An Efficient Algorithm for the Real-Time Generation of Synchronous Reference Signals

Corrado Guarino Lo Bianco, *IEEE*, *Member*

***Abstract*—Discontinuous reference signals reduce the performances of feedback control systems. This paper proposes an algorithm which generates smooth and synchronous set-points for electromechanical systems, starting from rough input signals. This result is achieved by filtering the discontinuous signals through a set of nonlinear filters whose output is given by smooth profiles subject to velocity and acceleration constraints. The reference synchronization is obtained by acting on the velocity bounds and can be customized to accomplish alternative tasks. In this work, for example, it is used to reduce friction power losses which affect electromechanical systems. The proposed algorithm is extremely compact and efficient, so that it can be implemented on control boards having limited memory and limited computational capabilities.**

***Index Terms*—Energy saving, Motion control, Synchronization, Tracking filters.**

I. Introduction

MOTION profiles for automatic machines are very commonly generated through real-time planners. Indeed, in many industrial applications reference signals cannot be planned in advance, being function of events that cannot be predicted. Early real-time planners were related to single-axis tasks [1] but, especially in robotic contexts, a number of real time planners for multi-axis systems [2]–[8] sharing common characteristics have recently appeared: generated trajectories are typically minimum-time and fulfill constraints on velocity, acceleration, and jerk.

Another type of generators have been developed in parallel with those previously mentioned in order to accomplish a slightly different task. In several applications, indeed, the reference signal is generated on-the-fly depending on the evolution of a master process. Such reference can be non-repetitive and discontinuous. Discontinuities are known to represent a problem for actuators, so reference signals must be suitably smoothed. Fig. 1 shows a typical situation in which a sequence of discontinuous set-points (dashed lines) is joined by means of minimum-time trajectories admitting bounded velocities and accelerations (solid lines). With some adaptations, real-time planners [2]–[8] could potentially be used to create the required linking profiles, but when such profiles must be generated through systems with limited computational capabilities like, e.g., microcontrollers for servo motor drives, other lightweight solutions are typically preferred. In repetitive contexts, smooth profiles can be planned offline. For example, in [9]–[11] linking curves were used to reduce system overshoots or oscillations. When discontinuous reference signals are not known in advance, the smoothing action is obtained through real-time methods. To this purpose, several strategies have been proposed in the literature based, for example, on the use of FIR filters [12], [13] or on neuro-fuzzy algorithms [14].

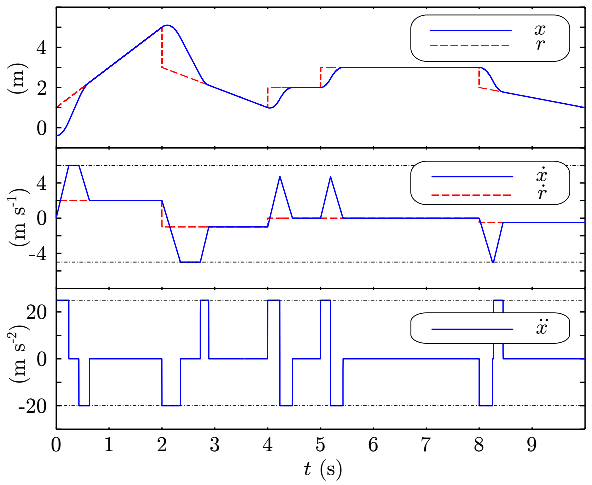

Fig. 1. A non-smooth reference signal r (dashed lines) is converted into a kinematically bounded signal x (solid lines). Dash-dotted lines indicate the velocity and the acceleration bounds.

A further class of real-time methods is based on the use of nonlinear feedback filters. The solutions proposed over the years have sometimes been conceived for continuous-time environments [15]–[18] or, in other contexts, for discrete-time applications [19]–[22].

The approaches based on FIR or nonlinear filters have several advantages. First of all, they are computationally efficient since filters are based on closed form equations. Secondly, filters can be implemented through extremely compact programs. For example, nonlinear filters which generate acceleration bounded profiles can be implemented with, roughly, twenty lines of C code, while jerk bounded versions, which allow smoother transients, but admit slightly higher evaluation times, require less than one hundred lines. Because of these characteristics, the smoothing filters can be implemented on control boards with limited computational capabilities like, e.g., embedded control boards or PLCs. Performances of FIR filters are generally comparable with those of nonlinear filters,

Manuscript received October 20, 2015; revised January 22, 2016 and March 17, 2016; accepted April 14, 2016.

C. Guarino Lo Bianco is with the Dipartimento di Ingegneria dell'Informazione, University of Parma, Italy; phone: +39 0521 905752, fax: +39 0521 905723; email: guarino@ce.unipr.it.

but the latter can also manage situations in which the given reference signals or the bounds change during transients: a novel trajectory is immediately generated toward the new reference by accomplishing the new limits. This is possible due to the use of internal controllers based on sliding mode techniques which, as known [23]–[25], robustly stabilize feedback systems. If the fact that generated transients are minimum-time and that reference signals are always reached without overshoot is taken into consideration, it becomes clear why this kind of filters has found application in several industrial or academic contexts [26]–[31].

In this paper, the filter synthesized in [21] is used to solve the following problem: given a single process involving several actuators whose reference signals are originally asynchronous and discontinuous, can they all be smoothed and synchronized by keeping the system productivity unchanged and improving the power efficiency? “Synchronized” is used here to indicate that, in case of transients, all generated smooth signals must join their corresponding references at the same sample time. The problem is motivated by a simple consideration: if all rough set-points refer to the same process, for example the manipulation of a single object, operations on the same object can only start after the end of the slowest transient. This implies that the fastest transients can be slowed down with no impact on the system productivity. The degrees of freedom achieved can be used to optimize a given cost index. In this paper, for example, since power efficiency represents an important issue [32]–[36], they will be used to reduce the power losses of an electromechanical system. This result is obtained by using a set of filters like the ones proposed in [21] and acting on their velocity bounds according to a strategy that is characterized by a negligible computational overhead. The resulting algorithm must indeed be sufficiently efficient and compact to be implemented on low cost control boards.

The trajectory synchronization technique obviously requires the a priori knowledge of all transient times, which are generally unknown in the case of real-time planners. The efficiency of such planners depends indeed on the fact that they never evaluate the complete optimal trajectory but, conversely, they generate it by making the right choices at each sample time. As a consequence, transient lengths are not known a priori and this can be limiting in the case of applications which require knowing in advance when the tracking condition will be achieved. The transient time estimation algorithm consequently represents another outcome of this paper. Its computational burden is comparable with the one of the filter.

The paper novelties can thus be summarized as follows:

- Synthesis of an algorithm for the efficient evaluation of the transient times of acceleration-constrained, minimum-time trajectories;
- Synchronization of multiple trajectories by acting on the velocity constraints;
- Use of the synchronization technique to reduce friction power losses.

It is worth to point out that the real synchronization of the actuators can be achieved only if the synchronous planner proposed in this paper is followed by appropriate closed loop controllers. Indeed, it must be recalled that, given a cluster of actuators, different tracking errors must be expected for each of them because of model uncertainties and variable loads: the controllers' performance might affect negatively the planner action. The problem can be solved by adopting synchronization techniques, acting at the control level, which guarantee uniform tracking errors. An extensive discussion on synchronous controllers is beyond the scope of this paper, nevertheless a short overview on some recent advances is proposed in the following for the reader's convenience. The structures of synchronous control schemes for multi-axis systems largely vary depending on the application considered. Some of them have a fixed structure [37] and are model-independent. Others, for example the one proposed in [38], obtain better performances by exploiting the system dynamics to synthesize the control loop, while further ones – see for example [39] – use adaptive techniques for the synchronization of multi-robot systems. All these aspects have been simultaneously taken into consideration in [40], where the proposed controller is adaptive and robust, it accounts for mutual interferences between axes and, finally, shows optimal performances even during transients. Interested readers are addressed to the existing literature for further details.

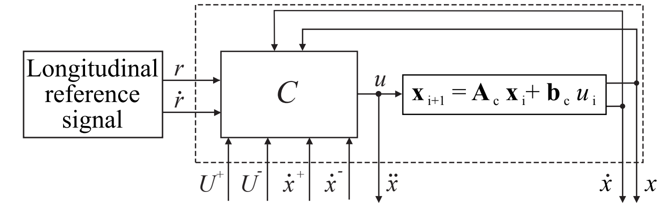
Longitudinal reference signal
r
$\dot{r}$
C
u
$\mathbf{x}_{i+1} = \mathbf{A}_c\,\mathbf{x}_i + \mathbf{b}_c\,u_i$
U^+
U^-
$\dot{x}^+$
$\dot{x}^-$
$\ddot{x}$
$\dot{x}$
x

Fig. 2. A schematic representation of the nonlinear filter. It converts a discontinuous reference $(r, \dot{r})$ into a continuous signal x whose first and second time derivatives are constrained between given bounds ($\dot{x}^- \leq \dot{x} \leq \dot{x}^+$ and $U^- \leq \ddot{x} \leq U^+$).

The paper is organized as follows. Section II briefly recalls the equations and the main characteristics of the nonlinear filter proposed in [21], then it analyzes and solves the transient-time estimation problem. Section III shows how multiple reference signals can be easily synchronized, by simultaneously reducing friction losses. The algorithm has been experimentally validated through laboratory tests whose results are discussed in Section IV. The work conclusions are drawn in Section V.

Notation

Operators $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ evaluate the ceiling and the floor of their arguments, respectively.

II. Evaluation of the transient time

The nonlinear filter shown in Fig. 2 represents the core of the synchronization strategy. It is composed by an algebraic, variable structure [23], nonlinear controller C which drives a discrete time system described by the following equation

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

where

$$\mathbf{A}_c = \left[\begin{array}{cc} 1 & T \\ 0 & 1 \end{array}\right]\ , \quad \mathbf{b}_c = \left[\begin{array}{c} \frac{T^2}{2} \\ T \end{array}\right]\ , \tag{2}$$

while $\mathbf{x}_i := [x_i \ \dot{x}_i]^T$ is the system state, u_i represents the system input and T the sampling time. It is easy to verify that (1) and (2) correspond to a chain of two discrete-time integrators. The filter behavior can be briefly summarized as follows: output x is a continuous signal that represents the best approximation of r compatibly with the following velocity and acceleration constraints

$$\dot{x}^- \leq \dot{x} \leq \dot{x}^+ \ , \tag{3}$$

$$U^- \leq \ddot{x} \leq U^+ \ . \tag{4}$$

Asymmetric bounds $\dot{x}^-$, $\dot{x}^+$, U^-, and U^+ can be freely assigned and can also be changed at run-time. Tracking transients of r are always minimum time compatibly with the constraints.

Let us indicate by $\mathbf{r}_i := [r_i \ \dot{r}_i]^T$ a reference signal given by steps and ramps like the ones shown in Fig. 1. By first applying the following transformation

$$\mathbf{y} = \mathbf{x} - \mathbf{r} \ , \tag{5}$$

and, then, the following one

$$\mathbf{z} = \left[\begin{array}{cc} \frac{1}{T^2} & \frac{1}{2T} \\ 0 & \frac{1}{T} \end{array} \right] \mathbf{y} \ , \tag{6}$$

the system equation becomes

$$\mathbf{z}_{i+1} = \mathbf{A}_d \, \mathbf{z}_i + \mathbf{b}_d \, u_i \ , \tag{7}$$

with

$$\mathbf{A}_d = \left[\begin{array}{cc} 1 & 1 \\ 0 & 1 \end{array} \right] \ , \quad \mathbf{b}_d = \left[\begin{array}{c} 1 \\ 1 \end{array} \right] \ . \tag{8}$$

Transformation (5) converts the original tracking problem into a stability problem: tracking condition $\mathbf{x} = \mathbf{r}$ is achieved when $\mathbf{y}$ converges toward the origin. Conversely, transformation (6) is used to eliminate sampling time T from the system equation. Evidently, because of (6), $\mathbf{z} = 0$ implies that $\mathbf{y} = 0$ and, in turn, that the tracking condition has been achieved. [21] shows that, by means of a proper controller C – whose equations are not reported here for conciseness – $\mathbf{z}$ can be forced toward the origin in minimum time by simultaneously satisfying constraints (3) and (4).

When the controller proposed in [21] is used, state $\mathbf{z}$ is attracted within a Boundary Layer (BL) of amplitude $U^+ - U^-$ created around a suitable sliding surface σ. Controller C has been designed so as to guarantee that, from any initial state $\mathbf{z}_0 := [z_{10} \ z_{20}]^T$, the BL is definitely reached in minimum time. Then, $\mathbf{z}$ is forced to remain within the BL and to slide toward the origin, still with a minimum-time transient: as stated earlier, this, in turn, implies that $\mathbf{x}$ reaches $\mathbf{r}$ in minimum time. Fig. 3 shows all possible typologies of trajectories in the $\mathbf{z}$-space. In the same figure, terms z_2^+ and z_2^- are defined as follows

$$z_2^+ := \frac{\dot{x}_i^+ - \dot{r}_i}{T} \ ,$$

$$z_2^- := \frac{\dot{x}_i^- - \dot{r}_i}{T} \ ,$$

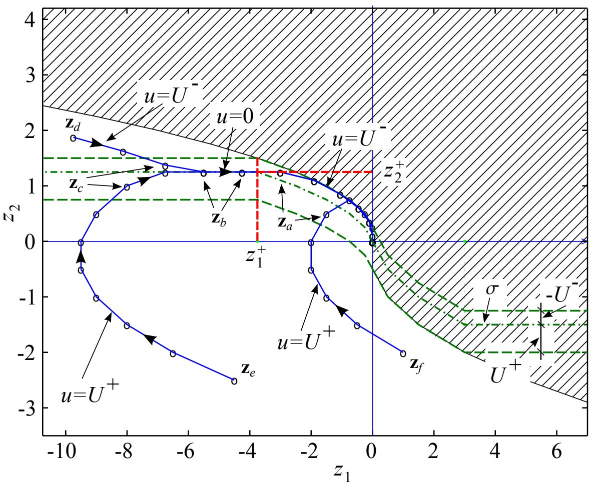

Fig. 3. Possible typologies of transients from generic states in the $\mathbf{z}$-space. Sliding surface σ is represented by dash-dotted lines, while the boundary layer is given by the area within the dashed lines.

and respectively represent the upper and lower velocity bounds converted into the $\mathbf{z}$-space, while z_1^+ and z_1^- are defined as follows

$$z_1^+ = -\left(\left\lfloor -\frac{z_2^+}{U^-} \right\rfloor + 1 \right) \left(z_2^+ + \left\lfloor -\frac{z_2^+}{U^-} \right\rfloor \frac{U^-}{2} \right) \ , \tag{9}$$

$$z_1^- = -\left(\left\lfloor -\frac{z_2^-}{U^+} \right\rfloor + 1 \right) \left(z_2^- + \left\lfloor -\frac{z_2^-}{U^+} \right\rfloor \frac{U^+}{2} \right) \ . \tag{10}$$

A short analysis of the filter synthesized in [21] is instrumental for the following discussion. As anticipated, controller C is designed so as to guarantee that convergence is always achieved in minimum time. In order to establish the number of sample times required for the convergence, all the transients which may occur must be scrutinized. It will be shown in the following that transient times can be deterministically and efficiently determined on the basis of the initial state of the filter. Without any loss of generality, the discussion will neglect initial states lying in the grayed area of Fig. 3 but, for symmetry reasons, the conclusions which are drawn in the following apply to them too.

If the initial state coincides with $\mathbf{z}_a := [z_{1a} \ z_{2a}]^T$ (see Fig. 3), i.e., if the state is within the BL and $z_1^+ \leq z_{1a} \leq 0$, the system reaches the origin in minimum time after m steps where

$$m := \left\lfloor \frac{1}{2} + \sqrt{\frac{1}{4} + 2 \left| \frac{z_{1a}}{U^-} \right|} \right\rfloor \ . \tag{11}$$

Equation (11) descends directly from the controller equations (see [21] and its references).

If the initial state is $\mathbf{z}_b := [z_{1b} \ z_2^+]^T$, i.e., the system velocity is equal to the admissible maximum, controller C imposes $u = 0$. By recalling that discrete time systems evolve according to the following equation

$$\mathbf{z}_k = \mathbf{A}_d^k \, \mathbf{z}_0 + \sum_{i=0}^{k} \mathbf{A}_d^{k-1} \mathbf{b}_d \, u_i \ , \tag{12}$$

where $\mathbf{A}_d$ and $\mathbf{b}_d$ are defined according to (8), with a few algebraic manipulations it can be asserted that, after k steps ($k \in \mathbb{N}$), the state will assume the following value

$$\mathbf{z}_k = \begin{bmatrix} z_{10} + k\, z_2^+ \\ z_2^+ \end{bmatrix} , \tag{13}$$

where $z_{10} = z_{1b}$. Practically, this implies that the state slides along z_2^+ until it crosses abscissa z_1^+ after k_b samples, where

$$k_b = \left\lceil \frac{z_1^+ - z_{1b}}{z_2^+} \right\rceil \tag{14}$$

can be easily derived from (13). Consequently, from $\mathbf{z}_b$ the origin is reached in $k_b + m$ steps.

From any state $\mathbf{z}_c := [z_{1c}\ z_{2c}]^T$ located within the BL, i.e., such that $z_{1c} < z_1^+$ and $z_2^+ - U^+ \leq z_{2c} \leq z_2^+ - U^-$, the upper velocity limit is always reached with a single step, thus the convergence toward the origin requires $k_b + m + 1$ sample times.

If the initial state coincides with $\mathbf{z}_d := [z_{1d}\ z_{2d}]$, where $z_{1d} < z_1^+$ and $z_{2d} > z_2^+ - U^-$, controller C pushes the state toward the feasible region in minimum time by imposing $u = U^-$. Still considering (12), a few manipulations permit verifying that the system evolves according to the following equation ($z_{10} = z_{1d}$, $z_{20} = z_{2d}$, $k \in \mathbb{N}$)

$$\mathbf{z}_k = \begin{bmatrix} z_{10} + k\, z_{20} + \dfrac{k(k+1)}{2}\, U^- \\ k\, U^- + z_{20} \end{bmatrix} \tag{15}$$

until the BL is reached, i.e., as long as condition $z_2 \leq z_2^+ - U^-$ holds. Equation (15) makes it possible to assert that this happens after $k = k_d$ sample times, where

$$k_d = \left\lceil \frac{z_2^+ - z_{2d} - U^-}{U^-} \right\rceil . \tag{16}$$

Similarly, from $\mathbf{z}_e := [z_{1e}\ z_{2e}]^T$ the BL is reached in minimum time with $u = U^+$. Equation (12) becomes

$$\mathbf{z}_k = \begin{bmatrix} z_{10} + k\, z_{20} + \dfrac{k(k+1)}{2}\, U^+ \\ k\, U^+ + z_{20} \end{bmatrix} , \tag{17}$$

where $z_{10} = z_{1e}$, $z_{20} = z_{2e}$, and $k \in \mathbb{N}$. In this case, the BL is reached when $z_2 \geq z_2^+ - U^+$, i.e., for $k = k_e$, where

$$k_e = \left\lceil \frac{z_2^+ - z_{2e} - U^+}{U^+} \right\rceil . \tag{18}$$

The last possible case concerns an initial state given by $\mathbf{z}_f := [z_{1f}\ z_{2f}]^T$. Differently for the last case, the state enters the BL through a point characterized by $z_1 \geq z_1^+$. The controller initially imposes $u = U^+$, so that the system evolves according to (17), with $z_{10} = z_{1f}$ and $z_{20} = z_{2f}$. After k_f steps the state enters the BL, whose lower bound is a composite curve made of segments whose vertexes have equation ($\tilde{k} \in \mathbb{N}\backslash 0$)

$$\mathbf{z}_{\tilde{k}} = \begin{bmatrix} \dfrac{\tilde{k}(\tilde{k}-1)}{2} U^- \\ -(\tilde{k}-1)U^- - U^+ \end{bmatrix} . \tag{19}$$

The evaluation of k_f is slightly more complex, since it requires establishing the values of k and $\tilde{k}$ for which the two piecewise linear curves (17) and (19) intersect each other. In order to simplify the evaluation of k_f, let us temporarily assume that k and $\tilde{k}$ are real numbers, i.e., replace both piecewise curves with a continuous-tangent approximation. It is then possible to solve equation $\mathbf{z}_k = \mathbf{z}_{\tilde{k}}$, whose terms are derived from (17) and (19), for $k > 0$ and $\tilde{k} > 0$. After a few algebraic manipulations and by discarding the unfeasible solutions, it can be verified that the two curves intersect for ($z_{10} = z_{1f}$, $z_{20} = z_{2f}$)

$$k = -1 - \frac{z_{20}}{U^+} + \frac{\sqrt{(2\,U^+ z_{10} - U^+ z_{20} - z_{20}^2)(U^+ - U^-)U^-}}{(U^+ - U^-)U^+} . \tag{20}$$

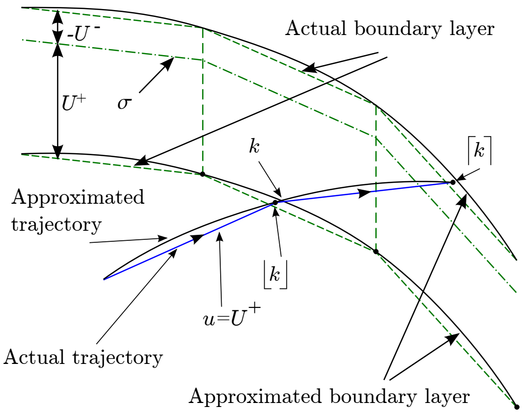

Fig. 4. Problems that could arise due to the approximation of the trajectory and of the BL with continuous curves: the point corresponding to $\lceil k \rceil$ is outside the actual BL. In this case, the correct value of k_f is $\lfloor k \rfloor$, which corresponds to the first point of the trajectory lying inside the BL.

This implies that after k_f steps, where

$$k_f = \lceil k \rceil , \tag{21}$$

the state enters the BL. The use of a continuous approximation for the evaluation of k_f, i.e., the approximation of the two piecewise linear curves with continuous-tangent curves, could lead to the problem shown in Fig. 4: the point corresponding to k_f is actually outside the BL. For this reason, after k_f has been evaluated, it is always necessary to check if the corresponding state is actually inside the BL and, if it is not, it is necessary to pose

$$k_f = \lfloor k \rfloor . \tag{22}$$

Once the state is inside the BL, the origin is finally reached in m steps, where m is given by (11). Summarizing, depending on the initial state, the number of sample times required to achieve convergence can be evaluated according to Table I. The same table also shows the number of elementary operations (additions, multiplications, etc.) that must be executed depending on the initial state. Reported data are directly extrapolated from the code used in the experiments of Section IV and consider worst-case situations, so that actual values are actually much lower. In conclusion, the procedure proposed here deterministically allows one to estimate, on the basis of the current state $\mathbf{z}$, the number of sampling cycles required to reach the tracking condition. Such information is acquired

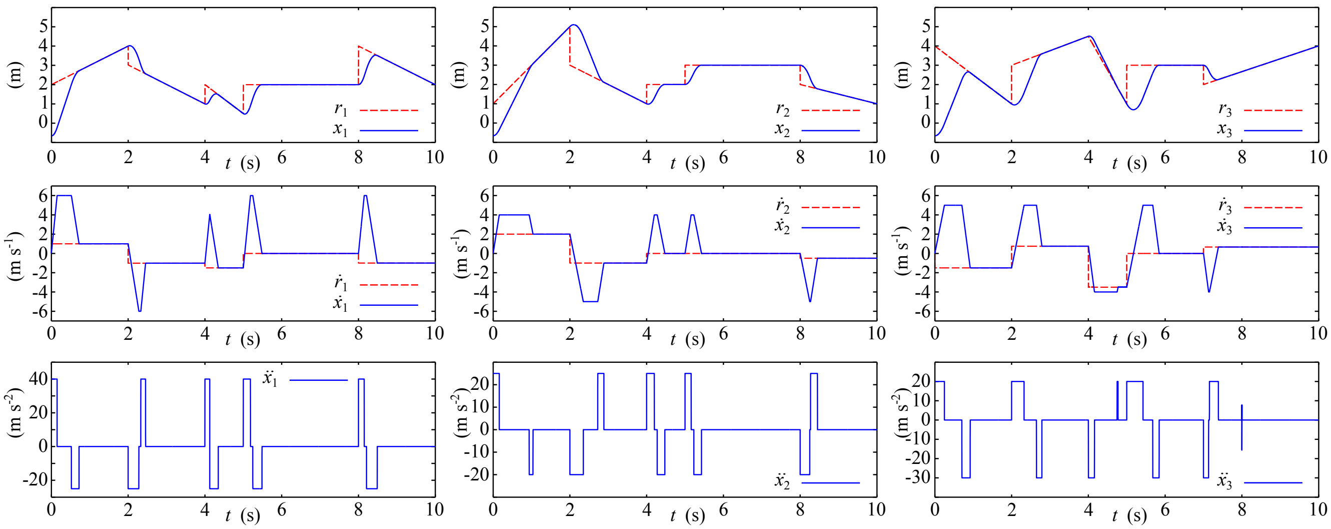

Fig. 5. Transients for the three independent reference signals.

TABLE I

NUMBER OF STEPS REQUIRED TO REACH THE ORIGIN AND MAXIMUM NUMBER OF ALGEBRAIC OPERATIONS. M STANDS FOR MULTIPLICATIONS OR DIVISIONS, A INDICATES ADDITIONS OR SUBTRACTIONS, WHILE S MEANS SQUARE ROOTS. TERMS m, k_b, k_d, k_e, AND k_f ARE EVALUATED ACCORDING TO (11), (14), (16), (18), AND (21) OR (22), RESPECTIVELY.

initial state	samples	algebraic operations
$\mathbf{z}_a$	m	28M + 25A + 3S
$\mathbf{z}_b$	$k_b + m$	31M + 29A + 3S
$\mathbf{z}_c$	$1 + k_b + m$	42M + 40A + 4S
$\mathbf{z}_d$	$k_d + 1 + k_b + m$	48M + 46A + 4S
$\mathbf{z}_e$	$k_e + 1 + k_b + m$	243M + 178A + 4S
$\mathbf{z}_f$	$k_f + m$	223M + 158A + 3S

TABLE II

KINEMATIC CONSTRAINTS USED FOR THE GENERATION OF THE THREE INDEPENDENT REFERENCE SIGNALS.

	U^-	U^+	$\dot x^-$	$\dot x^+$
signal 1	-25	40	-6	6
signal 2	-20	25	-5	4
signal 3	-30	20	-4	5

TABLE III

TRANSIENT TIMES FOR THE THREE REFERENCE SIGNALS. * HIGHLIGHTS THE LONGEST TRANSIENTS, WHILE t INDICATES THE STARTING TIME OF EACH TRANSIENT.

t (s)	0.0	2.0	4.0	5.0	7.0	8.0
signal 1	0.722	0.456	0.349	0.488	–	0.490*
signal 2	1.040*	0.887*	0.475	0.430	–	0.452
signal 3	0.922	0.790	0.775*	0.845*	0.389	–

through an algorithm whose computational burden is limited and known.

III. SYNCHRONIZATION OF MULTIPLE REFERENCE SIGNALS

The algorithm proposed in Section II can be used any time the transient time of each single filter needs to be known in advance. The trajectory synchronization technique proposed in the following represents one possible application of the above mentioned algorithm. Let us consider several independent discontinuous references, like the ones shown by the dashed lines in Fig. 5, which must be smoothly joined and synchronized by fulfilling, at the same time, the asymmetric bounds reported in Table II. The given constraints can be easily fulfilled by using one filter for each reference but, as shown in Fig. 5, such approach would produce different length transient times. This conclusion is also confirmed by Table III, which reports all transient times.

If all signals refer to a single process, the fastest transients can be slowed down so as to guarantee that the tracking condition is achieved simultaneously for all filters. This choice has no impact on the process productivity and the degrees of freedom acquired can be spent to optimize a given performance index. For example, in this paper they will be used to improve the energetic efficiency of an electromechanical system, but alternative choices are also possible. The proposed approach is based on a simplified strategy in order to limit the computational load. Potentially better performances could be obtained through a deeper analysis of the problem, however the proposed technique, despite its simplicity, covers very common applicative cases for which a consistent power saving can be achieved at a negligible computational burden.

Let us consider one of the filters and an initial configuration like the one shown in Fig. 6, for which $\dot r >= 0$ and $r_0 > x_0$, where r_0 and x_0 indicate the initial values of the reference signal and of the filter output, respectively. In the same figure, t_m indicates the minimum transient time of the current filter achieved by assuming a velocity upper bound equal to $\dot x^+$, i.e., equal to the user defined value, while t_M indicates the longest transient time detected among all the filters, estimated through the procedure proposed in Section II. By assuming a new upper bound $\hat v < \dot x^+$, a transient time $t_f \in [t_m, t_M]$ can be obtained: which is the best choice for $\hat v$ in terms of power efficiency? In order to answer to this question the actual velocity profile is approximated in the following by a straight line (see the

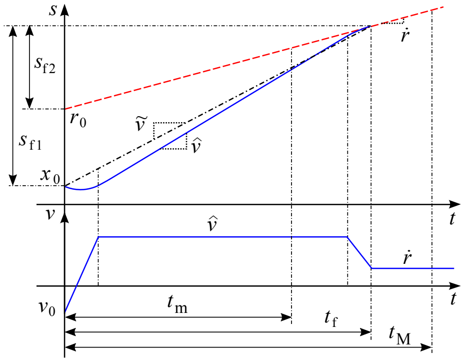

Fig. 6. Typical transient of filter output signal (solid line) toward the reference signal (dashed line) generated by the power saving strategy when $\dot{r} \geq 0$ and $x_0 < r_0$. The dash-dotted line represents an approximation of the actual transient.

dash-dotted line in Fig. 6) whose slope is $\widetilde{v}$: this practically implies the elimination of the parabolic blends and the use of infinite accelerations. This approximation has minimal impact on the power analysis. The choice of $\widetilde{v}$ can be used to modify the transient time, which ends when $r_0 + \dot{r}t_f = x_0 + \widetilde{v}t_f$, i.e., for

$$t_f = \frac{r_0 - x_0}{\widetilde{v} - \dot{r}} \,. \tag{23}$$

Thus, by acting on $\widetilde{v}$, $t_f \in [t_m, t_M]$ can be selected. The choice of $\widetilde{v}$ has evidently an impact on the power losses of a hypothetical mechanical system. For example, by assuming that they are mainly caused by viscous friction, which is actually true in many cases, it is possible to propose a performance index which depends on the squared value of the velocity. Practically, energy losses in the interval $[0, t_M]$ can be assumed proportional to the following index

$$E = \int_0^{t_M} \dot{x}^2(\tau) d\tau \,. \tag{24}$$

Bearing in mind (23), (24) can be posed into the following form

$$\begin{aligned} E &= \int_0^{t_f} \widetilde{v}^2 d\tau + \int_{t_f}^{t_M} \dot{r}^2 d\tau = (\widetilde{v}^2 - \dot{r}^2)\, t_f + \dot{r}^2 t_M \\ &= (r_0 - x_0)(\widetilde{v} + \dot{r}) + \dot{r}^2 t_M \,, \end{aligned}$$

i.e., since $r_0 - x_0 > 0$, energy losses linearly increase with $\widetilde{v}$.

From this analysis it is possible to evince that power losses can be reduced by minimizing $\widetilde{v}$ or, due to (23), by maximizing t_f, i.e., by imposing $t_f = t_M$.

Let us now come back to the actual transient trajectory. According to Fig. 6, it can be divided into three parts: the initial and the final ones admit velocities which change linearly, while in the second one the velocity is constant, so that it is possible to write

$$v_1(t) = v_0 + U^+ t, \quad t \in [0, t_1], \tag{25}$$

$$v_2(t) = \hat{v}, \quad t \in [0, t_2],$$

$$v_3(t) = \hat{v} + U^- t, \quad t \in [0, t_3], \tag{26}$$

where t_1, t_2, and t_3 represent the three partial transient times. For $t = t_1$, the velocity must be $v_1(t_1) = \hat{v}$, so that from (25) it can be evinced that

$$t_1 = \frac{\hat{v} - v_0}{U^+} \,. \tag{27}$$

Similarly, for t_3, with analogous manipulations, it is possible to obtain

$$t_3 = \frac{\dot{r} - \hat{v}}{U^-} \,, \tag{28}$$

and, finally, for t_2 we have

$$t_2 = t_f - t_1 - t_3 = t_f - \frac{\hat{v} - v_0}{U^+} - \frac{\dot{r} - \hat{v}}{U^-} \,. \tag{29}$$

The distance covered for $t_f = t_1 + t_2 + t_3$ can be obtained by integrating the three velocity functions. Thus, bearing in mind (27)–(29), it is possible to write

$$s_1 = \int_0^{t_1} v_1(\tau) d\tau = \frac{(v_0 - \hat{v})^2}{2U^+} - \frac{(v_0 - \hat{v})v_0}{U^+},$$

$$s_2 = \int_0^{t_2} v_2(\tau) d\tau = \left(t_f + \frac{v_0 - \hat{v}}{U^+} + \frac{\hat{v} - \dot{r}}{U^-}\right)\hat{v},$$

$$s_3 = \int_0^{t_3} v_3(\tau) d\tau = \frac{(\hat{v} - \dot{r})^2}{2U^-} - \frac{(\hat{v} - \dot{r})\hat{v}}{U^-}.$$

Consequently the total distance covered at time t_f is

$$\begin{aligned} s_{f_1} = &- \frac{(U^- - U^+)\hat{v}^2}{2U^-U^+} + \frac{(U^-U^+t_f + U^-v_0 - U^+\dot{r})\hat{v}}{U^-U^+} \\ &- \frac{U^-v_0^2 - U^+\dot{r}^2}{2U^-U^+} \,. \end{aligned} \tag{30}$$

During the same period, the reference signal covers a distance

$$s_{f_2} = r_0 + \dot{r}t_f \,. \tag{31}$$

Evidently, the two signals intersect in t_f if

$$s_{f_1} = r_0 - x_0 + s_{f_2} \,. \tag{32}$$

By substituting (30) and (31) in (32), by solving the resulting quadratic equation for $\hat{v}$, and by discarding the negative solution ($\hat{v}$ must be positive), it is possible to obtain the value of $\hat{v}$ which guarantees a transient time equal to t_f

$$\hat{v} = \frac{U^-U^+t_f + U^-v_0 - U^+\dot{r} + \widetilde{\Gamma}}{\Delta U}, \tag{33}$$

where

$$\begin{aligned} \widetilde{\Gamma} &:= \sqrt{U^-U^+[t_fU^-(t_fU^+ + 2\Delta v) + 2\,\Delta x\,\Delta U + \Delta v^2]}, \\ \Delta U &:= U^- - U^+, \\ \Delta v &:= v_0 - \dot{r}, \\ \Delta x &:= x_0 - r_0. \end{aligned}$$

The transient length which minimizes power losses is finally obtained by setting $t_f = t_M$.

An analogous reasoning shows that for $\dot{r} <= 0$ and $r_0 < x_0$ the optimal $\hat{v}$ is given by

$$\hat{v} = -\frac{U^-U^+t_f + U^+v_0 - U^-\dot{r} + \hat{\Gamma}}{\Delta U}, \tag{34}$$

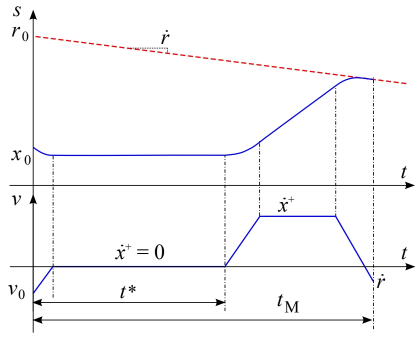

Fig. 7. A typical transient of the filter output signal (solid line) toward the reference signal (dashed line) generated by the power saving strategy when $\dot{r} < 0$ and $r_0 > x_0$.

where

$$\hat{\Gamma} := \sqrt{U^-U^+[t_fU^+(t_fU^- + 2\Delta v) - 2\,\Delta x\,\Delta U + \Delta v^2]}.$$

In both cases, the evaluation of the new bound requires executing 10 additions or subtractions, 15 multiplications or divisions, and 1 square root.

The solution adopted when $\dot{r} < 0$ and $r_0 > x_0$ is shown in Fig. 7. The power saving effect is simply obtained by setting $\dot{x}^+ = 0$, since this choice forces the filter to reduce $|\dot{x}|$. Sometimes – like in the example of Fig. 7 – the motor is even stopped. Evidently, the computational impact of this strategy is zero. Condition $\dot{x}^+ = 0$ is maintained as long as the transient time of the considered filter, which is continuously evaluated by the algorithm proposed in Section II, coincides with the one of the slowest transient: at time $t = t^*$ the original velocity bound is resumed, so that r is reached without overshoot at $t = t_M$. The same strategy is adopted when $\dot{r} > 0$ and $r_0 < x_0$ by imposing $\dot{x}^- = 0$.

Fig. 8 shows the transients obtained with the synchronization strategy proposed. As expected, the velocity upper and lower bounds become variable in order to guarantee that the three transients always end simultaneously. It is interesting to notice that at time $t = 7$ s the first and the second signals are not influenced by the transient of the third one and that, in the same way, transients of the first two references which occur at $t = 8$ s do not influence the third one. This behavior is due to a design choice: once the filter state is inside its BL, i.e., if the filter is tracking its reference signal, the bound scaling algorithm is disabled. This behavior, which can be penalizing in terms of power efficiency, has been introduced to avoid useless transients and overshoots, but can easily be removed so as to give priority to the energy saving strategy.

TABLE IV
ENERGY LOSSES QUANTIFIED, FOR EACH ONE OF THE SIX TRANSIENTS, THROUGH PERFORMANCE INDEX E (M^2 S^{-1}). t INDICATES THE STARTING TIME OF EACH TRANSIENT.

t (s)	0.0	2.0	4.0	5.0	7.0	8.0
Async	47.78	31.88	16.10	20.41	2.13	11.36
Sync	42.12	28.04	12.25	14.85	2.13	10.88

The energy losses of the six transients, quantified through index (24), are reported in Table IV. According to the expectations, viscous friction losses decrease due to the signal synchronization, as confirmed by the performance index E which drops from 129.66 m^2 s^{-1} to 110.27 m^2 s^{-1}, with a power saving equal to 14.95%. It must be noted that, according to the premises, the fifth transient ($t = 7.0$ s) is not modified, so that the corresponding power losses do not change.

It is worth highlighting that the given power gain only refers to losses that are caused by viscous friction during transients. Lower figures must be clearly expected when considering all possible losses and by also including the tracking periods in which, evidently, the two strategies cause the same power consumptions.

Table V reports a worst-case analysis concerning the overall computational complexity of the energy-saving strategy. The number of numerical operations linearly depends on N, i.e., on the number of reference signals that must be synchronized.

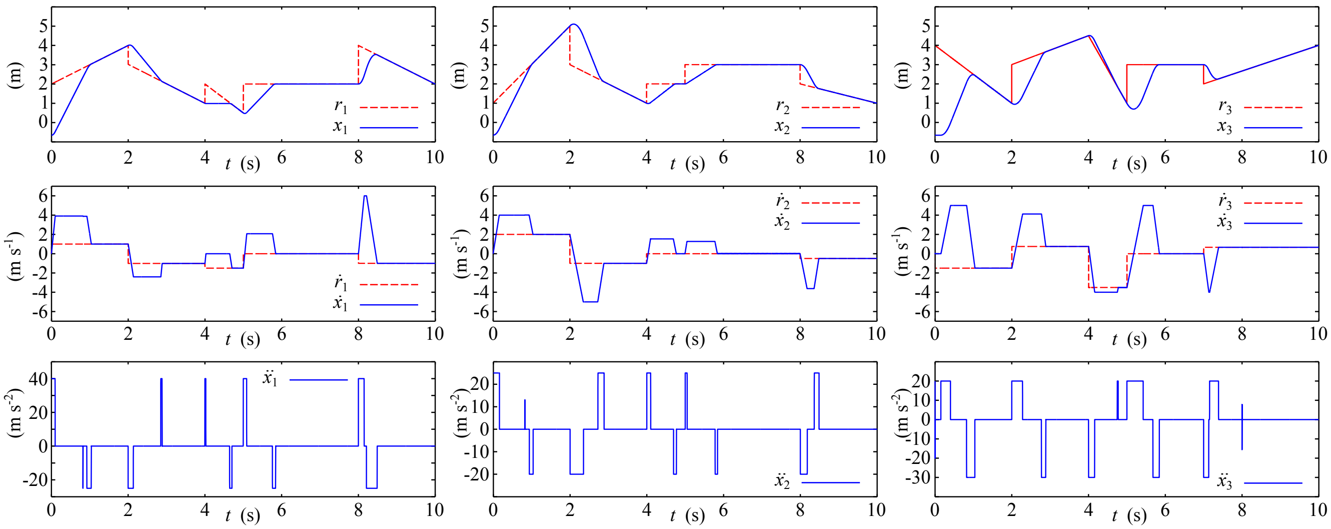

Fig. 8. Transients obtained through the synchronization of the three smooth references. Notice that, due to the synchronization strategy, velocity upper bounds continuously change.

TABLE V
WORST-CASE ANALYSIS OF THE COMPUTATIONAL COMPLEXITY OF THE ENERGY-SAVING ALGORITHM. N INDICATES THE NUMBER OF REFERENCE SIGNALS SIMULTANEOUSLY PROCESSED.

transient lengths estimation	N (243M + 178A + 4S)
bounds scaling	(N-1) (15M + 10A + 1S)
filters' update	N (25M + 18A + 1S)

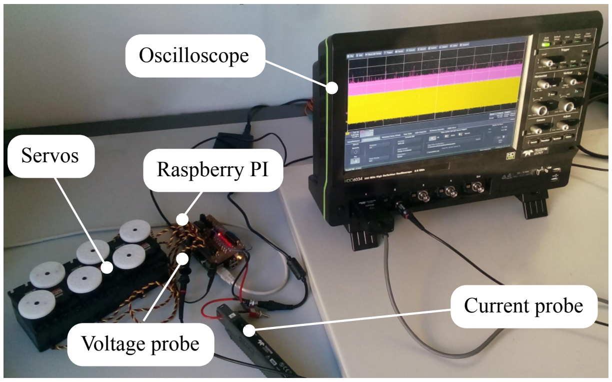

Fig. 9. The experimental test bench

Evidently, the most time-consuming part of the strategy is represented by the algorithm proposed in Section II for the transient length estimation. However, it is worth to point out that the worst-case analysis of Table V is based on the worst-case analysis of Table I: the actual computational burden is strongly affected by the evolution of the filters' state, so that much smaller figures can normally be expected – the total load is, roughly, just three times larger than the one required for the filters' update – and, when the tracking condition is eventually achieved, hardware resources are only dedicated to the filters' update.

IV. EXPERIMENTAL RESULTS

As stated in the introduction, the technique proposed has been conceived for systems with limited computational capabilities. For this reason, the tests reported in the following were carried out by means of an old generation Raspberry Pi board model B equipped with a single core ARM processor running at 700MHz .

The experimental test bench is shown in Fig. 9. It is composed of the Raspberry Pi board which, through a specifically developed interface "shield", drives an array of 6 servo-motors. Each motor is equipped with an internal position controller whose sample rate is equal to $20 \cdot 10^{-3}$ s. The same sample rate was adopted for the Raspberry Pi board.

The synchronization strategy aims at reducing the viscous friction losses. The experimental setup does not permit a direct evaluation of such losses, so efficiency improvements were estimated indirectly by measuring the electric power consumption of the six motors. This is clearly a limitation since electric consumptions are also affected by other losses. Just to mention the most important ones: servo-controller losses, power-bridge losses, Coulomb-friction losses, and copper losses. In any case, electric consumptions can give an idea of the efficiency improvements that can be achieved. They were measured by means of a LaCie HDC6034 oscilloscope equipped with current and voltage probes. Such instrument can acquire data at a sample rate of $40 \cdot 10^{-6}$ s, for a maximum period of 2000 s (roughly half an hour).

The discontinuous reference signals were directly generated by the Raspberry Pi board through a planner of random steps and ramps. Six independent sequences – one for each motor – were generated. In the same way, six sets of bounds $\dot{x}^-$, $\dot{x}^+$, U^-, and U^+ were randomly selected, one for each filter. Two different tests were carried out: the synchronization algorithm was disabled in the first one and enabled in the second one. In both experiments the random reference planner was initialized with the same seed so as to produce the same sequence of signals and bounds. Fig. 10 shows the initial 10 seconds of both experiments. It is possible to notice that references were changed at least every half second, and that also ramp slopes were randomly selected. In the first experiment the computational burden was only due to the evaluation of the 6 filters: an average computing time equal to $17.49 \cdot 10^{-6}$ s was measured. By activating the synchronization strategy, the average computing time raised to $50.42 \cdot 10^{-6}$ s: despite the increment, it is still compatible with the sample rate of the experiment ($20 \cdot 10^{-3}$ s).

The measured energy consumption, evaluated on a time horizon of 2000 s, was equal to 39548 J for the asynchronous strategy, while it decreased to 38713 J for the synchronous one, i.e., energy losses reduce by 2.1%. As expected this figure is smaller than the one given in Section III, but this is normal since the experimental measure includes all possible power losses and it also covers the tracking periods, i.e., those in which the two strategies are energetically indistinguishable.

A further test was then carried out to verify if the synchronization method can also be used for a larger number of reference signals. In particular, 100 independent and random set-points were simultaneously managed. Evidently, it was not possible to drive so many motors, consequently the experimental results concern the algorithmic efficiency only: still using the Raspberry Pi board, the average computational time of the entire algorithm is equal to $548.93 \cdot 10^{-6}$ s, i.e., it is still compatible with the sample rate used.

A conclusive remark concerns the maximum evaluation times. They have not been reported since they change at each run of the algorithm: the Raspberry PI board uses a Debian Wheezy operating system, which is inherently multitasking. Consequently, although a high priority was assigned to the planner, the maximum evaluation time changed casually and it was not correlated to the number of managed signals. Anyway, in the worst case, the maximum detected computing time was lower than $2 \cdot 10^{-3}$ s, i.e., it was smaller than the process sample time.

V. CONCLUSIONS

This paper has proposed several novel contributions. First of all, a transient-time estimation technique has been designed for trajectories subject to velocity and acceleration constraints. The considered trajectories start from generic interpolating

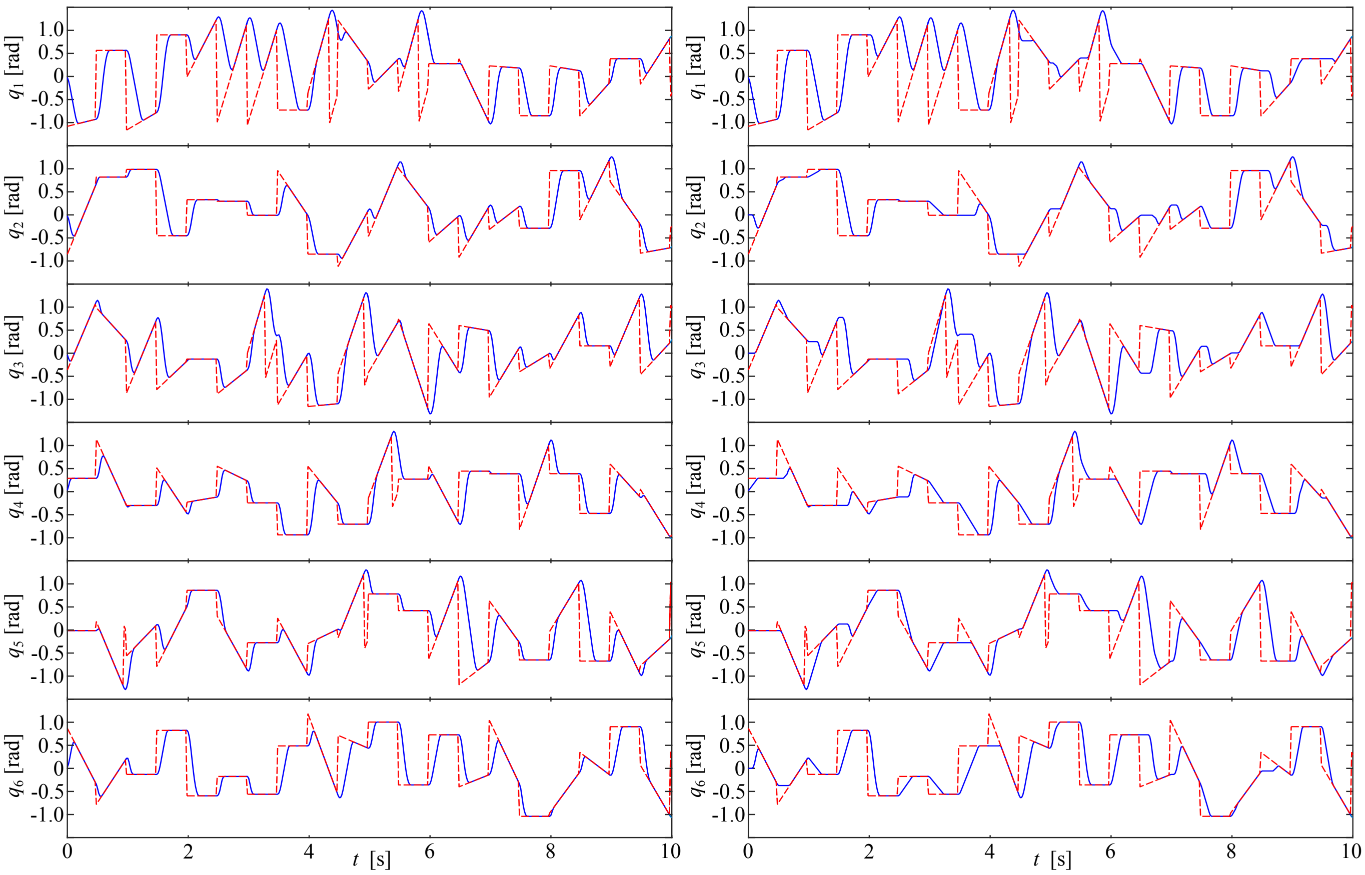

Fig. 10. The first 10 seconds of the two experiments. The discontinuous reference signals of the six motors are represented through dashed lines, solid lines are used for the filters output. The left figure shows the results corresponding to the experiment in which the filters were not synchronized, while the right one shows what happens due to the filters synchronization.

conditions and converge toward constant or ramp reference signals. The efficiency of the proposed technique derives from the exclusive use of closed-form expressions.

Then, the novel algorithm has been used to synchronize, through the generation of smooth trajectories, a set of independent reference signals. In particular, the synchronization is achieved by slowing down the fastest transients and by using the degrees of freedom thus acquired to improve the energetic efficiency of the system. The energy-saving strategy still uses closed-form expressions and has a worst-case computational burden that can be deterministically evaluated: as shown in the paper, in the worst case it linearly depends on the number of managed reference signals, but experimental results have shown that the actual dependence is less than linear. Another characteristic of the proposed technique is represented by the code compactness: the entire algorithm can be implemented with less than 400 lines of C language – the dimension of the executable file is equal to 70 kB – so that it can run on low performance control boards. In this paper, the reference synchronization strategy has been exploited to improve the power efficiency of a mechatronic system, but alternative performance indexes can be considered. Still remaining in an energy saving context, for example, a more accurate model could be introduced for power losses and more incisive actions could be taken to further improve the plant efficiency.

References

[1] S. Macfarlane and E. A. Croft, “Jerk-bounded manipulator trajectory planning: design for real-time applications,” *IEEE Trans. Robot. Autom.*, vol. 19, DOI 10.1109/TRA.2002.807548, no. 1, pp. 42–52, 2003.

[2] R. Haschke, E. Weitnauer, and H. Ritter, “On-line planning of timeoptimal, jerk-limited trajectories,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, DOI 10.1109/IROS.2008.4650924, 2008, pp. 3248–3253.

[3] T. Kröger and F. M. Wahl, “On-line trajectory generation: basic concepts for instantaneous reactions to unforeseen events,” *IEEE Trans. Robot.*, vol. 26, DOI 10.1109/TRO.2009.2035744, no. 1, pp. 94–111, Feb. 2010.

[4] X. Broquère, D. Sidobre, and I. Herrera-Aguilar, “Soft motion trajectory planner for service manipulator robot,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, DOI 10.1109/IROS.2008.4650608, 2008, pp. 2808–2813.

[5] B. Ezair, T. Tassa, and Z. Shiller, “Planning high order trajectories with general initial and final conditions and asymmetric bounds,” *Int. J. Robot. Res.*, vol. 33, DOI 10.1177/0278364913517148, no. 6, pp. 898–916, May. 2014.

[6] A. Frisoli, C. Loconsole, R. Bartalucci, and M. Bergamasco, “A new bounded jerk on-line trajectory planning for mimicking human movements in robot-aided neurorehabilitation,” *Robot. Auton. Syst.*, vol. 61, DOI 10.1016/j.robot.2012.09.003, no. 4, pp. 404–415, Apr. 2013.

[7] L. Messner, H. Gattringer, and H. Bremer, “Efficient Online Computation of Smooth Trajectories Along Geometric Paths for Robotic Manipulators ,” in *Multibody system dynamics, robotics, and control*, H. Gattringer and J. Gerstmayr, Eds. Wien: Springer-Verlag, 2013.

[8] M. Hehn and R. D’Andrea, “Real-time trajectory generation for quadrocopters,” *IEEE Trans. Robot.*, vol. 31, DOI 10.1109/TRO.2015.2432611, no. 4, pp. 877–892, Aug. 2015.

[9] K.-H. Rew and K.-S. Kim, “A closed-form solution to asymmetric motion profile allowing acceleration manipulation ,” *IEEE Trans. Ind.*

Electron., vol. 57, DOI 10.1109/TIE.2009.2036032, no. 7, pp. 2499–2506, Jul. 2010.

[10] F. Zou, D. Qu, J. Wang, and F. Xu, "Asymmetric trajectory planning for vacuum robot motion," in *Proc. IEEE Int. Conf. Robot. Autom.*, DOI 10.1109/ICRA.2011.5980588, May. 2011, pp. 1–4.

[11] C.-W. Ha, K.-H. Rew, and K.-S. Kim, "Robust zero placement for motion control of lightly damped systems," *IEEE Trans. Ind. Electron.*, vol. 60, DOI 10.1109/TIE.2012.2206334, no. 9, pp. 3857–3864, 2013.

[12] L. Biagiotti and C. Melchiorri, "FIR filters for online trajectory planning with time- and frequency-domain specifications," *Contr. Eng. Pract.*, vol. 20, DOI 10.1016/j.conengprac.2012.08.005, no. 12, pp. 1385–1399, 2012.

[13] L. Biagiotti, C. Melchiorri, and L. Moriello, "Optimal trajectories for vibration reduction based on exponential filters," *IEEE Trans. Contr. Syst. Technol.*, vol. 24, DOI 10.1109/TCST.2015.2460693, no. 2, pp. 609–622, 2016.

[14] L. Rutkowski, A. Przybyl, and K. Cpalka, "Novel online speed profile generation for industrial machine tool based on flexible neuro-fuzzy approximation," *IEEE Trans. Ind. Electron.*, vol. 59, DOI 10.1109/TIE.2011.2161652, no. 2, pp. 1238–1247, Feb. 2012.

[15] C. Guarino Lo Bianco, A. Tonielli, and R. Zanasi, "Nonlinear trajectory generator for motion control systems," in *Proc. IEEE Int. Conf. Ind. Electron. Soc.*, DOI 10.1109/IECON.1996.570952, Taiwan, Taipei, Aug. 1996, pp. 195–201.

[16] Y. Su and P. Mueller, "Smooth reference trajectory generation for industrial mechatronic systems under torque saturation," in *Proc. IFAC Symp. Mechatronic Syst.*, vol. 4 - PART 1, DOI 10.3182/20060912-3-DE-2911.00154, Heidelberg, Germany, Sept. 2006, pp. 896–901.

[17] X. Wei, J. Wang, and Z. Yang, "Robust smooth-trajectory control of nonlinear servo systems based on neural networks," *IEEE Trans. Ind. Electron.*, vol. 54, DOI 10.1109/TIE.2006.888784, no. 1, pp. 208–217, 2007.

[18] L. Biagiotti and R. Zanasi, "Online trajectory planner with constraints on velocity, acceleration and torque," in *Proc. IEEE Int. Symp. Ind. Electron.*, DOI 10.1109/ISIE.2010.5637558, Bari, Italy, Jul. 2010, pp. 274–279.

[19] R. Zanasi, C. Guarino Lo Bianco, and A. Tonielli, "Nonlinear filters for the generation of smooth trajectories," *Automatica*, vol. 36, DOI 10.1016/S0005-1098(99)00164-8, no. 3, pp. 439–448, Mar. 2000.

[20] C. Guarino Lo Bianco and R. Zanasi, "Smooth profile generation for a tile printing machine," *IEEE Trans. Ind. Electron.*, vol. 50, DOI 10.1109/TIE.2003.812284, no. 3, pp. 471–477, 2003.

[21] C. Guarino Lo Bianco and F. Wahl, "A novel second order filter for the real-time trajectory scaling," in *Proc. IEEE Int. Conf. Robot. Autom.*, DOI 10.1109/ICRA.2011.5979764, Shanghai, China, May. 2011, pp. 5813–5818.

[22] C. Guarino Lo Bianco and F. Ghilardelli, "A discrete-time filter for the generation of signals with asymmetric and variable bounds on velocity, acceleration, and jerk," *IEEE Trans. Ind. Electron.*, vol. 61, DOI 10.1109/TIE.2013.2284135, no. 8, pp. 4115–4125, Aug. 2014.

[23] V. I. Utkin, "Variable structure systems with sliding modes," *IEEE Trans. Autom. Contr.*, vol. 22, DOI 10.1109/TAC.1977.1101446, pp. 212–222, 1977.

[24] W. Kim, D. Shin, and C. C. Chung, "Microstepping using a disturbance observer and a variable structure controller for permanent-magnet stepper motors," *IEEE Trans. Ind. Electron.*, vol. 60, DOI 10.1109/TIE.2012.2198033, no. 7, pp. 2689–2699, 2013.

[25] M. V. Basin and P. C. Rodriguez-Ramirez, "Sliding mode controller design for stochastic polynomial systems with unmeasured states," *IEEE Trans. Ind. Electron.*, vol. 61, DOI 10.1109/TIE.2013.2240641, no. 1, pp. 387–396, 2014.

[26] B. Bona, M. Indri, and N. Smaldone, "Rapid prototyping of a model-based control with friction compensation for a direct-drive robot," *IEEE/ASME Trans. Mechatronics*, vol. 11, DOI 10.1109/TMECH.2006.882989, no. 5, pp. 576–584, Oct. 2006.

[27] M. Montanari, S. Peresada, C. Rossi, and A. Tilli, "Speed sensorless control of induction motors based on a reduced-order adaptive observer," *IEEE Trans. Contr. Sys. Technol.*, vol. 15, DOI 10.1109/TCST.2007.899714, no. 6, pp. 1049–1064, 2007.

[28] E. Mininno, F. Neri, F. Cupertino, and D. Naso, "Compact differential evolution," *IEEE Trans. Evol. Comput.*, vol. 15, DOI 10.1109/TEVC.2010.2058120, no. 1, pp. 32–54, 2011.

[29] D. Naso, F. Cupertino, and B. Turchiano, "NPID and adaptive approximation control of motion systems with friction," *IEEE Trans. Contr. Syst. Technol.*, vol. 20, DOI 10.1109/TCST.2011.2106128, no. 1, pp. 214–222, Jan. 2012.

[30] F. Cupertino, P. Giangrande, G. Pellegrino, and L. Salvatore, "End effects in linear tubular motors and compensated position sensorless control based on pulsating voltage injection," *IEEE Trans. Ind. Electron.*, vol. 58, DOI 10.1109/TIE.2010.2046577, no. 2, pp. 494–502, Feb. 2011.

[31] M. Corradini, V. Fossi, A. Giantomassi, G. Ippoliti, S. Longhi, and G. Orlando, "Minimal resource allocating networks for discrete time sliding mode control of robotic manipulators," *IEEE Trans. Ind. Inf.*, vol. 8, DOI 10.1109/TII.2012.2205395, no. 4, pp. 733–745, Nov. 2012.

[32] H. Kim and B. K. Kim, "Online minimum-energy trajectory planning and control on a straight-line path for three-wheeled omnidirectional mobile robots," *IEEE Trans. Ind. Electron.*, vol. 61, DOI 10.1109/TIE.2013.2293706, no. 9, pp. 4771–4779, Sep. 2014.

[33] Y. Wang, Y. Zhao, S. Bortoff, and K. Ueda, "A real-time energy-optimal trajectory generation method for a servomotor system," *IEEE Trans. Ind. Electron.*, vol. 62, DOI 10.1109/TIE.2014.2360077, no. 2, pp. 1175–1188, Feb. 2015.

[34] H.-K. Shin and B. K. Kim, "Energy-efficient gait planning and control for biped robots utilizing vertical body motion and allowable ZMP region," *IEEE Trans. Ind. Electron.*, vol. 62, DOI 10.1109/TIE.2014.2360152, no. 4, pp. 2277–2286, Apr. 2015.

[35] Z. Lu, K. Sekiyama, T. Aoyama, Y. Hasegawa, T. Kobayashi, and T. Fukuda, "Energetically efficient ladder descent motion with internal stress and body motion optimized for a multilocomotion robot," *IEEE Trans. Ind. Electron.*, vol. 62, DOI 10.1109/TIE.2015.2396872, no. 8, pp. 4972–4984, Aug. 2015.

[36] N. Uchiyama, Y. Ogawa, A. El Khalick M, and S. Sano, "Energy saving in five-axis machine tools using synchronous and contouring control and verification by machining experiment," *IEEE Trans. Ind. Electron.*, vol. 62, DOI 10.1109/TIE.2015.2437354, no. 9, pp. 5608–5618, Sep. 2015.

[37] D. Sun, X. Shao, and G. Feng, "A model-free cross-coupled control for position synchronization of multi-axis motions: theory and experiments," *IEEE Trans. Contr. Syst. Technol.*, vol. 15, DOI 10.1109/TCST.2006.883201, no. 2, pp. 306–314, Mar. 2007.

[38] Y. Xiao and K. Y. Zhu, "Optimal synchronization control of high-precision motion systems," *IEEE Trans. Ind. Electron.*, vol. 53, DOI 10.1109/TIE.2006.878317, no. 4, pp. 1160–1169, Jun. 2006.

[39] D. Sun and J. K. Mills, "Adaptive synchronized control for coordination of multirobot assembly tasks," *IEEE Trans. Robot. Autom.*, vol. 18, DOI 10.1109/TRA.2002.802229, no. 4, pp. 498–510, Aug. 2002.

[40] M. H. Cheng, Y. J. Li, and E. G. Bakhoum, "Controller synthesis of tracking and synchronization for multiaxis motion system," *IEEE Trans. Contr. Syst. Technol.*, vol. 22, DOI 10.1109/TCST.2013.2251344, no. 1, pp. 378–386, Jan. 2014.

Corrado Guarino Lo Bianco (M'04) graduated with honors in Electronic Engineering and received the Ph.D. degree in Control System Engineering from the University of Bologna, Italy, in 1989 and 1994, respectively.

Currently, he is with the Dipartimento di Ingegneria dell'Informazione of the University of Parma as Associate Professor on Industrial Robotics.

Corrado Guarino Lo Bianco is involved in researches concerning mobile and industrial robotics. In particular, his research interests comprise the smooth and optimal trajectory generation, the robot kinematics and dynamics, and the robot control.