UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXVII Ciclo

Real-time trajectory planning for systems subject to high order kinematic and dynamic constraints

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Corrado Guarino Lo Bianco

Dottorando: *Fabio Ghilardelli*

Gennaio 2015

To my parents,
for their support

Contents

Introduction

This thesis deals with the smooth trajectory generation problem for systems subject to high order dynamic and kinematic constraints. The task is achieved by formulating appropriate constrained optimization problems, which are handled by means of conventional linear programming algorithms, or alternatively by means of specifically devised nonlinear filters. Considered constraints are obviously represented by the dynamic and kinematic limits affecting mechatronic systems. Mechanical oscillations and vibrations are correlated to the smoothness of the reference signals, so that a smart planning guarantees reduced stresses and wears together with a general improvement in control loops performances. Assumed cost functions depend on the application. For example, a typical cost function is given by the total traveling time of the trajectory: Minimal execution times lead to improved productivity performances. Alternatively, the optimization problem could consider the minimization of the energy consumption associated to systems' transients: In this case trajectory planners must be able to generate profiles that minimize the motors' losses in order to save energy.

The planning issues described above can either be dealt with online techniques or offline strategies, i.e. by executing the overall planning stage before any movement can occur or by continuously adapting the reference trajectory during the task execution. In industrial contexts in particular, planning problems are typically solved with offline strategies by assuming the perfect knowledge of the dynamic model of controlled plants. This assumption obviously represents an evident approximation since systems' models are characterized by uncertainties. Moreover, in actual industrial contexts, systems must react promptly to unforeseen events: New generation machines must be driven through trajectories devised on-the-fly on the basis of the mutating environmental conditions. The online trajectory computation guarantees a reactive response to unpredictable situations generated, for example, by unmodelled obstacles or by a reaction to emergency situations. Furthermore, it can also be used to handle kinematic singularities online or undesired behaviors caused by model uncertainties.

The planning approaches proposed in this work are based on the so-called path-velocity decomposition paradigm: The trajectory is obtained by first designing the desired geometric path and then by assigning a time-law to the movement along it. The optimal planning problems proposed in this thesis are defined in this framework. They are aimed at finding real-time and time-optimal solutions which fulfill both the dynamic and the kinematic limits of the considered systems. This result is generally achieved by designing proper feedback schemes which are able to online modify unfeasible trajectories in order to fulfill the assigned limits. For this purpose, a dynamic filter is presented in Chapter 1. It is able to smooth a rough scalar trajectory compatibly with a given set of bounds concerning its first, second and third time derivatives in real time. The filter generates a new profile that is identical to the original trajectory only if this latter is feasible with respect to the limits. Conversely, if the given profile is not feasible, it is replaced by its best feasible approximation. The filter proposed in Chapter 1 is the core of the trajectory scaling systems designed in Chapters 2 and 3. Trajectory scaling systems have been introduced in literature in order to scale down the time-law of trajectories to constrain the dynamics of controlled systems within given limits. It is well-known indeed that saturation phenomena could affect the path-tracking performances of mechatronic systems. They can be avoided in the planning stage by taking into account the dynamic behavior of the plant, resulting in evident improvements to the path-tracking performances. Chapter 2 proposes a solution for a planning scenario in the configuration space: Constraints, which are managed in real time by the trajectory scaling system, are defined in the joint space and concern maximum joint velocities, acceleration and jerks. Furthermore, the scaling system is also able to manage limits on the joints' generalized forces and on their time derivatives, but this feature requires direct access to the control loops: If the commercial controller is not accessible, the scaling system can only manage the kinematic constraints.

Chapter 3 presents a trajectory scaling system in the operational space for industrial manipulators: Paths in Cartesian space could elicit the violation of joint limits more easily than paths in the configuration space. This happens especially in configurations that are close to kinematic singularities: Trajectories that pass close to singular points could elicit high joint velocities even in the event of slow Cartesian motions. Problems deriving from singular configurations are well-known in robotic literature. Historical approaches mainly propose solutions for redundant manipulators in offline contexts. Non-redundant manipulators cannot share the same solutions, so that alternative strategies must be proposed in order to manage singularities. The solution presented in Chapter 3 is based on a reduction in the longitudinal velocity along an assigned path, made in order to guarantee the fulfillment of a given set of constraints. The time-delay accumulated in order to guarantee feasibility is eliminated as soon as possible, depending on the status of motion of the system. The scaling system is able to simultaneously handle constraints in the configuration and joint spaces: The computational burden is compatible with online implementation. The proposed approach is tested on a six degrees of freedom manipulator. Since the manipulator controller is not accessible, only kinematic limits have been taken into account.

In Chapter 4, the trajectory planning problem is dealt with an alternative strategy, but still considering motions in the operational space and non-redundant manipulators. More precisely, singularities are avoided by admitting limited modifications of the path primitives. In the novel approach, the assigned time-laws and Cartesian paths are considered as "mandatory" for task accomplishment. Conversely, minor orientation changes to the tool frame are considered "admissible". The reason for this choice is represented by industrial applications like automatic gluing or arc welding, which do not allow any reduction in the longitudinal speed or changes to the Cartesian path. For such applications, the traveling time is mandatory whereas small orientation changes cannot adversely affect the correct task execution.

All the proposed strategies have been tested against actual robotic platforms such as a Comau Smart-Six manipulator and a Quanser Consulting 2R planar manipulator.

Some conclusions and some future work recommendations are proposed in the final chapter.

1 Real-time trajectory generation with dynamic filters

Smooth trajectory generation guarantees improved tracking performances for feedback control systems. For this reason, trajectories studied in this thesis will be characterized by proper continuity conditions and their dynamics will be accordingly bounded. Moreover, due to reasons linked to productivity, industrial applications normally require a minimization of the time necessary to complete assigned tasks. The traveling time minimization imposes high actuators' speeds and, consequently, mechanic and dynamic limits could easily be reached and violated. Such limits are correlated to the maximum velocities that motors can reach and to the maximum torque that actuators can provide. Torque constraints are related to the velocity change rates, i.e. to the accelerations that motors can tolerate. If these bounds are neglected at the planning stage, they could cause unexpected controllers' behaviors and lead to high tracking errors. Additionally, it is well-known that torque discontinuities cause unwanted oscillations that can endanger the control system stability, so that these must be strictly avoided.

In order to deal with the aforementioned issues, reference generators must be able to produce smooth trajectory profiles that are compliant with the kinematic and dynamic constraints of the mechanical system. In particular, the required degrees of smoothness imply that generated profiles must be continuous up to their second time derivative, i.e. the minimum degree of smoothness that is required in order to avoid acceleration discontinuities. Robotic literature proposes several solutions to the abovementioned problems, which can be roughly divided into two mainly categories: Offline methods and online approaches.

In early works, the reference generation problem was addressed by means of offline strategies. For example, minimum-time trajectories, which satisfy kinematic and dynamic constraints, have been planned in [1] and in [2] with the aid of a polyhedron search strategy. A similar problem has been studied in [3] by also considering constraints on the actuator torques: The solution has been found by adopting a sensitivity approach. A discretized version of the same problem has been solved in [4] by means of a linear programming strategy.

The offline solutions are only useful for particular applications where the controlled plant does not interact with the operating environment or where the environment is fully structured and known *a priori*. However, in a modern industrial context, automatic systems must be able to react to unpredictable and instantaneous changes: For example, human-robot interaction and knowledge limitation *a priori* are also becoming more frequent in industrial environments. For these reasons the planning stage has to be computed online, i.e. with execution times compatible with control loops updating, and the introduction of online and minimum-time trajectory generators are essential to support these increasingly challenging requirements.

The online trajectory generation problem has been mainly addressed in robotic literature through two different approaches: With direct planning methods, i.e. by resolving explicit online optimization problems (see [5, 6] for example) or with indirect methods, i.e. by generating trajectories via proper feedback systems. In [6] it has been shown that the optimization process typically converges in some milliseconds, a computational time that is not compatible with many real-time applications. A different approach has been presented in [7–9], where trajectories with bounded velocities, accelerations and jerks have been generated online by means of decision trees which only require a few microseconds. The branches of the decision trees have been defined by means of an offline investigation concerning all possible evolutions. The sole drawback of these approaches is represented by the complexity of the planning algorithm. This consideration is at the basis of the studies on the indirect planning methods, which aim for the same performances but are obtained with compact algorithms.

The early solutions based on indirect planning methods were based on second order filters that were able to constrain the resulting profile within given limits on maximum velocities and accelerations. Examples of these first milestones have been proposed in [10–13] and in [14–16] respectively for continuous and discrete-time frameworks. These kinds of planning strategies have been widely employed in actual real-time applications because, compared to previously cited direct planning methods, they are characterized by several advantages: Trajectories are not limited to rest-to-rest movements but, conversely, generic initial and final conditions can be handled, constraints can be changed in real time, and, finally, the code that is required for their implementation is extremely compact and efficient, so that it can be executed with industrial micro-controller boards.

In a continuous-time context, the first proposal of a jerk-limited feedback planner was introduced in [17] with a scheme able to generate trajectories characterized by bounded velocities, accelerations and jerks. The planner does not handle generic interpolation conditions but, conversely, can only drive the system toward the rest status.

The first third-order continuous-time solution, which was able to manage generic interpolating conditions, has been proposed in [18]. Its discrete-time implementation is affected by jerk chattering and transient overshoots (see [19]). The discrete-time scheme, which was later presented in [20] in order to deal with these problems, was only able to manage the jerk bounds. For this reason, this solution was later improved in [21] to simultaneously consider velocity, acceleration and jerk limits, and was subsequently revised in [22] by considering asymmetric jerk bounds: The jerk chattering problem was totally eliminated but, for particular interpolating conditions, the overshoot issue, caused by suboptimal transients, was still present.

The main goal of this chapter is to present a nonlinear variable-structure filter that solves all the issues still left open in [22] and the following challenging problem: Given a piecewise continuous input signal, the filter must be able to generate a smooth output reference that represents its best possible approximation compatibly with some assigned bounds on velocities, accelerations and jerks. The admissible bounds can be asymmetric and can be changed online. The bounds asymmetry is a novelty with respect to any other solution proposed in literature and is essential in order to handle applications like those described in [23, 24]. Furthermore, the filter always guarantees minimum-time transients and eliminates the overshoot problem that was affecting analogous filters proposed in [21, 22]. Many results discussed in this chapter have been presented early in [25].

The chapter organization can be summarized as follows. In §1.1 the trajectory scaling problem is formulated and the third-order discrete-time filter is proposed in §1.2. The convergence properties of the filter are analyzed in §1.3, in §1.4, and in §1.5. The performances of the filter are compared in §1.6 with those of the filter proposed in [21]. In the last section, §1.7, a numerical test case is discussed to highlight the main characteristics of the filter.

1.1 The optimal trajectory scaling problem

Let us introduce a preliminary definition:

Definition 1. *A function*

$$\begin{array}{rcl} f:[0,t_f] & \to & \mathbb{R} \\ t & \to & f_d := f(t) \end{array}$$

is feasible, and it is possible to write $f \in \mathscr{F}$, if it is continuous together with its first and second time derivatives and if it fulfills the following constraints

$$\begin{array}{ll} R^- \leq \dot{f}(t) \leq R^+, & \forall t \in [0,t_f], \\ S^- \leq \ddot{f}(t) \leq S^+, & \forall t \in [0,t_f], \\ U^- \leq \dddot{f}(t) \leq U^+, & \forall t \in [0,t_f]. \end{array}$$

where $R^-, S^-, U^- \in \mathbb{R}^-$ and $R^+, S^+, U^+ \in \mathbb{R}^+$ are freely assignable bounds.

The nonlinear discrete-time filter proposed in this chapter solves the following problem:

Problem 1. *Given a piecewise reference signal $r(t)$ made of steps, ramps or parabolas, evaluate an optimal output signal $x(t)$, which is continuous together with its first and second time derivatives, by solving one of the following optimality problems, selected on the basis of the feasibility of $x(0)$ and $r(t)$:*

a) $x(0) \notin \mathscr{F}$

$$\min_{x(t)} \; t_f$$

$$\textit{subject to} \qquad x(t_f) \in \mathscr{F}.$$

t_f *evidently indicates the instant in which* $x(t)$ *becomes feasible.*

b) $r(t) \notin \mathscr{F}$ *and* $x(0) \in \mathscr{F}$

$$\min_{x(t)} \int |r(\tau) - x(\tau)|\, d\tau$$

$$\textit{subject to} \qquad x(t) \in \mathscr{F}.$$

c) $r(t) \in \mathscr{F}$ *and* $x(0) \in \mathscr{F}$

$$\min_{x(t)} \; t_f$$

$$\textit{subject to} \qquad r(t_f) - x(t_f) = 0$$

$$x(t) \in \mathscr{F}, \quad \forall t \in [0, t_f].$$

t_f *is evidently the instant in which* $x(t)$ *hangs* $r(t)$*. The tracking condition must be obtained, compatibly with the given bounds, without overshoot.*

Problem 1 can be summarized as follows. If $x(0) \notin \mathscr{F}$, then feasibility must be gained in minimum time, otherwise two different situations could occur: If $r(t) \notin \mathscr{F}$, then $x(t)$ must be its best feasible approximation, while, $r(t)$ must be hanged in minimum time if $r(t) \in \mathscr{F}$. Evidently, feasibility represents the prior target of the system. The bounds could be time-varying and can also be changed during transients. Analogously, $r(t)$ could be modified at any time, so that the solution of *Problem* 1 must be evaluated at each sample time. Since control systems are mainly governed by means of digital controllers, a discrete time solution of *Problem* 1 is proposed. In the following, subscript $i \in \mathbb{N}$ indicates sampled variables that are acquired at time $t = iT$, where T is the system sampling time.

The problem is clearly similar to the one that was considered in [21] and [22], but, as a novelty, asymmetric jerk constraints are handled. Moreover, in [21] and [22], under particular operating conditions, some transients toward r were suboptimal and characterized by overshoot issues: The solution here proposed totally eliminates both problems. These improvements have required a complete redefinition of the filter control laws.

1.2 The discrete-time third order filter

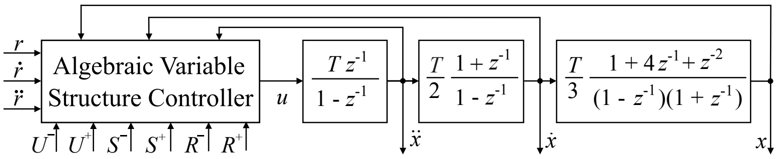
r
$\dot{r}$
$\ddot{r}$
Algebraic Variable
Structure Controller
u
$\frac{T z^{-1}}{1 - z^{-1}}$
$\frac{T}{2} \frac{1+z^{-1}}{1 - z^{-1}}$
$\frac{T}{3} \frac{1 + 4z^{-1} + z^{-2}}{(1 - z^{-1})(1 + z^{-1})}$
$\ddot{x}$
$\dot{x}$
x
U^{-}
U^{+}
S^{-}
S^{+}
R^{-}
R^{+}

Figure 1.1: The discrete-time system that solves *Problem* 1. The system is composed by a dynamic chain based on three integrators and an algebraic variable-structure controller.

The structure of the discrete-time filter is shown in Fig. 1.1 and it is the same that was proposed in [21] and in [22], i.e., it is made of a chain of three integrators. The Algebraic Variable Structure Controller (AVSC), that is based on sliding mode techniques [26, 27], has been completely rewritten with respect to the version presented in [21] in order to fulfill the new requirements. The AVSC uses a combination of appropriate Sliding Surfaces (SS) to robustly stabilize the system and to solve *Problem* 1. This section shows the control law that drives the AVSC.

The system dynamics is only due to the integrators chain and can be represented as follows

$$\mathbf{x}_{i+1} = \mathbf{A}\,\mathbf{x}_i + \mathbf{b}\,u_i\,, \tag{1.1}$$

where $\mathbf{x}_i := [x_i\ \dot{x}_i\ \ddot{x}_i]^T$ is the system state and

$$\mathbf{A} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix}. \tag{1.2}$$

Reference signal r_i is evaluated as follows

$$\mathbf{r}_{i+1} := \mathbf{A}\,\mathbf{r}_i\,, \tag{1.3}$$

where $\mathbf{r}_i := [r_i\ \dot{r}_i\ \ddot{r}_i]^T$. A step, a ramp, or a parabola can be generated depending on the initial values that are chosen for $\dot{r}_i$ and $\ddot{r}_i$. According to the hypothesis posed in *Problem* 1, $\dddot{r}_i = 0$.

The control law for the AVSC is defined by first consider the following change of coordinates $y_i := x_i - r_i$, $\dot{y}_i := \dot{x}_i - \dot{r}_i$, $\ddot{y}_i := \ddot{x}_i - \ddot{r}_i$, that places the system origin on the trajectory to be tracked. Due to (1.3), system (1.1) becomes

$$\mathbf{y}_{i+1} = \mathbf{A}\,\mathbf{y}_i + \mathbf{b}\,u_i\,, \tag{1.4}$$

where $\mathbf{A}$ and $\mathbf{b}$ coincide with (1.2), while $\mathbf{y}_i := [y_i\ \dot{y}_i\ \ddot{y}_i]^T$.

A further change of coordinates $\mathbf{y}_i = \mathbf{W}\,\mathbf{z}_i$, where

$$\mathbf{W} := \begin{bmatrix} T^3 & -T^3 & \frac{T^3}{6} \\ 0 & T^2 & -\frac{T^2}{2} \\ 0 & 0 & T \end{bmatrix}, \tag{1.5}$$

is required to eliminate sampling time T from matrices $\mathbf{A}$ and $\mathbf{b}$. System (1.4) becomes

$$\mathbf{z}_{i+1} = \mathbf{A}_d\,\mathbf{z}_i + \mathbf{b}_d\,u_i\,, \tag{1.6}$$

where $\mathbf{z}_i := [z_{1,i}\ z_{2,i}\ z_{3,i}]^T$ and

$$\mathbf{A}_d = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b}_d = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{1.7}$$

Matrix $\mathbf{W}$ is non singular and can be evaluated by means the following expression

$$\mathbf{W}^{-1} = \begin{bmatrix} \frac{1}{T^3} & \frac{1}{T^2} & \frac{1}{3T} \\ 0 & \frac{1}{T^2} & \frac{1}{2T} \\ 0 & 0 & \frac{1}{T} \end{bmatrix}. \tag{1.8}$$

Therefore, the inverse transformation $\mathbf{z}_i = \mathbf{W}^{-1}\,\mathbf{y}_i$ exists with certainty.

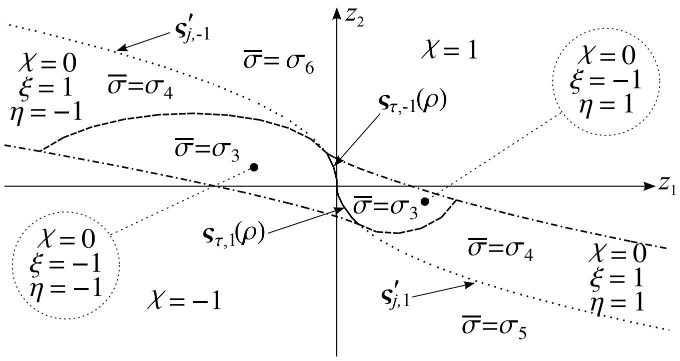
$\varsigma'_{j,\text{-}1}$
z_2
$\chi=0$
$\xi=1$
$\eta=-1$
$\overline{\sigma}=\sigma_4$
$\overline{\sigma}=\sigma_6$
$\chi=1$
$\varsigma_{\tau,\text{-}1}(\rho)$
$\chi=0$
$\xi=-1$
$\eta=1$
$\overline{\sigma}=\sigma_3$
z_1
$\overline{\sigma}=\sigma_3$
$\chi=0$
$\xi=-1$
$\eta=-1$
$\varsigma_{\tau,1}(\rho)$
$\chi=-1$
$\overline{\sigma}=\sigma_4$
$\chi=0$
$\xi=1$
$\eta=1$
$\varsigma'_{j,1}$
$\overline{\sigma}=\sigma_5$

Figure 1.2: Partitions of the (z_1, z_2)-plane that are used to select the most appropriate sliding surface. Curve $\varsigma'_{j,1}$ is given by the intersection between σ_4 and σ_5, while $\varsigma'_{j,-1}$ is given by the intersection between σ_4 and σ_6. Curves $\varsigma_{\tau,\eta}$ represent the borderlines between the areas where $\eta = 1$ and $\eta = -1$.

The proposed controller is designed to force state $\mathbf{z}$ toward the origin in minimum time by means of transients which fulfill the given constraints on velocity, acceleration, and jerk. If this result is achieved, then, $\mathbf{y} = \mathbf{W}\,\mathbf{z} = \mathbf{0}$ and, in turn, x, as desired, hangs reference r.

The AVSC, that is used to control system (1.6), depends on system state $\mathbf{z}$ and on the input reference signal, i.e., on $r, \dot{r}, \ddot{r}$. In particular, the following sliding mode control law has been explicitly designed to solve *Problem* 1 (sampling time i has been omitted in the following for conciseness, so that, e.g., command signal u_i is simply indicated as u)

$$u := \begin{cases} -U^{-}\mathrm{sat}\left(\dfrac{z_3-\sigma}{U^{-}}\right) & \text{if} \quad z_3-\sigma \geq 0 \\ -U^{+}\mathrm{sat}\left(\dfrac{z_3-\sigma}{U^{+}}\right) & \text{if} \quad z_3-\sigma < 0 \end{cases}, \tag{1.9}$$

where σ is a Sliding Surface (SS) that depends on z_1 and z_2, while $\mathrm{sat}(\cdot)$ represents a function that saturates its argument to ± 1. Evidently, due to (1.9), the jerk constraint is certainly satisfied since $u \in [U^{-}, U^{+}]$. Equation (1.9) also defines a Boundary Layer (BL) around the SS. Its upper bound is equal to $\sigma - U^{-}$, while its lower bound is equal to $\sigma - U^{+}$.

The SS has a variable structure that depends on $\mathbf{z}$. It is obtained by switching among several SSs according to the following rule

$$\sigma := \begin{cases} \sigma_1 & \text{if} \quad \sigma_1 < \overline{\sigma} \\ \overline{\sigma} & \text{if} \quad \sigma_2 \leq \overline{\sigma} \leq \sigma_1 \\ \sigma_2 & \text{if} \quad \overline{\sigma} < \sigma_2 \end{cases} . \tag{1.10}$$

The equations of σ_1 and σ_2 will soon be given, while $\overline{\sigma}$ is composed itself by several SSs. More precisely, the (z_1, z_2)-space, i.e., the space in which the first two components of the state span, is partitioned into the macro-areas that are shown in Fig. 1.2 and each of them is unambiguously identified by means of three parameters $\chi = -1, 0, 1$, $\eta = \pm 1$, and $\xi = \pm 1$. Depending on the state location, the most appropriate SS is chosen according to the following rules

$$\overline{\sigma} := \begin{cases} \sigma_3 & \text{if} \quad \chi = 0 \;\&\; \xi = -1 \\ \sigma_4 & \text{if} \quad \chi = 0 \;\&\; \xi = 1 \\ \sigma_5 & \text{if} \quad \chi = 1 \\ \sigma_6 & \text{if} \quad \chi = -1 \end{cases} . \tag{1.11}$$

Surface $\overline{\sigma}$, that is obtained by composing σ_3, σ_4, σ_5, and σ_6, is continuous and covers the whole (z_1, z_2)-space. All SSs are defined in the following:

1) Surfaces σ_1 and σ_2 (in the following $n = 1, 2$)

$$\sigma_n := -\frac{\gamma_n}{m_n} - \frac{m_n - 1}{2} \kappa_n + \overline{z}_3^+ , \tag{1.12}$$

where γ_n, κ_n, and m_n are evaluated as follows

$$z_3^- := \tfrac{S^- - \ddot{r}}{T}, \quad z_3^+ := \tfrac{S^+ - \ddot{r}}{T}, \tag{1.13}$$

$$\overline{\mathbf{z}}^+ := [\overline{z}_2^+ \; \overline{z}_3^+]^T := \left[\left(\tfrac{R^+ - \dot{r}}{T^2} - \tfrac{\ddot{r}}{2T} \right) \; - \tfrac{\ddot{r}}{T} \right]^T, \tag{1.14}$$

$$\overline{\mathbf{z}}^- := [\overline{z}_2^- \; \overline{z}_3^-]^T := \left[\left(\tfrac{R^- - \dot{r}}{T^2} - \tfrac{\ddot{r}}{2T} \right) \; - \tfrac{\ddot{r}}{T} \right]^T, \tag{1.15}$$

$$\hat{z}_3^+ := z_3^+ - \overline{z}_3^+ , \quad \hat{z}_3^- := z_3^- - \overline{z}_3^- ,$$

$$\hat{z}_2^+ := -\left\lceil -\tfrac{\hat{z}_3^+}{U^-} \right\rceil \left[\hat{z}_3^+ + \tfrac{U^-}{2} \left(\left\lceil -\tfrac{\hat{z}_3^+}{U^-} \right\rceil - 1 \right) \right] ,$$

$$\hat{z}_2^- := -\left\lceil -\tfrac{\hat{z}_3^-}{U^+} \right\rceil \left[\hat{z}_3^- + \tfrac{U^+}{2} \left(\left\lceil -\tfrac{\hat{z}_3^-}{U^+} \right\rceil - 1 \right) \right] ,$$

$$
\begin{aligned}
d_1 &:= z_2 - \overline{z}_2^+ \,, \quad d_2 := z_2 - \overline{z}_2^- \,, \\
\gamma_n &:= \begin{cases} \hat{z}_2^+ & \text{if } d_n < \hat{z}_2^+ \\ d_n & \text{if } \hat{z}_2^+ \leq d_n \leq \hat{z}_2^- \\ \hat{z}_2^- & \text{if } d_n > \hat{z}_2^- \end{cases} \,, \\
\kappa_n &:= \begin{cases} U^- & \text{if } \gamma_n \leq 0 \\ U^+ & \text{if } \gamma_n > 0 \end{cases} \,, \\
m_n &:= \left\lfloor \tfrac{1}{2} + \sqrt{\tfrac{1}{4} + 2\left|\tfrac{\gamma_n}{\kappa_n}\right|} \right\rfloor \,.
\end{aligned}
$$

and where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ respectively return the ceil and the floor of their arguments.

2) Surface σ_3

$$\sigma_3 := -\frac{2}{h(h+\tau)} z_1 - \frac{2h+\tau-1}{h(h+\tau)} z_2 - \frac{\tau(1-\tau^2)}{6h(h+\tau)} \alpha - \frac{2h^3 - 3h^2 + h + 3h^2\tau - 3h\tau}{6h(h+\tau)} \beta \,, \tag{1.16}$$

where

$$[\alpha \ \beta] := \begin{cases} [U^- \ U^+] & \text{if } \eta = 1 \\ [U^+ \ U^-] & \text{if } \eta = -1 \end{cases} \,, \tag{1.17}$$

while $h, \tau \in \mathbb{N}^+$ and $\eta = \pm 1$ are parameters that depend on z_1 and z_2. The role and the meaning of such parameters will be discussed in §1.3. The procedure that is used to devise σ_3 is proposed in the same Section.

3) Surface σ_4

$$\sigma_4 := \frac{n_1 \, \alpha + n_2 \, \beta + n_3 \, z_1 + n_4 \, z_2}{6 \, d_1} \,, \tag{1.18}$$

where

$$
\begin{aligned}
n_1 \quad &:= \quad 6\left[j^2 + \hat{\tau}(\hat{\tau}-1) + (2\hat{\tau}-1)j\right]\hat{\rho}^2 + 2\hat{\tau}\hat{\rho}\left[6j^2 + 9j(\hat{\tau}-1) + 2\hat{\tau}^2 + 1 - 9\hat{\tau}\right] \\
&\qquad +6\left[j^2 + (\hat{\tau}-2)j\right]\hat{\tau}^2 + \hat{\tau}^4 - 6\hat{\tau}^3 + 5\hat{\tau}^2 \,, \\
n_2 \quad &:= \quad 6h(1-h)(j\hat{\rho} + \hat{\rho}\,\hat{\tau} + j\hat{\tau}) - 2\hat{\rho}h(2 - 3h + h^2) - h\hat{\tau}(7 - 9h + 2h^2) \\
&\qquad +3\hat{\tau}^2 h(1-h) \,, \\
n_3 \quad &:= \quad -12(\hat{\tau} + \hat{\rho}) \,, \\
n_4 \quad &:= \quad -12(h+j)(\hat{\tau} + \hat{\rho}) + 12\hat{\rho}(1-\hat{\tau}) + 6\hat{\tau}(3-\hat{\tau}) \,, \\
d_1 \quad &:= \quad h\left[\hat{\tau}(\hat{\tau} - 2 + h + 2j) + \hat{\rho}(2j - 1 + h + 2\hat{\tau})\right] \,,
\end{aligned}
$$

and where α and β are defined according to (1.17), while $\hat{\tau}$ and $\hat{\rho}$ are evaluated by means of the following expressions

$$z^* := \begin{cases} z_3^+ & \text{if } \eta = 1 \\ z_3^- & \text{if } \eta = -1 \end{cases} , \tag{1.19}$$

$$\hat{\tau} := \left\lfloor -\tfrac{z^*}{\alpha} \right\rfloor , \tag{1.20}$$

$$\hat{\rho} := -\tfrac{z^*}{\alpha} - \hat{\tau} . \tag{1.21}$$

Parameters $h, j \in \mathbb{N}^+$, and $\eta = \pm 1$ only depend on z_1 and z_2. Their role and meaning will be discussed in §1.4, together with the method that is used to devise σ_4. It is worth to point out that, because of (1.19)–(1.21), $\hat{\rho} \in [0,1)$, while $\hat{\tau} \in \mathbb{N}^+$.

4) Surfaces σ_5 and σ_6

$$\sigma_5 := z_3^+ , \tag{1.22}$$

$$\sigma_6 := z_3^- . \tag{1.23}$$

The role of each SS is explained by means of the simple rest-to-rest transient that is shown in Fig. 1.3. The following limits have been assumed: $U^- = -10$ m s^{-3}, $U^+ = 20$ m s^{-3}, $S^- = -3.9$ m s^{-2}, $S^+ = 1.9$ m s^{-2}, $R^- = -0.95$ m s^{-1}, and $R^+ = 1.4$ m s^{-1}. Surface σ_3 is used to drive the system, in minimum time, toward the origin compatibly with the jerk constraint. As shown in the solid curves in Fig. 1.3, surface σ_3 does not account for the velocity and the acceleration limits, so that such bounds could be violated. The fulfillment of the acceleration constraint can potentially be achieved by using σ_5 and σ_6 in any area of the (z_1, z_2)-space in which σ_3 is unfeasible. Indeed, z_3^+ and z_3^-, that are defined according to (1.13), represent the equivalent bounds, in the $\mathbf{z}$-space, of S^+ and S^-. The dashed transients in Fig. 1.3 highlight that this solution, that is similar to the one that was used in [21, 22], has a drawback: An overshoot can appear. The filter will be used to generate reference signals for industrial machines, so that such overshoot is clearly undesired. Moreover, the transient is not minimum-time. Surface σ_4, when used in conjunction with σ_3, as shown by the dotted lines in Fig. 1.3, eliminates both issues. The last two surfaces, i.e., σ_1 and σ_2 defined in (1.12), are used to guarantee the fulfillment of the velocity limits: The

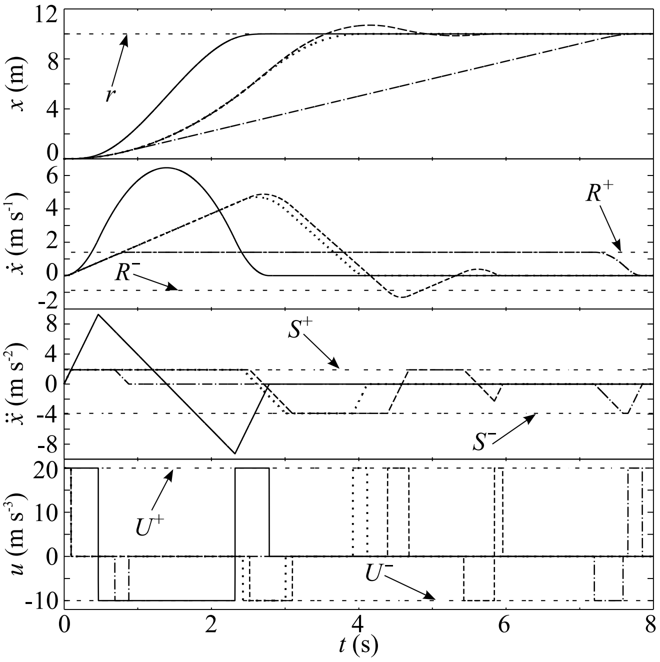

Figure 1.3: Comparison between transients that are achieved by using σ_3 (solid lines), σ_3, σ_5, and σ_6 (dashed lines), σ_3, σ_4, σ_5, and σ_6 (dotted lines), all the SSs (dash-dotted lines).

dash-dotted lines in Fig. 1.3 correspond to the system response that is obtained when all the surfaces are simultaneously used.

The role and the convergence properties of all the surfaces will be deeper analyzed in next Sections.

1.3 Design and convergence properties of σ_3

Surface σ_3 is designed to drive the system state toward the origin in minimum time by fulfilling, at the same time, the jerk constraint. Differently from the SS that was proposed in [20], it is able to handle asymmetric jerk bounds. The design and convergence properties of σ_3 are illustrated by forcing

$$\sigma = \sigma_3 \tag{1.24}$$

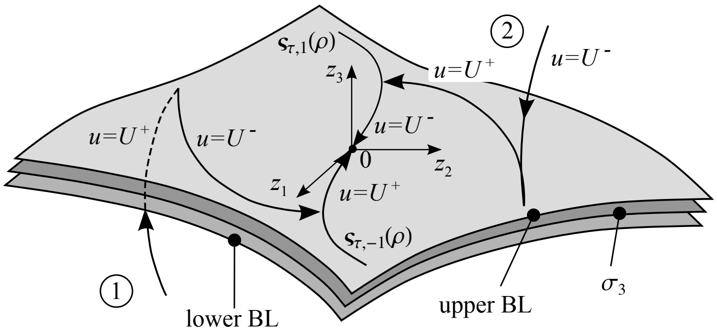

Figure 1.4: Schematic representation of some transients toward the origin. Surface σ_3 is shown together with its BL.

in (1.10) and by using (1.24) in (1.9) to drive the system (1.6). Equations (1.16) and (1.17) determine the shape of σ_3, while (1.9) wraps the SS within an appropriate BL.

The optimality of the transients from any generic state $\mathbf{z}$ can be understood with the aid of Fig. 1.4, that schematically shows two typical trajectories toward the origin. If the initial state is located below σ_3 (see Transient 1 in Fig. 1.4), the control law returns $u = U^+$, so that the z_3 component of $\mathbf{z}$, owing to (1.6) and (1.7), increases: σ_3 covers the whole (z_1, z_2)-space, so that the BL is certainly reached in minimum time. Once the state is inside the BL, command signal becomes, as it will soon be shown, $u = U^-$ and the state reaches curve $\boldsymbol{\varsigma}_{\tau,\eta}(\rho)$ where a new switch occurs. The origin is finally approached with $u = U^+$. According to the Pontryagin's maximum principle, the convergence is obtained in minimum time, since u is bang-bang and two switches have occurred. Similar transients are obtained for initial states that lie above σ_3 (see Transient 2 in Fig. 1.4).

The design of σ_3 and the proofs of the optimality of the transients from any generic state $\mathbf{z}$ within the BL to the origin are described as follows. First of all, it is possible to prove that all points $\mathbf{p}_{h,\tau,\eta}$, from which the origin can be reached in minimum time by means of a bang-bang command signal and with a single switch, have equation

$$\mathbf{p}_{h,\tau,\eta} = \begin{bmatrix} -\left[\frac{\tau(\tau-1)(\tau-2)}{6} + \frac{h\tau(h+\tau-2)}{2}\right]\alpha - \frac{h(h-1)(h-2)}{6}\beta \\ \left[\frac{\tau(\tau-1)}{2} + h\,\tau\right]\alpha + \frac{h(h-1)}{2}\beta \\ -\tau\,\alpha - h\,\beta \end{bmatrix} \tag{1.25}$$

where α and β, according to (1.17), depend on η. To this purpose, a first set of points, indicated in the following by $\mathbf{p}_{\tau,\eta}$, has been individuated by integrating backward system (1.6) from the origin. Two situations have been considered, each of them is denoted by a different value of η: If $\eta = 1$, then the system is driven with command signal $u = U^-$, while $u = U^+$ if $\eta = -1$. By means of this procedure, the following points have been obtained

$$\mathbf{p}_{\tau,\eta} = \left[-\frac{\tau}{6}\left(\tau^2 - 3\tau + 2\right)\alpha \quad \frac{\tau}{2}(\tau-1)\,\alpha \quad -\tau\,\alpha\right]^T,$$

where α depends on η because of (1.17), while $\tau \in \mathbb{N}^+$ indicates the number of back integrations that have occurred. Evidently, from any point $\mathbf{p}_{\tau,\eta}$, the origin can be reached in minimum time with τ steps by applying the maximum admissible command signal, i.e., $u = U^-$ if $\eta = 1$ or $u = U^+$ if $\eta = -1$.

From any point $\mathbf{p}_{\tau,\eta}$ a new stage of backward integrations returns points $\mathbf{p}_{h,\tau,\eta}$ as defined by (1.25). They are obtained by switching the command signal, so that from points $\mathbf{p}_{\tau,-1}$ command signal $u = U^-$ is assumed while, viceversa, $u = U^+$ from points $\mathbf{p}_{\tau,1}$. Evidently, bearing in mind the definitions of α and β given by (1.17), from any point $\mathbf{p}_{h,\tau,\eta}$ the origin is reached, by construction, by first applying command signal $u = \beta$, for $h \in \mathbb{N}^+$ steps and, then, by assuming $u = \alpha$ for $\tau \in \mathbb{N}^+$ steps. The control is clearly bang-bang, a single switch occurs, so that it is possible to assert that the transient toward the origin is, according to the Pontryagin's maximum principle, minimum-time.

Points $\mathbf{p}_{h,\tau,\eta}$, as shown in Fig. 1.5, completely cover the (z_1, z_2)-space, that is partitioned into two sectors depending on η. The borderline between the two sectors is given by curve $\boldsymbol{\varsigma}_{\tau,\eta}(\rho)$. The vertexes of σ_3 are obtained by adding vector $[0\,0\,\beta]^T$ to points $\mathbf{p}_{h,\tau,\eta}$. σ_3 is a composite SS that is made of flat quadrangles and that, evidently, covers the whole (z_1, z_2)-plane. Each quadrangle is indexed by h, τ, η. Given

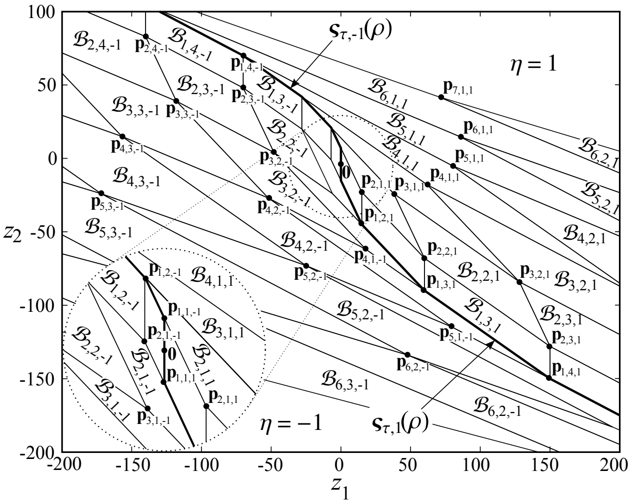

Figure 1.5: Projection of σ_3 on the (z_1, z_2)-plane. Curve $\boldsymbol{\varsigma}_{\tau,\eta}(\rho)$ separates the two zones that admit different values of η.

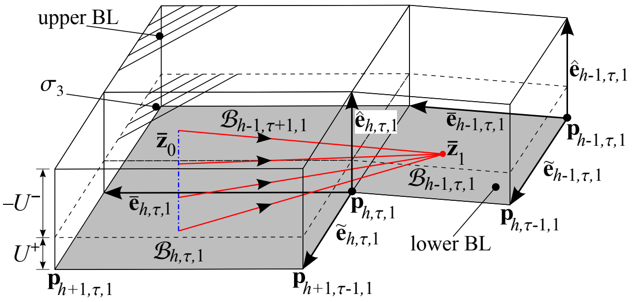
upper BL
σ_3
$\hat{\mathbf{e}}_{h\text{-}1,\tau,1}$
$\mathcal{B}_{h\text{-}1,\tau+1,1}$
$\hat{\mathbf{e}}_{h,\tau,1}$
$\bar{\mathbf{e}}_{h\text{-}1,\tau,1}$
$\mathbf{p}_{h\text{-}1,\tau,1}$
$\bar{\mathbf{z}}_0$
$\bar{\mathbf{z}}_1$
$\mathcal{B}_{h\text{-}1,\tau,1}$
$\tilde{\mathbf{e}}_{h\text{-}1,\tau,1}$
$-U^-$
$\bar{\mathbf{e}}_{h,\tau,1}$
$\mathbf{p}_{h,\tau,1}$
$\mathbf{p}_{h,\tau\text{-}1,1}$
$\tilde{\mathbf{e}}_{h,\tau,1}$
lower BL
U^+
$\mathcal{B}_{h,\tau,1}$
$\mathbf{p}_{h+1,\tau,1}$
$\mathbf{p}_{h+1,\tau\text{-}1,1}$

Figure 1.6: Single step evolution starting from box $\mathscr{B}_{h,\tau,1}$.

any point in the (z_1, z_2)-plane, the corresponding value of σ_3 is found by first individuating the quadrangle h, τ, η in which it is located – to this purpose, techniques similar to those proposed in [28] can be adopted – and, then, by using (1.16) and (1.17).

Equations (1.9), (1.16), (1.17), and (1.24) associate to each point $\mathbf{p}_{h,\tau,\eta}$ a planar sliding surface σ_3 and its BL. In particular, as shown in Fig. 1.6, a box $\mathscr{B}_{h,\tau,\eta}$, which upper/lower surfaces are given by the borders of the BL, is associated to each point

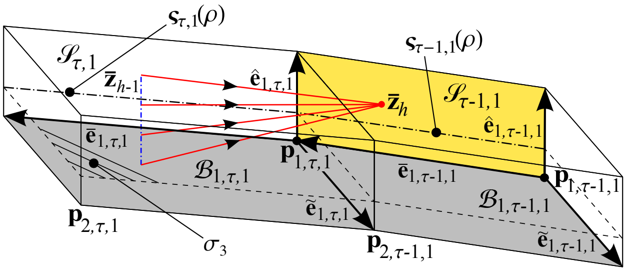
$\mathbf{s}_{\tau,1}(\rho)$
$\mathscr{S}_{\tau,1}$
$\bar{\mathbf{z}}_{h\text{-}1}$
$\hat{\mathbf{e}}_{1,\tau,1}$
$\bar{\mathbf{z}}_h$
$\mathbf{s}_{\tau-1,1}(\rho)$
$\mathscr{S}_{\tau\text{-}1,1}$
$\hat{\mathbf{e}}_{1,\tau\text{-}1,1}$
$\bar{\mathbf{e}}_{1,\tau,1}$
$\mathbf{p}_{1,\tau,1}$
$\mathcal{B}_{1,\tau,1}$
$\bar{\mathbf{e}}_{1,\tau\text{-}1,1}$
$\mathbf{p}_{1,\tau\text{-}1,1}$
$\mathcal{B}_{1,\tau\text{-}1,1}$
$\tilde{\mathbf{e}}_{1,\tau,1}$
$\mathbf{p}_{2,\tau,1}$
σ_3
$\mathbf{p}_{2,\tau\text{-}1,1}$
$\tilde{\mathbf{e}}_{1,\tau\text{-}1,1}$

Figure 1.7: Single step evolution form the box $\mathscr{B}_{1,\tau,1}$

$\mathbf{p}_{h,\tau,\eta}$. $\mathscr{B}_{h,\tau,\eta}$ can be formally defined as follows

$$\mathscr{B}_{h,\tau,\eta} := \{\mathbf{z} : \mathbf{z} = \mathbf{p}_{h,\tau,\eta} + \lambda\, \bar{\mathbf{e}}_{h,\tau,\eta} + \mu\, \tilde{\mathbf{e}}_{h,\tau,\eta} + \nu\, \hat{\mathbf{e}}_{h,\tau,\eta};\;\; \lambda, \mu, \nu \in [0,1)\} \tag{1.26}$$

where $\mathbf{p}_{h,\tau,\eta}$ is given by (1.25), while vectors $\bar{\mathbf{e}}_{h,\tau,\eta}, \tilde{\mathbf{e}}_{h,\tau,\eta}$, and $\hat{\mathbf{e}}_{h,\tau,\eta}$ are defined as follows

$$\bar{\mathbf{e}}_{h,\tau,\eta} := \alpha \left[-\tfrac{1}{2} \left[\tau(\tau-1) + h(h-1) + 2h\tau \right] \quad h+\tau \quad -1 \right]^T, \tag{1.27}$$

$$\tilde{\mathbf{e}}_{h,\tau,\eta} := (\alpha - \beta) \left[\tfrac{1}{2} h(h-1) \quad -h \quad 1 \right]^T, \tag{1.28}$$

$$\hat{\mathbf{e}}_{h,\tau,\eta} := \left[0 \quad 0 \quad \beta - \alpha \right]^T. \tag{1.29}$$

The vectors placement is shown in Fig. 1.6 for $\eta = 1$. Practically, vectors (1.27)–(1.29) represent a non-orthogonal reference frame that can be used to describe any point $\mathbf{z} \in \mathscr{B}_{h,\tau,\eta}$. More precisely, any point $\mathbf{z} \in \mathscr{B}_{h,\tau,\eta}$ can be alternatively represented by means of a vector of six elements $\bar{\mathbf{z}} := [h\ \tau\ \eta\ \lambda\ \mu\ \nu]^T$: The first three coordinates individuate the box, the last three define the position inside the box. In this representation, the origin assumes coordinates $\bar{\mathbf{z}} := [1\ 1\ \eta\ 0\ \frac{-\alpha}{\beta-\alpha}\ \frac{\beta}{\beta-\alpha}]^T$.

Bearing in mind these premises, it is possible to prove the optimality of the transients from any point that is located inside the BL. In particular, from any initial state $\mathbf{z}_0 \in \mathscr{B}_{h,\tau,\eta}$ or, equivalently, from $\bar{\mathbf{z}}_0 = [h\ \tau\ \eta\ \lambda\ \mu\ \nu]^T$, with $h, \tau \in \mathbb{N}^+$, the origin is certainly reached, if system (1.6) is controlled by means of (1.9), (1.16), (1.17), and

(1.24), according to the following sequence of steps:

i	$\bar{\mathbf{z}}_i$	u_{i+1}
0	$[h\ \tau\ \eta\ \lambda\ \mu\ \nu]^T$	$\nu\alpha+(1-\nu)\beta$
1	$[(h-1)\ \tau\ \eta\ \lambda\ \mu\ 0]^T$	β
2	$[(h-2)\ \tau\ \eta\ \lambda\ \mu\ 0]^T$	β
$\vdots$		β
$h-1$	$[1\ \tau\ \eta\ \lambda\ \mu\ 0]^T$	β
h	$[1\ (\tau-1)\ \eta\ \lambda\ 0\ (1-\mu)]^T$	$\mu\beta+(1-\mu)\alpha$
$h+1$	$[1\ (\tau-2)\ \eta\ \lambda\ 0\ 1]^T$	α
$\vdots$		α
$h+\tau-2$	$[1\ 1\ \eta\ \lambda\ 0\ 1]^T$	α
$h+\tau-1$	$[1\ 1\ \eta\ 0\ \frac{\alpha(\lambda-1)}{\beta-\alpha}\ 1]^T$	α
$h+\tau$	$[1\ 1\ \eta\ 0\ \frac{-\alpha}{\beta-\alpha}\ \frac{\beta-\lambda\alpha}{\beta-\alpha}]^T$	$\lambda\alpha$
$h+\tau+1$	$[1\ 1\ \eta\ 0\ \frac{-\alpha}{\beta-\alpha}\ \frac{\beta}{\beta-\alpha}]^T$	

Fig. 1.7 shows the transient that drives the system to switch the command law. More precisely, it is depicted the h-th step of the above sequence: After $h-1$ steps, the state reaches the box $\mathscr{B}_{1,\tau,\eta}$ with command law $u=\beta$ and the execution of a further single evolution step projects the state on the lateral surface of $\mathscr{B}_{1,\tau-1,\eta}$ box, alternatively indicated with $\bar{\mathbf{z}}_h=[1\ (\tau-1)\ \eta\ \lambda\ 0\ (1-\mu)]^T$: In this case, the command law is equal to $u=\mu\beta+(1-\mu)\alpha$ which indicates the switching instant. This lateral surface, pointed out in Fig. 1.7 with $\mathscr{S}_{\tau,1}$ for $\eta=1$, contains the curve $\varsigma_{\tau,\eta}(\rho)$, i.e., the borderline between two sectors characterized with different values of η. Therefore, $\bar{\mathbf{z}}_h\in\mathscr{S}_{\tau-1,1}\ \forall\lambda,\mu,\nu\in[0,1)$. Fig. 1.7 also depicts that all points $\mathscr{B}_{1,\tau,\eta}$, that share the same values of λ and μ, are mapped into the same point $\bar{\mathbf{z}}_h\in\mathscr{S}_{\tau-1,\eta}$, independently from ν. The next step, indicated in the above sequence with $i=h+1$, places the state on the upper edge of the surface $\mathscr{S}_{\tau-2,\eta}$ and the reached point is indicated with $\bar{\mathbf{z}}_{h+1}=[1\ (\tau-2)\ \eta\ \lambda\ 0\ 1]^T$.

The final convergence toward the origin, starting from a generic state $\mathbf{z}\in\mathscr{B}_{1,1,\eta}$, is shown with the aid of Fig. 1.8. The final transient corresponds to the last three steps in the above sequence. Fig. 1.8, in case of $\eta=1$, shows that independently from ν,

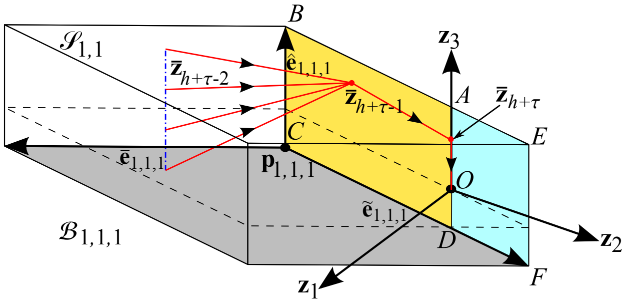
B
$\mathscr{S}_{1,1}$
$\mathbf{z}_3$
$\hat{\mathbf{e}}_{1,1,1}$
$\bar{\mathbf{z}}_{h+\tau\text{-}2}$
A
$\bar{\mathbf{z}}_{h+\tau}$
$\bar{\mathbf{z}}_{h+\tau\text{-}1}$
C
E
$\bar{\mathbf{e}}_{1,1,1}$
$\mathbf{p}_{1,1,1}$
O
$\tilde{\mathbf{e}}_{1,1,1}$
$\mathscr{B}_{1,1,1}$
D
$\mathbf{z}_2$
F
$\mathbf{z}_1$

Figure 1.8: Final evolution steps: If the system enters into box $\mathscr{B}_{1,1,1}$, it is forced in three steps toward the origin. Surface ABCD is contained into the plane individuated by vectors $\mathbf{z}_2$ and $\mathbf{z}_3$

the state evolves in one step in $\bar{\mathbf{z}}_{h+\tau-1}$ which lies on the surface ABCD. A further step drives the system, with command law $u = \alpha$, to point $\bar{\mathbf{z}}_{h+\tau}$ that lies on segment OA. The last transient drives the state toward the origin: The associated command law is $u = \lambda\alpha$. If $\eta = -1$, an analogous transient occurs: The state reaches first a point that lies on the surface ADEF, successively the AVSC drives the state on the segment OD with command law $u = \alpha$ and, finally, the state reaches the origin with command law $u = \lambda\alpha$.

After an additional analysis of the sequence of steps, it is evident that $h+\tau+1$ steps are required, i.e., only one more step with respect to the optimal transient from $\mathbf{p}_{h,\tau,\eta}$, to reach the origin. According to the Pontryagin's maximum principle, a single switch occurs and the command signal, apart from switching instants, is always equal to β or to α, i.e., by virtue of (1.17), it is equal to U^+ or to U^-. It is worth to point out that, at the switching instants, the command signal cannot be exactly equal to β or α because the switching times of discrete-time systems are fixed and cannot be freely selected. If the initial state is equal to $\mathbf{p}_{h,\tau,\eta}$, i.e., $\bar{\mathbf{z}}_0 = [h\ \tau\ \eta\ 0\ 0\ 0]^T$, the sequence converges to the origin in $h+\tau$ steps and u is, as desired, exactly bang-bang.

1.4 Design and convergence properties of σ_4

Surface σ_3 manages neither the velocity nor the acceleration constraints that, consequently, could be violated. The acceleration constraint could potentially be satisfied

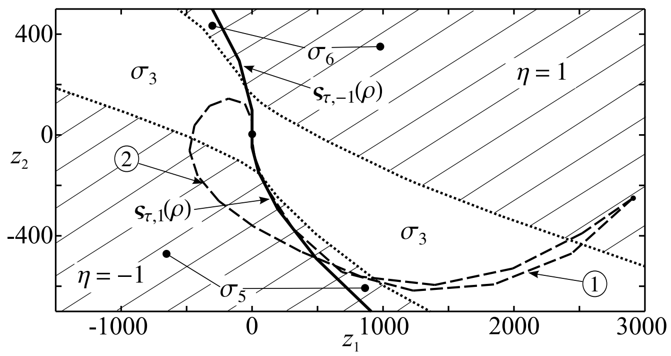

Figure 1.9: State transient in the (z_1, z_2)-space that is obtained by using σ_3 (Transient 1) and a combination of σ_3, σ_5, and σ_6 (Transient 2). Highlighted areas refer to regions in which σ_3 does not satisfy the acceleration constraint.

by simply bounding the third component of $\mathbf{z}$ within interval $[z_3^+, z_3^-]$. It was early anticipated that this result can be achieved by adopting σ_3 only in the areas of the (z_1, z_2)-space in which it is feasible, and by using σ_5 and σ_6 in the other zones. As shown in §1.2, this rough approach can cause overshoots and suboptimal transients. For this reason, in any zone of the (z_1, z_2)-plane in which σ_3 would lead to unfeasible accelerations, an alternative surface σ_4 is used.

The resulting partitions of the (z_1, z_2)-space are shown in Fig. 1.9. In the same figure, Transient 1 represents a typical state evolution that is obtained by only using σ_3, while Transient 2 is obtained by also adopting σ_5 and σ_6: In the first case, transitions are minimum-time, but the state enters in zones where σ_3 violates the acceleration constraint, in the second case, the given bounds are satisfied, but the transient is not time optimal and the convergence is obtained with an overshoot. The overshoot is generated every time the state crosses curve $\boldsymbol{\varsigma}_{\tau,\eta}(\rho)$, i.e., the curve obtained from the intersection between surfaces $\mathscr{S}_{\tau,\eta}$ and the SS σ_3 (see also Figs. 1.7 and 1.11).

In order to avoid the overshoot problem a new surface σ_4 is devised: σ_4 drives the state, in minimum-time and by satisfying the acceleration bound, toward regions where σ_3 cannot lose the feasibility. The behavior of σ_4 is highlighted with the aid of Fig. 1.10: The AVSC uses σ_4, σ_5, and σ_6 in order to drive the state toward σ_3 in minimum-time. With the introduction of σ_4, the partitioning scheme of the (z_1, z_2)-space coincides with the one shown in Fig. 1.2.

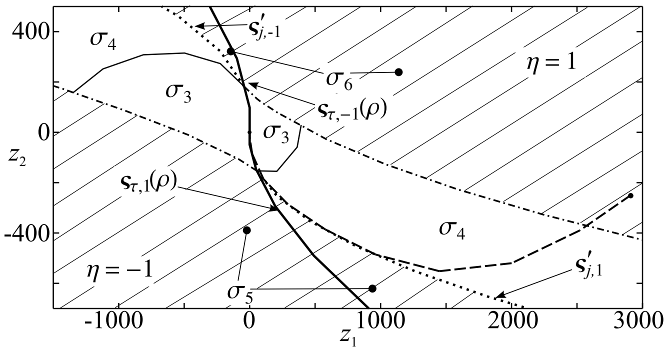

Figure 1.10: State transient in the (z_1, z_2)-space that is obtained by using σ_3, σ_4, σ_5, and σ_6. Highlighted areas refer to regions in which neither σ_3 nor σ_4 satisfy the acceleration constraint.

Surface σ_4, similarly to σ_3, has been planned by first individuating a set of points $\mathbf{q}_{h,j,\eta}$ from which the origin can be reached in minimum time by means of a trajectory that fulfills both the jerk and the acceleration constraints. The positions of such points are given by (1.30).

$$\mathbf{q}_{h,j,\eta} = \begin{bmatrix} \alpha\left\{\frac{\left[6(h+j)(2-\hat{\rho})-3(h+j)^2+9\hat{\rho}-11\right]\hat{\tau}}{6} + \frac{\left[(3-2h)j-j^2-(h-1)(h-2)\right]\hat{\rho}}{2}\right\} \\ \hfill -\alpha\left\{\frac{\hat{\tau}^3}{6} + \frac{(h+j+\hat{\rho}-2)\hat{\tau}^2}{2}\right\} - \frac{\beta h}{6}(h-1)(h-2) \\ \alpha\left[\frac{1}{2}\hat{\tau}(\hat{\tau}-3) + (j+\hat{\tau}+h-1)\hat{\rho} + \hat{\tau}(h+j)\right] + \frac{1}{2}\beta h(h-1) \\ -\alpha(\hat{\tau}+\hat{\rho}) - \beta h \end{bmatrix}, \quad (1.30)$$

The Pontryagin's maximum principle suggests that, because of the additional constraint, optimal transients must be bang-zero-bang. Fig. 1.11, which shows a situation in which $\eta = 1$, can be used to explain the synthesis of σ_4. By integrating backward from the origin system (1.6), with $u = U^-$, a monotonically increasing curve is obtained. An indefinite execution of the back integration process would clearly lead to the violation of upper bound z_3^+. In order preserve feasibility, in proximity of z_3^+ the command signal is switched to $u = 0$, and a set of points $\mathbf{r}_{j,1}$ is obtained. They are characterized by a third component equal to z_3^+, i.e., they are all strictly feasible with respect to the acceleration bound. Points $\mathbf{q}_{h,j,1}$ are finally obtained by means of a new stage of back integrations from $\mathbf{r}_{j,1}$, which is made by assuming

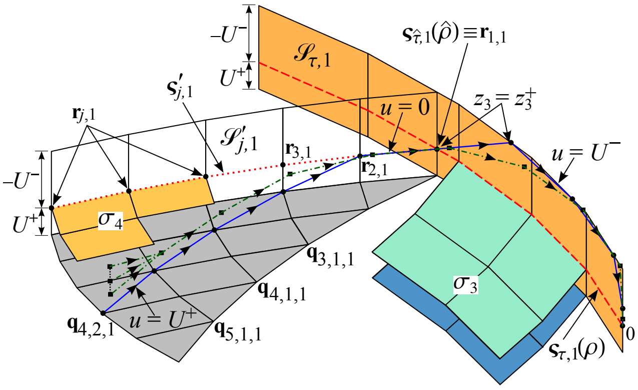

Figure 1.11: 3D view of a transient toward the origin for $\eta = 1$ starting from $\mathbf{q}_{2,4,1}$ (solid line) and from a generic point inside the BL (dash-dotted line). Point $\boldsymbol{\varsigma}_{\hat{\tau},1}(\hat{\rho})$ is reached in 5 steps, then the origin is gained by means of σ_3. The same number of steps is required for the second transient. Upper bound z_3^+ is never violated.

$u = U^+$. An analogous sequence of backward integrations is used for $\eta = -1$ in order to devise points $\mathbf{q}_{h,j,-1}$, but in this second case the command sequence becomes $u = U^+ \Rightarrow u = 0 \Rightarrow u = U^-$.

Fig. 1.11 also shows a typical approach to the origin from $\mathbf{q}_{4,2,1}$ ($h = 4$, $j = 2$, $\eta = 1$). If system (1.6) is driven with $u = U^+$, point $\mathbf{r}_{2,1}$ is reached after h steps (in the example $h = 4$). The transient is clearly minimum-time because the maximum available jerk has been used. As previously mentioned, points $\mathbf{r}_{j,1}$ are characterized, by construction, by a third component, equal to z_3^+, i.e., equal to the maximum allowable acceleration. Evidently, all points along curve $\boldsymbol{\varsigma}'_{j,1}$, which passes through points $\mathbf{r}_{j,1}$, possess the same property. In $\mathbf{r}_{2,1}$, command signal switches to $u = 0$ in order to avoid the violation of the acceleration bound, and the state slides along $\boldsymbol{\varsigma}'_{j,1}$ toward $\boldsymbol{\varsigma}_{\hat{\tau},1}(\hat{\rho})$ that is reached with further $j - 1$ steps (in the example $j = 2$, thus a single step is required). The transient is again minimum-time, since $u = 0$ is the maximum allowable command signal which guarantees the feasibility.

From $\boldsymbol{\varsigma}_{\hat{\tau},1}(\hat{\rho})$, the origin is finally gained in $\hat{\tau} + 2$ steps ($\hat{\tau} + 1$ if $\hat{\rho} = 0$) with $u = U^-$. It is possible to prove that this final transient exactly coincides with the final transient that can be obtained by means of σ_3: For this reason, the actual implementation of the control law uses σ_3 for the final convergence to the origin. In conclusion, from any point $\mathbf{q}_{h,j,\eta}$, the origin is reached after $h+j+\hat{\tau}+1$ steps ($h+j+\hat{\tau}$ if $\hat{\rho}=0$) by means of a bang-zero-bang command signal and by fulfilling the acceleration constraint. The transient is clearly time-optimal.

Vertexes of surface σ_4 are obtained by adding $[0\ \ 0\ \ \beta]^T$ to points $\mathbf{q}_{h,j,\eta}$. By construction, for $\eta = 1$, σ_4 monotonically decreases in function of h and, moreover, $\sigma_4 \le z_3^+$, i.e., it is feasible with respect to the upper acceleration constraint. On the contrary, for $\eta = -1$, σ_4 monotonically increases in function of h and $\sigma_4 \ge z_3^-$.

By assuming a command law given by (1.9), (1.17), (1.18), and

$$\sigma = \sigma_4\,, \tag{1.31}$$

a BL is created around σ_4. In the following it is shown that, by means of such control law, the origin is reached in minimum time from any generic state $\mathbf{z}$ that is located inside the BL. Similarly to σ_3, surface σ_4 attracts the system state with a command signal that is equal to U^+ or U^-, i.e., in minimum time. The transient toward the origin starts after the BL of σ_4 has been reached. Let us prove its optimality and feasibility. Command law (1.9), (1.17), (1.18), and (1.31) associates, to each point $\mathbf{q}_{h,j,\eta}$, a box $\mathscr{W}_{h,j,\eta}$ which upper and lower surfaces coincide with the BL limits and that is formally defined as follows

$$\mathscr{W}_{h,j,\eta} := \left\{\mathbf{z} : \mathbf{z} = \mathbf{q}_{h,j,\eta} + \lambda \bar{\mathbf{e}}'_{h,j,\eta} + \mu \tilde{\mathbf{e}}'_{h,j,\eta} + \nu \hat{\mathbf{e}}'_{h,j,\eta};\ \ \lambda, \mu, \nu \in [0,1)\right\} \tag{1.32}$$

where

$$\bar{\mathbf{e}}'_{h,j,\eta} := \begin{bmatrix} (-h-\hat{\tau}-j+1)\hat{\rho} - \hat{\tau}(\frac{1}{2}\hat{\tau}+h+j-\frac{3}{2}) \\ \hat{\tau}+\hat{\rho} \\ 0 \end{bmatrix} \alpha, \tag{1.33}$$

$$\tilde{\mathbf{e}}'_{h,j,\eta} := \left[-\tfrac{1}{2}h(h-1)\beta\ \ h\beta\ \ -\beta\right]^T, \tag{1.34}$$

$$\hat{\mathbf{e}}'_{h,j,\eta} := [0\ \ 0\ \ \beta-\alpha]^T. \tag{1.35}$$

Bearing in mind (1.30)–(1.35), any point $\mathbf{z} \in \mathscr{W}_{h,j,\eta}$ can be alternative represented as $\hat{\mathbf{z}} := [h\ j\ \eta\ \lambda\ \mu\ \nu]^T$.

The optimality of the transients, that are obtained by means of σ_4, is proved by verifying that, from any state $\mathbf{z} \in \mathscr{W}_{h,j,\eta}$, the origin is reached in $h+j+\hat{\tau}+1$ steps, i.e., the same number of steps that are required from $\mathbf{q}_{h,j,\eta}$ with the minimum-time control, and that the command signal is bang-zero-bang. In particular, by adopting command law (1.9), (1.17), (1.18), and (1.31), system (1.6) evolves, from any initial state $\mathbf{z}_0 \in \mathscr{W}_{h,j,\eta}$, with $h, j \in \mathbb{N}^+$ or, equivalently, from $\hat{\mathbf{z}}_0 = [h\ j\ \eta\ \lambda\ \mu\ \nu]^T$, as follows:

i	$\hat{\mathbf{z}}_i$	u_{i+1}
0	$[h\ j\ \eta\ \lambda\ \mu\ \nu]^T$	$\nu\alpha+(1-\nu)\beta$
1	$[(h-1)\ j\ \eta\ \lambda\ \mu\ 0]^T$	β
2	$[(h-2)\ j\ \eta\ \lambda\ \mu\ 0]^T$	β
$\vdots$		β
$h-1$	$[1\ j\ \eta\ \lambda\ \mu\ 0]^T$	β
h	$[1\ (j-1)\ \eta\ \lambda\ 0\ \frac{(1-\mu)\beta}{\beta-\alpha}]^T$	$\mu\beta$
$h+1$	$[1\ (j-2)\ \eta\ \lambda\ 0\ \frac{\beta}{\beta-\alpha}]^T$	0
$\vdots$		0
$h+j-2$	$[1\ 1\ \eta\ \lambda\ 0\ \frac{\beta}{\beta-\alpha}]^T$	

The transient is also shown by the dash-dotted line of Fig. 1.11, for $\eta = 1$. Practically, with a single step the state is projected on the lower surface of the BL. At step $h-1$, $\mathbf{z}$ reaches box $\mathscr{W}_{1,j,\eta}$, then, with a single step, it moves on the lateral surface $\mathscr{S}'_{j-1,\eta}$ of $\mathscr{W}_{1,j-1,\eta}$. It is possible to verify that points lying on $\boldsymbol{\varsigma}'_{j,\eta}$ have equation $[1\ j\ \eta\ \lambda\ 0\ \frac{\beta}{\beta-\alpha}]^T$, so that $\mathbf{z}_h$ is clearly located below $\boldsymbol{\varsigma}'_{j-1,\eta}$ when $\eta = 1$ (or above $\boldsymbol{\varsigma}'_{j-1,\eta}$ if $\eta = -1$), while the subsequent state, i.e., $\mathbf{z}_{h+1}$, exactly lies on $\boldsymbol{\varsigma}'_{j-2,\eta}$. Box $\mathscr{W}_{1,1,\eta}$ is reached, after $h+j-2$ steps with $u = 0$ and with an acceleration that is equal to z^*, i.e., to z_3^+ or to z_3^- depending on η: The whole transient is feasible with respect to the acceleration constraint.

The subsequent step is the most critical. It was early anticipated that the final transient toward the origin is achieved by means of σ_3: The state, with a single transient, enters in the area that is handled by such SS. As shown in Fig. 1.12, two alternative situations can arise depending on $\hat{\rho}$, both characterized by $u = 0$:

a) $\hat{\rho} \geq [(1-\lambda)(\hat{\tau}-1)]/[\hat{\tau}+2\lambda-1]$. The state evolves to $\bar{\mathbf{z}}_{h+j-1} = [1\ \hat{\tau}\ \eta\ a_1\ a_2\ a_3]^T$

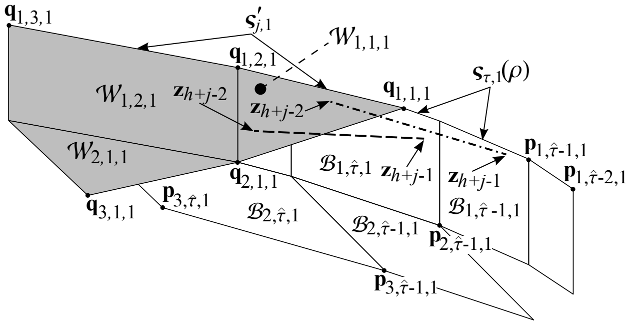
$\mathbf{q}_{1,3,1}$
$\mathbf{s}'_{j,1}$
$\mathcal{W}_{1,1,1}$
$\mathbf{q}_{1,2,1}$
$\mathbf{s}_{\tau,1}(\rho)$
$\mathcal{W}_{1,2,1}$
$\mathbf{z}_{h+j\text{-}2}$
$\mathbf{z}_{h+j\text{-}2}$
$\mathbf{q}_{1,1,1}$
$\mathbf{p}_{1,\hat{\tau}\text{-}1,1}$
$\mathcal{W}_{2,1,1}$
$\mathcal{B}_{1,\hat{\tau},1}$
$\mathbf{z}_{h+j\text{-}1}$
$\mathbf{z}_{h+j\text{-}1}$
$\mathbf{p}_{1,\hat{\tau}\text{-}2,1}$
$\mathbf{q}_{2,1,1}$
$\mathcal{B}_{1,\hat{\tau}\text{-}1,1}$
$\mathbf{q}_{3,1,1}$
$\mathbf{p}_{3,\hat{\tau},1}$
$\mathcal{B}_{2,\hat{\tau},1}$
$\mathcal{B}_{2,\hat{\tau}\text{-}1,1}$
$\mathbf{p}_{2,\hat{\tau}\text{-}1,1}$
$\mathbf{p}_{3,\hat{\tau}\text{-}1,1}$

Figure 1.12: 2D view in the (z_1, z_2)-space of the switching phase between σ_4 and σ_3 ($\eta = 1$). Starting from $\mathscr{W}_{1,1,1}$ the state always reaches, with a single step, $\mathscr{B}_{1,\hat{\tau},1}$ (dashed line) or $\mathscr{B}_{1,\hat{\tau}-1,1}$ (dash-dotted line).

where

$$
\begin{aligned}
a_1 &= \frac{(\lambda-1)(\hat{\tau}-1)+\hat{\rho}(\hat{\tau}+2\lambda-1)}{\hat{\tau}+1}\,,\\
a_2 &= \frac{(1-\hat{\rho})(\lambda-1)\alpha}{\beta-\alpha}\,,\\
a_3 &= \frac{\beta(\hat{\tau}+1)(1-\mu)+\alpha[\hat{\tau}(\lambda-1)(2-\hat{\rho})+\hat{\rho}(\lambda-1)]}{(\hat{\tau}+1)(\beta-\alpha)}\,.
\end{aligned}
$$

It is easy to verify that coefficients $a_1, a_2, a_3 \in [0, 1)$, so that $\bar{\mathbf{z}}_{h+j-1} \in \mathscr{B}_{1,\hat{\tau},\eta}$. Formal proofs are reported in §1.4.1;

b) $\hat{\rho} < [(1-\lambda)(\hat{\tau}-1)]/[\hat{\tau}+2\lambda-1]$. The state evolves to $\bar{\mathbf{z}}_{h+j-1} = [1\ (\hat{\tau}-1)\ \eta\ a_1\ a_2\ a_3]^T$ where

$$
\begin{aligned}
a_1 &= \frac{\hat{\rho}(\hat{\tau}-1)+\lambda(\hat{\tau}+2\hat{\rho}-1)}{\hat{\tau}-1}\,,\\
a_2 &= -\frac{\alpha\lambda\hat{\rho}(\hat{\tau}+1)}{(\hat{\tau}-1)(\beta-\alpha)}\,,\\
a_3 &= \frac{\beta(1-\mu)-\alpha(1-\lambda+\lambda\hat{\rho})}{\beta-\alpha}\,.
\end{aligned}
$$

Coefficients $a_1, a_2, a_3 \in [0, 1)$ (detailed proofs are reported, again, in §1.4.1), so that $\bar{\mathbf{z}}_{h+j-1} \in \mathscr{B}_{1,\hat{\tau}-1,\eta}$.

Once inside $\mathscr{B}_{1,\hat{\tau},\eta}$, the origin is reached with σ_3 according to the following steps (one less step is required from $\mathscr{B}_{1,\hat{\tau}-1,\eta}$):

i	$\bar{\mathbf{z}}_i$	u_{i+1}
$h+j-1$	$[1\ \hat{\tau}\ \eta\ a_1\ a_2\ a_3]^T$	$a_3\alpha+(1-a_3)\beta$
$h+j$	$[1\ (\hat{\tau}-1)\ \eta\ a_1\ 0\ (1-a_2)]^T$	$a_2\beta+(1-a_2)\alpha$
$h+j+1$	$[1\ (\hat{\tau}-2)\ \eta\ a_1\ 0\ 1]^T$	α
$\vdots$		α
$h+j+\hat{\tau}-2$	$[1\ 1\ \eta\ a_1\ 0\ 1]^T$	α
$h+j+\hat{\tau}-1$	$[1\ 1\ \eta\ 0\ \frac{\alpha(a_1-1)}{\beta-\alpha}\ 1]^T$	α
$h+j+\hat{\tau}$	$[1\ 1\ \eta\ 0\ \frac{-\alpha}{\beta-\alpha}\ \frac{\beta-a_1\alpha}{\beta-\alpha}]^T$	$a_1\alpha$
$h+j+\hat{\tau}+1$	$[1\ 1\ \eta\ 0\ \frac{-\alpha}{\beta-\alpha}\ \frac{\beta}{\beta-\alpha}]^T$	

Thus, starting from any state $\mathbf{z}_0 \in \mathscr{W}_{j,h,\eta}$, the origin is reached, in the worst case, in $h+j+\hat{\tau}+1$ steps, i.e., the same number of steps of the transients that are required from $\mathbf{q}_{j,h,\eta}$. The transient optimality is also proved by the bang-zero-bang behavior of the command signal, that admits $u = 0$ only when the acceleration is equal to the maximum admissible value.

1.4.1 Definition range proofs of a_1, a_2, a_3

In the following, it is proved that coefficients $a_1, a_2, a_3 \in [0,1)$ $\forall\hat{\rho} \in [0,1)$, $\lambda \in [0,1)$, $\mu \in [0,1)$, $[\alpha,\beta]$ defined in (1.17) and $\forall\hat{\tau} \geq 2$ if $\hat{\rho} < [(1-\lambda)(\hat{\tau}-1)]/[\hat{\tau}+2\lambda-1]$. It is worth to highlight that the definition range of $\hat{\tau}$ is $\hat{\tau} \geq 2$ when the system state evolves to $\bar{\mathbf{z}}_{h+j-1} = [1\ (\hat{\tau}-1)\ \eta\ a_1\ a_2\ a_3]^T$. It is easy to verify that, bearing in mind (1.19) and (1.20) and from $\hat{\tau} \in \mathbb{N}^+$, the system could evolve in $\mathscr{B}_{1,\hat{\tau}-1,\eta}$ only if the starting box $\mathscr{W}_{1,1,\eta}$ intercepts surface σ_3 inside a box $\mathscr{B}_{1,\hat{\tau},\eta}$ characterized by $\hat{\tau} \geq 2$: If the switching transient from σ_4 to σ_3 drives the system to box $\mathscr{B}_{1,1,\eta}$, only case a) can occur. This assertion is proved by substituting $\hat{\tau} = 1$ in the boundary relation

$$\hat{\rho} < \frac{(1-\lambda)(\hat{\tau}-1)}{\hat{\tau}+2\lambda-1}\,.$$

The analysis of the resulting condition highlights that $\hat{\rho} < 0$ and, bearing also in mind the definition (1.21), case b) becomes impossible. Hence, if the system evolves in

$\mathscr{B}_{1,\hat{\tau},\eta}$ with $\hat{\tau} \geq 2$ one of the two cases could arise. Conversely, if the destination box is characterized by $\hat{\tau} = 1$, only case a) can occur. After these premises, it is possible to demonstrate which is the definition interval for coefficients a_i where $i = 1, 2, 3$.

Property 1. *If* $\hat{\rho} \geq [(1-\lambda)(\hat{\tau}-1)]/[\hat{\tau}+2\lambda-1]$*, coefficient*

$$a_1 = \frac{(\lambda-1)(\hat{\tau}-1)+\hat{\rho}(\hat{\tau}+2\lambda-1)}{\hat{\tau}+1} \tag{1.36}$$

belongs to the following range: $a_1 \in [0,1)$ $\forall\lambda$, $\hat{\rho} \in [0,1)$*,* $\hat{\tau} \geq 1$.

Proof. Firstly, we prove that $a_1 \geq 0$. It is easy to verify that a_1 is proportional to $\hat{\rho}$ and, by hypothesis, $\hat{\rho} \in \left[\frac{(1-\lambda)(\hat{\tau}-1)}{\hat{\tau}+2\lambda-1}, 1\right)$. Consider the worst-case analysis on the boundary condition $\hat{\rho} = \frac{(1-\lambda)(\hat{\tau}-1)}{\hat{\tau}+2\lambda-1}$. By substituting such expression in (1.36) and after a few algebraic manipulations, it is possible to verify that $a_1 \geq 0\, \forall\lambda$, $\hat{\rho} \in [0,1)$, $\hat{\tau} \geq 1$.

The last part of the proof demonstrates that $a_1 < 1$. From the analysis of (1.36) it is evident that a_1 is a positive monotonic function of λ. Hence, it is possible to dominate the relation (1.36) by assuming $\lambda = 1$. Equation (1.36) becomes:

$$a_1 \leq \frac{\hat{\rho}(\hat{\tau}+1)}{\hat{\tau}+1} = \hat{\rho} < 1\,,$$

$\forall\lambda, \hat{\rho} \in [0,1), \hat{\tau} \geq 1$. □

Property 2. *If* $\hat{\rho} \geq [(1-\lambda)(\hat{\tau}-1)]/[\hat{\tau}+2\lambda-1]$*, coefficient*

$$a_2 = \frac{(1-\hat{\rho})(\lambda-1)\alpha}{\beta-\alpha} \tag{1.37}$$

belongs to the following range: $a_2 \in [0,1)$ $\forall\lambda$, $\hat{\rho} \in [0,1)$*,* $\forall\hat{\tau} \geq 1$.

Proof. It is possible to rearrange (1.37) as follows:

$$a_2 = \frac{-\alpha}{\beta-\alpha}(1-\hat{\rho})(1-\lambda)\,. \tag{1.38}$$

Bearing in mind (1.17), evidently $0 < \frac{-\alpha}{\beta-\alpha} < 1$. Equations (1.32) and (1.21) make it possible to assert that $\lambda < 1$ and $\hat{\rho} < 1$ and, in turn, that $a_2 > 0\, \forall\lambda$, $\hat{\rho} \in [0,1)$, $\forall\hat{\tau} \geq 1$. For the same reasons $a_2 < 1\, \forall\lambda$, $\hat{\rho} \in [0,1)$, $\forall\hat{\tau} \geq 1$. □

Property 3. *If* $\hat{\rho} \geq [(1-\lambda)(\hat{\tau}-1)]/[\hat{\tau}+2\lambda-1]$*, coefficient*

$$a_3 = \frac{\beta(\hat{\tau}+1)(1-\mu)+\alpha[\hat{\tau}(\lambda-1)(2-\hat{\rho})+\hat{\rho}(\lambda-1)]}{(\hat{\tau}+1)(\beta-\alpha)} \tag{1.39}$$

belongs to the following range: $a_3 \in [0,1)\ \forall\mu,\ \lambda,\ \hat{\rho} \in [0,1)$*,* $\forall\hat{\tau} \geq 1$.

Proof. By tacking into account the definition ranges of all the variables in (1.39), i.e., $[\alpha,\beta]$ as defined in (1.17), $\hat{\tau} \geq 1$, and $\mu,\ \lambda,\ \hat{\rho} \in [0,1)$, the proof of $a_3 \geq 0\, \forall\mu,\ \lambda,\ \hat{\rho} \in [0,1)$, $\forall\hat{\tau} \geq 1$ is straightforward.

The proof of condition $a_3 < 0$ is provided by a worst-case analysis: Relation (1.39) is a positive monotonic function of λ and can be dominated by considering $\lambda = 1$:

$$a_3 \leq \frac{\beta(\hat{\tau}+1)(1-\mu)}{(\hat{\tau}+1)(\beta-\alpha)} = (1-\mu)\frac{\beta}{(\beta-\alpha)} < 1$$

$\forall\mu \in [0,1)$, $\forall[\alpha,\beta]$ (see (1.17)), so that $0 \leq a_3 < 1$. □

Property 4. *If* $\hat{\rho} < [(1-\lambda)(\hat{\tau}-1)]/[\hat{\tau}+2\lambda-1]$ *and* $\hat{\tau} \geq 2$*, coefficient*

$$a_1 = \frac{\hat{\rho}(\hat{\tau}-1)+\lambda(\hat{\tau}+2\hat{\rho}-1)}{\hat{\tau}-1} \tag{1.40}$$

belongs to the following range: $a_1 \in [0,1)\ \forall\lambda,\ \hat{\rho} \in [0,1)$*,* $\forall\hat{\tau} \geq 2$.

Proof. As already done in Property 1, first consider a worst-case analysis of (1.40). a_1 is a positive monotonic function of $\hat{\rho}$: By substituting boundary condition $\hat{\rho} = \frac{(1-\lambda)(\hat{\tau}-1)}{\hat{\tau}+2\lambda-1}$ in (1.40), it is evident, after a few algebraic manipulation, that $a_1 = 1$. Since $\hat{\rho} < \frac{(1-\lambda)(\hat{\tau}-1)}{\hat{\tau}+2\lambda-1}$, the previous analysis proves that $a_1 < 1\ \forall\lambda,\ \hat{\rho} \in [0,1)$, $\forall\hat{\tau} \geq 2$.

The last part of the proof demonstrates that $a_1 \geq 0$. Equation (1.40) is a positive, monotonic function of λ: It can be dominated by considering $\lambda = 0$. After a few algebraic manipulations, (1.40) can be written as follows:

$$a_1 \geq \frac{\hat{\rho}(\hat{\tau}-1)}{\hat{\tau}-1} = \hat{\rho} \geq 0\,.$$

Hence, $a_1 \in [0,1)\ \forall\lambda,\ \hat{\rho} \in [0,1)$, $\forall\hat{\tau} \geq 2$. □

Property 5. *If $\hat{\rho} < [(1-\lambda)(\hat{\tau}-1)]/[\hat{\tau}+2\lambda-1]$ and $\hat{\tau} \geq 2$, coefficient*

$$a_2 = -\frac{\alpha\lambda\hat{\rho}(\hat{\tau}+1)}{(\hat{\tau}-1)(\beta-\alpha)} \tag{1.41}$$

belongs to the following range: $a_2 \in [0,1)$ $\forall\lambda,\ \hat{\rho} \in [0,1)$, $\forall\hat{\tau} \geq 2$.

Proof. It is easy to verity that $a_2 \geq 0$ by tacking into account the definition intervals of all its variables, i.e., $[\alpha,\beta]$ as defined by (1.17), $\hat{\tau} \geq 2$, and $\lambda,\ \hat{\rho} \in [0,1)$.

In order to demonstrate that $a_2 < 1$ let us make the following considerations: a_2 is a positive monotonic function of $\hat{\rho}$. By substituting boundary condition $\hat{\rho} = \frac{(1-\lambda)(\hat{\tau}-1)}{\hat{\tau}+2\lambda-1}$ in (1.41), coefficient a_2 is dominated as follows:

$$a_2 \leq -\frac{\alpha\lambda(1-\lambda)(\hat{\tau}+1)}{(\hat{\tau}+2\lambda-1)(\beta-\alpha)}\ . \tag{1.42}$$

A further step of worst-case analysis is necessary to obtain the final proof of condition $a_2 < 1$. Equation (1.42) can be dominated by considering $\hat{\tau} = 2$. After a few manipulations, (1.42) can be written as follows

$$a_2 \leq -\frac{\alpha}{\beta-\alpha}(1-\lambda) < 1\ ,$$

so that it is possible to assert that $a_2 \in [0,1)$ $\forall\lambda,\ \hat{\rho} \in [0,1)$, $\forall\hat{\tau} \geq 2$. □

Property 6. *If $\hat{\rho} < [(1-\lambda)(\hat{\tau}-1)]/[\hat{\tau}+2\lambda-1]$ and $\hat{\tau} \geq 2$, coefficient*

$$a_3 = \frac{\beta(1-\mu)-\alpha(1-\lambda+\lambda\hat{\rho})}{\beta-\alpha} \tag{1.43}$$

belongs to the following range: $a_3 \in [0,1)$ $\forall\mu,\ \lambda,\ \hat{\rho} \in [0,1)$, $\forall\hat{\tau} \geq 2$.

Proof. It is evident from the definitions of $[\alpha,\beta]$ (see (1.17)) and $\mu,\ \lambda,\ \hat{\rho} \in [0,1)$, that $a_3 \geq 0$. The proof of $a_3 < 1$ is archived by means of a worst-case analysis. Bearing in mind (1.43), the following assumptions are made: $(1-\mu) \leq 1$ $\forall\mu \in [0,1)$ and $(1-\lambda+\lambda\hat{\rho}) \leq 1$ $\forall\lambda,\hat{\rho} \in [0,1)$, so that it is possible to verify that

$$a_3 \leq \frac{\beta-\alpha}{\beta-\alpha} = 1$$

thus $a_3 \in [0,1)$ $\forall\mu,\ \lambda,\ \hat{\rho} \in [0,1)$, $\forall\hat{\tau} \geq 2$. □

Finally, it is possible to assert that, regardless of the box which is reached by a single step evolution starting from box $\mathscr{W}_{1,1,\eta}$, coefficients a_1, a_2, a_3 always belong to interval $[0,1)$.

1.5 Convergence properties of σ_1 and σ_2

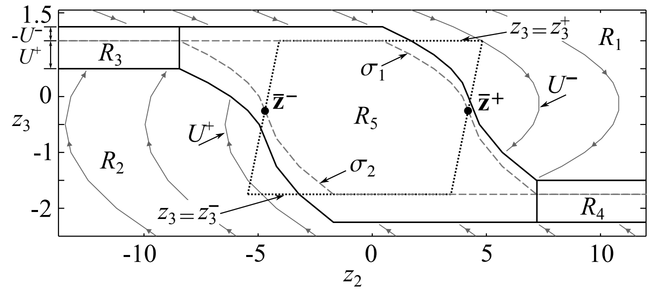

Figure 1.13: Partition induced in the (z_2, z_3)-plane by the AVSC together with the SSs σ_1 and σ_2 and with some system trajectories. The dotted quadrangle represents the feasible area in the (z_2, z_3)-plane obtained from the transformation of the velocity and acceleration limits.

Surfaces σ_1 and σ_2 are used to fulfill the velocity limits. To this aim, they force $\mathbf{z}$ inside the quadrangle (shown in Fig. 1.13) that has been obtained by converting the feasible domain of the $(\dot{x}, \ddot{x})$-space into an equivalent feasible domain in the (z_2, z_3)-space. The feasible domain of the $(\dot{x}, \ddot{x})$-space is represented by a rectangle that is bounded by the following four straight lines: $\ddot{x} = S^+$, $\ddot{x} = S^-$, $\dot{x} = R^+$, and $\dot{x} = R^-$.

Surfaces σ_1 and σ_2 are derived from analogous surfaces that were originally proposed in [29]. This is possible because the evolution of system (1.6) in the (z_2, z_3)-space does not depend on the z_1 component of the state and it is given by

$$\begin{bmatrix} z_{2,i+1} \\ z_{3,i+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_{2,i} \\ z_{3,i} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_i \,, \tag{1.44}$$

i.e., system equations coincide with those that were considered in [29], but the role of the pair z_1 and z_2 is now played by z_2 and z_3. Thus, by adopting the same command law that was proposed in [29], the same convergence properties are obtained. However, minor changes have been introduced in σ_1 and σ_2 in order to modify the convergence points: σ_1 guarantees the convergence toward $\overline{\mathbf{z}}^+$, while σ_2 guarantees the convergence toward $\bar{\mathbf{z}}^-$. Therefore, the convergence properties of σ_1 and σ_2 are revisited and reported in the following.

The evolution of the reduced system (1.44) is shown with the aid of Fig. 1.13. The same figure is also useful to demonstrate the convergence properties of σ_1 and σ_2. The (z_2, z_3)-plane can be subdivided into five regions R_i, $i = 1, 2, \ldots, 5$, where regions R_1 and R_2 are totally defined outside the BL, while the borders of the regions R_3, R_4 and R_5 share at least one of the BL obtained with the surfaces σ_1 or σ_2.

The following properties describe the behavior of the SSs σ_1 and σ_2: They force $\mathbf{z}$ inside the feasible quadrangle that is shown in Fig. 1.13.

Property 7. *For any (z_2, z_3) point lying in R_1 the AVSC returns $u = U^-$. Conversely, for any (z_2, z_3) point lying in R_2 the AVSC returns $u = U^+$.*

Proof. Consider a point (z_2, z_3) that lies in R_1: Three cases could arise depending on the relation (1.10).

1. $\overline{\sigma}(z_1, z_2) > \sigma_1(z_2)$. Due to (1.10) the selected SS is $\sigma = \sigma_1(z_2)$. By assuming these premises in (1.10) and in (1.9), the AVSC returns $u = U^-$ because the point is outside the BL, i.e., from (1.9), $z_3 - \sigma > 1$.

2. $\sigma_1(z_2) \leq \overline{\sigma}(z_1, z_2) \leq \sigma_2(z_2)$. The selected SS is $\overline{\sigma}(z_2, z_3)$. Even in this situation, the (z_2, z_3) point is located outside the BL and the command law, that is chosen by the AVSC, is again $u = U^-$.

3. $\overline{\sigma}(z_1, z_2) < \sigma_2(z_2)$. Even in the last case, the point (z_2, z_3) is located outside the BL. The chosen command law is evidently $u = U^-$ because, from (1.9), $z_3 - \sigma > 1$.

Similar considerations holds in case the point (z_2, z_3) is located in region R_2. Conversely from any point that lies in R_1, the AVSC chooses, bearing also in mind (1.9), the command law $u = U^+$. □

Remark 1. *Since the maximum control law is assumed in both regions R_1 and R_2, i.e., $u = U^-$ in region R_1 and $u = U^+$ in region R_2, one of the regions R_3, R_4, and*

R_5 is certainly reached in minimum-time compatibly with the given bounds (see the system evolution trajectories in Fig. 1.13).

Property 8. *Any point (z_2, z_3), that lies in R_3, is forced in a single step on σ_1 and, then, it slides toward R_5 with command signal $u = 0$. Conversely, any point in R_4 is forced in a single step on σ_2 and, then, it slides toward R_5 with $u = 0$.*

Proof. Consider a point (z_2, z_3) that lies inside region R_3. Independently from the position of $\overline{\sigma}(z_1, z_2)$, it is easy to verify that (1.10) returns $\sigma = \sigma_1 = \sigma_2 = z_3^+ \geq 0$. According to (1.44), z_3 evolves as follows

$$z_{3,i+1} = z_{3,i} + u_i \, . \tag{1.45}$$

The command signal, due to (1.9) and (1.10), becomes $u_i = -U^-\left(\frac{z_{3,i}-\sigma_1}{U^-}\right) = -z_{3,i} + z_3^+$, so that (1.45) returns $z_{3,i+1} = z_3^+$ independently from the starting point $z_{3,i}$, i.e. the system (1.44) reaches σ_1 in a single step. If the starting point lies on σ_1 in R_3, i.e. $(z_2, z_3) = (z_2, z_3^+)$, the AVSC command law assumes $u_i = 0$. Hence, z_3 remains on the σ_1 SS with a constant and positive value.

The evolution of z_2 can be deduced again by (1.44) and it follows the law:

$$z_{2,i+1} = z_{2,i} + z_{3,i} + u_i \, . \tag{1.46}$$

Even in this case, from any point inside R_3, the AVSC command law becomes $u_i = -z_{3,i} + z_3^+$ and, from (1.46), z_2 evolves in $z_{2,i+1} = z_{2,i} + z_3^+$, i.e. the point shifts right with step equal to z_3^+ that corresponds to the maximum acceleration value in the **z**-space.

Therefore, independently from the starting point inside R_3, any (z_2, z_3) point reaches the SS σ_1 in a single step and, in turn, slides right in minimum-time, i.e. with increments equal to z_3^+ (the AVSC returns $u = 0$), since it enters with certainty in R_5.

It is possible to archive the same conclusion, with similar reasonings, if the starting point is located inside R_4. In this scenario, the role of σ_1 is assumed by σ_2 and the system state slides left in minimum-time with increments equal to $z_3^- \leq 0$. □

Remark 2. *Property 8 asserts that if the system state enters in R_3 or R_4, it is entrapped into those regions and it enters with certainty in R_5 with minimum-time transients, i.e. with the maximum acceleration allowable with respect to the assigned bounds. Besides, the acceleration limits are not violated because $z_3 = z_3^+$ in R_3 and $z_3 = z_3^-$ in R_4.*

Properties 7 and 8 ensure that from any point in the (z_2, z_3)-plane the system evolves, with minimum-time transients, to region R_5. Next properties show that the system state cannot abandon region R_5 and, furthermore, it converges toward the origin with the minimum number of steps.

Property 9. *Consider a starting point (z_2, z_3) belonging to R_5. The system remains inside R_5 and converges toward the origin or one of the two points $\bar{\mathbf{z}}^+$, $\bar{\mathbf{z}}^-$ (defined in (1.14) and (1.15) and shown in Fig. 1.13).*

Proof. A further change of coordinates must be first applied to reduced system (1.44) in order to move the σ_1 (or σ_2) convergence point from $\bar{\mathbf{z}}^+$ (or $\bar{\mathbf{z}}^-$) to the new point $\bar{\mathbf{t}}^+$ (or $\bar{\mathbf{t}}^-$) defined in the following (see (1.49) and (1.50)). The main purpose of the coordinate transformation is to convert reduced system (1.44) into the analogous one presented in [21] and, in turn, in [15]. The change of coordinates transforms any point that lies on the (z_2, z_3)-plane in a new point defined in the $\mathbf{t}$-space:

$$\begin{bmatrix} t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_{2,i} \\ z_{3,i} \end{bmatrix} + \begin{bmatrix} \frac{\ddot{r}}{2T} \\ \frac{\ddot{r}}{T} \end{bmatrix}, \tag{1.47}$$

i.e the (z_2, z_3)-state is shifted, in the new space, by $[\frac{\ddot{r}}{2T} \;\; \frac{\ddot{r}}{T}]^T$. Due the affine transformation (1.47), the reduced discrete system maintains the same dynamics of (1.44). Bearing in mind (1.13)–(1.15), the equivalent bounds in the $\mathbf{z}$-space are transformed, owing to (1.47), as follows:

$$t_3^- := \tfrac{S^-}{T}, \quad t_3^+ := \tfrac{S^+}{T}, \tag{1.48}$$

$$\bar{\mathbf{t}}^+ := [\bar{t}_2^+ \; \bar{t}_3^+]^T := \left[\tfrac{R^+ - \dot{r}}{T^2} \;\; 0\right]^T, \tag{1.49}$$

$$\bar{\mathbf{t}}^- := [\bar{t}_2^- \; \bar{t}_3^-]^T := \left[\tfrac{R^- - \dot{r}}{T^2} \;\; 0\right]^T. \tag{1.50}$$

The use of (1.48)–(1.50) in place of (1.13)–(1.15), makes it possible to define the transformed SSs $\widetilde{\sigma}_1$ and $\widetilde{\sigma}_2$ as follows ($n = 1,2$).

$$\widetilde{\sigma}_n := -\frac{\widetilde{\gamma}_n}{\widetilde{m}_n} - \frac{\widetilde{m}_n - 1}{2}\widetilde{\kappa}_n \, , \tag{1.51}$$

where, $\widetilde{\gamma}_n$, $\widetilde{\kappa}_n$, and $\widetilde{m}_n$ are evaluated as:

$$\begin{aligned}
\hat{t}_2^+ &:= -\left\lceil -\tfrac{t_3^+}{U^-} \right\rceil \left[t_3^+ + \tfrac{U^-}{2}\left(\left\lceil -\tfrac{t_3^+}{U^-} \right\rceil - 1 \right) \right] \, , \\
\hat{t}_2^- &:= -\left\lceil -\tfrac{t_3^-}{U^+} \right\rceil \left[t_3^- + \tfrac{U^+}{2}\left(\left\lceil -\tfrac{t_3^-}{U^+} \right\rceil - 1 \right) \right] \, , \\
\widetilde{d}_1 &:= t_2 - \overline{t}_2^+ \, , \quad \widetilde{d}_2 := t_2 - \overline{t}_2^- \, ,
\end{aligned}$$

$$\widetilde{\gamma}_n := \begin{cases} \hat{t}_2^+ & \text{if } \widetilde{d}_n < \hat{t}_2^+ \\ \widetilde{d}_n & \text{if } \hat{t}_2^+ \leq \widetilde{d}_n \leq \hat{t}_2^- \\ \hat{t}_2^- & \text{if } \widetilde{d}_n > \hat{t}_2^- \end{cases} \quad \widetilde{\kappa}_n := \begin{cases} U^- & \text{if } \widetilde{\gamma}_n \leq 0 \\ U^+ & \text{if } \widetilde{\gamma}_n > 0 \end{cases} \quad \widetilde{m}_n := \left\lfloor \tfrac{1}{2} + \sqrt{\tfrac{1}{4} + 2\left| \tfrac{\widetilde{\gamma}_n}{\widetilde{\kappa}_n} \right|} \right\rfloor \, .$$

By analyzing (1.48)–(1.51), it is easy to verify that the transformed reduced system has the same structure of the reduced system described in [21]. The sole difference is represented by the jerk bound asymmetry, but a similar asymmetric case is discussed in [29]. Hence, by assuming the same conclusions proposed in the aforementioned papers, any (z_2, z_3) point, belonging to R_5, is attracted towards point $\overline{\mathbf{t}}^+$ if the AVSC chooses $\widetilde{\sigma}_1$ (or $\overline{\mathbf{t}}^-$ if it chooses $\widetilde{\sigma}_2$) and, in turn, towards point $\overline{\mathbf{z}}^+$ (or $\overline{\mathbf{z}}^-$) in the $\mathbf{z}$-space. Alternatively, the system is driven in minimum-time toward the origin by $\overline{\sigma}$.

It is worth to remember that surface $\overline{\sigma}$ is designed to force the system toward the origin in minimum-time. However, $\overline{\sigma}$ would drive the system outside the feasible area because it does not manage the velocity constraints (see §1.3 and §1.4 for further details): In this eventuality, the AVSC selects σ_1 (or σ_2) and the system is "parked" in $\overline{\mathbf{z}}^+$ (or $\overline{\mathbf{z}}^-$) waiting until $\overline{\sigma}$ newly enters in R_5. □

The reduced system behavior in the $\mathbf{z}$ coordinates can be better explained with the aid of Fig. 1.14 which shows several system trajectories in case of convergence toward σ_1 (Fig. 1.14a) or σ_2 (Fig. 1.14b). All trajectories first hang one of the two SSs with $u = U^+$ or $u = U^-$, i.e., with the maximum admissible command signal. After that, the system slides on the right with $u = 0$ if the AVSC chooses σ_1 and it

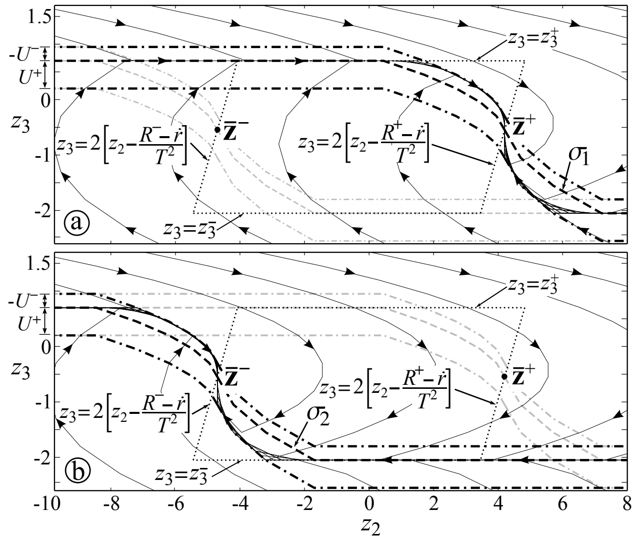

Figure 1.14: System trajectories in the (z_2, z_3)-space obtained by assuming: a) $\sigma = \sigma_1$ and b) $\sigma = \sigma_2$. SSs σ_1 and σ_2 are indicated by means of dashed lines and are surrounded by their BLs (dash dotted lines). The dotted quadrangle contours the feasible area. z_3^+ and z_3^- are defined according to (1.13).

converges toward to $\overline{\mathbf{z}}^+$. Conversely, if the system first hangs σ_2, then it slides left along σ_2 with $u = 0$ and then it reaches $\overline{\mathbf{z}}^-$ in minimum-time. Those transients are, evidently, minimum-time due to the bang-zero-bang behavior of the command law.

Property 10. *Points $\overline{\mathbf{z}}^+$ and $\overline{\mathbf{z}}^-$, defined in (1.14) and (1.15), i.e. the two points corresponding to states $(R^-, 0)$ and $(R^+, 0)$ in the $(\dot{x}, \ddot{x})$-space, are left with certainty in finite time.*

Proof. Consider the evolution of the system (1.6) and (1.7) when the starting point is located in $\overline{\mathbf{z}}^+$ and preliminarily assume $\overline{\sigma} > \sigma_1$: The system evolves as follows

$$z_{3,i+1} = z_{3,i} = \overline{z}_3^+ \ , \tag{1.52}$$

$$z_{2,i+1} = z_{2,i} + z_{3,i} = z_{2,i} + \overline{z}_3^+ \ , \tag{1.53}$$

$$z_{1,i+1} = z_{1,i} + z_{2,i} + z_{3,i} = z_{1,i} + z_{2,i} + \overline{z}_3^+ \ , \tag{1.54}$$

Without loss of generality, consider $\overline{z}_3^+ \geq 0$, so that the AVSC chooses σ_1 in (1.10)

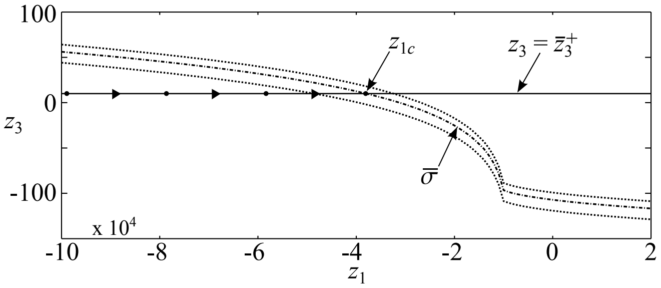

Figure 1.15: $\overline{\sigma}(z_1, z_2)$ evolution and its BL projected in the (z_1, z_3)-plane drawn for $z_2 = \overline{z}_2^+ = 400$. $\overline{\sigma}$ monotonically decreases in function of z_1. The system transients along z_1 is depicted by means dots and arrows.

with command law $u = 0$. In the (z_2, z_3)-plane the reduced system (1.44) evolves along a plane characterized by $z_3 = \overline{z}_3^+$. According to (1.54), z_1 evolves in the (z_1, z_2)-plane with steps equal to $z_{2,i} + \overline{z}_3^+$. The system behavior can be easily understood with the aid of Fig. 1.15 which shows the shape of $\overline{\sigma}(z_1, z_2)$ in the (z_1, z_3)-plane together with its BL: $\overline{\sigma}$ monotonically decreases in function of z_1. This characteristic applies over the whole (z_1, z_2)-plane. In $\overline{\mathbf{z}}^+$, $\sigma_1 = \overline{z}_3^+ > 0$ and, in turn $\overline{\sigma} > \overline{z}_3^+$. The positivity of $\overline{\sigma}$ implies that the current value of z_1 is located on the left of z_{1c}, where z_{1c} is the point where surface $\overline{\sigma}$ crosses line $z_3 = \overline{z}_3^+$, i.e. the solution of $\overline{\sigma}(z_1, z_2) = \overline{z}_3^+$. Since z_1 increases, surface $\overline{\sigma}$ is reached with certainty. As soon as $\overline{\sigma} < z_3^+$, the AVSC switches (owing to (1.10)) and starts using $\overline{\sigma}$. Hence, convergence point $\overline{\mathbf{z}}^+$ is abandoned in minimum-time (z_1 evolves with increments equals to $\overline{z}_2^+$, i.e. the maximum allowable velocity compatible with the assigned constraints) and the system starts a transient toward the origin following $\overline{\sigma}$ with control law $u = U^-$ (see §1.3 and §1.4 for the convergence proofs of σ_3 and σ_4).

The same reasoning can be made by assuming $\overline{z}_3^+ < 0$.

Analogous considerations hold when $\overline{\sigma} < \sigma_2$ and the system is initially locked in $\overline{\mathbf{z}}^-$. In that case, as soon as $\overline{\sigma} \geq \sigma_2$ the AVSC selects $\overline{\sigma}$ and, with command law $u = U^+$, the system converges toward the origin in minimum-time. □

1.6 Comparisons with respect to the previous version

The performances of the new filter are compared with those of an analogous filter devised in [21]. As claimed in §1.1, the filter here presented eliminates a suboptimal behavior that was affecting the version proposed in [21]: Depending on the reference signal and on the bounds, transients of [21] could be non-minimum-time and, in turn, undesired overshoots could appear.

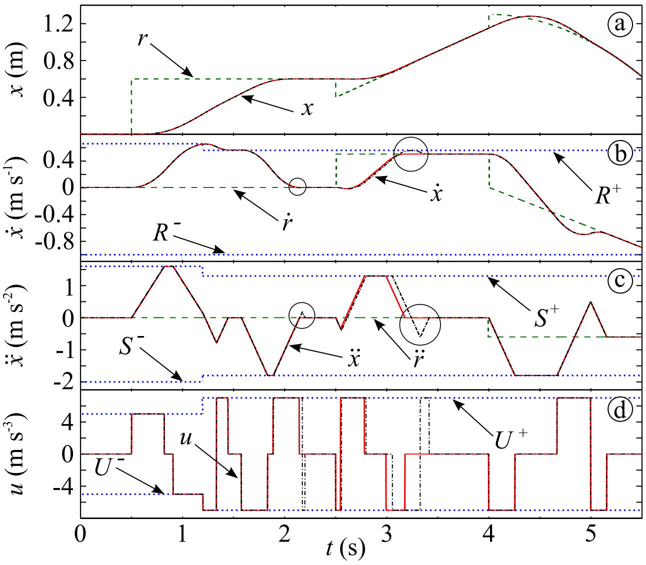

Figure 1.16: Comparisons between the outputs of the novel filter (solid lines) and those proposed in [21] (dash-dotted lines): Reference signal (dashed lines) is reached in minimum-time and by fulfilling the imposed bounds (dotted lines). Circles highlight the configurations for which the new filter returns shorter transients.

The filters performances are highlighted by executing the same sequence of a step, a ramp followed by a parabola as shown in Fig. 1.16. The reference signal is non-smooth and unfeasible with respect to the bounds and, as asserted in §1.1, the filter output is the best possible approximation of the input signal. The assigned bounds on the jerk signal have been assumed symmetric for comparison reasons. The bounds are changed online and are assigned as in Table 1.1.

In particular, user-defined bounds are changed at $t = 1.2$ s during a transient and

Table 1.1: Kinematic bounds used for the comparisons with the filter proposed in [21].

t	R^+	R^-	S^+	S^-	U^+	U^-
[s]	[m s^{-1}]	[m s^{-1}]	[m s^{-2}]	[m s^{-2}]	[m s^{-3}]	[m s^{-3}]
0.0	0.65	-1	1.6	-2	5	-5
1.2	0.56	-1	1.3	-1.8	7	-7

both filters reach the reference signal without overshoot. The circles of Fig. 1.16 show, conversely, two cases in which the new filter returns better solutions. Improvements are especially evident for the second of the two transients, which has been shortened by 0.3 s and the hanging overshoot has been totally eliminated. It is interesting to notice that the overshoot of the transient toward the parabolic reference is common to both approaches. This implies that the solution found by [21] was already minimum-time compatibly with the constraints: This overshoot can only be eliminated by loosening the given bounds.

1.7 A test case

The discrete-time filter has been tested in a simulation environment and its performances have been evaluated in case of asymmetric constraints, by using an input signal that has been appositely devised in order to highlight the system capabilities. The test has been executed in a real-time environment, based on RTAI [30].

As shown in Fig. 1.17, the reference signal is discontinuous and it is given by a combination of steps, ramps and parabolas. Assigned constraints are online changed and have been assigned according to Table 1.2. Fig. 1.17 shows the obtained results. Fig. 1.17a depicts the original discontinuous signal (dashed line) compared with the output of the proposed filter. Figs. 1.17b-d show the velocities, accelerations and jerks signals: In dotted lines are shown the dynamic bounds, dashed lines highlight the reference signal.

The filter, as expected, mimics the input signal by means of an output that fulfills the given velocity, acceleration, and jerk constraints. The reference signal is always

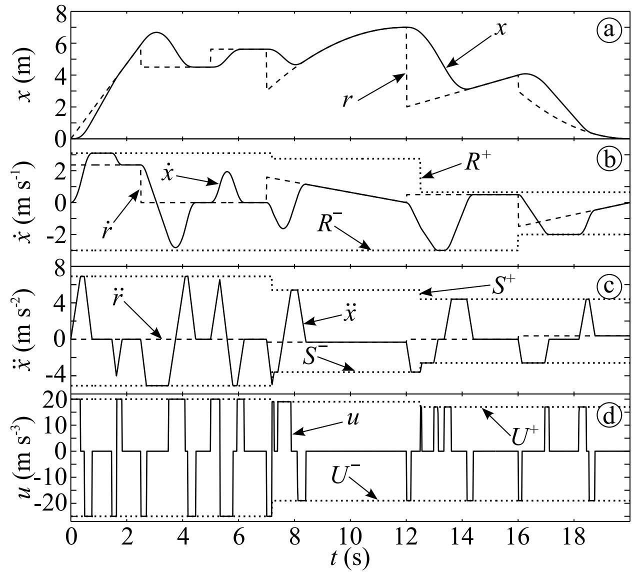

Figure 1.17: Results of the test case. The reference signal (dashed line) is tracked at the best by the filter output (solid lines). Given constraints are always satisfied (dotted lines).

Table 1.2: Kinematic bounds used for the test case.

t	R^+	R^-	S^+	S^-	U^+	U^-
[s]	[m s^{-1}]	[m s^{-1}]	[m s^{-2}]	[m s^{-2}]	[m s^{-3}]	[m s^{-3}]
0.0	3.1	-3	6.9	-5.1	20	-25
6.4	2.75	-3	5.4	-3.6	19	-19
12.5	0.65	-3	4.4	-2.6	17	-19
16	0.65	-2	4.4	-2.6	17	-19

reached without overshoot even when, at $t = 7.2$ s and $t = 12.5$ s, the bounds are changed in the middle of a transient. Transients are always minimum time compatibly with the constraints. The jerk has a bang-zero-bang behavior and the chattering phenomenon is totally avoided. The filter has been executed with a sampling time $T = 1 \cdot 10^{-3}$ s. An average evaluation time of $5.24 \cdot 10^{-6}$ s has been measured on a PC equipped with an Intel Core2 Duo processor @ 3.00 GHz: Even considering less efficient processors, the computational burden would still be compatible with the sampling times of many industrial applications.

2

Trajectory scaling systems in the configuration space

Any industrial plant is affected by physical limits that could worsen motion control. These limits can be managed at control level or by synthesizing proper reference signals. In particular, ranges of controlled variables are typically limited, admissible motor velocities and accelerations are bounded, actuator torques are finite, etc.. The controller performances worsen every time saturation occurs and, in the worst situations, feedback systems can also be destabilized. For this reason, industrial controllers are necessarily designed tacking this phenomenon into account. Approaches that are typically used to prevent the problem are based on anti-windup schemes or on model predictive controllers. Both techniques require specially devised control schemes. Alternatively, it is possible to manage system limitations by properly planning reference signals. The main advantage of this choice is that it permits the use of much simpler control structures.

Early trajectory-based approaches were conceived to be executed offline [31–33] through the resolution of an optimization process. Unfortunately, offline techniques cannot account for model uncertainties, trajectory changes or unexpected disturbance in the system, so that given bounds cannot be fulfilled with certainties. In order to solve these issues, online techniques have been developed. Online approaches are typically based on the path-velocity decomposition paradigm [34]: The trajectory is composed of a path and a Longitudinal Time-Law (LTL). By correctly acting on the

LTL, trajectory feasibility is preserved and path tracking performances are significantly improved. Algorithms based on this concept are also known as Trajectory Scaling Systems (TSS) because trajectory feasibility is maintained by properly slowing (or scaling) down the LTL in order to fulfill the kinematic and the dynamic limits.

The scaling problem has been traditionally solved by considering two planning frameworks: Trajectories could be planned in the configuration space or in the operational space. This chapter focusses on defining and solving a scaling problem in the configuration space. The trajectory is scaled down in order to fulfill a given set of kinematic bounds on joint velocities, accelerations and jerks, and of dynamic bounds on the generalized force and force derivatives.

Alternatively, the scaling problem can also be handled by considering paths planned in the operational space: The introduction of bounds defined in the Cartesian space requires a more challenging problem to be solved with respect to the one posed above because of the non-linear relationships between the operational and configuration spaces. The scaling problem in the operational space will be formulated and solved in next Chapter 3.

A closer examination of robotic literature about the path planning in the configuration space reveals that the early works were developed in the eighties in an offline context. The interpolation between joint configurations can be accomplished by posing a polynomial interpolation problem. In [1] a total traveling time minimization problem has been solved offline by simultaneously managing limits on joint velocities, accelerations and jerks. The jerk constraints were added in order to prevent, or, better, to minimize mechanical stresses. This historical approach had a major drawback: It does not consider, in the optimization process, the manipulator dynamics and, as a consequence, control performances of the actual implementation could worsen.

The open problem of the integration of the robot dynamics in the optimization procedure has been considered in [35]. In an offline context, paths were planned by considering limitations on the manipulator dynamics and constraints on the joint velocities and accelerations. More precisely, the manipulator dynamics were accounted for by considering the actuator torques and by converting them into equivalent limits on joint accelerations. The trajectory feasibility was maintained with the aid of an iterative algorithm, which added new spline control points every time the optimization process was unable to find feasible solutions. In the approach proposed in [35] the jerk constraints were not managed. More recent approaches adopt fifth-order splines in order to also manage the jerk limits. In [5], for example, the system dynamics has been handled by considering torque and torque derivative limits. In a similar solution presented in [36], torque and torque-derivative constraints were mapped into equivalent bounds used to find the best control points configuration of a cubic spline, in order to simultaneously satisfy the given kinematic and dynamic limits.

In order to obtain a more efficient solution to the optimization problem, a suboptimal strategy for systems subject to kinematic constraints on velocities, accelerations and jerks and to dynamic limits on actuators' torques has been proposed in [37]. It is based on the use of a dynamic programming approach. The proposed approach approximates the time-optimal solution to any desired degree of accuracy.

The optimization problem, which is at the basis of the planning method, has been subsequently solved in a discrete-time framework. The discretization process could add undesired approximations to the optimization procedure which could lead to feasibility issues. The main reasons which motivated the discretization of the planning problem have been summarized in [38]:

- Commonly manipulators are governed through discrete-time controllers;
- Dynamics equations, which rule the manipulators' motions, are highly coupled and require demanding computational resources for their computations.

A significant step forward for the resolution of path planning problems in real-time frameworks has been presented in [39]. Although the solution found in that paper is based on an offline investigation, it proposes an interesting idea based on the Pontryagin's maximum principle: Minimum-time trajectories can be obtained from the generation of discrete curves that, at each sample time, assume maximum or minimum allowable acceleration with respect to the given limits. The acceleration constraints and the generalized force limits caused by the actuators' electronics have been considered in [39] by converting them into equivalent bounds on the joint accelerations. According to this strategy, it is not necessary to solve a complex optimization problem to constrain the whole trajectory but, at each sample time, it is sufficient to choose a proper acceleration value on the basis of the current kinematic and dynamic configuration. The acceleration signal can admit only two values which correspond to the given acceleration limits. The solution has been investigated offline through an iterative algorithm based on a forward integration with maximum acceleration command followed by a backward integration procedure obtained, from the trajectory end-point, by assuming the maximum deceleration command. Pontryagin's maximum principle postulates that, for the problem at hand, trajectories, which admit a single switch in the acceleration command, are minimum-time. Moreover, paper [39] has introduced the use of the phase plane s-$\dot{s}$. It represents the evolution of the velocity signal with respect to the distance along the path s. The admissible region has been defined as the region where the generated profile is feasible with respect to the given bounds.

The planning technique proposed in [39] has been extended in [40] where, at the planning stage, bounds in the operational space have been considered and the phase plane definition has been extended. The idea introduced by Bobrow *et. al.* has been improved in subsequent years in order to obtain a time-minimum trajectory for 6 Degrees of Freedom (DoF) robots [41]. Furthermore, the gripper forces' constraints have been introduced in [42].

Previously cited methods are unfortunately not compatible with real-time implementations. Even considering suboptimal solutions, the solution of optimization problems requires computational times of several milliseconds. If lower computation times are required, i.e. execution times are comparable with control loops' sample times, profile generations can be managed in an "indirect" way: Trajectories can be generated by means of appropriate feedback schemes that smooth rough unfeasible signals in order to fulfill the given bounds. Early works on this topic have been presented in [10] and [43] where, in a continuous time framework, trajectories for mechatronic systems with bounded joint velocities and accelerations were generated by means of a continuous time filter. The limits are changed in real-time and are symmetric. The technique has been successively discretized in [14] but, in this case, only the symmetric acceleration limits has been managed.

By exploiting the path-velocity decomposition paradigm [34], paper [16] proposed the first version of the Trajectory Scaling System (TSS): The trajectory is generated by composing a Longitudinal Time-Law (LTL) and a path. The TSS is placed between the LTL and the path generators and it scales the LTL in order to fulfill the given limits. These constraints, defined in the joint space, must be converted into equivalent bounds for the LTL. In [16] the limits on the maximum joint velocities and accelerations have been considered. The problem has been revised and extended in [23] in order to manage high-order kinematic and dynamic constraints. In particular, the TSS was modified to be able to generate smooth profiles for systems subject to constraints on: Joint velocities, accelerations and jerks and generalized forces and force derivatives acting on the joints. A preliminary experimental study on the TSS presented here has been accomplished in [44]. It is worth noting which are the main differences between the results proposed in this chapter and those presented in [44]: In the previous version, the planner was able to manage bounds on the joint velocities and accelerations online as well as the generalized forces acting on the joints. The TSS presented in the following is also able to constrain joint jerks and the derivative of the generalized forces acting on the joints.

The scaling problem proposed in [23] is revised in the following sections and tested in an experimental environment in order to prove the robustness of the proposed method with respect to paths that differently solicit the dynamic behavior of the system. Moreover, §2.4 proposes a solution for the feasibility problems that could arise when the motors rotate at high speed. Most of the following results have been presented in [45].

2.1 The scaling problem

The scaling problem can be summarized as follows: A trajectory, planned according to the path-velocity decomposition paradigm [34], must be proper scaled in order to fulfill the dynamic or kinematic constraints and to track at best the desired trajectory. The problem is solved by properly slow down the Longitudinal Time-Law

(LTL) of an appositely design scaling system. The scaling system, for configuration in witch the given limits cannot be violated, must also eliminate accumulated delays in minimum time. Indicating with $\mathbf{q} := [q_1 \; q_2 \; \dots \; q_N]^T$ the vector of the generalized variables used to describe the joint displacement, the considered kinematic/dynamic constraints are related to the maximum joint velocities, accelerations, and jerk and to the maximum generalized forces and their first order derivatives acting on the manipulator joints. In particular, it is possible to denote with $\boldsymbol{\tau} := [\tau_1 \; \tau_2 \; \dots \; \tau_N]^T$ and with $\dot{\boldsymbol{\tau}} := [\dot{\tau}_1 \; \dot{\tau}_2 \; \dots \; \dot{\tau}_N]^T$ the vector of the generalized forces and their derivatives, respectively. The scaling system must to modify the LTL in order to fulfill the following user-defined limits ($i = 1, 2, \dots, N$)

$$\underline{\tau}_i \leq \tau_i \leq \overline{\tau}_i \, , \tag{2.1}$$

$$\underline{\dot{\tau}}_i \leq \dot{\tau}_i \leq \overline{\dot{\tau}}_i \, , \tag{2.2}$$

$$\underline{\dot{q}}_i \leq \dot{q}_i \leq \overline{\dot{q}}_i \, , \tag{2.3}$$

$$\underline{\ddot{q}}_i \leq \ddot{q}_i \leq \overline{\ddot{q}}_i \, , \tag{2.4}$$

$$\underline{\dddot{q}}_i \leq \dddot{q}_i \leq \overline{\dddot{q}}_i \, , \tag{2.5}$$

where underlined and overlined terms respectively indicate lower and upper user-defined bounds on the considered entities and N is the dimension of the configuration space.

The planning scenario is introduced as follows (more details in [23]): System trajectories are defined by users or by an automatic planning systems according to the path-velocity decomposition paradigm [34], i.e. they are obtained by combining a LTL $u(t)$ with a path that is defined in the configuration space in the following manner

$$\begin{array}{rcc} \mathbf{f} : [0, u_f] & \rightarrow & \mathbb{R}^N \\ u & \rightarrow & \mathbf{q}_d := \mathbf{f}(u) \end{array} \, , \tag{2.6}$$

where scalar u parametrizes the path, u_f is its final value. The LTL is directly parametrized in function of the time and it is a monotonic function defined as follows

$$\begin{array}{rcc} u : [0, t_f] & \rightarrow & [0, u_f] \\ t & \rightarrow & u_d := u(t) \end{array} \, , \tag{2.7}$$

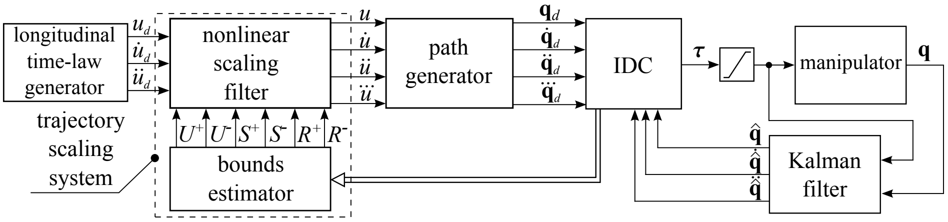

Figure 2.1: The control scheme used to drive the manipulator.

where t_f is the total traveling time. The overall trajectory is obtained by combining (2.6) and (2.7): $\mathbf{q}_d = \mathbf{f}[u(t)]$. Desired trajectory $\mathbf{q}_d$ can be differentiated by means of a chain differentiation rule in the following manner

$$\dot{\mathbf{q}}_d(u,\dot{u}) = \mathbf{f}'(u)\dot{u}\,, \tag{2.8}$$

$$\ddot{\mathbf{q}}_d(u,\dot{u},\ddot{u}) = \mathbf{f}''(u)\dot{u}^2 + \mathbf{f}'(u)\ddot{u}\,, \tag{2.9}$$

$$\dddot{\mathbf{q}}_d(u,\dot{u},\ddot{u},\dddot{u}) = \mathbf{f}'''(u)\dot{u}^3 + 3\mathbf{f}''(u)\dot{u}\ddot{u} + \mathbf{f}'(u)\dddot{u}\,. \tag{2.10}$$

Functions $\mathbf{f}(u)$ and $u(t)$ must be characterized by continuity properties. More precisely, for the sake of smoothness, the third derivative of $\mathbf{f}(u)$ with respect to u must be piecewise continuous, i.e., $\mathbf{f}'''(u) \in \mathscr{C}_p([0,u_f])$, while its first derivative, for regularity reasons, must be different from zero, i.e., $\mathbf{f}'(u) \neq \mathbf{0}$. Analogously, $\dddot{u}(t) \in \mathscr{C}_p([0,t_f])$. The scaling system, that was first proposed in [23], can be described with the aid of Fig. 2.1. A manipulator, that is affected by known dynamic/kinematic limits, is driven by means of an Inverse Dynamic Controller (IDC). The system actuators are not equipped with velocity and acceleration sensors and, moreover, the acquired position signals are particularly noisy. For this reason, the control feedback has been closed by means of a Kalman filter, that evidently introduces a non ideal behavior in the loop, since performances also depend on the accuracy of the model it uses.

Reference trajectories are obtained by combining the output of a LTL generator with the input of a path generator. A TSS is placed between them in order to generate, starting from the user defined reference $u_d(t)$, a signal $u(t)$ which derivatives fulfill the following equivalent bounds

$$R^- \leq \dot{u} \leq R^+, \qquad S^- \leq \ddot{u} \leq S^+, \qquad U^- \leq \dddot{u} \leq U^+\,. \tag{2.11}$$

Indeed, if (2.11) are satisfied, trajectory $\mathbf{q}_d(t)$, as will be proved in §2.3, is feasible with respect to the assigned kinematic/dynamic constraints. Bounds R^+, R^-, S^+, S^-, U^+, and U^- are evaluated at run time by the bound estimator block on the basis of the manipulator status of motion and of the controller status. This implies that bounds change at each sample time and that they are not repetitive even in case of cyclic trajectories like those that are later considered, due to the non ideal behavior of the system and to the presence of external disturbances.

The generation of a feasible signal $u(t)$ is demanded to the nonlinear scaling filter described in the Chapter 1. It is worth to resume its main characteristics: The filter returns $u(t) = u_d(t)$ if $u_d(t)$ is feasible, otherwise it generates an output $u(t)$ that is the best possible feasible approximation of $u_d(t)$ and that fulfills condition $\dddot{u}(t) \in \mathscr{C}_p([0,t_f])$. If $u_d(t)$, starting from an unfeasible state, newly becomes feasible, it is hanged by $u(t)$ in minimum time.

2.2 The manipulator model

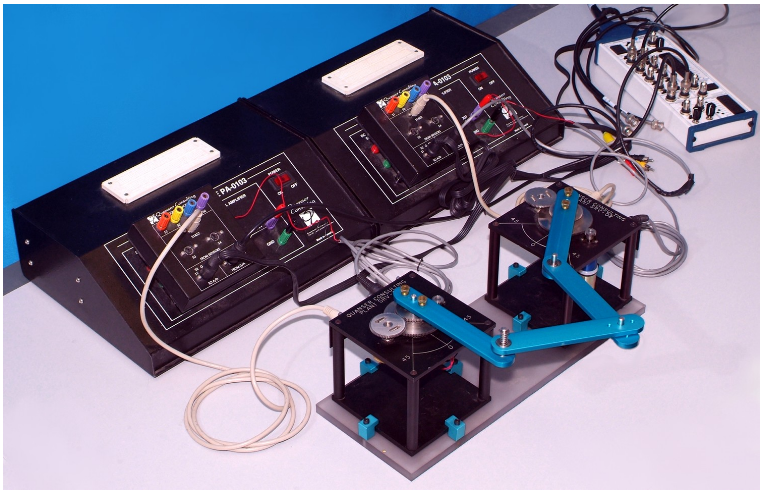

Figure 2.2: The experimental setup.

The experimental setup, that has been used for the validation of the scaling system, is shown in Fig. 2.2. It is constituted by a four links, closed-chain, planar manipulator,

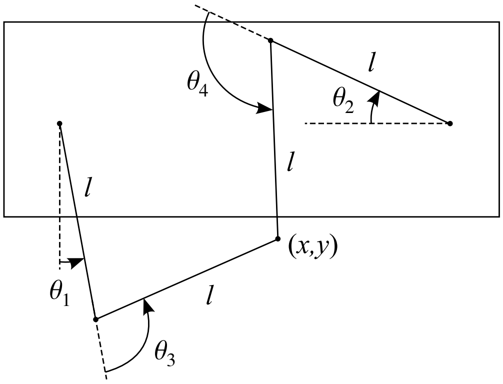

Figure 2.3: A schematic representation of the four links planar manipulator.

that is developed and distributed by Quanser Consulting Inc. for the fast prototyping of controllers. It is schematically represented in Fig. 2.3. Joints 1 and 2 are actuated by means of two DC motors, while all the other joints are passive. All links have the same length l and the same mass m_l. The system is described in the configuration space by angles θ_1 and θ_2.

The system dynamic equations can be obtained by means of the Euler-Lagrange approach. The potential energy of the whole system is constant, thus its evaluation is not necessary. The kinetic energy of the two motors and of the two links solidly connected to them can be expressed as follows

$$E_1 = \frac{1}{2} J_m \dot{\theta}_1^2 \,, \qquad E_2 = \frac{1}{2} J_m \dot{\theta}_2^2 \,,$$

where J_m represents the equivalent inertia, referred to the output shaft, of the combined system motor plus link.

An exact definition of the kinetic energy associated to links 3 and 4 should lead to a complex model. Conversely, a simplified model can be obtained by assuming some approximations. If $\theta_1 \simeq \theta_2 \simeq 0$ it is possible to write

$$E_3 = \frac{1}{2} J_l \dot{\theta}_2^2 + \frac{1}{2} m_l l^2 \dot{\theta}_1^2 \,, \qquad E_4 = \frac{1}{2} J_l \dot{\theta}_1^2 + \frac{1}{2} m_l l^2 \dot{\theta}_2^2 \,,$$

where J_l is the link inertia around the joint axis. Practically, kinetic energy associated to link 3 is supposed to be given by a translational component due to motor 1 and a rotational component caused by motor 2. This is clearly an approximation, since both components actually depend on θ_1, θ_2, $\dot{\theta}_1$, and $\dot{\theta}_2$. Model accuracy worsen if the system works far from the origin. These approximations are inserted in the IDC synthesis and in the Kalman filter model. The scaling system must also to be able to manage these system uncertainties in order to guarantee feasible trajectories.

The inverse dynamics equations are obtained by differentiating the Lagrangian equation $L = E_1 + E_2 + E_3 + E_4$, and by adding the dissipative effects ($i = 1,2$)

$$\tau_i = (J_m + J_l + m_l l^2)\ddot{\theta}_i + F\,\dot{\theta}_i\ , \tag{2.12}$$

where F is the viscous friction coefficient. Dissipative nonlinear terms have been neglected, thus introducing a further approximation.

The system input is represented by the motors voltages, so that the model has been rearranged in order to admit the same inputs. A reduction gear is placed between the output shaft and the motor shaft. Its reduction gear ratio is defined as follows

$$K_g := \frac{\theta_{m_i}}{\theta_i}\ , \tag{2.13}$$

where θ_{m_i} indicates the angular position of the motor shaft. Consequently, the relationship between output torques τ_i and motor torques τ_{m_i} is

$$\tau_i = K_g\,\tau_{m_i}\ . \tag{2.14}$$

Motor torques and feeding currents i_m are correlated by the electromechanics equation

$$\tau_{m_i} = K_m i_{m_i}\ , \tag{2.15}$$

where K_m is the motor torque constant. Finally, the electric equation expresses motor feeding voltages v_{m_i} in function of motor currents i_{m_i} and motor angular velocities $\dot{\theta}_{m_i}$

$$v_{m_i} = R_m i_{m_i} + K_m \dot{\theta}_{m_i}\ , \tag{2.16}$$

where R_m is the rotor resistance, while K_m is the back electromotive force constant, that is supposed to be equal to the motor torque constant.

By combining (2.12)–(2.16), after a few algebraic manipulations, it is possible to write the following dynamic equations ($i = 1,2$)

$$v_{m_i} = \alpha \ddot{\theta}_i + \beta \dot{\theta}_i \,, \tag{2.17}$$

where

$$\alpha = \frac{R_m}{K_m K_g}(J_m + J_l + m_l l^2) \,, \quad \beta = \frac{R_m}{K_m K_g} F + K_m K_g \,.$$

Due to the abovementioned simplifications, the model is linear and decoupled. This is clearly an approximation that can potentially affect the performances of the scaling system.

Hence, the simplified model of the whole 2R planar robot can be written as follows

$$\boldsymbol{\tau} = \mathbf{H}\ddot{\mathbf{q}} + \mathbf{F}\dot{\mathbf{q}} \,, \tag{2.18}$$

where $\boldsymbol{\tau} := [v_{m_1}\ v_{m_2}]^T$, $\mathbf{q} := [\theta_1\ \theta_2]^T$, while

$$\mathbf{H} := \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}, \quad \mathbf{F} := \begin{bmatrix} \beta & 0 \\ 0 & \beta \end{bmatrix}.$$

Instead of the feedforward torque controller with a feedback PD action proposed in [29], here an IDC is used due to its better performances. It is based on model (2.18) and it is characterized by the following equation

$$\boldsymbol{\tau} = \hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_d + \hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) + \hat{\mathbf{v}}(\mathbf{q},\dot{\mathbf{q}}) + \hat{\mathbf{H}}(\mathbf{q})\left(\mathbf{k}_p^T\mathbf{e} + \mathbf{k}_v^T\dot{\mathbf{e}} + \mathbf{k}_i^T \int \mathbf{e}\right) \,, \tag{2.19}$$

where symbol ˆ points out that the controller uses an estimated version of dynamics parameters α and β that are derived by means of an identification procedure, while $\mathbf{k}_p^T$, $\mathbf{k}_v^T$, and $\mathbf{k}_i^T$ are diagonal matrices that contain positive gains. The generalized joint variables derivatives are obtained from the Kalman filtering stage.

The time derivative of (2.19) can be written as follows (see [23])

$$\begin{aligned}\dot{\boldsymbol{\tau}} = {} & \dot{\hat{\mathbf{H}}}(\mathbf{q},\dot{\mathbf{q}})\ddot{\mathbf{q}}_d + \hat{\mathbf{H}}(\mathbf{q})\dddot{\mathbf{q}}_d + \hat{\mathbf{D}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + 2\hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\ddot{\mathbf{q}} + \hat{\mathbf{L}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{E}}(\mathbf{q},\dot{\mathbf{q}})\ddot{\mathbf{q}} \\ & + \hat{\mathbf{H}}(\mathbf{q})\left(\mathbf{k}_p^T\dot{\mathbf{e}} + \mathbf{k}_v^T\ddot{\mathbf{e}} + \mathbf{k}_i^T\mathbf{e}\right)s + \dot{\hat{\mathbf{H}}}(\mathbf{q},\dot{\mathbf{q}})\left(\mathbf{k}_p^T\mathbf{e} + \mathbf{k}_v^T\dot{\mathbf{e}} + \mathbf{k}_i^T\int\mathbf{e}\right) \,. \end{aligned} \tag{2.20}$$

Because of (2.9) and (2.10), (2.19) and (2.20) can be expressed in function of u and its derivatives

$$\boldsymbol{\tau}(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}}) = \hat{\mathbf{a}}_1(u;\mathbf{q})\ddot{u} + \hat{\mathbf{a}}_2(u,\dot{u};\mathbf{q}) + \hat{\mathbf{a}}_3(\mathbf{q},\dot{\mathbf{q}}) \,, \tag{2.21}$$

$$\dot{\boldsymbol{\tau}}(u,\dot{u},\ddot{u},\dddot{u};\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) = \hat{\mathbf{b}}_1(u;\mathbf{q})\dddot{u} + \hat{\mathbf{b}}_2(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}}) + \hat{\mathbf{b}}_3(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) \,, \tag{2.22}$$

where

$$\hat{\mathbf{a}}_1(u;\mathbf{q}) = \hat{\mathbf{b}}_1(u;\mathbf{q}) := \hat{\mathbf{H}}(\mathbf{q})\mathbf{f}'(u) \,, \tag{2.23}$$

$$\hat{\mathbf{a}}_2(u,\dot{u};\mathbf{q}) := \hat{\mathbf{H}}(\mathbf{q})\mathbf{f}''(u)\dot{u}^2 \,, \tag{2.24}$$

$$\hat{\mathbf{a}}_3(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) := \hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) + \hat{\mathbf{v}}(\mathbf{q},\dot{\mathbf{q}}) + \hat{\mathbf{H}}(\mathbf{q})\left(\mathbf{k}_p^T\mathbf{e} + \mathbf{k}_v^T\dot{\mathbf{e}} + \mathbf{k}_i^T\int\mathbf{e}\right), \tag{2.25}$$

$$\hat{\mathbf{b}}_2(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}}) := \dot{\hat{\mathbf{H}}}(\mathbf{q},\dot{\mathbf{q}})[\mathbf{f}''(u)\dot{u}^2 + \mathbf{f}'(u)\ddot{u}] + \hat{\mathbf{H}}(\mathbf{q})[\mathbf{f}'''(u)\dot{u}^3 + 3\mathbf{f}''(u)\dot{u}\,\ddot{u}] \,, \tag{2.26}$$

$$\begin{aligned}\hat{\mathbf{b}}_3(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) := {} & \hat{\mathbf{D}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + 2\hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\ddot{\mathbf{q}} + \hat{\mathbf{L}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{E}}(\mathbf{q},\dot{\mathbf{q}})\ddot{\mathbf{q}} \\ & + \hat{\mathbf{H}}(\mathbf{q})\left(\mathbf{k}_p^T\dot{\mathbf{e}} + \mathbf{k}_v^T\ddot{\mathbf{e}} + \mathbf{k}_i^T\mathbf{e}\right) + \dot{\hat{\mathbf{H}}}(\mathbf{q},\dot{\mathbf{q}})\left(\mathbf{k}_p^T\mathbf{e} + \mathbf{k}_v^T\dot{\mathbf{e}} + \mathbf{k}_i^T\int\mathbf{e}\right).\end{aligned} \tag{2.27}$$

and where $\hat{\mathbf{a}}_1(u;\mathbf{q}) = \hat{\mathbf{b}}_1(u;\mathbf{q}) = [\hat{a}_{11},\hat{a}_{12},\ldots,\hat{a}_{1N}]^T$, $\hat{\mathbf{a}}_2(u,\dot{u};\mathbf{q}) = [\hat{a}_{21},\hat{a}_{22},\ldots,\hat{a}_{2N}]^T$, $\hat{\mathbf{b}}_2(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}}) = [\hat{b}_{21},\hat{b}_{22},\ldots,\hat{b}_{2N}]^T$, $\hat{\mathbf{a}}_3(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) = [\hat{a}_{31},\hat{a}_{32},\ldots,\hat{a}_{3N}]^T$, $\hat{\mathbf{b}}_3(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) = [\hat{b}_{31},\hat{b}_{32},\ldots,\hat{b}_{3N}]^T$, while $\mathbf{e} := \mathbf{q} - \mathbf{q}_d$, $\dot{\mathbf{e}} := \dot{\mathbf{q}} - \dot{\mathbf{q}}_d$, and $\ddot{\mathbf{e}} := \ddot{\mathbf{q}} - \ddot{\mathbf{q}}_d$ represent, respectively, the trajectory tracking error and its first and second time derivatives. Also in this case, symbol ˆ points out that the controller uses an estimated model of the manipulator.

2.3 Evaluation of the equivalent longitudinal constraints

Firstly, the generalized force and the generalized force derivative constraints have to be converted to an equivalent version that can be used to bound the scaling filter dynamics. Equations (2.1) and (2.2), owing to (2.21) and (2.22), can be reformulated in the following manner ($i = 1,2,\ldots,N$):

$$\underline{\tau}_i \leq \hat{a}_{1i}(u;\mathbf{q})\ddot{u} + \hat{a}_{2i}(u,\dot{u};\mathbf{q}) + \hat{a}_{3i}(\mathbf{q},\dot{\mathbf{q}}) \leq \overline{\tau}_i \,, \tag{2.28}$$

Table 2.1: Expressions used for the evaluation of the equivalent longitudinal bounds.

	$f_i' > 0$	$f_i' < 0$	$f_i' = 0$
$\overline{\sigma}_i$	$(\dddot{\overline{q}}_i - f_i'''\dot{u}^3 - 3f_i''\dot{u}\ddot{u})/f_i'$	$(\dddot{\underline{q}}_i - f_i'''\dot{u}^3 - 3f_i''\dot{u}\ddot{u})/f_i'$	∞
$\underline{\sigma}_i$	$(\dddot{\underline{q}}_i - f_i'''\dot{u}^3 - 3f_i''\dot{u}\ddot{u})/f_i'$	$(\dddot{\overline{q}}_i - f_i'''\dot{u}^3 - 3f_i''\dot{u}\ddot{u})/f_i'$	$-\infty$
$\overline{\mu}_i$	$(\ddot{\overline{q}}_i - f_i''\dot{u}^2)/f_i'$	$(\ddot{\underline{q}}_i - f_i''\dot{u}^2)/f_i'$	∞
$\underline{\mu}_i$	$(\ddot{\underline{q}}_i - f_i''\dot{u}^2/)f_i'$	$(\ddot{\overline{q}}_i - f_i''\dot{u}^2)/f_i'$	$-\infty$
$\overline{\zeta}_i$	$\dot{\overline{q}}_i/f_i'$	$\dot{\underline{q}}_i/f_i'$	∞
$\underline{\zeta}_i$	$\dot{\underline{q}}_i/f_i'$	$\dot{\overline{q}}_i/f_i'$	$-\infty$
	$\hat{b}_{1i} > 0$	$\hat{b}_{1i} < 0$	$\hat{b}_{1i} = 0$
$\overline{\gamma}_i$	$[\dot{\overline{\tau}}_i - \hat{b}_{2i}(\dot{u},\ddot{u}) - \hat{b}_{3i}]/\hat{b}_{1i}$	$[\dot{\underline{\tau}}_i - \hat{b}_{2i}(\dot{u},\ddot{u}) - \hat{b}_{3i}]/\hat{b}_{1i}$	∞
$\underline{\gamma}_i$	$[\dot{\underline{\tau}}_i - \hat{b}_{2i}(\dot{u},\ddot{u}) - \hat{b}_{3i}]/\hat{b}_{1i}$	$[\dot{\overline{\tau}}_i - \hat{b}_{2i}(\dot{u},\ddot{u}) - \hat{b}_{3i}]/\hat{b}_{1i}$	$-\infty$
	$\hat{a}_{1i} > 0$	$\hat{a}_{1i} < 0$	$\hat{a}_{1i} = 0$
$\overline{\beta}_i$	$[\overline{\tau}_i - \hat{a}_{2i}(\dot{u}) - \hat{a}_{3i}]/\hat{a}_{1i}$	$[\underline{\tau}_i - \hat{a}_{2i}(\dot{u}) - \hat{a}_{3i}]/\hat{a}_{1i}$	∞
$\underline{\beta}_i$	$[\underline{\tau}_i - \hat{a}_{2i}(\dot{u}) - \hat{a}_{3i}]/\hat{a}_{1i}$	$[\overline{\tau}_i - \hat{a}_{2i}(\dot{u}) - \hat{a}_{3i}]/\hat{a}_{1i}$	$-\infty$

$$\dot{\underline{\tau}}_i \leq \hat{b}_{1i}(u;\mathbf{q})\dddot{u} + \hat{b}_{2i}(u,\dot{u},\ddot{u};\mathbf{q},\dot{\mathbf{q}}) + \hat{b}_{3i}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) \leq \dot{\overline{\tau}}_i \,. \tag{2.29}$$

The relations (2.28) and (2.29) can be rearranged in order to convert the original generalized force bounds into equivalent limits compatible with the longitudinal time-law. To this purpose, for example, joint i fulfills (2.28) if $\ddot{u} \in \left[\underline{\beta}_i\, \overline{\beta}_i\right]$ where $\underline{\beta}_i$ and $\overline{\beta}_i$ are defined in Table 2.1. The same reasoning must be applied for each joint at the same time, so that the generalized force constraint (2.1) is satisfied only if $\ddot{u} \in \bigcap_{i=1}^{N} \left[\underline{\beta}_i\, \overline{\beta}_i\right]$. Similar considerations apply for the derivative of the generalized force constraint: Considered constraint (2.2) holds only if $\dddot{u} \in \left[\underline{\gamma}_i\, \overline{\gamma}_i\right]$ where $\underline{\gamma}_i$ and $\overline{\gamma}_i$ are defined, also in this case, in Table 2.1.

The second family of the contemplated bounds are related to the joint velocities, accelerations and jerks. Similar to the above procedure, these constraints must to be converted into equivalent bounds for the longitudinal time-law. Hence, by considering relations (2.8)–(2.10), joint space bounds (2.3)–(2.5) can be reformulated in

$$\dot{\underline{q}}_i \leq f_i{}'(u)\dot{u} \leq \dot{\overline{q}}_i \,, \tag{2.30}$$

$$\underline{\ddot{q}}_i \leq f_i{}''(u)\dot{u}^2 + f_i{}'(u)\ddot{u} \leq \overline{\ddot{q}}_i\,, \tag{2.31}$$

$$\underline{\dddot{q}}_i \leq f_i{}'''(u)\dot{u}^3 + 3f_i{}''(u)\dot{u}\ddot{u} + f_i{}'(u)\dddot{u} \leq \overline{\dddot{q}}_i\,. \tag{2.32}$$

Follow the same reasoning applied for relations (2.28) and (2.29), it is possible to assert that the velocity bound (2.3) is satisfied only if $\dot{u} \in \bigcap_{i=1}^{N} \left[\underline{\zeta}_i\ \overline{\zeta}_i\right]$, the acceleration bound (2.4) is fulfilled only if $\ddot{u} \in \bigcap_{i=1}^{N} \left[\underline{\mu}_i\ \overline{\mu}_i\right]$, and the jerk bound (2.5) holds only if $\dddot{u} \in \bigcap_{i=1}^{N} [\underline{\sigma}_i\ \overline{\sigma}_i]$. The values $\underline{\zeta}_i$, $\overline{\zeta}_i$, $\underline{\mu}_i$, $\overline{\mu}_i$, $\underline{\sigma}_i$, and $\overline{\sigma}_i$ are given in Table 2.1.

In conclusion, equivalent bounds R^+, R^-, S^+, S^-, U^+, and U^- are obtained from (2.1)–(2.5), by also considering (2.28)–(2.32). In particular, they are calculated as follows

$$R^- := \max_{i=1,\dots,N}\{\underline{\zeta}_i\}, \quad R^+ := \min_{i=1,\dots,N}\{\overline{\zeta}_i\}, \tag{2.33}$$

$$S^- := \max_{i=1,\dots,N}\{\underline{\beta}_i, \underline{\mu}_i\}, \quad S^+ := \min_{i=1,\dots,N}\{\overline{\beta}_i, \overline{\mu}_i\}, \tag{2.34}$$

$$U^- := \max_{i=1,\dots,N}\{\underline{\gamma}_i, \underline{\sigma}_i\}, \quad U^+ := \min_{i=1,\dots,N}\{\overline{\gamma}_i, \overline{\sigma}_i\}. \tag{2.35}$$

2.4 Feasibility problems

Feasibility problems, that could arise when real-time scaling systems are used, can be explained with the aid of Fig. 2.4, that shows some hypothetical scenarios in which feasibility is lost with certainty. In all the considered cases, $\dot{u}$ is supposed to be positive, but a similar analysis can be performed to account for problems that could arise when $\dot{u} < 0$.

Firstly, consider Case a, in which $\dot{u}$ reaches R^+. Owing to the continuity of $\ddot{u}$, the velocity cannot be immediately reduced, so that the constraint is certainly violated. Short-time constraints violations do not alter the system performances – motors slightly violate the maximum speed for brief periods – but the situation can become critical if, as shown in Fig. 2.4, $\ddot{u}$ has reached its lower bound S^-: $\dot{u}$ decrements slowly, constraints are violated for long periods, and the system stability is endangered. A similar situation occurs in Case b, but, in this second case, the constraint violation is caused by the saturation on $\dddot{u}$.

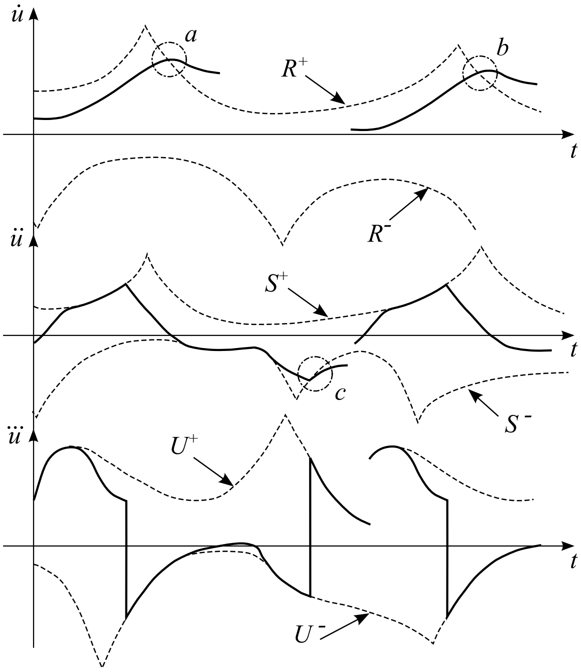

Figure 2.4: Possible constraints violations.

An even worst situation occurs in Case c: $\dddot{u}$ reaches U^+, so that the acceleration limit cannot be fulfilled. Unfortunately, the acceleration constraint is correlated with the maximum admissible motor torques: In case of violations motors saturate, thus the controller performances worsen or, in the worst cases, stability is lost.

Critical situations are evidently caused by the proximity to zero of the acceleration and of the jerk bounds, thus the obvious solution is based on a proper relaxation of such bounds. In order to understand how it can be done, in the following is briefly resumed the equations that are used for the constraints evaluation.

Equations (2.33)–(2.35) confirm that, for any point along the path, the equivalent constraints are influenced by the status of motion of the system, i.e., by $\dot{u}$ and $\ddot{u}$. Thus, by acting on $\dot{u}$ and $\ddot{u}$, it is possible to modify the longitudinal bounds.

In particular, considering first $\ddot{u}$ and Case a of Fig. 2.4, in the following is analyzed the relation between the bounds defined in Table 2.1 and the feasibility problems. As previously reported, problems are caused by values of S^-, and, in turn, of

$\underline{\beta}_i$ and $\underline{\mu}_i$, that are too close to zero. Terms $\underline{\mu}_i$ approach zero if terms f_i' assume large values or if products $f_i''\dot{u}^2$ dominate over $\underline{\ddot{q}}_i$ (or over $\overline{\ddot{q}}_i$). In the first case nothing can be done, since terms f_i' only depend on the path shape. In the second case, an appropriate reduction of $\dot{u}$ can produce an enlargement of the bounds. A similar consideration holds for terms $\underline{\beta}_i$: No action can be taken if their approach to zero is caused by $\hat{a}_{3i}$ or by $\hat{a}_{1i}$. Conversely, bearing also in mind (2.24), if problems derive from the amplitude of $\hat{a}_{2i}(\dot{u})$, positive effects can, again, be obtained by reducing $\dot{u}$. In both cases, the reduction of $\dot{u}$ can also be justified through physical reasons. Terms $\underline{\beta}_i$ are directly correlated to (2.1), while terms $\underline{\mu}_i$ descend from (2.4): By decreasing $\dot{u}$, motor torques and accelerations are certainly reduced, so that constraint violations are less probable.

Similar conclusions can be drawn for Cases *b* and *c*: The modules of S^+, S^-, U^+, and U^- generally increase if $\dot{u}$ is decreased, e.g., by properly downscaling R^+, when $\dot{u} > 0$, or R^-, when $\dot{u} < 0$. It is worth to mention that productivity reasons suggest to avoid global reductions of R^+ (or of R^-) that, conversely, should be limited to the sole critical points. In the following §2.4.1 and §2.4.2 are reported strategies that properly scale down R^+ or R^- in order to limit feasibility problems that could afflict the TSS.

2.4.1 Local scaling of R^+ and R^-

Several techniques have been tested in order to properly scale R^+ and R^-. The best performances in terms of promptness and reliability have been achieved by means of Algorithm 1. It is executed at the same sample rate of the main control loop and it is based on a simple threshold approach. Three different thresholds, namely $\overline{S}$, $\overline{U}$, and $\widetilde{U}$, have been chosen respectively on S^-, U^+, and U^-. If one of such thresholds is violated, i.e., if one of the three bounds approaches zero, R^+ is posed equal to zero, so that, necessarily, $\dot{u}$ decreases. This causes an enlargement of the acceleration and of the jerk bounds, so that the original limit R^+ is restored in a short time: As shown in §2.5, sufficiently large bounds on the acceleration and the jerk can be achieved by adopting an upper velocity constraint that properly choppers between 0 and R^+.

Thresholds $\overline{S}$, $\overline{U}$, and $\widetilde{U}$ must be properly chosen. In order to achieve the best performances, the actual shapes of R^+, R^-, S^+, and S^- should be known in advance.

Algorithm 1 Downscale R^+ and R^- when the acceleration and jerk limits are not sufficient to guarantee the fulfillment of the velocity bounds.

```
Evaluate $R^+$
**if** $(\dot{u} > 0)$ **and** $\Big[(|S^-| < \overline{S})$ **or** $(U^+ < \overline{U})$ **or** $(|U^-| < \widetilde{U})\Big]$ **then**
    $R^+ = 0$
**end if**
Evaluate $R^-$
**if** $(\dot{u} < 0)$ **and** $\Big[(S^+ < \overline{S})$ **or** $(|U^-| < \overline{U})$ **or** $(U^+ < \widetilde{U})\Big]$ **then**
    $R^- = 0$
**end if**
```

For example, in Case *a* of Fig. 2.4, the knowledge of the slope of R^+ could be used to properly scale bound S^-. Unfortunately, longitudinal bounds depend on the system status and, consequently, they are only roughly known, being influenced by many external factors. For this reason, $\overline{S}$, $\overline{U}$, and $\widetilde{U}$ are chosen with a heuristic rule. In general, sufficiently good performances are verified by adopting an initial value of $\overline{S}$ that is one order of magnitude larger than the average longitudinal velocity, while $\overline{U}$ and $\widetilde{U}$ should be two orders of magnitude larger. Tunings can be later refined on the actual plant.

It is worth to remark that the technique here proposed solves many of the issues generated by an improper choice of the longitudinal time-law, but, since bounds S^+, S^-, U^+, and U^- are also influenced by terms that do not directly depend on $\dot{u}$, feasibility cannot be guaranteed with certainty: The system can still enter into regions from which the feasibility cannot be maintained. The convergence toward unfeasible regions is pointed out by a sudden shrinkage of the feasible intervals: U^+ and U^- approach each other (and/or the same happens to S^+ and S^-), so that it is no more possible to respect the given constraints. If this happens, the user must choose a different time-law or, alternatively, he must globally scale down R^+ and R^-. In case of cyclic time-laws, a completely automatic approach, like the one that is proposed in the next section, can be adopted.

2.4.2 Global scaling of R^+ and R^-

If the local approach, presented in §2.4.1, is not able to solve all feasibility issues, remaining problems can be handled by further scaling the equivalent velocity bounds according to the following equations

$$\widetilde{R}^+ = \psi R^+, \qquad \widetilde{R}^- = \psi R^-,$$

where $\psi \in [0.1, 1]$. Scaling factor ψ applies to the whole trajectory, so it cannot be chosen too small in order to guarantee a proper tracking of $u_d(t)$.

Algorithm 2 Evaluation of the error signal used by the global bound-scaling system. M indicates the number of samples occurring in a period T, while e_j is the error signal at time $t = jT$.

```
$\tilde{e} = \infty$; $\bar{e} = \infty$
for $i = 1$ to $M$ do
    if $\dot{u}_i > 0$ then
        $e_i = \min\{U_i^+ - \overline{U}, |U_i^-| - \widetilde{U}\}$
    else
        $e_i = \min\{|U_i^-| - \overline{U}, U_i^+ - \widetilde{U}\}$
    end if
    if $e_i < \tilde{e}$ then
        $\tilde{e} = e_i$
    end if
    $e_i = \min\{S_i^+ - \overline{S}, |S_i^-| - \overline{S}\}$
    if $e_i < \bar{e}$ then
        $\bar{e} = e_i$
    end if
end for
$e_j = [\mathrm{sign}(\bar{e})\ \&\ \mathrm{sign}(\tilde{e})]\ |\bar{e}\,\tilde{e}|$
```

The selection of ψ can be automated in case of cyclic trajectories by adopting a strategy that maximizes its value compatibly with the assigned constraints: If the local bound-scaling system proposed in §2.4.1 is sufficient to guarantee the feasibility, the global scaling system must converge to $\psi = 1$.

Term ψ is obtained by means of the following discrete-time PI regulator, characterized by a sample time T that is equal to the trajectory period

$$\psi_j = \psi_{j-1} + K_P\, e_j + (K_I\, T - K_P)\, e_{j-1}\ , \tag{2.36}$$

where e_j is evaluated according to Algorithm 2.

The controller task is to maintain U^+, $|U^-|$, S^+, and $|S^-|$ as close as possible to the given thresholds, i.e., to $\overline{U}$, $\widetilde{U}$, and $\overline{S}$, by increasing ψ. If such thresholds, due to the considered trajectory, cannot be violated, the controller converges toward $\psi = 1$, otherwise lower values of ψ are obtained. Since ψ is bounded between 0.1 and 1, the controller is equipped with an anti-windup system.

2.5 Experimental results

The TSS and, in particular, the bound-scaling techniques proposed in §2.4 for the avoidance of feasibility issues, have been tested on the planar 2R robot that is shown in Fig. 2.2. As asserted in §2.2, the system is not equipped with torque control loops, so that its motors are directly voltage driven. The model coefficients, that have been obtained through an identification procedure, have been successively used, according to control approach proposed in [23], to implement the IDC. The evaluation of the equivalent bounds R^+, R^-, S^+, S^-, U^+, and U^-, following the procedure defined in §2.3, has been accomplished by assuming the following limits ($i = 1, 2$): $\dot{\overline{q}}_i = -\dot{\underline{q}}_i = 6$ rad s^{-1}, $\ddot{\overline{q}}_i = -\ddot{\underline{q}}_i = 70$ rad s^{-2}, $\dddot{\overline{q}}_i = -\dddot{\underline{q}}_i = 3 \cdot 10^3$ rad s^{-3}, $\overline{\tau}_i = -\underline{\tau}_i = 5$ V, and $\dot{\overline{\tau}}_i = -\dot{\underline{\tau}}_i = 35$ V s^{-1}. The scaling algorithm has been checked by means of the following three paths

Path 1:

$$\mathbf{f}_1(u) = \begin{bmatrix} \theta_{10} + a\cos(2\pi u) \\ \theta_{20} + a\sin(2\pi u) \end{bmatrix}, \tag{2.37}$$

where $\theta_{10} = 8$ deg, $\theta_{20} = 3$ deg, and $a = 25$ deg;

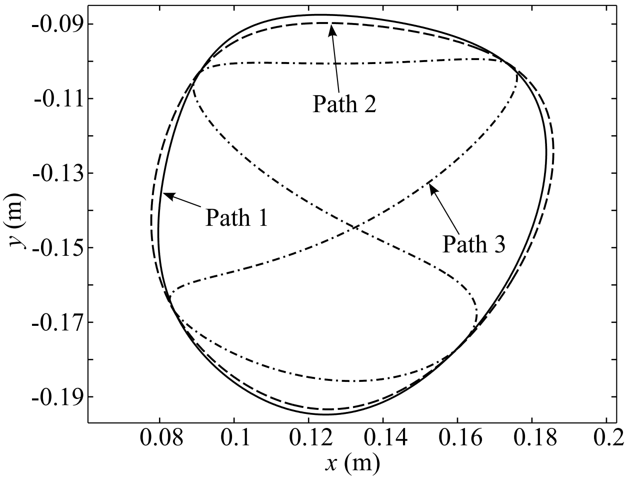

Figure 2.5: The three paths that have been used for the experiments.

Path 2:

$$\mathbf{f}_2(u) = \begin{bmatrix} \theta_{10} + a\cos(2\pi u) + b\cos(6\pi u) \\ \theta_{20} + a\sin(2\pi u) + b\sin(6\pi u) \end{bmatrix}, \tag{2.38}$$

where $\theta_{10} = 8$ deg, $\theta_{20} = 3$ deg, $a = 25$ deg, and $b = 1$ deg;

Path 3:

$$\mathbf{f}_3(u) = \begin{bmatrix} \theta_{10} + a\cos(2\pi u) \\ \theta_{20} + a\sin(4\pi u) \end{bmatrix}, \tag{2.39}$$

where $\theta_{10} = 8$ deg, $\theta_{20} = 3$ deg, and $a = 20$ deg. Paths 1 and 2, as shown in Fig. 2.5, have very similar shapes, but they differently solicit the control system. This can be evinced from Fig. 2.6, that compares functions $\mathbf{f}_i'''(u)$ of the three paths. In particular, values assumed by $\mathbf{f}_2'''(u)$ are higher than those assumed by $\mathbf{f}_1'''(u)$, so that, due to the expressions of $\underline{\sigma}_i$, $\overline{\sigma}_i$, $\underline{\mu}_i$, and $\overline{\mu}_i$ defined in Table 2.1, bounds U^+ and U^- of Path 2 create feasibility issues. From this point of view, Path 3 is, thus, the most critical.

The following three thresholds have been used for the bound-scaling system: $\overline{U} = 500$, $\widetilde{U} = 20$, and $\overline{S} = 15$. They were chosen on the basis of the criteria that were previously specified, with the exception of $\widetilde{U}$. Indeed, experimental tests have shown that, for the proposed trajectories, small values of $\widetilde{U}$ do not cause constraint violations, so that $\widetilde{U}$ is only used to guarantee that $U^- < 0$.

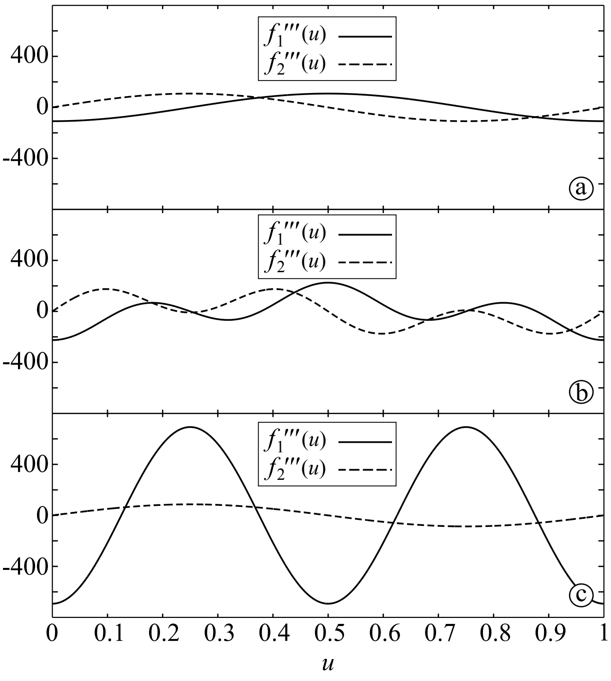

Figure 2.6: Function $\mathbf{f}'''(u)$ for the three paths.

For Path 1, as shown in Fig. 2.7, the outputs of the scaling filter fulfill the equivalent longitudinal bounds, so that conditions (2.1)–(2.5), as proved by Fig. 2.8, are fulfilled as well. Fig. 2.7a also proves a relevant characteristic of the proposed scaling system: Differently from other approaches (see e.g. [46]), time delays, that have been accumulated in order to satisfy the constraints, are eliminated, thus the system efficiency is preserved. The LTL has been specifically chosen unfeasible in order to mimic a situation in which the user is unaware of the system limits: Without the aid of the two bound-scaling mechanisms, the system stability is lost in all the three test cases. Fig. 2.7b highlights the combined actions of the two bound-scaling systems. In particular, the commutations of $\widetilde{R}^{+}$, that are pointed out by means of dash-dotted circles, are caused by the local mechanism, while the effects of the global strategy are demonstrated by the reduced values of the velocity limits: In order to preserve feasibility, ψ has converged to 0.47, i.e., the user selected bounds have been scaled by more than the 50%.

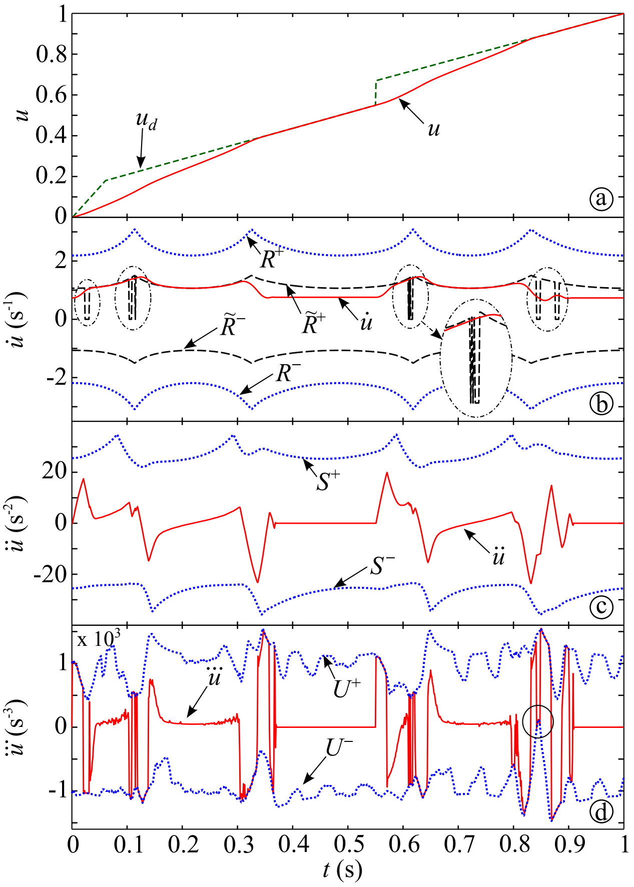

Figure 2.7: Path 1: a) Position reference signal u_d (dashed line) compared with the filter output u (solid line); b) filter output $\dot{u}$ (solid line) compared with velocity bounds $\widetilde{R}^+$ and $\widetilde{R}^-$ (dashed lines) and with unscaled bounds R^+ and R^-; c) and d) bounds S^+ and S^-, U^+, and U^- (dotted lines) compared with filter outputs $\ddot{u}$ and $\dddot{u}$ (solid line).

Experimental results have also proved that the best performances are obtained thanks to the combined action of the two bound-scaling stages. Indeed, if the local bound-scaling system is disabled and the initial value of ψ is chosen far from its optimal value, the system becomes unstable during the first revolutions. Conversely, still considering a disabled local bound-scaling system, if ψ is initially chosen close to its optimal value, stability is preserved, but lower final values of ψ are obtained, i.e., worse system performances are detected.

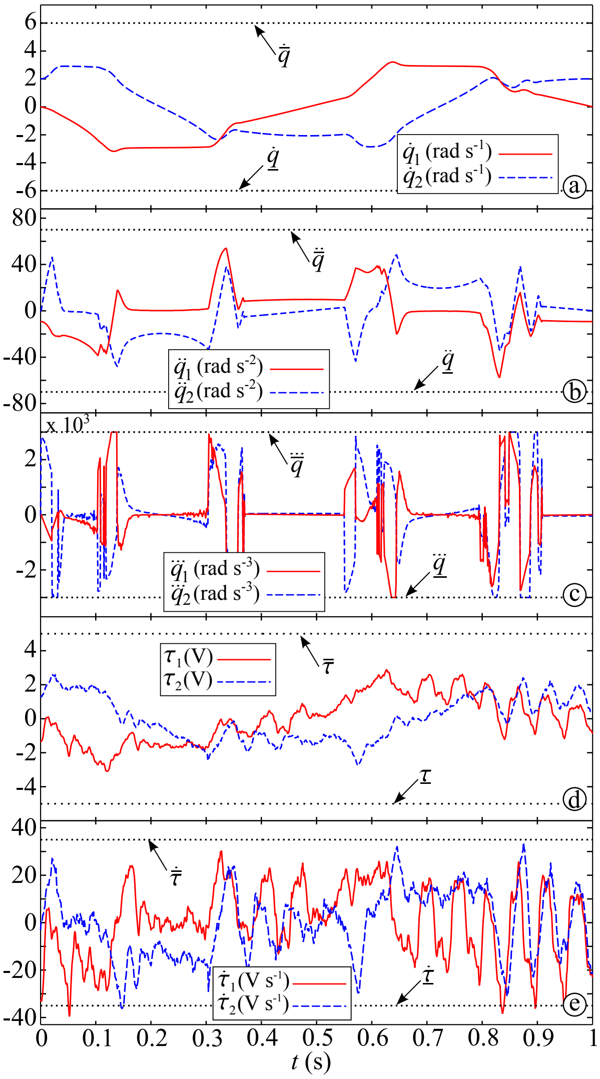

Figure 2.8: Path 1: a) joint velocities; b) joint accelerations; c) joint jerks; d) motor feeding voltages; e) derivatives of the motor feeding voltages.

Similar behaviors have been verified for the other two paths. In particular, since the same time-law is used for Path 1 and Path 2, worse performances were expected for the second path due to the larger values assumed by $\mathbf{f}_2'''(u)$. On the contrary, thanks to the bound-scaling system, similar performances have been obtained and the results are, thus, omitted for the sake of conciseness.

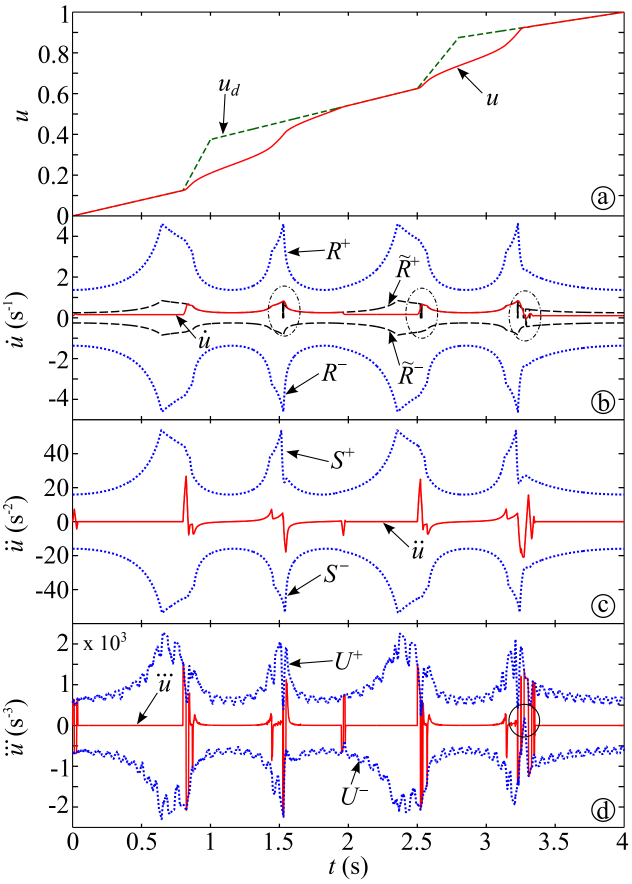

Figure 2.9: Path 3: a) Position reference signal u_d (dashed line) compared with the filter output u (solid line); b) filter output $\dot{u}$ (solid line) compared with velocity bounds $\widetilde{R}^+$ and $\widetilde{R}^-$ (dashed lines) and with unscaled bounds R^+ and R^-; c) and d) bounds S^+ and S^-, U^+, and U^- (dotted lines) compared with filter outputs $\ddot{u}$ and $\dddot{u}$ (solid line).

Conversely, different results have been obtained for the critical Path 3. In particular, owing to the very large values of $\mathbf{f}_3'''(u)$, the global bound-scaling system presented in §2.4.2, has converged to a smaller value of ψ, i.e., $\psi = 0.18$, in order to fulfill, as proved by Fig. 2.9, the equivalent longitudinal bounds.

The solid circles in Figs. 2.7d and 2.9d show that, in order to guarantee that $U^- < 0$ and $U^+ > 0$, higher values of ψ cannot be achieved.

The good behavior of the TSS is proved by the small path tracking errors that have been detected for the three cases. In particular, the following root-mean-square and peak errors have been measured:

- Path 1: $e_{rms} = 4.40 \cdot 10^{-4}$ m, $e_{max} = 1.11 \cdot 10^{-3}$ m;
- Path 2: $e_{rms} = 4.81 \cdot 10^{-4}$ m, $e_{max} = 1.62 \cdot 10^{-3}$ m;
- Path 3: $e_{rms} = 4.02 \cdot 10^{-4}$ m, $e_{max} = 1.27 \cdot 10^{-3}$ m.

It is interesting to notice that, since the system constraints are always fulfilled, similar tracking errors have been obtained for the three examples: They are mainly due to the IDC tuning.

The TSS defined in the configuration space permits also to maintain the trajectory feasibility with respect to system uncertainties (introduced in the identification process and in the IDC's realization) and to paths that different solicit the system dynamic behavior. The experimental results, reported in the previous version of the TSS [44], highlight the robustness of the TSS with respect system uncertainties: In those experiments the estimated dynamics parameters $\hat{\alpha}$ and $\hat{\beta}$ have been perturbed (increased and decreased by 30%) and the resulting path tracking errors have been proved the TSS robustness.

On the other hand, if trajectories are defined in the operational space, a different scaling algorithm has to be developed: The user-defined limits could potentially be defined directly in the Cartesian space and a different evaluation of the equivalent bounds on the longitudinal time-law has to be formalized. The problem formulation and a possible solution will be reported in the next chapter.

3

Trajectory scaling systems in the operational space

Planning trajectories in the operational space could elicit unfeasible joint velocities and accelerations that are mainly due to kinematic singularities. A kinematic singularity is a point defined in the operational space where, even in the event of slow Cartesian motions, high joint velocities and accelerations must be expected. Dealing with singularities is a problem that is widely discussed in literature and several techniques have been proposed in order to avoid singular points. The problem has often been solved offline because of the highly coupled relations that exist between the configuration space and the operational space.

The singularity avoidance problem can be addressed in several ways depending on the number of independent joints of the manipulator. A manipulator is defined redundant if the number of independent joints is greater than the DoFs of the robot or non-redundant in the other case.

In case of redundant manipulators, the solution to the problem can be found by acting on the available DoFs that derive from the exploitation of the null space in the Jacobian matrix (see, for example [47,48]). For non-redundant manipulators, known approaches could be roughly divided into two categories. In the first, minor path changes are admitted in order to avoid critical configurations, while in the second, planned trajectories are slowed down in the vicinity of singular configurations in order to preserve accurate path tracking. The choice of the most appropriate approach is dictated by the manipulator tasks. If the execution time is mandatory, the first strategy is preferred – a solution based on this approach will be presented in next Chapter 4. Conversely, if deviations from the planned path are not admissible, kinematic bounds must be handled with the second approach. The aim of this chapter is to present a scaling system developed according the second approach.

Robotic literature proposes only a few works which address problems in the Cartesian space. Early works have been proposed in [39] and in [49] where, in an offline context, a trajectory optimization problem in the Cartesian space has been solved in order to constrain joint accelerations and jerks respectively. As stated in Chapter 2, if the trajectory planner must be computed in real-time, an approach based on Trajectory Scaling Systems (TSS) could be used in order to constrain Cartesian trajectories within some specified bounds. In this chapter it will be shown that even for trajectories in the Cartesian space, despite the increased problem complexity, the TSS computational time is fully compatible with real-time implementation. Recently, trajectories planned in the operational space have been, for the first time, handled online in [50] to satisfy velocity and acceleration constraints. In the following sections a more challenging problem will be defined and solved: The TSS has been modified to manage jerk constraints in real-time. §3.4 is dedicated to comparisons between the TSS proposed in [50] and the novel TSS.

3.1 Problem formulation

The scaling problem here formulated is similar to the problem exposed in §2.1 but, in this case, trajectories are defined in the operational space. Besides the limits on joint velocities, accelerations and jerks, constraints directly expressed in the operational space must be real-time evaluated and managed in order to guarantee smooth movements of the robot end effector and the overall trajectory feasibility. The scaling problem can be formulated in the following manner.

Given a trajectory in the operational space and a set of kinematic constraints, the TSS here proposed appropriately modifies the longitudinal speed in order to preserve the path tracking. To this aim, trajectories are planned according to the path-velocity decomposition paradigm [34], an approach that is also suited for planning scenarios in the operational space. Practically, trajectories are obtained as a combination of a path in the operational space and a LTL. By defining s as the Euclidean distance from the beginning of the path, measured along the path itself, the LTL is specified by assigning the following function

$$\begin{array}{rcc} s : [0, t_f] & \rightarrow & \mathbb{R}^+ \\ t & \rightarrow & s := s(t) \end{array} , \tag{3.1}$$

where t_f is the total traveling time. Along this and the following sections, term “path” will indicate the combination of the positions and the orientations that are assumed by the tool frame or by the wrist frame during the motion. Positions of the robot tool frame are specified through a function $\mathbf{p}_T(s)$ that is defined as follows

$$\begin{array}{rcc} \mathbf{p}_T : [0, s_f] & \rightarrow & \mathbb{R}^3 \\ s & \rightarrow & \mathbf{p}_T := \mathbf{p}_T(s) \end{array} , \tag{3.2}$$

where $s_f = s(t_f)$ is the path length. Orientations can be specified in two different manners, depending on the planning strategy. They can be expressed, e.g., by the following function

$$\begin{array}{rcc} \mathbf{\Phi}_T : [0, s_f] & \rightarrow & \mathbb{R}^3 \\ s & \rightarrow & \mathbf{\Phi}_T := \mathbf{\Phi}_T(s) \end{array} , \tag{3.3}$$

where $\mathbf{\Phi}_T$ indicates a tool frame orientation that is expressed through any minimal representation. Alternatively, orientations could also be described by means of a rotation matrix

$$\begin{array}{rcc} {}^0_T\mathbf{R} : [0, s_f] & \rightarrow & \mathbb{R}^9 \\ s & \rightarrow & {}^0_T\mathbf{R} := {}^0_T\mathbf{R}(s) \end{array} . \tag{3.4}$$

Depending on the selected planning scheme, the trajectory is thus given by functions $\mathbf{p}_T(t) = \mathbf{p}_T[s(t)]$ and $\mathbf{\Phi}_T(t) = \mathbf{\Phi}_T[s(t)]$ or, alternatively, by $\mathbf{p}_T(t) = \mathbf{p}_T[s(t)]$ and ${}^0_T\mathbf{R}(t) = {}^0_T\mathbf{R}[s(t)]$. Position and orientations are measured with respect to an inertial frame. Without any loss of generality, it will coincide with the manipulator 0 frame.

Differently from the TSS in the configuration space presented in Chapter 2, here, manipulator control loops cannot be accessed and, therefore, the TSS manages only

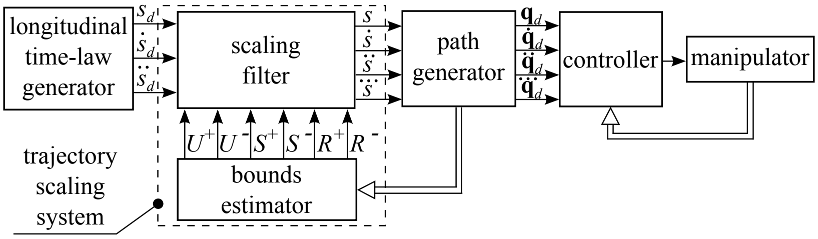

Figure 3.1: The overall manipulator control scheme. The dashed box surrounds the trajectory scaling system.

kinematic constraints. In addition, operational space trajectories are supposed to be known. From these trajectories, it is possible to obtain an analogous path $\mathbf{q}_d(t)$ in the joint space by solving an inverse kinematic problem. $\mathbf{q}_d(t)$ is the reference signal for the manipulator controller.

The TSS, shown in Fig. 3.1, is a dynamic system that is inserted between the LTL generator and the path generator: It prevents constraint violations by properly reducing longitudinal speeds, accelerations and jerks. An advantage of this scaling technique is immediately evident: The TSS does not require data from the manipulator controller, which internal structure can even be unknown.

The scheme of the TSS is similar to that one presented in Chapter 2 for problems in the configuration space. It is made of a general purpose, nonlinear scaling filter and a bounds estimator. The first modifies the nominal LTL, that is expressed through function $s_d(t)$, in order to keep its first, second and third time derivatives within given bounds, i.e., the filter output is a signal $s(t)$ that fulfills the following inequalities

$$R^- \leq \dot{s} \leq R^+ \, , \quad S^- \leq \ddot{s} \leq S^+ \, , \quad U^- \leq \dddot{s} \leq U^+ \, , \tag{3.5}$$

where R^+, R^-, S^+, S^-, U^+, and U^- are freely assignable limits. The scaling filter, formed by a chain of three discrete time integrators, has been widely described in Chapter 1. It is worth to summarize its main characteristics: Output signal $s(t)$ tracks $s_d(t)$ if this latter is feasible, otherwise $s(t)$ becomes the best possible feasible approximation of $s_d(t)$. Moreover, $s(t)$ always hangs in minimum time, compatibly with the imposed constraints, any feasible $s_d(t)$. This property can be used, as will be shown in §3.4, to generate minimum-time trajectories.

The second block of the TSS is the bounds estimator, that converts the kinematic constraints, which affect the manipulator, into equivalent limits for the LTL. It is totally different from the analogous implementation that has been presented in Chapter 2, since it must handle constraints that are defined in the operational space. Its computational complexity is necessarily higher, because the required conversions involve the existing relationships between the configuration and the operational spaces. Nevertheless, the proposed formulation, as shown in §3.4, is perfectly compatible with real-time implementation. The bounds estimator handles several different constraints. For example, velocities, accelerations and jerks of the end effector must be kept within some given limits. Thus, by defining the desired generalized velocity of the tool frame as

$$\widetilde{\mathbf{v}}_T := \left[\begin{array}{c} \mathbf{v}_T \\ \hline \boldsymbol{\omega}_T \end{array}\right] = [\widetilde{v}_{T_1}\ \widetilde{v}_{T_2}\ \widetilde{v}_{T_3}\ \widetilde{v}_{T_4}\ \widetilde{v}_{T_5}\ \widetilde{v}_{T_6}]^T \in \mathbb{R}^6 \tag{3.6}$$

where $\mathbf{v}_T$ and $\boldsymbol{\omega}_T$ respectively represent the desired linear and angular velocities in the operational space, by indicating with

$$\widetilde{\mathbf{a}}_T := \left[\begin{array}{c} \mathbf{a}_T \\ \hline \boldsymbol{\alpha}_T \end{array}\right] = [\widetilde{a}_{T_1}\ \widetilde{a}_{T_2}\ \widetilde{a}_{T_3}\ \widetilde{a}_{T_4}\ \widetilde{a}_{T_5}\ \widetilde{a}_{T_6}]^T \in \mathbb{R}^6 \tag{3.7}$$

the desired generalized acceleration, where $\mathbf{a}_T$ and $\boldsymbol{\alpha}_T$ respectively indicate the desired linear and angular accelerations, and by posing with

$$\widetilde{\mathbf{j}}_T := \left[\begin{array}{c} \mathbf{j}_T \\ \hline \boldsymbol{\iota}_T \end{array}\right] = [\widetilde{j}_{T_1}\ \widetilde{j}_{T_2}\ \widetilde{j}_{T_3}\ \widetilde{j}_{T_4}\ \widetilde{j}_{T_5}\ \widetilde{j}_{T_6}]^T \in \mathbb{R}^6 \tag{3.8}$$

the desired generalized jerk, where $\mathbf{j}_T$ and $\boldsymbol{\iota}_T$ respectively indicate the desired linear and angular jerks, following constraints must be simultaneously satisfied ($k = 1,2,\ldots,6$)

$$\underline{v}_k \leq \widetilde{v}_{T_k} \leq \overline{v}_k\ , \tag{3.9}$$

$$\underline{a}_k \leq \widetilde{a}_{T_k} \leq \overline{a}_k\ . \tag{3.10}$$

$$\underline{j}_k \leq \widetilde{j}_{T_k} \leq \overline{j}_k\ . \tag{3.11}$$

Constraint vectors $\underline{\mathbf{v}} := [\underline{v}_1\, \underline{v}_2\, \underline{v}_3\, \underline{v}_4\, \underline{v}_5\, \underline{v}_6]^T \in (\mathbb{R}^-)^6$, $\bar{\mathbf{v}} := [\bar{v}_1\, \bar{v}_2\, \bar{v}_3\, \bar{v}_4\, \bar{v}_5\, \bar{v}_6]^T \in (\mathbb{R}^+)^6$, $\underline{\mathbf{a}} := [\underline{a}_1\ \underline{a}_2\ \underline{a}_3\ \underline{a}_4\ \underline{a}_5\ \underline{a}_6]^T \in (\mathbb{R}^-)^6$, $\bar{\mathbf{a}} := [\bar{a}_1\ \bar{a}_2\ \bar{a}_3\ \bar{a}_4\ \bar{a}_5\ \bar{a}_6]^T \in (\mathbb{R}^+)^6$, $\underline{\mathbf{j}} := [\underline{j}_1\, \underline{j}_2\, \underline{j}_3\, \underline{j}_4\, \underline{j}_5\, \underline{j}_6]^T \in (\mathbb{R}^-)^6$, and $\bar{\mathbf{j}} := [\bar{j}_1\, \bar{j}_2\, \bar{j}_3\, \bar{j}_4\, \bar{j}_5\, \bar{j}_6]^T \in (\mathbb{R}^+)^6$ are assigned by users depending on the desired motion smoothness.

Bounds on joint velocities, accelerations, and jerks, i.e., $\dot{q}_i$, $\ddot{q}_i$, and $\dddot{q}_i$ respectively, must also be taken into account. It has been previously pointed out that trajectories in the operational space could produce joint velocities and accelerations that are beyond the actuators capabilities. In order to prevent path tracking problems the following limits must be fulfilled

$$\underline{\dot{q}}_i \le \dot{q}_i \le \bar{\dot{q}}_i\,, \tag{3.12}$$

$$\underline{\ddot{q}}_i \le \ddot{q}_i \le \bar{\ddot{q}}_i\,, \tag{3.13}$$

$$\underline{\dddot{q}}_i \le \dddot{q}_i \le \bar{\dddot{q}}_i\,, \tag{3.14}$$

where $i = 1, 2, \ldots, N$, while N is equal to the number of independent joints. Bounds $\underline{\dot{\mathbf{q}}} := [\underline{\dot{q}}_1\, \underline{\dot{q}}_2 \ldots \underline{\dot{q}}_N]^T \in (\mathbb{R}^-)^N$ and $\bar{\dot{\mathbf{q}}} := [\bar{\dot{q}}_1\, \bar{\dot{q}}_2 \ldots \bar{\dot{q}}_N]^T \in (\mathbb{R}^+)^N$ typically coincide with the motor speed limits, $\underline{\ddot{\mathbf{q}}} := [\underline{\ddot{q}}_1\, \underline{\ddot{q}}_2 \ldots \underline{\ddot{q}}_N]^T \in (\mathbb{R}^-)^N$ and $\bar{\ddot{\mathbf{q}}} := [\bar{\ddot{q}}_1\, \bar{\ddot{q}}_2 \ldots \bar{\ddot{q}}_N]^T \in (\mathbb{R}^+)^N$ are used to reduce the mechanical stress, while $\underline{\dddot{\mathbf{q}}} := [\underline{\dddot{q}}_1\, \underline{\dddot{q}}_2 \ldots \underline{\dddot{q}}_N]^T \in (\mathbb{R}^-)^N$ and $\bar{\dddot{\mathbf{q}}} := [\bar{\dddot{q}}_1\, \bar{\dddot{q}}_2 \ldots \bar{\dddot{q}}_N]^T \in (\mathbb{R}^+)^N$ are introduced to limit the rate of joint accelerations: Continuous accelerations make it possible to drastically reduce the elicitation of oscillatory modes.

Longitudinal velocities, accelerations, and jerks along the path can be bounded, by imposing

$$\underline{\dot{s}} \le \dot{s} \le \bar{\dot{s}}\,, \tag{3.15}$$

$$\underline{\ddot{s}} \le \ddot{s} \le \bar{\ddot{s}}\,, \tag{3.16}$$

$$\underline{\dddot{s}} \le \dddot{s} \le \bar{\dddot{s}}\,, \tag{3.17}$$

where $\underline{\dot{s}} \in \mathbb{R}^-$, $\bar{\dot{s}} \in \mathbb{R}^+$, $\underline{\ddot{s}} \in \mathbb{R}^-$, $\bar{\ddot{s}} \in \mathbb{R}^+$, $\underline{\dddot{s}} \in \mathbb{R}^-$, and $\bar{\dddot{s}} \in \mathbb{R}^+$ are user-defined limits respectively on longitudinal velocities, accelerations and jerks.

In next §3.2, it will be shown how constraints (3.9)–(3.14) can be converted into equivalent bounds R^-, R^+, S^-, S^+, U^-, and U^+ for the LTL. No adaptation is required for constraints (3.15)–(3.17), that can be immediately applied to (3.5).

3.2 The equivalent bounds evaluation for the LTL

As known, velocities, accelerations, and jerks of the end effector can be evaluated as follows

$$
\begin{aligned}
\widetilde{\mathbf{v}}_T &= \mathbf{J}_T\,\dot{\mathbf{q}}\,, && (3.18)\\
\widetilde{\mathbf{a}}_T &= \dot{\mathbf{J}}_T\,\dot{\mathbf{q}} + \mathbf{J}_T\,\ddot{\mathbf{q}}\,, && (3.19)\\
\widetilde{\mathbf{j}}_T &= \ddot{\mathbf{J}}_T\,\dot{\mathbf{q}} + 2\dot{\mathbf{J}}_T\,\ddot{\mathbf{q}} + \mathbf{J}_T\,\dddot{\mathbf{q}}\,, && (3.20)
\end{aligned}
$$

where $\mathbf{J}_T = \mathbf{J}_T(\mathbf{q})$ is the system geometric Jacobian, while $\dot{\mathbf{J}}_T = \dot{\mathbf{J}}_T(\mathbf{q})$ and $\ddot{\mathbf{J}}_T = \ddot{\mathbf{J}}_T(\mathbf{q})$ are its first and second time derivative, respectively.

If $\mathbf{J}_T$ is not singular, by inverting (3.18)–(3.20) it immediately descends that

$$
\begin{aligned}
\dot{\mathbf{q}} &= \mathbf{J}_T^{-1}\,\widetilde{\mathbf{v}}_T\,, && (3.21)\\
\ddot{\mathbf{q}} &= \mathbf{J}_T^{-1}(\widetilde{\mathbf{a}}_T - \dot{\mathbf{J}}_T\,\dot{\mathbf{q}}) = \mathbf{J}_T^{-1}(\widetilde{\mathbf{a}}_T - \dot{\mathbf{J}}_T\,\mathbf{J}_T^{-1}\,\widetilde{\mathbf{v}})\,, && (3.22)\\
\dddot{\mathbf{q}} &= \mathbf{J}_T^{-1}(\widetilde{\mathbf{j}}_T - \ddot{\mathbf{J}}_T\,\dot{\mathbf{q}} - 2\dot{\mathbf{J}}_T\,\ddot{\mathbf{q}})\\
&= \mathbf{J}_T^{-1}\left[\widetilde{\mathbf{j}}_T - \ddot{\mathbf{J}}_T\,\mathbf{J}_T^{-1}\,\widetilde{\mathbf{v}}_T - 2\dot{\mathbf{J}}_T\,\mathbf{J}_T^{-1}(\widetilde{\mathbf{a}}_T - \dot{\mathbf{J}}_T\,\mathbf{J}_T^{-1}\,\widetilde{\mathbf{v}}_T)\right]. && (3.23)
\end{aligned}
$$

Equations (3.21)–(3.23) are expressed in function of variables of the configuration space and are instrumental for the evaluation of the equivalent longitudinal bounds, provided that they could be posed in function of s.

To this purpose, path $[\mathbf{p}_T(s),\ {}^{0}_{T}\mathbf{R}(s)]$ is first converted into an equivalent path $\mathbf{q}_T(s)$ in the configuration space, which is subsequently used to evaluate

$$\mathbf{J}_T(s) = \mathbf{J}[\mathbf{q}(s)] \qquad \text{and} \qquad \mathbf{J}_T^{-1}(s) = \mathbf{J}^{-1}[\mathbf{q}(s)]\,.$$

Linear velocity $\mathbf{v}_T$ can be obtained from $\mathbf{p}_T(s)$ by means of the chain differentiation rule, i.e.

$$\mathbf{v}_T(s,\dot{s}) = \frac{d\mathbf{p}_T(s)}{ds}\,\frac{ds}{dt} = \frac{d\mathbf{p}_T(s)}{ds}\,\dot{s} := \mathbf{v}_T(s)\,\dot{s}\,. \qquad (3.24)$$

A similar result can also be obtained for $\boldsymbol{\omega}_T := [\omega_x\ \omega_y\ \omega_z]^T$. The differentiation rule of rotational matrices, i.e. ${}^{0}_{T}\dot{\mathbf{R}} = \mathbf{S}(\boldsymbol{\omega}_T)\,{}^{0}_{T}\mathbf{R}$, where $\mathbf{S}(\boldsymbol{\omega}_T)$ is a skew symmetric matrix that is defined as follows

$$\mathbf{S}(\boldsymbol{\omega}_T) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix},$$

makes it possible to write

$$\mathbf{S}[\boldsymbol{\omega}_T(s,\dot{s})] = {}^0_T\dot{\mathbf{R}}\,{}^0_T\mathbf{R}^T = \frac{d\,\left[{}^0_T\mathbf{R}(s)\right]}{ds}\,{}^0_T\mathbf{R}^T(s)\,\dot{s}\,. \tag{3.25}$$

Owing to (3.25), $\boldsymbol{\omega}_T(s,\dot{s})$ can evidently be written as follows

$$\boldsymbol{\omega}_T(s,\dot{s}) = \boldsymbol{\omega}_T(s)\,\dot{s}\,. \tag{3.26}$$

Bearing in mind (3.24) and (3.26), (3.18) can be posed into the following form

$$\widetilde{\mathbf{v}}_T(s,\dot{s}) = \widetilde{\mathbf{v}}_T(s)\,\dot{s}\,, \tag{3.27}$$

so that, by using the chain differentiation rule, its first and second order derivative are given by

$$\widetilde{\mathbf{a}}_T(s,\dot{s}) = \widetilde{\mathbf{v}}'_T(s)\,\dot{s}^2 + \widetilde{\mathbf{v}}_T(s)\,\ddot{s}\,, \tag{3.28}$$

$$\widetilde{\mathbf{j}}_T(s,\dot{s},\ddot{s}) = \widetilde{\mathbf{v}}''_T(s)\,\dot{s}^3 + 3\,\widetilde{\mathbf{v}}'_T(s)\,\dot{s}\,\ddot{s} + \widetilde{\mathbf{v}}_T(s)\,\dddot{s}\,, \tag{3.29}$$

where $\widetilde{\mathbf{v}}'_T(s) := [d\widetilde{\mathbf{v}}_T(s)]/(ds)$ and $\widetilde{\mathbf{v}}''_T(s) := [d^2\widetilde{\mathbf{v}}_T(s)]/(ds^2)$. Hence, equations (3.27)–(3.29) permit to express bounds (3.9)–(3.11) in function of the curvilinear abscissa s.

Relations (3.12)–(3.14) must also be expressed in function of the curvilinear abscissa. By means of (3.27), (3.18) can be written as follows

$$\dot{\mathbf{q}}(s,\dot{s}) = \mathbf{a}(s)\,\dot{s}\,, \tag{3.30}$$

where

$$\mathbf{a}(s) := \mathbf{J}_T^{-1}(s)\,\widetilde{\mathbf{v}}_T(s)\,. \tag{3.31}$$

As shown in [51], $\widetilde{\mathbf{v}}_T(s)$ can be obtained from the knowledge of the path equations, so that, given s, it is certainly possible to evaluate $\mathbf{a}(s)$. Consequently, in the following, $\mathbf{a}(s)$ is supposed to be known.

The critical part of the procedure to express the joints' constraints in function of the curvilinear abscissa is represented by the explicit derivation of the first and the second time derivative of the Jacobian in function of s. An efficient procedure for the evaluation of $\dot{\mathbf{J}}_T$ and $\ddot{\mathbf{J}}_T$ is proposed in §3.5, where it is also shown that $\dot{\mathbf{J}}_T$ and $\ddot{\mathbf{J}}_T$ assume the following structure

$$\begin{aligned}\dot{\mathbf{J}}_T(s,\dot{s}) &= \mathbf{J}'_T(s)\,\dot{s}\,,\\ \ddot{\mathbf{J}}_T(s,\dot{s},\ddot{s}) &= \mathbf{J}''_T(s)\,\dot{s}^2 + \mathbf{J}'_T(s)\,\ddot{s}\,.\end{aligned}$$

Thus, bearing also in mind (3.27)–(3.29), (3.22) and (3.23) can be respectively written as follows

$$\ddot{\mathbf{q}}(s,\dot{s},\ddot{s}) = \mathbf{a}(s)\,\ddot{s} + \mathbf{b}(s)\,\dot{s}^2\,, \tag{3.32}$$

$$\dddot{\mathbf{q}}(s,\dot{s},\ddot{s},\dddot{s}) = \mathbf{a}(s)\,\dddot{s} + 3\,\mathbf{b}(s)\,\dot{s}\,\ddot{s} + \mathbf{c}(s)\,\dot{s}^3\,, \tag{3.33}$$

where $\mathbf{a}(s)$ is given by (3.31), while

$$\mathbf{b}(s) := \mathbf{J}_T^{-1}(s)\,[\widetilde{\mathbf{v}}'_T(s) - \mathbf{J}'_T(s)\,\mathbf{a}(s)]\,, \tag{3.34}$$

$$\mathbf{c}(s) := \mathbf{J}_T^{-1}(s)[\widetilde{\mathbf{v}}''_T(s) - \mathbf{J}''_T(s)\,\mathbf{a}(s) - 2\,\mathbf{J}'_T(s)\,\mathbf{b}(s)]\,. \tag{3.35}$$

Since $\widetilde{\mathbf{v}}'_T(s)$ and $\widetilde{\mathbf{v}}''_T(s)$ can be analytically obtained from the path equations, all terms that are required for the evaluation of $\mathbf{b}(s)$ and $\mathbf{c}(s)$ are known.

Previous results (3.30), (3.32) and (3.33) permit to pose constraints (3.12)–(3.14) in the following manner ($i = 1,2,\ldots,N$)

$$\underline{\dot{q}}_i \leq a_i(s)\,\dot{s} \leq \overline{\dot{q}}_i\,, \tag{3.36}$$

$$\underline{\ddot{q}}_i \leq a_i(s)\,\ddot{s} + b_i(s)\,\dot{s}^2 \leq \overline{\ddot{q}}_i\,, \tag{3.37}$$

$$\underline{\dddot{q}}_i \leq a_i(s)\,\dddot{s} + 3\,b_i(s)\,\dot{s}\,\ddot{s} + c_i(s)\,\dot{s}^3 \leq \overline{\dddot{q}}_i\,. \tag{3.38}$$

Similar to the procedure presented in §2.3, equations (3.36)–(3.38) must evidently be simultaneously satisfied for any $i = 1,2,\ldots,N$, and they can be rearranged in order to convert the original joint bounds into equivalent constraints for the curvilinear coordinate. For example, joint i fulfills (3.36) if $\dot{s} \in \left[\underline{\zeta}_i, \overline{\zeta}_i\right]$ where $\underline{\zeta}_i$ and $\overline{\zeta}_i$ are given

Table 3.1: Equivalent longitudinal bounds.

	$a_i > 0$	$a_i < 0$	$a_i = 0$
$\overline{\sigma}_i$	$(\overline{\dddot{q}}_i - c_i\dot{s}^3 - 3b_i\dot{s}\ddot{s})/a_i$	$(\underline{\dddot{q}}_i - c_i\dot{s}^3 - 3b_i\dot{s}\ddot{s})/a_i$	∞
$\underline{\sigma}_i$	$(\underline{\dddot{q}}_i - c_i\dot{s}^3 - 3b_i\dot{s}\ddot{s})/a_i$	$(\overline{\dddot{q}}_i - c_i\dot{s}^3 - 3b_i\dot{s}\ddot{s})/a_i$	$-\infty$
$\overline{\mu}_i$	$(\overline{\ddot{q}}_i - b_i\dot{s}^2)/a_i$	$(\underline{\ddot{q}}_i - b_i\dot{s}^2)/a_i$	∞
$\underline{\mu}_i$	$(\underline{\ddot{q}}_i - b_i\dot{s}^2/)a_i$	$(\overline{\ddot{q}}_i - b_i\dot{s}^2)/a_i$	$-\infty$
$\overline{\zeta}_i$	$\overline{\dot{q}}_i/a_i$	$\underline{\dot{q}}_i/a_i$	∞
$\underline{\zeta}_i$	$\underline{\dot{q}}_i/a_i$	$\overline{\dot{q}}_i/a_i$	$-\infty$

	$\widetilde{v}_{T_k} > 0$	$\widetilde{v}_{T_k} < 0$	$\widetilde{v}_{T_k} = 0$
$\overline{\gamma}_k$	$(\overline{j}_k - \widetilde{v}''_{T_k}\dot{s}^3 - 3\widetilde{v}'_{T_k}\dot{s}\ddot{s})/\widetilde{v}_{T_k}$	$(\underline{j}_k - \widetilde{v}''_{T_k}\dot{s}^3 - 3\widetilde{v}'_{T_k}\dot{s}\ddot{s})/\widetilde{v}_{T_k}$	∞
$\underline{\gamma}_k$	$(\underline{j}_k - \widetilde{v}''_{T_k}\dot{s}^3 - 3\widetilde{v}'_{T_k}\dot{s}\ddot{s})/\widetilde{v}_{T_k}$	$(\overline{j}_k - \widetilde{v}''_{T_k}\dot{s}^3 - 3\widetilde{v}'_{T_k}\dot{s}\ddot{s})/\widetilde{v}_{T_k}$	$-\infty$
$\overline{\beta}_k$	$(\overline{a}_k - \widetilde{v}'_{T_k}\dot{s}^2)/\widetilde{v}_{T_k}$	$(\underline{a}_k - \widetilde{v}'_{T_k}\dot{s}^2)/\widetilde{v}_{T_k}$	∞
$\underline{\beta}_k$	$(\underline{a}_k - \widetilde{v}'_{T_k}\dot{s}^2/)\widetilde{v}_{T_k}$	$(\overline{a}_k - \widetilde{v}'_{T_k}\dot{s}^2)/\widetilde{v}_{T_k}$	$-\infty$
$\overline{\alpha}_k$	$\overline{v}_k/\widetilde{v}_{T_k}$	$\underline{v}_k/\widetilde{v}_{T_k}$	∞
$\underline{\alpha}_k$	$\underline{v}_k/\widetilde{v}_{T_k}$	$\overline{v}_k/\widetilde{v}_{T_k}$	$-\infty$

in Table 3.1. The same inclusion must simultaneously apply for all joints, so that velocity constraint (3.12) is satisfied only if $\dot{s} \in \bigcap_{i=1}^{N}\left[\underline{\zeta}_i, \overline{\zeta}_i\right]$. Similar considerations apply on the acceleration and on the jerk constraints, thus equations (3.37) and (3.38) are satisfied, and, in turn, also (3.13) and (3.14), only if $\ddot{s} \in \bigcup_{i=1}^{N}[\underline{\mu}_i, \overline{\mu}_i]$ and only if $\dddot{s} \in \bigcup_{i=1}^{N}[\underline{\sigma}_i, \overline{\sigma}_i]$, respectively.

A similar procedure can also be used to transform the limits in the operational space. By virtue of (3.27)–(3.29), bounds (3.9)–(3.11) become ($k = 1, 2, \dots, 6$)

$$\underline{v}_k \leq \widetilde{v}_{T_k}(s)\,\dot{s} \leq \overline{v}_k\ , \tag{3.39}$$

$$\underline{a}_k \leq \widetilde{v}_{T_k}(s)\,\ddot{s} + \widetilde{v}'_{T_k}(s)\,\dot{s}^2 \leq \overline{a}_k\ , \tag{3.40}$$

$$\underline{j}_k \leq \widetilde{v}_{T_k}(s)\,\dddot{s} + 3\,\widetilde{v}'_{T_k}(s)\,\dot{s}\,\ddot{s} + \widetilde{v}''_{T_k}(s)\,\dot{s}^3 \leq \overline{j}_k\ . \tag{3.41}$$

The structure of (3.39)–(3.41) is evidently the same of (3.36)–(3.38), so that it is possible to assert that the trajectory is feasible only if $\dot{s} \in \bigcap_{k=1}^{6}[\underline{\alpha}_k, \overline{\alpha}_k]$, $\ddot{s} \in \bigcap_{k=1}^{6}\left[\underline{\beta}_k, \overline{\beta}_k\right]$ and $\dddot{s} \in \bigcap_{k=1}^{6}\left[\underline{\gamma}_k, \overline{\gamma}_k\right]$ where coefficients $\underline{\alpha}_k$, $\overline{\alpha}_k$, $\underline{\beta}_k$, $\overline{\beta}_k$, $\underline{\gamma}_k$, and $\overline{\gamma}_k$ are defined according to Table 3.1.

Finally, constraints (3.15)–(3.17) are directly defined in the curvilinear-coordinate space, thus they do not require any transformation.

In conclusion, time-law $s(t)$ is feasible if the following conditions are simultaneously fulfilled

$$\dot{s} \in \left[R^-, R^+\right] , \tag{3.42}$$

$$\ddot{s} \in \left[S^-, S^+\right] , \tag{3.43}$$

$$\dddot{s} \in \left[U^-, U^+\right] , \tag{3.44}$$

where ($i = 1, 2, \ldots, N$, $k = 1, 2, \ldots, 6$)

$$R^- := \max_{i,k}\{\underline{\alpha}_k, \underline{\zeta}_i, \underline{\dot{s}}\}, \qquad R^+ := \min_{i,k}\{\overline{\alpha}_k, \overline{\zeta}_i, \overline{\dot{s}}\}, \tag{3.45}$$

$$S^- := \max_{i,k}\{\underline{\beta}_k, \underline{\mu}_i, \underline{\ddot{s}}\}, \qquad S^+ := \min_{i,k}\{\overline{\beta}_k, \overline{\mu}_i, \overline{\ddot{s}}\}, \tag{3.46}$$

$$U^- := \max_{i,k}\{\underline{\gamma}_k, \underline{\sigma}_i, \underline{\dddot{s}}\}, \qquad U^+ := \min_{i,k}\{\overline{\gamma}_k, \overline{\sigma}_i, \overline{\dddot{s}}\}. \tag{3.47}$$

3.3 Feasibility problems

The scaling mechanism proposed in this chapter is subject to the same feasibility issues already pointed out in §2.4 for trajectories in the configuration space. The problem can be summarized as follows: Depending on the system status of motion, interval $[S^-, S^+]$ could become very small. Obviously, when this happens the dynamics of the speed signal is strongly limited, so that it can only be slowly changed. Unfortunately, the velocity bounds, i.e., R^- and R^+, are not constant: If they change too rapidly the feasibility is lost since the available dynamics is not sufficient to permit fast speed changes. A similar problem arises for the acceleration signal when interval $[U^-, U^+]$ tends to vanish. In §2.4 a solution is provided for trajectories in the configuration space: Roughly speaking, if $[S^-, S^+]$ and/or $[U^-, U^+]$ become too small, it is possible to invert such trend by reducing in advance the longitudinal speed. This result is achieved by properly downscaling velocity bound R^+ (more details on the strategy can be found in §2.4). This principle is generally valid, provided that critical situations could be known in advance. Preliminary tests have revealed that the technique adopted in §2.4 for trajectories in the joint space is not suited for those in the operational space. In case of paths passing close to kinematic singularities, indeed, acceleration and jerk feasibility intervals are normally well open along the whole trajectory, then they suddenly collapse to zero when the trajectory passes very close to a singular point: The available time is not sufficient for any reaction.

An alternative strategy, also presented §2.4, for cyclic trajectories could be theoretically reused for the problem here presented. However, the assumption made in §3.1 is that trajectories are not repetitive and that any reaction to possible problems must be taken in real-time. For this reason, alternative approaches have been investigated such to promptly react to undesired behaviors caused by singular point. It is known that an approach to a singularity can be detected by analyzing the singular values of the Jacobian matrix associated to the end effector. However techniques based on the singular value decomposition did not give acceptable results, partially because of their computational burden, but mainly because they do not reveal singularities with a sufficient time margin.

Conversely, an approach based on an analysis in the configuration space, despite its simplicity, has given the most promising results. More precisely, singularities can be easily detected through checks in the joint space. For example, wrist singularities of a 6R anthropomorphic manipulator like the one considered in §3.4, can be predicted by observing the fifth joint variable: A wrist alignment corresponds to a singular configuration and it is pointed out by condition $q_5 \simeq 0$. Consequently, bounds R^- and R^+ can be downscaled as follows

$$\left[\widetilde{R}^-,\, \widetilde{R}^+\right] = \begin{cases} \left[R^-,\, \psi R^+\right], & \text{if } \dot{s} \geq 0 \\ \left[\psi R^-,\, R^+\right], & \text{otherwise} \end{cases}, \tag{3.48}$$

where ψ is given by

$$\psi = \begin{cases} \operatorname{sat}\left[1 + \frac{|\dot{s}|}{v_R}\left(\frac{|q_5|}{\overline{q}_5} - 1\right)\right], & \text{if } |q_5| < \overline{q}_5 \\ 1, & \text{otherwise} \end{cases}. \tag{3.49}$$

Function $\operatorname{sat}(\cdot)$ saturates its output between 0 and 1.

Threshold $\overline{q}_5$ is chosen such to guarantee a reasonably anticipated reaction of the system, while v_R coincides with the maximum longitudinal speed that can be admitted for the considered application: More incisive reactions are provided when $\dot{s}$ approaches v_R. The new bounds, i.e., $\widetilde{R}^-$ and $\widetilde{R}^+$, replace R^- and R^+, so that (3.42) is replaced by

$$\dot{s} \in \left[\widetilde{R}^-, \widetilde{R}^+\right] .$$

Similar detection methods can be used to manage other singular configurations of the manipulator.

It is worth to mention that the bound-scaling system proposed in §2.4 should still be used to manage possible problems which could occur even in uncritical zones because of, for example, reference signals characterized by too demanding longitudinal speeds.

3.4 Experimental results

The experimental tests have been executed on a real Comau Smart-Six 6-1.4 manipulator and have been fitted to compare the tracking performances of the TSS presented in the above sections, i.e. the Jerk Constrained TSS (JC-TSS) with the performances of its previous release proposed in [50], i.e. the Acceleration Constrained TSS (AC-TSS). It is important to remark the main difference between the two systems: In the JC-TSS is introduced the management of jerk constraints. The performances of these systems have been successively compared with those of the commercial trajectory generator, as shown in the last part of this section.

The trajectory planner of the commercial controller has been replaced with a novel reference generator equipped with one of the two TSSs. Conversely, joint motors are still driven with the control loops of the commercial controller. The trajectory updating time is equal to $2 \cdot 10^{-3}$ s.

The first test sets have been concerned the execution of a horizontal linear trajectory from $\mathbf{p}_A = [0.25\ 0.83\ 1.095]^T$ to $\mathbf{p}_B = [-0.25\ 0.83\ 1.095]^T$, which passes close to a wrist singularity located at $\overline{\mathbf{p}}_T = [0\ 0.83\ 1.077]^T$ and that, consequently, it could

Table 3.2: Kinematic limits that have been used for the experiments. Jerk bounds only apply to the JC-TSS.

$\underline{\dot{\mathbf{q}}}$	$[-10\ -10\ -10\ -7.5\ -10\ -9]^T$
$\overline{\dot{\mathbf{q}}}$	$[10\ 10\ 10\ 7.5\ 10\ 9]^T$
$\underline{\ddot{\mathbf{q}}}$	$[-100\ -100\ -100\ -100\ -100\ -100]^T$
$\overline{\ddot{\mathbf{q}}}$	$[100\ 100\ 100\ 100\ 100\ 100]^T$
$\underline{\dddot{\mathbf{q}}}$	$[-4000\ -4000\ -4000\ -4000\ -4000\ -4000]^T$
$\overline{\dddot{\mathbf{q}}}$	$[4000\ 4000\ 4000\ 4000\ 4000\ 4000]^T$

$\underline{\mathbf{v}}_T$	$[-0.4\ -0.4\ -0.4\ -10\ -10\ -10]^T$
$\overline{\mathbf{v}}_T$	$[0.4\ 0.4\ 0.4\ 10\ 10\ 10]^T$
$\underline{\mathbf{a}}_T$	$[-5\ -5\ -5\ -100\ -100\ -100]^T$
$\overline{\mathbf{a}}_T$	$[5\ 5\ 5\ 100\ 100\ 100]^T$
$\underline{\mathbf{j}}_T$	$[-400\ -400\ -400\ -1000\ -1000\ -1000]^T$
$\overline{\mathbf{j}}_T$	$[400\ 400\ 400\ 1000\ 1000\ 1000]^T$

require unfeasible velocities for joints 4 and 6. The tool frame orientation is kept constant and, by assuming the RPY notation, it is posed equal to $\mathbf{\Phi}_{T,0} = [\pi/2\ 0\ \pi/2]^T$ rad. The nominal LTL is given by a step signal defined as follows

$$s_d(t) := \begin{cases} 0 \text{ m} & t = 0 \\ s_f \text{ m} & t > 0 \end{cases},$$

$\dot{s}_d(t) = 0$ m s^{-1}, and $\ddot{s}_d(t) = 0$ m s^{-2}, where $s_f = 0.5$ m is equal to the path length. The TSSs react to the nominal LTL by generating smooth and minimum-time profiles which are compatible with user-defined limits. The upper and lower bounds, defined in (3.9)–(3.14), are computed by assuming the values defined in Table 3.2, while the limits for the constraints along the path, defined in (3.15)–(3.17), are evaluated by imposing $\underline{\dot{s}} = -0.4$ m s^{-1}, $\overline{\dot{s}} = 0.4$ m s^{-1}, $\underline{\ddot{s}} = -15$ m s^{-2}, $\overline{\ddot{s}} = 15$ m s^{-2}, $\underline{\dddot{s}} = -1000$ m s^{-3}, $\overline{\dddot{s}} = 1000$ m s^{-3}. For the AC-TSS, obviously, jerk constraints have not been applied.

As explained in §3.2, the assigned bounds in the joint and Cartesian spaces are online converted into equivalent limits for the LTL through (3.42)–(3.44) and Table 3.1.

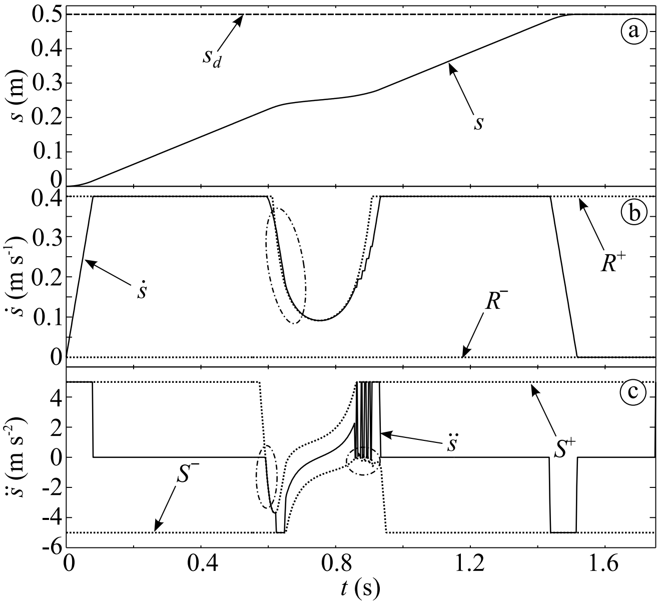

Figure 3.2: Outputs of the AC-TSS compared with the equivalent longitudinal bounds: (a) Longitudinal time-law $s(t)$; (b) Longitudinal velocity $\dot{s}(t)$; (c) Longitudinal acceleration $\ddot{s}(t)$. Dash-dotted lines point out minor constraints violations.

The R^- bound has been imposed equal to 0 in order to avoid backward movements. The output of the AC-TSS is shown in Fig. 3.2. Since transients are minimum-time, the acceleration suddenly assumes its maximum value and the velocity increases until its upper limits is reached. During this phase the motion is essentially limited by the constraints in the operational space. Fig. 3.3 shows the same transient achieved with the JC-TSS. The motion is still minimum-time and the cruising speed is reached with a bang-zero-bang jerk signal. Transient time is evidently longer due to the imposed continuity on the acceleration.

In the vicinity of the singular point, bounds on joint velocities and accelerations become dominant and the longitudinal speed is reduced in order to maintain the feasibility. Figs. 3.2 and 3.3 make it possible to appreciate the different behaviors of the two scaling systems. In particular, while the AC-TSS shows an evident chattering on the acceleration signal (highlighted with dash dotted circles in Fig. 3.2c), the JC-TSS admits much smoother transients. Fig. 3.3 is also useful to understand the importance

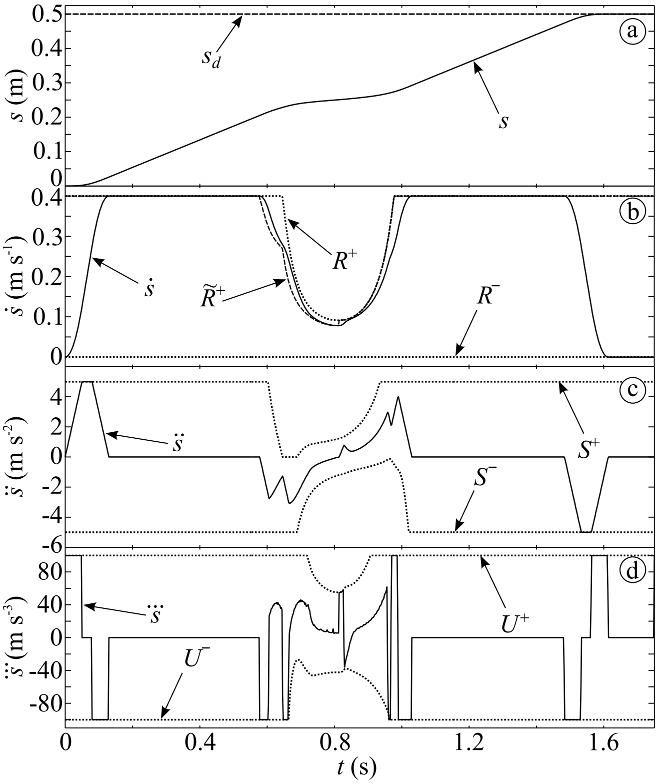

Figure 3.3: Outputs of the JC-TSS compared with the equivalent longitudinal bounds: (a) Longitudinal time-law $s(t)$; (b) Longitudinal velocity $\dot{s}(t)$; (c) Longitudinal acceleration $\ddot{s}(t)$; (d) Longitudinal jerk $\dddot{s}(t)$.

of the downscaling action on R^+, obtained by means of (3.48) and (3.49) and by assuming $v_R = 0.4$ m s^{-1} and $\overline{q}_5 = 0.1$ rad. The JC-TSS, indeed, owing to the continuity on the acceleration signal, normally reacts less promptly than the AC-TSS to sudden changes of the velocity limits and, consequently, constraints' violations could occur more easily. Equation (3.48) anticipates critical situations by reducing in advance upper bound R^+. It is worth to mention that also the AC-TSS could benefit from the use of the same downscaling strategy since, as shown in Fig. 3.2b, it can be subject as well to minor constraint violations. The fulfillment of the equivalent constraints implies, it turn, that (3.9)–(3.14) are satisfied. Fig. 3.4 compares the resulting joint velocities, accelerations, and jerks for joints 4 and 6, i.e. the most solicited joints, with the corre-

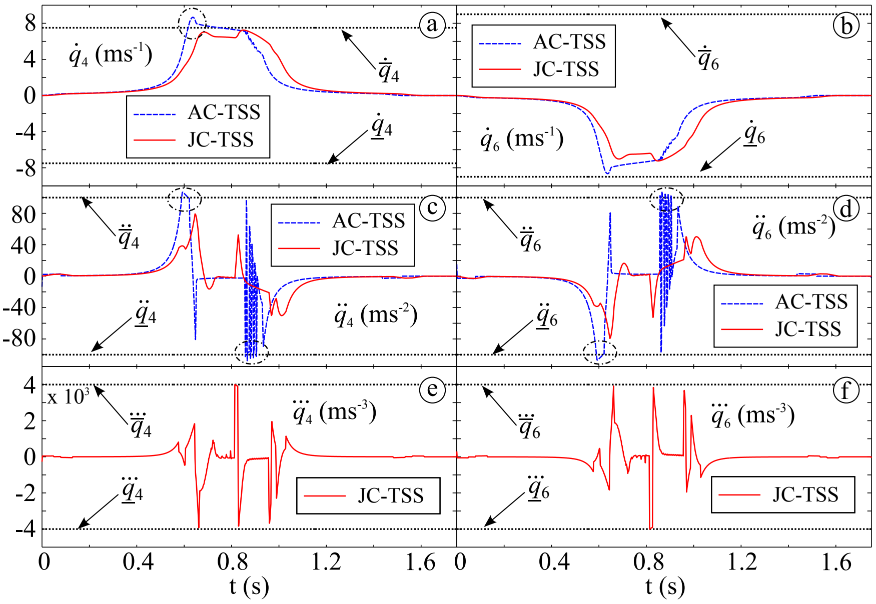

Figure 3.4: Joint reference signals generated by the AC-TSS and by the JC-TSS for joints 4 and 6: (a) and (b) Joint velocities; (c) and (d) Joint accelerations; (e) and (f) Joint jerks.

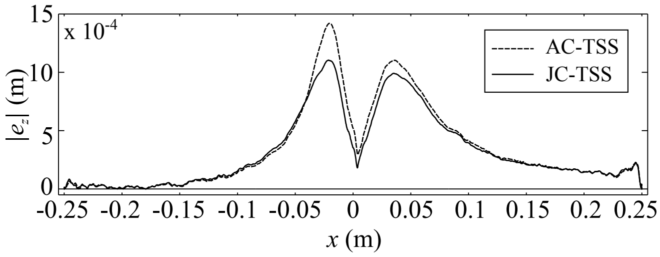

Figure 3.5: Trajectory tracking errors along the x axis.

sponding user defined bounds. Both scaling systems generate references which fulfill the assigned constraints, even if minor violations can be detected for the AC-TSS: They are caused by the above mentioned violations of the equivalent bounds. Moreover, the AC-TSS produces an evident chattering on the acceleration signal which can be potentially troublesome depending on the characteristics of the manipulator controller.

Robust comparisons between the two TSSs can be achieved by considering the

Table 3.3: Statistics on trajectory tracking errors.

	AC-TSS	JC-TSS
$e_{z,max}$ [m]	$1.44 \cdot 10^{-3}$	$1.10 \cdot 10^{-3}$
$E\left[\,\lvert e_z \rvert\,\right]$ [m]	$3.20 \cdot 10^{-4}$	$3.01 \cdot 10^{-4}$
var(e_z) [m^2]	$13.14 \cdot 10^{-8}$	$9.61 \cdot 10^{-8}$
$e_{z,rms}$ [m]	$4.84 \cdot 10^{-4}$	$4.32 \cdot 10^{-4}$
$\int_T \lvert e_z \rvert$ [m]	$2.86 \cdot 10^{-1}$	$2.69 \cdot 10^{-1}$

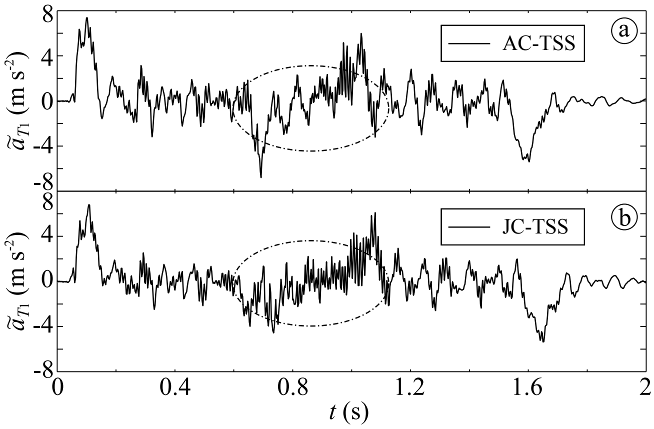

Figure 3.6: Accelerations measured along the x axis. During the trajectory scaling phase, the AC-TSS excites self-oscillations at a frequency close to 14 Hz (see the area surrounded by the dash-dotted lines).

trajectory tracking errors in the operational space. To this purpose, the Cartesian reference signal has been compared with the end-effector position derived from the joint encoders through the direct kinematic procedure. Fig. 3.5 shows the tracking errors for the z axis measured along the direction of motion, i.e. the x axis, while tracking errors along the x and the y axes have not been reported being negligible. The two TSSs have similar performances at the beginning and at the end of each trajectory, while differences become significant in the vicinity of the singularity: The smoother accelerations, which characterize the JC-TSS transients, guarantee better control loops performances and, in turn, tracking errors of reduced amplitude. The numerical comparisons proposed in Table 3.3 further confirm that smaller tracking errors can be achieved by means of the JC-TSS.

In addition, the solicitations acting on the payload have been verified by means of a further experiment. The end-effector of the manipulator has been equipped with an accelerometer in order to measure the structural vibrations. Again, solicitations along the y and the z axes are negligible, so that they have not been reported. From Fig. 3.6 it is possible to evince that differences between the two TSSs are generally minimal, but they become evident close to the singularity. In particular, during the trajectory scaling phase the AC-TSS, differently from the JC-TSS, excites oscillations at a frequency which is close to 14 Hz.

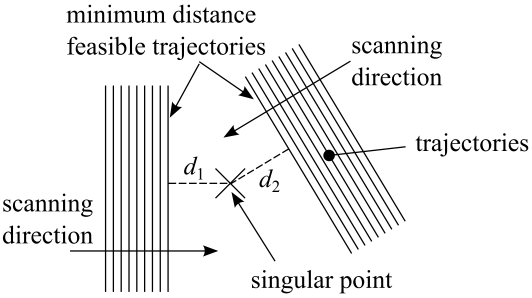

Figure 3.7: Scanning procedure: The singular point is approached from several directions by means of parallel trajectories. For each direction, minimum distance d_i corresponding to the last feasible segment is stored.

The last set of experiments has been conceived to verify the minimum distance from the singular point that can be reached with a feasible trajectory. Several approaching directions have been considered, all of them lying in the xz-plane with $y = 0.83$ m. The scanning procedure is schematically reported in Fig. 3.7. A set of parallel straight lines have been planned with the same longitudinal velocity and with Cartesian distance between two of them equal to $1 \cdot 10^{-3}$ m. From each direction, the scanning procedure has been stopped when the feasibility has been lost and it has been stored the minimum distance from $\overline{\mathbf{p}}_T$. The experiment has been repeated by adopting the planner of commercial controller, the AC-TSS, and the JC-TSS. Not only, all tests have also been repeated for different longitudinal speeds. Fig. 3.8, that collect the stored information, summarizes the achieved results. The use of any of the

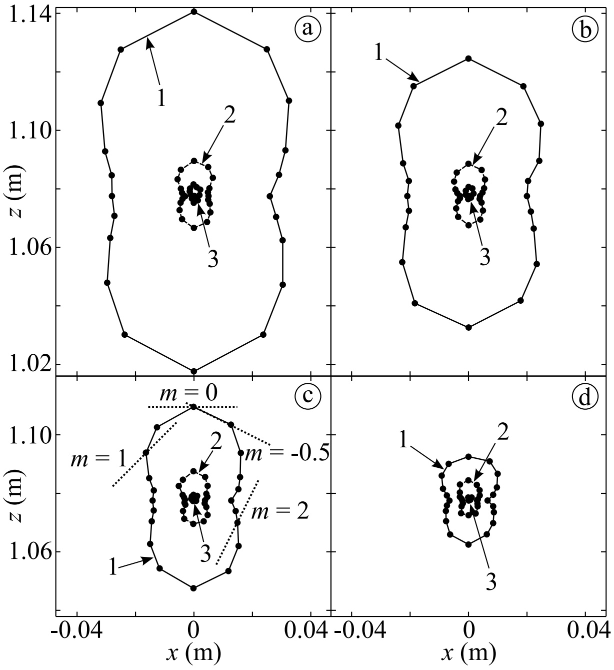

Figure 3.8: Minimum distance points from singularity achieved by executing linear segments and for different slopes *m*. Points have been obtained (1) without any TSS, (2) with the JC-TSS, and (3) with the AC-TSS. Experiments refer to different longitudinal speeds: (a) 0.4 m s^{-1}; (b) 0.3 m s^{-1}; (c) 0.2 m s^{-1}; (d) 0.1 m s^{-1}.

two TSSs evidently permits closer approaches to the singularity. Improvements are especially evident at the highest speeds. Notice that the AC-TSS, owing to the discontinuity of its acceleration signal, reacts more promptly to critical configurations, so that feasibility is even preserved in areas that cannot be reached with the aid of the JC-TSS but, as depicted in Figs. 3.5 and 3.6, it solicits oscillatory modes that smoother trajectories neglect.

The last aspect, that has been considered for the compared evaluation of both TSSs, has been concerned in the estimation of planners' computational burdens that have been measured on a RTAI [30] based PC equipped with an Intel Core2 Duo E8400 @ 3.00 GHz processor. The execution time of the AC-TSS is in the range of between $7 \cdot 10^{-6}$ s and $142 \cdot 10^{-6}$ s with an average computational time equal to

$35\cdot 10^{-6}$ s. Conversely, the execution time of the JC-TSS is in the range of between $11\cdot 10^{-6}$ s and $153\cdot 10^{-6}$ s with an average computational time equal to $49\cdot 10^{-6}$ s. The JC-TSS, since it also constrains the jerk signal, is characterized by greater computational burdens. Despite this drawback, the JC-TSS, and even more so the AC-TSS, is totally compatible with real-time implementation.

3.5 Efficient evaluation of $\mathbf{J}'_T(s)$ and $\mathbf{J}''_T(s,\dot{s})$

Some preliminary considerations are instrumental for the synthesis of the Jacobian derivatives. For a manipulator whose frames are assigned according to the modified Denavit-Hartenberg procedure, the Jacobian matrix associated to tool frame T – which is located on the Nth link and, consequently, is rigidly connected to frame N – has the following structure

$$\mathbf{J}_T = \left[\begin{array}{c} \mathbf{J}_{v_T} \\ \hline \mathbf{J}_{\omega_T} \end{array}\right] = \left[\begin{array}{c|c|c|c} \mathbf{j}_T^{v_1} & \mathbf{j}_T^{v_2} & \cdots & \mathbf{j}_T^{v_N} \\ \hline \mathbf{j}_T^{\omega_1} & \mathbf{j}_T^{\omega_2} & \cdots & \mathbf{j}_T^{\omega_N} \end{array}\right],$$

where $(k = 1,2,\ldots,N)$

$$\mathbf{j}_T^{v_k} = \begin{cases} \hat{\mathbf{z}}_k & \text{prismatic joint} \\ \hat{\mathbf{z}}_k \times [\mathbf{p}_T - \mathbf{p}_k] & \text{revolute joint} \end{cases}, \tag{3.50}$$

$$\mathbf{j}_T^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \hat{\mathbf{z}}_k & \text{revolute joint} \end{cases}, \tag{3.51}$$

and where $\mathbf{p}_T$ and $\mathbf{p}_k$ are respectively the positions of the tool frame and of the kth frame w.r.t. frame 0. $\hat{\mathbf{z}}_k$ is the $\hat{\mathbf{z}}$ unit vector of frame k, described w.r.t. frame 0 and can be evaluated as follows: $\hat{\mathbf{z}}_k = {}^0_k\mathbf{R}\hat{\mathbf{z}}^*$, where $\hat{\mathbf{z}}^* = [0\ 0\ 1]^T$.

Analogously, the jacobian matrix of the generic i-th frame has the following structure

$$\mathbf{J}_i := \left[\begin{array}{c} \mathbf{J}_{v_i} \\ \hline \mathbf{J}_{\omega_i} \end{array}\right] = \left[\begin{array}{c|c|c|c|c|c|c} \mathbf{j}_i^{v_1} & \mathbf{j}_i^{v_2} & \cdots & \mathbf{j}_i^{v_i} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \mathbf{j}_i^{\omega_1} & \mathbf{j}_i^{\omega_2} & \cdots & \mathbf{j}_i^{\omega_i} & \mathbf{0} & \cdots & \mathbf{0} \end{array}\right],$$

where $(k = 1,2,\ldots,i)$

$$\mathbf{j}_i^{v_k} = \begin{cases} \hat{\mathbf{z}}_k & \text{prismatic joint} \\ \hat{\mathbf{z}}_k \times [\mathbf{p}_i - \mathbf{p}_k] & \text{revolute joint} \end{cases}, \tag{3.52}$$

$$\mathbf{j}_i^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \hat{\mathbf{z}}_k & \text{revolute joint} \end{cases} . \tag{3.53}$$

Evidently, by comparing (3.51) with (3.53) it is immediately possible to conclude that, for any element $\mathbf{j}_i^{\omega_k}$ which is different from zero, it holds $\mathbf{j}_i^{\omega_k} = \mathbf{j}_T^{\omega_k}$. Conversely, the equality $\mathbf{j}_i^{v_k} = \mathbf{j}_T^{v_k}$, still considering elements $\mathbf{j}_i^{v_k}$ which are different from zero, only holds, according to (3.50) and (3.52), for prismatic joints.

Similar expressions can also be devised for the evaluation of the Jacobian derivatives. For example, $\dot{\mathbf{J}}_T$ has the following structure

$$\dot{\mathbf{J}}_T := \left[\begin{array}{c} \dot{\mathbf{J}}_{v_T} \\ \hline \dot{\mathbf{J}}_{\omega_T} \end{array} \right] = \left[\begin{array}{c|c|c|c} \dot{\mathbf{j}}_T^{v_1} & \dot{\mathbf{j}}_T^{v_2} & \cdots & \dot{\mathbf{j}}_T^{v_N} \\ \hline \dot{\mathbf{j}}_T^{\omega_1} & \dot{\mathbf{j}}_T^{\omega_2} & \cdots & \dot{\mathbf{j}}_T^{\omega_N} \end{array} \right] ,$$

where $(k = 1, 2, \ldots, N)$

$$\dot{\mathbf{j}}_T^{v_k} = \begin{cases} \dot{\hat{\mathbf{z}}}_k & \text{prismatic joint} \\ \dot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_T - \mathbf{p}_k] + \hat{\mathbf{z}}_k \times [\mathbf{v}_T - \mathbf{v}_k] & \text{revolute joint} \end{cases} , \tag{3.54}$$

$$\dot{\mathbf{j}}_T^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \dot{\hat{\mathbf{z}}}_k & \text{revolute joint} \end{cases} , \tag{3.55}$$

while $\dot{\hat{\mathbf{z}}}_k$ can be evaluated in the following manner

$$\dot{\hat{\mathbf{z}}}_k = \frac{d}{dt} \left({}^0_k\mathbf{R} \, \hat{\mathbf{z}}^* \right) = \frac{d}{dt} \left({}^0_k\mathbf{R} \right) \hat{\mathbf{z}}^* = \mathbf{S}(\boldsymbol{\omega}_k) \, {}^0_k\mathbf{R} \, \hat{\mathbf{z}}^* = \boldsymbol{\omega}_k \times \hat{\mathbf{z}}_k . \tag{3.56}$$

The analogous matrix $\dot{\mathbf{J}}_i$, for the generic i-th frame, admits the following, similar representation

$$\dot{\mathbf{J}}_i := \left[\begin{array}{c} \dot{\mathbf{J}}_{v_i} \\ \hline \dot{\mathbf{J}}_{\omega_i} \end{array} \right] = \left[\begin{array}{c|c|c|c|c|c|c} \dot{\mathbf{j}}_i^{v_1} & \dot{\mathbf{j}}_i^{v_2} & \cdots & \dot{\mathbf{j}}_i^{v_i} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \dot{\mathbf{j}}_i^{\omega_1} & \dot{\mathbf{j}}_i^{\omega_2} & \cdots & \dot{\mathbf{j}}_i^{\omega_i} & \mathbf{0} & \cdots & \mathbf{0} \end{array} \right] ,$$

where $(k = 1, 2, \ldots, i)$

$$\dot{\mathbf{j}}_i^{v_k} = \begin{cases} \dot{\hat{\mathbf{z}}}_k & \text{prismatic joint} \\ \dot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_i - \mathbf{p}_k] + \hat{\mathbf{z}}_k \times [\mathbf{v}_i - \mathbf{v}_k] & \text{revolute joint} \end{cases} ,$$

$$\dot{\mathbf{j}}_i^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \dot{\hat{\mathbf{z}}}_k & \text{revolute joint} \end{cases} .$$

The considerations, that were made for the Jacobian matrix, also apply to its first derivative, so that it is always possible to assume $\dot{\mathbf{j}}_i^{\omega_k} = \dot{\mathbf{j}}_T^{\omega_k}$, while $\dot{\mathbf{j}}_i^{v_k} = \dot{\mathbf{j}}_T^{v_k}$ is only true for prismatic joints.

Finally, $\ddot{\mathbf{J}}_T$ can be evaluated, by extending the abovementioned procedure, as follows

$$\ddot{\mathbf{J}}_T := \left[\begin{array}{c} \ddot{\mathbf{J}}_{v_T} \\ \hline \ddot{\mathbf{J}}_{\omega_T} \end{array}\right] = \left[\begin{array}{c|c|c|c} \ddot{\mathbf{j}}_T^{v_1} & \ddot{\mathbf{j}}_T^{v_2} & \cdots & \ddot{\mathbf{j}}_T^{v_N} \\ \hline \ddot{\mathbf{j}}_T^{\omega_1} & \ddot{\mathbf{j}}_T^{\omega_2} & \cdots & \ddot{\mathbf{j}}_T^{\omega_N} \end{array}\right] ,$$

where $(k = 1, 2, \ldots, N)$

$$\ddot{\mathbf{j}}_T^{v_k} = \begin{cases} \ddot{\hat{\mathbf{z}}}_k & \text{prismatic joint} \\ \ddot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_T - \mathbf{p}_k] + 2\dot{\hat{\mathbf{z}}}_k \times [\mathbf{v}_T - \mathbf{v}_k] + \hat{\mathbf{z}}_k \times [\mathbf{a}_T - \mathbf{a}_k] & \text{revolute joint} \end{cases} ,$$

$$\ddot{\mathbf{j}}_T^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \ddot{\hat{\mathbf{z}}}_k & \text{revolute joint} \end{cases} ,$$

and, analogously, $\ddot{\mathbf{J}}_i$ has the following representation

$$\ddot{\mathbf{J}}_i := \left[\begin{array}{c} \ddot{\mathbf{J}}_{v_i} \\ \hline \ddot{\mathbf{J}}_{\omega_i} \end{array}\right] = \left[\begin{array}{c|c|c|c|c|c|c} \ddot{\mathbf{j}}_i^{v_1} & \ddot{\mathbf{j}}_i^{v_2} & \cdots & \ddot{\mathbf{j}}_i^{v_i} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \ddot{\mathbf{j}}_i^{\omega_1} & \ddot{\mathbf{j}}_i^{\omega_2} & \cdots & \ddot{\mathbf{j}}_i^{\omega_i} & \mathbf{0} & \cdots & \mathbf{0} \end{array}\right] ,$$

where $(k = 1, 2, \ldots, i)$

$$\ddot{\mathbf{j}}_i^{v_k} = \begin{cases} \dot{\hat{\mathbf{z}}}_k & \text{prismatic joint} \\ \ddot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_i - \mathbf{p}_k] + 2\dot{\hat{\mathbf{z}}}_k \times [\mathbf{v}_i - \mathbf{v}_k] + \hat{\mathbf{z}}_k \times [\mathbf{a}_i - \mathbf{a}_k] & \text{revolute joint} \end{cases} , \tag{3.57}$$

$$\ddot{\mathbf{j}}_i^{\omega_k} = \begin{cases} \mathbf{0} & \text{prismatic joint} \\ \ddot{\hat{\mathbf{z}}}_k & \text{revolute joint} \end{cases} . \tag{3.58}$$

The term $\ddot{\hat{\mathbf{z}}}_k$ can be evaluated, by using the chain differentiation rule, from (3.56), in the following manner

$$\ddot{\hat{\mathbf{z}}}_k = \boldsymbol{\alpha}_k \times \hat{\mathbf{z}}_k + \boldsymbol{\omega}_k \times \dot{\hat{\mathbf{z}}}_k = \boldsymbol{\alpha}_k \times \hat{\mathbf{z}}_k + \boldsymbol{\omega}_k \times (\boldsymbol{\omega}_k \times \hat{\mathbf{z}}_k) . \tag{3.59}$$

Again, $\ddot{\mathbf{j}}_i^{\omega_k} = \ddot{\mathbf{j}}_T^{\omega_k}$, while $\ddot{\mathbf{j}}_i^{v_k} = \ddot{\mathbf{j}}_T^{v_k}$ only applies for prismatic joints.

The use of the path-velocity decomposition approach requires alternative techniques for the evaluation of the Jacobian matrix and of its first two time derivatives: Those matrices must to be defined in function of the curvilinear abscissa s. Since those matrices must be often evaluated in real-time, the devised approach must be efficient. An efficient solution for the evaluation of $\mathbf{J}_T(s)$ and $\dot{\mathbf{J}}_T(s,\dot{s})$ has been original proposed in [50], so that it is briefly summarized in the following. After, the procedure is extended for the evaluation of the second order derivative of the Jacobian matrix, i.e. $\ddot{\mathbf{J}}_T(s,\dot{s},\ddot{s})$.

The terms of $\mathbf{J}_T(s)$ can be immediately derived from (3.50) and (3.51), so that for prismatic joints it is possible to assume

$$\mathbf{j}_T^{\omega_k}(s) := \mathbf{0}\,, \tag{3.60}$$

$$\mathbf{j}_T^{v_k}(s) := \hat{\mathbf{z}}_k(s)\,, \tag{3.61}$$

while, for revolute joints, it holds

$$\mathbf{j}_T^{\omega_k}(s) := \hat{\mathbf{z}}_k(s)\,, \tag{3.62}$$

$$\mathbf{j}_T^{v_k}(s) := \mathbf{j}_T^{\omega_k}(s) \times \Delta_k(s)\,, \tag{3.63}$$

where $\hat{\mathbf{z}}_k(s) := {}^0_k\mathbf{R}(s)\,\hat{\mathbf{z}}^*$ and

$$\Delta_k(s) := \mathbf{p}_T(s) - \mathbf{p}_k(s)\,. \tag{3.64}$$

The synthesis of the terms of $\dot{\mathbf{J}}_T(s,\dot{s})$ has required some more steps (details are omitted for conciseness and, here, it is reported its main results). It has been shown in [50] that for revolute joints the following equation apply

$$\dot{\hat{\mathbf{z}}}_k = \dot{s}\,\mathbf{j}'^{\,\omega_k}_T(s) = \dot{s}[\mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}_T^{\omega_k}(s)]\,, \tag{3.65}$$

while for prismatic joints it is possible to assert

$$\dot{\hat{\mathbf{z}}}_k = \dot{s}\,\mathbf{j}'^{\,v_k}_T(s) = \dot{s}[\mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}_T^{v_k}(s)]\,. \tag{3.66}$$

The equations of $\dot{\mathbf{j}}_T^{\omega_k}(s,\dot{s})$ and $\dot{\mathbf{j}}_T^{v_k}(s,\dot{s})$ are obtained from (3.54) and (3.55) by considering substitutions (3.30), (3.31), and (3.64)–(3.66). They can be represented as follows

$$\dot{\mathbf{j}}_T^{\omega_k}(s,\dot{s}) := \dot{s}\,\mathbf{j}'^{\,\omega_k}_T(s)\,, \tag{3.67}$$

$$\dot{\mathbf{j}}_T^{v_k}(s,\dot{s}) := \dot{s}\,\mathbf{j}'^{\,v_k}_T(s)\,, \tag{3.68}$$

where, for prismatic joints,

$$\mathbf{j}'^{\,\omega_k}_T(s) := \mathbf{0}\,, \tag{3.69}$$

$$\mathbf{j}'^{\,v_k}_T(s) := \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s)\times\mathbf{j}_T^{v_k}(s)\,, \tag{3.70}$$

while, for revolute joints,

$$\mathbf{j}'^{\,\omega_k}_T(s) := \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s)\times\mathbf{j}_T^{\omega_k}(s)\,, \tag{3.71}$$

$$\mathbf{j}'^{\,v_k}_T(s) := \mathbf{j}'^{\,\omega_k}_T(s)\times\Delta_k(s)+\mathbf{j}_T^{\omega_k}(s)\times[\mathbf{J}_{v_T}(s)-\mathbf{J}_{v_k}(s)]\mathbf{a}(s)\,. \tag{3.72}$$

It is important to mention that, because of the structure of (3.67) and (3.68), the derivative of the Jacobian matrix can always be posed in the following form

$$\dot{\mathbf{J}}_T(s,\dot{s}) = \mathbf{J}'_T(s)\,\dot{s}\,, \tag{3.73}$$

with

$$\mathbf{J}'_T(s) := \left[\begin{array}{c} \mathbf{J}'_{v_T}(s) \\ \hline \mathbf{J}'_{\omega_T}(s) \end{array}\right] = \left[\begin{array}{c|c|c|c} \mathbf{j}'^{\,v_1}_T(s) & \mathbf{j}'^{\,v_2}_T(s) & \cdots & \mathbf{j}'^{\,v_N}_T(s) \\ \hline \mathbf{j}'^{\,\omega_1}_T(s) & \mathbf{j}'^{\,\omega_2}_T(s) & \cdots & \mathbf{j}'^{\,\omega_N}_T(s) \end{array}\right].$$

The same property evidently applies for the Jacobians of the intermediate frames, so that it is licit to assume for any generic frame *i* that

$$\dot{\mathbf{J}}_{v_i}(s,\dot{s}) = \mathbf{J}'_{v_i}(s)\,\dot{s}\,, \tag{3.74}$$

$$\dot{\mathbf{J}}_{\omega_i}(s,\dot{s}) = \mathbf{J}'_{\omega_i}(s)\,\dot{s}\,. \tag{3.75}$$

The trajectory scaling method proposed in §3.1 requires also the knowledge of $\ddot{\mathbf{J}}_T(s,\dot{s},\ddot{s})$. The starting point is represented by (3.57) and (3.58). According to (3.58), for prismatic joint $\ddot{\mathbf{j}}_T^{\omega_k}=\mathbf{0}$, while for revolute joints $\ddot{\mathbf{j}}_T^{\omega_k}=\ddot{\mathbf{z}}_k$.

From (3.18) it is possible to evince that

$$\boldsymbol{\omega}_k = \mathbf{J}_{\omega_k}\,\dot{\mathbf{q}}\,. \tag{3.76}$$

Its differentiation leads to the following expression

$$\boldsymbol{\alpha}_k = \dot{\mathbf{J}}_{\omega_k}\,\dot{\mathbf{q}} + \mathbf{J}_{\omega_k}\,\ddot{\mathbf{q}}\,. \tag{3.77}$$

By substituting (3.76) and (3.77) into (3.59) it is possible to obtain

$$\ddot{\hat{\mathbf{z}}}_k = (\dot{\mathbf{J}}_{\omega_k}\,\dot{\mathbf{q}} + \mathbf{J}_{\omega_k}\,\ddot{\mathbf{q}}) \times \hat{\mathbf{z}}_k + \mathbf{J}_{\omega_k}\,\dot{\mathbf{q}} \times (\mathbf{J}_{\omega_k}\,\dot{\mathbf{q}} \times \hat{\mathbf{z}}_k)\,. \tag{3.78}$$

After the analysis of the terms in (3.78), some of them have already been expressed in function of s, $\dot{s}$ and $\ddot{s}$: $\ddot{\mathbf{q}}(s,\dot{s},\ddot{s})$ is obtained in (3.32), $\dot{\mathbf{J}}_{\omega_k}(s,\dot{s})$ is given by (3.75), while $\mathbf{J}_{\omega_k}(s)$ is a matrix composed by terms $\hat{\mathbf{z}}_k(s)$ that, in turn, are given by (3.62). The abovementioned substitutions make it possible to rewrite (3.78) as follows

$$\begin{aligned}\ddot{\hat{\mathbf{z}}}_k(s,\dot{s},\ddot{s}) = \Big\{&[\mathbf{J}'_{\omega_k}(s)\,\mathbf{a}(s) + \mathbf{J}_{\omega_k}(s)\,\mathbf{b}(s)] \times \mathbf{j}_T^{\omega_k}(s)\\ &+ \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times [\mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}_T^{\omega_k}(s)]\Big\}\,\dot{s}^2 + \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}_T^{\omega_k}(s)\,\ddot{s}\,,\end{aligned}$$

and, consequently, for revolute joints the generic term $\ddot{\mathbf{j}}_T^{\omega_k}$ is given by

$$\ddot{\mathbf{j}}_T^{\omega_k}(s,\dot{s},\ddot{s}) := \ddot{\hat{\mathbf{z}}}_k(s,\dot{s},\ddot{s}) = \mathbf{j}''^{\,\omega_k}_T(s)\,\dot{s}^2 + \mathbf{j}'^{\,\omega_k}_T(s)\,\ddot{s}\,, \tag{3.79}$$

where $\mathbf{j}'^{\,\omega_k}_T(s)$ is defined according to (3.67), while

$$\mathbf{j}''^{\,\omega_k}_T(s) := [\mathbf{J}'_{\omega_k}(s)\,\mathbf{a}(s) + \mathbf{J}_{\omega_k}(s)\,\mathbf{b}(s)] \times \mathbf{j}_T^{\omega_k}(s) + \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}'^{\,\omega_k}_T(s)\,. \tag{3.80}$$

The evaluation of terms $\ddot{\mathbf{j}}_T^{v_k}(s,\dot{s},\ddot{s})$ requires a similar procedure. For prismatic joints, because of (3.57), term $\ddot{\mathbf{j}}_T^{v_k}(s,\dot{s},\ddot{s})$ still coincides with $\ddot{\hat{\mathbf{z}}}_k$, so that it can be derived from (3.78) by applying the same substitutions. The sole difference is represented by term $\hat{\mathbf{z}}_k$ that, according to (3.61), is now given by $\mathbf{j}_T^{v_k}$. Consequently it is possible to assert

$$\begin{aligned}\ddot{\mathbf{j}}_T^{v_k}(s,\dot{s},\ddot{s}) = \Big\{&[\mathbf{J}'_{\omega_k}(s)\,\mathbf{a}(s) + \mathbf{J}_{\omega_k}(s)\,\mathbf{b}(s)] \times \mathbf{j}_T^{v_k}(s)\\ &+ \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times [\mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}_T^{v_k}(s)]\Big\}\,\dot{s}^2 + \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}_T^{v_k}(s)\,\ddot{s}\\ = {}&\mathbf{j}''^{\,v_k}_T(s)\,\dot{s}^2 + \mathbf{j}'^{\,v_k}_T(s)\,\ddot{s}\,,\end{aligned}$$

where $\mathbf{j}'^{v_k}_T(s)$ is given by (3.70), while

$$\mathbf{j}''^{v_k}_T(s) := [\mathbf{J}'_{\omega_k}(s)\,\mathbf{a}(s) + \mathbf{J}_{\omega_k}(s)\,\mathbf{b}(s)] \times \mathbf{j}^{v_k}_T(s) + \mathbf{J}_{\omega_k}(s)\,\mathbf{a}(s) \times \mathbf{j}'^{v_k}_T(s)\,. \tag{3.81}$$

Equation (3.57) is also the starting point for the evaluation of terms $\ddot{\mathbf{j}}^{v_k}_T(s,\dot{s},\ddot{s})$ for revolute joints. The linear velocity and acceleration of any frame k, including tool frame T, are given by

$$\begin{aligned}
\mathbf{v}_k &= \mathbf{J}_{v_k}\,\dot{\mathbf{q}}\,,\\
\mathbf{a}_k &= \dot{\mathbf{J}}_{v_k}\,\dot{\mathbf{q}} + \mathbf{J}_{v_k}\,\ddot{\mathbf{q}}\,,
\end{aligned}$$

so that the second equation of (3.57) can be rewritten as follows

$$\ddot{\mathbf{j}}^{v_k}_T = \ddot{\hat{\mathbf{z}}}_k \times [\mathbf{p}_T - \mathbf{p}_k] + 2\,\dot{\hat{\mathbf{z}}}_k \times [\mathbf{J}_{v_T} - \mathbf{J}_{v_k}]\,\dot{\mathbf{q}} + \hat{\mathbf{z}}_k \times [(\dot{\mathbf{J}}_{v_T} - \dot{\mathbf{J}}_{v_k})\,\dot{\mathbf{q}} + (\mathbf{J}_{v_T} - \mathbf{J}_{v_k})\,\ddot{\mathbf{q}}]$$

The analytic expression of $\ddot{\mathbf{j}}^{v_k}_T(s,\dot{s},\ddot{s})$ is obtained by applying the following substitutions: $\ddot{\hat{\mathbf{z}}}_k(s,\dot{s},\ddot{s})$ is given by (3.79), $\mathbf{p}_T(s) - \mathbf{p}_k(s)$ coincides with (3.64), $\dot{\hat{\mathbf{z}}}_k(s,\dot{s})$ is given by (3.65), $\dot{\mathbf{q}}(s,\dot{s})$ is given by (3.30), $\ddot{\mathbf{q}}(s,\dot{s},\ddot{s})$ is given by (3.32), and, finally, $\dot{\mathbf{J}}_{v_k}$ and $\dot{\mathbf{J}}_{v_T}$ can be evaluated by means of (3.74). As a consequence, $\ddot{\mathbf{j}}^{v_k}_T(s,\dot{s},\ddot{s})$ assumes the following expression

$$\begin{aligned}
\ddot{\mathbf{j}}^{v_k}_T(s,\dot{s},\ddot{s}) = {} & \left[\mathbf{j}''^{\omega_k}_T(s)\,\dot{s}^2 + \mathbf{j}'^{\omega_k}_T(s)\,\ddot{s}\right] \times \Delta_k(s) + 2\,\mathbf{j}'^{\omega_k}_T(s) \times [\mathbf{J}_{v_T}(s) - \mathbf{J}_{v_k}(s)]\mathbf{a}(s)\,\dot{s}^2\\
& + \mathbf{j}^{\omega_k}_T(s) \times \left\{[\mathbf{J}'_{v_T}(s) - \mathbf{J}'_{v_k}(s)]\,\mathbf{a}(s)\,\dot{s}^2 + [\mathbf{J}_{v_T}(s) - \mathbf{J}_{v_k}(s)]\,[\mathbf{a}(s)\,\ddot{s} + \mathbf{b}(s)\,\dot{s}^2]\right\}\\
= {} & \mathbf{j}''^{v_k}_T(s)\,\dot{s}^2 + \mathbf{j}'^{v_k}_T(s)\ddot{s}\,.
\end{aligned}$$

where $\mathbf{j}'^{v_k}_T(s)$ is given by (3.72), while

$$\begin{aligned}
\mathbf{j}''^{v_k}_T(s) = {} & \mathbf{j}''^{\omega_k}_T(s) \times \Delta_k(s) + 2\,\mathbf{j}'^{\omega_k}_T(s) \times [\mathbf{J}_{v_T}(s) - \mathbf{J}_{v_k}(s)]\,\mathbf{a}(s)\\
& + \mathbf{j}^{\omega_k}_T(s) \times \Big\{[\mathbf{J}'_{v_T}(s) - \mathbf{J}'_{v_k}(s)]\,\mathbf{a}(s) + [\mathbf{J}_{v_T}(s) - \mathbf{J}_{v_k}(s)]\,\mathbf{b}(s)\Big\}\,.
\end{aligned} \tag{3.82}$$

It is thus clear that, independently from the type of joint, terms $\ddot{\mathbf{j}}^{v_k}_T(s,\dot{s},\ddot{s})$ and $\ddot{\mathbf{j}}^{\omega_k}_T(s,\dot{s},\ddot{s})$ always admit the following structures

$$\ddot{\mathbf{j}}^{v_k}_T(s,\dot{s},\ddot{s}) = \mathbf{j}''^{v_k}_T(s)\,\dot{s}^2 + \mathbf{j}'^{v_k}_T(s)\,\ddot{s}\,, \tag{3.83}$$

Table 3.4: Terms that are required for the evaluation of the second time derivative of the Jacobian matrix.

	$\mathbf{j}_T''^{v_k}(s)$	$\mathbf{j}_T'^{v_k}(s)$	$\mathbf{j}_T''^{\omega_k}(s)$	$\mathbf{j}_T'^{\omega_k}(s)$
prismatic	(3.81)	(3.70)	$\mathbf{0}$	$\mathbf{0}$
revolute	(3.82)	(3.72)	(3.80)	(3.71)

$$\ddot{\mathbf{j}}_T^{\omega_k}(s,\dot{s},\ddot{s}) = \mathbf{j}_T''^{\omega_k}(s)\,\dot{s}^2 + \mathbf{j}_T'^{\omega_k}(s)\,\ddot{s}\,, \tag{3.84}$$

whose terms are defined according to Table 3.4 . Consequently, the second derivative of the Jacobian matrix can be formally represented as follows

$$\ddot{\mathbf{J}}_T(s,\dot{s},\ddot{s}) = \mathbf{J}_T''(s)\,\dot{s}^2 + \mathbf{J}_T'(s)\,\ddot{s}\,. \tag{3.85}$$

4

Singularity avoidance system: Online orientation modification

As stated in Chapter 3, planning trajectories in the operational space could easily result in exceeding the physical limits of electromechanical systems especially when trajectories pass close to singular points. By considering trajectory planning problems for non-redundant manipulators, online reaction to these critical situations is a challenging problem that has only been partially dealt with in robotic literature. As shown in Chapter 3, the path-velocity decomposition paradigm [34] could be an appropriate scheme to deal with kinematic singularities. It has been previously mentioned that this result can be achieved through two different planning strategies: By slowing down the longitudinal velocity of the robot or by introducing slight changes to the tool orientation in order to preserve the execution time of the task. The former methodology has been widely discussed in Chapter 3, while the latter is presented in this chapter.

The main goal of the real-time trajectory planner proposed here is to preserve the user-defined time-law and Cartesian path: If an assigned trajectory passes close to singular configurations, small changes to the tool orientation can be tolerated in order to maintain joint velocities and accelerations confined within the robot physical limits. The original trajectory is restored as soon as the kinematic conditions do not endanger trajectory feasibility.

While many industrial processes require the preservation of both paths and time-laws, robotic literature does not propose a wide solution spectrum to the aforementioned problem. Several robotic application fields exist where the planned time-law is mandatory for proper task execution: For example automatic painting [52,53], gluing or welding [54], in which alterations of the motion longitudinal speed would cause task quality losses. For those applications, the problem is dealt with, when possible, by means of offline constrained optimization algorithms. For example, for painting applications, trajectories can be planned by considering performance indexes like paint coating uniformity [55], task execution time [56], or energy consumption [57].

In real-time environments, trajectory feasibility cannot be checked in advance, so that appropriate systems are required for the online handling of constraints. For this purpose, it has been noticed that for many industrial applications, small changes to the tool orientation minimally affect the quality of the final result. For example, papers [58,59] have shown that paint coating quality is mainly affected by longitudinal velocity and by accurate Cartesian-path tracking, while small orientation errors (± 20 deg) only slightly influence the final result. Even larger errors can be admitted in case of thermally sprayed coatings [60]. This is the underlying assumption adopted in [61] for the solution of the feasibility problem: Constraints on joint velocities and accelerations are fulfilled by introducing minimal changes to the tool orientation. Such changes are evaluated through a non-linear programming algorithm which is executed at each sample time. The method is explicitly suited for reduced speed applications since, on principle, during the approach to singular points it only permits minimal changes to the tool-frame orientation: When critical regions are finally reached, more significant actions must be taken for singularity avoidance, which could alone lead to constraint violations. Recent approaches deal with the real-time orientation modification problem by solving an optimization problem at each sample time [62]. The trajectory updating time has been set equal to $10 \cdot 10^{-3}$ s, i.e. it is much higher than the usual sampling times of control systems.

The Singularity Avoidance System (SAS) described in this chapter handles the same problem considered in [61] and in [62], but the proposed solution is conceived to work even in conjunction with high speeds in the operational space. This result is achieved by ensuring anticipated actions with respect to singularities. Roughly speaking, as soon as a singular point is detected, the tool-frame orientation is properly modified by fulfilling the given constraints, so that close to critical points, only minor adaptations are required. The following requirements have been assumed for the SAS synthesis: Given a provisional and possibly unfeasible trajectory defined in the operational space and passing close to a singular point, the SAS must smoothly introduce, when necessary and in real-time, small variations to the tool-frame orientation in order to guarantee feasible joint velocities and accelerations and preserve the assigned Cartesian-path and longitudinal speed. According to the premises, only minor orientation changes are admitted.

Following the definition of the problem and the explanation of the proposed solution (§4.1), the blocks that constitute the SAS are described in §4.2, §4.3, and §4.4. Afterwards, the proof of the effectiveness of the proposed method is reported in §4.5 in a simulation environment and in §4.6 in experimental tests based on the Comau Smart-Six manipulator.

4.1 Problem definition and proposed solution

Trajectories in the operational space are typically specified by defining proper time functions which describe position and orientation of tool frame T. For example, they can be expressed through the following signals described w.r.t. an inertial frame (typically represented by manipulator frame 0)

${}^{0}_{T}\mathbf{R}(t)$	Rotation matrix: orientation of frame T;
$\mathbf{p}_T(t)$	Position of frame T;
$\boldsymbol{\omega}_T(t)$	Angular velocity of frame T;
$\mathbf{v}_T(t)$	Linear velocity of frame T;
$\boldsymbol{\alpha}_T(t)$	Angular acceleration of frame T;
$\mathbf{a}_T(t)$	Linear acceleration of frame T.

According to the scheme shown in Fig. 4.1, the Cartesian planner is normally followed by an inverse kinematics block which returns an equivalent trajectory in the

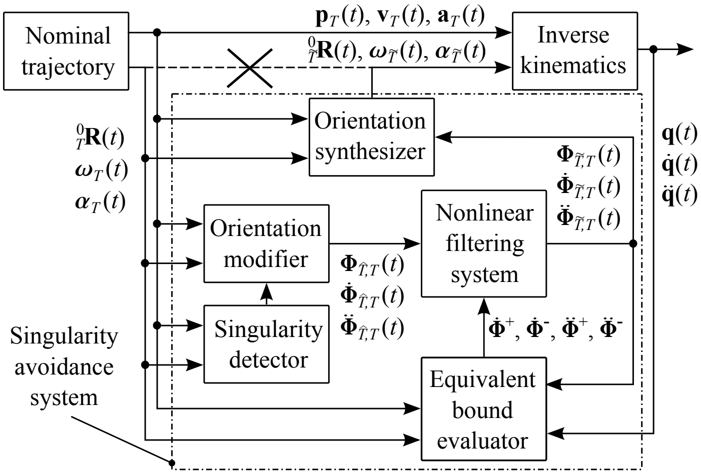
Nominal trajectory
$\mathbf{p}_T(t)$, $\mathbf{v}_T(t)$, $\mathbf{a}_T(t)$
${}^0_{\tilde{T}}\mathbf{R}(t)$, $\boldsymbol{\omega}_{\tilde{T}}(t)$, $\boldsymbol{\alpha}_{\tilde{T}}(t)$
Inverse kinematics
${}^0_T\mathbf{R}(t)$
$\boldsymbol{\omega}_T(t)$
$\boldsymbol{\alpha}_T(t)$
Orientation synthesizer
$\mathbf{q}(t)$
$\dot{\mathbf{q}}(t)$
$\ddot{\mathbf{q}}(t)$
$\boldsymbol{\Phi}_{\tilde{T},T}(t)$
$\dot{\boldsymbol{\Phi}}_{\tilde{T},T}(t)$
$\ddot{\boldsymbol{\Phi}}_{\tilde{T},T}(t)$
Nonlinear filtering system
Orientation modifier
$\boldsymbol{\Phi}_{\hat{T},T}(t)$
$\dot{\boldsymbol{\Phi}}_{\hat{T},T}(t)$
$\ddot{\boldsymbol{\Phi}}_{\hat{T},T}(t)$
Singularity detector
$\dot{\boldsymbol{\Phi}}^+$, $\dot{\boldsymbol{\Phi}}^-$, $\ddot{\boldsymbol{\Phi}}^+$, $\ddot{\boldsymbol{\Phi}}^-$
Singularity avoidance system
Equivalent bound evaluator

Figure 4.1: A schematic representation of the singularity avoidance system.

configuration space that, in turn, is used to drive the joint actuators.

When trajectories are planned in real time, it is not possible to guarantee in advance their feasibility w.r.t. the physical limits of the system. In particular, joint velocities could exceed the given bounds or available motor torques could be insufficient to guarantee the required joint accelerations. If $\mathbf{q} \in \mathbb{R}^N$ is the vector of the joint generalized variables, the following requirements must be fulfilled:

$$\underline{\dot{\mathbf{q}}} \leq \dot{\mathbf{q}} \leq \bar{\dot{\mathbf{q}}}, \tag{4.1}$$

$$\underline{\ddot{\mathbf{q}}} \leq \ddot{\mathbf{q}} \leq \bar{\ddot{\mathbf{q}}}, \tag{4.2}$$

where $\underline{\dot{\mathbf{q}}}$, $\underline{\ddot{\mathbf{q}}} \in (\mathbb{R}^-)^N$, and $\bar{\dot{\mathbf{q}}}$, $\bar{\ddot{\mathbf{q}}} \in (\mathbb{R}^+)^N$ represent proper bounds for joint velocities and accelerations.

In order to avoid possible feasibility issues, the standard planning scheme is modified according to Fig. 4.1. In particular, while the position signals are sent unchanged to the inverse kinematics block, the orientation signals are preprocessed by the Singularity Avoidance System (SAS) which slightly modifies the tool frame orientation in order to preserve feasibility.

The first block of the SAS is the Singularity Detector (SD). It detects possible singular configurations along the nominal trajectory and raises an alarm in order to activate the SAS. Such alarm is then read by the Orientation Modifier (OM), which specifies, through an auxiliary frame $\hat{T}$, a new candidate orientation for the tool frame in order to skip the singularity. The reciprocal orientation between nominal reference frame T and $\hat{T}$ is described by means of vector $\mathbf{\Phi}_{\hat{T},T} := [\alpha\ \beta\ \gamma]^T$, which corresponds to a Roll-Pitch-Yaw (RPY) minimal notation. Since only minor trajectory modifications can be admitted, $\mathbf{\Phi}_{\hat{T},T}$ is generally close to zero. The SD and the OM blocks are described in §4.2.

The trajectory of frame $\hat{T}$ can still be unfeasible. In fact, the avoidance of singular points, by itself, is not sufficient to guarantee that the modified trajectory is feasible w.r.t. (4.1) and (4.2). On the contrary, any trajectory change that is made in configurations which are close to singularities could potentially require additional joint velocities and accelerations, thus worsening an already critical situation. For this reason, the OM block is followed by a Nonlinear Filtering System (NFS) which constrains the dynamics of $\mathbf{\Phi}_{\hat{T},T}$ between proper bounds. The NSF is made by three independent scalar filters, each of them acting on a single component of $\mathbf{\Phi}_{\hat{T},T}$. The NFS returns a signal $\mathbf{\Phi}_{\tilde{T},T}$ which represents the best possible approximation of $\mathbf{\Phi}_{\hat{T},T}$ satisfying the following inequalities

$$\dot{\mathbf{\Phi}}^- \leq \dot{\mathbf{\Phi}}_{\tilde{T},T} \leq \dot{\mathbf{\Phi}}^+, \tag{4.3}$$

$$\ddot{\mathbf{\Phi}}^- \leq \ddot{\mathbf{\Phi}}_{\tilde{T},T} \leq \ddot{\mathbf{\Phi}}^+ \tag{4.4}$$

where $\dot{\mathbf{\Phi}}^-, \ddot{\mathbf{\Phi}}^- \in (\mathbb{R}^-)^3$ and $\dot{\mathbf{\Phi}}^+, \ddot{\mathbf{\Phi}}^+ \in (\mathbb{R}^+)^3$ are proper limits that are devised by the Equivalent Bound Evaluator (EBE) block starting from the actual constraints of the actuators and from the manipulator status of motion. As shown in §4.4, if (4.3) and (4.4) are satisfied, then (4.1) and (4.2) hold with certainty and the trajectory is feasible. The internal structure of the NFS filters, since they must constrain velocities and accelerations of the auxiliary frame $\mathbf{\Phi}_{\tilde{T},T}$, has been extensively described in [29]. Conversely, the filter proposed in Chapter 1 is able to constrain also the jerk signal and, for the problem at hand, is oversized. In order to obtain lower computational burden, the filter presented in [29] is the best choice.

The last block shown in Fig. 4.1, i.e. the Orientation Synthesizer (OS), returns ${}^{0}_{\tilde{T}}\mathbf{R}(t)$, $\boldsymbol{\omega}_{\tilde{T}}(t)$, and $\boldsymbol{\alpha}_{\tilde{T}}(t)$ of the modified trajectory, which are obtained, according to the procedure described in §4.3, from ${}^{0}_{T}\mathbf{R}(t)$, $\boldsymbol{\omega}_T(t)$, and $\boldsymbol{\alpha}_T(t)$ and from the knowledge of $\boldsymbol{\Phi}_{\widetilde{T},T}$, $\dot{\boldsymbol{\Phi}}_{\widetilde{T},T}$, and $\ddot{\boldsymbol{\Phi}}_{\widetilde{T},T}$.

4.2 Singularity Detector and Orientation Modifier blocks

As specified in §4.1, the SD block detects the insurgence of singular configurations and raises an alarm that activates the OM block. As early anticipated, the determination of the alarm signal represents a crucial point because singular configurations must be detected sufficiently in advance in order to have enough time to smoothly modify the tool orientation.

Singular configurations depend on the manipulator pose, and can be detected by analyzing the manipulability index. According to [63], the manipulability index associated to $\mathbf{q}$ is defined as follows

$$j(\mathbf{q}) = \frac{\sigma_m(\mathbf{q})}{\sigma_M(\mathbf{q})} \in [0,1]$$

where $\sigma_m(\mathbf{q})$ and $\sigma_M(\mathbf{q})$ are, respectively, the minimum and the maximum singular values associated to the Jacobian matrix of frame T. If $j(\mathbf{q})$ becomes equal to zero, the manipulator is in a singular configuration.

The alarm function exploits the reciprocal of the manipulability index

$$a(\mathbf{q}) := \frac{1}{j(\mathbf{q})},$$

and its time derivative

$$b(\mathbf{q}) := \left| \frac{d}{dt} \frac{1}{j(\mathbf{q})} \right| .$$

The typical shapes of $a(\mathbf{q})$ and $b(\mathbf{q})$, for a trajectory passing close to a singularity, are shown in Fig. 4.2. Sudden changes of $b(\mathbf{q})$ announce that a critical point is approaching then, closer to such point, $b(\mathbf{q})$ goes to zero but $a(\mathbf{q})$ raises, so that the alarm signal can simply be given by the following logic function

$$SD(\mathbf{q}) := \begin{cases} \text{true} & \text{if } \left[\, a(\mathbf{q}) \geq \overline{a} \,\right] \text{ or } \left[\, b(\mathbf{q}) \geq \overline{b} \,\right] \\ \text{false} & \text{otherwise} \end{cases} \tag{4.5}$$

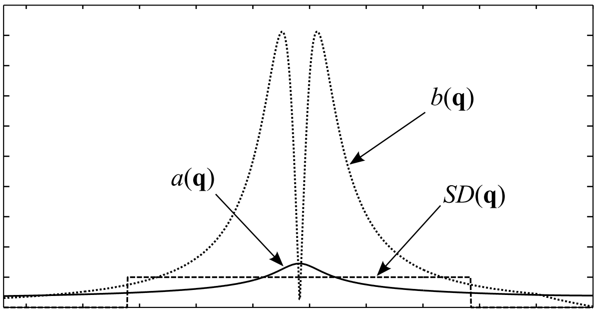

Figure 4.2: Signal $a(\mathbf{q})$ (solid line) and $b(\mathbf{q})$ (dotted line) that are used for the evaluation of alarm signal $SD(\mathbf{q})$ (dashed line).

where $\overline{a}$ and $\overline{b}$ represent appropriate thresholds.

Once the alarm has been raised, the OM block synthesizes the new candidate orientation for the tool frame, which is expressed through an auxiliary frame $\hat{T}$. The reciprocal orientation between nominal reference frame T and modified reference frame $\hat{T}$ is represented by means of an RPY minimal notation and it is given by

$$\mathbf{\Phi}_{\hat{T},T}(\mathbf{q}) := c(\mathbf{q})\ \hat{\mathbf{\Phi}}_{\hat{T},T}(\mathbf{q})\ , \tag{4.6}$$

where $\hat{\mathbf{\Phi}}_{\hat{T},T}(\mathbf{q})$ is a vector which fixes the “direction” of the reciprocal rotation between T and $\hat{T}$, while $c(\mathbf{q})$ is a scalar which represents the amplitude of such rotation. $c(\mathbf{q})$ depends on longitudinal velocity $\mathbf{v}_T$ and on the distance between end effector and singular point: Larger rotations are admitted close to critical configurations and for high speeds, according to the following function

$$c(\mathbf{q}) := l(\mathbf{v}_T)\ a(\mathbf{q})\ , \tag{4.7}$$

where

$$l(\mathbf{v}_T) := \lambda_1\,|\mathbf{v}_T| + \lambda_2 \tag{4.8}$$

is chosen such to guarantee that changes of the tool-frame orientation are increasingly proportional to $\mathbf{v}_T$.

Vector $\hat{\mathbf{\Phi}}_{\hat{T},T}(\mathbf{q})$ is chosen by inspecting the surroundings of the nominal trajectory according to the following equation

$$\hat{\boldsymbol{\Phi}}_{\hat{T},T}(\mathbf{q}) = r\frac{[\overline{\alpha}\,\overline{\beta}\,\overline{\gamma}]^T}{\left\|[\overline{\alpha}\,\overline{\beta}\,\overline{\gamma}]^T\right\|},$$

where $\overline{\alpha},\overline{\beta},\overline{\gamma} \in \{-1,0,1\}$ and $r = 1 \cdot 10^{-3}$ rad. Practically, $\hat{\boldsymbol{\Phi}}_{\hat{T},T}(\mathbf{q})$ is chosen by considering all possible permutations of $\overline{\alpha},\overline{\beta},\overline{\gamma} \in \{-1,0,1\}$ and by checking, for each of them, the corresponding manipulability index: The value of $\hat{\boldsymbol{\Phi}}_{\hat{T},T}(\mathbf{q})$ which admits the higher manipulability index is used in (4.6) for the determination of the new tool-frame orientation.

4.3 The Orientation Synthesizer block

The OS is the SAS output block which combines the nominal trajectory with $\boldsymbol{\Phi}_{\widetilde{T},T} := [\alpha\ \beta\ \gamma]^T$, $\dot{\boldsymbol{\Phi}}_{\widetilde{T},T} := [\dot{\alpha}\ \dot{\beta}\ \dot{\gamma}]^T$, and $\ddot{\boldsymbol{\Phi}}_{\widetilde{T},T} := [\ddot{\alpha}\ \ddot{\beta}\ \ddot{\gamma}]^T$, i.e. with the signals that are provided by the NFS and whose synthesis is discussed in next §4.4. Changes only affect the tool frame orientation and, consequently, its angular velocity and acceleration while, according to the premises, signals $\mathbf{p}_T(t)$, $\mathbf{v}_T(t)$, and $\mathbf{a}_T(t)$ are left unchanged in order to preserve the geometry of the Cartesian path.

The modified orientation of the tool frame can be evaluated according to the following expression

$${}^{0}_{\widetilde{T}}\mathbf{R} = {}^{0}_{T}\mathbf{R}\,{}^{T}_{\widetilde{T}}\mathbf{R}(\boldsymbol{\Phi}_{\widetilde{T},T}),$$

where ${}^{T}_{\widetilde{T}}\mathbf{R}(\boldsymbol{\Phi}_{\widetilde{T},T})$ is the rotation matrix corresponding to minimal notation $\boldsymbol{\Phi}_{\widetilde{T},T}$.

The relative angular velocity between frame $\widetilde{T}$ and frame T can always be expressed as follows

$${}^{T}\boldsymbol{\omega}_{\widetilde{T},T} = \mathbf{T}(\boldsymbol{\Phi}_{\widetilde{T},T})\,\dot{\boldsymbol{\Phi}}_{\widetilde{T},T}, \tag{4.9}$$

where $\mathbf{T}(\boldsymbol{\Phi}_{\widetilde{T},T})$ is a proper matrix which correlates angular velocities to $\dot{\boldsymbol{\Phi}}_{\widetilde{T},T}$. The structure of $\mathbf{T}(\boldsymbol{\Phi}_{\widetilde{T},T})$ depends on the minimal notation. For the RPY notation, it can be easily proved that it is given by

$$\mathbf{T}(\boldsymbol{\Phi}_{\widetilde{T},T}) = \begin{bmatrix} 1 & 0 & \sin(\beta) \\ 0 & \cos(\alpha) & -\sin(\alpha)\cos(\beta) \\ 0 & \sin(\alpha) & \cos(\alpha)\cos(\beta) \end{bmatrix}. \tag{4.10}$$

In the following, the argument of $\mathbf{T}(\cdot)$ will be omitted for conciseness.

Bearing in mind (4.9), it is immediately possible to derive the angular velocity of frame $\widetilde{T}$ by means of the following equation

$$\boldsymbol{\omega}_{\widetilde{T}} = \boldsymbol{\omega}_T + \boldsymbol{\omega}_{\widetilde{T},T} = \boldsymbol{\omega}_T + {}^0_T\mathbf{R}^T\boldsymbol{\omega}_{\widetilde{T},T} = \boldsymbol{\omega}_T + {}^0_T\mathbf{R}\,\mathbf{T}\dot{\mathbf{\Phi}}_{\widetilde{T},T}\,. \tag{4.11}$$

The angular acceleration of $\widetilde{T}$ can be evidently obtained by deriving (4.11), i.e. according to the following expression

$$\begin{aligned}\boldsymbol{\alpha}_{\widetilde{T}} &= \boldsymbol{\alpha}_T + {}^0_T\dot{\mathbf{R}}\,\mathbf{T}\dot{\mathbf{\Phi}}_{\widetilde{T},T} + {}^0_T\mathbf{R}\,\dot{\mathbf{T}}\dot{\mathbf{\Phi}}_{\widetilde{T},T} + {}^0_T\mathbf{R}\,\mathbf{T}\ddot{\mathbf{\Phi}}_{\widetilde{T},T}\\ &= \boldsymbol{\alpha}_T + \boldsymbol{\omega}_T \times {}^0_T\mathbf{R}\,\mathbf{T}\dot{\mathbf{\Phi}}_{\widetilde{T},T} + {}^0_T\mathbf{R}\left[\dot{\bar{\mathbf{T}}}\dot{\bar{\mathbf{\Phi}}}_{\widetilde{T},T} + \mathbf{T}\ddot{\mathbf{\Phi}}_{\widetilde{T},T}\right],\end{aligned} \tag{4.12}$$

where $\dot{\bar{\mathbf{\Phi}}}_{\widetilde{T},T} := [\dot{\alpha}\dot{\beta}\ \dot{\gamma}\dot{\alpha}\ \dot{\beta}\dot{\gamma}]^T$ and

$$\dot{\bar{\mathbf{T}}} := \begin{bmatrix} 0 & 0 & \cos(\beta) \\ -\sin(\alpha) & -\cos(\alpha)\cos(\beta) & \sin(\alpha)\sin(\beta) \\ \cos(\alpha) & -\sin(\alpha)\cos(\beta) & -\cos(\alpha)\sin(\beta) \end{bmatrix}.$$

4.4 The Equivalent Bounds Evaluator block

As stated in §4.1, the SAS scheme requires that bounds $\dot{\underline{\mathbf{q}}}$, $\ddot{\underline{\mathbf{q}}}$, $\dot{\overline{\mathbf{q}}}$, and $\ddot{\overline{\mathbf{q}}}$ must be converted into equivalent bounds on $\dot{\mathbf{\Phi}}^-, \ddot{\mathbf{\Phi}}^-, \dot{\mathbf{\Phi}}^+$, and $\ddot{\mathbf{\Phi}}^+$. Such conversion depends on the current displacement of the tool frame and, more precisely, on its current position, assumed to coincide with $\mathbf{p}_{\widetilde{T}} = \mathbf{p}_T$, and on its current orientation, assumed to be given by ${}^0_{\widetilde{T}}\mathbf{R}$. The solution of the inverse kinematic problem for ${}^0_{\widetilde{T}}\mathbf{R}$ and $\mathbf{p}_T$ returns the current value of $\mathbf{q}$, which can be used for the evaluation of the Jacobian matrix $\mathbf{J}_{\widetilde{T}}$ that, for this reason, from now on will be supposed to be known.

In non-singular configurations – remember that the proposed approach is used to avoid such configurations – it is possible to write

$$\dot{\mathbf{q}} = \mathbf{J}_{\widetilde{T}}^{-1}\bar{\mathbf{v}}_{\widetilde{T}}\,,$$

where $\bar{\mathbf{v}}_{\widetilde{T}} := [\mathbf{v}_{\widetilde{T}}^T\ \boldsymbol{\omega}_{\widetilde{T}}^T]^T$ represents the generalized velocity of $\widetilde{T}$. By partitioning $\mathbf{J}_{\widetilde{T}}^{-1}$ into two sub-matrices such that

$$\mathbf{J}_{\widetilde{T}}^{-1} := \left[\mathbf{J}_{v_{\widetilde{T}}}^{-1} \mid \mathbf{J}_{\omega_{\widetilde{T}}}^{-1}\right], \tag{4.13}$$

it is possible to write

$$\dot{\mathbf{q}} = \mathbf{J}_{v_{\widetilde{T}}}^{-1}\mathbf{v}_{\widetilde{T}} + \mathbf{J}_{\omega_{\widetilde{T}}}^{-1}\boldsymbol{\omega}_{\widetilde{T}} \,.$$

Velocity $\mathbf{v}_{\widetilde{T}}$ coincides with $\mathbf{v}_T$, while $\boldsymbol{\omega}_{\widetilde{T}}$ is given by (4.11), so that constraint condition (4.1) can be rewritten as follows

$$\underline{\dot{\mathbf{q}}} \leq \mathbf{J}_{v_{\widetilde{T}}}^{-1}\mathbf{v}_T + \mathbf{J}_{\omega_{\widetilde{T}}}^{-1}\left[\boldsymbol{\omega}_T + {}^{0}_{T}\mathbf{R}\,\mathbf{T}\,\dot{\mathbf{\Phi}}_{\widetilde{T},T}\right] \leq \dot{\overline{\mathbf{q}}},$$

or, analogously,

$$\dot{\widetilde{\mathbf{q}}}^{-} \leq \mathbf{J}_{\omega_{\widetilde{T}}}^{-1}\,{}^{0}_{T}\mathbf{R}\,\mathbf{T}\,\dot{\mathbf{\Phi}}_{\widetilde{T},T} \leq \dot{\widetilde{\mathbf{q}}}^{+}, \tag{4.14}$$

where

$$\dot{\widetilde{\mathbf{q}}}^{-} := \underline{\dot{\mathbf{q}}} - \mathbf{J}_{\widetilde{T}}^{-1}\overline{\mathbf{v}}_T, \tag{4.15}$$

$$\dot{\widetilde{\mathbf{q}}}^{+} := \dot{\overline{\mathbf{q}}} - \mathbf{J}_{\widetilde{T}}^{-1}\overline{\mathbf{v}}_T. \tag{4.16}$$

Any feasible $\dot{\mathbf{\Phi}}_{\widetilde{T},T}$ must fulfill (4.14). All the terms in (4.15) and (4.16) are known: $\underline{\dot{\mathbf{q}}}$ and $\dot{\overline{\mathbf{q}}}$ are defined by the users, $\mathbf{J}_{\widetilde{T}}^{-1}$ is the inverse Jacobian of frame $\widetilde{T}$, $\overline{\mathbf{v}}_T$ is the generalized velocity of frame T. In the same way, many of the terms in (4.14) are known: $\mathbf{J}_{\omega_{\widetilde{T}}}^{-1}$ derives from (4.13), ${}^{0}_{T}\mathbf{R}$ is the rotation matrix of frame T, and $\mathbf{T}$ is given by (4.10).

The acceleration constraints can be similarly handled. It is well known that

$$\overline{\mathbf{a}}_{\widetilde{T}} = \dot{\mathbf{J}}_{\widetilde{T}}\dot{\mathbf{q}} + \mathbf{J}_{\widetilde{T}}\ddot{\mathbf{q}},$$

which can be solved for $\ddot{\mathbf{q}}$, thus leading to

$$\ddot{\mathbf{q}} = \mathbf{J}_{\widetilde{T}}^{-1}\left[\overline{\mathbf{a}}_{\widetilde{T}} - \dot{\mathbf{J}}_{\widetilde{T}}\dot{\mathbf{q}}\right] = \mathbf{J}_{\widetilde{T}}^{-1}\left[\overline{\mathbf{a}}_{\widetilde{T}} - \dot{\mathbf{J}}_{\widetilde{T}}\mathbf{J}_{\widetilde{T}}^{-1}\overline{\mathbf{v}}_{\widetilde{T}}\right]. \tag{4.17}$$

By using the partitioning scheme suggested in (4.13), it is possible to rewrite (4.17) as follows

$$\ddot{\mathbf{q}} = \mathbf{J}_{v_{\widetilde{T}}}^{-1}\mathbf{a}_{\widetilde{T}} + \mathbf{J}_{\omega_{\widetilde{T}}}^{-1}\boldsymbol{\alpha}_{\widetilde{T}} - \mathbf{J}_{\widetilde{T}}^{-1}\dot{\mathbf{J}}_{\widetilde{T}}\mathbf{J}_{\widetilde{T}}^{-1}\overline{\mathbf{v}}_{\widetilde{T}},$$

which, in turn, can be written, because of (4.12) and since $\mathbf{a}_{\widetilde{T}} = \mathbf{a}_T$, as follows

$$\ddot{\mathbf{q}} = \mathbf{J}_{v_{\widetilde{T}}}^{-1}\mathbf{a}_T + \mathbf{J}_{\omega_{\widetilde{T}}}^{-1}\left\{\boldsymbol{\alpha}_T + \boldsymbol{\omega}_T \times {}^{0}_{T}\mathbf{R}\,\mathbf{T}\,\dot{\mathbf{\Phi}}_{\widetilde{T},T} + {}^{0}_{T}\mathbf{R}\left[\dot{\overline{\mathbf{T}}}\dot{\overline{\mathbf{\Phi}}}_{\widetilde{T},T} + \mathbf{T}\,\ddot{\mathbf{\Phi}}_{\widetilde{T},T}\right]\right\} - \mathbf{J}_{\widetilde{T}}^{-1}\dot{\mathbf{J}}_{\widetilde{T}}\mathbf{J}_{\widetilde{T}}^{-1}\overline{\mathbf{v}}_{\widetilde{T}}\,. \tag{4.18}$$

By substituting (4.18) into (4.2), after a few manipulations it is possible to obtain

$$\tilde{\ddot{\mathbf{q}}}^{-} \leq \mathbf{J}_{\omega_{\tilde{T}}}^{-1}\,{}^{0}_{T}\mathbf{R}\,\mathbf{T}\,\ddot{\mathbf{\Phi}}_{\tilde{T},T} \leq \tilde{\ddot{\mathbf{q}}}^{+}\,, \tag{4.19}$$

where

$$\begin{aligned}
\tilde{\ddot{\mathbf{q}}}^{-} &:= \underline{\ddot{\mathbf{q}}} - \mathbf{J}_{\tilde{T}}^{-1}\left[\bar{\mathbf{a}}_{T} - \dot{\mathbf{J}}_{\tilde{T}}\mathbf{J}_{\tilde{T}}^{-1}\bar{\mathbf{v}}_{\tilde{T}}\right] - \mathbf{J}_{\omega_{\tilde{T}}}^{-1}\left\{\boldsymbol{\omega}_{T} \times {}^{0}_{T}\mathbf{R}\,\mathbf{T}\,\dot{\mathbf{\Phi}}_{\tilde{T},T} + {}^{0}_{T}\mathbf{R}\,\dot{\bar{\mathbf{T}}}\,\dot{\bar{\mathbf{\Phi}}}_{\tilde{T},T}\right\}\,,\\
\tilde{\ddot{\mathbf{q}}}^{+} &:= \bar{\ddot{\mathbf{q}}} - \mathbf{J}_{\tilde{T}}^{-1}\left[\bar{\mathbf{a}}_{T} - \dot{\mathbf{J}}_{\tilde{T}}\mathbf{J}_{\tilde{T}}^{-1}\bar{\mathbf{v}}_{\tilde{T}}\right] - \mathbf{J}_{\omega_{\tilde{T}}}^{-1}\left\{\boldsymbol{\omega}_{T} \times {}^{0}_{T}\mathbf{R}\,\mathbf{T}\,\dot{\mathbf{\Phi}}_{\tilde{T},T} + {}^{0}_{T}\mathbf{R}\,\dot{\bar{\mathbf{T}}}\,\dot{\bar{\mathbf{\Phi}}}_{\tilde{T},T}\right\}\,.
\end{aligned}$$

Vectors $\tilde{\ddot{\mathbf{q}}}^{-}$ and $\tilde{\ddot{\mathbf{q}}}^{+}$ can be evaluated from the knowledge of the nominal trajectory and of the current values of $\mathbf{\Phi}_{\tilde{T},T}$ and $\dot{\mathbf{\Phi}}_{\tilde{T},T}$. Evidently, any feasible $\ddot{\mathbf{\Phi}}_{\tilde{T},T}$ must fulfill (4.19).

Equations (4.14) and (4.19) represent the starting point for converting the original bounds on $\dot{\mathbf{q}}$ and on $\ddot{\mathbf{q}}$ into equivalent bounds on $\dot{\mathbf{\Phi}}_{\tilde{T},T}$ and on $\ddot{\mathbf{\Phi}}_{\tilde{T},T}$. However, such conversion is not straightforward since $\mathbf{q}$ and $\mathbf{\Phi}_{\tilde{T},T}$ are dimensionally different. More in details, (4.14) and (4.19) represent a set of $2N$ independent constraints on $\dot{\mathbf{q}}$ and on $\ddot{\mathbf{q}}$ that must be converted into 6 equivalent constraint equations on $\dot{\mathbf{\Phi}}_{\tilde{T},T}$ and on $\ddot{\mathbf{\Phi}}_{\tilde{T},T}$. The conversion clearly admits some degrees of freedom that can be used to obtain the best possible performances from the trajectory modification strategy. This is the reason why the equivalent bounds are obtained by solving a Linear Programming (LP) problem which returns a set of proper equivalent limits $\dot{\mathbf{\Phi}}^{-} := [\dot{\alpha}^{-}\ \dot{\beta}^{-}\ \dot{\gamma}^{-}]^{T}, \ddot{\mathbf{\Phi}}^{-} := [\ddot{\alpha}^{-}\ \ddot{\beta}^{-}\ \ddot{\gamma}^{-}]^{T}$ and $\dot{\mathbf{\Phi}}^{+} := [\dot{\alpha}^{+}\ \dot{\beta}^{+}\ \dot{\gamma}^{+}]^{T}, \ddot{\mathbf{\Phi}}^{+} := [\ddot{\alpha}^{+}\ \ddot{\beta}^{+}\ \ddot{\gamma}^{+}]^{T}$ for $\dot{\mathbf{\Phi}}_{\tilde{T},T}$ and on $\ddot{\mathbf{\Phi}}_{\tilde{T},T}$ which guarantee that (4.14) and (4.19) – and, in turn, (4.1) and (4.2) – are certainly satisfied when (4.3) and (4.4) hold.

Available degrees of freedom can be used to improve the behavior of the SAS. More precisely, its best performances can be obtained when bounds on $\dot{\mathbf{\Phi}}_{\tilde{T},T}$ and $\ddot{\mathbf{\Phi}}_{\tilde{T},T}$ are kept well open: In fact, this permits a good reactivity of the trajectory modifier which can rapidly react to critical situations.

By defining $\mathbf{A} := \mathbf{J}_{\omega_{\tilde{T}}}^{-1}\,{}^{0}_{T}\mathbf{R}\,\mathbf{T}$ and bearing in mind (4.14) and (4.19), it is possible formulate the following semi-infinite minimax optimization problem

$$\max_{\substack{\dot{\mathbf{\Phi}}^{+},\ddot{\mathbf{\Phi}}^{+} \in (\mathbb{R}^{+})^{3}\\ \dot{\mathbf{\Phi}}^{-},\ddot{\mathbf{\Phi}}^{-} \in (\mathbb{R}^{-})^{3}}} \ \min_{i=1,\dots,6}\{\Gamma_{i}^{+} - \Gamma_{i}^{-}\} \tag{4.20}$$

subject to

$$\dot{\tilde{\mathbf{q}}}^{-} \leq \mathbf{A}\dot{\mathbf{\Phi}}_{\tilde{T},T} \leq \dot{\tilde{\mathbf{q}}}^{+} \qquad \forall\dot{\mathbf{\Phi}}_{\tilde{T},T} \in [\dot{\mathbf{\Phi}}^{-},\dot{\mathbf{\Phi}}^{+}], \tag{4.21}$$

$$\ddot{\tilde{\mathbf{q}}}^{-} \leq \mathbf{A}\ddot{\mathbf{\Phi}}_{\tilde{T},T} \leq \ddot{\tilde{\mathbf{q}}}^{+} \qquad \forall\ddot{\mathbf{\Phi}}_{\tilde{T},T} \in [\ddot{\mathbf{\Phi}}^{-},\ddot{\mathbf{\Phi}}^{+}], \tag{4.22}$$

where Γ_i^+ and Γ_i^- are the components of vectors $\mathbf{\Gamma}^+ := [k\,\dot{\alpha}^+ \; k\,\dot{\beta}^+ \; k\,\dot{\gamma}^+ \; \ddot{\alpha}^+ \; \ddot{\beta}^+ \; \ddot{\gamma}^+]^T$ and $\mathbf{\Gamma}^- := [k\,\dot{\alpha}^- \; k\,\dot{\beta}^- \; k\,\dot{\gamma}^- \; \ddot{\alpha}^- \; \ddot{\beta}^- \; \ddot{\gamma}^-]^T$, respectively, and k is used to correctly weight the velocity and acceleration bounds. For the problem at hand, in particular, $k = 10$ guarantees acceleration bounds 10 times larger than velocity bounds.

Matrix $\mathbf{A}$, which appears in (4.21) and (4.22) is certainly non-singular. As stated in §4.2, indeed, $\mathbf{\Phi}_{\tilde{T},T}$ is always kept close to zero, so that $\mathbf{T}(\mathbf{\Phi}_{\tilde{T},T})$, owing to (4.10), is almost an identity matrix. Furthermore, the avoidance of singular configurations also guarantees that $\mathbf{J}_{\omega_{\tilde{T}}}^{-1} \in \mathbb{R}^{N\times 3}$, i.e. the rightmost part of the inverse Jacobian, is certainly full rank. Finally, since ${}^{0}_{T}\mathbf{R} \in \mathbb{R}^{3\times 3}$ is always non-singular, it is possible to conclude that $\mathbf{A}$ is a full rank $N \times 3$ real matrix.

Equations (4.21) and (4.22) are clearly linear. This property can be used to convert both semi-infinite constraints into a set of finite constraints. For example, the feasibility of (4.21) can be checked by ignoring the interior points of box $[\dot{\mathbf{\Phi}}^{-},\dot{\mathbf{\Phi}}^{+}]$ and by only inspecting its vertexes. More precisely, the feasibility of (4.21) is guaranteed if the following vertex points are feasible

$$\dot{\mathbf{\Phi}}_1 := [\dot{\alpha}^- \; \dot{\beta}^- \; \dot{\gamma}^-]^T,\; \dot{\mathbf{\Phi}}_2 := [\dot{\alpha}^- \; \dot{\beta}^- \; \dot{\gamma}^+]^T,\; \dot{\mathbf{\Phi}}_3 := [\dot{\alpha}^- \; \dot{\beta}^+ \; \dot{\gamma}^-]^T,\; \dot{\mathbf{\Phi}}_4 := [\dot{\alpha}^- \; \dot{\beta}^+ \; \dot{\gamma}^+]^T,$$
$$\dot{\mathbf{\Phi}}_5 := [\dot{\alpha}^+ \; \dot{\beta}^- \; \dot{\gamma}^-]^T,\; \dot{\mathbf{\Phi}}_6 := [\dot{\alpha}^+ \; \dot{\beta}^- \; \dot{\gamma}^+]^T,\; \dot{\mathbf{\Phi}}_7 := [\dot{\alpha}^+ \; \dot{\beta}^+ \; \dot{\gamma}^-]^T,\; \dot{\mathbf{\Phi}}_8 := [\dot{\alpha}^+ \; \dot{\beta}^+ \; \dot{\gamma}^+]^T.$$

The same concept applies for (4.22), so that (4.20) can be easily reconverted into the following LP problem

$$\max_{\lambda,\dot{\mathbf{\Phi}}^{+},\ddot{\mathbf{\Phi}}^{+},\dot{\mathbf{\Phi}}^{-},\ddot{\mathbf{\Phi}}^{-}} \{\lambda\}$$

subject to

$$\lambda \leq \Gamma_i^+ - \Gamma_i^- \qquad i = 1,2,\ldots,6 \tag{4.23}$$

$$\dot{\mathbf{\Phi}}^{+} \geq 0 \tag{4.24}$$

$$\ddot{\mathbf{\Phi}}^{+} \geq 0 \tag{4.25}$$

$$\dot{\boldsymbol{\Phi}}^{-} \leq 0 \tag{4.26}$$

$$\ddot{\boldsymbol{\Phi}}^{-} \leq 0 \tag{4.27}$$

$$\mathbf{A}\,\dot{\boldsymbol{\Phi}}_k \leq \tilde{\dot{\mathbf{q}}}^{+} \qquad k = 1,2,\ldots,8, \tag{4.28}$$

$$\mathbf{A}\,\dot{\boldsymbol{\Phi}}_k \geq \tilde{\dot{\mathbf{q}}}^{-} \qquad k = 1,2,\ldots,8, \tag{4.29}$$

$$\mathbf{A}\,\ddot{\boldsymbol{\Phi}}_k \leq \tilde{\ddot{\mathbf{q}}}^{+} \qquad k = 1,2,\ldots,8, \tag{4.30}$$

$$\mathbf{A}\,\ddot{\boldsymbol{\Phi}}_k \geq \tilde{\ddot{\mathbf{q}}}^{-} \qquad k = 1,2,\ldots,8. \tag{4.31}$$

LP problems have been studied for a long time and many excellent solvers exist. Thus, a quite natural choice should be to use one of them to solve the above linear problem. For instance, `CPLEX` is one of the fastest and most reliable among the available LP solvers but, unfortunately, for the application at hand it has a drawback. Each linear problem must be solved within the sampling time of the process, i.e. in a few milliseconds. While `CPLEX` is extremely fast in solving linear problems, it wastes a considerable amount of time – at least, compatibly with the sampling time – for the preparation of the environment that is required by the solver. For the considered application, a new problem must be formulated and solved at each sample time, so that, instead of `CPLEX`, a custom LP solver has been implemented by posing a great care in choosing the right strategies in order to reduce computational times. The proposed solution is based on the well-known simplex algorithm (see, e.g., [64]). While such algorithm has an exponential worst-case complexity, it works very well in practice. It has been observed that the number of iterations required by the simplex algorithm is of the same order of magnitude of the number of constraints of the problem. Problem (4.23)-(4.31) has more than 200 constraints but only 13 variables. Thus, rather than applying the simplex algorithm to (4.23)-(4.31), it is convenient to apply it to its dual, where the number of constraints is equal to 13, i.e. the number of variables of problem (4.23)-(4.31). As known, the optimal value of the original problem is equal to the optimal value of the dual problem, while the solution of the original problem can be derived from the dual solution, e.g., by the complementarity conditions (details on the latter assertion can be found, again, in [64]).

An important feature of the simplex algorithm, which has been specifically exploited, is the so-called “warm-start”, i.e. the possibility of solving an LP problem by starting from the solution of another LP problem, when this latter is just a slight perturbation of the former one. This is indeed the case at hand: Each new LP problem (4.23)-(4.31) differs from the preceding one for what concerns matrix **A** and vectors $\dot{\tilde{\mathbf{q}}}^+, \dot{\tilde{\mathbf{q}}}^-, \ddot{\tilde{\mathbf{q}}}^+, \ddot{\tilde{\mathbf{q}}}^-$, but the new entries are just slight perturbations of the old ones.

The optimal solution of an LP problem is strictly related to its optimal basis. Still referring to [64] for details, the optimal solutions of the LP problem and of its dual, as well as their common optimal value, can be easily found once the optimal basis is known. By denoting as B_i^* the optimal basis for problem (4.23)-(4.31) at the i-th sampling instant, the solution of two linear systems allows to establish, at sampling instant $i+1$, whether B_i^* is:

- feasible both for problem (4.23)-(4.31) and for its dual, in which case it is also an optimal basis for the two problems and no further computation is needed (notice that while the optimal basis does not change, the values of the variables and the optimal value usually do);

- feasible for problem (4.23)-(4.31) but not for its dual, in which case the simplex algorithm is applied by using B_i^* as a starting basis (usually, very few iterations are needed to recover optimality);

- not feasible for problem (4.23)-(4.31) but feasible for its dual, in which case the dual simplex algorithm is applied by using B_i^* as a starting basis (again, very few iterations are needed to recover optimality);

- not feasible for problem (4.23)-(4.31) and for its dual, in which case some efforts must be spent, e.g., by applying the *two-phase* simplex method, in order to recover feasibility (or dual feasibility) and start with the simplex (or dual simplex) algorithm (notice that this is also needed at the first sampling instant).

The most time-consuming case is obviously the last one, where some additional effort is needed before the simplex or the dual simplex algorithm is started. However, what has been observed in practice is that most of the runs fall into the first case (at least in 99.56% of the cases), i.e. the new solution is obtained almost instantaneously, while

the fourth case occurs very seldom (at most in 0.04% of the cases). Test experiments executed on a Intel Core2 Duo PC @ 3.00 GHz have shown that the computational times for the LP problems are in range of between $0.11 \cdot 10^{-3}$ s and $1.03 \cdot 10^{-3}$ s, with an average execution time equal to $0.13 \cdot 10^{-3}$ s.
Equivalent bounds $\dot{\mathbf{\Phi}}^{-}, \ddot{\mathbf{\Phi}}^{-}, \dot{\mathbf{\Phi}}^{+}$, and $\ddot{\mathbf{\Phi}}^{+}$, once evaluated by the custom LP solver, are sent to the NFS for the generation of the smooth signals that are used to change the tool orientation. The description of the NFS is omitted for brevity, but its structure and behavior have been proposed and extensively discussed in [29]. As stated in §4.1, it is important to remark that the NFS needs to constrain the orientation modification within bounds on $\dot{\mathbf{\Phi}}_{\widetilde{T},T}$ and $\ddot{\mathbf{\Phi}}_{\widetilde{T},T}$. The filter proposed in Chapter 1 is also able to constrain the jerk signal. In order to limit the computational burden of the whole SAS, the simplified version of the filter proposed in [29] is here used for the NFS synthesis.

4.5 Simulation results

The SAS behavior has been firstly verified via a set of tests computed in a simulation environment based on the Comau Smart-Six kinematics. Conversely, next §4.6 is devoted to present experimental tests executed on the actual manipulator. The user-defined constraint vectors, i.e. $\underline{\dot{\mathbf{q}}}$, $\overline{\dot{\mathbf{q}}}$, $\underline{\ddot{\mathbf{q}}}$, and $\overline{\ddot{\mathbf{q}}}$, have been posed equal to $(i = 1,2,\ldots,N)$: $\underline{\dot{q}}_i = -10$ rad s^{-1}, $\overline{\dot{q}}_i = 10$ rad s^{-1}, $\underline{\ddot{q}}_i = -100$ rad s^{-2} and $\overline{\ddot{q}}_i = 100$ rad s^{-2}. The choice has been motivated by the maximum joint velocity of the actual robot for the velocity constraints and by obtaining smooth transients for the acceleration limits. The SAS parameters have been experimentally tuned such to guarantee orientation changes in the range ±0.1 rad (±5.7 deg) at the maximum longitudinal speed considered for the experiments ($|\mathbf{v}_T| = 0.4$ m s^{-1}). In particular, thresholds in (4.5) and coefficients of (4.8) have been respectively appointed equal to $\overline{a} = 9$, $\overline{b} = 7.5$, $\lambda_1 = 7/450$, and $\lambda_2 = 2/4500$.

In the first test case, the nominal trajectory is defined as a straight line from $\mathbf{p}_A = [0.65\ 0.83\ 1.12]^T$ m to $\mathbf{p}_B = [-0.2\ 0.83\ 1.12]^T$ m. The nominal orientation of the tool-frame is kept constant and, by assuming the RPY notation, it is posed equal

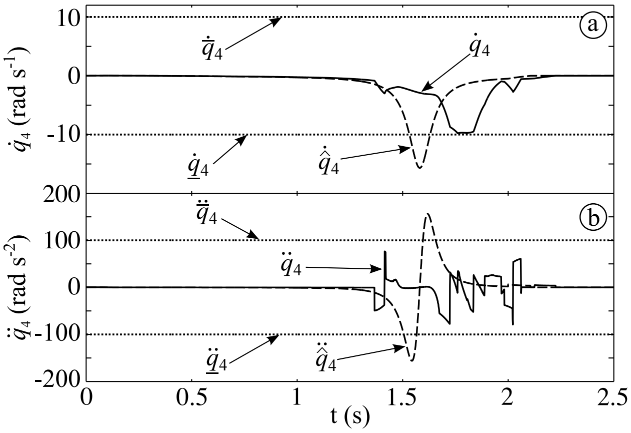

Figure 4.3: (a) Velocities and (b) accelerations of joint 4 obtained with the SAS (solid lines) compared with those obtained without the SAS (dashed lines). Dotted lines highlight the assigned bounds.

to $\mathbf{\Phi}_{T,0} = [\pi/2\; 0\; \pi/2]^T$ rad. The trajectory passes close to a wrist singularity located at $\bar{\mathbf{p}}_T = [0\; 0.83\; 1.077]^T$ m with a longitudinal velocity equal to 0.4 m s^{-1}: If the SAS is not used, velocities and accelerations of joints 4 and 6 become unfeasible. Fig. 4.3 shows, in particular, the behavior of joint 4: Without the SAS (dashed line) the velocity and the acceleration limits are violated while, conversely, feasibility is preserved when the SAS is active (solid line). Similar transients are obtained for joint 6. As stated in §4.4, this result is achieved with the aid of the nonlinear filters which keep the first and the second time derivatives of $\mathbf{\Phi}_{\tilde{T},T}$ between the equivalent bounds evaluated by the EBE. This assertion can be verified through Figs. 4.4 and 4.5, which compare components α and β of $\mathbf{\Phi}_{\tilde{T},T}$ with the homologous components $\hat{\alpha}$ and $\hat{\beta}$ of $\mathbf{\Phi}_{\hat{T},T}$. Transients related to the third component of $\mathbf{\Phi}_{\tilde{T},T}$, i.e. γ, have been omitted since for all the considered tests the SAS always imposes $\gamma = 0$. In non critical regions, $\mathbf{\Phi}_{\hat{T},T}$ is kept equal to zero, so that the manipulator is driven by the nominal trajectory. Conversely, when the SD detects a singularity, the OM block modifies the tool orientation by generating a proper signal $\mathbf{\Phi}_{\hat{T},T}$ (see the dashed lines of Figs. 4.4 and 4.5). Such signal is generally discontinuous and unfeasible w.r.t. the given joint limits. For this reason, it is filtered by the NFS which generates a smooth

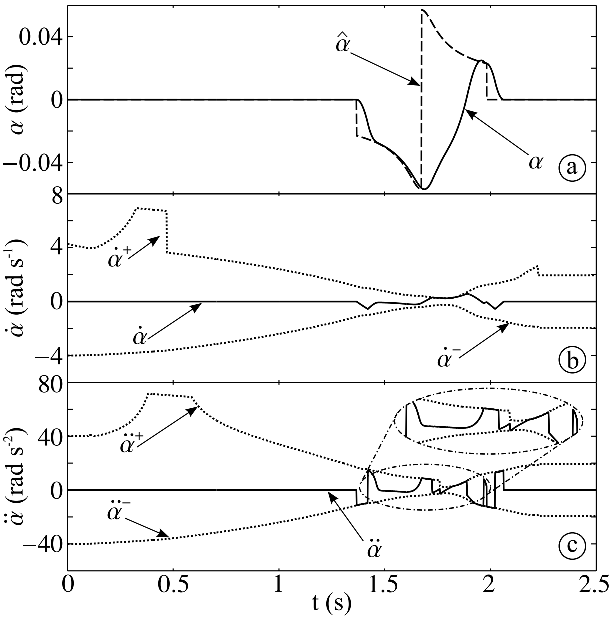

Figure 4.4: Time shapes of α, $\dot{\alpha}$, and $\ddot{\alpha}$ (solid lines). Dotted lines represent the equivalent bounds on $\dot{\alpha}$ and on $\ddot{\alpha}$. Dashed line $\hat{\alpha}$ represents the non-smooth signal provided by the OM.

signal $\boldsymbol{\Phi}_{\tilde{T},T}$ (solid lines) whose first and second time derivatives fulfill equivalent bounds $\dot{\boldsymbol{\Phi}}^-, \ddot{\boldsymbol{\Phi}}^-$ and $\dot{\boldsymbol{\Phi}}^+, \ddot{\boldsymbol{\Phi}}^+$, thus preserving the trajectory feasibility.

The effectiveness of the proposed method depends on the distance between the trajectory and the singular point: When trajectories pass close to the singular configuration more orientation changing effort is required in order to maintain the system feasibility. To this purpose, a different test set has been conceived to verify the system performances for trajectories passing at different distances from $\overline{\mathbf{p}}_T$: It has been generated a set of linear paths from $\mathbf{p}_A = [0.65\ 0.83\ z]^T$ m to $\mathbf{p}_B = [-0.2\ 0.83\ z]^T$ m for $z \in [1.05, 1.12]$ m and executed with a longitudinal velocity equal to 0.4 m s^{-1}. Fig. 4.6 shows the achieved shapes for α and β (also in all of this transients, the SAS has chosen $\gamma = 0$). As expected, for trajectories passing closer to $\overline{\mathbf{p}}_T$, the SAS introduces higher orientation changes. For the considered longitudinal velocity, interval $z \in [1.065, 1.092]$ m is still precluded (remember that singularity occurs at

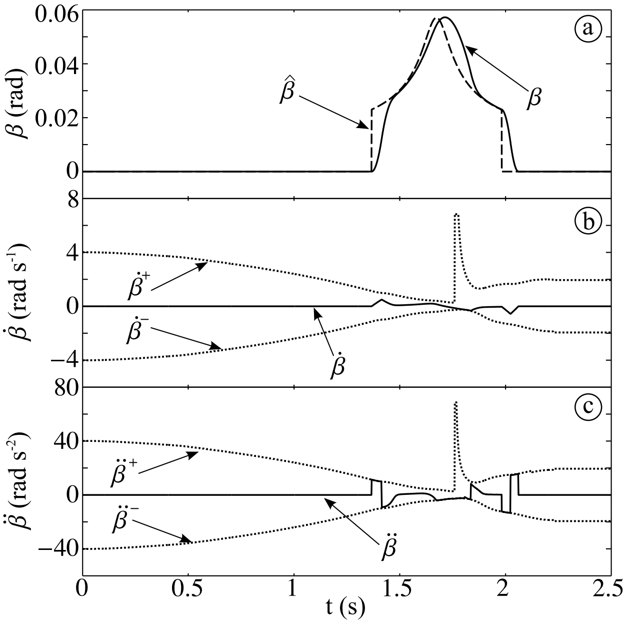

Figure 4.5: Time shapes of β, $\dot{\beta}$, and $\ddot{\beta}$ (solid lines). Dotted lines represent the equivalent bounds on $\dot{\beta}$ and on $\ddot{\beta}$. Dashed line $\hat{\beta}$ represents the non-smooth signal provided by the OM.

$z = 1.077$ m) but for lower velocities (0.1 m s^{-1}) the singular point can even be crossed.

The last set of tests has been conceived to verify the SAS performances for trajectories approaching $\overline{\mathbf{p}}_T$ from different directions. These tests follow the same procedure highlighted in §3.4 for the evaluation of the behavior of the TSS. It is worth to synthetically resume the singularity approaching procedure, in particular with the aid of Fig. 3.7: For each direction, a sequence of parallel trajectories lying in the plane $y = 0.83$ m and converging toward $\overline{\mathbf{p}}_T$ has been executed. All trajectories are characterized by the same longitudinal velocity and the distance between two of them is equal to $1 \cdot 10^{-3}$ m. For each direction the scanning process ends when feasibility is lost: The minimum distance from $\overline{\mathbf{p}}_T$ of the last feasible trajectory is stored and subsequently shown in Fig. 4.7, where the performances that can be obtained with the SAS are compared with those obtained without the SAS: the latter performances have been obtained with the aid of the commercial controller (as the same manner

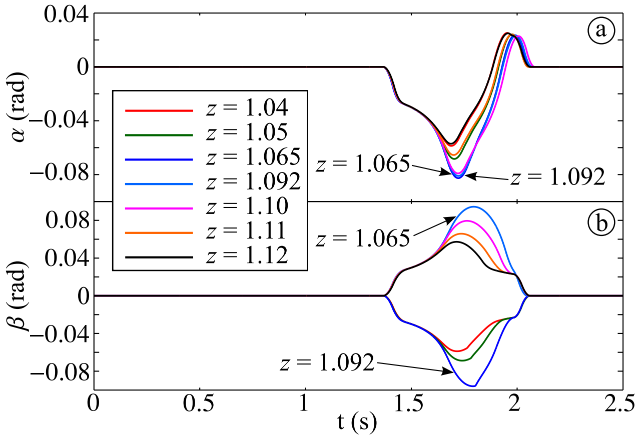

Figure 4.6: Orientation modification in function of the distance from the singularity point. Larger deviations are obtained for trajectories closer to $\overline{\mathbf{p}}_T$.

as points indicated with (1) in Fig. 3.8). The acquired results have highlighted that SAS performances are influenced by the approaching direction but, in any case, the SAS has been permitted to plan feasible trajectory closer to $\overline{\mathbf{p}}_T$ with respect to the performances of the commercial controller: Improvements are especially evident for vertical ($m = \infty$) and horizontal trajectories ($m = 0$). As shown in Fig. 4.7, similar results have been achieved independently from the longitudinal velocity.

It is worth to mention a final consideration about the proposed methods for the singularity management: Although the intent of the SAS and the TSS was different, it is important to emphasize that both strategies have been allowed to generate feasible trajectories that are closer to $\overline{\mathbf{p}}_T$ with respect to the behavior of the commercial controller.

4.6 Experimental results

The SAS performances have been finally checked by means of a set of actual experiments executed on a Comau Smart SiX 6-1.4 manipulator. They concern the execution of vertical and horizontal trajectories lying in the plane $y = 0.83$ m and passing close to a wrist singularity. More precisely, vertical trajectories pass $5 \cdot 10^{-3}$ m far

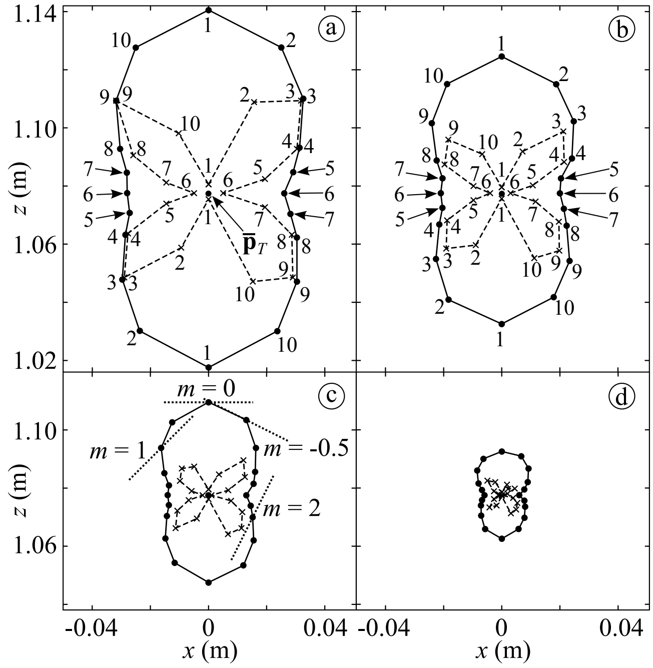

Figure 4.7: Minimum achievable feasible distance from the singularity point with the SAS (dashed lines) or without it (solid lines). Results achieved for a longitudinal velocity equal to: (a) 0.4 ms^{-1}; (b) 0.3 ms^{-1}; (c) 0.2 ms^{-1}; (d) 0.1 ms^{-1}. Slopes of the linear segments: (1) m=0; (2) m=-1/2; (3) m=-1; (4) m=-2; (5) m=-4; (6) $m = \infty$; (7) m=4; (8) m=2; (9) m=1; (10) m=1/2.

from the singularity, while horizontal ones pass at $3.5 \cdot 10^{-3}$ m: In both cases they represent the minimum-distance trajectories that can be safely executed by means of the SAS for a maximum longitudinal velocity equal to 0.4 m s^{-1}. The trajectory updating time is equal to $2 \cdot 10^{-3}$ s and the performances are evaluated on the basis of the path tracking errors in the operational space. Those errors is mainly due to the system dynamics and depend on the controller characteristics and on the elasticity of the structure. They can be mitigated by improving the controller performances or by using smooth reference signals. Clearly, the trajectory planner here proposed can only provide smooth references since, as stated in §4.1, control loops have not been accessible.

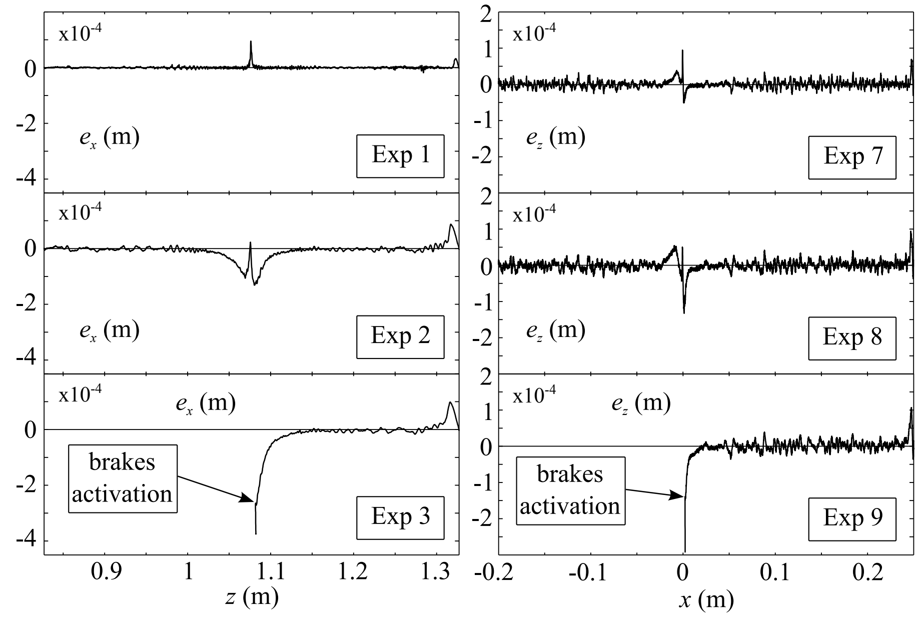

Figure 4.8: Path tracking errors for vertical trajectories (Experiments 1, 2, and 3) and for horizontal trajectories (Experiments 7, 8, and 9). In all of these experiments the SAS has been disabled.

The experiments have been executed with different longitudinal speeds. In Experiment 1 and Experiment 7 a vertical and an horizontal trajectories have been executed by assuming a longitudinal speed equal to 0.01 m s^{-1} and with the SAS disabled. Fig. 4.8 shows the path tracking error along the x axis for Experiment 1 and along the z axis for Experiment 7. The tracking errors along the y and z axis for vertical trajectories and along the x and y axis for horizontal ones have not been reported being negligible. The main purpose of these two experiments have been sought in the commercial controller performances. Due the extremely low velocity, the path tracking error, calculated from the motors' encoders through the forward kinematics, can be neglected far from the singular value but, in the proximity of the singularity, it has roughly assumed $1 \cdot 10^{-4}$ m for both the experiments. Those experimental results have been highlighted that the controller tracking performances worsen as soon as joint velocities rise.

In Experiments 2 and 8 trajectories are executed with a longitudinal speed respec-

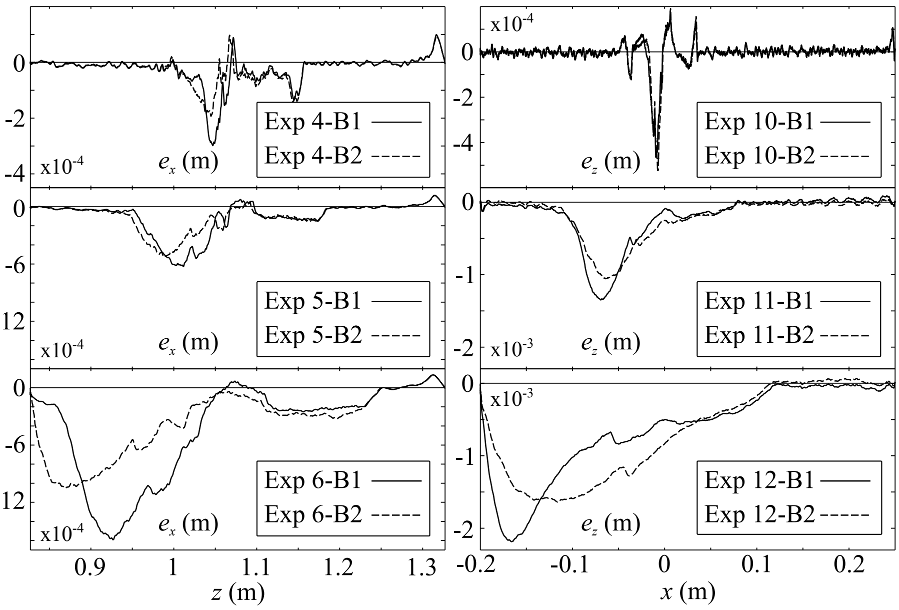

Figure 4.9: Path tracking errors for vertical trajectories (SAS enabled) for Experiments 4, 5, and 6. Solid lines represent errors obtained with bounds B1, while dashed lines indicate errors acquired with bounds B2.

tively equal to 0.07 m s^{-1} and 0.02 m s^{-1}, which represent the maximum velocities that can be feasibly maintained without the SAS assistance. Then, in Experiments 3 and 9, still keeping the SAS disabled, they are further increased to 0.1 m s^{-1} and to 0.03 m s^{-1} respectively: Joint speeds become unfeasible, so that the manipulator is stopped by the controller. In Experiments 4, 5, 6, 10, 11, and 12 the SAS has been activated. For each experiment two alternative sets of bounds have been considered ($i = 1, 2, \ldots, 6$):

B1) $\underline{\dot{q}}_i = -8$ rad s^{-1}, $\bar{\dot{q}}_i = 8$ rad s^{-1}, $\underline{\ddot{q}}_i = -40$ rad s^{-2}, and $\bar{\ddot{q}}_i = 40$ rad s^{-2};
B2) $\underline{\dot{q}}_i = -10$ rad s^{-1}, $\bar{\dot{q}}_i = 10$ rad s^{-1}, $\underline{\ddot{q}}_i = -25$ rad s^{-2}, and $\bar{\ddot{q}}_i = 25$ rad s^{-2}.

Specifically, B2 is characterized by higher velocity constraints and lower acceleration limits with respect to the B1 set. The use of two alternative sets of bounds makes it possible to point out that, with a proper choice of the SAS constraints, it is possible to improve the controller tracking performances. Practically, when B2 is used, trajectories require lower torques, so that closed-loop controllers are less solicited and track-

Table 4.1: Experimentally measured maximum path tracking errors and maximum angular deviations from the nominal path.

	vertical trajectories				horizontal trajectories			
	Exp.	long. speed $[\mathrm{m\ s^{-1}}]$	max error $\times 10^{-4}$ [m]	max $\lvert\alpha\rvert$, $\lvert\beta\rvert$ [deg]	Exp.	long. speed $[\mathrm{m\ s^{-1}}]$	max error $\times 10^{-4}$ [m]	max $\lvert\alpha\rvert$, $\lvert\beta\rvert$ [deg]
SAS off	1	0.01	0.958	–	7	0.01	0.973	–
	2	0.07	1.31	–	8	0.02	1.32	–
	3	0.1	2.76	–	9	0.03	1.93	–
SAS on	4-B1	0.1	2.99	1.04	10-B1	0.03	4.65	0.391
	4-B2	0.1	1.93	1.04	10-B2	0.03	5.28	0.391
	5-B1	0.2	6.28	3.00	11-B1	0.2	13.5	2.86
	5-B2	0.2	5.26	2.99	11-B2	0.2	10.6	3.02
	6-B1	0.4	15.9	6.58	12-B1	0.4	21.9	4.57
	6-B2	0.4	10.5	6.22	12-B2	0.4	16.4	5.07

ing errors reduce. This characteristic is particularly evident at high speeds as shown by Fig. 4.9 and it is also confirmed by the data reported in Table 4.1. In all experiments, trajectories are feasible w.r.t. the given limits. For vertical paths the maximum achievable velocity increases from 0.07 $\mathrm{m\ s^{-1}}$ to 0.4 $\mathrm{m\ s^{-1}}$ with a maximum angular deviation of the end-effector equal to 6.22 deg, while for horizontal paths it passes from 0.02 $\mathrm{m\ s^{-1}}$ to 0.4 $\mathrm{m\ s^{-1}}$ with a maximum angular deviation equal to 5.07 deg: Orientation changes are evidently compatible with those admissible for the industrial applications considered at the beginning of this chapter.

The computational burden introduced by the whole SAS execution is in range of between $0.163 \cdot 10^{-3}$ s and $1.373 \cdot 10^{-3}$ s with an average computational time equal to $0.254 \cdot 10^{-3}$ s. These performances have been evaluated on a RTAI [30] based PC equipped with an Intel Core 2 Duo E8400 @ 3.00 GHz processor. The SAS is fully compatible with real-time implementation. By comparing these execution times with those highlighted in §4.4, i.e. computational times that are required for the resolution of the optimization problem, it is evident that a large part of the time is spent by the

EBE block computation.

In conclusion, it is important to remark that the orientation changes introduced by the SAS behavior are feasible with respect to the bounds on joint velocities and accelerations and are sufficiently smooth in order to avoid undesired oscillations and mechanical stresses. Path tracking errors, measured on the actual manipulator, were proportional to the longitudinal speed and were mainly due to control loops' performances.

Conclusion and future studies

This thesis has introduced and proposed solutions to the challenging problem of real-time trajectory generation for systems subject to high order kinematic and dynamic constraints. Proposed problems have been addressed by means of specifically devised feedback schemes able to modify, in real-time, a provisional and potentially non-feasible trajectory in order to fulfill a given set of constraints. Chapters 2 and 3 have introduced the trajectory scaling problem: By assuming the path-velocity decomposition paradigm for the trajectory planning phase (i.e. the trajectory is obtained from the composition of a geometric path and of a time-law), longitudinal velocity has been modified online in order to preserve accurate path tracking and to fulfill the physical limits of the mechatronic system under investigation: Chapter 2 has considered trajectories planned in the configuration space while Chapter 3 has been focused on trajectories planned in the Cartesian space. An alternative planner for trajectories defined in the Cartesian space has been discussed in Chapter 4 where a Singularity Avoidance System (SAS) has been devised. Unlike the scaling systems, the SAS slightly modifies the orientation of planned trajectories in order to maintain joint velocities and accelerations confined within the robot's physical limits. The SAS assumes that the user-defined longitudinal velocity is "mandatory" for correct task execution.

The choice of the most appropriate solution depends on the manipulator task. Several applications exist where it is possible to slow down the longitudinal velocity in order to maintain accurate path tracking such as, for example automated drawing or cutting applications: These applications do not admit deviations from the planned path. Conversely, if the execution time and the Cartesian path preservation are mandatory, small changes in the tool orientation can be tolerated in order to fulfill the system's physical limits. Applications like automatic painting, gluing or welding are typical examples in which alterations of the longitudinal speed would cause task quality losses.

The scaling system proposed in Chapter 3 has been successfully tested in an experimental setup based on an actual 6R anthropomorphic manipulator. The experimental tests were conceived in order to compare two different implementations of TSSs: A first version that manages bounds on velocities and accelerations (AC-TSS) and another one that is also able to constrain jerks (JC-TSS). The bounds have been defined both in the joint and in the Cartesian spaces. The experimental results have proven that trajectories with bounded jerks ensure smoother movements with respect to those achievable with the AC-TSS. Conversely, the AC-TSS is more reactive in the vicinity of singular configurations. It is important to mention that the evaluation of the equivalent bounds used for the definition of the time-law also requires the evaluation of the inverse Jacobian. As a consequence, considered paths cannot directly cross singular points: Such configurations can only be managed by admitting minor path modifications, as has been done in the solution provided in Chapter 4.

The SAS presented in Chapter 4 has been tested both in a simulated and experimental setup. The simulation results have proven the effectiveness of the proposed strategy, which is able to handle singular points also in the event of high-speed trajectories by admitting a maximum tool orientation variation of 6 deg for high longitudinal speeds (0.4 m s^{-1}). The experimental setup, based on a 6R anthropomorphic manipulator, has highlighted negligible path tracking errors. In the event of slow motions, the SAS is able to generate feasible trajectories even where paths directly cross singular configurations.

It is important to note that the computational burden of the proposed planners is compatible with real-time implementation. In particular, the execution time of the AC-TSS is in the range of between $7 \cdot 10^{-6}$ s and $142 \cdot 10^{-6}$ s with an average time equal to $35 \cdot 10^{-6}$ s. Conversely, the execution time of the JC-TSS is in the range of between $11 \cdot 10^{-6}$ s and $153 \cdot 10^{-6}$ s with an average time computational equal to $49 \cdot 10^{-6}$ s. Since it must also consider constraints on the jerk signal, the JC-TSS is characterized by greater computational burdens, which are, however, still compatible with the sampling times of many control systems. The computational burden introduced by the SAS execution is in the range of between $163 \cdot 10^{-6}$ s and $1373 \cdot 10^{-6}$ s with an average evaluation time equal to $254 \cdot 10^{-6}$ s: The computational load is still compatible with that admissible for real-time implementation.

Recommendations for future studies

Several research themes have been left open by this thesis. In particular, concerning the discrete-time filter used for the generation of scalar smooth profiles, it would be interesting to address the problem of the synchronization of several of these in order to plan multidimensional trajectories for robotic applications. The problem can be summarized as follows: If two or more trajectories must be synchronized in order to obtain complex cooperative motions, i.e. consider the case of multiple electric cams, it is necessary to scale the fastest of them in order to guarantee the simultaneous end of the motion. A possible solution to this issue could be to provide synchronization constraints between the sliding surfaces that govern the filters.

As far as the TSSs are concerned, an interesting improvement can be certainly be the introduction of the generalized force and the force derivative constraints in the solutions provided for paths planned in the operational space. This would require a complete reformulation of the equivalent bounds evaluator procedure and more in-depth knowledge of the controlled system model: Accurate dynamic models must be developed and their parameters must be identified. Such models must then be integrated into the equivalent bounds evaluation blocks.

Another research field, still related to the TSS implementation, concerns the inversion of the Jacobian matrix. As it is well-known, Jacobians become singular close to singularities, so that they cannot be inverted. It could be interesting to verify which path tracking performances can be achieved when using an approximated Jacobian inverse. If the performances do not deteriorate excessively and remain compatible with the task guidelines, a solution that permits crossing singular configurations could certainly represent a major improvement for the proposed TSSs.

Possible improvements in the SAS strategy concern the efficient evaluation of the best orientation modification and the reduction of the minimum achievable distances from singular points. For the first open problem, research could focus on the investigation of a novel method for the synthesis of the most promising orientation: Methods based on the manipulability index are computationally onerous, so that suboptimal, but more efficient, solutions should be used in order to deal with this issue. The minimum achievable distances from singular points are evidently correlated to the longitudinal velocity, but they are also somehow correlated to the shape of the manipulability ellipsoid: The performances of the SAS also depend on the singular point approaching directions. It could be interesting to study changing behaviors of the SAS, chosen depending on the approaching directions.

Another research branch could try to unify the SAS and the TSS in a common framework. A general approach to singularity management could be used to solve a wide variety of trajectory planning problems, since a hybrid solution, obtained from the combination of the two methods, could potentially benefit from the advantages of both: Path feasibility could be preserved also in trajectories that pass very close to or, even, cross singular configurations. Evidently, the hybrid approach could only be used in applications that admit variations of both the longitudinal speed and the planned orientation.

Bibliography

[1] C.-S. Lin, P.-R. Chang, and J.Y.S. Luh. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Trans. Automatic Control*, AC-28(12):1066–1074, 1983.

[2] G. Lini, L. Consolini, and A. Piazzi. Minimum-time constrained velocity planning. In *Med. Conf. on Contr. and Autom., MED09*, pages 748–753, 2009.

[3] A. De Luca, L. Lanari, and G. Oriolo. A sensitivity approach to optimal spline robot trajectories. *Automatica*, 27(3):535–539, 1991.

[4] S. Engleder. Time-optimal motion planning and control of an electrohydraulically actuated toggle mechanism . *Mechatronics*, 17(8):448–456, 2007.

[5] S. Macfarlane and E. A. Croft. Jerk-bounded manipulator trajectory planning: design for real-time applications. *IEEE Trans. on Rob. and Autom.*, 19(1):42–52, 2003.

[6] L. Messner, H. Gattringer, and H. Bremer. Efficient Online Computation of Smooth Trajectories Along Geometric Paths for Robotic Manipulators . In H. Gattringer and J. Gerstmayr, editors, *Multibody system dynamics, robotics, and control*. Springer-Verlag, Wien, 2013.

[7] R. Haschke, E. Weitnauer, and H. Ritter. On-line planning of timeoptimal, jerk-limited trajectories. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 08*, pages 3248–3253, 2008.

[8] T. Kröger and F. M. Wahl. On-Line Trajectory Generation: Basic Concepts for Instantaneous Reactions to Unforeseen Events. *IEEE Trans. on Rob.*, 26(1):94–111, Feb. 2010.

[9] X. Broquère, D. Sidobre, and I. Herrera-Aguilar. Soft motion trajectory planner for service manipulator robot. In *Proc. of the 2008 IEEE/RSJ Int. Conf. on Intell. Rob. and Systems, IROS 08*, pages 2808–2813, 2008.

[10] C. Guarino Lo Bianco, A. Tonielli, and R. Zanasi. Nonlinear trajectory generator for motion control systems. In *IECON96 - 22nd Int. Conf. of the IEEE Ind. Electr. Society*, pages 195–201, Taiwan, Taipei, August 1996.

[11] Y. Su and P.C. Mueller. Smooth reference trajectory generation for industrial mechatronic systems under torque saturation. In *4th IFAC Symp. on Mechatr. Sys.*, volume 4 - PART 1, pages 896–901, Heidelberg, Germany, Sept. 2006.

[12] X. Wei, J. Wang, and Z. Yang. Robust Smooth-Trajectory Control of Nonlinear Servo Systems Based on Neural Networks. *IEEE Trans. Ind. Electr.*, 54(1):208–217, 2007.

[13] L. Biagiotti and R. Zanasi. Online trajectory planner with constraints on velocity, acceleration and torque. In *Proc. of the IEEE Int. Symp. on Ind. Electr., ISIE2010*, pages 274–279, Bari, Italy, Jul. 2010.

[14] R. Zanasi, C. Guarino Lo Bianco, and A. Tonielli. Nonlinear filters for the generation of smooth trajectories. *Automatica*, 36(3):439–448, March 2000.

[15] C. Guarino Lo Bianco and R. Zanasi. Smooth profile generation for a tile printing machine. *IEEE Trans. on Ind. Electr.*, 50(3):471–477, 2003.

[16] O. Gerelli and C. Guarino Lo Bianco. Nonlinear variable structure filter for the online trajectory scaling. *IEEE Trans. on Ind. Electr.*, 56(10):3921–3930, Oct. 2009.

[17] S. Liu. An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators. In *Proc. of the seventh Int. Work. on Advanced Motion Control*, pages 365–370, 2002.

[18] R. Zanasi and R. Morselli. Third order trajectory generator satisfying velocity, acceleration and jerk constraints. In *Proc. of the IEEE Int. Conf. on Contr. Appl.*, pages 1165–1170, Glasgow, UK, Sept. 2002.

[19] L. Biagiotti and C. Melchiorri. *Trajectory planning for automatic machines and robots*. Springer, Berlin, Heidelberg, Germany, 2008.

[20] R. Zanasi and R. Morselli. Discrete minimum time tracking problem for a chain of three integrators with bounded input. *Automatica*, 39(9):1643–1649, 2003.

[21] O. Gerelli and C. Guarino Lo Bianco. A discrete-time filter for the on-line generation of trajectories with bounded velocity, acceleration, and jerk. In *IEEE Int. Conf. on Rob. and Autom., ICRA2010*, pages 3989–3994, Anchorage, AK, May 2010.

[22] C. Guarino Lo Bianco and F. Ghilardelli. Third order system for the generation of minimum-time trajectories with asymmetric bounds on velocity, acceleration, and jerk. In *Workshop on Robot Motion Planning: Online, Reactive, and in Real-time 2012 IEEE/RSJ Int. Conf. on Int. Rob. and Sys., (IROS 2012)*, pages 137–143, Algarve, Portugal, Oct 2012.

[23] C. Guarino Lo Bianco and O. Gerelli. Online trajectory scaling for manipulators subject to high-order kinematic and dynamic constraints. *IEEE Trans. on Rob.*, 27(6):1144–1152, December 2011.

[24] D. Kubus, D. Inkermann, T. Vietor, and F. M. Wahl. Joint Actuation Based on Highly Dynamic Torque Transmission Elements – Concept and Control Ap-

proaches. In *Proc. of IEEE Int. Conf. on Rob. and Autom., ICRA 2011*, pages 2777–2784, Shanghai, China, May 2011.

[25] C. Guarino Lo Bianco and F. Ghilardelli. A Discrete-Time filter for the generation of signals with asymmetric and variable bounds on velocity, acceleration, and jerk. *IEEE Trans. on Ind. Electr.*, 61(8):4115–4125, Aug 2014.

[26] V. I. Utkin. Variable structure systems with sliding modes. *IEEE Trans. on Autom. Contr.*, 22:212–222, 1977.

[27] V. I. Utkin. *Sliding Modes in Control and Optimization*. Springer-Verlag, New York, NY, 1992.

[28] R. Morselli. *Nonlinear trajectory generators for motion control systems*. PhD thesis, 2003.

[29] C. Guarino Lo Bianco and F.M. Wahl. A novel second order filter for the real-time trajectory scaling. In *IEEE Int. Conf. on Rob. and Autom., ICRA 2011*, pages 5813–5818, Shanghai, China, May 2011.

[30] R. Bucher and S. Balemi. Rapid Controller Prototyping with Matlab/Simulink and Linux. In *6th IFAC Symp. on Adv. in Control Educ., ACE2003*, Oulu, Finland, Jun. 2003.

[31] J. M. Hollerbach. Dynamic scaling of manipulator trajectories. *J Dyn Sys Meas Control*, 106(1):102–106, 1984.

[32] A. De Luca and R. Farina. Dynamic scaling of trajectories for robots with elastic joints. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2436–2442, Washington, DC, May 2002.

[33] J. Kieffer, A. Cahill, and M. James. Robust and accurate time-optimal path-tracking control for robot manipulators. *IEEE Trans Robot Automat*, 13(6):880–890, 1997.

[34] K. Kant and S.W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *Int. J. Robot. Res.*, 5(3):72–89, 1986.

[35] B. K. Kim and K. G. Shin. Minimum-time path planning for robot arms and their dynamics. *IEEE Trans. on Systems, Man, and Cybernetics*, 15(2):213–223, Mar/Apr 1985.

[36] D. Constantinescu and E. A. Croft. Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *J. Robot. Syst.*, 17(5):233–249, 2000.

[37] K. G. Shin and N. D. McKay. A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE Trans. on Automatic Control*, AC-31(6):491–500, June 1986.

[38] H. H. Tan and R. B. Potts. Minimum time trajectory planner for the discrete dynamic robot model with dynamic constraints. *IEEE J. of Robotics and Automation*, 4(2):174–185, April 1988.

[39] J. E. Bobrow, S. Dubowsky, and J .S. Gibson. Time-optimal control of robotics manipulators along specified paths. *Int. J. Robot. Res.*, 4(3):3–17, 1985.

[40] K. G. Shin and N. D. McKay. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, 30(6):531–541, Jun. 1985.

[41] F. Pfeiffer and R. Johanni. A concept for manipulator trajectory planning. *IEEE J. of Robotics and Automation*, RA-3(2):115–123, 1987.

[42] Z. Shiller and S Dubowsky. Robot path planning with obstacles, actuator, gripper, and payload constraints. *The Int'l J. of Robotics Research*, 8(6):3–18, 1989.

[43] R. Zanasi, A. Tonielli, and C. Guarino Lo Bianco. Nonlinear filter for smooth trajectory generation. In *NOLCOS98 - 4th Nonlinear Control Systems Design Symp.*, volume 1, pages 245–250, Enschede, the Netherlands, July 1998.

[44] C. Guarino Lo Bianco, F. Ghilardelli, and D. Kubus. Experimental validation of a time scaling algorithm for robotics systems. In *IEEE Int. Conf. on Rob. and Biomim. (ROBIO2012)*, pages 2044–2049, 2012.

[45] C. Guarino Lo Bianco and F. Ghilardelli. Techniques to preserve the stability of a trajectory scaling algorithm. In *IEEE Int. Conf. on Rob. and Autom. (ICRA2013)*, pages 870–876, 2013.

[46] O. Dahl and L. Nielsen. Torque-limited path following by online trajectory time scaling. *IEEE Trans Robot Automat*, 6(5):554–561, 1990.

[47] T. Yoshikawa. Analysis and control of robot manipulators with redundancy. In *1st Int. Symp. Robotics Research*. MIT press, 1984.

[48] J. M. Hollerbach and K. Suh. Redundancy resolution of manipulators through torque optimization. *IEEE Jou. of Robotics and Automation*, 3(4):308–316, Aug 1987.

[49] K.J. Kyriakopoulos and G.N. Saridis. Minimum jerk path generation. In *IEEE Int. Conf. on Rob. and Autom., ICRA'88*, pages 364–369 vol.1, apr 1988.

[50] C. Guarino Lo Bianco and F. Ghilardelli. Real-time planner in the operational space for the automatic handling of kinematic constraints. *IEEE Trans. on Autom. Sci. and Eng.*, 11(3):730 – 739, 2014.

[51] L. Sciavicco, B. Siciliano, L. Villani, and G. Oriolo. *Robotics: Modelling, planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer-Verlag, Berlin, Germany, 2011.

[52] D.C. Conner, A. Greenfield, P.N. Atkar, A.A. Rizzi, and H. Choset. Paint deposition modeling for trajectory planning on automotive surfaces. *IEEE Trans. on Autom. Sci. and Eng.*, 2(4):381–392, Oct 2005.

[53] P. Hertling, L. Hog, R. Larsen, J.W. Perram, and H.G. Petersen. Task curve planning for painting robots. I. Process modeling and calibration. *IEEE Trans. on Rob. and Autom.*, 12(2):324–330, Apr 1996.

[54] J. Peng, Q. Chen, J. Lu, J. Jin, and C.A. van Luttervelt. Real time optimization of robotic arc welding based on machine vision and neural networks. In *The 1998 IEEE Int. Conf. on Industrial Electronics, Control, and Instrumentation, IECON'98*, pages 1279–1283, 1998.

[55] S.-H. Suh, I.-K. Woo, and S.-K. Noh. Development of an Automatic Trajectory Planning System (ATPS) for spray painting robots. In *IEEE Int. Conf. on Rob. and Autom., ICRA91*, pages 1948–1955, 1991.

[56] R. Ramabhadran and J. K. Antonio. Fast solution techniques for a class of optimal trajectory planning problems with applications to automated spray coating. *IEEE Trans. on rob. and aut.*, 13(4):519–530, 1997.

[57] V. Potkonjaka, G. S. Dordević, D. Kostić, and M. Rašić. Dynamics of anthropomorphic painting robot: Quality analysis and cost reduction. *Robotics and Autonomous Systems*, 32(1):17–38, 2000.

[58] P.J. From and J.T. Gravdahl. A Real-Time Algorithm for Determining the Optimal Paint Gun Orientation in Spray Paint Applications. *IEEE Trans. on Autom. Science and Eng.*, 7(4):803–816, 2010.

[59] P.J. From, J. Gunnar, and J.T. Gravdahl. Optimal Paint Gun Orientation in Spray Paint Applications - Experimental Results. *IEEE Trans. on Autom. Science and Eng.*, 8(2):438–442, Apr. 2011.

[60] W. Tillmann, E. Vogli, and B. Krebs. Influence of the Spray Angle on the Characteristics of Atmospheric Plasma Sprayed Hard Material Based Coatings. *J. of Thermal Spray Tech.*, 17(5-6):948–955, 2008.

[61] G. Schreiber, M. Otter, and G. Hirzinger. Solving the singularity problem of non-redundant manipulators by constraint optimization. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys., IROS'99*, volume 3, pages 1482–1488, 1999.

[62] W. Decrè, H. Bruyninckx, and J. De Schutter. Extending the iTaSC constraint-based robot task specification framework to time-independent trajectories and

user-configurable task horizons. In *IEEE Int. Conf. on Rob. and Autom. (ICRA2013)*, pages 1941–1948, 2013.

[63] T. Yoshikawa. Manipulability of robot mechanisms. *Int. J. of Robotics Research*, 4(2):3–9, 1985.

[64] R.J. Vanderbei. *Linear Programming: Foundations and Extensions (fourth edition)*. International Series in Operations Research & Management Science. Springer, New York, 2014.

Acknowledgements

So, here I am! The last page of this thesis and the last day of my student career. Now, it is time for acknowledgements.

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Corrado Guarino Lo Bianco for his continuous and essential guidance, for all the inspiring discussions we had, and for the help with scientific and non-scientific problems.

Besides my supervisor, I would like to thank Prof. Marco Locatelli for his immense knowledge in the optimization topic. I would also like to thank the head of the Control Group, Prof. Aurelio Piazzi, for his encouragement. Then, I have to express my gratitude to Prof. Stefano Caselli, head of the Robotics and Intelligent Machines Laboratory, for giving me the opportunity to test my researches on actual mechatronic systems.

My sincere thanks also goes to all the other colleagues I had the pleasure to share this experience with, in particular my office mate: I surely miss the endless phone calls of Mario Sabbatelli and the always interesting conversations with Fabio Oleari.

I would remember the fantastic experience that I have done at the Technischen Universität of Braunschweig, Germany. In particular, I wish to thank the head of the Institut für Robotik und Prozessinformatik, Prof. Friedrich M. Wahl, for the opportunity he gave me to work with his wonderful group. I also thank friends Daniel Kubus and Dirk Buchholz for making me feel at home every day during my stay in Germany.

Last, and most important, a huge thanks goes to my family and, in particular, to my parents, for their endless support and their patience. To them I dedicate this thesis.

Fabio