Time-optimal regulation of a chain of integrators with saturated input and internal variables: an application to trajectory planning

Luigi Biagiotti * **Roberto Zanasi** *

* *DII, University of Modena and Reggio Emilia*
41100 Modena, Italy
(e-mail: luigi.biagiotti@unimore.it,roberto.zanasi@unimore.it)

Abstract: The design of a planner for time-optimal trajectories with constraints on velocity, acceleration, jerk, ..., is translated into a regulation problem for a chain of integrators with saturations not only in the input but also in all the internal (state) variables. Then the problem is solved by designing a regulator, based on the variable structure control, able to steer the state vector to the origin in minimum time, being compliant with all the constraints. For this purpose, a modular structure with a cascade of controllers, each one devoted to the regulation to the origin of a specific component of the state vector, is demonstrated to be effective and ideally suitable to cope with systems of any order. Analytical examples are provided for filters of first, second and third order.

Keywords: Trajectory planning; Optimal control; Switching Surfaces;

1. INTRODUCTION

The planning of time-optimal trajectories subject to kinematic constraints has been faced in a number of works, leading to both offline algorithms for the computation of trajectories typically based on dynamic programming, Shin and Mckay [1986], Singh and Leu [1987] or other different optimization methods, Bobrow et al. [1985], Lee and Lee [1997], and to on-line planners, able to compute in real-time the trajectory once that the desired final configuration (position, velocity, etc.) has been specified, Lo Bianco et al. [2000], Zanasi and Morselli [2002].
In particular, in Lo Bianco et al. [2000] a second order filter is proposed, able to generate trajectories with continuity of position and velocity profiles and with a discontinuous but limited acceleration, while in Zanasi and Morselli [2002] a filter of third order is built with the purpose of on-line planning trajectories continuous in position, velocity and acceleration (and with limited jerk). In this work the approach based on variable structure control with the computation of the sliding surface by backward integration (already applied in Lo Bianco et al. [2000], Zanasi and Morselli [2002]) is combined with a cascade structure of the controllers which provides a clear interpretation of the final controller and allows an immediate (although computationally complex) extension to filters of order higher than three. This means that, following the proposed approach, it is possible to design filters providing as output trajectories with an higher degree of smoothness (i.e. degree of continuity of the derivatives of the position profile), which in some applications may be necessary in order to avoid undesired effects, such as vibrations, see Lambrechts et al. [2005], Barre et al. [2005].

2. PROBLEM FORMULATION

The dynamic nonlinear filter of n-th order proposed in this paper, and shown in Fig. 1, generates on-line a time optimal trajectory $x(t)$ that tracks at best the reference signal $r(t)$, satisfying desired constraints on the first n derivatives of $x(t)$, i.e.

$$\underline{x}_i \leq x_i(t) \leq \overline{x}_i, \qquad i = 1, \ldots, n \tag{1}$$

where $x_i(t) = x^{(n-i)}(t) = \dfrac{d^{n-i}x(t)}{dt^{n-i}}$, $i = 1, \ldots, n$. Note that the constant parameters $\underline{x}_i < 0$, $\overline{x}_i > 0$ are in general not symmetric, that is $\underline{x}_i \neq -\overline{x}_i$. Moreover, they can sometimes modified on-line.
The reference signal $r(t)$ is generally given by a first rough trajectory generator providing for instance piece-wise constant profiles according to the task to be performed, or it is the result of an external input, such as the commands of a human operator. Obviously, the signal $r(t)$ can be actually tracked only if it is compliant with the constraints (1). Additionally, the hypothesis that the n-th derivative of $r(t)$ is null, i.e. $r^{(n)}(t) \equiv 0$, is assumed.
The filter is composed by a chain of n integrators and by a nonlinear controller able to nullify the tracking error in

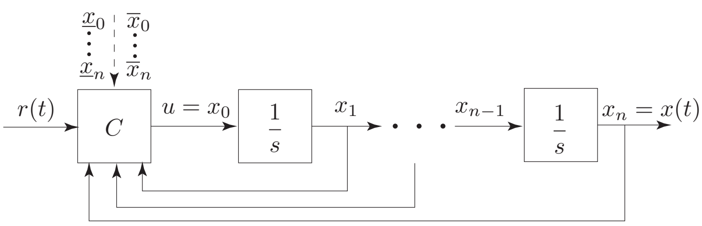


Fig. 1. Structure of a generic trajectory filter of n-th order.

minimum time, being compliant with the above mentioned constraints. Therefore, given the system

$$\begin{cases} \dot{x}_1 = u \\ \dot{x}_2 = x_1 \\ \quad \vdots \\ \dot{x}_n = x_{n-1} \end{cases} \tag{2}$$

or, with a more compact notation,

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u \tag{3}$$

with the state vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_{n-1}, x_n]^T = [x^{(n-1)}, x^{(n-2)}, \ldots, x^{(1)}, x]^T$, and the matrices

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & \cdots & & 0 \\ 1 & 0 & 0 & \cdots & & 0 \\ 0 & 1 & 0 & \cdots & & 0 \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

we define the tracking error as $\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{r}$, where $\boldsymbol{r} = [r_1, r_2, \ldots, r_{n-1}, r_n]^T = [r^{(n-1)}, r^{(n-2)}, \ldots, r^{(1)}, r]^T$. Because of the hypothesis $r^{(n)}(t) \equiv 0$, the dynamics of the error variable is equal to (3), i.e.

$$\dot{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{B}u. \tag{4}$$

The aim of the controller C is therefore to steer the vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_{n-1}, y_n]^T$ to the origin, guaranteeing that for any value of the time t

$$\underline{y}_i(t) \le y_i(t) \le \overline{y}_i(t), \qquad i = 1, \ldots, n-1 \tag{5}$$

where the limit values are computed as

$$\underline{y}_i(t) = \underline{x}_i - r_i(t), \qquad \overline{y}_i(t) = \overline{x}_i - r_i(t)$$

and therefore they are not constant since they depend on the reference input $r(t)$. It is worth noticing that because the input signal and its derivatives are compliant with constraints (1), i.e. $\underline{x}_i \le r_i(t) \le \overline{x}_i$, $i = 1\ldots, n$, the limits on y_i are such that $\underline{y}_i(t) \le 0$ and $\overline{y}_i(t) \ge 0$, $i = 1\ldots, n$, $\forall t$. Moreover, the additional constraint on the control action

$$\underline{y}_0 = \underline{x}_0 \le u(t) \le \overline{x}_0 = \overline{y}_0 \tag{6}$$

is assumed.

The regulation of (4) with a saturation on the control action, as in (6), has been coped and solved in different ways, see Teel [1992], Marchand and Hably [2005]. In particular, the time-optimal control of (4) is described in many textbooks on optimal control theory (at least for $n = 2$ and $n = 3$), see for instance Lee and Markus [1967] among many others. In this work, the chain of integrators must be regulated to zero by taking into account constraints not only on the control action $u(t)$ but also on all the intermediate variables $y_i(t)$. The solution has been found in designing a nonlinear controller obtained by nesting n controllers, each one devoted to nullify a specific element of the error vector $\boldsymbol{y}$ in minimum time. According to this design philosophy, the third order controller is built over the second order filter which is based on the first order filter. This structure can be easily iterated in order to obtain higher order trajectory generators.

3. CONTROLLERS DESIGN

All the controllers have the same general structure, illustrated in Fig. 2: the i-th controller acts on the control variable $u_i \in \{\, \underline{x}_{i-1},\ 0,\ \overline{x}_{i-1}\}$ to steer the state vector $\boldsymbol{x}_i = [x_1, \ldots, x_i]^T$ to $\boldsymbol{r}_i = [r_{i,1}, \ldots, r_{i,i}]^T$. Note that, the input vector $\boldsymbol{r}_i$ of the system S_{i-1} is of length i. Accordingly the controller C_i must generate a control output of the same size. In particular, the output of the i-th controller will be $\boldsymbol{u}_i = [0, \ldots, 0, u_i]^T$ where the term u_i is piecewise constant and accordingly its derivatives are null. In the following sections, with a little abuse of notation the output of the generic i-th controller is simple denoted by the scalar u_i. Moreover, for the sake of simplicity, in the generic section devoted to the filter of i-th order both the variables $r_{i,k}$ and $y_{i,k} = x_i - r_{i,k}$, $k = 1, \ldots, i$, components of $\boldsymbol{r}_i$ and $\boldsymbol{y}_i$ respectively, are simply denoted with r_k and $y_k = x_i - r_k$. However it is necessary to remember that r_1 or y_1 related to the first order filter are different from those defined for second or third order filters.

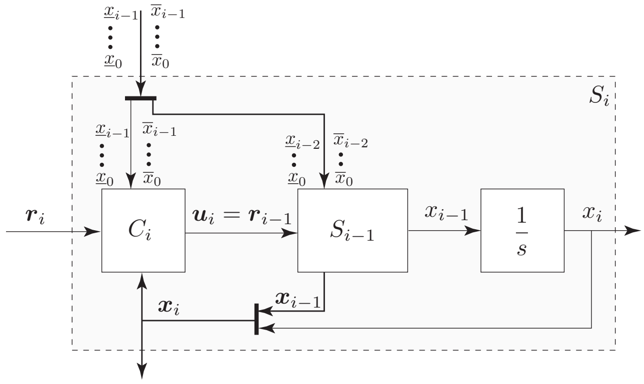


Fig. 2. Structure of the i-th control loop ($S_0 = 1$).

3.1 First order filter

Proposition 1. Given the system

$$\dot{x}_1 = u$$

composed by a single integrator, the control law

$$C_1: \quad u_1(r_1, x_1) = \begin{cases} \overline{x}_0, & \text{if } y_1 < 0 \\ 0, & \text{if } y_1 = 0 \\ \underline{x}_0, & \text{if } y_1 > 0 \end{cases} \tag{7}$$

satisfies the constraint $\underline{x}_0 \le x_0(t) \le \overline{x}_0$ (with $x_0(t) = u(t) = \dot{x}_1(t)$) and forces the state variable x_1 to reach the value $x_1 = r_1$ (r_1 constant) in minimum time.

Proof. The control signal is bounded by definition and therefore the variable x_0, related to x_1 by $\dot{x}_0 = x_1$, is $\underline{x}_0 \le x_0(t) \le \overline{x}_0$. The error variable $y_1 = x_1 - r_1$ is considered, whose dynamics is described by $\dot{y}_1 = u$. By applying the constant control $u_1(t) = \hat{u}_1$ ($\hat{u}_1 = \overline{x}_0$, if $y_1 < 0$ or $\hat{u}_1 = \underline{x}_1$, if $y_1 > 0$), the tracking error $y_1(t) = \int_{t0}^{t} u(t)dt = \hat{u}_1(t - t_0) + y_{1_0}$ tends to the origin at the maximum speed. When y_1 equals 0, the control action switches and becomes $u_1(t) = 0$.

Remark 2. By defining a special type of inverse sign function, i.e.

$$\text{ISgn}(\overline{x}, \underline{x}, \xi) = \begin{cases} \underline{x}, & \xi > 0 \\ 0, & \xi = 0 \\ \overline{x}, & \xi < 0. \end{cases}$$

the definition of the controller C_1 in (7) becomes

$$u_1(r_1, x_1) = \text{ISgn}(\overline{x}_0, \underline{x}_0, y_1).$$

3.2 Second order filter

The first order controller is now exploited to design a second order filter able to steer the state vector $[x_1, x_2]^T$ to the desired values $[r_1, r_2]^T$.

Proposition 3. Given the second order system composed by a cascade of two integrators and by the controller C_1:

$$\begin{cases} \dot{x}_1 = u_1(u_2, x_1) \\ \dot{x}_2 = x_1 \end{cases} \quad (8)$$

where u_2 is the new control input, the control law

$$C_2 : u_2(\boldsymbol{r}_2, \boldsymbol{x}_2) = \begin{cases} \overline{x}_1, & \text{if } y_2 < h_2(y_1) \text{ or } y_2 = h_2^-(y_1) \\ 0, & \text{if } y_2 = y_1 = 0 \\ \underline{x}_1, & \text{if } y_2 > h_2(y_1) \text{ or } y_2 = h_2^+(y_1) \end{cases} \quad (9)$$

with

$$h_2(y_1) = \begin{cases} h_2^-(y_1) = \dfrac{y_1^2}{2\overline{x}_0}, & \text{if } y_1 < 0 \\ 0, & \text{if } y_1 = 0 \\ h_2^+(y_1) = \dfrac{y_1^2}{2\underline{x}_0}, & \text{if } y_1 > 0 \end{cases} \quad (10)$$

satisfies the constraint $\underline{x}_1 \le x_1(t) \le \overline{x}_1$ (and obviously $\underline{x}_0 \le x_0(t) \le \overline{x}_0$) and forces the variable x_2 to reach the value r_2 (and x_1 the desired value r_1) in minimum time.

Proof. Since the second order filter is based on (7) the bounds on $x_0(t)$ are automatically guaranteed, while the compliance with the constraints on $x_1(t)$ is assured by the fact that the controller (9) provides a control output $u_2 \in [\underline{x}_1, \overline{x}_1]$ used as reference signal by the inner control loop which guarantees that such a value is never exceeded. We consider the dynamics of the error variables $y_1 = x_1 - r_1$, $y_2 = x_2 - r_2$, that, because of the relationship between $r_1(t)$ and $r_2(t)$, i.e. $\dot{r}_2(t) = r_1(t)$, and the hypothesis $\dot{r}_1(t) = 0$, becomes

$$\begin{cases} \dot{y}_1 = u_1(u_2, y_1 + r_1) \\ \dot{y}_2 = y_1. \end{cases} \quad (11)$$

The controller C_2 should ideally steer the error to the origin by means of two arcs of trajectory obtained with $u_2(t) = \overline{x}_1$ or $u_2(t) = \underline{x}_1$ according to the initial conditions $[y_{1_0}, y_{2_0}]^T$, and then $u_2(t) = r_1(t)$. In this way, the controller firstly applies an extremal control value in order to nullify the error variable y_2 in the fastest way, then it "brakes" by imposing $u_2(t) = r_1(t)$ with the purpose of stopping exactly on $y_2 = 0$. At the end of this second phase, the inner controller C_1 must assure that $x_1 - r_1 = y_1$ is equal to zero. In this manner at the end of motion the state $\boldsymbol{y}_2 = [y_1, y_2]^T$ is exactly on $\mathbf{0}$. By representing the system trajectories in the phase plane, it is straightforward to compute the switching locus, i.e. the curve $y_2 = h_2(y_1)$ of the phase plane where the control action must change. This curve can be obtained by backward integrating the system dynamics (11), composed by the filter S_1 and an additional integrator, from the origin by considering a step input u_2 of generic magnitude $\hat{u}_2$. When the input $\hat{u}_2$ is applied to C_1, it provides a constant control action $u_1 = \hat{u}_1$ ($\hat{u}_1 = \overline{x}_0$ or $\hat{u}_1 = \underline{x}_0$) that aims at reducing the error $x_1 - \hat{u}_2$. As a consequence the state of the system for $-t_1 \le t \le t_0 = 0$ is

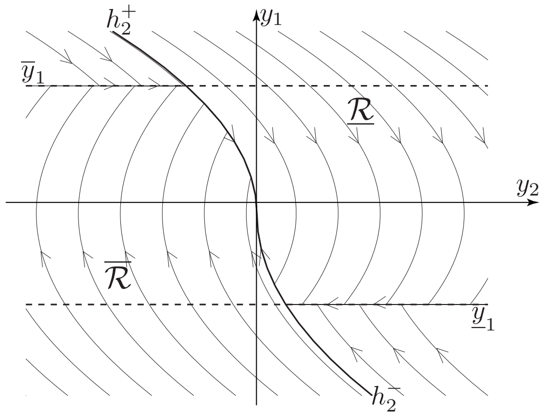


Fig. 3. Phase portrait of the second order filter.

$$\boldsymbol{y}_2 = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -\hat{u}_1 t \\ \dfrac{\hat{u}_1 t^2}{2} \end{bmatrix}. \quad (12)$$

Eliminating the parameter t from (12), one obtain the explicit expression of the trajectory passing through the origin [1], that is

$$y_2 = \frac{y_1^2}{2\hat{u}_1}. \quad (13)$$

The value of $\hat{u}_1$ in (12) and (13) depends on the reference signal applied to C_1 which coincides with the control $\hat{u}_2$ provided by C_2, and therefore with r_1, being (13) the last tract of the trajectory.
If $x_1 - \hat{u}_2 = x_1 - r_1 < 0$ then the control action provided by C_1 is $\hat{u}_1 = \overline{x}_0$, and the state $\boldsymbol{y}_2$ goes to the origin along the curve $y_2 = h_2^-(y_1) = \frac{y_1^2}{2\overline{x}_0}$. Conversely if $x_1 - \hat{u}_2 = x_1 - r_1 > 0$ the control action provided by C_1 is $\hat{u}_1 = \underline{x}_0$ and the trajectory passing through the origin is $y_2 = h_2^+(y_1) = \frac{y_1^2}{2\underline{x}_0}$.
In this manner, as shown in Fig. 3 where some system trajectories for different initial conditions are reported, the phase space is split into two parts by the switching curve:

- in $\overline{\mathcal{R}}$ the controller composed by the cascade of C_2 and C_1 steers the system state towards $y_2 = h_2^-(y_1)$ or $y_2 = \underline{y}_1$ and then to the origin;
- in $\underline{\mathcal{R}}$ the controller steers the system state towards $y_2 = h_2^+(y_1)$ or $y_2 = \overline{y}_1$ and then to the origin.

The system is therefore globally stable. Note that according to the definition reported in (9), the controller C_2 does not apply the ideal control sequence

$$u_2 = \underline{x}_1 \text{ or } \overline{x}_1 \xrightarrow{switch} r_1$$

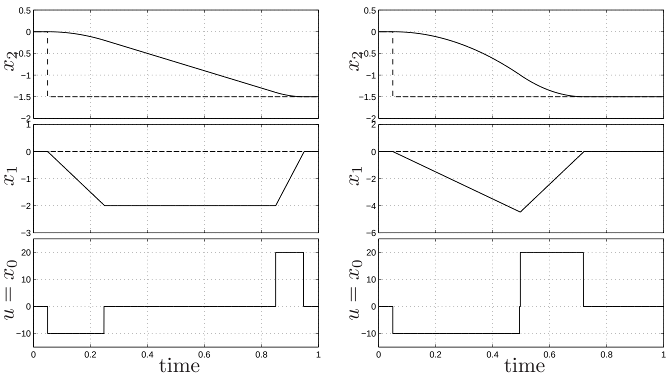
but imposes the same ideal dynamics by constraining the state $\boldsymbol{y}_2$ on the curve $y_2 = h_2(y_1)$ by applying a switching control action.

Remark 4. The controller C_2, defined by (9), can be expressed with a more compact notation as

[1] The equation (13) is a particular case of the generic trajectory passing trough the point (y_{1_0}, y_{2_0}), whose expression is

$$y_2 = h_2(y_1, y_{1_0}, y_{2_0}) = \frac{y_1^2 - y_{1_0}^2}{2\hat{u}_1} + y_{2_0}, \qquad \hat{u}_1 \ne 0.$$

Moreover, similarly to trajectories through the origin, it is possible to define generic trajectories obtained for $\hat{u}_1 = \overline{x}_0$ and $\hat{u}_1 = \underline{x}_0$, which are denoted with $h_2^-(y_1, y_{1_0}, y_{2_0})$ and $h_2^+(y_1, y_{1_0}, y_{2_0})$, respectively.



(a) (b)

Fig. 4. Output of the second order filter with a step reference input r_2, when the bound on x_1 is reached (a) and not reached (b).

x_0	$\leftarrow$	acc. (a)	x_1	$\leftarrow$	vel. (v)
x_2	$\leftarrow$	pos.			
$\underline{x}_0$	$\leftarrow$	$\mathtt{a}_{min}$	$\overline{x}_0$	$\leftarrow$	$\mathtt{a}_{max}$
$\underline{x}_1$	$\leftarrow$	$\mathtt{v}_{min}$	$\overline{x}_1$	$\leftarrow$	$\mathtt{v}_{max}$

Table 1. Symbols for the definition of the second order trajectory generator.

$$u_2(\boldsymbol{r}_2, \boldsymbol{x}_2) = \text{ISgn}_2\big(\overline{x}_1, \underline{x}_1, y_2 - h_2(y_1), y_1\big)$$

where

$$\text{ISgn}_2(\overline{x}, \underline{x}, \xi_1, \xi_2) = \begin{cases} \text{ISgn}(\overline{x}, \underline{x}, \xi_1), & \xi_1 \neq 0 \\ \text{ISgn}(\overline{x}, \underline{x}, \xi_2), & \xi_1 = 0. \end{cases}$$

The response of the second order system to a constant reference input r_2 ($r_1 = \dot{r}_2 = 0$) is reported in Fig. 4.(a) and Fig. 4.(b). Note that, when the limit values of x_1 are reached, as in Fig. 4.(a), the control action $u(t) = x_0(t)$ switches two times, from $\overline{x}_0$ to 0, and then from 0 to $\underline{x}_0$, or viceversa. Conversely if $\underline{x}_1$ or $\overline{x}_1$ are not reached the control $u(t)$ directly switches from $\overline{x}_0$ to $\underline{x}_0$ or viceversa, see Fig. 4.(b).

The second order system can be adopted as second order position trajectory generator with constraints on speed and acceleration: in this case it is sufficient to consider the substitutions of Tab. 1. The filter designed in this manner can be used to plan on-line time-optimal trajectories with continuous velocity and bounded acceleration which track at best an external reference signal.

3.3 Third order filter

In order to define the third order filter, the outer controller C_3 is defined.

Proposition 5. Given the third order system composed by a cascade of three integrators and by the controllers C_1 and C_2:

$$\begin{cases} \dot{x}_1 = u_2(u_1(u_3, \dot{u}_3, x_2, x_1), x_1) \\ \dot{x}_2 = x_1 \\ \dot{x}_3 = x_2 \end{cases}$$

where u_3 and $\dot{u}_3$ are the new control inputs, the control law C_3:

$$u_3(\boldsymbol{r}_3, \boldsymbol{x}_3) = \begin{cases} \overline{x}_2, & \text{if } y_3 < h_3(y_2, y_1) \text{ or } y_2 = h_3^-(y_2, y_1) \\ 0, & \text{if } y_3 = y_2 = y_1 = 0 \\ \underline{x}_2, & \text{if } y_3 > h_3(y_2, y_1) \text{ or } y_2 = h_3^+(y_2, y_1) \end{cases} \tag{14}$$

with

$$h_3(y_1, y_2) = \begin{cases} h_3^-(y_1, y_2) & \text{if } y_2 < h_2(y_1) \text{ or } y_2 = h_2^-(y_1) \\ 0 & \text{if } y_2 = y_1 = 0 \\ h_3^+(y_1, y_2)) & \text{if } y_2 > h_2(y_1) \text{ or } y_2 = h_2^+(y_1) \end{cases} \tag{15}$$

where h_3^- and h_3^+ are defined by (16) and (17) respectively, satisfies the constraint $\underline{x}_2 \leq x_2(t) \leq \overline{x}_2$ (and obviously $\underline{x}_1 \leq x_1(t) \leq \overline{x}_1$ and $\underline{x}_0 \leq x_0(t) \leq \overline{x}_0$) and forces the vector $\boldsymbol{x}_3$ to reach the desired value $\boldsymbol{x}_3 = \boldsymbol{r}_3$ in minimum time.

Proof. The constraints on $x_0(t)$ and $x_1(t)$ are automatically guaranteed by the inner controllers C_1 and C_2, while the compliance with the bounds on the $x_3(t)$ descends from the fact that the control signal $u_3(t)$, which is the reference signal for C_2, always ranges in $[\underline{x}_2, \overline{x}_2]$. The philosophy and the structure of the controller C_3 are the same of the inner controller C_2. By considering the dynamics of the error variables $\boldsymbol{y}_3 = [y_1, y_2, y_3]^T$:

$$\begin{cases} \dot{y}_1 = u_2(u_1(u_3, \dot{u}_3, y_2 + r_2, y_1 + r_1), y_1 + r_1) \\ \dot{y}_2 = y_1 \\ \dot{y}_3 = y_2 \end{cases} \tag{18}$$

the controller is built with the purpose of regulating in minimum time the last component of the vector $\boldsymbol{y}_3$, i.e. y_3, and requiring the internal control loops to nullify the other components. Analogously to C_2, this can be accomplish, by steering $\boldsymbol{y}_3$ to the origin with two arcs of trajectory: the former with the constant control $u_3(t) = \overline{x}_2$ or $u_3(t) = \underline{x}_2$ depending on the initial conditions $[y_{1_0}, y_{2_0}, y_{3_0}]^T$ and the latter with $u_3(t) = r_2(t)$. In this case the switching locus is a surface in a three dimensional space, that can be built by considering all the trajectories passing through the origin obtained with the generic control $u_3(t) = r_2(t)$. Such trajectories can be computed by backward integrating the error dynamics (18) with a generic input $r_2(t)$. It is therefore necessary to study the behavior of the system composed by the cascade of second order filter S_2 and the additional integrator. When $r_2(t)$ is given to the system S_2, the control action $u(t)$ directly applied to the chain of integrators is a sequence of constant value segments, see Fig. 4. In particular, as highlighted in Sec. 3.2, two different situations may occur:

(1) while the output $x_2(t)$ of the system S_2 tends to the desired value $r_2(t)$ in minimum time, the variable $x_1(t)$ reaches the boundary limit $\underline{x}_1$ or $\overline{x}_1$: in this case the control sequence is $u(t) = \hat{u}_2 \to 0 \to \hat{u}_1$, where $(\hat{u}_1, \hat{u}_2) = (\underline{x}_0, \overline{x}_0)$ or $(\hat{u}_1, \hat{u}_2) = (\overline{x}_0, \underline{x}_0)$, depending on the initial value of the state $\boldsymbol{y}_3$ and, in particular, of the two components regulated by controllers C_1 and C_2, i.e. y_1 and y_2;

(2) the variable $x_2(t)$ equals $r_2(t)$ but the limits $\underline{x}_1$ or $\overline{x}_1$ are never reached: in this case the control $u(t)$ applied to the chain of integrators directly switches from $\hat{u}_2$ to $\hat{u}_1$ without the intermediate segment with $u(t) = 0$. Also in this case the values $(\hat{u}_1, \hat{u}_2)$ correspond to

$$h_3^-(y_2,y_1) = \begin{cases} \dfrac{\overline{y}_1^4\overline{x}_0^2 + ((y_1-\overline{y}_1)^3(3y_1+\overline{y}_1) - 12(y_1-\overline{y}_1)^2y_2\overline{x}_0 + 12y_2^2\overline{x}_0^2)\underline{x}_0^2}{24\overline{y}_1\underline{x}_0^2\overline{x}_0^2}, & \text{if } y_2 > h_2^-(y_1,\underline{y}_1,h_2(\underline{y}_1)) \\ \dfrac{2y_1y_2\underline{x}_0(\underline{x}_0-\overline{x}_0)\overline{x}_0 + (y_1^2-2y_2\overline{x}_0)(\sqrt{(y_1^2-2y_2\overline{x}_0)\underline{x}_0(\underline{x}_0-\overline{x}_0)}(-2\underline{x}_0+\overline{x}_0) - 2y_1\underline{x}_0(\underline{x}_0-\overline{x}_0))}{6\underline{x}_0(\underline{x}_0-\overline{x}_0)\overline{x}_0^2}, & \text{otherwise} \end{cases} \quad (16)$$

$$h_3^+(y_2,y_3) = \begin{cases} \dfrac{\underline{y}_1^4\underline{x}_0^2 + ((y_1-\underline{y}_1)^3(3y_1+\underline{y}_1) - 12(y_1-\underline{y}_1)^2y_2\underline{x}_0 + 12y_2^2\underline{x}_0^2)\overline{x}_0^2}{24\underline{y}_1\underline{x}_0^2\overline{x}_0^2}, & \text{if } y_2 < h_2^+(y_1,\overline{y}_1,h_2(\overline{y}_1)) \\ \dfrac{2y_1y_2\underline{x}_0(\underline{x}_0-\overline{x}_0)\overline{x}_0 + (y_1^2-2y_2\underline{x}_0)(\sqrt{-(y_1^2-2y_2\underline{x}_0)\overline{x}_0(\underline{x}_0-\overline{x}_0)}(2\overline{x}_0-\underline{x}_0) - 2y_1\overline{x}_0(\underline{x}_0-\overline{x}_0))}{6\overline{x}_0(\underline{x}_0-\overline{x}_0)\underline{x}_0^2}, & \text{otherwise} \end{cases} \quad (17)$$

$(\underline{x}_0, \overline{x}_0)$, or $(\overline{x}_0, \underline{x}_0)$, according to the initial value of y_1 and y_2.

In order to build the switching surface we start from the simpler case (2). The trajectory of $\boldsymbol{y}_3$ can be computed by backward integrating from the origin the dynamics of the chain of three integrators fed by an input sequence composed by a segment of duration $-t_1$ with $u(t) = \hat{u}_1$ followed by a segment of duration $-t_2$ with $u(t) = \hat{u}_2$:

$$\boldsymbol{y}_3 = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -t_1\hat{u}_1 - t_2\hat{u}_2 \\ \dfrac{t_1(t_1+2t_2)\hat{u}_1 + t_2^2\hat{u}_2}{2} \\ \dfrac{-t_1(t_1^2+3t_1t_2+3t_2^2)\hat{u}_1 - t_2^3\hat{u}_2}{6} \end{bmatrix}. \quad (19)$$

Eliminating the parameters t_1 and t_2 from (19), one obtains the explicit expression of the trajectory:

$$y_3 = \frac{1}{6\hat{u}_1(\hat{u}_1-\hat{u}_2)\hat{u}_2^2}\Big[2y_1y_2\hat{u}_1(\hat{u}_1-\hat{u}_2)\hat{u}_2 + +(y_1^2-2y_2\hat{u}_1)\big(\sqrt{(y_1^2-2y_2\hat{u}_1)\hat{u}_1(\hat{u}_1-\hat{u}_2)} (-2\hat{u}_1+\hat{u}_2) - 2y_1\hat{u}_1(\hat{u}_1-\hat{u}_2)\big)\Big]. \quad (20)$$

The expression of the trajectory (20) is valid if the limit values of y_1 (ore equivalently, the limit values of x_1) are not reached. Otherwise (case (1)), it is necessary to compute the trajectory in a different way. In particular, as shown in Fig. 5, the regions $\overline{\mathcal{R}}$ and $\underline{\mathcal{R}}$ are both split in two sub-regions, with subscript s (saturated) and ns (not saturated) according to the fact that $\overline{y}_1$ or $\underline{y}_1$ is reached or not. The limit trajectory (dividing $\mathcal{R}_s$ and $\mathcal{R}_{ns}$) is the one passing exactly through the point $(\hat{y}_1, \hat{y}_2) = \big(\overline{y}_1, h_2(\overline{y}_1)\big)$ or $(\hat{y}_1, \hat{y}_2) = \big(\underline{y}_1, h_2(\underline{y}_1)\big)$ and has the expression $y_2 = h_2^-\big(y_1, \overline{y}_1, h_2(\overline{y}_1)\big)$, if $(y_1, y_2) \in \overline{\mathcal{R}}$ or $y_2 = h_2^+\big(y_1, \underline{y}_1, h_2(\underline{y}_1)\big)$, otherwise.

In case (1), that occurs if the projection of the trajectory of $\boldsymbol{y}_3$ on the plane $y_3 = 0$ lies on $\mathcal{R}_s$, it is necessary to backward integrate from the origin the dynamics of the integrators chain with an input sequence composed by a segment of duration $-t_1$ with $u(t) = \hat{u}_1$, followed by a segment of duration $-t_2$ with $u(t) = 0$ and finally a segment of duration $-t_3$ with $u(t) = \hat{u}_2$:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -t_1\hat{u}_1 - t_3\hat{u}_2 \\ \dfrac{t_1(t_1+2(t_2+t_3))\hat{u}_1 + t_3^2\hat{u}_2}{2} \\ \dfrac{-t_1(t_1^2+3t_1(t_2+t_3)+3(t_2+t_3)^2)\hat{u}_1 - t_3^3\hat{u}_2}{6} \end{bmatrix} \quad (21)$$

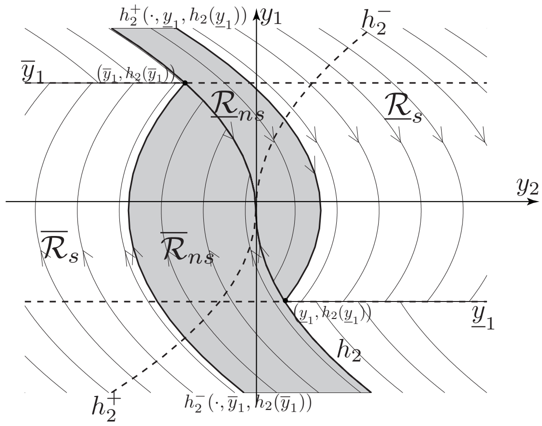


Fig. 5. Phase portrait of the second order filter S_2 with the regions corresponding to different control sequences.

with a further constraint tied to the limits on y_1 (x_1):

$$-t_1 = \frac{\hat{y}_1}{\hat{u}_1} \quad (22)$$

Note that the substitution of (22) in (21) is equivalent to backward integrate the system dynamics from $\hat{y}_1$ instead of the origin with an input composed by two tracts: the former of duration $-t_2$ with $u(t) = 0$ and the latter of duration $-t_3$ with $u(t) = \hat{u}_2$.

Finally, the expression of the surface obtained by eliminating the parameters t_1, t_2 and t_3 from (21) and (22) results

$$y_3 = \frac{1}{24\hat{y}_1\hat{u}_1^2\hat{u}_2^2}\Big(\hat{y}_1^4\hat{u}_2^2 + ((y_1-\hat{y}_1)^3(3y_1+\hat{y}_1) - -12(y_1-\hat{y}_1)^2y_2\hat{u}_2 + 12y_2^2\hat{u}_2^2)\hat{u}_1^2\Big) \quad (23)$$

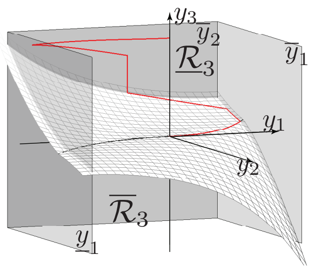
with $(\hat{u}_1, \hat{u}_2, \hat{y}_1) = (\underline{x}_0, \overline{x}_0, \overline{y}_1)$ if the projection of the trajectory on the plane $y_3 = 0$ starts in $\overline{\mathcal{R}}_s$ and $(\hat{u}_1, \hat{u}_2, \hat{y}_1) = (\overline{x}_0, \underline{x}_0, \underline{y}_1)$ if it starts in $\underline{\mathcal{R}}_s$.

The complete switching surface $y_3 = h_3(y_1, y_2)$ is obtained by combining (20) and (23), with the proper values of $\hat{u}_1$ and $\hat{u}_2$.

Similarly to the controller C_2, C_3 does not apply the ideal control sequence

$$u_3(t) = \overline{x}_2 \text{ or } \underline{x}_2 \xrightarrow{switch} r_2$$

but imposes the same ideal dynamics by forcing the state $\boldsymbol{y}$ on the surface $y_3 = h_3(y_1, y_2)$ by applying a switching control action. When the trajectory hits this surface the controller C_2 steers the trajectory towards the curve whose projection on the plane $y_3 = 0$ is $y_2 = h_2(y_1)$ and finally



(a) (b)

Fig. 6. Trajectories of the error variable $\boldsymbol{y}_3$ with the controller C_3 in case that the maximum value of x_2 is reached (a) and not reached (b).

x_0	$\leftarrow$	jerk (j)	x_1	$\leftarrow$	acc. (a)
x_2	$\leftarrow$	vel.(v)	x_3	$\leftarrow$	pos.
$\underline{x}_0$	$\leftarrow$	$\mathtt{j}_{min}$	$\overline{x}_0$	$\leftarrow$	$\mathtt{j}_{max}$
$\underline{x}_1$	$\leftarrow$	$\mathtt{a}_{min}$	$\overline{x}_1$	$\leftarrow$	$\mathtt{a}_{max}$
$\underline{x}_2$	$\leftarrow$	$\mathtt{v}_{min}$	$\overline{x}_2$	$\leftarrow$	$\mathtt{v}_{max}$

Table 2. Symbols for the definition of the third order trajectory generator.

the trajectory moves along this curve until the origin is reached. Therefore, the overall control is globally stable.

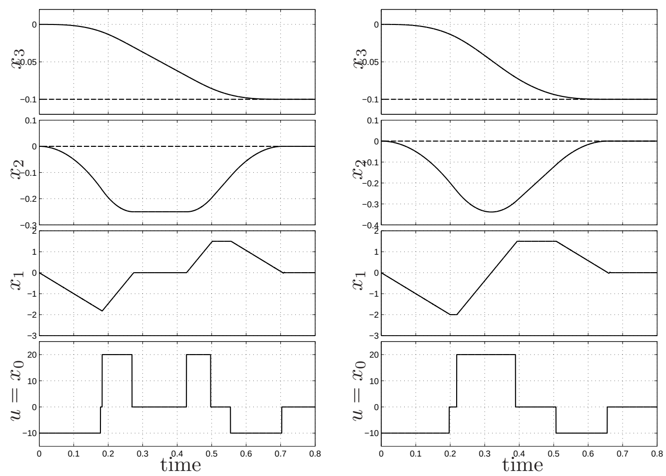
Remark 6. The controller C_3, defined by (9), can be expressed with a more compact notation as

$$u_3(\boldsymbol{r}_3, \boldsymbol{x}_3) = \mathrm{ISgn}_3(\overline{x}_2, \underline{x}_2, y_3 - h_3(y_1, y_2), y_2 - h_2(y_1), y_1)$$

where

$$\mathrm{ISgn}_3(\overline{x}, \underline{x}, \xi_1, \xi_2, \xi_3) = \begin{cases} \mathrm{ISgn}(\overline{x}, \underline{x}, \xi_1), & \xi_1 \neq 0 \\ \mathrm{ISgn}(\overline{x}, \underline{x}, \xi_2), & \xi_1 = 0, \xi_2 \neq 0 \\ \mathrm{ISgn}(\overline{x}, \underline{x}, \xi_3), & \xi_1 = 0, \xi_2 = 0. \end{cases}$$

The figure 6 shows the trajectories of $\boldsymbol{y}_3$ when a step reference input is applied to the filter and two different situations occur: in Fig. 6.(a) the limit value of y_2 is reached before the trajectory hits the switching surface, while in Fig. 6.(b) this does not happen. The corresponding profiles of $[x_1, x_2, x_3]^T$ are reported in Fig. 7. Note that in this case the filter used as trajectory generator



(a) (b)

Fig. 7. Output of the third order filter with a step reference input r_3 corresponding to the trajectories of the error dynamics of Fig. 6.

(with the substitutions reported in Tab. 2), guarantees the continuity of velocity, acceleration and also a bounded jerk.

4. CONCLUSIONS

In this paper the design of a filter for on-line trajectory generation with constraints on velocity, acceleration and jerk has been translated into a regulation problem for a chain of integrators with bounds in the internal (state) variables. A nested structure of the controllers based on variable structure control allows a great simplification of the design procedures, which are all based on the same concept: regulating to zero the last element of the error vector $\boldsymbol{y}_i$, leaving the task of nullifying the other components to the inner control loops. This modular structure is particularly suitable for considering filters of order higher than three. The possibility of extending the proposed approach to systems of higher order is strictly related to the capability of computing the switching surface, and therefore to the capability of solve systems of polynomial equations. With respect to this problem it is worth noticing that in the literature some techniques aiming at systematize the calculations have been proposed, see Walther et al. [2001].

REFERENCES

P.-J. Barre, R. Bearee, P. Borne, and E. Dumetz. Influence of a jerk controlled movement law on the vibratory behaviour of high-dynamics systems. *Journal of Intelligent and Robotic Systems*, 42(3):275–293, 2005.

J.E. Bobrow, S. Dubowsky, and J.S. Gibson. Time-optimal control of robotic manipulators along specified paths. *The Int. Journal of Robotics Research*, 4:3–17, 1985.

P. Lambrechts, M. Boerlage, and M. Steinbuch. Trajectory planning and feedforward design for electromechanical motion systems. *Control Engineering Practice*, 13:145–157, 2005.

E. B. Lee and L. Markus. *Foundations of Optimal Control Theory*. John Wiley and sons, 1967.

Y. Dae Lee and B. Hee Lee. Genetic trajectory planner for a manipulator with acceleration parametrization. *j-jucs*, 3(9):1056–1073, 1997.

C. G. Lo Bianco, A. Tonielli, and R. Zanasi. Nonlinear filters for the generation of smooth trajectories. *Automatica*, 36:439–448, 2000.

N. Marchand and A. Hably. Nonlinear stabilization of multiple integrators with bounded controls. *Automatica*, 41:2147–2152, 2005.

K.G. Shin and N.D. Mckay. A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE Trans. on automatic Control*, 6:491–500, 1986.

S. Singh and M.C. Leu. Optimal trajectory generation for robotic manipulators using dynamic programming. *Journal of dynamic systems, measurement, and control*, 109:88–96, 1987.

A.R. Teel. Global stabilization and restricted tracking for multiple integrators with bounded controls. *Systems and Control Letters*, 18:165–171, 1992.

U. Walther, T.T. Georgiou, and A. Tannenbaum. On the computation of switching surfaces in optimal control: A grbner basis approach. *IEEE Transactions on Automatic Control*, 46(4):534–540, 2001.

R. Zanasi and R. Morselli. Third order trajectory generator satisfying velocity, acceleration and jerk constraints. In *Proc. of Int. Conf. on Control Applications*, 2002.