MOTION_CONTROL 2010, Convegno Nazionale ANIPLA, Milano, Italy, 10-11 November 2010.

Dynamic Filters for Online Planning Optimal Trajectories

L. Biagiotti[1], C. Melchiorri[2], R. Zanasi[1]

1. DII, University of Modena and Reggio Emilia
Via Vignolese 905, 41100 Modena, Italy
email: {luigi.biagiotti, roberto.zanasi}@unimore.it

2. DEIS, University of Bologna,
Via Risorgimento 2, 40136 Bologna, Italy
email: claudio.melchiorri@unibo.it

Abstract

Filters for time-optimal trajectory generation can be obtained in different ways with quite different performances and complexity levels. However, one can easily observe that the configurations of such filters are based on two main schemes: systems composed by a chain of integrators with a feedback control and systems formed by a sequence of Finite Impulse Response (FIR) filters disposed in a cascade configuration. Both trajectory generators are characterized by the order n that defines the degree of continuity of the resulting trajectory and both can be designed according to a modular approach that allows to obtain the n-th order filter from that of $n-1$ order. In this paper, after having presented the structure and the analytical expression of the two types of trajectory filters, their common features (possibility of generating online time-optimal trajectories under constraints of velocity, acceleration, jerk, etc.) but especially the main differences are analyzed. In particular, the possible applications of the two systems are considered. Trajectory filters with a feedback loop are able to track generic input signals (and not only step functions) guaranteing the compliance of the output with the given constraints but are characterized by a complexity that limits their use to the third order. Conversely, the simple structure and the low computational cost make open-loop filters desirable for point-to point motions, even with an high degree of smoothness ($n = 4$ or 5). Moreover, the low-pass response of this type of filters allows to characterize (and to design) the output trajectory from a frequency point of view, but, on the other hand, it may cause large delays and distortions between input and the output signals.

1 Introduction

Generation of time-optimal trajectories subject to kinematic constraints (on velocity, acceleration, jerk, etc.) plays an important role in all those industrial applications where the execution time of motion profiles, and accordingly the total duration of tasks to be performed, is critical. For this reason, a large number of papers addressing this problem is available in the literature, including both offline algorithms for the computation of trajectories typically based on dynamic programming, [1, 2] or other different optimization methods, [3, 4], and on-line planners, able to compute in real-time the trajectory once that the desired final configuration (position, velocity, etc.) has been specified, [5, 6]. In particular, in [5] 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 [6] 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).

Besides the possibility of generating online trajectories compliant with given constraints, the structure of the dynamic filters may have relevant implications on the spectral content of the motion profile. As a matter of fact, the need of high velocities often leads to the excitation of eigenfrequencies of machines/robots caused by structural flexibilities and may produce vibrations and large tracking errors. For this reason, a number of works copes the problem of filtering preplanned trajectories in order to reduce residual vibrations. The available methods range from low-pass and notch filters to input shaping techniques, see [7] for a comparative overview. With this respect, the features of the filter for trajectory planning can be exploited in order to take into account frequency specifications and not only time-domain constraints.

The paper is organized as follows. In Sec. 2 the main concepts tied to time-optimal trajectories and dynamic filters for online trajectory generation are presented, then in Sections 3 and 4 the analytical expressions of trajectory generators based on feedback controllers and on cascades of FIR filters are provided together with their main properties. The two different types of trajectory generators are finally compared in Sec. 5, in particular with respect to the different problems that they can solve (point-to-point optimal trajectory generation, smoothing of pre-planned trajectories, etc.). Concluding remarks are provided in the last section.

2 Optimal trajectories and dynamic filters

The optimization process of trajectories subject to constraints on velocity, acceleration, jerk, ..., leads to the so-called multi-segment trajectories, i.e. trajectories composed by several tracts properly joined, each one characterized by a specific analytical expression, and in which the velocity, the acceleration, or higher derivatives (depending on the required order of continuity) are saturated to the maximum allowed value. By imposing constraints on the first n derivatives, i.e.

$$q_{min}^{(i)} \leq q_n^{(i)}(t) \leq q_{max}^{(i)}, \qquad i = 1, \ldots, n \tag{1}$$

one obtains a trajectory $q_n(t)$ of class $\mathcal{C}^{n-1}$, that is with the first $n-1$ derivatives that are continuous, while the n-th derivative $q_n^{(n)}(t)$ is a piece-wise constant function whose values belong to a set $\{q_{min}^{(n)}, 0, q_{max}^{(n)}\}$. The number n is called order of the trajectory. Typical examples of multi-segment trajectories are the “trapezoidal velocity” trajectories or the “double S velocity” trajectories, of order two and three, respectively.

Usually, these trajectories are defined by analytic functions, indeed rather complex, but in the literature different methods based on dynamic systems have been proposed for their generation. The dynamic filters for trajectory planning generate on-line a time optimal trajectory $q_n(t)$ that tracks at best a reference signal $r(t)$, satisfying desired constraints on the first n derivatives of $q_n(t)$. The reference signal $r(t)$ is generally given by a first coarse trajectory generator providing for instance a piecewise constant profile which defines the desired final positions, or is an external input, given for example by a human operator.

Among the different trajectory generators, it is possible to identify two main categories according to their structure:

1. trajectory generators characterized by a control loop, see for instance Fig. 1;
2. trajectory generators characterized by a cascade configuration, like in Fig. 10.

Besides the different structure, the two groups enjoy peculiar features that make each type of generator preferable for a specific application. In particular, open-loop systems have a very simple structure and imply a computational cost rather lower than similar closed-loop generators. Because of their simple structure, they can be easily generalized in order to implement high order trajectories (with $n = 4$ or $n = 5$). On the other hand, this kind of trajectory planners suffers from some limitations that can be overcome by adopting closed-loop filters. As a matter of fact, filters with a cascade configuration only work with symmetric constraints. Moreover, when used for online planning trajectories the desired bounds cannot be changed in runtime and a new trajectory cannot start before the current motion profile has executed without violating the desired limit values. On the contrary, closed-loop trajectory generators allow to modify the limits of velocity, acceleration, jerk, etc. and to start a new motion at any time.
Finally, while the responses to a step input of the two type of dynamic filters are equal (the point-to-point optimal trajectory for a given set of constraints is unique), the behaviors of open- and closed-loop planners are very different when used to filter generic input profiles, as highlighted in Sec. 5.

3 Closed-loop trajectory generator

The trajectory planner based on feedback regulation is composed by a chain of n integrators and by a nonlinear controller able to nullify in minimum time the tracking error between the reference input $r(t)$ and the integrators output $x_n(t) = q_n(t)$, being compliant with constraints (1), see Fig. 1 [8]. More formally, given the system

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

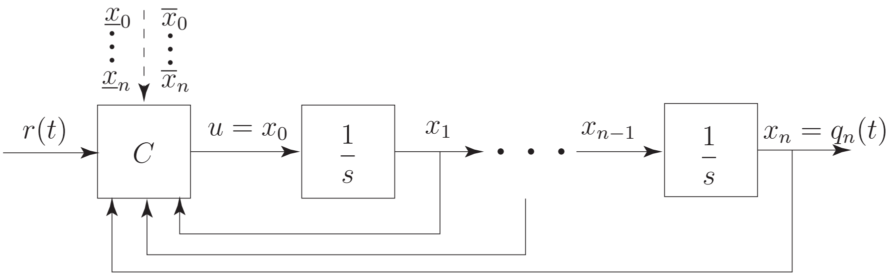


Figure 1: Structure of a closed-loop trajectory filter of n-th order.

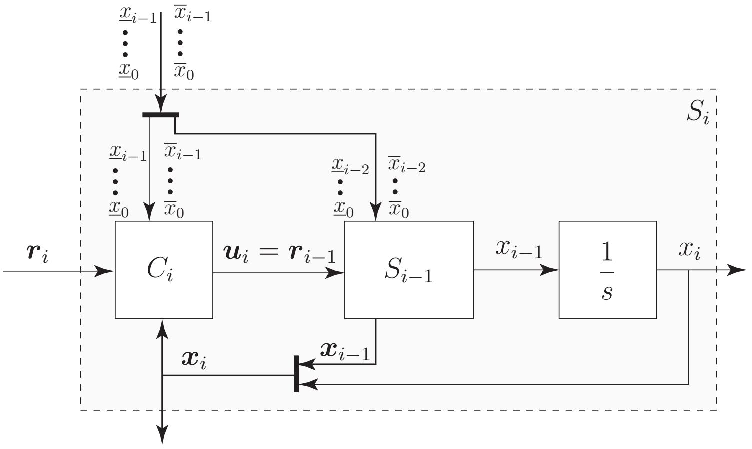


Figure 2: Structure of the i-th control loop ($S_0 = 1$).

with the state vector $\boldsymbol{x} = [x_1, x_2, \dots, x_{n-1}, x_n]^T = [q_n^{(n-1)}, q_n^{(n-2)}, \dots, q_n^{(1)}, q_n]^T$ and the constraints on its components

$$\underline{x}_i \le x_i(t) \le \overline{x}_i, \qquad i = 1, \dots, n \tag{3}$$

where $x_i(t) = q^{(n-i)}(t)$ and $\underline{x}_i = q_{min}^{(n-i)}$, $\overline{x}_i = q_{max}^{(n-i)}$, $i = 1, \dots, n$, aim of the controller C is to steer to the origin the tracking error defined as $\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{r}$, being $\boldsymbol{r} = [r_1, r_2, \dots, r_{n-1}, r_n] = [r^{(n-1)}, r^{(n-2)}, \dots, r^{(1)}, r]^T$. Note that the vector of the inputs is composed by the reference signal and its derivatives (up to $n-1$). Moreover it is assumed that $r^{(n)}(t) \equiv 0$. Obviously, the controller C must not only nullify all the components y_i of the error vector, but also guarantee that for any value of the time t

$$\underline{y}_i \le y_i(t) \le \overline{y}_i, \qquad i = 1, \dots, n-1 \tag{4}$$

where the limit values are computed as

$$\underline{y}_i = \underline{x}_i - r_i, \qquad \overline{y}_i = \overline{x}_i - r_i.$$

Note that the bounds on y_i are not constant since they depend on the reference input $r(t)$. Finally, the problem of time-optimal trajectory generation is translated into a problem of regulation of a chain of n integrators with constraints. In order to obtain a modular solution, that for the design of the n-th order trajectory planner takes advantage of the filter of order $n-1$, the nonlinear controller is obtained by nesting n controllers, each one devoted to nullify a specific element of the error vector $\boldsymbol{y}$ in minimum time. For instance, 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. 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 $\boldsymbol{x}_i = [x_1, \dots, x_i]^T$ to $\boldsymbol{r}_i = [r_{i,1}, \dots, r_{i,i}]^T$. Note that, the $(i-1)$-th system S_{i-1} has a input vector $\boldsymbol{r}_i$ 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 $\boldsymbol{u}_i = [0, \dots, 0, u_i]^T$ where

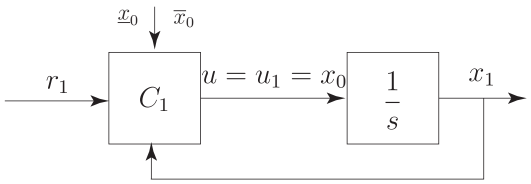


Figure 3: Block-scheme representation of the first order filter.

the term u_i is piecewise constant and accordingly its derivatives are null. In the following, 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.

3.0.1 First order filter

Given the system of Fig. 3, composed by a single integrator

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

the control law

$$C_1 : \quad u_1(r_1, x_1) = \begin{cases} \overline{x}_0, & \text{if } y_1 < 0 \\ 0, & \text{if } y_1 = 0 \\ \underline{x}_0, & \text{if } y_1 > 0 \end{cases} \tag{5}$$

satisfies the constraint $\underline{x}_0 \leq x_0(t) \leq \overline{x}_0$ (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.

The first order filter cannot be adopted in practical applications since it implies an impulsive acceleration but is the starting point for higher order trajectory planners.

3.0.2 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 $[r_1, r_2]^T$, see Fig. 4. Given the second order system composed by a cascade of two integrators and by the controller C_1:

$$\begin{cases} \dot{x}_1 = & u_1(u_2, x_1) \\ \dot{x}_2 = & x_1 \end{cases} \tag{6}$$

where u_2 is the new control input, the control law

$$C_2 : u_2(\boldsymbol{r}_2, \boldsymbol{x}_2) = \begin{cases} \overline{x}_1, & \text{if } y_2 < h_2(y_1) \text{ or } y_2 = h_2^-(y_1) \\ 0, & \text{if } y_2 = y_1 = 0 \\ \underline{x}_1, & \text{if } y_2 > h_2(y_1) \text{ or } y_2 = h_2^+(y_1) \end{cases} \tag{7}$$

x_0	$\leftarrow$	$q_2^{(2)}(t)$	x_1	$\leftarrow$	$q_2^{(1)}(t)$	x_2	$\leftarrow$	$q_2(t)$
$\underline{x}_0$	$\leftarrow$	$q_{min}^{(2)}$	$\underline{x}_1$	$\leftarrow$	$q_{min}^{(1)}$			
$\overline{x}_0$	$\leftarrow$	$q_{max}^{(2)}$	$\overline{x}_1$	$\leftarrow$	$q_{max}^{(1)}$			

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

with

$$h_2(y_1) = \begin{cases} h_2^-(y_1) = \dfrac{y_1^2}{2\overline{x}_0}, & \text{if } y_1 < 0 \\ 0, & \text{if } y_1 = 0 \\ h_2^+(y_1) = \dfrac{y_1^2}{2\underline{x}_0}, & \text{if } y_1 > 0 \end{cases} \tag{8}$$

satisfies the constraint $\underline{x}_1 \leq x_1(t) \leq \overline{x}_1$ (and obviously $\underline{x}_0 \leq x_0(t) \leq \overline{x}_0$) and forces the variable x_2 to reach the value r_2 (and x_1 the desired value r_1) in minimum time. The behavior of the second order system is summarized in Fig. 5: the application of any reference input moves the error $\boldsymbol{y}$ away from the origin of the phase space and the controller acts so that $\boldsymbol{y}$ is firstly steered toward the curve defined by $y_2 = h_2(y_1)$ and then moves to the origin along this curve. The function $y_2 = h_2(y_1)$ defines the so-called switching locus, i.e. the set of points where the control action $u(t)$ must switch from the maximum to the minimum value and viceversa. In general, only one switch is necessary to reach the origin. Note that, when the trajectory reaches the value $\underline{y}_1$ or $\overline{y}_1$, it proceeds along the curve $y_1 = \underline{y}_1$ or $y_1 = \overline{y}_1$, respectively. Therefore, if the trajectory starts in the region between $y_1 = \underline{y}_1$ and $y_1 = \overline{y}_1$, the limit values are never exceeded.
With the substitutions of Tab. 1. the second order system can be adopted as second order position trajectory generator with constraints on speed and acceleration. The responses of the trajectory generator to a constant reference input r_2 ($r_1 = \dot{r}_2 = 0$) are reported in Fig. 6(a) and Fig. 6(b). In the former case the limit value of the velocity is reached, while in the latter the maximum velocity is not attained. Figure 6(c) shows the output of the filter when the reference r_1 is a constant and therefore r_2 varies like a ramp. The system tries to

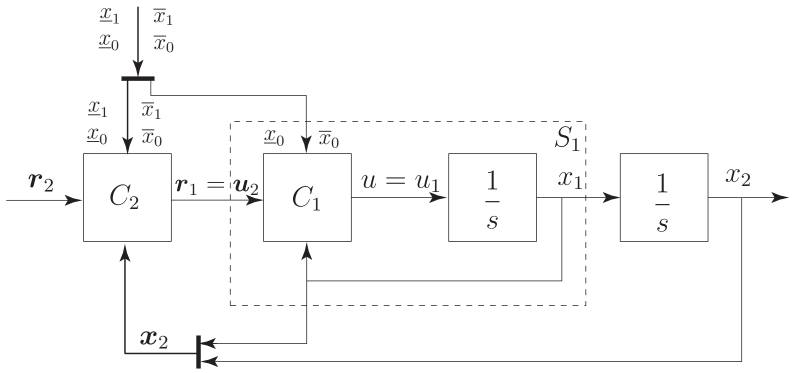

$\underline{x}_1$ $\overline{x}_1$
$\underline{x}_0$ $\overline{x}_0$
$\underline{x}_1$ $\overline{x}_1$
$\underline{x}_0$ $\overline{x}_0$
$\underline{x}_0$ $\overline{x}_0$
S_1
$\boldsymbol{r}_2$
C_2
$\boldsymbol{r}_1 = \boldsymbol{u}_2$
C_1
$u = u_1$
$\frac{1}{s}$
x_1
$\frac{1}{s}$
x_2
$\boldsymbol{x}_2$

Figure 4: Block-scheme representation of the second order filter.

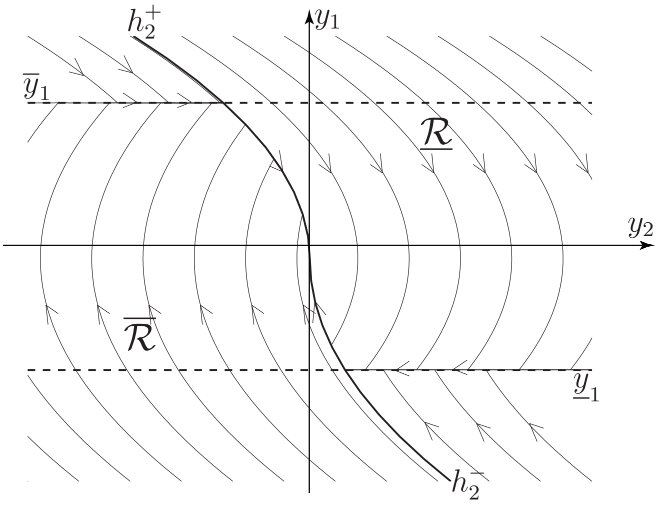


Figure 5: Phase portrait of the second order filter.

reach the reference signal at the maxim allowed speed and when this occurs it continues to track r_2 without error.
Note that in the three cases, the limit values of the acceleration are not symmetric, being $q^{(2)}_{min} = -10$ and $q^{(2)}_{max} = 20$.

3.0.3 Third order filter

In order to define the third order filter, the outer controller C_3 is defined, as shown in Fig. 7. Given the third order system composed by a cascade of three integrators and by the controllers C_1 and C_2:

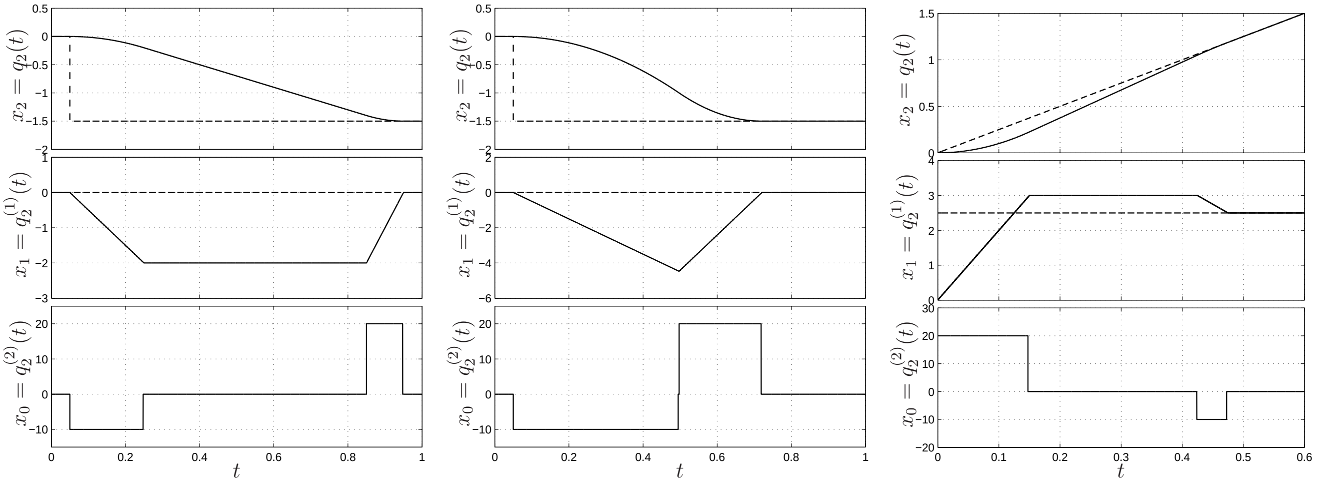
$$\begin{cases} \dot{x}_1 = u_2(u_1(u_3, \dot{u}_3, x_2, x_1), x_1) \\ \dot{x}_2 = x_1 \\ \dot{x}_3 = x_2 \end{cases}$$

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

$$u_3(\boldsymbol{r}_3, \boldsymbol{x}_3) = \begin{cases} \overline{x}_2, & \text{if } y_3 < h_3(y_2, y_1) \text{ or } y_2 = h_3^-(y_2, y_1) \\ 0, & \text{if } y_3 = y_2 = y_1 = 0 \\ \underline{x}_2, & \text{if } y_3 > h_3(y_2, y_1) \text{ or } y_2 = h_3^+(y_2, y_1) \end{cases} \tag{9}$$

with

$$h_3(y_1, y_2) = \begin{cases} h_3^-(y_1, y_2) & \text{if } y_2 < h_2(y_1) \text{ or } y_2 = h_2^-(y_1) \\ 0 & \text{if } y_2 = y_1 = 0 \\ h_3^+(y_1, y_2)) & \text{if } y_2 > h_2(y_1) \text{ or } y_2 = h_2^+(y_1) \end{cases} \tag{10}$$



(a) (b) (c)

Figure 6: 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) and output of the filter with a ramp reference input r_2 (c).

where h_3^- and h_3^+ are defined by

$$h_3^-(y_2,y_1)=\begin{cases}\dfrac{\overline{y}_1^4\overline{x}_0^2+((y_1-\overline{y}_1)^3(3y_1+\overline{y}_1)-12(y_1-\overline{y}_1)^2y_2\overline{x}_0+12y_2^2\overline{x}_0^2)\underline{x}_0^2}{24\overline{y}_1\underline{x}_0^2\overline{x}_0^2}, & \text{if } y_2>h_2^-(y_1,\underline{y}_1,h_2(\underline{y}_1))\\[2ex] \dfrac{2y_1y_2\underline{x}_0(\underline{x}_0-\overline{x}_0)\overline{x}_0+(y_1^2-2y_2\overline{x}_0)(\sqrt{(y_1^2-2y_2\overline{x}_0)\underline{x}_0(\underline{x}_0-\overline{x}_0)}(-2\underline{x}_0+\overline{x}_0)-2y_1\underline{x}_0(\underline{x}_0-\overline{x}_0))}{6\underline{x}_0(\underline{x}_0-\overline{x}_0)\overline{x}_0^2}, & \text{otherwise}\end{cases} \tag{11}$$

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

satisfies the constraint $\underline{x}_2 \le x_2(t) \le \overline{x}_2$ (and obviously $\underline{x}_1 \le x_1(t) \le \overline{x}_1$ and $\underline{x}_0 \le x_0(t) \le \overline{x}_0$) and forces the vector $\boldsymbol{x}_3$ to reach the desired value $\boldsymbol{x}_3 = \boldsymbol{r}_3$ in minimum time. Note that the design philosophy and the structure of controller C_3 are exactly the same of controllers C_1 and C_2. The function $y_3 = h_3(y_2, y_1)$ defines a surface that splits the three-dimensional error space in two regions where the extremal control $u(t) = \underline{x}_2$ or $u(t) = \overline{x}_2$ is applied in order to steer the points $\boldsymbol{y}_3$ towards the surface itself. When the surface is met, the control action switches its value and $\boldsymbol{y}_3$ goes to the origin remaining on this surface. The figure 8 shows the trajectories of $\boldsymbol{y}_3$ when a step reference input is applied to the filter and two different situations occur: in Fig. 8(a) the limit value of y_2 is reached before the trajectory hits the switching surface, while in Fig. 8(b) this does not happen. The corresponding profiles of $[x_1, x_2, x_3]^T$ are reported in Fig. 9. In this case the filter used as trajectory generator (with the substitutions reported in Tab. 2), guarantees the continuity of velocity, acceleration and also the boundedness of jerk. Obviously, the filter is able to track in minimum time not only step inputs, but also ramp and even parabolic signals.

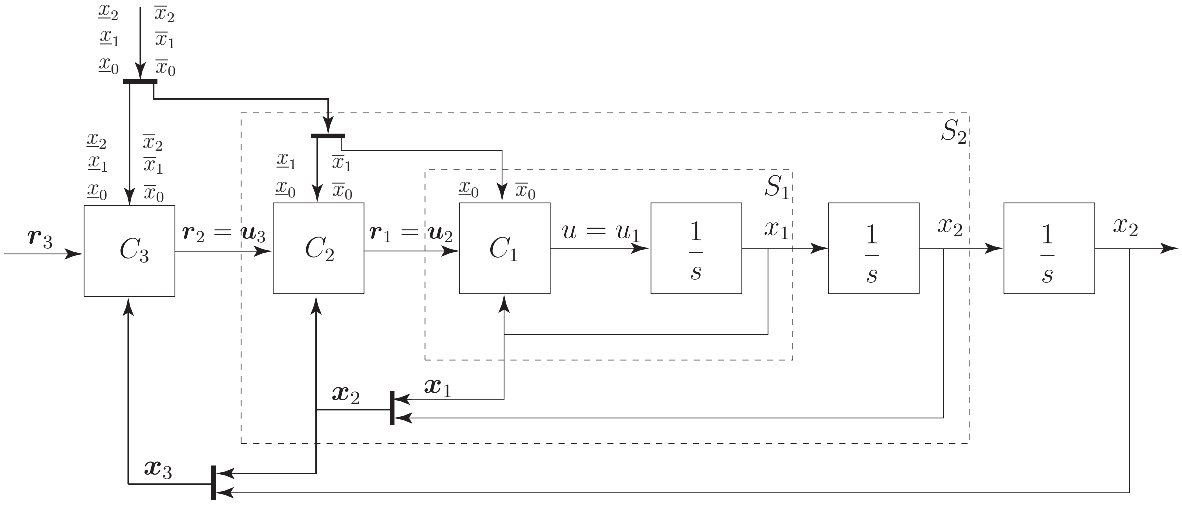

$\underline{x}_2$ $\underline{x}_1$ $\underline{x}_0$
$\overline{x}_2$ $\overline{x}_1$ $\overline{x}_0$
$\underline{x}_2$ $\underline{x}_1$ $\underline{x}_0$
$\overline{x}_2$ $\overline{x}_1$ $\overline{x}_0$
$\underline{x}_1$ $\underline{x}_0$
$\overline{x}_1$ $\overline{x}_0$
$\underline{x}_0$ $\overline{x}_0$
S_2
S_1
$\boldsymbol{r}_3$
C_3
$\boldsymbol{r}_2 = \boldsymbol{u}_3$
C_2
$\boldsymbol{r}_1 = \boldsymbol{u}_2$
C_1
$u = u_1$
$\frac{1}{s}$
x_1
$\frac{1}{s}$
x_2
$\frac{1}{s}$
x_2
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
$\boldsymbol{x}_3$

Figure 7: Block scheme representation of a third order filter.

x_0	$\leftarrow$	$q_3^{(3)}(t)$	x_1	$\leftarrow$	$q_3^{(2)}(t)$	x_2	$\leftarrow$	$q_3^{(1)}(t)$	x_3	$\leftarrow$	$q_3(t)$
$\underline{x}_0$	$\leftarrow$	$q_{min}^{(3)}$	$\underline{x}_1$	$\leftarrow$	$q_{min}^{(2)}$	$\underline{x}_2$	$\leftarrow$	$q_{min}^{(1)}$			
$\overline{x}_0$	$\leftarrow$	$q_{max}^{(3)}$	$\overline{x}_1$	$\leftarrow$	$q_{max}^{(2)}$	$\underline{x}_2$	$\leftarrow$	$q_{max}^{(1)}$			

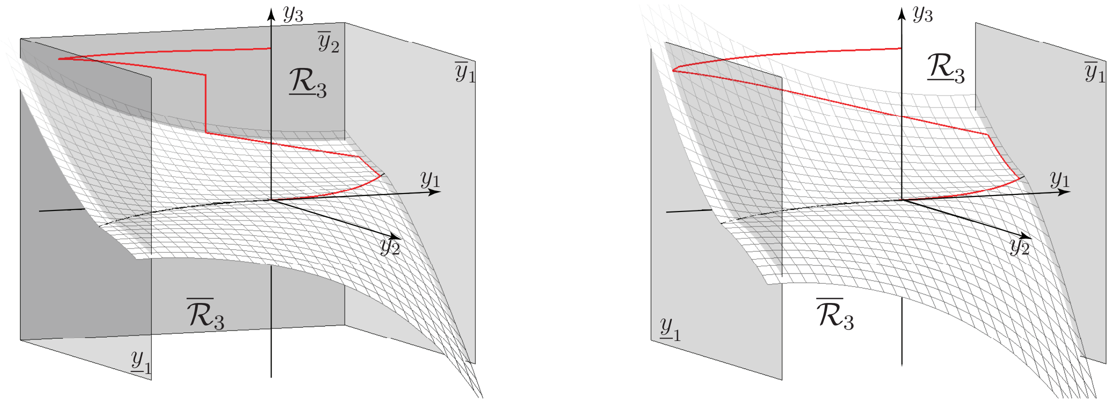
Table 2: Symbols for the definition of the third order trajectory generator.

Thanks to the modular structure, the procedure for filters design can be further iterated in order to obtain higher order trajectory planners. However, the solution, simple from a conceptual point of view, becomes very complicated because of the computation of the analytical expression of the switching surface. For this reason, the use of closed-loop trajectory generators is usually limited to second and third order.

Moreover, is worth noticing that the filter responses reported in Fig. 6 and Fig. 9 are obtained in the ideal case of continuous-time systems. However, a practical implementation on a digital controller requires discrete-time systems. It is therefore necessary to approximate the dynamic parts of the filter, in particular the chain of integrators, with equivalent discrete-time elements, e.g. by considering the integrators obtained by discretization with the backward differences method:

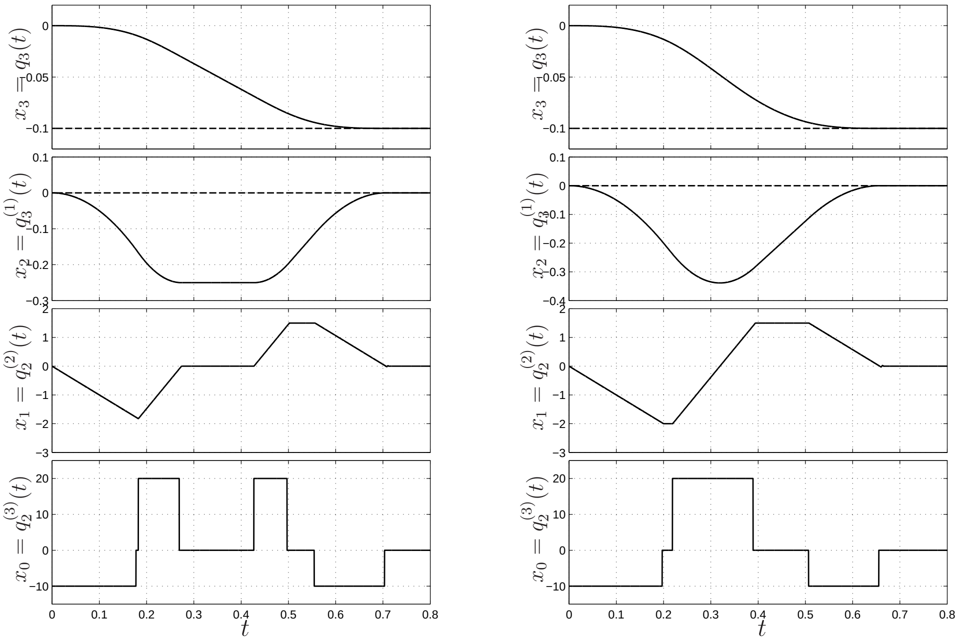
$$\frac{1}{s} \rightarrow \frac{T_s}{1 - z^{-1}}$$

where T_s is the sampling time. Unfortunately, such an approximation introduces in the system some chattering, i.e. a rapid variation of the control action when the error $\boldsymbol{y}$ approaches the origin, see Sec. 5. In order to suppress such oscillations, one should design the control system directly in the discrete-time domain [9, 10], but the problem is still under investigation at least for the third order filter [11].



(a) (b)

Figure 8: 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).



(a) (b)

Figure 9: Output of the third order filter with a step reference input r_3 corresponding to the trajectories of the error dynamics of Fig. 8.

4 Open-loop trajectory generator

Under the condition of symmetric constraints, i.e.

$$q_{min}^{(i)} = -q_{max}^{(i)}, \qquad i = 1, \dots, n+1$$

one can prove that a multi-segment trajectory of order n can be obtained by filtering a step input with a cascade of n dynamic filters, each one characterized by the transfer function

$$M_i(s) = \frac{1}{T_i}\frac{1 - e^{-sT_i}}{s} \tag{13}$$

where the parameter T_i (in general different for each filter composing the chain) is a time length, see Fig. 10. Therefore, the expression of the optimal trajectory of order n can be written as

$$q_n(t) = \underbrace{h \cdot 1(t)}_{=r(t)} * m_1(t) * \dots * m_{n-1}(t) * m_n(t) \tag{14}$$

where $1(t)$ denote the unit-step function and h is the desired displacement between the current configuration and the desired (final) one. Note that (14) can be also formulated in a recursive way, according to

$$q_n(t) = q_{n-1}(t) * m_n(t) \tag{15}$$

where $q_0(t) = r(t) = h \cdot 1(t)$. The smoothness of the trajectory, that is the number of continuous derivatives, is strictly tied to the number of filters composing the chain. If we consider n filters, the resulting trajectory will be of class $\mathcal{C}^{n-1}$. By increasing the smoothness of the trajectory, the duration augments as well. As a matter of fact, the total duration of a trajectory planned by means of n dynamic systems $M_i(s)$ is simple given by the sum of the lengths of impulse response of each filter, i.e.

$$T_{tot} = T_1 + T_2 + \dots + T_n.$$

The parameters T_i can be set with the purpose of imposing desired bounds on velocity, acceleration, jerk and higher derivatives, by simply assuming

$$\begin{aligned} T_1 &= \frac{|h|}{q_{max}^{(1)}} \\ T_i &= \frac{q_{max}^{(i-1)}}{q_{max}^{(i)}}, \quad i = 2, \dots, n \end{aligned} \tag{16}$$

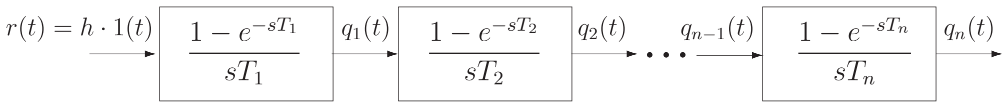

$r(t) = h \cdot 1(t)$
$\frac{1 - e^{-sT_1}}{sT_1}$
$q_1(t)$
$\frac{1 - e^{-sT_2}}{sT_2}$
$q_2(t)$
$q_{n-1}(t)$
$\frac{1 - e^{-sT_n}}{sT_n}$
$q_n(t)$

Figure 10: System composed by n filters for the computation of an optimal trajectory of class $\mathcal{C}^{n-1}$.

with the constraints

$$\begin{array}{rcl} T_1 & \geq & T_n + \ldots + T_2 \\ T_2 & \geq & T_n + \ldots + T_3 \\ & \vdots & \\ T_{n-2} & \geq & T_n + T_{n-1} \\ T_{n-1} & \geq & T_n. \end{array} \tag{17}$$

These conditions are necessary to guarantee that the output trajectory never exceeds the limits $q_{max}^{(i)}$, $i = 1, \ldots, n$. On the other hand, they restrict the types of trajectories that the filter is able to generate. In particular, the feasible trajectories are those that reach all the bounds $q_{max}^{(i)}$, while trajectories with a triangular profile of velocity, acceleration, etc. with $\max_t |q^{(i)}(t)| < q_{max}^{(i)}$ cannot be planned.

4.1 Derivatives of a generic trajectory

A trajectory generator should provide not only the position profile of the trajectory but also the related profiles of velocity, acceleration, jerk, etc. which are needed to implement feed-forward actions on the controlled system. While the closed-loop trajectory planner directly provides these signals, that are the components of the state vector $\boldsymbol{x}$, in the case of the cascade of n filters, it is necessary to purposely compute their value. However, the calculation of the derivatives of a trajectory $q_n(t)$ is straightforward by considering the definition (14). In fact,

$$\begin{array}{rcl} q_n^{(1)}(t) & = & q_{n-1}(t) * m_n^{(1)}(t) \\ & = & q_{n-1}(t) * \dfrac{1}{T_n}\Big(\delta(t) - \delta(t - T_n)\Big) \\ & = & \dfrac{1}{T_n}\Big(q_{n-1}(t) - q_{n-1}(t - T_n)\Big). \end{array} \tag{18}$$

The generic derivative of k-th order, can be calculated in a recursive manner as

$$q_n^{(k)}(t) = \frac{1}{T_n}\left(q_{n-1}^{(k-1)}(t) - q_{n-1}^{(k-1)}(t - T_n)\right) \tag{19}$$

with $q_{n-k}^{(0)}(t) = q_{n-k}(t)$. Figure 11 shows the block-scheme representation of the filter for the computation of the trajectory and its derivatives which has been obtained by iterating and Laplace transforming (19). Note that the filter of Fig. 11 gives a closed form expression (in terms of Laplace transform) of the derivatives and does not simply provide their numerical values.

4.2 Frequency characterization of the trajectory/filter

The expression of the trajectory $q_n(t)$ based on a cascade of linear dynamic filters allows to readily deduce its spectrum. As a matter of fact, by considering the Laplace transform of (14):

$$Q_n(s) = \frac{h}{s} \cdot M_1(s) \cdot M_2(s) \cdot \ldots \cdot M_n(s). \tag{20}$$

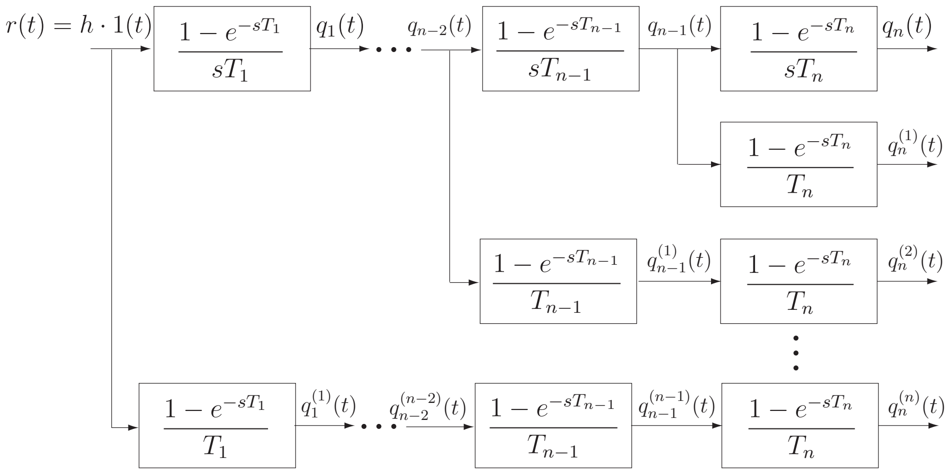

$r(t) = h \cdot 1(t)$
$\frac{1-e^{-sT_1}}{sT_1}$
$q_1(t)$
$q_{n-2}(t)$
$\frac{1-e^{-sT_{n-1}}}{sT_{n-1}}$
$q_{n-1}(t)$
$\frac{1-e^{-sT_n}}{sT_n}$
$q_n(t)$
$\frac{1-e^{-sT_n}}{T_n}$
$q_n^{(1)}(t)$
$\frac{1-e^{-sT_{n-1}}}{T_{n-1}}$
$q_{n-1}^{(1)}(t)$
$\frac{1-e^{-sT_n}}{T_n}$
$q_n^{(2)}(t)$
$\frac{1-e^{-sT_1}}{T_1}$
$q_1^{(1)}(t)$
$q_{n-2}^{(n-2)}(t)$
$\frac{1-e^{-sT_{n-1}}}{T_{n-1}}$
$q_{n-1}^{(n-1)}(t)$
$\frac{1-e^{-sT_n}}{T_n}$
$q_n^{(n)}(t)$

Figure 11: System composed by n filters for the computation of an optimal trajectory of class $\mathcal{C}^{n-1}$ and of all the derivatives of order $i = 1, \ldots, n$.

the Fourier Transform of $q_n(t)$ immediately descends, being the restriction of $Q_n(s)$ to the imaginary axis, i.e. $Q_n(j\omega)$. Therefore, the closed form expression of $Q_n(j\omega)$ is given by the products of the Fourier signal corresponding to the input $h \cdot 1(t)$ and the frequency responses of the filters composing the trajectory generator:

$$Q_n(j\omega) = \frac{h}{j\omega} \cdot M_1(j\omega) \cdot M_2(j\omega) \cdot \ldots \cdot M_n(j\omega)$$

where

$$\begin{aligned} M_i(j\omega) &= \frac{1}{T_i}\frac{1-e^{j\omega T_i}}{j\omega} \\ &= e^{-j\frac{\omega T_i}{2}} \frac{\sin\left(\frac{\omega T_i}{2}\right)}{\frac{\omega T_i}{2}}. \end{aligned} \tag{21}$$

Since the frequency characterization of trajectories, including their derivatives and, in particular, the acceleration [12], is a useful tool to predict vibratory phenomena in the systems to which the trajectories are applied, it is necessary to obtain the expression of the spectrum of the generic k-th derivative of $q_n(t)$. Because of the properties of Laplace transforms, this result is straightforward. As a matter of fact, the Laplace transform of $q_n^{(k)}(t)$ is simply given by

$$Q_n^{(k)}(s) = s^k Q_n(s)$$

and therefore the expression of the spectrum of $q_n^{(k)}(t)$ is

$$\begin{aligned} Q_n^{(k)}(j\omega) &= (j\omega)^k \cdot Q_n(j\omega) \\ &= h\cdot\, (j\omega)^{k-1}\cdot\, M_1(j\omega) \cdot M_2(j\omega) \cdot \ldots \cdot M_n(j\omega) \end{aligned}$$

In conclusion, the amplitude spectrum of $q_n(t)$ and its derivatives, i.e. $|Q_n^{(k)}(j\omega)|$, is given by the product of two main elements:

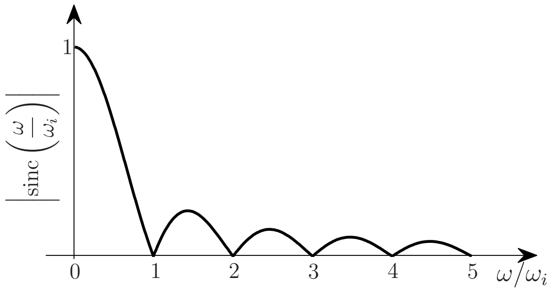


Figure 12: Magnitude of the frequency response of the filter $M_i(s)$.

- a power of ω, i.e. ω^{k-1}, being k the order of the derivative;
- the (magnitude of the) frequency response of the a chain of n filters $M_i(s)$.

The frequency response of the cascade of filters is given by the product of the single frequency responses $M_i(j\omega)$, $i = 1, \ldots, n$, whose magnitude is

$$|M_i(j\omega)| = \left|\frac{\sin\left(\frac{\omega T_i}{2}\right)}{\frac{\omega T_i}{2}}\right| = \left|\operatorname{sinc}\left(\frac{\omega}{\omega_i}\right)\right|$$

where sinc(·) denotes the normalized sinc function defined as $\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ and $\omega_i = \frac{2\pi}{T_i}$. Note that the function $|M_i(j\omega)|$, shown in Fig. 12, is equal to zero for $\omega = k\,\omega_i$, with k integer. This properties can be profitably exploited to properly chose the parameters of the trajctory/filter with the purpose of nullify the spectrum of the trajectory at critical frequencies, for instance the eigenfrequencies of the plant. For this aim, if ω_r denotes a resonant frequency, it is sufficient to assume

$$\omega_i = \frac{\omega_r}{l} \quad \Leftrightarrow \quad T_i = l\,\frac{2\pi}{\omega_r}, \quad l = 1, 2, \ldots. \tag{22}$$

Note that, with these considerations the same conclusions as in [13] has been reached and generalized. As a matter of fact, with reference to a double S speed trajectory in [13] it is recognized that in order to suppress residual vibrations due to the dominating vibratory mode of an axis of motion it is necessary to assume the duration of the "jerk period" (in which the jerk remains constant) equal to a multiple of the natural period of the vibrational mode. According to (22) the reduction of residual vibrations caused by resonant frequencies of the plant can be achieved with multi-segment trajectories of any order provided that the time constant T_i of a filter $M_i(s)$ is l times, l integer, the dominating natural period $\frac{2\pi}{\omega_r}$.

4.3 Evaluation of the trajectory by means of FIR filters

Although the expression of a generic trajectory is usually provided in the continuous-time domain by means of an analytic function of time t, it is generally used as reference signal for a computer controlled system and therefore needs to be evaluated at discrete time instants, which depend on the sampling time T_s of the controller. For this reason, like in the case of closed-loop trajectory planners, it is necessary to express the trajectory filter in the discrete-time domain. Starting from the expression of the continuous trajectory planner, obtained by connecting n filters $M_i(s)$ in a cascade configuration fed by a step function, it is possible to deduce an equivalent discrete-time system by discretizing the filters with one of the techniques available in the literature [14] and providing as input the sequence obtained by sampling with a period T_s the continuous step function. In particular the adoption of backward differences method leads to a digital filter composed by a chain of FIR filters, whose transfer function results

$$M_i(z) \quad = \quad \frac{1}{N_i} \frac{1 - z^{-N_i}}{1 - z^{-1}} \tag{23}$$

where

$$N_i = \frac{T_i}{T_s} \tag{24}$$

is the number of samples (not null) of the filter response, which is also equal to the number of elements composing the FIR filter (usually called *taps*) as they appear in the equivalent (nonrecursive) formulation

$$M_i(z) = \frac{1}{N_i} + \frac{1}{N_i} z^{-1} + \frac{1}{N_i} z^{-2} + \ldots + \frac{1}{N_i} z^{-N_i - 1}. \tag{25}$$

Note that (25) is the expression of a moving average filter, which averages the last N_i samples.

Finally, the expression of $Q_n(z)$ representing the discrete-time trajectory in the z-domain results

$$Q_n(z) = \frac{h}{1 - z^{-1}} \cdot M_1(z) \cdot M_2(z) \cdot \ldots \cdot M_n(z)$$

where $\frac{1}{1-z^{-1}}$ is the z-transform of

$$1(k) = \begin{cases} 1, & \text{for } k = 0, 1, 2, \ldots \\ 0, & \text{for } k < 0. \end{cases}$$

It is worth highlighting that the temporal sequence $q_n(k) = \mathcal{Z}^{-1}\{Q_n(z)\}$ only approximates the corresponding continuous-time trajectory $q_n(t)$ as shown in Fig. 13, where the samples of $q_3(k)$ are compared with the profile of the corresponding third order trajectory $q_3(t)$. In this case the sampling time $T_s = 0.02$ has been intentionally assumed quite large, if compared with the total duration of the trajectory, in order to point out the approximation error. However, it is possible to prove that when T_s goes to zero, such an error vanishes. From a practical point of view, this means that for sampling periods small enough, the sequence $q_n(k)$ can be used in lieu of the corresponding function $q_n(t)$ without appreciable differences, and therefore the bank of n FIR filters shown in Fig. 14 fed with sampled step functions (defining the desired final positions) can be adopted to generate online the trajectory of order n.

Also from a frequency point of view $Q_n(j\omega)$ is approximated by the spectrum of $Q_n(z)$, that can be computed as

$$\begin{aligned} Q_n(e^{j\omega T_s}) &= \frac{h}{1 - e^{-j\omega T_s}} M_1(e^{j\omega T_s}) M_2(e^{j\omega T_s}) \ldots M_n(e^{j\omega T_s}) \\ &\approx \frac{h}{j\omega T_s} M_1(e^{j\omega T_s}) M_2(e^{j\omega T_s}) \ldots M_n(e^{j\omega T_s}) \end{aligned} \tag{26}$$

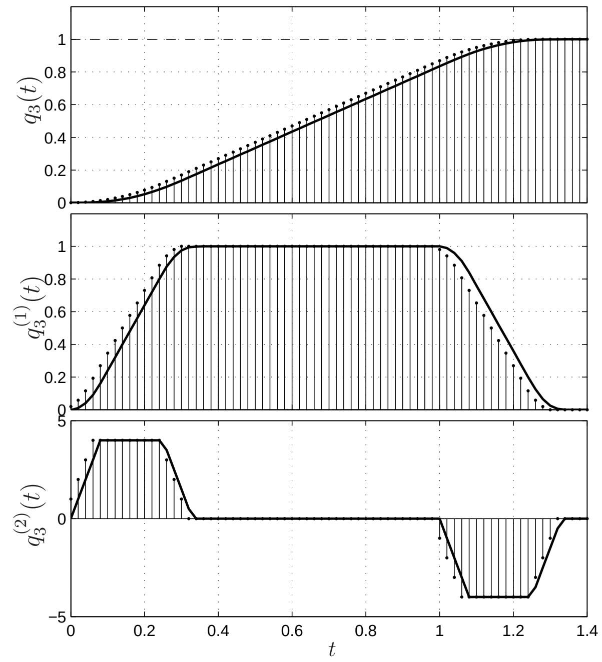


Figure 13: Comparison between the output of the third order continuous-time filter (solid line) and that of corresponding cascade of FIR filters (dots) obtained by discretization ($T_s = 0.02$).

with

$$\begin{aligned} M_i(e^{j\omega T_s}) &= \frac{1}{N_i}\frac{1-e^{-jN_i\omega T_s}}{1-e^{-j\omega T_s}} \\ &\approx \frac{1}{N_i}\frac{1-e^{-j\omega T_i}}{j\omega T_s} && (27) \\ &= \frac{1}{T_i}\frac{1-e^{-j\omega T_i}}{j\omega} && (28) \end{aligned}$$

where the Taylor series expansion of exponential function truncated at the first order, i.e.

$$e^{-j\omega T_s} \approx 1 - j\omega T_s$$

has been used. Note that because of the approximation, (26) and (27) are valid only only if ωT_s is small enough. As a consequence the smaller the sampling time T_s is, the wider the frequency range of validity of (26) will be. Within this range, the considerations of

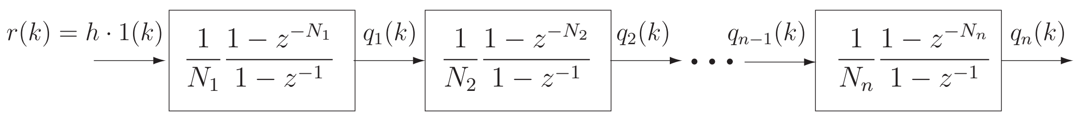

$r(k) = h \cdot 1(k)$
$\frac{1}{N_1}\frac{1-z^{-N_1}}{1-z^{-1}}$
$q_1(k)$
$\frac{1}{N_2}\frac{1-z^{-N_2}}{1-z^{-1}}$
$q_2(k)$
$q_{n-1}(k)$
$\frac{1}{N_n}\frac{1-z^{-N_n}}{1-z^{-1}}$
$q_n(k)$

Figure 14: System composed by n moving average filters for the computation of an optimal trajectory of class $\mathcal{C}^{n-1}$ at discrete time-instants kT_s.

Sec. 4.2 remain valid and therefore the desired trajectory $q_n(t)$ can be planned by means of the chain of discrete-time filters, whose characteristic parameters N_i are directly related to the periods T_i defining the trajectory by means of (24).

Finally, it is worth noticing that from a computational point of view the structure proposed in Fig. 14 for the generation of time-optimal trajectories results very efficient. As a matter of fact the i-th FIR filter is characterized by the differences equation

$$q_i(k) = q_i(k-1) + \frac{1}{N_i}(q_{i-1}(k) - q_{i-1}(k-N_i)), \quad i = 1, \ldots, n$$

and for the evaluation of q_i at the k-th sampling time two additions and one multiplication are necessary. Therefore the trajectory of order n requires n multiplications and $2n$ additions. Note that the general expression of multi-segment trajectories based on polynomials is

$$q_n(t) = a_{i,n}t^n + a_{i,n-1}t^{i,n-1} + a_{i,1}t + a_{i,0}, \quad i = 1, \ldots, 2^n - 1$$

where the coefficients $a_{i,j}$ must be properly computed for each of the 2^n-1 tracts. Obviously some of these coefficients are null in specific segments, but in the worst case an efficient evaluation[1] of the trajectory for a given value of t needs at least n multiplications and n additions. Moreover, in the case of the direct evaluation of the the analytic expression of the trajectory it is necessary a search algorithm to determine which segment must be considered at a specific value of time t and a switch statement to apply a different expression for each tract. For this reason the expression based on FIR filters is preferable to the standard analytic expression of multi-segment trajectories both in terms of implementation complexity and computational costs.

4.4 Applicative examples

The trajectory generator composed by n running average filters owns two main features which make it superior over classical pre-planned optimal trajectories, namely

- the possibility of planning the trajectory online by simply changing the input signal composed by step functions;
- the clear frequency characterization of the trajectory obtained as output of the filters' chain which makes possible choosing the value of the characteristic parameter T_i of each filter on the basis of the desired trajectory spectrum.

These features are now exploited to perform tasks that with the analytic expression of optimal trajectory would require complicated procedures while are immediate with FIR filters.

4.4.1 Multi-point optimal trajectories

The planning of complex trajectories composed by several segments joining a set of via-points p_j, $j = 0, \ldots, m$ can be simply achieved by feeding the discrete-time filter of Fig. 14

[1] For the estimation of the complexity of the polynomial evaluation the so-called Horner formula has been assumed.

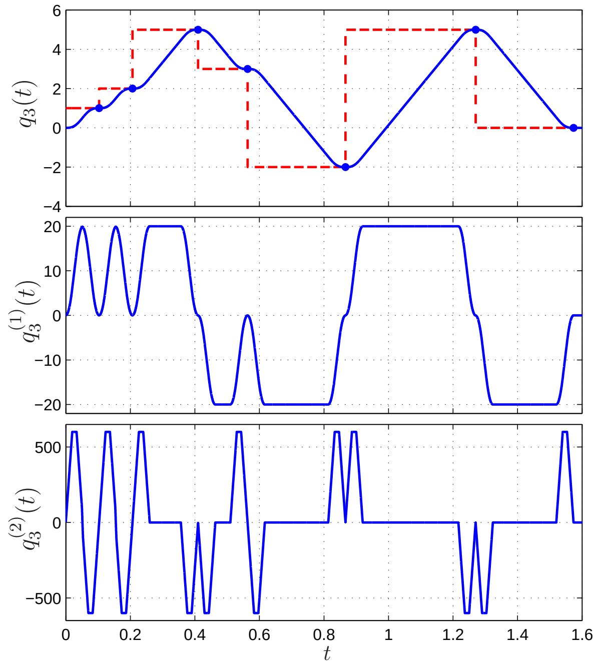


Figure 15: Complex motion obtained with a third order trajectory passing through a sequence of via-points $\boldsymbol{p} = \{0, 1, 2, 5, 3, -2, 5, 0\}$, with the constraints $q^{(1)}_{max} = 15$ m/s, $q^{(2)}_{max} = 600$ m/s^2, $q^{(3)}_{max} = 30000$ m/s^3.

with a staircase function whose constant values are the desired (final) positions p_j. For instance, in Fig. 15 a motion profiles composed by a sequence of third order trajectories $q_3(t)$ (double S speed trajectories) is shown. The trajectory generator is formed by three running average filters, defined by the time constants

$$
\begin{aligned}
T_{1,j} &= \frac{|h_j|}{q^{(1)}_{max}} = \frac{|h_j|}{15} \text{ s} \\
T_2 &= \frac{q^{(1)}_{max}}{q^{(2)}_{max}} = 0.025 \text{ s} \rightarrow N_2 = \frac{0.025}{0.0001} = 250 \\
T_3 &= \frac{q^{(2)}_{max}}{q^{(3)}_{max}} = 0.02 \text{ s} \rightarrow N_3 = \frac{0.02}{0.0001} = 200
\end{aligned}
$$

where $h_j = p_j - p_{j-1}$, $j = 1, \ldots, m$ represents the displacement of the j-th tract and the sampling time is $T_s = 0.0001$s. Note that, the limit values of velocity, acceleration, jerk, are generally constant and therefore N_2 and N_3 are fixed before the motion starts. Conversely, in order to guarantee a constant velocity with different displacements h_j, it is necessary to change online the structure of the first FIR filter composing the trajectory planner. In particular, it is needed to adapt the number of taps composing the filter to the desired displacement value h_j, according to the formula $N_{1,j} = \text{round}\left(\frac{|h_j|}{T_s \cdot q^{(1)}_{max}}\right)$. Obviously, this operation must be performed whenever the input signal defining the final position changes and it is necessary to guarantee that the FIR filter's output remains constant. Since a new via-points is given only when the previous one has been reached, when the input changes all the FIR filters (and in particular the first one) are in a steady-state condition. For the first filter fed with constant signals, this means that both the output and all the internal states are equal to the input value. As a consequence, when additional taps are added to $M_1(z)$, in order to keep the output unchanged, it is necessary to set the values of the new internal states equal to those of the existing states. When some taps are eliminated, the values of residual internal states are not modified.

With this simple procedure, a sequence of double S speed trajectories compliant with the desired constraints and crossing the given set of points can be planned online by applying to the chain of $n = 3$ moving average filters the signal obtained by sampling with a period T_s the continuous-time function

$$r(t) = \sum_{j=1}^{m} p_j \cdot 1(t - t_j)$$

where t_j is the starting time-instant of the j-th trajectory, that can be computed recursively from t_{j-1} according to

$$t_j = t_{j-1} + T_{1,j-1} + T_2 + T_3, \text{ with } t_1 = 0. \tag{29}$$

4.4.2 Multi-segment trajectories with frequency specifications

In previous example the parameters of the trajectory generator are obtained on the basis of constraints (velocity, acceleration, jerk) expressed in the time-domain, but as already mentioned the structure of the filter makes it particularly suitable for taking into account frequency constraints, which may arise because of critical frequencies of the plant that must track the motion profile. In this way, it is possible to combine the advantages of time-optimal multi-segment trajectories with those of the approaches that filter the input trajectories in order to properly shape their spectrum, see [7] for a comprehensive overview on this argument. It is worth noticing that closed-loop trajectory filters, based on nonlinear controllers, do not provide a frequency characterization of the output on the basis of the input and of the filters parameters and therefore cannot take into account frequency specifications.

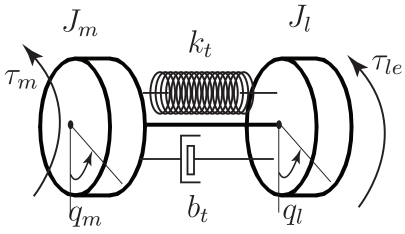
In order to demonstrate the effectiveness of the proposed procedure for trajectory planning we consider the standard motion system shown in Fig. 16, composed by two inertias with an elastic transmission lightly damped [15, 16, 17], whose model (from the motor position q_m to the load position q_l) corresponds to the following transfer function

$$G_{ml} = \frac{Q_l(s)}{Q_m(s)} = \frac{2\delta\omega_n s + \omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \tag{30}$$

with

$$\omega_n = \sqrt{\frac{k_t}{J_l}}, \qquad \delta = \frac{b_t}{2\sqrt{k_t J_l}}.$$

The parameter of the system, reported in Fig. 16, are derived from [17], as well as the trajectory constraints ($q_{max}^{(1)} = 250$ rad/s, $q_{max}^{(2)} = 5000$ rad/s^2, $q_{max}^{(3)} = 5 \times 10^5$ rad/s^3).



Parameter	Symbol	Value	Unit
Motor inertia	J_m	0.72×10^{-5}	$\mathrm{kg\,m^2}$
Load inertia	J_l	0.23×10^{-5}	$\mathrm{kg\,m^2}$
Spring stiffness	k_t	0.156	$\mathrm{N\,m}$
Internal damping	b_t	1.0×10^{-5}	$\mathrm{N\,m\,s}$

Figure 16: Lumped constant model of a motion system with elastic transmission and values of parameters.

With these parameters the resonant frequency of the system is $\omega_r \approx \omega_n = 260$ rad/s.

We suppose that an ideal control system imposes to the (rotor) inertia J_m the desired motion profile, that is $q_m(t) = q_{ref}(t)$, being $q_{ref}(t)$ a trajectory obtained with the trajectory generator, and we analyze the effect of a particular choice of the filter's parameters on the dynamic behavior of the plant and in particular on the the tracking error, defined as $\varepsilon(t) = q_{ref}(t) - q_l(t) = q_m(t) - q_l(t)$. Obviously, the choice of the parameters of the filter is critical only when the spectral components of the trajectory are appreciable in the neighborhood of the eigenfrequency of the plant. If this occurs, the design of the trajectory neglecting the dynamic characteristics of the plant may lead to large tracking errors and residual vibrations when the motion should stop. For instance, the response of the motion system $G_{ml}(s)$ to a trapezoidal velocity trajectory obtained by assuming $T_1 = 0.064$s and $T_2 = 0.032$s ($T_{tot} = 0.096$s) is shown in Fig. 17(a) along with the error $\varepsilon(t)$. Note the considerable value of the error at the end of motion, in particular if compared with the error obtained by applying to the system the trapezoidal velocity trajectory of Fig. 18, characterized by the same total duration $T_{tot} = 0.096$s but obtained with $T_1 = 3T_0$ and $T_2 = T_0$ (being $T_0 = \frac{2\pi}{\omega_r} = 0.0242$ s the natural period of $G_{ml}(s)$). By analyzing the spectral contents of the two trajectories, and in particular of the acceleration profiles, it is possible to easily explain such results. Note that the dynamic relation between the reference trajectory and the tracking error ε, obtained from (30) after simple algebraic manipulations, is given by

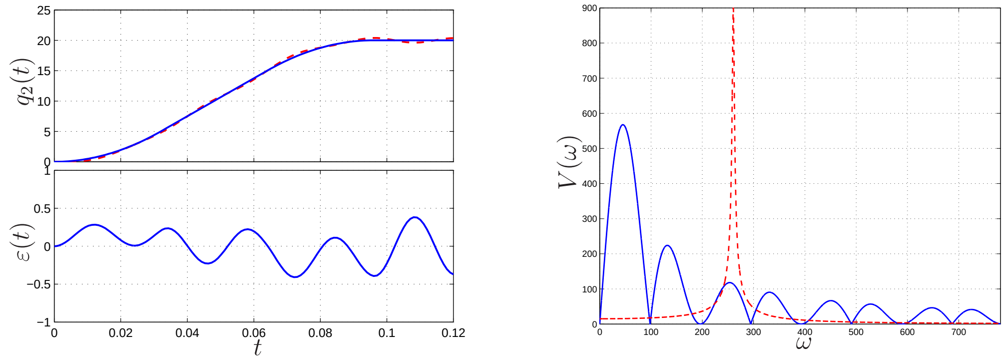
$$G_\varepsilon(s) = \frac{s^2}{s^2 + 2\delta\omega_n s + \omega_n^2}$$

that is ε can be obtained by applying the second derivative of the trajectory, i.e. the acceleration profile, to a second order system (without zeros) characterized by a natural frequency ω_n and a damping ration δ. For this reason in Fig. 17(b) and Fig. 18(b) the spectrum $|Q_n^{(2)}(j\omega)|$ of the acceleration profile, hereafter denoted with $V(\omega)$, is compared with the magnitude of the frequency response of $G_\varepsilon(s)$ (properly scaled for the sake of clarity). From considerations of Sec.4.2 it follows that the parameters

$$T_1 = 3T_0 \Leftrightarrow \omega_1 = \frac{\omega_r}{3}$$

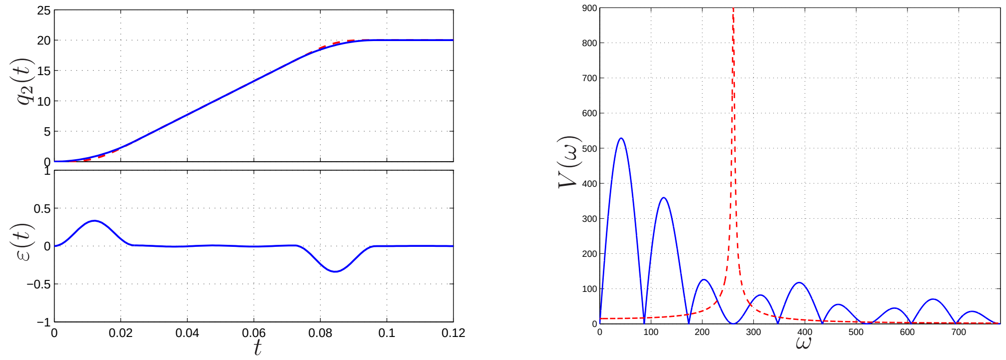
and

$$T_2 = T_0 \Leftrightarrow \omega_2 = \omega_r$$



(a) (b)

Figure 17: Response of the elastic system G_{ml} to a trapezoidal velocity trajectory obtained with $T_1 = 0.064$s and $T_2 = 0.032$s: tracking error (a) and frequency spectrum of the acceleration (b).



(a) (b)

Figure 18: Response of the elastic system G_{ml} to a trapezoidal velocity trajectory obtained with $T_1 = 3T_0$ and $T_2 = T_0$: tracking error (a) and frequency spectrum of the acceleration (b).

lead to $|M_1(j\omega_r)| = 0$, and $|M_2(j\omega_r)| = 0$, and therefore they introduce in $V(\omega)$ a zero of multiplicity two for $\omega = \omega_r$. This implies that not only

$$V(\omega_r) = 0,$$

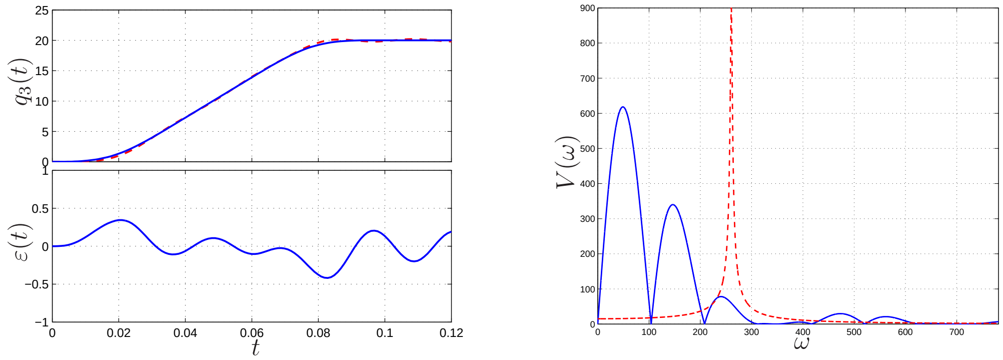
but also

$$\frac{dV(\omega)}{d\omega}_{|\omega=\omega_r} = 0$$

and, as a consequence, in the neighborhood of the resonant frequency ω_r the slope of the function $V(\omega)$ is small and therefore the function remains limited in a broader range of frequencies.

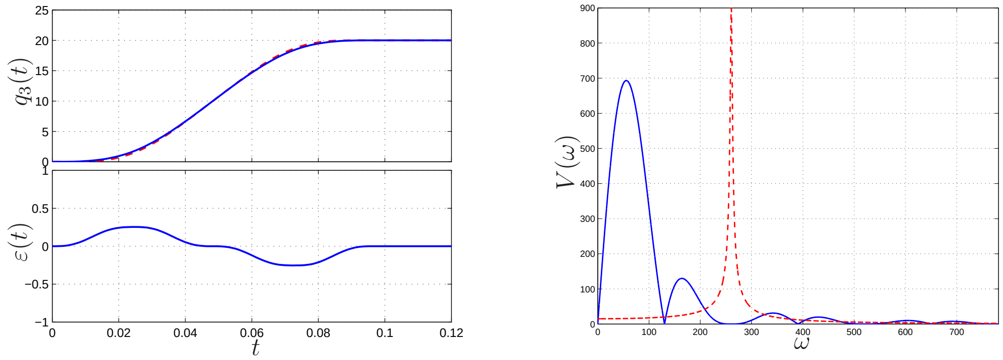
The use of double S velocity trajectories (with limited jerk) can further improve the results in terms of magnitude of the tracking error as highlighted in a number of work, see [16, 15, 17].

But also in this case the choice of the filter/trajectory parameters has a strong influence on the system output, as shown in Fig. 19 and Fig. 20, where two double S velocity trajectories of the same duration are compared. In particular, a choice of the time constants T_1, T_2,



(a) (b)

Figure 19: Response of the elastic system G_{ml} to a double S velocity trajectory obtained with $T_1 = 5/2T_0$, $T_2 = 3/4T_0$ and $T_3 = 3/4T_0$: tracking error (a) and frequency spectrum of the acceleration (b).



(a) (b)

Figure 20: Response of the elastic system G_{ml} to a double S velocity trajectory obtained with $T_1 = 2T_0$, $T_2 = T_0$ and $T_3 = T_0$: tracking error (a) and frequency spectrum of the acceleration (b).

T_3 that does not take into account the presence of a resonant peak into the system[2] does not produce any improvement in the tracking performances with respect to lower order trajectories such as trapezoidal velocity trajectories, see Fig. 19. On the contrary, by assuming the parameters $T_1 = 2T_0$, $T_2 = T_0$ and $T_3 = T_0$ the spectrum of the trajectory $V(\omega_r)$ has a zero of multiplicity three for $\omega = \omega_r$, and therefore

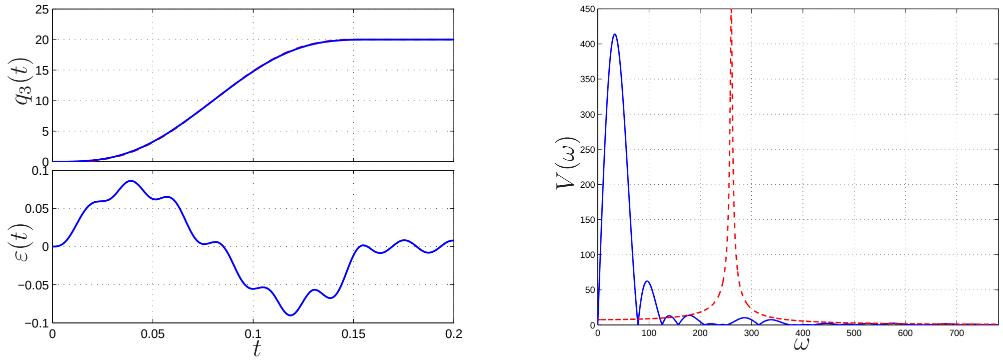
$$V(\omega_r) = \frac{dV(\omega)}{d\omega}\Big|_{\omega=\omega_r} = \frac{d^2V(\omega)}{d\omega^2}\Big|_{\omega=\omega_r} = 0.$$

[2] Note that the values

$$T_1 = \frac{5}{2}T_0 \Leftrightarrow \omega_1 = \frac{2}{5}\omega_r$$

$$T_2 = T_3 = \frac{3}{4}T_0 \Leftrightarrow \omega_2 = \omega_3 = \frac{4}{3}\omega_r$$

do not guarantee that $V(\omega_r) = 0$, as shown in Fig. 19(b).



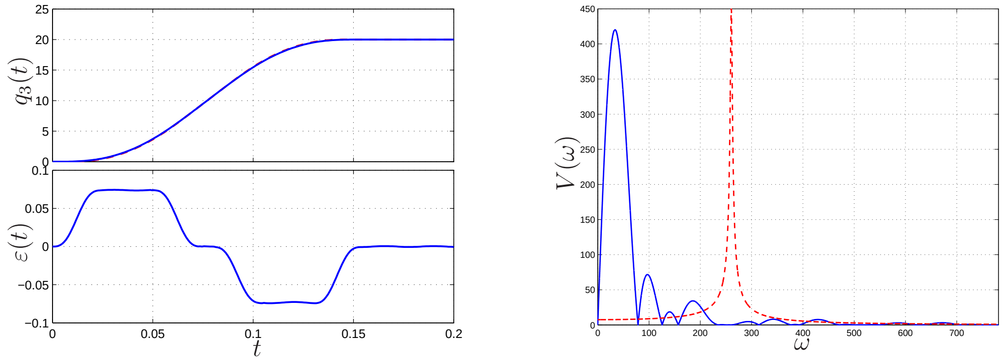
(a) (b)

Figure 21: Response of the elastic system G_{ml} to a double S velocity trajectory obtained with $T_1 = |h|/q^{(1)}_{max}$, $T_2 = q^{(1)}_{max}/q^{(2)}_{max}$ and $T_3 = T_1 - T_2$: tracking error (a) and frequency spectrum of the acceleration (b).

From a practical point of view, this means that the values of the function $V(\omega)$ in the neighborhood of the resonant frequency are smaller than those of the trapezoidal velocity trajectory of fig. 18, whose parameters T_i are obtained with similar considerations. As a consequence, the tracking error of the plant is reduced. In particular for the trapezoidal velocity trajectory the maximum value of the error is $\max|\varepsilon_{tr}| = 0.3395$, while for the double S velocity trajectory $\max|\varepsilon_{2S}| = 0.2536$, with a reduction $\Delta|\varepsilon| \approx -25\%$.

In the examples discussed so far, only the constraints due to the dynamic behavior of the plant have been considered, while the bounds velocity, acceleration, etc. have not been taken into account. As a consequence, the peak values of $q^{(1)}(t)$, $q^{(2)}(t)$, etc. depend on the choice of the parameters T_i. For instance, in the case of the double S speed trajectory of Fig. 20(a), the values of such parameters lead to $q^{(1)}_{max} = 276$ rad/s (being the displacement $h = 20$rad), $q^{(2)}_{max} = 11454$rad/s^2, $q^{(3)}_{max} = 474750$rad/s^3. However, the interpretation of multi-segment trajectories as a bank of filters allows to combine temporal and frequency constraints. This feature must be profitably exploited when the actuation system imposes some physical limits and the load introduces undesired dynamical modes. For instance, with reference to the plant $G_{ml}(s)$, if the actuation system is capable of providing a maximum speed $q^{(1)}_{max}$ and a maximum acceleration $q^{(2)}_{max}$, a minimum time double S velocity trajectory can be obtained by assuming $T_1 = h/q^{(1)}_{max}$ and $T_2 = q^{(1)}_{max}/q^{(2)}_{max}$, while the parameter T_3 is set to the minimum value compliant with constraints (17), that is $T_3 = T_1 - T_2$. However, although the error is about one order smaller than the error of trajectories of Fig. 18 and Fig. 20 obtained by tacking into account only the dynamical model of the plant[3], it exhibits some oscillations when the trajectory stops, as highlighted in Fig. 21(a). In order to reduce these residual vibrations one can set the free parameters T_3 to T_0 in order to make $V(\omega)$ null for $\omega = \omega_r$, see Fig. 22. In this way, the resonant mode of the plant is not excited and, at the end of the motion, the error ε goes immediately to zero. Obviously the bounds

[3] This is due to the total duration of the trajectory obtained with temporal constraints, $T_{tot} = 0.16$s, which is pretty higher than the duration of the other trajectories, $T_{tot} = 0.0965$s. For this reason the magnitude of spectral components at high frequencies is considerably reduced.



(a) (b)

Figure 22: Response of the elastic system G_{ml} to a double S velocity trajectory obtained with $T_1 = |h|/q^{(1)}_{max}$, $T_2 = q^{(1)}_{max}/q^{(2)}_{max}$ and $T_3 = T_0$: tracking error (a) and frequency spectrum of the acceleration (b).

on velocity and acceleration are satisfied, as shown in Fig. 23.

In conclusion, the formulation of multi-segment trajectories based on dynamic open-loop filters allows to consider both temporal and frequency specifications. In particular, if bounds on the first m derivatives $q_n^{(i)}(t)$, $i = 1, \ldots, m$ are given, and additionally it is necessary that $V(\omega) = 0$ for l critical frequencies $\omega_{r,i}$, $i = 1, \ldots, l$, the trajectory order must be assumed $n = m + l$. Then, the former m parameters T_i must be selected according to (16), while the latter l parameters on the basis of (22). In general, the durations T_i, $i = m + 1, \ldots, m + l$ are lower than T_i, $i = 1, \ldots, m$. If this does not occur, it means that the constraints due to dynamical reasons are not consistent with the one or more of the limits on velocity, acceleration, jerk, etc. but it may be necessary to reduce such limit values (alternatively one can neglect such bounds that are met in any case, because of the frequency constraints).

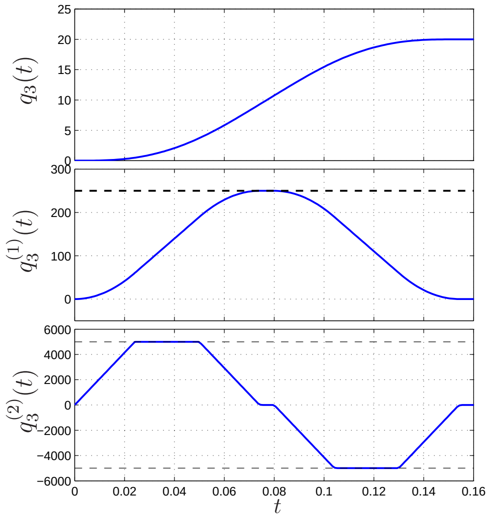
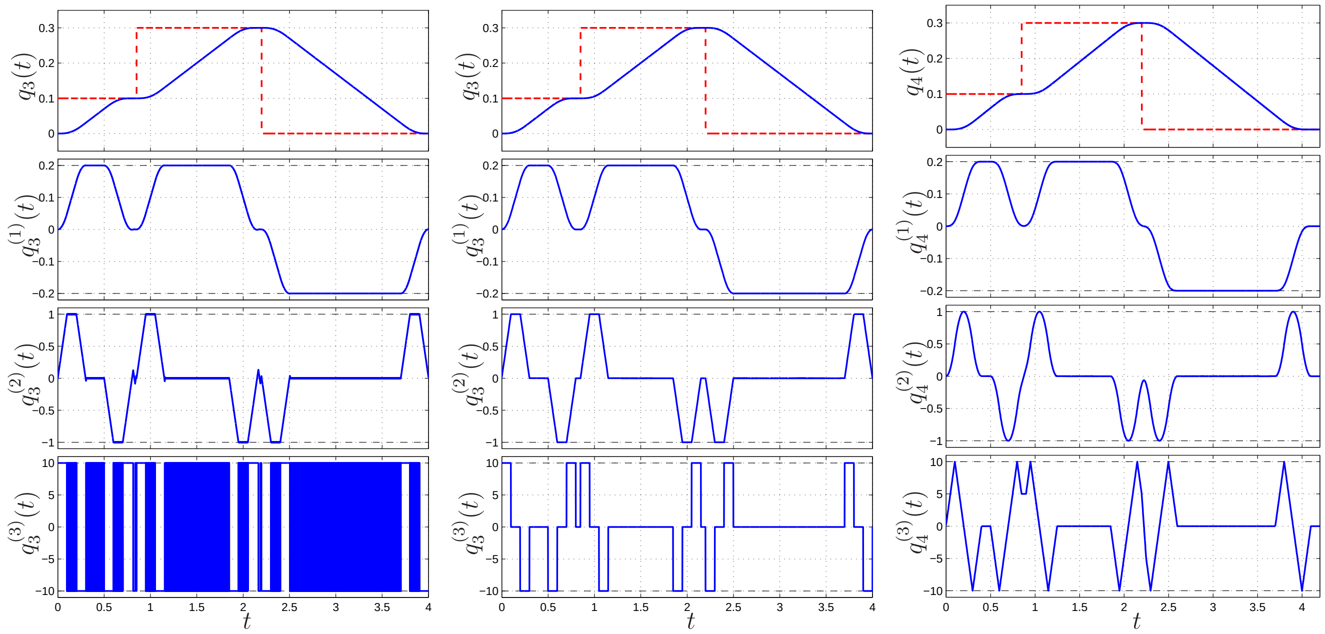


Figure 23: Profiles of position, velocity and acceleration of the trajectory considered in Fig. 22.

5 A comparison between closed-loop and open-loop filters

In this section, the main differences between closed- and open-loop trajectory planners are highlighted, in particular, with respect to their performances and the different problems that they can solve. As already mentioned, closed-loop filters are superior to analogous open-loop systems in terms of capacities and flexibility (asymmetric bounds, no additional constraints among the filter parameters such as (18), possibility of change the desired limits in runtime) but they are also affected by some limitations (presence of chattering superimposed to the output, high complexity of the implementation, high computational burden, maximum order $n = 3$). For this reason the choice of a particular type of filter must be performed according to the specific application to be carried out. For instance, in standard tasks where online generation of point-to-point trajectories is required, if the desired bounds of velocity, acceleration, jerk, etc. meet the conditions (18), open-loop filters are preferable. As a matter of fact, as reported in Fig. 24(a) and Fig. 24(b) the two generators provide the same trajectory but with very different costs in terms of computational complexity. Moreover, in Fig. 24(a) one can easily observe the oscillations in the jerk profile and the overshoots in the acceleration. Figure 24(c) shows the fourth order trajectory $q_4(t)$ obtained by applying the same stepwise input function to a chain of four running average filters. With a little increase of complexity, one can obtain the time-optimal trajectory with the desired degree of smoothness.

On the other hand, if the input function changes before that the previous tract of trajectory ends, the open-loop generator cannot guarantee the compliance with the constraints, see Fig. 25(b). On the contrary, closed-loop filters always fulfill the prescribed constraints, as shown in Fig. 25(b). Moreover, as already mentioned, in case of feedback controlled trajectory generators, it is possible to consider asymmetric limits, i.e. $q_{max}^{(i)} \neq -q_{min}^{(i)}$, or even change the values of the bounds in runtime. In this case, the controller will act so



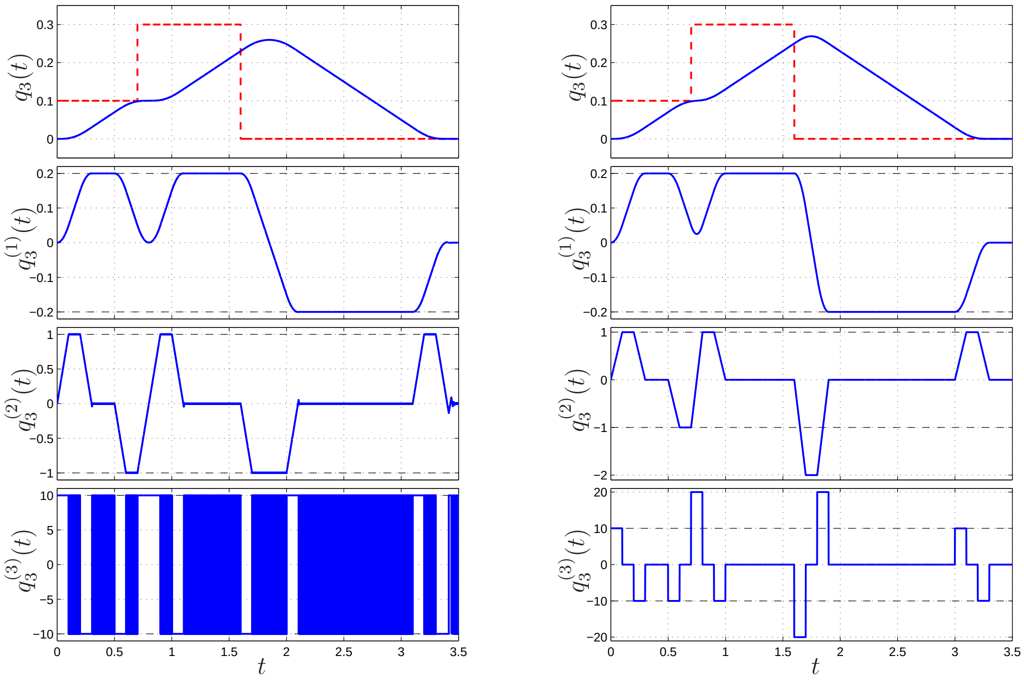
(a) (b) (c)

Figure 24: Comparison between the output of the third order closed-loop filter (a) and those of a third order (b) and fourth order (c) open-loop filter with the same input $r(t)$ composed by step functions.

that the new limits are satisfied in minimum time. For instance, in Fig. 26(b) the behavior of the filter is shown when the minimum value of velocity is increased from -0.2 to -0.1. As soon as this occurs, the controller modifies the velocity of the motion profile in order to meet the new limit but without violating the other constraints on acceleration and jerk. Conversely, by adopting an open-loop generator it is not possible to change the constraints during the planning of a trajectory, since this requires a structural modification of the filters composing it, leading to discontinuities of the response. In fact, the number of taps of the FIR filters is directly related to the desired bounds by means of (24).

Finally, both closed- and open-loop filters can be used to modify pre-planned trajectories with the purpose of making them compliant with the desired bounds and not only to generate point-to-point multi-segment trajectories. Also in this case the two types of dynamic systems behave quite differently. In Fig. 27 the responses to a reference input $r(t)$ composed by ramp functions (therefore with constant velocity and impulsive acceleration) are reported. The closed-loop filters tries to reach in minimum-time and then to track the reference $r(t)$, satisfying the given constraints. On the contrary, the third order open-loop filter produces a smoother but delayed version of the input. Note that the output signal reaches the constant velocity of the ramps but that all the other constraints are not reached. Moreover, since the input signal is continuous the resulting trajectory has continuous jerk. Therefore it is possible to consider a second order trajectory filter, with the parameters

$$T_1 = \frac{q^{(1)}_{max}}{q^{(2)}_{max}}, \qquad T_2 = \frac{q^{(2)}_{max}}{q^{(3)}_{max}}$$



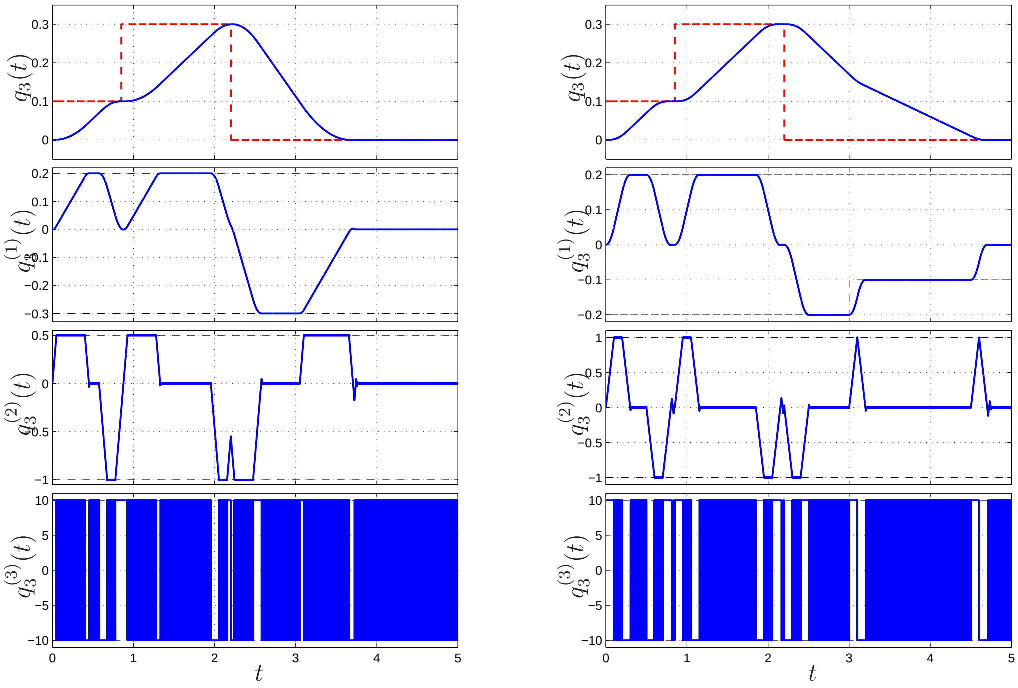
(a) (b)

Figure 25: Comparison between the output of the third order closed-loop filter (a) and that of a cascade of three moving average filters (b) with the same input $r(t)$ composed by step functions, that are applied in time instants that are not consistent with (29).

to assure that all the constraints are satisfied. In this way, the total (constant) delay between the input and the output is reduced, being $T_{tot} = T_1 + T_2$, see Fig. 27(c).

Despite the hypothesis $r^{(4)} = 0$ assumed in the design of the third order closed-loop filter, the variable structure control allows to consider input signals $r(t)$ that do not fulfill this condition. In Fig. 28(a) the response of the filter to a generic input is illustrated. When the desired bounds are met the output follows exactly the input but when the reference overcomes such values the filter limits the output. In this case the reference input may represent the commands provided by an external system or a human operator, and the trajectory filter can be profitably exploited to make them compatible with the physical limits that are present in any plant.

On the contrary open-loop filters behave like a standard low-pass filter, and therefore if the input signal is "too fast" the output will result quite deformed (and delayed), see Fig. 28(b). As a consequence, the chain of FIR filters is not suitable for an application to generic inputs, even if the possibility to select the filters'parameters on the basis of frequency specifications can be of great help when it is necessary to plan time-optimal trajectories for plants with critical frequencies.



(a) (b)

Figure 26: Response of the third order closed-loop filter to a stepwise input function under asymmetric constraints of velocity and acceleration (a) and with a variation of the bound on the velocity (b).

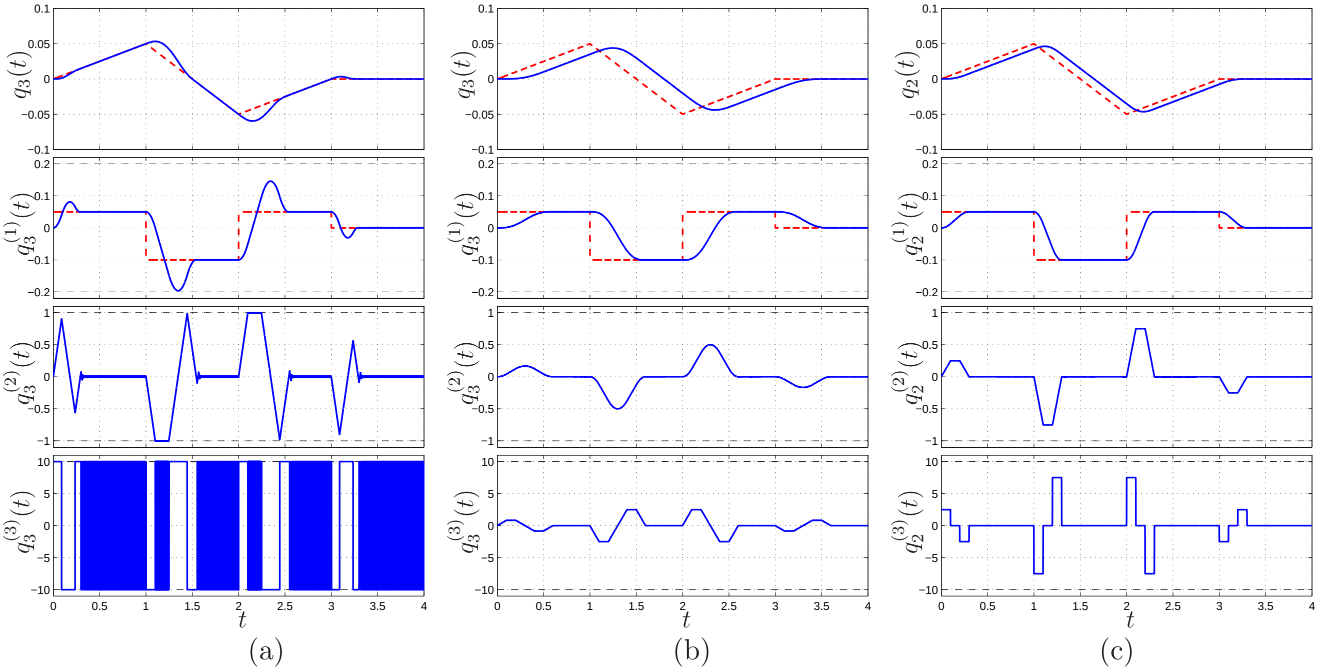
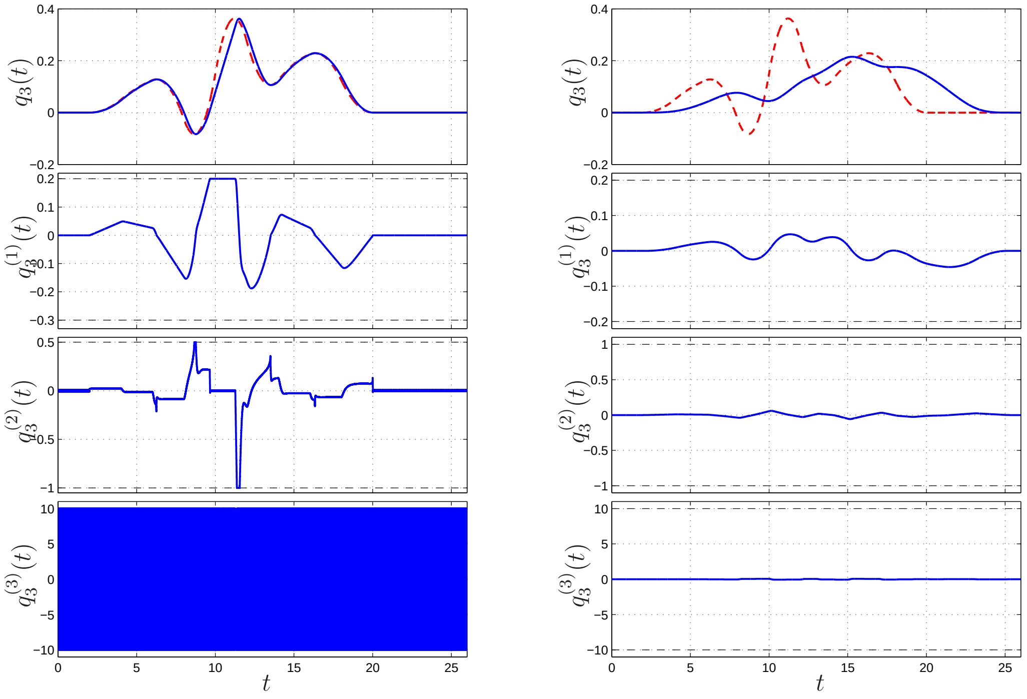


Figure 27: Comparison between the output of the third order closed-loop filter (a) and those of a third order (b) and a second order (c) open-loop filter with the same input $r(t)$ composed by ramp functions.



(a) (b)

Figure 28: Comparison between the output of the third order closed-loop filter (a) and that of a cascade of three moving average filters (b) with a generic input $r(t)$.

6 Conclusions

In this paper, generation of time-optimal trajectories by means of dynamical filters is considered. In particular, the different closed- and open-loop schemes are analyzed and compared. Trajectory generator with a feedback loop can be used to filter generic input signals with the purpose of make them compliant with the given bounds on velocity, acceleration, jerk, etc.. The filter acts so that the output signal follows the input with no delay if all the constraints are satisfied. Otherwise, the output is limited. In case of a stepwise input function the output is a multi-segment time-optimal trajectory. The same trajectory can be obtained by applying a step signal to a chain of FIR filters, characterized by a simple structure and a low computational cost that make them ideal for point-to point motions, even with an high degree of smoothness ($n = 4$ or 5).

References

[1] 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.

[2] 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.

[3] 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.

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

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

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

[7] N.C. Singer, W.E. Singhose, and W.P. Seering. Comparison of filtering methods for reducing residual vibration. *European Journal of Control*, 5:208–218, 1999.

[8] L. Biagiotti and R. Zanasi. Time-optimal regulation of a chain of integrators with saturated input and internal variables: an application to trajectory planning. In *NOLCOS 2010, 8th IFAC Symposium on Nonlinear Control Systems*, Bologna, September 2010.

[9] C. Guarino Lo Bianco and R. Zanasi. Smooth profile generation for a tile printing machine. *Industrial Electronics, IEEE Transactions on*, 50(3):471 – 477, jun. 2003.

[10] 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.

[11] O. Gerelli and C.G.L. Bianco. A discrete-time filter for the on-line generation of trajectories with bounded velocity, acceleration, and jerk. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3989 –3994, may. 2010.

[12] L. Biagiotti and C. Melchiorri. *Trajectory Planning for Automatic Machines and Robots*. Springer Berlin Heidelberg, 2008.

[13] A. Olabi, R. Bare, O. Gibaru, and M. Damak. Feedrate planning for machining with industrial six-axis robots. *Control Engineering Practice*, 18(5):471–482, May 2010.

[14] G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital Control of Dynamic Systems*. Ellis-Kagle Press, third edition, 1998.

[15] 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*, 42:275–293, 2005.

[16] P.H. Meckl and P.B. Arestides. Optimized s-curve motion profiles for minimum residual vibration. In *Proceedings of the American Control Conference*, pages 2627–2631, Philadelphia, Pennsylvania, June 1998.

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