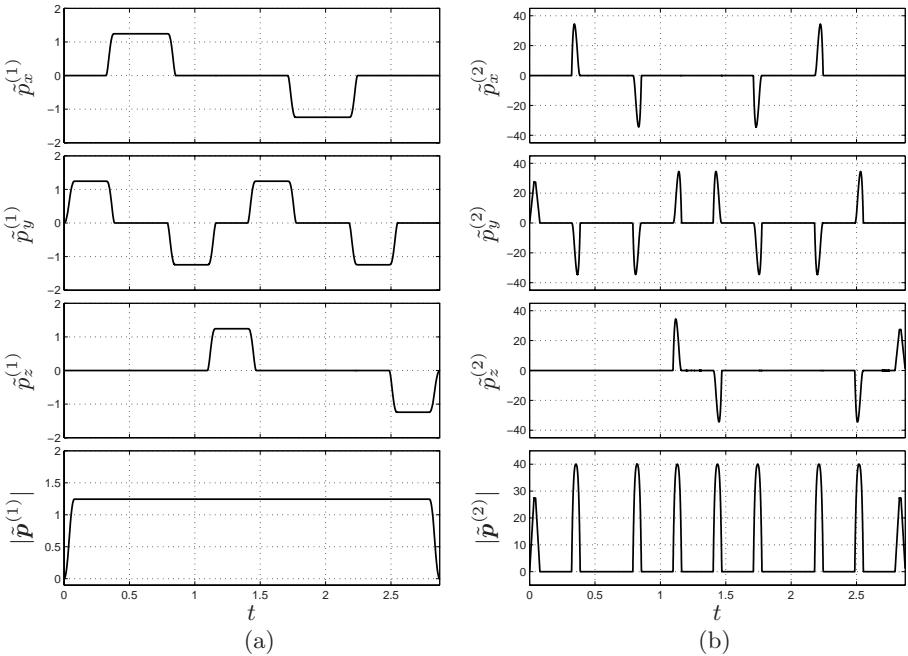
see Fig. 9.21. Also in this case the feed rate is constant, and the duration  $T = 2.8720$  of the trajectory is considerably larger than the original one.

An alternative approach for the definition of  $u(t)$ , which allows to handle in a simple manner the constraints on the maximum speed/acceleration, consists in considering the segments of the trajectory separately, applying to each of them a different motion law. For example, a simple constant velocity trajectory may be assumed in the Bézier tracts, and a double S profile in the linear segments. Obviously, these motion laws have to be designed in such a way to guarantee the continuity of the velocity and acceleration profiles along the geometric path. For this reason, this method can be directly applied to the curve  $\mathbf{p}(u)$  before the reparameterization (9.13).

Given the trajectory segments  $\mathbf{p}_i(u)$ ,  $i = 0, \dots, 14$  with  $u \in [0, 1]$ , the first step consists in computing the constraints on  $u(t)$  which limit the velocity  $|\tilde{\mathbf{p}}^{(1)}(t)|$  and acceleration  $|\tilde{\mathbf{p}}^{(2)}(t)|$  below the desired values. In this example, the maximum values are

$$v_{max} = 1.82, \quad a_{max} = 40, \quad j_{max} = 822.54$$

from which the constraints on  $u(t)$  result



**Fig. 9.21.** Velocity and acceleration (components and magnitude) of a trajectory composed by linear and 5-th degree Bézier segments, with a motion law which limits the maximum acceleration.

$$\left\{ \begin{array}{l} \dot{u}_{max,i} = \frac{v_{max}}{\left| \frac{d\mathbf{p}_i(u)}{du} \right|_{max}} \\ \ddot{u}_{max,i} = \frac{a_{max}}{\left| \frac{d^2\mathbf{p}_i(u)}{du^2} \right|_{max}} \\ \ddot{u}_{max,i} = \frac{j_{max}}{\left| \frac{d^3\mathbf{p}_i(u)}{du^3} \right|_{max}} \end{array} \right. \quad (9.16)$$

for straight line segments ( $i$  even), and

$$\dot{u}_{max,i} = \min \left\{ \frac{v_{max}}{\left| \frac{d\mathbf{p}_i(u)}{du} \right|_{max}}, \sqrt{\frac{a_{max}}{\left| \frac{d^2\mathbf{p}_i(u)}{du^2} \right|_{max}}} \right\}$$

if  $\mathbf{p}_i(u)$  is a Bézier function<sup>7</sup> ( $i$  odd).

Once the maximum values of speed and acceleration of each tract are computed (in particular the velocities in the Bézier segments), it is possible to deduce the initial and final velocities of the linear tracts (with the only exception of the initial and final segments for which, respectively, initial and final velocities are null) that guarantee the continuity of the overall trajectory velocity:

$$\left\{ \begin{array}{l} \dot{u}_{0,i} = \dot{u}_{max,i-1} \left| \frac{d\mathbf{p}_{i-1}(u)}{du} \right|_{u=1} \\ \quad \left| \frac{d\mathbf{p}_i(u)}{du} \right|_{u=0} \\ \dot{u}_{1,i} = \dot{u}_{max,i+1} \left| \frac{d\mathbf{p}_{i+1}(u)}{du} \right|_{u=0} \\ \quad \left| \frac{d\mathbf{p}_i(u)}{du} \right|_{u=1} \end{array} \right. \quad i \text{ even.} \quad (9.17)$$

Then, for each linear tract, a double S trajectory from  $u_{min} = 0$  to  $u_{max} = 1$  and starting at  $t = t_i$  (being  $t_i$  the final time of previous segment) is computed, while a motion law<sup>8</sup> with a constant velocity  $\dot{u}_{max,i}$ , given by (9.17), is adopted for Bézier blends. The geometric path defined by  $\tilde{\mathbf{p}}(t) = \mathbf{p}(u)$ , with  $u = u(t)$ , does not change, but the resulting velocities/accelerations, shown in Fig. 9.22, are rather different from the ones obtained in previous examples. Note that in each tract either the maximum acceleration or the maximum velocity is reached, and the duration of the trajectory ( $T = 2.24$ ) is considerably reduced with respect to the previous cases, in which the velocity has been reduced along the overall geometric path to meet the constraint on  $\mathbf{a}_{max}$ .

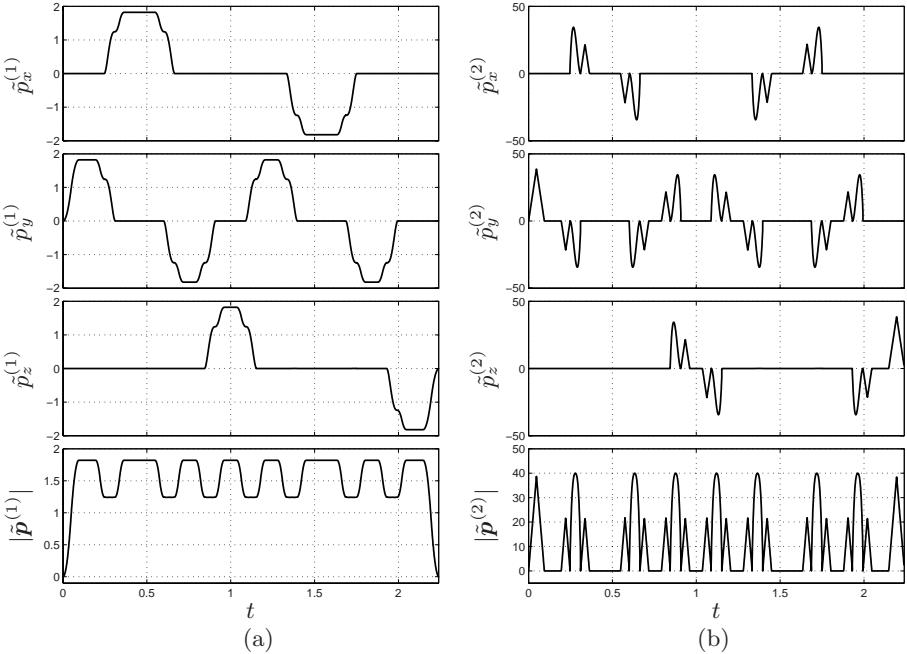
For what concerns the orientation, it is possible to define the related trajectory as a function of the time  $t$ , by assuming  $\theta_i(t)$  in (9.15) as a 5-th degree polynomial, computed with the conditions

$$\begin{aligned} t_{0,i} &= t_i, & t_{1,i} &= t_{i+1} \\ \theta_{0,i} &= 0, & \theta_{1,i} &= \theta_{t,i} \\ \dot{\theta}_{0,i} &= 0, & \dot{\theta}_{1,i} &= 0 \\ \ddot{\theta}_{0,i} &= 0, & \ddot{\theta}_{1,i} &= 0. \end{aligned}$$

---

<sup>7</sup> In this case, it is assumed a motion law with a constant velocity  $u(t) = u_{max,i}$ .

<sup>8</sup> Also in this case the constraints are  $u_{min} = 0$ ,  $u_{max} = 1$ , with the initial time  $t = t_i$ .



**Fig. 9.22.** Velocity and acceleration (components and magnitude) of a trajectory composed by linear and 5-th degree Bézier segments, with a motion law optimized in each tract.

### 9.6.2 B-spline trajectory

The same set of points used in the previous section is considered:

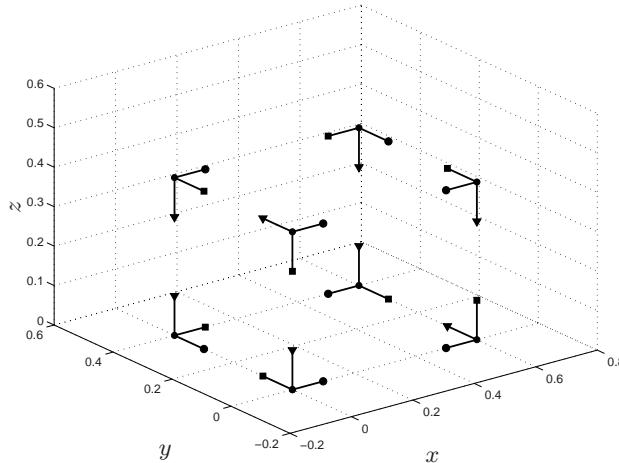
$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} 0 & 0.0 & 0.6 & 0.6 & 0.6 & 0.6 & 0.0 & 0.0 & 0 \\ 0 & 0.4 & 0.4 & 0.0 & 0.0 & 0.4 & 0.4 & 0.0 & 0 \\ 0 & 0.0 & 0.0 & 0.0 & 0.4 & 0.4 & 0.4 & 0.4 & 0 \end{bmatrix}.$$

The goal is now to use a cubic B-spline for their interpolation, as described in Sec. 8.4. Also in this case the orientation is specified by means of local frames, described by rotation matrices defined for each data point, see Fig. 9.23. In particular, the frames related to the points  $q_k$ ,  $k = 0, \dots, 8$  are

$$\mathbf{R} = [\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4, \mathbf{R}_5, \mathbf{R}_6, \mathbf{R}_7, \mathbf{R}_0]$$

where the matrices  $\mathbf{R}_k$  are those defined in the previous section. In this case, also the time instants at which the points must be interpolated by the trajectory are specified:

$$\mathbf{t} = [0, 0.5, 0.8, 1.1, 1.6, 1.9, 2.2, 2.5, 3].$$



**Fig. 9.23.** Via-points and related local frames considered for B-spline interpolation.

In the computation of the B-spline (see Sec. 8.4), it is therefore convenient to assume  $\bar{u}_k = t_k$ ,  $k = 0, \dots, 8$ . As a consequence the knots which define the spline are

$$\boldsymbol{u} = [0, 0, 0, 0, 0.5, 0.8, 1.1, 1.6, 1.9, 2.2, 2.5, 3, 3, 3, 3].$$

In order to determine the trajectory for the orientation, the rotation matrices defining the desired configurations are transformed into the corresponding Roll-Pitch-Yaw angles, obtaining the sequence

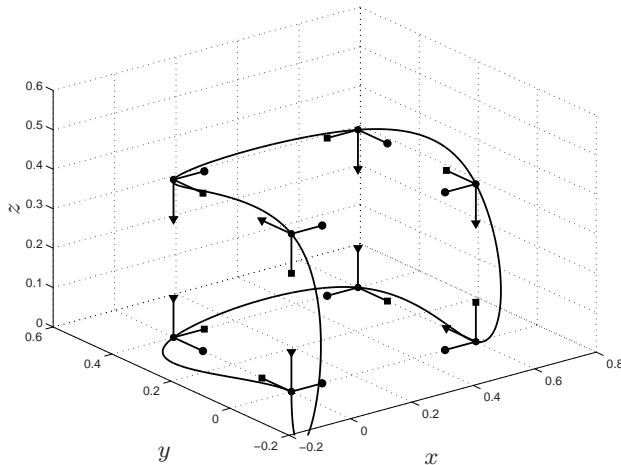
$$\begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} 0 & -1.57 & 3.14 & 3.14 & 3.14 & -1.57 & 0.00 & 0.00 & 0 \\ 0 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0 \\ 0 & 0.00 & 0.00 & 1.57 & 3.14 & 3.14 & 3.14 & -1.57 & 0 \end{bmatrix}$$

which can be interpolated with the same technique used for the position. The synchronization between position and orientation is implicitly obtained, since the same knot vector is used in both cases. The resulting control points for the B-splines defining the position and the orientation are

$$\boldsymbol{P}_{\text{pos}} = \begin{bmatrix} 0 & 0.00 & -0.39 & 0.80 & 0.52 & 0.51 & 0.81 & -0.21 & 0.10 & 0.00 & 0 \\ 0 & 0.13 & 0.41 & 0.51 & -0.20 & -0.19 & 0.49 & 0.50 & -0.24 & 0.00 & 0 \\ 0 & 0.00 & -0.01 & 0.01 & -0.08 & 0.48 & 0.39 & 0.35 & 0.61 & -0.40 & 0 \end{bmatrix}^T$$

and

$$\boldsymbol{P}_{\text{or}} = \begin{bmatrix} 0 & -0.52 & -5.04 & 4.93 & 1.73 & 6.58 & -3.78 & 1.00 & -0.49 & 0.00 & 0 \\ 0 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0 \\ 0 & 0.00 & 0.22 & -0.45 & 2.17 & 3.54 & 2.62 & 4.96 & -5.56 & 1.57 & 0 \end{bmatrix}^T$$

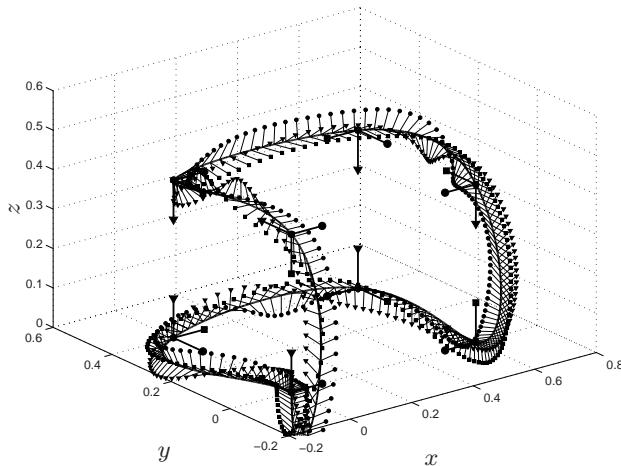


**Fig. 9.24.** Position trajectory obtained with a B-spline curve.

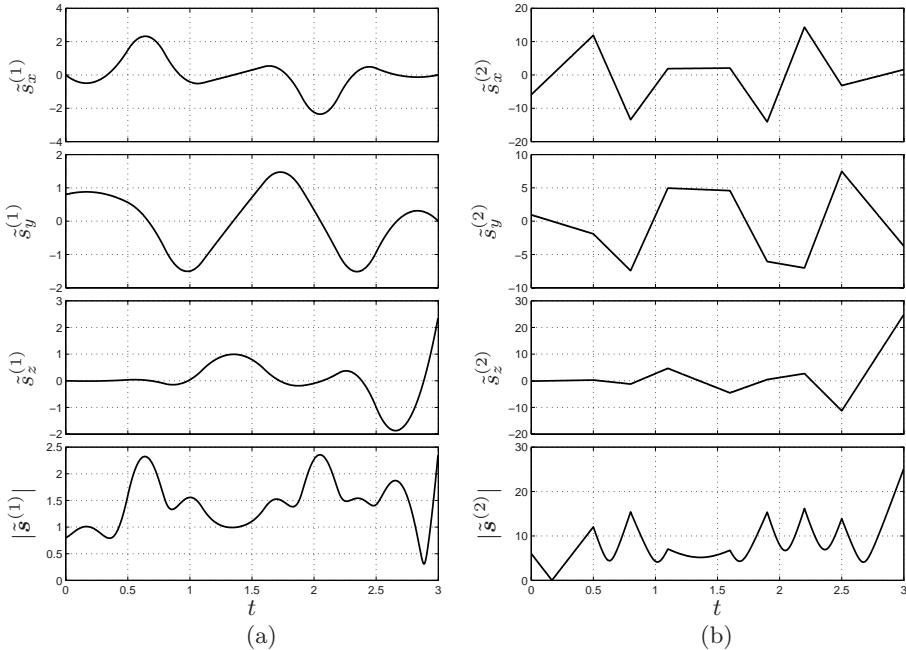
respectively. Figure 9.24 shows the B-spline curve  $s_{\text{pos}}(u)$  which defines the position, while in Fig. 9.25 the frames corresponding to the profiles of Roll-Pitch-Yaw angles described by  $s_{\text{ori}}(u)$  are reported.

These trajectories are obtained by imposing the simple motion law  $u(t) = t$  to the two B-spline curves:

$$\tilde{s}(t) = s(u)|_{u=t}.$$



**Fig. 9.25.** Trajectory for the orientation superimposed to the B-spline position trajectory.



**Fig. 9.26.** Velocity and acceleration (components and magnitude) of a the position trajectory represented by a B-spline curve, with the parameterization  $u(t) = t$ .

In this way, the data points are interpolated at time instants  $t_k = \bar{u}_k$ , and the resulting velocities/accelerations are discontinuous at initial and final points, see Fig. 9.26. This depends on the fact that the parameterization  $u(t) = t$  is characterized by a non-null initial and final velocity  $\dot{u}(t)$  and acceleration  $\ddot{u}(t)$ . In order to obtain null initial and final velocities/accelerations, it is therefore necessary to assume a parameterization with

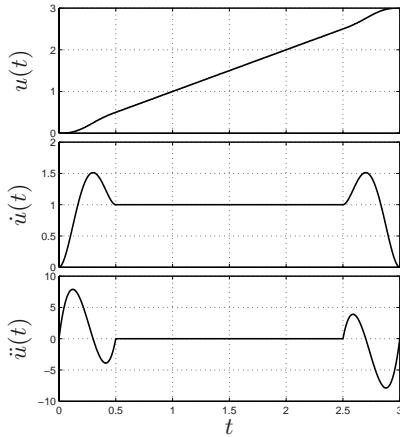
$$\begin{aligned}\dot{u}(t_0) &= 0, & \dot{u}(t_8) &= 0, \\ \ddot{u}(t_0) &= 0, & \ddot{u}(t_8) &= 0.\end{aligned}$$

For this purpose, the function  $u(t) = t$  has been modified by assuming in the time intervals  $[t_0, t_1]$  and  $[t_7, t_8]$  5-th degree polynomial blends, computed with the conditions

$$\begin{aligned}u(t_0) &= \bar{u}_0, & u(t_1) &= \bar{u}_1, \\ \dot{u}(t_0) &= 0, & \dot{u}(t_1) &= 1, \\ \ddot{u}(t_0) &= 0, & \ddot{u}(t_1) &= 0,\end{aligned}$$

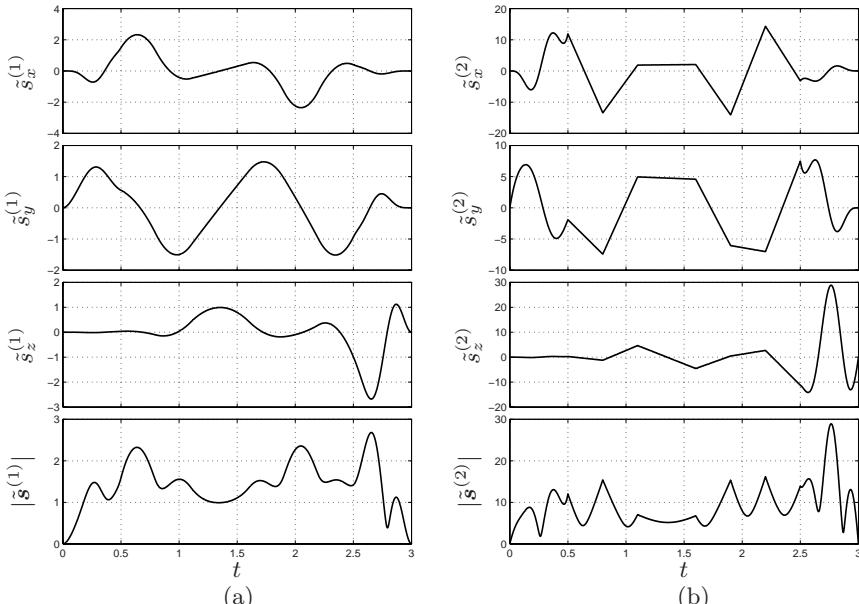
and

$$\begin{aligned}u(t_7) &= \bar{u}_7, & u(t_8) &= \bar{u}_8, \\ \dot{u}(t_7) &= 1, & \dot{u}(t_8) &= 0, \\ \ddot{u}(t_7) &= 0, & \ddot{u}(t_8) &= 0.\end{aligned}$$



**Fig. 9.27.** Smooth parameterization  $u(t)$  for the computation of B-spline trajectories.

The profiles of the modified function  $u(t)$  are reported in Fig. 9.26. Note that with the new parameterization, the desired interpolation time instants are maintained, since  $\bar{u}_k = t_k$ ,  $k = 0, \dots, 8$ . Moreover, the velocity and acceleration profiles have now null initial and final values, see Fig. 9.28.



**Fig. 9.28.** Velocity and acceleration (components and magnitude) of the position trajectory represented by a B-spline curve, with the parameterization of Fig. 9.27.

### 9.6.3 Smoothing B-spline trajectory

In this example, a trajectory approximating a large number of points is considered. In particular, only the position is taken into account, assuming a constant orientation of the tool along the path. Given the points shown in Fig. 9.29, disposed on a plane, the goal consists in finding a curve which approximates them and that must be tracked with a constant velocity.

For this purpose a smoothing B-spline is adopted, see Sec. 8.7. The knots for the computation of  $s(u)$  are assumed with a cord-length distribution in the interval  $[0, 1]$ , and the control points are computed according to the algorithm of Sec. 8.7.1, with unitary weights  $w_k$ , and the coefficient  $\lambda = 10^{-6}$ . Figure 9.30 shows the geometric path described by  $s(u)$ . It is clear that the value of  $\lambda$  assumed in this example guarantees small curvatures, but on the other hand the approximation error ( $\epsilon_{max} = 0.0252$ ) is noticeable.

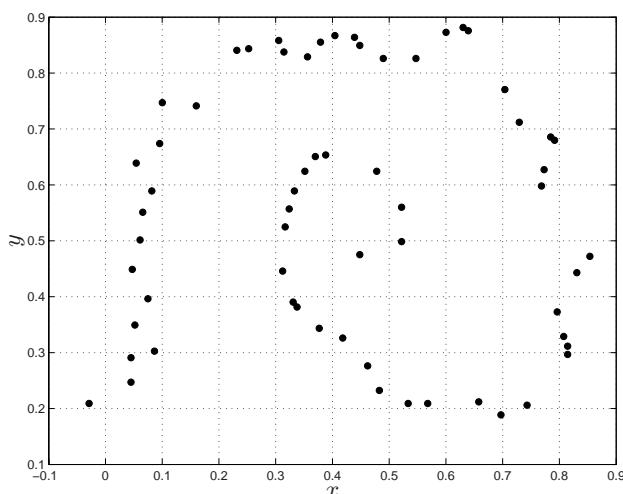
A double S feed rate profile is used to define the motion law  $u(t)$ , with the advantage of obtaining a constant velocity along the most part of the curve and smooth initial/final phases (i.e with null initial/final velocities and accelerations). By adopting the same method exploited in the first example, and explained in details in Sec. 9.5,  $u(t)$  is defined by imposing in (9.12) the values

$$v_k = q^{ss}(kT_s), \quad a_k = \ddot{q}^{ss}(kT_s)$$

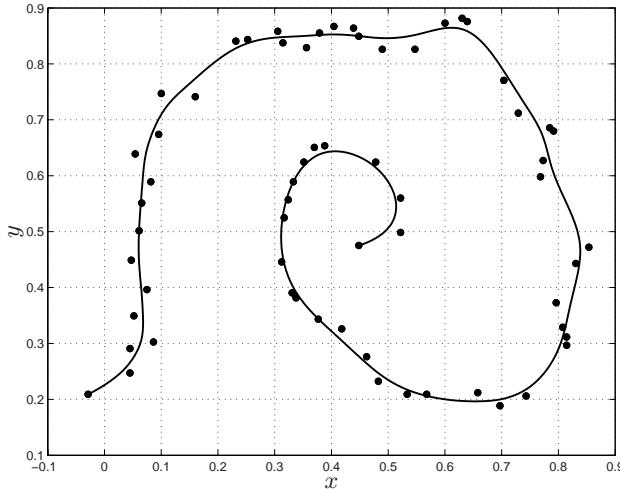
where  $q^{ss}(t)$  is double S trajectory computed for a displacement  $h = 2.9096$  (equal to the length of  $s(u)$ ) and with the conditions

$$v_{max} = 2, \quad a_{max} = 40, \quad j_{max} = 200.$$

The profiles of velocity and acceleration are reported in Fig. 9.31. Note that  $|\tilde{s}^{(1)}(t)|$  is characterized by an ideal double S profile, while the acceleration

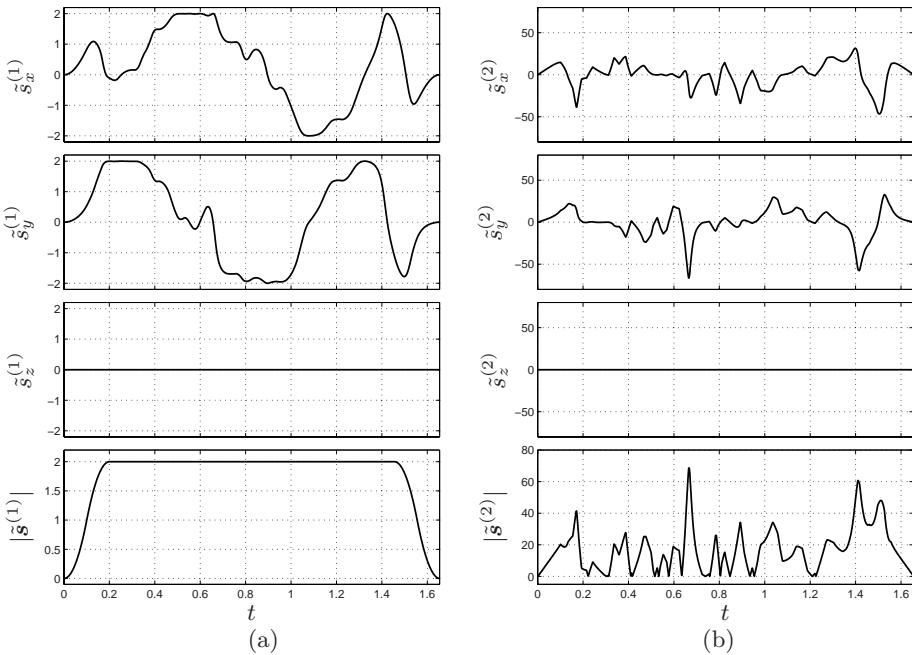


**Fig. 9.29.** Via-points used for the approximation with a smoothing B-spline.

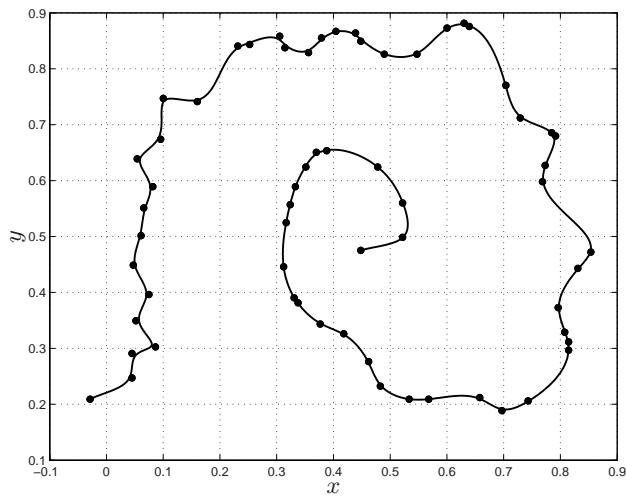


**Fig. 9.30.** Smoothing B-spline approximating a set of data points, for  $\lambda = 10^{-6}$ .

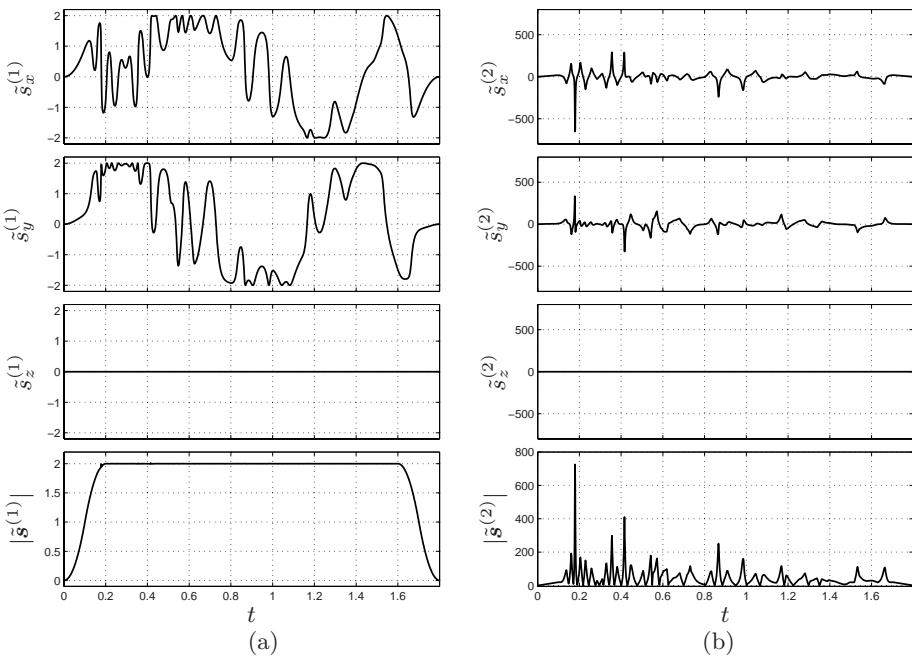
has a behavior extremely variable due to the curvature of the geometric path. If smaller values of  $\lambda$  are considered, the approximation error can be reduced at the expense of higher curvature values. For instance, for  $\lambda = 10^{-8}$  the maximum distance of the spline curve from the via-points is  $\epsilon_{max} = 0.0073$ , see Fig. 9.32. By adopting the same feed rate of the case  $\lambda = 10^{-6}$  with a double S shape, the magnitude of the velocity has the same profiles while its components have a more oscillating behavior, see Fig. 9.33(a). This is due to the fast variations on the velocity direction required to “track” the points. These fast variations are also the cause of the high values of the curvature and the acceleration shown in Fig. 9.33(b).



**Fig. 9.31.** Velocity and acceleration (components and magnitude) of the position trajectory represented by a smoothing B-spline approximating a set of data points, for  $\lambda = 10^{-6}$ .



**Fig. 9.32.** Smoothing B-spline approximating a set of data points, for  $\lambda = 10^{-8}$ .



**Fig. 9.33.** Velocity and acceleration (components and magnitude) of the position trajectory represented by a smoothing B-spline approximating a set of data points, for  $\lambda = 10^{-8}$ .

### 9.6.4 B-spline approximation of a trajectory based on motion primitives

Because of their simple use, in many industrial applications the definition of the geometric path is often performed by means of motion primitives, such as straight lines, circles, etc. Unfortunately, when linear and circular motions are mixed in order to obtain complex paths, discontinuities in the acceleration profile are inevitable, because of the centripetal component of the acceleration in rotational motions. The only way to avoid these discontinuities is to adopt a motion law with null velocity at the transition points between linear and circular segments. An alternative approach consists in approximating the path with an intrinsically continuous trajectory, for instance based on cubic (or higher degree) B-spline curves. In the example described below, the trajectory to be approximated is the one shown in Fig. 8.17, where the orientation is not taken into account. This trajectory can be approximated by interpolating the points

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} 0 & 1.00 & 1.70 & 2.00 & 2.00 & 2.00 & 2.00 & 2.00 & 2.00 & 2.00 \\ 0 & 0.00 & 0.29 & 0.99 & 1.70 & 2.00 & 1.70 & 1.00 & 0.01 & 0.00 \\ 0 & 0.00 & 0.00 & 0.00 & 0.29 & 0.99 & 1.70 & 2.00 & 2.00 & 2.00 \end{bmatrix}$$

obtained by “sampling” the trajectory itself, see Fig. 9.34 where the original trajectory, and the approximating cubic B-spline are reported. In particular, the knots (chosen according to a cord-length distribution in the interval  $[0, 1]$ ) and the control points (computed with the algorithm of Sec. 8.4) which define the B-spline curve are

$$\mathbf{u} = [0, 0, 0, 0, 0.15, 0.26, 0.38, 0.49, 0.61, 0.73, 0.84, 0.99, 1, 1, 1, 1]$$

and

$$\mathbf{P} = \begin{bmatrix} 0 & 0.33 & 0.91 & 1.79 & 2.05 & 1.98 & 2.00 & 1.99 & 2.00 & 2.00 & 2.00 & 2.00 \\ 0 & 0 & -0.07 & 0.20 & 1.00 & 1.77 & 2.10 & 1.78 & 0.92 & 0.33 & 0.00 & 0 \\ 0 & 0 & -0.00 & 0.01 & -0.05 & 0.20 & 0.99 & 1.79 & 2.07 & 2.00 & 2.00 & 2.00 \end{bmatrix}^T.$$

The maximum approximation error, measured in terms of Hausdorff<sup>9</sup> distance, is  $\epsilon_{max} = 0.0194$ , and the profiles of velocity and acceleration of the trajectory  $\tilde{\mathbf{s}}(t)$ , obtained by imposing on  $\mathbf{s}(u)$  a feed rate described by a double S trajectory with the conditions

---

<sup>9</sup> Given two curves, and more generally two sets of data, the Hausdorff distance provides the maximum distance of a set to the nearest point in the other set. More formally, the Hausdorff distance between two parametric curves  $f_i(s_i)$  is defined as

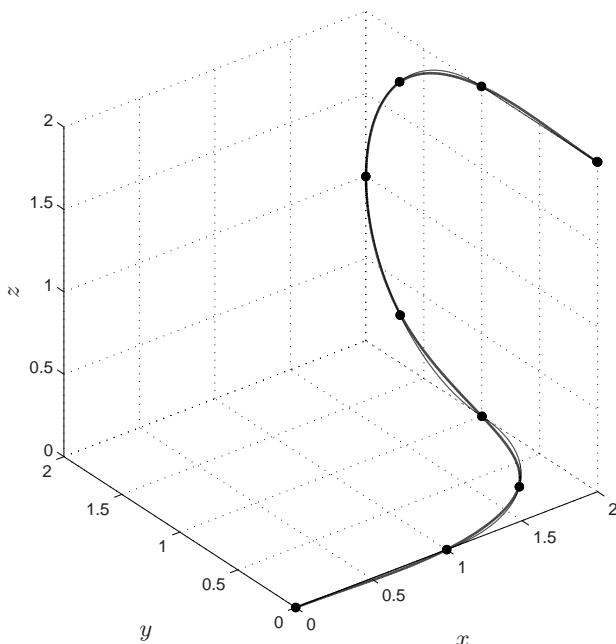
$$d_H(f_1, f_2) = \max \left\{ \frac{\max_{s_1} \min_{s_2} |f_1(s_1) - f_2(s_2)|}{\max_{s_2} \min_{s_1} |f_1(s_1) - f_2(s_2)|} \right\}. \quad (9.18)$$

$$v_{max} = 2, \quad a_{max} = 40, \quad j_{max} = 200,$$

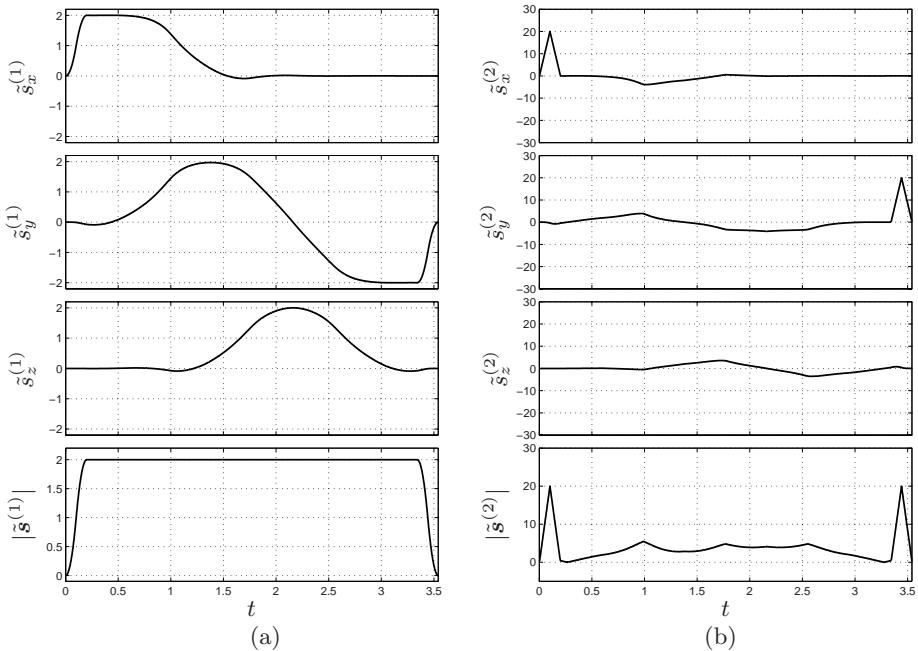
are reported in Fig. 9.35.

Note that the acceleration is continuous, although it is quite different from that one would expect when a trajectory composed by linear and circular segments is tracked with a constant velocity (acceleration/deceleration phases apart). As a matter of fact, the acceleration should be null in the straight line segments and constant along circular paths.

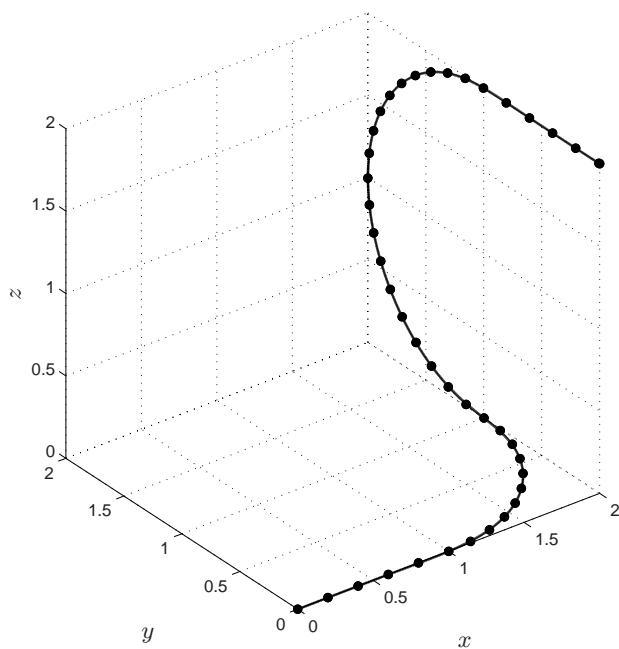
If we increase the number of points of the original trajectory which are interpolated by the B-spline curve, the result is considerably improved. For instance, by assuming that the number of points  $q_k$  is 42, the maximum approximation error is  $\epsilon_{max} = 0.0032$ , as shown in Fig. 9.36. Moreover, also the velocity and acceleration profiles obtained by assuming the same feed rate profile of the previous case are more similar to the ones of a trajectory composed by linear and circular segments, see Fig. 9.37. After the initial acceleration phase, the acceleration is null along the straight line segments, while during circular motions it is practically constant.



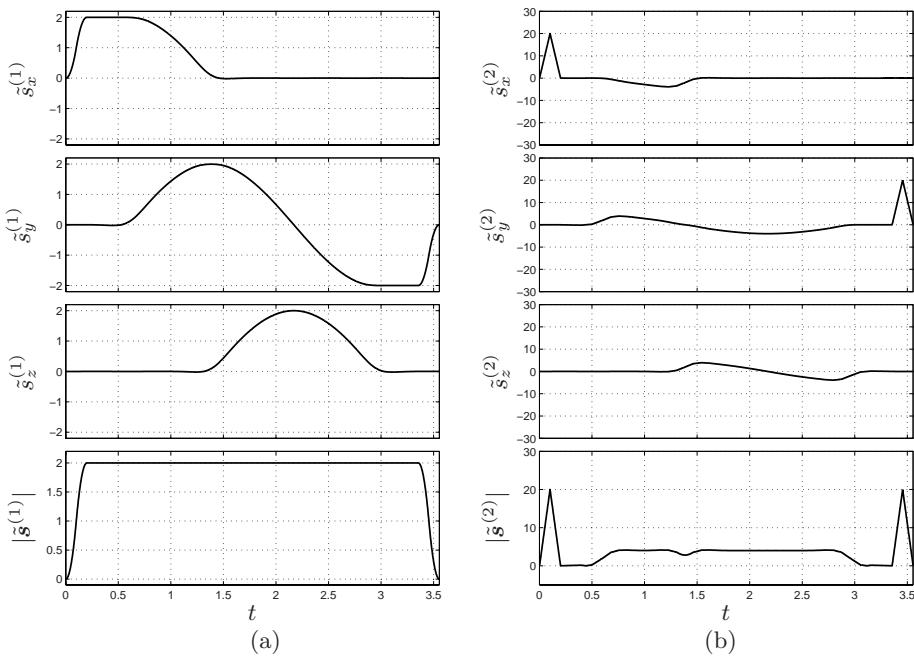
**Fig. 9.34.** Trajectory based on motion primitives, approximated by a B-spline curve with 12 control points.



**Fig. 9.35.** Velocity and acceleration (components and magnitude) of a B-spline curve approximating a trajectory based on motion primitives, with a double S feed rate profile.



**Fig. 9.36.** Trajectory based on motion primitives, approximated by a B-spline curve with 42 control points.



**Fig. 9.37.** Velocity and acceleration (components and magnitude) of a B-spline curve approximating a trajectory based on motion primitives, with a double S feed rate profile.

## **Part IV**

---

### **Appendices**

# A

---

## Numerical Issues

In this Appendix, the expression of the parameters  $a_i$  of normalized polynomials up to degree 21, Sec. 2.1.7, the expressions of the parameters necessary to compute the trajectory '4-3-4', Sec. 3.6, and of the trajectory obtained as a composition of polynomial and trigonometric segments, Sec. 3.11, are reported.

Efficient algorithms for the evaluation of polynomial functions and the solution of tridiagonal systems are also illustrated, see Sec. 4.4.1, and Sec. 4.4.3.

### A.1 Parameters of normalized polynomials $q_N(\tau)$

The coefficients  $a_i$  of the normalized polynomials  $q_N(\tau)$  up to degree 21 with unitary displacement  $h = q_1 - q_0 = 1$  and duration  $T = \tau_1 - \tau_0 = 1$  (without loss of generality, the value  $\tau_0 = 0$  is assumed)

$$q_N(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + \dots + a_n\tau^n = \sum_{i=0}^n a_i\tau^i$$

computed assuming null boundary conditions on velocity, acceleration, jerk, ..., are reported in Tab. A.1. The coefficients of the polynomials  $\dot{q}_N(\tau)$  and  $\ddot{q}_N(\tau)$  defining the velocity and acceleration profiles are reported in Tab. A.2 and Tab. A.3 respectively.

	3	5	7	9	11	13	15	17	19	21
$a_0$	0	0	0	0	0	0	0	0	0	0
$a_1$	0	0	0	0	0	0	0	0	0	0
$a_2$	3	0	0	0	0	0	0	0	0	0
$a_3$	-2	10	0	0	0	0	0	0	0	0
$a_4$	-	-15	35	0	0	0	0	0	0	0
$a_5$	-	6	-84	126	0	0	0	0	0	0
$a_6$	-	-	70	-420	462	0	0	0	0	0
$a_7$	-	-	-20	540	-1980	1716	0	0	0	0
$a_8$	-	-	-	-315	3465	-9009	6435	0	0	0
$a_9$	-	-	-	-	70	-3080	20020	-40040	24310	0
$a_{10}$	-	-	-	-	-	1386	-24024	108108	-175032	92378
$a_{11}$	-	-	-	-	-	-252	16380	-163800	556920	-755820
$a_{12}$	-	-	-	-	-	-	-6006	150150	-1021020	2771340
$a_{13}$	-	-	-	-	-	-	924	-83160	1178100	-5969040
$a_{14}$	-	-	-	-	-	-	-	25740	-875160	8314020
$a_{15}$	-	-	-	-	-	-	-	-3432	408408	-7759752
$a_{16}$	-	-	-	-	-	-	-	-	-109395	4849845
$a_{17}$	-	-	-	-	-	-	-	-	12870	-1956240
$a_{18}$	-	-	-	-	-	-	-	-	-	461890
$a_{19}$	-	-	-	-	-	-	-	-	-	-48620
$a_{20}$	-	-	-	-	-	-	-	-	-	-1939938
$a_{21}$	-	-	-	-	-	-	-	-	-	184756

**Table A.1.** Per column: coefficients  $a_i$  of the normalized polynomials  $q_N(\tau)$  with degree  $n = 3, 5, \dots, 21$ , with null boundary conditions on their derivatives up to order 10. The degree of the polynomials is  $n = 2n_c + 1$ , being  $n_c$  the number of null initial (and final) conditions.

	2	4	6	8	10	12	14	16	18	20
$a_0$	0	0	0	0	0	0	0	0	0	0
$a_1$	6	0	0	0	0	0	0	0	0	0
$a_2$	-6	30	0	0	0	0	0	0	0	0
$a_3$	-	-60	140	0	0	0	0	0	0	0
$a_4$	-	30	-420	630	0	0	0	0	0	0
$a_5$	-	-420	-2520	2772	0	0	0	0	0	0
$a_6$	-	-140	3780	-13860	12012	0	0	0	0	0
$a_7$	-	-	-2520	27720	-72072	51480	0	0	0	0
$a_8$	-	-	630	-27720	180180	-360360	218790	0	0	0
$a_9$	-	-	-	13860	-240240	1081080	-1750320	923780	0	0
$a_{10}$	-	-	-	-2772	180180	-1801800	6126120	-8314020	3879876	
$a_{11}$	-	-	-	-	-72072	1801800	-12252240	33256080	-38798760	
$a_{12}$	-	-	-	-	-12012	-1081080	15315300	-77597520	174594420	
$a_{13}$	-	-	-	-	-	360360	-12252240	116396280	-465585120	
$a_{14}$	-	-	-	-	-	-51480	6126120	-116396280	814773960	
$a_{15}$	-	-	-	-	-	-	-1750320	77597520	-977728752	
$a_{16}$	-	-	-	-	-	-	218790	-33256080	814773960	
$a_{17}$	-	-	-	-	-	-	-	8314020	-465585120	
$a_{18}$	-	-	-	-	-	-	-	-923780	174594420	
$a_{19}$	-	-	-	-	-	-	-	-	-38798760	
$a_{20}$	-	-	-	-	-	-	-	-	3879876	

**Table A.2.** Per column: coefficients  $a_i$  of the normalized polynomials  $\dot{q}_N(\tau)$  with degree  $n = 2, 4, \dots, 20$ .

	1	3	5	7	9	11	13	15	17	19
$a_0$	6	0	0	0	0	0	0	0	0	0
$a_1$	-12	60	0	0	0	0	0	0	0	0
$a_2$	-	-180	420	0	0	0	0	0	0	0
$a_3$	-	120	-1680	2520	0	0	0	0	0	0
$a_4$	-	-	2100	-12600	13860	0	0	0	0	0
$a_5$	-	-	-840	22680	-83160	72072	0	0	0	0
$a_6$	-	-	-	-17640	194040	-504504	360360	0	0	0
$a_7$	-	-	-	-	5040	-221760	1441440	-28892880	1750320	0
$a_8$	-	-	-	-	-	124740	-2162160	9729720	-15752880	8314020
$a_9$	-	-	-	-	-	-	-27720	1801800	-18018000	-83140200
$a_{10}$	-	-	-	-	-	-	-	-792792	19819800	-134774640
$a_{11}$	-	-	-	-	-	-	-	-	-	365816880
$a_{12}$	-	-	-	-	-	-	-	-	-	-426786360
$a_{13}$	-	-	-	-	-	-	-	-	-	-
$a_{14}$	-	-	-	-	-	-	-	-	-	-
$a_{15}$	-	-	-	-	-	-	-	-	-	-
$a_{16}$	-	-	-	-	-	-	-	-	-	-
$a_{17}$	-	-	-	-	-	-	-	-	-	-
$a_{18}$	-	-	-	-	-	-	-	-	-	-
$a_{19}$	-	-	-	-	-	-	-	-	-	-

**Table A.3.** Per column: coefficients  $a_i$  of the normalized polynomials  $\ddot{q}_N(\tau)$  with degree  $n = 1, 3, \dots, 19$ .

## A.2 Parameters of the Trajectory '4-3-4'

A trajectory '4-3-4' is defined by means of 14 parameters, see Sec. 3.6. Some of these parameters are computed on the basis of the initial and final conditions

$$\begin{aligned} a_{0l} &= q_0, & a_{1l} &= 0, & a_{2l} &= 0, \\ a_{0t} &= q_a, \\ a_{0s} &= q_1, & a_{1s} &= 0, & a_{2s} &= 0. \end{aligned}$$

The remaining seven parameters are computed by solving the system with seven equations introduced in Sec. 3.6. Let us define for the sake of simplicity the constants

$$m_0 = 7T_s(2T_l + T_t) + 3T_t(3T_l + 2T_t), \quad h_a = q_a - q_0, \quad h_b = q_b - q_a, \quad h_1 = q_1 - q_b.$$

The parameters are

$$\begin{aligned} a_{4l} &= \frac{-3T_s T_t (T_s + 2T_t) h_a + T_l (-2T_t^2 h_1 + T_s^2 (-2h_a + h_b) + 3T_s T_t (-h_a + h_b))}{T_s m_0} \\ a_{3l} &= \frac{4T_s T_t (T_s + 2T_t) h_a + T_l (2T_t^2 h_1 + T_s^2 (4h_a - h_b) + T_t (6T_s h_a - 3T_s h_b))}{T_s m_0} \\ a_{3t} &= \frac{T_t (2T_t^2 h_1 + 2T_s^2 h_a + T_s T_t (6h_a - h_b)) + T_l (6T_t^2 h_1 - T_s^2 h_b - 6T_s T_t h_b)}{T_s m_0} \\ a_{2t} &= \frac{3(-2T_s T_t (T_s + 2T_t) h_a + T_l (-2T_t^2 h_1 + T_s^2 h_b + 3T_s T_t h_b))}{T_s m_0} \\ a_{1t} &= \frac{T_t (-2T_t^2 h_1 + 3T_s T_t (2h_a + h_b) + T_s^2 (4h_a + h_b))}{T_s m_0} \\ a_{4s} &= \frac{3T_s T_t (2T_l + T_t) h_1 + T_t^2 (3T_l + 2T_t) h_1 + T_s^2 (2T_t h_a - 3T_l h_b - T_t h_b)}{T_t m_0} \\ a_{3s} &= \frac{4T_s T_t (2T_l + T_t) h_1 + 2T_t^2 (3T_l + 2T_t) h_1 + T_s^2 (2T_t h_a - 3T_l h_b - T_t h_b)}{T_t m_0}. \end{aligned}$$

## A.3 Solution of the Equation $\mathbf{M} \mathbf{k} = \mathbf{q}$

The coefficients  $\mathbf{a}_1$ ,  $\mathbf{a}_2$ ,  $k_{ij}$ , solutions of the equation (3.56), with null boundary conditions ( $\mathbf{v}_0 = \mathbf{v}_7 = 0$ ), are computed as follows.

If the elements  $m_{i,j}$  of matrix  $\mathbf{M}$  are known, one can define the parameters

$$\begin{aligned}
s_1 &= (-m_{1,1} + m_{2,1} + m_{3,1} + m_{4,1}) \\
s_2 &= (m_{13,2} + m_{14,2} + m_{15,2} + m_{16,2}) \\
s_3 &= (m_{10,1} + m_{11,1} + m_{12,1}) \\
s_4 &= (m_{5,2} + m_{6,2} + m_{7,2} + m_{8,2}) \\
s_5 &= (-m_{1,1} + m_{2,1}) \\
T_{13} &= (T_1 + T_2 + T_3) \\
T_{23} &= (T_2 + T_3) \\
T_{24} &= (T_2 + T_3 + T_4) \\
T_{25} &= (T_2 + T_3 + T_4 + T_5) \\
T_{26} &= (T_2 + T_3 + T_4 + T_5 + T_6) \\
T_{27} &= (T_2 + T_3 + T_4 + T_5 + T_6 + T_7) \\
T_{14} &= (T_1 + T_2 + T_3 + T_4) \\
T_{15} &= (T_1 + T_2 + T_3 + T_4 + T_5) \\
T_{16} &= (T_1 + T_2 + T_3 + T_4 + T_5 + T_6) \\
T_{17} &= (T_1 + T_2 + T_3 + T_4 + T_5 + T_6 + T_7) \\
T_{34} &= (T_3 + T_4) \\
T_{35} &= (T_3 + T_4 + T_5) \\
T_{36} &= (T_3 + T_4 + T_5 + T_6) \\
T_{37} &= (T_3 + T_4 + T_5 + T_6 + T_7) \\
T_{45} &= (T_4 + T_5) \\
T_{46} &= (T_4 + T_5 + T_6) \\
T_{47} &= (T_4 + T_5 + T_6 + T_7) \\
T_{57} &= (T_5 + T_6 + T_7)
\end{aligned}$$

$$\begin{aligned}
D_1 = & (-s_2 s_1 + (s_3 - m_{9,1}) s_4 \\
& + (-T_{14} m_{1,1} + T_{24} m_{2,1} + T_{34} m_{3,1} + T_{44} m_{4,1}) m_{5,2} + \\
& + (-T_{15} m_{1,1} + T_{25} m_{2,1} + T_{35} m_{3,1} + T_{45} m_{4,1}) m_{6,2} \\
& + (-T_{16} m_{1,1} + T_{26} m_{2,1} + T_{36} m_{3,1} + T_{46} m_{4,1}) m_{7,2} \\
& + (-T_{17} m_{1,1} + T_{27} m_{2,1} + T_{37} m_{3,1} + T_{47} m_{4,1}) m_{8,2}).
\end{aligned}$$

Then, the solution of equation (3.56) is

$$\mathbf{a}_1 = -s_4(q_0 - q_1)/D_1$$

$$\mathbf{a}_2 = s_1(q_0 - q_1)/D_1$$

$$k_{11} = (m_{1,1}s_4(q_0 - q_1))/D_1$$

$$k_{21} = ((m_{1,1} - m_{2,1})s_4(q_0 - q_1))/D_1$$

$$k_{31} = ((m_{1,1} - m_{2,1} - m_{3,1})s_4(q_0 - q_1))/D_1$$

$$k_{41} = (-s_1s_4(q_0 - q_1))/D_1$$

$$k_{51} = (-s_1(m_{6,2} + m_{7,2} + m_{8,2})(q_0 - q_1))/D_1$$

$$k_{61} = (-s_1(m_{7,2} + m_{8,2})(q_0 - q_1))/D_1$$

$$k_{71} = (-s_1m_{8,2}(q_0 - q_1))/D_1$$

$$k_{12} = ((-s_2s_1 + (s_3 - T_{14}m_{1,1} + T_{24}m_{2,1} + T_{34}m_{3,1} + T_4m_{4,1})m_{5,2} +$$

$$+ (s_3 - T_{15}m_{1,1} + T_{25}m_{2,1} + T_{35}m_{3,1} + T_{45}m_{4,1})m_{6,2} +$$

$$+ (s_3 - T_{16}m_{1,1} + T_{26}m_{2,1} + T_{36}m_{3,1} + T_{46}m_{4,1})m_{7,2} +$$

$$+ (s_3 - T_{17}m_{1,1} + T_{27}m_{2,1} + T_{37}m_{3,1} + T_{47}m_{4,1})m_{8,2})q_0 - s_4m_{9,1}q_1)(1/D_1)$$

$$k_{22} = ((-s_2s_1 + (m_{11,1} + m_{12,1} + T_{24}s_5 + T_{34}m_{3,1} + T_4m_{4,1})m_{5,2} +$$

$$+ (m_{11,1} + m_{12,1} + T_{25}s_5 + T_{35}m_{3,1} + T_{45}m_{4,1})m_{6,2} +$$

$$+ (m_{11,1} + m_{12,1} + T_{26}s_5 + T_{36}m_{3,1} + T_{46}m_{4,1})m_{7,2} +$$

$$+ (m_{11,1} + m_{12,1} + T_{27}s_5 + T_{37}m_{3,1} + T_{47}m_{4,1})m_{8,2})q_0 +$$

$$+ (m_{10,1} - m_{9,1} - m_{1,1}T_1)s_4q_1)(1/D_1)$$

$$k_{32} = ((-s_2s_1 + m_{12,1}s_4 + (T_{34}s_5 + T_{34}m_{3,1} + T_4m_{4,1})m_{5,2} +$$

$$+ (T_{35}s_5 + T_{35}m_{3,1} + T_{45}m_{4,1})m_{6,2} + (T_{36}s_5 + T_{36}m_{3,1} + T_{46}m_{4,1})m_{7,2} +$$

$$+ (T_{37}s_5 + T_{37}m_{3,1} + T_{47}m_{4,1})m_{8,2})q_0 +$$

$$+ ((m_{10,1} + m_{11,1} + T_2m_{2,1} - m_{9,1}) - m_{1,1}(T_1 + T_2))s_4q_1)(1/D_1)$$

$$k_{42} = ((-s_2 + T_4m_{5,2} + T_{45}m_{6,2} + T_{46}m_{7,2} + T_{47}m_{8,2})s_1)q_0 +$$

$$+ ((s_3 - T_{123}m_{1,1} + T_{23}m_{2,1} + T_3m_{3,1} - m_{9,1})s_4)q_1)(1/D_1)$$

$$k_{52} = ((-m_{14,2} + m_{15,2} + m_{16,2}) + T_5m_{6,2} + (T_5 + T_6)m_{7,2} + T_{57}m_{8,2})s_1q_0 +$$

$$+ (-m_{13,2}s_1 + (-T_{14}m_{11} + T_{24}m_{2,1} + T_{34}m_{3,1} + T_4m_{4,1} + s_3 - m_{9,1})s_4)q_1)(1/D_1)$$

$$k_{62} = ((-m_{15,2} + m_{16,2}) + T_6m_{7,2} + (T_6 + T_7)m_{8,2})s_1q_0$$

$$+ (-m_{13,2} + m_{14,2})s_1 + (s_3 - m_{9,1})s_4$$

$$+ (-T_{14}m_{1,1} + T_{24}m_{2,1} + T_{34}m_{3,1} + T_4m_{4,1})m_{5,2}$$

$$+ (-T_{15}m_{1,1} + T_{25}m_{2,1} + T_{35}m_{3,1} + T_{45}m_{4,1})(m_{6,2} + m_{7,2} + m_{8,2}))q_1)(1/D_1)$$

$$k_{72} = ((T_7m_{8,2} - m_{16,2})s_1)q_0 + (-m_{13,2} + m_{14,2} + m_{15,2})s_1 +$$

$$+ (s_3 - m_{9,1})s_4 + (-T_{14}m_{1,1} + T_{24}m_{2,1} + T_{34}m_{3,1} + T_4m_{4,1})m_{5,2} +$$

$$+ (-T_{15}m_{1,1} + T_{25}m_{2,1} + T_{35}m_{3,1} + T_{45}m_{4,1})m_{6,2} +$$

$$+ (-T_{16}m_{1,1} + T_{26}m_{2,1} + T_{36}m_{3,1} + T_{46}m_{4,1})(m_{7,2} + m_{8,2}))q_1)(1/D_1).$$

## A.4 Efficient Evaluation of Polynomial Functions

Given a polynomial of degree  $n$ , represented in a standard form as

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

the so-called *Horner method* provides an efficient technique for its evaluation in a point  $x_0$ , requiring only  $n$  multiplications and  $n$  additions. This technique is based on the observation that a polynomial can be written in a “nested multiplication form” as

$$p(x) = ((\dots ((a_n x + a_{n-1}) x + a_{n-2}) \dots) x + a_1) x + a_0.$$

Therefore, the evaluation of  $p(x)$  can be performed recursively by assuming  $b_n = a_n$  and performing for each  $a_k$  the following computation

```
for k = n - 1 : -1 : 0 do
     $b_k = a_k + x_0 b_{k+1}$ 
end loop (k)
```

Then  $p(x_0) = b_0$ . With the same technique, it is possible to compute the value of the derivative of the polynomial  $\dot{p}(x)$  in  $x_0$ . By assuming  $c_n = b_n$ , the evaluation of  $\dot{p}(x_0)$  proceeds as follows

```
for k = n - 1 : -1 : 1 do
     $c_k = b_k + x_0 c_{k+1}$ 
end loop (k)
```

and finally  $\dot{p}(x_0) = c_1$ .

## A.5 Numerical Solution of Tridiagonal Systems

### A.5.1 Tridiagonal systems

The solution of algebraic linear systems, such as

$$\mathbf{A} \mathbf{x} = \mathbf{d}$$

is computationally simple and efficient when the system matrix  $\mathbf{A}$  results tridiagonal. In this case, the system has the form

$$\begin{bmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & 0 & \vdots \\ 0 & a_3 & b_3 & \ddots & \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}$$

and the solution can be obtained in  $\mathcal{O}(n)$  operations by applying the so-called *Thomas algorithm* [97].

The algorithm overwrites the original arrays (if this is not desirable, it is convenient to copy the original arrays) and is based on two steps:

### 1. Forward Elimination

for  $k = 2 : 1 : n$  do

$$m = \frac{a_k}{b_{k-1}}$$

$$b_k = b_k - m c_{k-1}$$

$$d_k = d_k - m d_{k-1}$$

end loop ( $k$ )

### 2. Backward substitution

$$x_n = \frac{d_n}{b_n}$$

for  $k = n - 1 : -1 : 1$  do

$$x_k = \frac{d_k - c_k x_{k+1}}{b_k}$$

end loop ( $k$ )

This method cannot be used if  $b_1 = 0$ . In this case, it is possible to eliminate the variable  $x_2 = \frac{d_1}{c_1}$  and then solve the resulting system of  $n - 1$  unknowns, which is still a tridiagonal system.

### A.5.2 Cyclic tridiagonal systems

When the linear system has a cyclic form, namely

$$\begin{bmatrix} b_1 & c_1 & 0 & \cdots & 0 & a_1 \\ a_2 & b_2 & c_2 & 0 & & 0 \\ 0 & a_3 & b_3 & \ddots & \vdots & \\ \vdots & 0 & \ddots & \ddots & 0 & \\ 0 & \ddots & & & c_{n-1} & \\ c_n & 0 & \cdots & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \quad (\text{A.1})$$

the solution can be found by exploiting the algorithm for standard tridiagonal systems, by applying the *Sherman-Morrison formula*, [98]. In particular, the system (A.1) can be rewritten as

$$(\bar{\mathbf{A}} + \mathbf{u} \mathbf{v}^T) \mathbf{x} = \mathbf{d} \quad (\text{A.2})$$

where

$$\bar{\mathbf{A}} = \begin{bmatrix} 0 & c_1 & 0 & \cdots & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & & 0 \\ 0 & a_3 & b_3 & & \ddots & \vdots \\ \vdots & 0 & & \ddots & & 0 \\ 0 & & \ddots & & & c_{n-1} \\ 0 & 0 & \cdots & 0 & a_n & \left(b_n - \frac{a_1 c_n}{b_1}\right) \end{bmatrix}$$

and

$$\mathbf{u}^T = [b_1 \ 0 \ 0 \ \dots \ 0 \ c_n], \quad \mathbf{v}^T = [1 \ 0 \ 0 \ \dots \ 0 \ a_1/b_1].$$

At this point, it is sufficient to solve the two following auxiliary problems (as detailed in the previous section):

$$\begin{aligned} \bar{\mathbf{A}} \mathbf{y} &= \mathbf{d} \\ \bar{\mathbf{A}} \mathbf{q} &= \mathbf{u}. \end{aligned}$$

The solution of (A.2) can be found as

$$\mathbf{x} = \mathbf{y} - \frac{\mathbf{v}^T \mathbf{y}}{1 + (\mathbf{v}^T \mathbf{q})} \ \mathbf{q}.$$

# B

---

## B-spline, Nurbs and Bézier curves

### B.1 B-spline Functions

Splines are piecewise polynomial functions widely used to interpolate sets of data points or to approximate functions, curves and surfaces. On this topic a wide literature exists, see among many others the excellent books [38, 99, 100]. A particularly efficient technique for the computation of splines is based on so called *B-splines*, or *Basic-splines*. The reason for such a name is that a generic spline can be obtained as a linear combination of a proper number of basis functions, the B-splines ( $B_j^p(u)$ ), i.e.

$$\mathbf{s}(u) = \sum_{j=0}^m \mathbf{p}_j B_j^p(u), \quad u_{min} \leq u \leq u_{max}$$

where the coefficient  $\mathbf{p}_j$ ,  $j = 0, \dots, m$ , called control points, define the curve and can be computed by imposing approximation/interpolation conditions on the given set of data points, as explained in Chapter 8. This representation modality is called *B-form*.

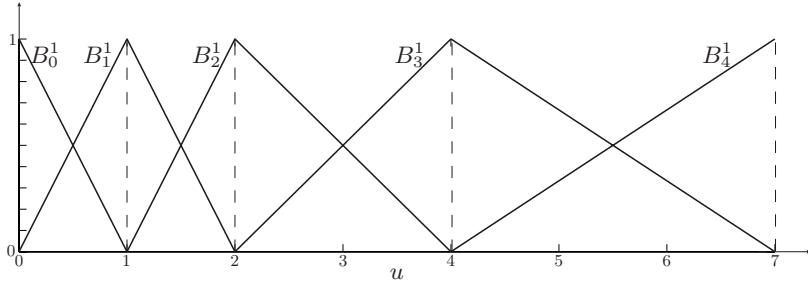
#### B.1.1 B-spline basis functions

Let  $\mathbf{u} = [u_0, \dots, u_{n_{knot}}]$  be a vector of real numbers (called *knots*), with  $u_j \leq u_{j+1}$ . The  $j$ -th *B-spline basis function* of degree  $p$  (or equivalently of order  $p + 1$ ) is defined, in a recursive manner, as

$$B_j^0(u) = \begin{cases} 1, & \text{if } u_j \leq u < u_{j+1} \\ 0, & \text{otherwise} \end{cases}$$

$$B_j^p(u) = \frac{u - u_j}{u_{j+p} - u_j} B_j^{p-1}(u) + \frac{u_{j+p+1} - u}{u_{j+p+1} - u_{j+1}} B_{j+1}^{p-1}(u), \quad p > 0.$$

Note that



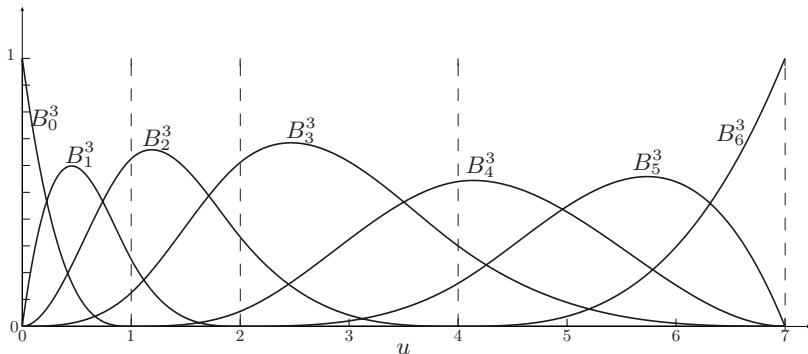
**Fig. B.1.** B-spline basis functions of degree 1 defined on  $\mathbf{u} = [0, 0, 1, 2, 4, 7, 7]$ .

1.  $B_j^p(u)$  is a piecewise polynomial, defined  $\forall u \in [u_{min}, u_{max}]$ .
2.  $B_j^p(u)$  is equal to zero everywhere except in the interval  $u \in [u_j, u_{j+p+1})$ , see Fig. B.1 and Fig. B.2.
3. The interval  $[u_i, u_{i+1})$  is called *i-th knot span*; it can be of zero length, in case knots are coincident<sup>1</sup>.
4. The B-spline basis functions are normalized so that

$$\sum_{j=0}^m B_j^p(u) = 1, \quad \forall u \in [u_0, u_{n_{knot}}] \quad (\text{partition of the unity}).$$

5. In every knot span  $[u_i, u_{i+1})$  at most  $p + 1$  basis functions  $B_j^p$  are not null, namely  $B_{i-p}^p, \dots, B_i^p$  (see Fig. B.2); this is illustrated in the graph of Fig. B.3 (*truncated triangular table*), referred to cubic functions, which shows the dependencies of the 3-rd degree basis functions on  $B_3^0$  (which is the only 0-degree term different from zero in the interval  $[u_3, u_4)$ ).

Given a value of  $u \in [u_i, u_{i+1})$ , it is possible to evaluate the basis functions with a simple and efficient algorithm. By taking into account the observation



**Fig. B.2.** Cubic basis functions defined on  $\mathbf{u} = [0, 0, 0, 0, 1, 2, 4, 7, 7, 7]$ .

<sup>1</sup> The difference between knots and breakpoints is that breakpoints are the set of *distinct* knot values.

$B_0^0$  $B_0^1$  $B_1^0$  $B_0^2$  $B_1^1$  $B_0^3$  $B_2^0$  $B_1^2$  $B_1^3$  $B_2^1$  $B_1^3$  $B_3^0$  $B_2^2$  $B_1^3$  $B_3^1$  $B_2^2$  $B_2^3$  $B_4^0$  $B_3^1$  $B_2^3$  $B_3^2$  $B_3^3$  $B_2^3$  $B_4^1$  $B_4^2$  $B_3^3$  $\vdots$  $\vdots$  $\vdots$  $\vdots$  $\vdots$ 

**Fig. B.3.** Truncated triangular table of a cubic B-spline function.

4, for each knot span it is sufficient to evaluate  $p + 1$  basis functions. The following procedure, written in C language, computes the nonvanishing basis functions at  $u$  and stores the results in  $B[j]$  ( $j = 0 \dots p + 1$ )

```
void BasisFuncs(int i, double u, int p, double U[], double B[])
/*
Input:    i    - knot span including u
          u    - value of the independent variable
          p    - degree of the spline
          U[] - Knot vector
Output:   B[] - value of the nonvanishing basis function at u
*/
{
    int j,r;
    double temp, acc;
    double DR[MAX_P], DL[MAX_P];
    B[0]=1;
    for (j=1; j<=p;j++)
    {
        DL[j] = u - U[i+1-j];
        DR[j] = U[i+j] - u;
        acc = 0.0;
        for (r=0; r<= j-1;r++)
        {
            DL[j] = DL[j] * DR[r]/(U[i+r+1]-U[i+r]);
            DR[j] = DR[j] * DR[r]/(U[i+r+1]-U[i+r]);
        }
        B[j] = acc;
    }
}
```

```

        temp = B[r]/(DR[r+1] + DL[j-r]);
        B[r] = acc + DR[r+1]*temp;
        acc = DL[j-r]*temp;
    }
    B[j] = acc;
}
}

```

where  $i$  is the index of the knot span which includes  $u$ . In particular, for a given  $i$ ,  $B[j]$  provides the value of the basis function  $B_{i-p+j}^p$ .

A further problem consists in finding  $i$ , for a given value of  $u$  and of the knot vector  $U$ . The following algorithm<sup>2</sup>, based on a binary search, provides the solution:

```

int WhichSpan(double u, double U[], int n_knot, int p)
/*
Input:   u      - value of the independent variable
         U[]    - Knot vector
         n_knot - length of U[] - 1
         p      - degree of the spline
Output:  mid    - index of the knot span including u
*/
{
    int high, low, mid;

    high = n_knot - p;
    low = p;

    if (u == U[high])
        mid = high;
    else
    {
        mid = (high+low)/2;
        while ((u<U[mid])||(u>=U[mid+1]))
        {
            if (u==U[mid+1])
                mid = mid+1; /* knot with multiplicity >1 */
            else
            {
                if (u > U[mid])
                    low = mid;
                else
                    high=mid;
            }
            mid = (high+low)/2;
        }
    }
}

```

---

<sup>2</sup> These algorithms, implemented in C language, do not work for  $u = u_{max}$ . It is therefore necessary to consider separately this case (for  $u = u_{max}$  the last basis function has a unit value, while all the other functions are null) or to avoid that this condition occurs by assuming  $u = u_{max} - \epsilon$ , where  $\epsilon$  is a small positive number.

```

        }
    }
}
return mid;
}

```

**Example B.1** Given  $p = 3$ ,  $\mathbf{u} = [0, 0, 0, 0, 1, 2, 4, 7, 7, 7, 7]$  and  $u = 1.5$ , then  $u \in [u_4, u_5]$ ,  $i = 4$ , and the nonvanishing basis functions are

$$B_1^3 = 0.0313, \quad B_2^3 = 0.5885, \quad B_3^3 = 0.3733, \quad B_4^3 = 0.0069.$$

The shape of all the basis functions defined on  $\mathbf{u}$  is shown in Fig. B.2.  $\square$

### B.1.2 Definition and properties of B-splines

Given the B-spline basis functions defined on the nonuniform knot vector (of size  $n_{knot}$ )

$$\mathbf{u} = [\underbrace{u_{min}, \dots, u_{min}}_{p+1}, \underbrace{u_{p+1}, \dots, u_{n_{knot}-p-1}}, \underbrace{u_{max}, \dots, u_{max}}_{p+1}] \quad (\text{B.1})$$

a  $p$  degree *B-spline curve* is defined as

$$\mathbf{s}(u) = \sum_{j=0}^m \mathbf{p}_j B_j^p(u), \quad u_{min} \leq u \leq u_{max} \quad (\text{B.2})$$

where  $\mathbf{p}_j$ ,  $j = 0, \dots, m$  are the *control points*, and form the so-called *control polygon*. Therefore, to represent a spline curve in the *B-form*, it is necessary to provide:

1. The integer  $p$ , defining the degree of the spline.
2. The vector of the knots  $\mathbf{u}$ .
3. The coefficients (control points)  $\mathbf{P} = [\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{m-1}, \mathbf{p}_m]$  of  $\mathbf{s}(u)$ .

**Example B.2** Fig. B.4 shows a cubic B-spline curve ( $p = 3$ ), together with its control polygon defined by<sup>3</sup>

$$\begin{aligned} \mathbf{P} &= [\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{m-1}, \mathbf{p}_m] \\ &= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & -3 & 4 & 5 & -5 & -6 \end{bmatrix}. \end{aligned}$$

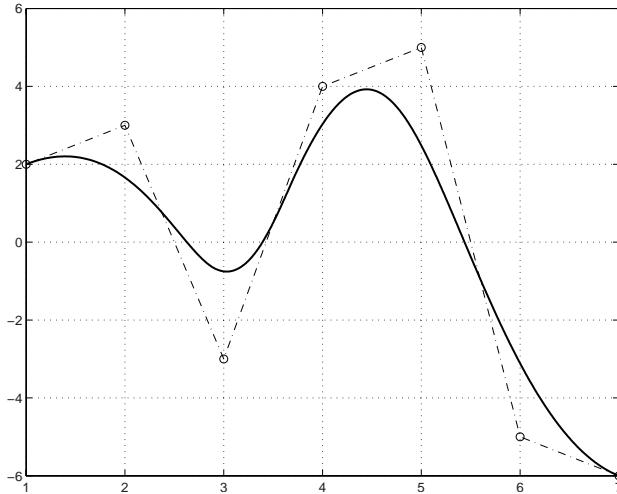
The knot vector is

$$\mathbf{u} = [0, 0, 0, 0, 1, 2, 4, 7, 7, 7, 7].$$

$\square$

---

<sup>3</sup> In the procedure `BSplinePoint`, reported below and used to evaluate the B-spline for a given value of  $u$ , the vector  $\mathbf{P}$  is defined as  $\mathbf{P} = \{1, 2, 3, 4, 5, 6, 7, 2, 3, -3, 4, 5, -5, -6\}$ .



**Fig. B.4.** Cubic B-spline and its control polygon.

The main properties of a B-spline, useful for generation of trajectories, are:

1. The degree  $p$  of the spline, the number  $m + 1$  of control points, and the number  $n_{knot} + 1$  of knots are related by  $n_{knot} = m + p + 1$ .
2.  $s(u)$  is differentiable infinite times in the interior of the knot intervals, and it is  $p - k$  times continuously differentiable at a knot of multiplicity  $k$ , e.g. a cubic B-spline ( $p = 3$ ) has two continuous derivatives at a knot with multiplicity one.

**Example B.3** Fig. B.5 shows a cubic B-spline curve ( $p = 3$ ), passing through the control points of Example B.2:

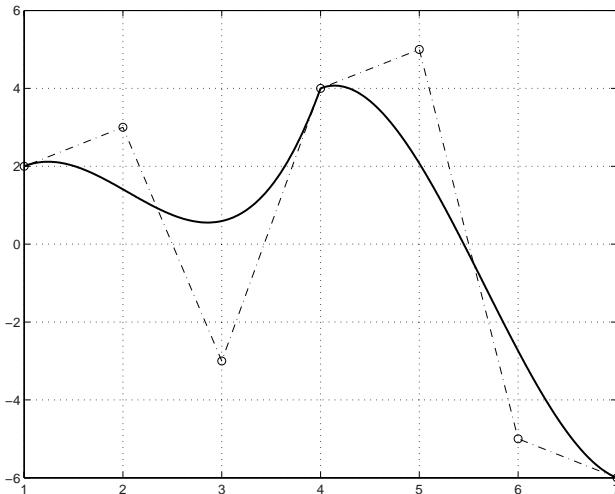
$$\begin{aligned}\mathbf{P} &= [\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{m-1}, \mathbf{p}_m] \\ &= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & -3 & 4 & 5 & -5 & -6 \end{bmatrix}.\end{aligned}$$

In this case the knot vector has a multiple element at its interior, i.e.

$$\mathbf{u} = [0, 0, 0, 0, 2, 2, 2, 7, 7, 7, 7].$$

As a consequence the spline curve is characterized by a discontinuity.  $\square$

3. Endpoints interpolation:  $s(u_{min}) = \mathbf{p}_0$  and  $s(u_{max}) = \mathbf{p}_m$ .
4. The curve is invariant under affine transformations, i.e. translations, rotations, scalings, shears, and they can be applied to  $s(u)$  by applying them to the control points.
5. Local modifications: the change of a control point  $\mathbf{p}_j$  modifies only  $s(u)$  in the interval  $[u_j, u_{j+p+1}]$ .



**Fig. B.5.** Cubic B-spline with a multiple knots in the interior of the knot vector.

6. A B-spline curve can be scaled in time by applying a transformation on the knots; in particular by assuming  $\mathbf{u}' = \lambda \mathbf{u}$ , the  $i$ -th derivative of the B-spline is scaled by a factor  $1/\lambda^i$ .

**Example B.4** The components of the first and second derivatives of the B-spline curves defined by the control points of previous example, and computed for

$$\mathbf{u} = [0, 0, 0, 0, 1, 2, 4, 7, 7, 7, 7]$$

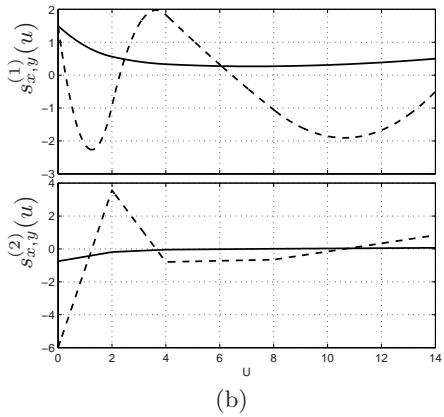
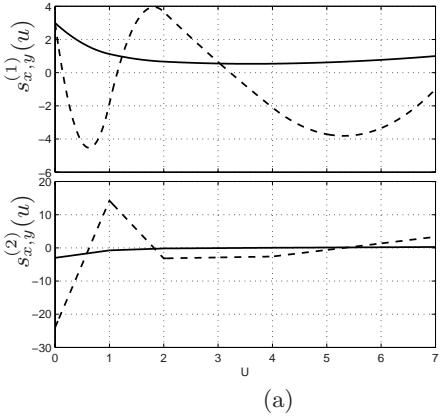
and  $\mathbf{u}' = 2\mathbf{u}$  are reported in Fig. B.6. In the latter case the velocity is half the velocity produced by the original spline, while the acceleration is four times smaller. □

7. The control polygon represents a piecewise linear approximation of the curve. In general, the lower the degree, the closer the curve follows the control polygon, see Example B.5.

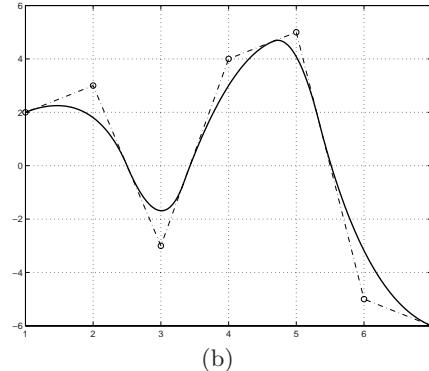
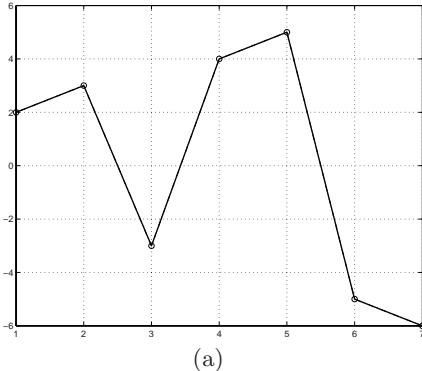
**Example B.5** The splines of degree 1 and degree 2, computed with the control points of Example B.2, namely

$$\begin{aligned}\mathbf{P} &= [\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{m-1}, \mathbf{p}_m] \\ &= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & -3 & 4 & 5 & -5 & -6 \end{bmatrix}\end{aligned}$$

are reported in Fig. B.7. Note the different approximation of the control polygon produced by the curve according to its degree. In particular, for  $p = 1$  the spline overlaps its control polygon. □



**Fig. B.6.** Components of the first and second derivatives ( $x$  solid,  $y$  dashed) of the B-spline curves computed for  $u$  (a) and  $u' = 2u$  (b).



**Fig. B.7.** B-spline curves of degree 1 (a) and 2 (b) computed with the same control points.

### B.1.3 Evaluation of a B-spline curve

For a fixed value  $\bar{u}$  of the independent variable, a spline of degree  $p$  can be computed by considering only the  $p + 1$  basis functions which are not null in the  $i$ -th span interval, which includes  $\bar{u}$ :

$$s(\bar{u}) = \sum_{j=i-p}^i p_j B_j^p(\bar{u}). \quad (\text{B.3})$$

Therefore, the value of  $s(\bar{u})$  can be calculated according to the following three steps:

1. Find the index  $i$  of the knot interval to which  $\bar{u}$  belongs (by means of the function `WhichSpan(u,U,p)`).

2. Given  $i$ , compute the basis functions at  $\bar{u}$  by means of the function `BasisFuncs(i,u,p,U)`.
3. Compute  $s(\bar{u})$ , by equation (B.3), which can be rewritten as

$$s(\bar{u}) = \sum_{j=0}^p p_{i-p+j} B_{i-p+j}^p(\bar{u}). \quad (\text{B.4})$$

The algorithm for the computation of a spline function for a given value of  $u$  is

```

void BSplinePoint( double u, double U[], int n_knot, int p,
                    double P[], int d, double s[])
{
    /*
Inputs:u      - value of the independent variable
       U[]     - Knot vector
       n_knot - length of U[] -1
       p       - degree of the spline
       P[]     - Control point vector
       d       - dimensions of a control point (2 in 2D, 3 in 3D, etc.)
Output:s[]    - value of the B-spline at u
    */

    {
        double B[MAX_P];
        int i, k, j;

        i= WhichSpan(u, U, n_knot, p);
        BasisFuncs(i, u, p, U, B);

        for (k = 0; k<d; k++) /* For each components of the B-spline*/
        {
            s[k] = 0;
            for (j = 0; j<=p; j++)
            {
                s[k] = s[k] + P[k*(n_knot-p) + i-p+j]*B[j];
            }
        }
    }
}

```

It is worth noticing that the above algorithm implements the equation (B.4).

#### B.1.4 Derivative of a B-spline curve

The derivative of a B-spline curve  $s(u)$  defined on the nonuniform knot vector

$$\mathbf{u} = [\underbrace{u_{min}, \dots, u_{min}}_{p+1}, \underbrace{u_{p+1}, \dots, u_{n_{knot}-p-1}, u_{max}, \dots, u_{max}}_{p+1}] \quad (\text{B.5})$$

can be obtained by differentiating the basis functions  $B_j^p(u)$  in (B.2):

$$\mathbf{s}^{(1)}(u) = \sum_{j=0}^m \mathbf{p}_j B_j^{p(1)}(u), \quad u_{min} \leq u \leq u_{max}. \quad (\text{B.6})$$

Since the derivative of a basis function element is

$$B_j^{p(1)}(u) = \frac{p}{u_{j+p} - u_j} B_j^{p-1}(u) - \frac{p}{u_{j+p+1} - u_{j+1}} B_{j+1}^{p-1}(u) \quad (\text{B.7})$$

the derivative of the overall curve is a spline defined on

$$\mathbf{u}' = [\underbrace{u_{min}, \dots, u_{min}}_p, u_{p+1}, \dots, u_{n_{knot}-p-1}, \underbrace{u_{max}, \dots, u_{max}}_p] \quad (\text{B.8})$$

by

$$\mathbf{s}^{(1)}(u) = \sum_{j=0}^{m-1} \mathbf{q}_j B_j^{p-1}(u), \quad u_{min} \leq u \leq u_{max} \quad (\text{B.9})$$

where the new control points  $\mathbf{q}_j$  are computed as

$$\mathbf{q}_j = p \frac{\mathbf{p}_{j+1} - \mathbf{p}_j}{u_{j+p+1} - u_{j+1}}. \quad (\text{B.10})$$

In many cases, it is necessary to compute the generic  $k$ -th derivative of the B-spline function  $\mathbf{s}(u)$ , i.e.

$$\mathbf{s}^{(k)}(u) = \sum_{j=0}^m \mathbf{p}_j B_j^{p(k)}(u), \quad u_{min} \leq u \leq u_{max}.$$

This can be obtained by computing the functions  $B_j^{p(k)}(u)$ . An efficient algorithm for the computation of the  $k$ -th derivative of  $B_j^p(u)$ , in terms of the basis functions  $B_j^{p-k}(u), \dots, B_{j+k}^{p-k}(u)$  defined on  $\mathbf{u}$ , is

$$B_j^{p(k)}(u) = \frac{p!}{(p-k)!} \sum_{i=0}^k a_{k,i} B_{j+i}^{p-k} \quad (\text{B.11})$$

with

$$\begin{aligned} a_{0,0} &= 1 \\ a_{k,0} &= \frac{a_{k-1,0}}{u_{j+p-k+1} - u_j} \\ a_{k,i} &= \frac{a_{k-1,i} - a_{k-1,i-1}}{u_{j+p+i-k+1} - u_{j+i}}, \quad i = 1, \dots, k-1 \\ a_{k,k} &= \frac{-a_{k-1,k-1}}{u_{j+p+1} - u_{j+k}}. \end{aligned}$$

Note that  $k$  cannot exceed  $p$  (all the derivatives of order  $k > p$  are null). Moreover, it is possible that the denominators involving knots differences become zero; in this case the quotient is defined to be zero.

The algorithm (B.11) for the computation of the basis functions and their derivatives up to the order  $n$  can be implemented as a routine C in the following way [38].

```

typedef double Matrix[MAX_P+1][MAX_P+1];

void DersBasisFuns(double u, int j, int p, int n, double U[], 
                    Matrix Ders)
{
/*
Inputs: u      - value of the independent variable
        j      - index of the knot span, which includes u
        p      - degree of the spline
        n      - max degree of differentiation of B-spline
                basis functions
        U[]    - Knot vector
Output: Ders[][] - values of B-spline basis functions and theirs
derivatives at u
*/
    double DR[MAX_P], DL[MAX_P];
    Matrix Du, a;
    double acc, temp, d;
    int i, r, k, s1, s2, rk, pk, i1, i2;

    Du[0][0] = 1.0;
    for (j=1; j<=p; j++)
    {
        DL[j] = u - U[i+1-j];
        DR[j] = U[i+j]-u;
        acc = 0.0;
        for (r=0;r<j;r++)
        {
            Du[j][r] = DR[r+1] + DL[j-r];
            temp = Du[r][j-1] / Du[j][r];
            Du[r][j] = acc + DR[r+1] * temp;
            acc = DL[j-r] * temp;
        }
        Du[j][j] = acc;
    }

    for (j=0; j<=p; j++)
        Ders[0][j] = Du[j][p];
}

```

```

for (r=0; r<=p; r++)
{
    s1=0;
    s2=1;
    a[0][0] = 1.0;
    for (k=1; k<=n; k++)
    {
        d = 0.0;
        rk = r - k;
        pk = p - k;
        if (r >= k)
        {
            a[s2][0] = a[s1][0] / Du[pk+1][rk];
            d = a[s2][0] * Du[rk][pk];
        }
        if (rk >= -1)
            j1 = 1;
        else
            j1 = -rk;
        if (r-1 <= pk)
            j2 = k - 1;
        else
            j2 = p - r;
        for (j=j1; j<=j2; j++)
        {
            a[s2][j] = (a[s1][j] - a[s1][j-1]) / Du[pk+1][rk+j];
            d += a[s2][j] * Du[rk+j][pk];
        }
        if (r <= pk)
        {
            a[s2][k] = -a[s1][k-1] / Du[pk+1][r];
            d += a[s2][k] * Du[r][pk];
        }
        Ders[k][r] = d;
        j = s1; s1 = s2; s2 = j;
    }
}
r = p;
for (k=1; k<=n; k++)
{
    for (j=0; j<=p; j++) Ders[k][j] *= r;
    r *= (p-k);
}
}

```

The array  $\text{Ders}[\cdot][\cdot]$  contains the nonzero basis functions and their derivatives. In particular,  $\text{Ders}[k][j]$  provides the values of  $B_{i-p+j}^p(u)$ ,  $j = 0, \dots, p$ , computed for  $u \in [u_j, u_{j+1}]$ . The other elements of  $B_{i-p+j}^p(u)$  are null.

**Example B.6** The basis functions of degree three ( $p = 3$ ), and the related derivatives, defined on the knots vector  $\mathbf{u} = [0, 0, 0, 0, 1, 2, 4, 7, 7, 7, 7]$  and computed at  $\bar{u} = 4.5$  (therefore the knot span index is  $i = 6$ ), are<sup>4</sup>

$$\begin{aligned}\text{Ders}[0][0] &= 0.1736, \quad \text{Ders}[0][1] = 0.5208, \quad \text{Ders}[0][2] = 0.3009, \quad \text{Ders}[0][3] = 0.0046, \\ \text{Ders}[1][0] &= -0.2083, \quad \text{Ders}[1][1] = -0.1250, \quad \text{Ders}[1][2] = 0.3055, \quad \text{Ders}[1][3] = 0.0277, \\ \text{Ders}[2][0] &= 0.1666, \quad \text{Ders}[2][1] = -0.3000, \quad \text{Ders}[2][2] = 0.0222, \quad \text{Ders}[2][3] = 0.1111, \\ \text{Ders}[3][0] &= -0.0666, \quad \text{Ders}[3][1] = 0.2800, \quad \text{Ders}[3][2] = -0.4355, \quad \text{Ders}[3][3] = 0.2222.\end{aligned}$$

which correspond to

$$\begin{aligned}B_3^3 &= 0.1736, \quad B_4^3 = 0.5208, \quad B_5^3 = 0.3009, \quad B_6^3 = 0.0046, \\ B_3^{3(1)} &= -0.2083, \quad B_4^{3(1)} = -0.1250, \quad B_5^{3(1)} = 0.3055, \quad B_6^{3(1)} = 0.0277, \\ B_3^{3(2)} &= 0.1666, \quad B_4^{3(2)} = -0.3000, \quad B_5^{3(2)} = 0.0222, \quad B_6^{3(2)} = 0.1111, \\ B_3^{3(3)} &= -0.0666, \quad B_4^{3(3)} = 0.2800, \quad B_5^{3(3)} = -0.4355, \quad B_6^{3(3)} = 0.2222.\end{aligned}$$

All the other terms  $B_j^{3(k)}$  are null.

□

### B.1.5 Conversion from B-form to Piecewise Polynomial form (pp-form)

The representation of a spline as a piecewise polynomial (the so-called *pp-form*) can be useful to perform the pointwise evaluation of the curve for a given  $\bar{u}$ . The definition of a piecewise polynomial spline of degree  $p$  consists of:

1. A strictly increasing sequence of points  $\mathbf{u}^* = [u_0^*, \dots, u_m^*]$ , the so-called *breakpoints* (in this case multiple points are not admitted).
2. The  $p+1$  coefficients of each polynomial  $\mathbf{p}_j(u) = \mathbf{a}_{0,j} + \mathbf{a}_{1,j}u + \dots + \mathbf{a}_{p,j}u^p$  forming the spline

$$s(u) = \mathbf{p}_j(u), \quad \text{if } u_j^* \leq u \leq u_{j+1}^*, \quad j = 0, \dots, m-1.$$

The conversion from B-form to pp-form is straightforward by using the procedures for the evaluation and the differentiation of a B-spline. As a matter of fact the piecewise polynomial form can be found according to the following two steps:

1. Find the vector of the breakpoints  $\mathbf{u}^*$  from the knot vector  $\mathbf{u}$ , by checking whether each knot has a multiplicity greater than one.

---

<sup>4</sup> In this case, `MAX_P=6`.

2. Compute the coefficients  $\alpha_{k,j}$ ,  $k = 0 \dots p$ , of the  $j$ -th polynomial function by evaluating the spline and its derivatives at the breakpoint  $u_j^*$ ; the expression of the coefficients is

$$\alpha_{k,j} = \frac{\frac{d^k s(u)}{du^k}|_{u=u_j^*}}{k!}.$$

**Example B.7** The pp-form of the B-spline reported in example B.2 is a spline of degree 3, with 5 distinct breakpoints<sup>5</sup>

$$\mathbf{u}^* = [0, 1, 2, 4, 7]$$

whose coefficients<sup>6</sup> are

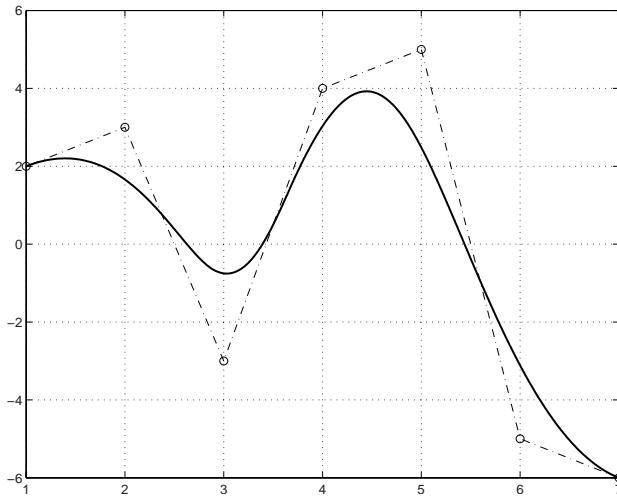
$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} a_{p,0}^x & \cdots & a_{1,0}^x & a_{0,0}^x \\ a_{p,0}^y & \cdots & a_{1,0}^y & a_{0,0}^y \\ \hline a_{p,1}^x & \cdots & a_{1,1}^x & a_{0,1}^x \\ a_{p,1}^y & \cdots & a_{1,1}^y & a_{0,1}^y \\ \hline \vdots & & & \\ \hline a_{p,j}^x & \cdots & a_{1,j}^x & a_{0,j}^x \\ a_{p,j}^y & \cdots & a_{1,j}^y & a_{0,j}^y \end{bmatrix} \\ &= \begin{bmatrix} 0.37 & -1.50 & 3.00 & 1.00 \\ 6.37 & -12.00 & 3.00 & 2.00 \\ \hline 0.09 & -0.37 & 1.12 & 2.87 \\ -2.90 & 7.12 & -1.87 & -0.62 \\ \hline 0.01 & -0.08 & 0.66 & 3.72 \\ 0.04 & -1.58 & 3.66 & 1.72 \\ \hline 0.01 & -0.02 & 0.54 & 4.86 \\ 0.32 & -1.30 & -2.10 & 3.10 \end{bmatrix}. \end{aligned}$$

Note that the spline has dimension 2, and for this reason the coefficients are denoted with superscripts  $x$  and  $y$ .  $\square$

---

<sup>5</sup> As a consequence j=4.

<sup>6</sup> The representation adopted in Matlab is used.



**Fig. B.8.** A cubic B-spline and its control polygon.

## B.2 Definition and Properties of Nurbs

A Non Uniform Rational B-Spline (Nurbs) curve of degree  $p$  is defined as

$$\mathbf{n}(u) = \frac{\sum_{j=0}^m \mathbf{p}_j w_j B_j^p(u)}{\sum_{j=0}^m w_j B_j^p(u)}, \quad u_{min} \leq u \leq u_{max} \quad (\text{B.12})$$

where  $\mathbf{p}_j$ ,  $j = 0, \dots, m$  are the *control points*, forming the so-called *control polygon*,  $w_j$  are proper *weights* and  $B_j^p(u)$  are the B-spline basis functions of degree  $p$  defined on the nonuniform knot vector (of size  $n_{knot} + 1$ )

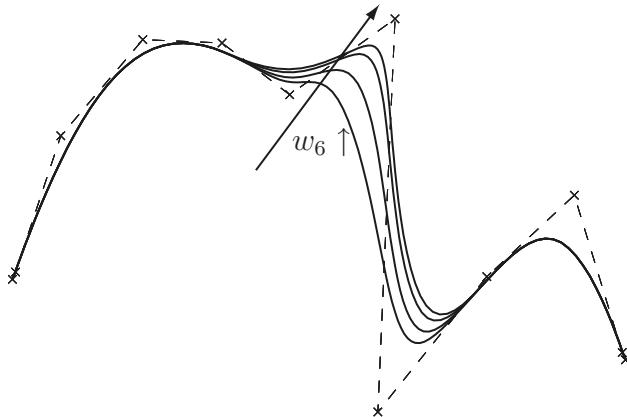
$$\mathbf{u} = [\underbrace{u_{min}, \dots, u_{min}}_{p+1}, u_{p+1}, \dots, u_{n_{knot}-p-1}, \underbrace{u_{max}, \dots, u_{max}}_{p+1}]. \quad (\text{B.13})$$

By setting

$$N_j^p(u) = \frac{w_j B_j^p(u)}{\sum_{i=0}^m w_i B_i^p(u)}, \quad u_{min} \leq u \leq u_{max} \quad (\text{B.14})$$

it is possible to rewrite (B.12) in the form

$$\mathbf{n}(u) = \sum_{j=0}^m \mathbf{p}_j N_j^p(u), \quad u_{min} \leq u \leq u_{max}. \quad (\text{B.15})$$



**Fig. B.9.** Local modification of a Nurbs trajectory by acting on a weight.

The  $N_j^p(u)$  are piecewise rational functions, called the *rational basis functions*. Note that if the weights are constant and equal, i.e.  $w_j = \bar{w} \neq 0, \forall j$ , then<sup>7</sup>  $N_j^p(u) = B_j^p(u)$ ; therefore, B-splines are a special case of Nurbs curves. All the properties stated for the B-spline hold for Nurbs:

1. Endpoints interpolation:  $\mathbf{n}(u_{min}) = \mathbf{p}_0$  and  $\mathbf{n}(u_{max}) = \mathbf{p}_m$ .
2. The curve is invariant under affine transformations, i.e. translations, rotations, scalings, shears, that can be applied to  $\mathbf{n}(u)$  by applying them to the control points.
3. Local modifications: the change of a control point  $\mathbf{p}_j$  or of a weight  $w_j$  modifies  $\mathbf{n}(u)$  only in the interval  $[u_j, u_{j+p+1}]$ , see Fig. B.9.
4. The control polygon represents a piecewise linear approximation of the curve. In general, the lower the degree of the Nurbs, the closer the curve follows the control polygon.

An efficient way to represent (and evaluate) Nurbs curves is based on homogeneous coordinates. In the three-dimensional case, for a given set of control points  $\mathbf{p}_j = [p_{x,j}, p_{y,j}, p_{z,j}]^T$  and weights  $w_j$ , it is possible to construct the weighted control points  $\mathbf{p}_j^w = [w_j p_{x,j}, w_j p_{y,j}, w_j p_{z,j}, w_j]^T \in \mathbb{R}^4$ , which define the *nonrational* B-spline

---

<sup>7</sup> In this case, (B.14) becomes

$$N_j^p(u) = \frac{\bar{w} B_j^p(u)}{\bar{w} \sum_{i=0}^m B_i^p(u)}, \quad u_{min} \leq u \leq u_{max}$$

and, by considering that  $\sum_{j=0}^m B_j^p(u) = 1$ , it follows  $N_j^p(u) = B_j^p(u)$ .

$$\mathbf{n}^w(u) = \sum_{j=0}^m \mathbf{p}_j^w B_j^p(u), \quad u_{min} \leq u \leq u_{max}. \quad (\text{B.16})$$

The representations (B.15) and (B.16) are equivalent in the sense that they are related by means of a bijective map. By applying the perspective map

$$\mathbf{p} = H(\mathbf{p}^w) = \begin{cases} \left[ \frac{p_x^w}{w}, \frac{p_y^w}{w}, \frac{p_z^w}{w} \right]^T, & \text{if } w \neq 0 \\ \text{direction of } [p_x^w, p_y^w, p_z^w]^T, & \text{if } w = 0 \end{cases} \quad (\text{B.17})$$

it is possible to transform  $\mathbf{n}^w(u)$  in the corresponding rational B-spline

$$\mathbf{n}(u) = H(\mathbf{n}^w(u)) = H \left( \sum_{j=0}^m \mathbf{p}_j^w B_j^p(u) \right).$$

In particular, the expression (B.16) is used to evaluate the Nurbs curve for a given value of the variable  $u$ , by exploiting the algorithm reported in Sec. B.1.3

$$\mathbf{n}(u) \xrightarrow{H^{-1}} \mathbf{n}^w(u) \xrightarrow{\text{Alg.(B.1.3)}} \mathbf{n}^w(\bar{u}) = \begin{bmatrix} n_x^w \\ n_y^w \\ n_z^w \\ w \end{bmatrix} \xrightarrow{H} \mathbf{n}(\bar{u}) = \begin{bmatrix} \frac{n_x^w}{w} \\ \frac{n_y^w}{w} \\ \frac{n_z^w}{w} \\ \frac{n_x^w}{w} \end{bmatrix}.$$

### B.3 Definition and Properties of Bézier Curves

A Bézier curve of degree  $m$  is defined as

$$\mathbf{b}(u) = \sum_{j=0}^m B_j^m(u) \mathbf{p}_j, \quad 0 \leq u \leq 1 \quad (\text{B.18})$$

where the coefficients  $\mathbf{p}_j$  are the *control points*, and the basis functions  $B_j^m(u)$  are  $m$ -th degree *Bernstein polynomials* defined by

$$B_j^m(u) = \binom{m}{j} u^j (1-u)^{m-j} \quad (\text{B.19})$$

with the binomial coefficients given by

$$\binom{m}{j} = \frac{m!}{j! (m-j)!}.$$

The binomial coefficients, for  $j = 0, \dots, m$ , form the rows of Pascal's triangle shown in Tab. B.1 for  $m = 0, \dots, 9$ .

$m$										
0	1									
1	1	1								
2	1	2	1							
3	1	3	3	1						
4	1	4	6	4	1					
5	1	5	10	10	5	1				
6	1	6	15	20	15	6	1			
7	1	7	21	35	35	21	7	1		
8	1	8	28	56	70	56	28	8	1	
9	1	9	36	84	126	126	84	36	9	1
										...

**Table B.1.** Pascal's triangle.

**Example B.8** For  $m = 1$ , the basis functions are  $B_0^1(u) = 1 - u$  and  $B_1^1(u) = u$ , therefore the curve is  $\mathbf{b}(u) = (1 - u)\mathbf{p}_0 + u\mathbf{p}_1$ , that is a straight line from  $\mathbf{p}_0$  to  $\mathbf{p}_1$ .  $\square$

**Example B.9** For  $m = 3$ , the Bézier curve has the form

$$\mathbf{b}(u) = (1 - u)^3 \mathbf{p}_0 + 3u(1 - u)^2 \mathbf{p}_1 + 3u^2(1 - u) \mathbf{p}_2 + u^3 \mathbf{p}_3.$$

A 2-dimensional curve obtained with

$$\mathbf{p}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{p}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} 1 \\ 2.5 \end{pmatrix}, \quad \mathbf{p}_3 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

is illustrated in Fig. B.10.  $\square$

From the previous example, it results that:

- The *control polygon* (formed by the control points) approximates the shape of the curve;
- $\mathbf{p}_0 = \mathbf{b}(0)$  and  $\mathbf{p}_m = \mathbf{b}(1)$ .
- The tangent directions in  $\mathbf{p}_0$  and  $\mathbf{p}_m$  are parallel to  $\mathbf{p}_1 - \mathbf{p}_0$  and  $\mathbf{p}_m - \mathbf{p}_{m-1}$ .
- The curve is completely contained in the convex hull formed by its control points.

### B.3.1 Evaluation of a Bézier curve

The value of a Bézier curve for a given  $\bar{u}$  can be computed, according to the *de Casteljau algorithm*, as  $\mathbf{b}(\bar{u}) = \mathbf{p}_j^m$  with  $\mathbf{p}_j^m$  defined in a recursive way:

$$\mathbf{p}_j^k(\bar{u}) = \begin{cases} (1 - \bar{u}) \mathbf{p}_j^{k-1}(\bar{u}) + \bar{u} \mathbf{p}_{j+1}^{k-1}(\bar{u}) & \text{if } \begin{cases} k = 1, \dots, m \\ j = 0, \dots, m-k \end{cases} \\ \mathbf{p}_j & \text{if } \begin{cases} k = 0 \\ j = 0, \dots, m. \end{cases} \end{cases} \quad (\text{B.20})$$

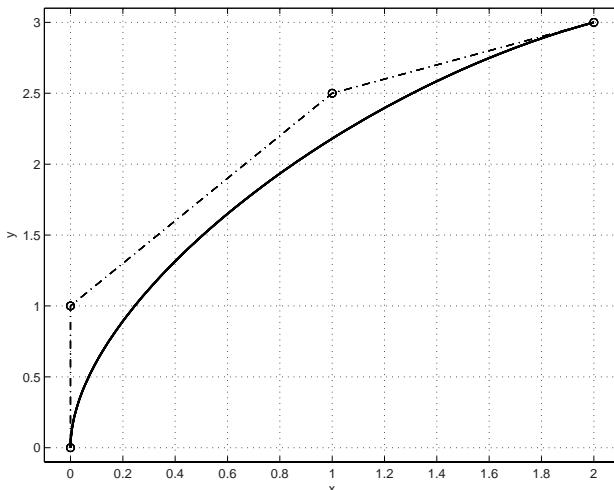
This algorithm, although less efficient from a computational point of view with respect to classical polynomial evaluation methods (e.g. the Horner formula, reported in Sec. A.4), is affected by smaller round-off errors. The procedure for the computation of the Bézier curve  $\mathbf{b}_u$  of degree  $m$  defined by the control points  $\mathbf{P}[]$  (for the sake of simplicity a one-dimensional case is considered) for a given value is

```
void DeCasteljau(int m, double u, double P[], double b_u[])
{
    int i, k;
    double Q[MAX_M];

    for (i=0; i<=m; i++)
        Q[i]=P[i];

    for (k=1; k<=m; k++)
        for (i=0; i<=m-k; i++)
            Q[i] = (1.0-u)*Q[i]+u*Q[i+1];

    b_u[0] = Q[0];
}
```



**Fig. B.10.** 3 – rd degree Bézier curve, in the 2-dimensional space.

Obviously, in order to evaluate the curve for a given value  $u$  it is always possible to transform a curve from the Bézier form to a standard polynomial form. For the trajectories computation, the curves commonly adopted are of degree 3, 4, and 5. In these cases, the curve can be represented as

$$\mathbf{b}(u) = \sum_{i=0}^m \mathbf{a}_i u^i \quad (\text{B.21})$$

where the coefficients  $\mathbf{a}_j$  are

$$\begin{aligned} \text{for } m = 3 & \left\{ \begin{array}{l} \mathbf{a}_0 = \mathbf{p}_0 \\ \mathbf{a}_1 = -3\mathbf{p}_0 + 3\mathbf{p}_1 \\ \mathbf{a}_2 = 3\mathbf{p}_0 - 6\mathbf{p}_1 + 3\mathbf{p}_2 \\ \mathbf{a}_3 = -\mathbf{p}_0 + 3\mathbf{p}_1 - 3\mathbf{p}_2 + \mathbf{p}_3 \end{array} \right. \\ \text{for } m = 4 & \left\{ \begin{array}{l} \mathbf{a}_0 = \mathbf{p}_0 \\ \mathbf{a}_1 = -4\mathbf{p}_0 + 4\mathbf{p}_1 \\ \mathbf{a}_2 = 6\mathbf{p}_0 - 12\mathbf{p}_1 + 6\mathbf{p}_2 \\ \mathbf{a}_3 = -4\mathbf{p}_0 + 12\mathbf{p}_1 - 12\mathbf{p}_2 + 4\mathbf{p}_3 \\ \mathbf{a}_4 = 5\mathbf{p}_0 - 4\mathbf{p}_1 + 6\mathbf{p}_2 - 4\mathbf{p}_3 + \mathbf{p}_4 \end{array} \right. \\ \text{for } m = 5 & \left\{ \begin{array}{l} \mathbf{a}_0 = \mathbf{p}_0 \\ \mathbf{a}_1 = -5\mathbf{p}_0 + 5\mathbf{p}_1 \\ \mathbf{a}_2 = 10\mathbf{p}_0 - 20\mathbf{p}_1 + 10\mathbf{p}_2 \\ \mathbf{a}_3 = -10\mathbf{p}_0 + 30\mathbf{p}_1 - 30\mathbf{p}_2 + 10\mathbf{p}_3 \\ \mathbf{a}_4 = 5\mathbf{p}_0 - 20\mathbf{p}_1 + 30\mathbf{p}_2 - 20\mathbf{p}_3 + 5\mathbf{p}_4 \\ \mathbf{a}_5 = -\mathbf{p}_0 + 5\mathbf{p}_1 - 10\mathbf{p}_2 + 10\mathbf{p}_3 - 5\mathbf{p}_4 + \mathbf{p}_5. \end{array} \right. \end{aligned}$$

More generally, the relationship between the control points defining a Bézier curve and the coefficients  $\mathbf{a}_i$  of the standard polynomial representation (B.21) is

$$\mathbf{a}_i = \frac{m!}{(m-i)!} \sum_{j=0}^i \frac{(-1)^{j+i}}{j!(i-j)!} \mathbf{p}_j \quad (\text{B.22})$$

where  $m$  is the degree of the Bézier curve.

### B.3.2 Derivatives of a Bézier curve

The derivative of a  $m$ -th degree Bézier curve is a Bézier curve (of degree  $m-1$ ) defined by

$$\mathbf{b}^{(1)}(u) = m \sum_{i=0}^{m-1} B_i^{m-1}(u) (\mathbf{p}_{i+1} - \mathbf{p}_i). \quad (\text{B.23})$$

From eq. (B.23), the values of the end derivatives of a Bézier curve can be easily deduced

$$\begin{aligned}\mathbf{b}^{(1)}(0) &= m(\mathbf{p}_1 - \mathbf{p}_0), & \mathbf{b}^{(2)}(0) &= m(m-1)(\mathbf{p}_0 - 2\mathbf{p}_1 + \mathbf{p}_2), \\ \mathbf{b}^{(1)}(1) &= m(\mathbf{p}_m - \mathbf{p}_{m-1}), & \mathbf{b}^{(2)}(1) &= m(m-1)(\mathbf{p}_m - 2\mathbf{p}_{m-1} + \mathbf{p}_{m-2}).\end{aligned}\tag{B.24}$$

Notice that the  $k$ -th derivative at an endpoint depends only on the  $k+1$  control points at that end.

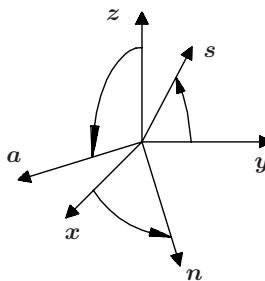
# C

---

## Representation of the Orientation

The problem of describing the orientation of a rigid body in the 3D space can be solved in different ways. This problem is of particular relevance in robotics, where in general the end effector must be both positioned and oriented in order to execute a given task.

The most common mathematical tools for describing the orientation of a rigid body are: rotation matrices, the angle-axis representation, the Euler angles, and the Roll-Pitch-Yaw angles. For a detailed description of these mathematical operators, refer to the wide literature available for example in robotics: see [12], [11], [52] among many others.



**Fig. C.1.** Generic rotation between two frames.

### C.1 Rotation Matrices

A rotation matrix  $\mathbf{R}$  is a  $3 \times 3$  positive definite matrix whose column vectors describe a reference frame  $\mathcal{F}_1$  with respect to a base frame  $\mathcal{F}_0$ , Fig. C.1. If  $\mathbf{n}, \mathbf{s}, \mathbf{a}$  are the unit vectors associated to the axes of  $\mathcal{F}_1$ , then the rotation matrix  $\mathbf{R}$  is defined as

$$\mathbf{R} = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix}.$$

Rotation matrices have several important properties:

1.  $\det\{\mathbf{R}\} = +1$ .
2.  $\mathbf{R}$  is an orthogonal matrix, i.e.

$$\mathbf{n}^T \mathbf{s} = \mathbf{n}^T \mathbf{a} = \mathbf{s}^T \mathbf{a} = 0, \quad \|\mathbf{n}\| = \|\mathbf{s}\| = \|\mathbf{a}\| = 1$$

or, in a more compact form  $\mathbf{R}^T \mathbf{R} = \mathbf{I}$  ( $3 \times 3$  identity matrix).

3. Since  $\mathbf{R}$  is orthogonal, it follows also that  $\mathbf{R}^{-1} = \mathbf{R}^T$ .

### C.1.1 Elementary rotation matrices

An *elementary rotation* is a rotation about one of the axes of the base frame  $\mathcal{F}_0$ . The rotation matrices  $\mathbf{R}_x$ ,  $\mathbf{R}_y$ , and  $\mathbf{R}_z$  expressing these rotations have the form

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (\text{C.1})$$

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (\text{C.2})$$

$$\mathbf{R}_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{C.3})$$

By properly composing these elementary rotations, it is possible to compute the rotation matrix of a generic rotation in space.

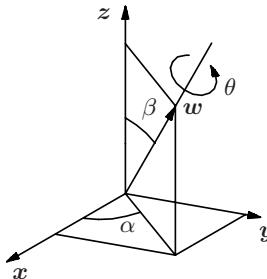
## C.2 Angle-Axis Representation

Often, one is interested in expressing a rotation of a given angle about an arbitrary axis in space.

Let us define a base frame  $\mathcal{F}_0$ , and a unit vector  $\mathbf{w} = [w_x, w_y, w_z]^T$ , see Fig. C.2. The rotation matrix  $\mathbf{R}_{\mathbf{w}}(\theta)$  specifying a rotation of an angle  $\theta$ , positive in the counter-clockwise direction, about  $\mathbf{w}$  is defined as

$$\mathbf{R}_{\mathbf{w}}(\theta) = \begin{bmatrix} w_x^2(1 - c_\theta) + c_\theta & w_x w_y(1 - c_\theta) - w_z s_\theta & w_x w_z(1 - c_\theta) + w_y s_\theta \\ w_x w_y(1 - c_\theta) + w_z s_\theta & w_y^2(1 - c_\theta) + c_\theta & w_y w_z(1 - c_\theta) - w_x s_\theta \\ w_x w_z(1 - c_\theta) - w_y s_\theta & w_y w_z(1 - c_\theta) + w_x s_\theta & w_z^2(1 - c_\theta) + c_\theta \end{bmatrix} \quad (\text{C.4})$$

where  $c_\theta = \cos(\theta)$ ,  $s_\theta = \sin(\theta)$ . Note that



**Fig. C.2.** Angle-Axis representation.

$$\mathbf{R}_{-\mathbf{w}}(-\theta) = \mathbf{R}_{\mathbf{w}}(\theta).$$

It may be of interest to solve the inverse problem, i.e. given a generic rotation matrix  $\mathbf{R}$

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (\text{C.5})$$

define the equivalent axis  $\mathbf{w}$  and angle  $\theta$ . If  $\sin(\theta) \neq 0$  ( $\theta \neq k\pi$ ), these parameters are given by

$$\theta = \cos^{-1} \left( \frac{r_{11} + r_{22} + r_{33} - 1}{2} \right) \quad (\text{C.6})$$

$$\mathbf{w} = \frac{1}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}. \quad (\text{C.7})$$

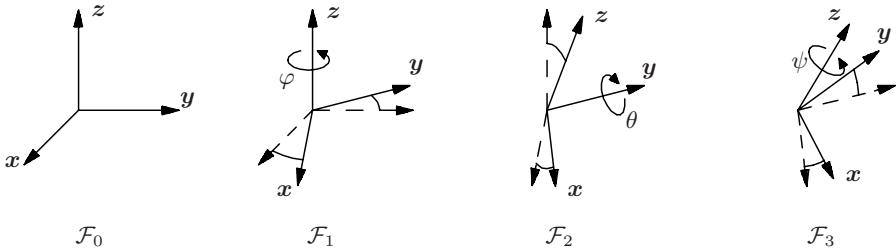
If  $\sin(\theta) = 0$ , it is necessary to analyze the particular expression assumed by  $\mathbf{R}$  and compute the equations for  $\theta = 0, \pi$ . Note that for  $\theta = 0$  the rotation axis  $\mathbf{w}$  is arbitrary.

### C.3 Euler Angles

The Euler angles, as well as the Roll-Pitch-Yaw angles, are *minimum representations* of rotations, in the sense that only three parameters (i.e. three angles  $\varphi, \theta, \psi$ ) are used to express an arbitrary rotation in the 3D space. Commonly, the ZYZ representation is chosen for the Euler angles, that is three consecutive rotations about the  $\mathbf{z}_0$  ( $\varphi$ ),  $\mathbf{y}_1$  ( $\theta$ ),  $\mathbf{z}_2$  ( $\psi$ ) axes of the *current* reference frame (the frame obtained by applying the rotations), see Fig. C.3.

The rotation matrix  $\mathbf{R}_{Euler}$  corresponding to these three rotations is defined as

$$\mathbf{R}_{Euler}(\varphi, \theta, \psi) = \begin{bmatrix} c_\varphi c_\theta c_\psi - s_\varphi s_\psi & -c_\varphi c_\theta s_\psi - s_\varphi c_\psi & c_\varphi s_\theta \\ s_\varphi c_\theta c_\psi + c_\varphi s_\psi & -s_\varphi c_\theta s_\psi + c_\varphi c_\psi & s_\varphi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}. \quad (\text{C.8})$$

**Fig. C.3.** Euler angles.

For the inverse problem, i.e given a generic rotation matrix  $\mathbf{R}$  as in eq. (C.5) define the three angles  $\varphi, \theta, \psi$ , there are two possibilities:

1.  $r_{13}^2 + r_{23}^2 \neq 0$ , then  $\sin \theta \neq 0$ . There are two sets of solutions depending on the value assigned to the sign of  $\theta$ . If  $0 < \theta < \pi$ , i.e.  $\sin \theta > 0$ , from (C.8) one obtains

$$\begin{cases} \varphi = \text{atan2}(r_{23}, r_{13}) \\ \theta = \text{atan2}(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}) \\ \psi = \text{atan2}(r_{32}, -r_{31}) \end{cases} \quad (\text{C.9})$$

or, considering  $-\pi < \theta < 0$  ( $\sin \theta < 0$ ),

$$\begin{cases} \varphi = \text{atan2}(-r_{23}, -r_{13}) \\ \theta = \text{atan2}(-\sqrt{r_{13}^2 + r_{23}^2}, r_{33}) \\ \psi = \text{atan2}(-r_{32}, r_{31}) \end{cases} \quad (\text{C.10})$$

where  $\text{atan2}(y, x)$  is the four quadrant arctangent of  $x$  and  $y$ .

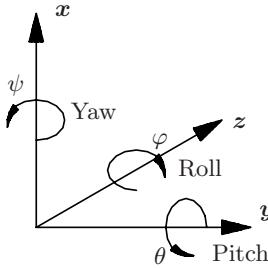
2.  $r_{13}^2 + r_{23}^2 = 0$ . In this case  $\theta = 0, \pi$  and  $\cos \theta = \pm 1$ . By choosing  $\theta = 0$ , i.e.  $\cos \theta = 1$  one obtains

$$\begin{cases} \theta = 0 \\ \varphi + \psi = \text{atan2}(r_{21}, r_{11}) = \text{atan2}(-r_{12}, r_{11}). \end{cases} \quad (\text{C.11})$$

Vice versa, if  $\theta = \pi$ , ( $\cos \theta = -1$ ) one obtains

$$\begin{cases} \theta = 0 \\ \varphi - \psi = \text{atan2}(-r_{21}, -r_{11}) = \text{atan2}(-r_{12}, -r_{11}). \end{cases} \quad (\text{C.12})$$

In both cases there are infinite solutions, since only the sum (difference) of  $\varphi$  and  $\theta$  can be determined. In fact, being  $\theta = 0, \pi$ , the rotations  $\varphi, \psi$  are about parallel axes and therefore it is not possible to distinguish between them.



**Fig. C.4.** RPY angles.

## C.4 Roll-Pitch-Yaw Angles

The Roll-Pitch-Yaw (RPY) representation indicates three consecutive rotations about the axes of the *base* reference frame  $\mathcal{F}_0$ : of an angle  $\varphi$  (Yaw) about  $x$ , of an angle  $\theta$  (Pitch) about  $y$  and of an angle  $\psi$  (Roll) about  $z$ , see Fig. C.4.

The rotation matrix  $\mathbf{R}_{RPY}$  corresponding to these three rotations is defined as

$$\mathbf{R}_{RPY}(\varphi, \theta, \psi) = \begin{bmatrix} c_\varphi c_\theta & -s_\varphi c_\psi + c_\varphi s_\theta s_\psi & s_\varphi s_\psi + c_\varphi s_\theta c_\psi \\ s_\varphi c_\theta & c_\varphi c_\psi + s_\varphi s_\theta s_\psi & -c_\varphi s_\psi + s_\varphi s_\theta c_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix} \quad (\text{C.13})$$

For the inverse problem, i.e. given a generic rotation matrix  $\mathbf{R}$  as in (C.5) define the three angles  $\varphi, \theta, \psi$ , there are two possibilities:

1.  $r_{11}^2 + r_{21}^2 \neq 0 \rightarrow \cos \theta \neq 0$ : one obtains

$$\begin{cases} \varphi = \text{atan2}(r_{21}, r_{11}) \\ \theta = \text{atan2}(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \\ \psi = \text{atan2}(r_{32}, r_{33}) \end{cases}$$

with  $\theta \in [-\pi/2, \pi/2]$ , or

$$\begin{cases} \varphi = \text{atan2}(-r_{21}, -r_{11}) \\ \theta = \text{atan2}(-r_{31}, -\sqrt{r_{32}^2 + r_{33}^2}) \\ \psi = \text{atan2}(-r_{32}, -r_{33}) \end{cases}$$

- if  $\theta \in [\pi/2, 3\pi/2]$ .
2.  $r_{11}^2 + r_{21}^2 = 0 \rightarrow \cos \theta = 0$ :  $\theta = \pm\pi/2$  and infinite solution exist (sum or difference of  $\psi$  and  $\varphi$ ).

It may be convenient to assign a value to an angle (e.g.  $\varphi$  or  $\psi$  equal to  $\pm 90^\circ$ ) and then compute the remaining one

$$\begin{cases} \theta = \pm\pi/2 \\ \varphi - \psi = \text{atan2}(r_{23}, r_{13}) = \text{atan2}(-r_{12}, r_{22}). \end{cases}$$

# D

---

## Spectral Analysis and Fourier Transform

### D.1 Fourier Transform of a Continuous Time Function

Let  $x(t)$  be a function  $\mathcal{T} \rightarrow \mathbb{R}$ , where  $\mathcal{T}$  is the time domain and  $\mathbb{R}$  the set of real numbers. The function  $x(t)$  has *finite energy* if the integral

$$\int_{-\infty}^{+\infty} x(t)^2 dt$$

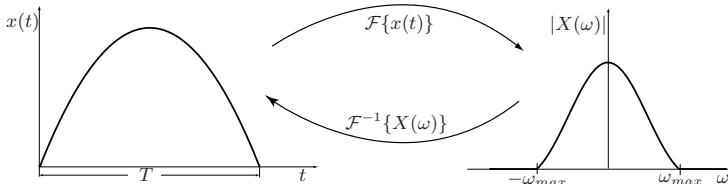
has a finite value. Notice that this property, for physical reasons, holds for the accelerations/velocities of all the trajectories  $q(t)$  presented in this book.

For functions  $x(t)$  with finite energy, it is possible to define the *Fourier transform* and the *inverse Fourier transform* (see Fig. D.1) as

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt, \quad x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega)e^{j\omega t} d\omega. \quad (\text{D.1})$$

By considering the Fourier transform in polar coordinates, i.e.  $X(\omega) = |X(\omega)|e^{j\varphi(\omega)}$ , its inverse can be rewritten as

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |X(\omega)|e^{j(\omega t + \varphi(\omega))} d\omega$$



**Fig. D.1.** Fourier transform of a bandlimited (and finite length) signal  $x(t)$ .

or, equivalently,

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |X(\omega)| [\cos(\omega t + \varphi(\omega)) + j \sin(\omega t + \varphi(\omega))] d\omega.$$

Since  $x(t) \in \mathbb{R}$ , it follows that

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |X(\omega)| \cos(\omega t + \varphi(\omega)) d\omega$$

and, by taking into account that  $|X(\omega)| \cos(\omega t + \varphi(\omega))$  is an even function, one obtains

$$x(t) = \frac{1}{\pi} \int_0^{+\infty} |X(\omega)| \cos(\omega t + \varphi(\omega)) d\omega$$

and finally

$$x(t) = \int_0^{+\infty} V(\omega) \cos(\omega t + \varphi(\omega)) d\omega \quad (\text{D.2})$$

where

$$\begin{aligned} V(\omega) &= \frac{|X(\omega)|}{\pi}, & \omega \geq 0 \\ \varphi(\omega) &= \arg\{X(\omega)\}, & \omega \geq 0. \end{aligned}$$

Equation (D.2) expresses the function  $x(t)$  as the “summation” of an infinite number of sinusoidal terms, each of them with frequency  $\omega$ , amplitude  $V(\omega)$  and phase  $\varphi(\omega)$ .

### D.1.1 Main properties of the Fourier transform

The main properties of the Fourier transform, which can be of interest for the analysis of trajectories, are now summarized. If  $x(t) \leftrightarrow X(\omega)$ ,  $y(t) \leftrightarrow Y(\omega)$  then:

**1. Linearity:**

$$\alpha x(t) + \beta y(t) \leftrightarrow \alpha X(\omega) + \beta Y(\omega), \quad \forall \alpha, \beta \in \mathbb{C}.$$

**2. Scaling:**

$$x(\lambda t) \leftrightarrow \frac{1}{|\lambda|} X\left(\frac{\omega}{\lambda}\right), \quad \forall \lambda \in \mathbb{R}, \lambda \neq 0.$$

**3. Shifting:**

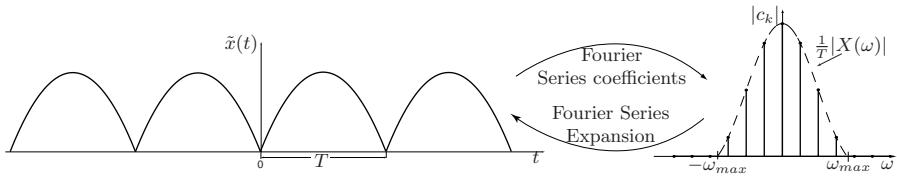
$$x(t - \lambda) \leftrightarrow e^{j\lambda t} X(\omega), \quad \forall \lambda \in \mathbb{R}.$$

**4. Derivative:**

$$\frac{dx(t)}{dt} \leftrightarrow j\omega X(\omega).$$

**5. Energy Theorem (Parseval):**

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega.$$



**Fig. D.2.** Fourier series coefficients of a periodic signal  $\tilde{x}(t)$ .

## D.2 Fourier Series of a Periodic Continuous Function

Let us consider a periodic function  $\tilde{x}(t)$  with period  $T$ , i.e.  $\tilde{x}(t+T) = \tilde{x}(t)$ . Its Fourier transform consists of impulses that in the frequency domain are spaced by  $\omega_0 = 2\pi/T$ , [101]. This result leads to the standard *Fourier Series (FS)* expansion (in the exponential form)

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}, \quad \omega_0 = \frac{2\pi}{T}$$

where the coefficients  $c_k$  are computed as

$$c_k = \frac{1}{T} \int_0^T \tilde{x}(t) e^{-jk\omega_0 t} dt.$$

If the periodic function  $\tilde{x}(t)$  is obtained by repeating the finite length function  $x(t) \in [0, T]$ , the coefficients of the Fourier series are related to the Fourier transform of  $x(t)$  by

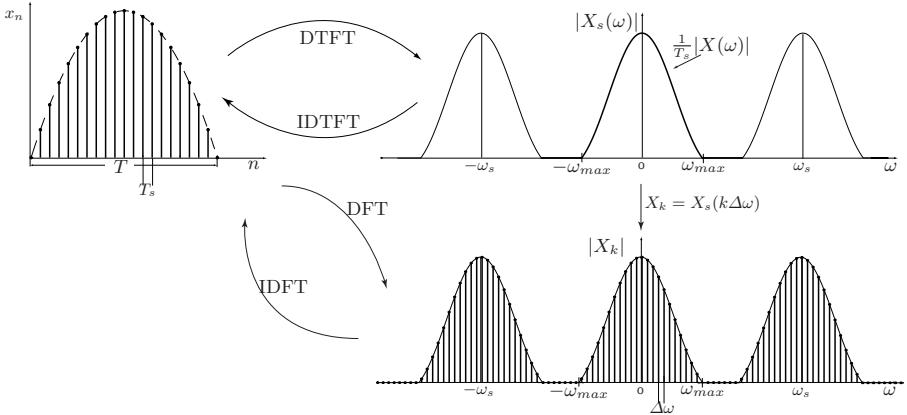
$$c_k = \frac{1}{T} X(k\omega_0), \quad \omega_0 = \frac{2\pi}{T}$$

that is they can be computed by sampling (and scaling by  $1/T$ ) the Fourier transform  $X(\omega)$ , as shown in Fig. D.2. As in the case of the Fourier transform, when a real signal  $\tilde{x}(t)$  is considered, the Fourier series can be written as the summation of harmonic functions:

$$\tilde{x}(t) = v_0 + \sum_{k=1}^{\infty} v_k \cos(k\omega_0 t + \varphi_k), \quad \omega_0 = \frac{2\pi}{T}$$

where

$$v_0 = c_0, \quad \begin{cases} v_k = 2|c_k| \\ \varphi_k = \arg(c_k) \end{cases}, \quad k > 0.$$



**Fig. D.3.** Discrete time Fourier transform and discrete Fourier transform of a sequence  $x_n$ .

### D.3 Fourier Transform of a Discrete Time Function

The Fourier transform of a sequence<sup>1</sup>  $x_n$ , which represents the values of a continuous-time function  $x(t) \in [0, T]$  at the discrete time instants  $t = nT_s$  where  $T_s$  is the sampling period, is defined as

$$X_s(\omega) = \sum_{n=-\infty}^{\infty} x_n e^{-j\omega nT_s}$$

which is periodic, with period  $\omega_s = \frac{2\pi}{T_s}$ . In this case, the inverse Fourier transform is

$$x_n = \frac{1}{\omega_s} \int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} X_s(\omega) e^{j\omega nT_s} d\omega.$$

It is possible to show that the Fourier transform of  $x_n$  consists of periodically repeated copies of the Fourier transform of  $x(t)$  (scaled by  $1/T_s$ ), i.e.

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega + k\omega_s). \quad (\text{D.3})$$

For the proof see [65]. If the Nyquist condition on the sampling time is satisfied (i.e.  $T_s < \frac{\pi}{\omega_{max}}$ , where  $\omega_{max}$  is such that  $|X(\omega)| \simeq 0, \forall \omega > \omega_{max}$ ), the spectrum  $X(\omega)$  can be obtained by multiplying  $X_s(\omega)$  by  $T_s$  in the interval  $\omega \in [-\frac{\pi}{T_s}, \frac{\pi}{T_s}]$ .

<sup>1</sup> Called Discrete Time Fourier Transform (DTFT).

### D.3.1 Discrete Fourier transform

If  $\tilde{x}_n$  is a periodic sequence with period  $N$ , i.e.  $\tilde{x}_n = \tilde{x}_{n+rN}$ , it is possible to write it as a Fourier series corresponding to a sum of complex exponential functions with frequencies that are multiples of the fundamental frequency  $\frac{2\pi}{N}$ :

$$\tilde{x}_n = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}_k e^{j(\frac{2\pi}{N})kn} \quad (\text{D.4})$$

where  $\tilde{X}_k$  are the *Discrete Fourier Series (DFS)* coefficients, defined as

$$\tilde{X}_k = \sum_{n=0}^{N-1} \tilde{x}_n e^{-j(\frac{2\pi}{N})kn}.$$

Note that the sequence  $\tilde{X}_k$  is periodic with period  $N$ , i.e.  $\tilde{X}_k = \tilde{X}_{k+rN}$ . Often, the discrete Fourier series is rewritten in terms of the complex quantity

$$W_N = e^{-j(\frac{2\pi}{N})}.$$

With this notation, the expression of DFS and of its inverse are respectively

$$\tilde{X}_k = \sum_{n=0}^{N-1} \tilde{x}_n W_N^{kn}, \quad \tilde{x}_n = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}_k W_N^{-kn}.$$

In case of a periodic sequence obtained by repeating a finite length sequence  $x_n$  (such that  $x_n = 0$  outside the range  $0 \leq n \leq N - 1$ ), that is

$$\tilde{x}_n = \sum_{r=-\infty}^{\infty} x_{n-rN},$$

the discrete Fourier series  $\tilde{X}_k$  is related to the Fourier transform of  $x_n$ , called *Discrete Fourier Transform (DFT)*, by means of

$$X_k = \begin{cases} \tilde{X}_k, & 0 \leq k \leq N - 1 \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, the DFT and its inverse (IDFT) are respectively defined as

$$X_k = \begin{cases} \sum_{n=0}^{N-1} x_n W_N^{kn}, & 0 \leq k \leq N - 1 \\ 0, & \text{otherwise} \end{cases} \quad (\text{D.5})$$

and

$$x_n = \begin{cases} \frac{1}{N} \sum_{k=0}^{N-1} X_k W_N^{-kn}, & 0 \leq k \leq N-1 \\ 0, & \text{otherwise.} \end{cases}$$

The DFT is characterized by properties similar to those of the Fourier transform for continuous functions (i.e. linearity, scaling, shifting, and so on), but the main advantage of adopting the DFT is the fact that very efficient algorithms for its computation exist (*Fast Fourier Transform (FFT)*). By means of these numerical methods the discrete Fourier transform can be computed in  $\mathcal{O}(N \log_2 N)$  operations, while the use of the definition (D.5) would require  $\mathcal{O}(N^2)$  operations. For the details about the FFT implementation see [19, 65]. It is possible to relate the DFT of the finite length sequence  $x_n$ , obtained by sampling the continuous function  $x(t) \in [0, T]$ , with the corresponding discrete time Fourier transform:

$$X_k = X_s(\omega) \Big|_{\omega = 2\pi k / NT_s}.$$

By considering that  $T_s = T/N$ , it is easy to show that the coefficients of the DFT are the values of  $X_s(\omega)$  sampled with a frequency period  $\Delta\omega = 2\pi/T$ :

$$X_k = X_s(k\Delta\omega). \quad (\text{D.6})$$

## D.4 Fourier Analysis of Signals Using DFT (and FFT)

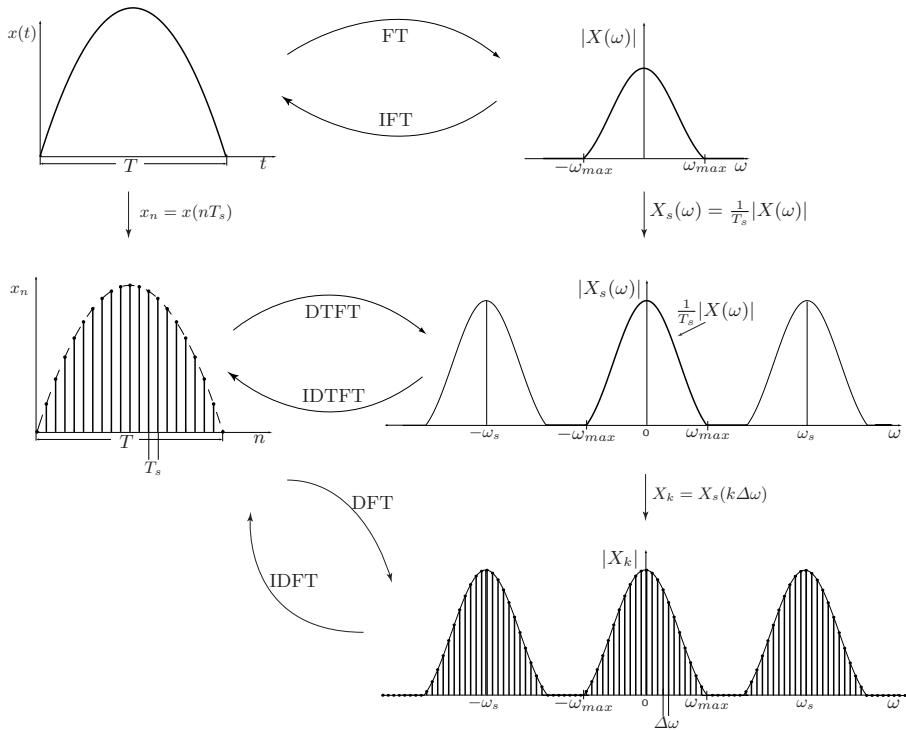
One of the main application of the DFT is the analysis of the frequency content of continuous-time signals. Given a bandlimited signal (possibly with the adoption of an anti-aliasing filter)  $x(t)$ , it is possible to obtain a discrete-time sequence  $x_n$  by sampling  $x(t)$  with a period  $T_s$  ( $x_n = x(nT_s)$ ). Since the sampling operation in the time domain corresponds to the periodic repetition in the frequency domain, as expressed by (D.3), the Fourier transform of the continuous signal  $x(t)$  is simply a restriction of the (continuous) Fourier transform of the discrete-time signal  $x_n$ :

$$X(\omega) = \begin{cases} T_s X_s(\omega), & \omega \in \left[-\frac{\omega_s}{2}, \frac{\omega_s}{2}\right] \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, by applying the DFT to  $x_n$ , which provides the value of  $X_s(\omega)$  at discrete frequencies  $k2\pi/T$  (see (D.6)), it is possible to numerically evaluate  $X(\omega)$ , being

$$X_k = X_s(k\Delta\omega) = \frac{1}{T_s} X(k\Delta\omega), \quad \text{with } \Delta\omega = \frac{2\pi}{T}.$$

The relationships among continuous- and discrete- time functions and their transforms are summarized in Fig. D.4.



**Fig. D.4.** Continuous and discrete Fourier transforms.

---

## References

1. Y. Xiao, K. Zhu, and H.C. Liaw. Generalized synchronization control of multi-axis motion systems. *Control Engineering Practice*, 13:809–819, 2005.
2. J.V. Gerwen. Electronic camming and gearing. *Assembly Automation*, 19:35–38, 1999.
3. C. Melchiorri. *Traiettorie per Azionamenti Elettrici*. Progetto Leonardo. Esculapio Ed., Bologna, I, second edition, 2003.
4. M.A. Gonzales-Palacios and J. Angeles. *Cam Synthesis*, volume 26 of *Solid Mechanics and its Applications*. Kluver Academic, 1993.
5. J. Angeles and C.S. Lopez-Cajun. *Optimization of cam mechanisms*. Kluwer Academic Publ., 1991.
6. F.Y. Chen. *Mechanics and Design of Cam Mechanisms*. Pergamon Press Inc., 1982.
7. P.W. Jensen. *Cam design and manufacture*. New York Industrial Press, 1965.
8. P.L. Magnani and G. Ruggeri. *Meccanismi per macchine automatiche*. UTET, 1986.
9. R.L. Norton. *Design of machinery*. McGraw-Hill, 1992.
10. Merriam-Webster dictionary, url: <http://www.m-w.com/dictionary/trajecotry>.
11. J. Angeles. *Fundamentals of robotic mechanical systems*. Springer-Verlag, 1997.
12. B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer-Verlag, Berlin, Heidelberg, 2008.
13. Thomas R. Kurfess, editor. *Robotics and Automation Handbook*. CRC Press, 2000.
14. Z. Koloc and M. Vaclavik. *Cam Mechanisms*, volume 14 of *Studies in Mechanical Engineering*. Elsevier, 1993.
15. A.S. Gutman. To avoid vibration - try this new cam profile. *Product engineering*, 25:42–48, Dec. 1961.
16. F. Freudenstein. On the dynamics of high-speed cam profiles. *International Journal of Mechanical Sciences*, 1:342–349, 1960.
17. S.A. Bazaz and B. Tondu. Minimum time on-line joint trajectory generator based on low order spline method for industrial manipulators. *Robotics and Autonomous Systems*, 29:257–268, 1999.
18. G. Strang. *Linear Algebra and Its Applications*. Thomson Brooks/Cole, fourth edition, 2006.

19. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, third edition, 2007.
20. G.J. Borse. *Numerical Methods with MATLAB*. PWS Publishing Company, 1997.
21. G.E. Forsythe. Generation and use of orthogonal polynomials for data-fitting with a digital computer. *Journal of Society for Industrial and Applied Mathematics*, 5:74–88, 1957.
22. C.W. Clenshaw and J. G. Hayes. Curve and surface fitting. *Journal of Applied Mathematics*, 1:164–183, 1965.
23. J.C. Mason and David C. Handscomb. *Chebyshev Polynomials*. CRC Press, 2002.
24. T. J. Rivlin. *The Chebyshev Polynomials*. Tracts in Pure & Applied Mathematics. John Wiley & Sons, 1974.
25. A. C. R. Newbery. Interpolation by algebraic and trigonometric polynomials. *Mathematics of Computation*, 20(96):597–599, 1966.
26. A. C. R. Newbery. Trigonometric interpolation and curve-fitting. *Mathematics of Computation*, 24(112):869–876, 1970.
27. T. Lyche and R. Winther. A stable recurrence relation for trigonometric B-splines. *Journal of Approximation Theory*, 25:266–279, 1979.
28. T. Lyche, L. L. Schumaker, and S. Stanley. Quasi-interpolants based on trigonometric splines. *Journal of Approximation Theory*, 95(2):280–309, 1998.
29. M. Neamtu, H. Pottmann, and L.L. Schumaker. Designing Nurbs cam profiles using trigonometric splines. *Journal of Mechanical Design, Transactions of the ASME*, 120(2):175–180, 1998.
30. E. Dyllong and A. Visioli. Planning and real-time modifications of a trajectory using spline techniques. *Robotica*, 21:475–482, 2003.
31. I.J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. *Quarterly of Applied Mathematics*, 4:45–99, 1946.
32. C. Reinsch. Smoothing by spline function. *Numerische Mathematik*, 10:177–183, 1967.
33. T. Lyche and L. L. Schumaker. Procedures for computing smoothing and interpolating natural splines. *Communications of the ACM*, 17(8):463 – 467, 1974.
34. B. Cao, G.I. Dodds, and G.W. Irwin. Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots. *Proceedings of IEE Conference on Control Theory and Applications*, 144:467–475, 1997.
35. R. L. Eubank. *Nonparametric Regression and Spline Smoothing*. Marcel Dekker, 1999.
36. C. Lee and Y. Xu. Trajectory fitting with smoothing splines using velocity information. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA '00*, San Francisco, CA, 2000.
37. L. Biagiotti and C. Melchiorri. Smooth trajectories for high-performance multi-axes automatic machines. In *Proc. 4th IFAC Symposium on Mechatronic Systems*, Heidelberg, G, Sept. 2006.
38. L. Piegl and W. Tiller. *The Nurbs Book*. Springer-Verlag, second edition, 1997.
39. A. De Luca, L. Lanari, and G. Oriolo. A sensitivity approach to optimal spline robot trajectories. *Automatica*, 27(3):535–539, 1991.

40. C. G. Lo Bianco and Aurelio Piazzi. Minimum-time trajectory planning of mechanical manipulators under dynamic constraints. *International Journal of Control*, 75(13):967–980, 2002.
41. A. Piazzi and A. Visioli. Global minimum-jerk trajectory planning of robot manipulator. *IEEE Transaction on Industrial Electronics*, 47(1):140–149, 2000.
42. W. Hoffmann and T. Sauer. A spline optimization problem from robotics. *Rendiconti di matematica*, 26 (7):221–230, 2006.
43. C.-S. Lin, P.-R. Chang, and J.Y.S. Luh. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Transaction on Automatic Control*, 28(12):1066–1074, 1983.
44. D. Simon. Data smoothing and interpolation using eighth order algebraic splines. *IEEE Transactions on Signal Processing*, 52(4):1136–1144, 2004.
45. H. Park. Choosing nodes and knots in closed B-spline curve interpolation to point data. *Computer-Aided Design*, 33:967–975, 2001.
46. C. Edwards, E. Fossas, and L. Fridman, editors. *Advances in Variable Structure and Sliding Mode Control*, volume 334 of *Lecture Notes in Control and Information Sciences*. Springer Verlag, 2006.
47. A. Sabanovic, L. Fridman, and S. K. Spurgeon, editors. *Variable Structure Systems: From Principles to Implementation*. IEE Book Series, 2004.
48. R. Zanasi and R. Morselli. Third order trajectory generator satisfying velocity, acceleration and jerk constraints. In *Proceedings of the 2002 International Conference on Control Applications*, Glasgow, UK, 2002.
49. R. Zanasi, C. Guarino Lo Bianco, and A. Tonielli. Nonlinear filter for smooth trajectory generation. In *Proceedings of the IFAC Symposium on Nonlinear Control Systems, NOLCOS'98*, Enschede, NL, 1998.
50. R. Zanasi, C. Guarino Lo Bianco, and A. Tonielli. Nonlinear filters for the generation of smooth trajectories. *Automatica*, 36:439–448, March 2000.
51. J.M. Hollerbach. Dynamic scaling of manipulator trajectories. *Journal of Dynamic Systems, Measurement and Control*, 106:102–106, 1983.
52. R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
53. L.-W. Tsai. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John Wiley & Sons, 1999.
54. H. W. Beaty and J. L. Kirtley. *Electric Motor Handbook*. McGraw-Hill, 1998.
55. B.K. Fussell and C.K. Taft. Brushless DC motor selection. In *Electrical Electronics Insulation Conference*, pages 345–353, Rosemont, IL, USA, Sept. 1995.
56. R. Fredrik, J. Hans, and W. Jan. Optimal selection of motor and gearbox in mechatronic applications. *Mechatronics*, 16(1):63–72, 2006.
57. P. Meckl and W. Seering. Minimizing residual vibration for point-to-point motion. *ASME Journal of Vibration, Acoustics, Stress, and Reliability in Design*, 107:378–382, 1985.
58. R.L. Norton. *Cam Design and Manufacturing Handbook*. Industrial Press, 2002.
59. K. Itao and K. Kanzaki. High-speed positioning with polydyne cams. *Review of the electrical communication laboratories*, 21(1-2):12–22, 1973.
60. T.R. Thoren, H.H. Engermann, and D.A. Stoddart. Cam design as related to valve train dynamics. *SAE Quarterly Transactions*, 6:1–14, 1952.
61. W.M. Dudley. New methods in valve cam design. *SAE Quarterly Transactions*, 2:19–33, 1948.

62. D.A. Stoddart. Polydyne cam design - III. *Machine Design*, 25(3):149–164, 1953.
63. E.E. Peisekah. Improving the polydyne cam design method. *Russian Engineering Journal*, 46:25–27, 1966.
64. J.-G. Sun, R.W. Longman, and F. Freudenstein. Determination of appropriate cost functionals for cam-follower design using optimal control theory. In *Proceedings of the American Control Conference*, San Diego, CA, 1984.
65. A. V. Oppenheim and R. W. Schafer. *Discrete-time signal processing*. Prentice-Hall, Upper Saddler River, NJ, second edition, 1999.
66. T.W. Parks and C.S. Burrus. *Digital Filter Design*. John Wiley & Sons, New York, 1987.
67. S. Winder. *Analog and Digital Filter Design*. Elsevier, 2002.
68. W. Singhose, N. Singer, and W. Seering. Comparison of command shaping methods for reducing residual vibration. In *Proceedings of the European Control Conference, ECC'95*, volume 2, pages 1126–1131, Rome, I, 1995.
69. N.C. Singer and W.P. Seering. Preshaping command inputs to reduce system vibration. *ASME Journal of Dynamic Systems, Measurement and Control*, 112:76–82, 1990.
70. W. Singhose, W. Seering, and N. Singer. Shaping inputs to reduce vibration: a vector diagram approach. In *Proceedings of the IEEE Conference on Robotics and Automation, ICRA'90*, Cincinnati, OH, 1990.
71. W. Singhose, W. Seering, and N. Singer. Residual vibration reduction using vector diagrams to generate shaped inputs. *ASME Journal of Mechanical Design*, 116:654–659, 1994.
72. W.E. Singhose, L.J. Porter, T.D. Tuttle, and N.C. Singer. Vibration reduction using multi-hump input shapers. *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, 119:320–326, 1997.
73. K. Ogata. *Discrete-time control systems*. Prentice-Hall, second edition, 1995.
74. T.D. Tuttle and W.P. Seering. A zero-placement technique for designing shaped inputs to suppress multiple-mode vibration. In *American Control Conference*, Baltimore, Maryland, 1994.
75. S. Devasia, D. Chen, and B. Paden. Nonlinear inversion-based output tracking. *IEEE Transactions on Automatic Control*, AC-41:930–942, 1996.
76. L.R. Hunt and G. Meyer. Stable inversion for nonlinear systems. *Automatica*, 33:1549–1554, 1997.
77. A. Visioli and A. Piazzi. A toolbox for input-output system inversion. *International Journal of Computers, Communications and Control*, 2:388–402, 2007.
78. D. Pallastrelli and A. Piazzi. Stable dynamic inversion of nonminimum-phase scalar linear systems. In *16th IFAC World Congress on Automatic Control*, Prague, CZ, 2005.
79. Tsuneo Yoshikawa. *Foundations of Robotics. Analysis and Control*. The MIT Press, 1990.
80. W. Khalil and E. Dombre. *Modeling, Identification and Control of Robots*. Hermes Penton Science, 2002.
81. F. L. Lewis, D. M. Dawson, and C. T. Abdallah. *Robot Manipulator Control Theory and Practice*. Control Engineering. Marcel Dekker, second edition, 2004.
82. Z. Yang and E. Red. On-line cartesian trajectory control of mechanisms along complex curves. *Robotica*, 15:263–274, 1997.

83. B.A. Barsky and T. D. Derose. Geometric continuity of parametric curves: three equivalent characterizations. *IEEE Computer Graphics and Applications*, 9(6), 1989.
84. B.A. Barsky and T. D. Derose. Geometric continuity of parametric curves: construction of geometrically continuous splines. *IEEE Computer Graphics and Applications*, 9(6), 1989.
85. P. J. Davis. *Interpolation and Approximation*. Dover, 1976.
86. G. M. Phillips. *Interpolation and Approximation by Polynomials*. CMS books in mathematics. Springer, 2003.
87. J. Park, S.Nam, and M. Yang. Development of a real-time trajectory generator for nurbs interpolation based on the two-stage interpolation method. *International Journal of Advanced Manufacturing Technology*, 26(4):359–365, 2005.
88. C. Blanc and C. Schlick. Accurate parametrization of conics by Nurbs. *IEEE Computer Graphics and Applications*, 16(6):64–71, 1996.
89. H. Akima. A new method of interpolation and smooth curve fitting based on local procedures. *Journal of the Association for Computing Machinery*, 17(4):589 – 602, 1970.
90. F.-C. Wang and P. K. Wright. Open architecture controllers for machine tools, Part 2: a real time quintic spline interpolator. *Journal of Manufacturing Science and Engineering*, 120:425–432, 1998.
91. F.-C. Wang and D. C. H. Yang. Nearly arc-length parameterized quintic-spline interpolation for precision machining. *Computer Aided Design*, 25(5):281–288, 1993.
92. R. Volpe. Task space velocity blending for real-time trajectory generation. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'93*, Atlanta, Georgia, US, 1993.
93. J. Lloyd and V. Hayward. Real-time trajectory generation using blend functions. In *Proceedings of IEEE International Conference on Robotics and Automation, ICRA'91*, Sacramento, CA, USA, 1991.
94. R. V. Fleisig and A. D. Spence. A constant feed and reduced angular acceleration interpolation algorithm for multi-axis machining. *International Journal of Machine Tools and Manufacture*, 33(1):1–15, 2001.
95. R.-S. Lin. Real-time surface interpolator for 3-D parametric surface machining on 3-axis machine tools. *International Journal of Machine Tools and Manufacture*, 40:1513–1526, 2000.
96. C.-W. Cheng and M.-C. Tsai. Real-time variable feed rate Nurbs curve interpolator for CNC machining. *International Journal of Advanced Manufacturing Technology*, 23:865–873, 2004.
97. S.D. Conte and C. de Boor. *Elementary Numerical Analysis: An Algorithmic Approach*. McGraw-Hill, third edition, 1981.
98. W. H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in FORTRAN: The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.
99. C. de Boor. *A Practical Guide to Spline*, volume 27 of *Applied Mathematical Sciences*. Springer Verlag, 1978.
100. E.V. Shikin and A.I. Plis. *Handbook on Splines for the User*. CRC, 1995.
101. A. Papoulis. *Signal Analysis*. McGraw-Hill, New York, 1984.

---

# Index

- Acceleration
  - centripetal, 416
  - coefficient of, 250
  - constraint, 135, 216, 230, 245, 415, 416
  - root mean square (RMS), 249
  - spectrum, 287, 299, 303, 323
  - tangential, 416
- Angle
  - Euler, 491
  - Roll-Pitch-Yaw, 493
- Angle-axis, 490
- Approximation, 180, 346, 364, 368, 371
  - B-spline, 364
  - global, 364
  - Hausdorff distance, 449
  - least square minimization, 364
  - with prescribed tolerance, 187
- Arc length, 353
- B-spline, 194, 359, 376, 397, 440, 467
  - approximation, 364, 371
  - basis function, 467
  - basis function evaluation, 469
  - boundary conditions, 196, 373, 378
  - continuity, 207
  - control point, 471
  - control polygon, 471
  - cubic, 360, 371
  - cyclic conditions, 197, 200, 379, 382
  - degree, 467
  - differentiation, 475
  - evaluation, 474
- extra-knot, 199
- interpolation, 360
  - knot, 195, 359, 366, 377, 467
  - knots choice, 195, 377
  - mixed interpolation approximation, 368
  - of order five, 204, 384
  - of order four, 197, 379
  - order, 467
  - partition of the unity, 468
  - properties, 471
  - smoothing, 346, 371, 445
- Bézier, 32, 483
  - Bézier curve, 393, 406, 483
    - cubic, 395
    - derivative, 486
    - evaluation, 484
    - interpolation, 395
    - quintic, 400
  - Bell trajectory, *see* Double S
  - Bernstein polynomial, 32
  - Bernstein polynomial, 483
  - Binormal vector, 353
  - Bode diagram, 286
  - Breakpoint, 188, 359, 468
    - centripetal distribution, 189
    - choice, 188
    - cord length distribution, 189
    - equally spaced, 189
- Cam, 4
  - electronic, 4, 241
  - mechanical, 4, 241

- Cartesian space, 341
- Chebyshev polynomial, 162
- Computer Numerical Control (CNC), 391
- Condition number, 152
- Constraint
  - acceleration, 230
  - jerk, 230
  - velocity, 230
- Continuity, 343, 349, 399, 420
  - derivative, 343
  - geometric, 343, 345
  - parametric, 343
- Continuous torque, 246
- Control point, 484
- Control polygon, 484
- Cubic polynomial, 23, 166
  - coefficient of acceleration, 253
  - coefficient of velocity, 253
  - frequency spectrum, 292
  - vibration, 277
- Cubic spline, 166
  - approximation, 180
  - assigned initial and final velocities, 169, 175
  - assigned initial and final velocities and acceleration, 177
  - breakpoint, 188
  - choice of the time instants, 188
  - duration optimization, 189
  - frequency spectrum, 299, 300
  - interpolation, 167
  - periodic, 172
  - properties, 168
  - smoothing, 180
- Curvature vector, 345, 384
  - computation from via-points, 394
- Curvilinear coordinate, 353
- Cyclic conditions, 165, 173, 197, 379
- Cycloidal trajectory, 43, 235
  - coefficient of acceleration, 253
  - coefficient of velocity, 253
  - frequency spectrum, 293, 295, 300
  - modified, 127, 229
  - normalized form, 235
  - vibration, 275
- de Casteljau algorithm, 484
- Diagram speed-torque, 245
- Discrete Fourier Series (DFS), 499
- Discrete Fourier Transform (DFT), 295, 499
- Discrete Time Fourier Transform (DTFT), 498
- Double S, 79, 209, 256
  - coefficient of acceleration, 256
  - coefficient of velocity, 256
  - computation for negative displacement, 90
  - duration, 101
  - flux diagram for parameters' computation, 90
  - frequency spectrum, 290
  - online computation, 93, 209
  - with preassigned durations of the different phases, 102
  - with zero initial and final velocities, 90
- Dynamics
  - inversion, 306, 330
  - mechanical system, 247, 305
  - non-minimum phase, 331
  - robot, 237
- Electric motor, 245
  - continuous torque, 246
  - diagram speed-torque, 245, 247, 254
  - peak torque, 246
  - rated speed, 246
- Electronic cam, 4
- Elliptic trajectory, 45
  - frequency spectrum, 295, 300
  - vibration, 276
- Exponential trajectory, 47
- Extra-Insensitive (EI) shaper, 324
- Fast Fourier Transform (FFT), 500
  - frequency analysis, 500
- Feed rate, 421
  - constant, 421
  - double S, 434
  - variable, 424
- Feedforward, 306, 330
- Fifteen segments trajectory, 107
- Filtering, 318
  - Extra-Insensitive (EI) shaper, 324
  - input shaping, 318
  - low-pass, 318

- Zero Vibration (ZV) shaper, 319
- Fourier, 51, 52
  - aperiodic continuous function, 495
  - discrete series, 499
  - discrete time function, 498
  - discrete transform, 295, 499
  - fast transform, 500
  - frequency analysis, 500
  - periodic continuous function, 299, 497
- Fourier series, 52, 299, 497
- Fourier transform, 285, 291, 303, 495
  - properties, 496
- Frenet frame, 353
- Frequency, 303–305
  - analysis, 285
  - modification, 303, 304
- Freudenstein, 54, 55, 253, 283, 284, 299, 300
- Frobenius norm, 365
- Geometric path, 7, 342
- Gutman, 53, 253, 282, 299, 300
- Harmonic trajectory, 42, 235
  - coefficient of acceleration, 253
  - coefficient of velocity, 253
  - frequency spectrum, 293, 295, 300
  - normalized form, 235
  - vibration, 274
- Hausdorf distance, 449
- Horner formula, 463
- Interpolation, 164, 346, 347, 359, 368, 375
  - B-spline, 359
  - cubic splines, 167
  - global, 359
  - Lagrange formula, 154
  - linear, 406
  - local, 393
  - polynomial, 151
  - trigonometric polynomial, 164
- Jerk
  - constraint, 209, 230, 416
- Kinematics, 7
  - inverse, 7, 342
- Knot, 359, 366
- centripetal distribution, 360
- choice, 366
- cord length distribution, 360
- equally spaced, 360
- Lagrange formula, 154
- Laplace transform, 322
- Linear interpolation, 406, 429
- Master-slave system, 241
- Matlab
  - precision, 153
- Matrix
  - diagonal, 365
  - Frobenius norm, 365
  - trace, 365
- Mechanical
  - task, 247
- Mechanical cam, 4
- Mechanical model, 266, 305, 331
  - $n$  degrees of freedom, 267
  - modeling error, 326
  - nonlinear, 269, 270
  - one degree of freedom, 266, 322
- Modified cycloidal
  - Alt modification, 128
  - distortion angle, 129
  - frequency spectrum, 299, 300
  - Wildt modification, 128
- Motion law, 7, 342, 415, 418
  - double S, 419
- Motion primitive, 343, 449
  - circular arc, 356
  - straight line, 356
- Motor sizing, 245
- Multipoint trajectory, 393
  - B-spline, *see* B-spline
  - intermediate velocities computation, 25
  - orthogonal polynomial, 155
  - polynomial, 151
  - spline, *see* Spline
  - trigonometric polynomial, 164
- Neville, 154
- Normal vector, 353
- Normalized form, 31, 230, 291, 457
  - cubic polynomial, 231
  - cycloidal trajectory, 235

- exponential trajectory, 48
- harmonic trajectory, 235
- polynomial of degree five, 231
- polynomial of degree seven, 232
- polynomials of higher degree, 30
- Normalized polynomials, 231, 457
- Nurbs, 391, 481
  - evaluation, 483
  - weights, 392
- Orientation, 342, 347, 429, 488
  - angle-axis, 349, 490
  - Euler angles, 342, 348, 491
  - Roll-Pitch-Yaw angles, 342, 348, 493
  - rotation matrix, 347, 489
- Orthogonal polynomial, 155
  - Chebyshev, 162
  - frequency spectrum, 299, 300
- Parameterization, 31, 230, 457
- Parametric curve, 341
- Peak torque, 246
- Periodic condition, *see* Cyclic conditions
- Polydyne, 305
- Polynomial, 15, 23, 26, 28, 119, 151
  - Chebyshev, 162
  - cubic, 23, 231, 233, 292, 299, 300
  - Lagrange formula, 154
  - of degree five, 26, 231, 233, 253, 299, 300
  - of degree seven, 28, 232, 233, 253
  - orthogonal, 155, 299, 300
  - trigonometric, 164
  - vibration, 278
- Polynomial evaluation, 463
  - Horner formula, 463
- Reparameterization, 396, 415
- Robot
  - dynamics, 237
  - end effector, 347
  - kinematics, 7
- Root mean square (RMS), 249
- Rotation, 350, 488
  - angle-axis, 490
  - Euler angles, 491
  - matrix, 489
  - Roll-Pitch-Yaw angles, 493
- Rotation matrix, 357
- Saturation, 228
  - dynamic, 228
  - kinematic, 228
- Scaling
  - dynamic, 236
  - geometric, 223
  - kinematic, 230
  - time, 228, 303, 396, 416
- Schoenberg, 166
- Seven segments trajectory, *see* Double S
- Sherman-Morrison formula, 465
- Shift
  - time, 224, 226, 396
- Sinusoidal trajectory
  - modified, 124, 253, 299, 300
- Smoothness, 343, 371
- Snap, 204, 306
  - continuity, 204
- Spectrum
  - residual, 49
- Spline
  - approximation, 364
  - B-basis, 467
  - B-form, 471
  - B-spline, 359, 364, 467
  - Basic, 467
  - clamped, 182
  - conversion from PP-form to B-Form, 479
  - cubic, 166, 299, 300
  - cyclic conditions, 173, 197, 379
  - interpolation, 359
  - natural, 167
  - Nurbs, 481
  - periodic, 167
  - PP-form, 479
  - trigonometric, 165
- Splinedyne, 318
- Synchronization, 66, 241
  - trapezoidal trajectory, 66
- Tangent vector, 345, 353, 384, 389
  - computation from via-points, 394
  - interpolation, 389
- Thomas algorithm, 465
- Time scaling, 228
  - constant, 229, 239
- Torque
  - constraint, 237

- continuous, 249
- inertial, 247
- reflected, 247
- root mean square (RMS), 249
- Trajectory
  - '4-3-4', 118
  - 3D space, 341
  - asymmetric constant acceleration, 295, 300
  - asymmetric constant acceleration, 21
  - based on Fourier series, 51
  - Cartesian space, 341
  - circle, 355
  - constant acceleration, 18, 295, 300
  - constant acceleration with cycloidal/cubic blends, 144, 461
  - constant velocity, 17, 287, 295, 300
  - constant velocity/acceleration with cycloidal or harmonic blends, 133
  - constraint on the acceleration, 135, 216, 230, 245, 415, 416
  - constraint on the velocity, 133, 230, 245, 416
  - cubic polynomial, 23, 231, 253, 277, 292, 299, 300
  - cyclic, 165, 172, 248, 299, 379
  - cycloidal, 43, 127, 142, 148, 235, 253, 275, 293, 295, 300
  - double S, 79, 148, 209, 256, 290
  - elliptic, 45, 276, 295, 300
  - exponential, 47
  - fifteen segments, 107
  - filtering, *see* Filtering
  - frequency analysis, 285
  - Freudenstein 1-3, 54, 253, 283, 299, 300
  - Freudenstein 1-3-5, 55, 253, 284, 299, 300
  - geometric modification of, 223
  - Gutman 1-3, 53, 253, 282, 299, 300
  - harmonic, 42, 235, 253, 274, 293, 295, 300
  - helix, 354
  - linear trajectory with circular blends, 279
  - linear with circular blends, 59, 299, 300
  - linear with parabolic blends, 62
  - linear with polynomial blends, 76
  - minimum time, 140, 231, 233, 241
  - modified cycloidal, 127, 299, 300
  - modified sinusoidal, 124, 253, 299, 300
  - modified trapezoidal, 119, 253, 281, 299, 300
  - motion primitive, 356
  - multi-dimensional, 6, 341
  - multipoint, 24, 28, 151, 346
  - nonlinear filter, 208
  - normalized, 31, 230, 291, 457
  - one-dimensional, 3
  - online computation of the double S, 93
  - online planning, 208
  - optimization, 208, 241
  - orthogonal polynomial, 155, 299, 300
  - parabolic, 148
  - piecewise polynomial, 117
  - polydyne, 305
  - polynomial, 15, 119
  - polynomial of degree five, 26, 231, 253, 278, 299, 300
  - polynomial of degree seven, 28, 232, 253
  - spline, 166, 299, 300
  - splinedyne, 318
  - straight line, 356
  - synchronization, 241
  - translation, 223
  - trapezoidal, 62, 148, 216, 256, 280, 287, 299, 300
  - trigonometric, 42, 144
  - trigonometric polynomial, 164
  - with preassigned acceleration, 65
  - with preassigned acceleration and velocity, 65
  - Transfer function, 322, 329, 330
    - inverse, 330
  - Transform
    - Fourier, 52, 285, 291
    - Laplace, 286, 322, 330
    - Zeta, 329
  - Trapezoidal, 62, 216, 256
    - coefficient of acceleration, 256
    - coefficient of velocity, 256
    - duration, 69
    - frequency spectrum, 287, 299, 300

- modified, 119, 253, 281, 299, 300
- multipoint, 67, 74
- vibration, 280
- with finite initial and final velocities, 70
  - with preassigned acceleration, 65
  - with preassigned acceleration and velocity, 65
  - with preassigned durations, 69
- Tridiagonal system, 464
  - cyclic, 465
  - Sherman-Morrison formula, 465
  - Thomas algorithm, 465
- Trigonometric polynomial, 164
- Trigonometric spline, 165
- Trigonometric trajectory, 42
- Uniform parameterization, 397, 423, 436
- Vandermonde matrix, 151
  - condition number, 152
- Velocity
  - coefficient of, 250
  - constraint, 230, 245, 415, 416
  - root mean square (RMS), 249
- Via-point, 343
- Vibration, 47, 57, 265, 319
- Zero Vibration (ZV) shaper, 319