PERGAMON

Automatica 36 (2000) 439–448

www.elsevier.com/locate/automatica

Brief Paper

Nonlinear filters for the generation of smooth trajectories☆

R. Zanasi[a,*], C. Guarino Lo Bianco[b], A. Tonielli[c]

[a]DIS, Dipartimento di Scienze dell'Ingeneria, University of Modena and Reggio Emilia, Via Araldi 270, 41100 Modena, Italy
[b]Dipartimento di Ingegneria dell'Informazione, University of Parma, Parco Area delle Scienze 181A, 43100 Parma, Italy
[c]DEIS, Department of Electronics, Computer and System Science, University of Bologna, V.le Risorgimento 2, 40136 Bologna, Italy

Received 25 February 1998; received in final form 21 June 1999

Abstract

A nonlinear variable structure (VS) system is presented, suitable for smooth trajectory generation in motion control system. Joined to a rough reference trajectory generator (step, ramp, etc.) it provides smooth output signals according to user-selectable bounds on trajectory derivatives. The bounds on output derivatives can be changed during system operation without impairing system stability. Moreover, minimum time response with guaranteed no overshoot is ensured both in the continuous and discrete-time implementations. The main contribution of the paper lies in the discrete-time version of the trajectory generator. The performance of the proposed nonlinear filters is tested through simulation experiments. Simulation results concerning the use of the discrete filter in a position tracking control system are also presented.

Keywords: Trajectories; Smoothing filters; Optimal trajectory; Variable structure systems; Discrete digital dynamic control

1. Introduction

Motion Control systems are complex feedback–feedforward control systems where the availability of smooth reference trajectories, with bounded time derivatives, is a key point to get high dynamics and accurate tracking performance. Bounds on derivatives of reference trajectories are primarily a consequence of energetic and constructive limitations in available actuators as, for example, voltage and current limits in electric drives, leading to velocity and acceleration bounds on controlled output variables. In many applications, variable bounds (task-dependent) have to be imposed on time derivatives of each trajectory to get the specified performance, see for example Leonhard (1985), and Rossi and Tonielli (1992,1994).

In robotics, the problem of smooth generation of position trajectories is usually solved by computing optimal profiles, see Brady (1982). Off-line optimisation routines are used and some a priori information on the desired trajectory is required. Dynamic saturations induced by bounded velocities or accelerations are, as a consequence, handled by constraining the optimisation problem. If the desired trajectory changes, the tracking profile are also to be changed and optimisation routines run again.

A different approach is proposed in this paper. Although mainly motivated by the aim of designing a system which on-line generates task-dependent trajectories in position tracking control problems, the proposed system can be easily adopted in tracking problems of other state variables. Smooth trajectories are generated at the output of a state-variable filter controlled by a nonlinear feedback dynamic controller, designed so that the filter output tracks "at best" an external reference without exceeding (on-line) user-selectable bounds on output derivatives. In position control problems "tracking at best" means, for example, that a position step reference is transformed into a smooth signal which tracks the step in the smallest time achievable with velocities and accelerations compatible with the task and without overshoot. The basic idea was first proposed in Guarino Lo Bianco, Tonielli and Zanasi (1996), where the basic features were described. An improved version of the trajectory smoother for the continuous-time case was presented in Zanasi, Guarino Lo Bianco and Tonielli (1998). In this

☆This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor A. Bagchi under the direction of Editor T. Basar.

* Corresponding author. Tel.: + 39-59-360-266; fax: + 39-59-270-477.

E-mail addresses: roberto.zanasi@unimo.it (R. Zanasi), guarino@ce.unipr.it (C. Guarino Lo Bianco), atonielli@deis.unibo.it (A. Tonielli)

0005-1098/00/$ - see front matter
PII: S0005-1098(99)00164-8

paper, after a quick review of the previous results, the optimal design of the trajectory smoother for the discrete-time design is given. Comparable performances are obtained by the continuous and discrete-time designs.

The design of the trajectory smoother is mainly based on variable structure (VS) control techniques, see Utkin (1977). With a proper choice of a sliding surface and a boundary layer, it is possible to guarantee good performances both in transient and in final steady-state conditions. The second-order filter is considered in this paper, corresponding to dynamic limitations on first and second derivatives. Extensions to higher-order cases are under study. The kernel of the trajectory smoother is built around a nonlinear state-variable filter; this ensures that bounded first and second output derivatives are available to generate feedforward control actions.

The paper is organised as follows. In Section 2, the Problem is stated. In Section 3 the optimal structure of the trajectory smoother for the continuous-time case is given: the dynamic behaviour and the stability are studied in the case of time-varying reference signals. The continuous-time case described in this section must be considered as a motivation and inspiration for the discrete-time controllers presented in the following Section 4, where the design of the trajectory smoother for the discrete-time case is considered: the optimal structure is given and its main characteristics are tested in simulation. In Section 5, simulations concerning a practical application example are reported: the trajectory smoother is used in the control system of the angular position of a DC motor driving a mechanical tool along an assembly line.

2. Problem statement

Let us consider the feedback system shown in Fig. 1 composed by the cascade of n integrators, where $r(t)$ is the input, $x(t)$ is the output and $x^{(i)}$ (for $i = 1, \ldots, n$) denotes the ith time derivative of the output. The scope of the paper is to design the nonlinear controller such that the output $x(t)$ tracks the reference signal $r(t)$ while limiting the time derivatives $x^{(i)}$ to given (variable) values: $|x^{(i)}| < x_M^{(i)}$.

The problem has a solution only if the time derivatives of the reference signal are "almost always" bounded as, for example, when tracking steps, square waves, sawtooth, etc. This implies some consequences:

1. perfect tracking is achievable only when the derivatives of the reference signal are smaller than the maximum admissible output derivatives: $|r^{(i)}| < x_M^{(i)}$;
2. if the derivatives of the reference signal are larger than the maximum admissible output derivatives, perfect tracking is not achievable. According to the problem position this condition cannot last for an infinite time. As soon as reference derivatives return within the admissible limits, the tracking can be achieved again.

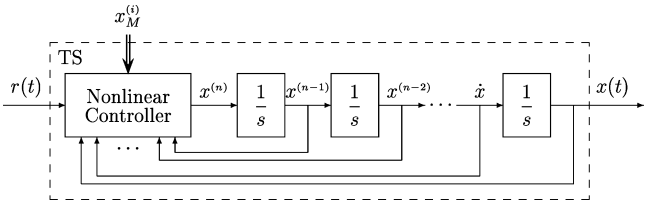


Fig. 1. Nonlinear trajectory smoother (TS).

By adding a linear state-feedback controller to the cascade of integrators a linear state-variables filter is obtained. If the bandwidth of the closed-loop system is sufficiently larger than that of the reference signal, the system output tracks the reference and its derivatives. State-variable filters are often used to reconstruct derivatives of noisy signals. The saturation of the state variables of a linear state-variables filter does not solve the stated problem, owing to the linear characteristic of the state-feedback controller. A nonlinear controller must be designed. A variable structure controller is adopted to solve the following control problem.

Control Problem (CP). *Design a nonlinear state-variables filter whose output $x(t)$ tracks "at best" the reference signal $r(t)$ and its derivatives with the following specifications*:

(S1) *when a smooth reference is applied* (*derivatives smaller than the bound values*), *the tracking condition $x(t) = r(t)$ is ensured $\forall t > t_0$*;

(S2) *when a discontinuous reference trajectory or a non-smooth reference* (*derivatives larger than the bound values*) *is applied*, *the tracking is lost. As soon as the smooth reference is re-established*, *the tracking must be achieved again in "minimum time" and without overshoot*;

(S3) *output derivatives must be bounded*; *the bounds are time variable and they can also change during tracking transients*:

$$|x^{(i)}| < x_M^{(i)} \quad (i = 1, \ldots, n) \tag{1}$$

(S4) *bounded output derivatives are available as additional outputs*.

The second-order filter design is considered in this paper for both the continuous-time and the discrete-time design of position tracking system. Higher-order filter design is under investigation. The following notations are used. The filter variables are denoted as x, $\dot{x}$, $\ddot{x}$, where $\dot{x} = x^{(1)}$ and $\ddot{x} = x^{(2)}$. For simplicity and without loss of generality, bounds on the output derivatives are assumed to be symmetrical, namely $|\dot{x}| < \dot{x}_M$ for the speed bound and $|\ddot{x}| < \ddot{x}_M = U$ for the acceleration bound.

In the following development the first derivative of the reference signal is assumed to be known. Although this condition ensures some properties when tracking high-order reference signals (ramp, parabolas, etc.) it is not strictly required to track steps or square waves signals.

3. Continuous-time case

The continuous-time development of the controller is original, although it follows from well-known theoretical results from optimal control, see Lewis (1986), and sliding mode control, see Utkin (1977). The continuous-time case is reported here because its development was the motivation and the inspiration for the design of the discrete version which represents the main contribution of the paper. The structure of the trajectory smoother (TS) for the continuous-time second-order case, is shown in Fig. 2.

A first choice for controller C is the following.

Controller 1. *The control signal $u(t)$ is chosen as follows*:

$$C1: \quad u = -U \operatorname{sgn}\sigma \frac{1+\operatorname{sgn}(\dot{x}\operatorname{sgn}\sigma+\dot{x}_M)}{2}, \tag{2}$$

$$\sigma = y + \frac{|\dot{y}|\dot{y}}{2U}, \tag{3}$$

where $y = x - r$ is the tracking error, $\dot{y} = \dot{x} - \dot{r}$ is the velocity error, $\dot{r}$ is the continuous-time derivative of the reference signal r, $\operatorname{sgn}(\cdot) = \pm 1$ is the sign function and $\sigma = 0$ is the adopted sliding mode surface.

Result 1. *Controller C1 solves the Control Problem.*

Proof. The system trajectories in the phase plane $(y, \dot{y})$ are shown in Fig. 3. The sliding mode surface $\sigma = 0$ is composed of two parabolas, p_1 and p_2. Lines $\dot{y}_M = \dot{x}_M - \dot{r}$ and $\dot{y}_m = -\dot{x}_M - \dot{r}$ correspond to upper and lower bounds for velocity $\dot{y}$. Note that, since $\dot{y} = \dot{x} - \dot{r}$, the following correspondence holds:

$$|\dot{x}| \le \dot{x}_M \leftrightarrow \dot{y}_m \le \dot{y} \le \dot{y}_M. \tag{4}$$

Parabolas p_1, p_2 and lines $\dot{y}_M$, $\dot{y}_m$ divide the phase plane $(y, \dot{y})$ into four regions: R_1, R_2, R_3 and R_4 (see Fig. 3). Only two different types of trajectories are present in the system: straight lines in regions R_1 and R_2 and parabolas in regions R_3 and R_4.

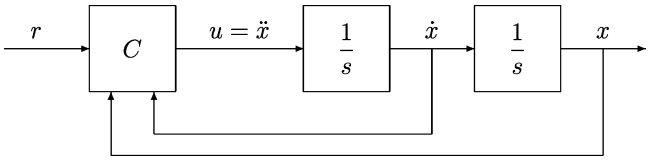


Fig. 2. Nonlinear trajectory smoother for the continuous-time case: r reference signal, x output position, $\dot{x}$ velocity, $\ddot{x}$ acceleration.

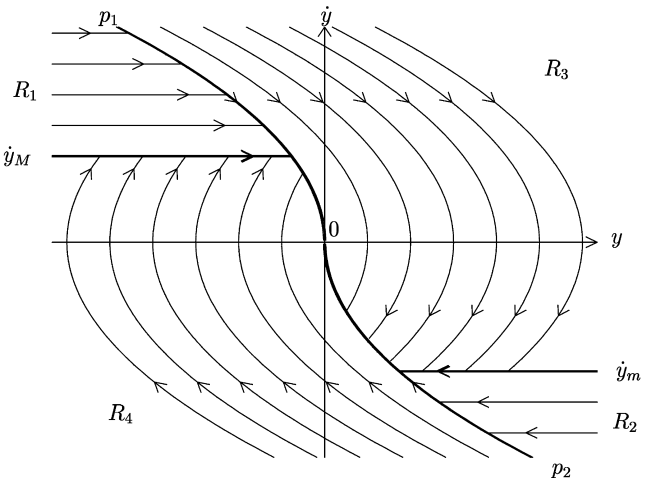


Fig. 3. System trajectories when controller $C1$ is used: y position error; $\dot{y}$ velocity error; $\dot{y}_M$ and $\dot{y}_m$ are the upper and lower bounds for the velocity error.

Let $\mathbf{y} = (y, \dot{y})$ denote the state of the system in the phase plane. Since $\dot{x} = \dot{y} + \dot{r}$, from (4) and (2) it follows that the term

$$\frac{1+\operatorname{sgn}(\dot{x}\operatorname{sgn}\sigma+\dot{x}_M)}{2} = \begin{cases} 0 & \text{if } \mathbf{y}\in R_1,\ \mathbf{y}\in R_2, \\ 1 & \text{if } \mathbf{y}\in R_3,\ \mathbf{y}\in R_4 \end{cases} \tag{5}$$

in expression (2), forces $u = 0$ when state $\mathbf{y}$ belongs to regions R_1 and R_2. The velocity bound $|\dot{x}| \le \dot{x}_M$ is surely satisfied during normal system operation, while it is satisfied after a finite time if the initial condition $\dot{\mathbf{y}}(0)$ belongs to regions R_1 or R_2. The control signal $u(t) = \ddot{x}$ can assume only three different constant values: (1) $u(t) = 0$ when $\mathbf{y}\in R_1$ and $\mathbf{y}\in R_2$; (2) $u(t) = -U$ when $\mathbf{y}\in R_3$; (3) $u(t) = U$ when $\mathbf{y}\in R_4$. In regions R_3 and R_4 the system trajectories are parabolas

$$y(t) - \frac{\dot{y}^2(t)}{2u_0} = y_0 - \frac{\dot{y}_0^2}{2u_0}, \tag{6}$$

where $u_0 = \pm U$ is the input and $(y_0, \dot{y}_0)$ is the initial condition. From Fig. 3 it is evident that the controlled system is globally stable: all system trajectories reach the sliding surface $\sigma = 0$ in a finite time. Then the state $\mathbf{y}$ moves along this surface until it reaches the desired final position $y(t) = 0, \rightarrow x(t) = r(t)$. This condition is reached in "minimum-time", without overshoot, with bounded velocity and bounded acceleration. Moreover, it is clear that the parameters $\dot{x}_M$ and U can be freely changed during system operation without influencing the system stability. Result 1 is proved. □

A drawback of controller $C1$ is the presence of an "ideal sliding mode": when the state $\mathbf{y}$ reaches the sliding

mode surface $\sigma = 0$, the control signal $u(t)$ starts switching at infinite frequency between the two values $\pm U$, see Eq. (2). At each instant, the "mean value' of this switching at infinite frequency is equal to the equivalent control $u_{eq}(t)$, i.e. the control needed to keep the state on the sliding mode surface. This "ideal sliding mode" cannot be generated by a real physical system. In fact, all actual implementations of the control law (2) always lead to chattering in the output, because of their "finite" switching frequency. The chattering can be reduced by introducing a boundary layer around the sliding surface. This can be done by substituting in (2) the sign function $\mathrm{sgn}(\cdot)$ with a saturation function $\mathrm{sat}(\cdot)$. Nevertheless, this solution does not guarantee for the absence of overshoot and, moreover, a long settling time is expected in the neighbourhood of the final position, owing to the very small damping coefficient of the control law. To cope with these problems, the following controller is proposed.

Controller 2. *The control signal $u(t)$ is chosen as follows*:

$$C2: \quad u = -U\,\mathrm{sat}(\lambda\sigma)\frac{1+\mathrm{sgn}(\dot{x}\,\mathrm{sgn}\,\sigma+\dot{x}_M)}{2}, \tag{7}$$

$$\sigma = y + \frac{\max\{|\dot{y}|,k\}\dot{y}}{2U} + \frac{3}{2\lambda}\mathrm{sat}\left(\frac{\dot{y}}{k}\right), \tag{8}$$

where $k=\sqrt{U/\lambda}$, $y=x-r$ is the tracking error, $\dot{y}=\dot{x}-\dot{r}$ is the velocity error and $\dot{r}$ is the continuous-time derivative of the reference signal r.

The two main differences between controllers $C1$ and $C2$ are:

(1) in (7) the sign function $\mathrm{sgn}(\sigma)$ has been replaced by the saturation function $\mathrm{sat}(\lambda\sigma)=\{\lambda\sigma$ if $|\lambda\sigma|\le 1;\mathrm{sgn}(\lambda\sigma)$ if $|\lambda\sigma|>1\}$, that is, a boundary layer has been introduced in the vicinity of sliding surface $\sigma=0$;
(2) the sliding mode variable σ defined in (8) has been properly modified in order to guarantee the absence of overshoot.

Result 2 (Main result for the continuous-time case). *Controller $C2$ solves the Control Problem in the minimum time compatible with the presence of the boundary layer. In the neighbourhood of the tracking condition $x(t)=r(t)$, the dynamics of the system is exponentially stable with real and coincident eigenvalues: $s_{1,2}=-\sqrt{U\lambda}$. Moreover, it behaves as a type 2 system, i.e.:*

(1) *If the reference signal is a ramp ($r=r_0t$), the steady-state tracking error is zero.*
(2) *If the reference signal is a parabola ($r=at^2$), the steady-state tracking error is constant.*

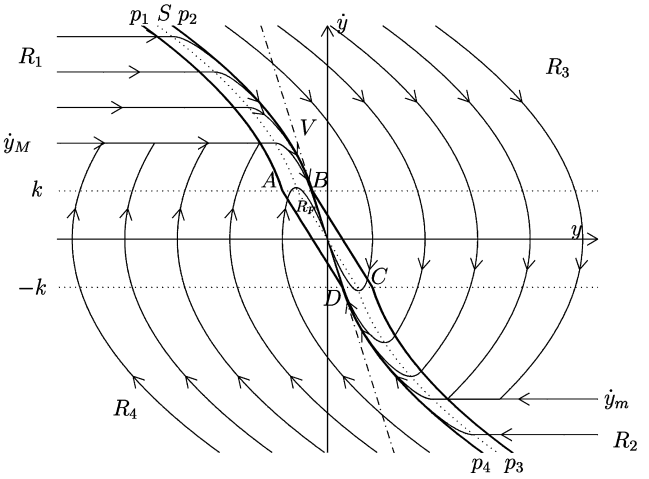


Fig. 4. System trajectories when controller $C2$ is used.

Proof. The system dynamics is analysed in the following two cases: (1) behaviour in large; (2) behaviour in the vicinity of the final position.

(1) *Behaviour "in large"*. The system trajectories are shown in Fig. 4. The sliding mode surface $\sigma=0$, i.e. the dotted curve S in the centre of the boundary layer, is composed by two parabolas and one segment. This particular shape has been chosen to avoid overshoot during transients. The system state $\mathbf{y}$ belongs to the boundary layer if $|\lambda\sigma|<1$. The amplitude of the boundary layer is $\beta=2/\lambda$, completely determined by λ. When $\lambda\to\infty$, the amplitude $\beta\to 0$, $k\to 0$, the boundary layer tends to disappear and controller $C2$ tends to $C1$. The boundary layer is delimited by the parabolas p_1, p_2, p_3, p_4 and the two segments A–D and B–C (see Fig. 4). In contrast to other boundaries, parabolas p_2 and p_4 are "real system trajectories" and therefore they cannot be crossed by any other trajectory.

Owing to this particular choice of the boundary layer, it is evident that the system is globally stable "in large". In fact, when the system state $\mathbf{y}$ is outside the boundary layer, the control signal $u(t)$ assumes values ($u=\pm U$) which surely force the state $\mathbf{y}$ towards the linear region. Consequently, no matter the initial condition, all the trajectories reach the boundary layer in a finite time and, then, they are forced to move towards the region delimited by the four points A, B, C, D. The boundedness of the velocity during transients is guaranteed by the presence of term (5) in expression (7).

(2) *Behaviour in the vicinity of the final position*. Let $R_F=\{(y,\dot{y}):|\dot{y}|\le k,|\lambda\sigma|\le 1\}$ denote the region of the phase plane delimited by the four points A, B, C and D. When the state $\mathbf{y}\in R_F$, the control law (7) and (8) is linear and the system dynamics is described by the following differential equation:

$$\ddot{y}+2\sqrt{U\lambda}\dot{y}+U\lambda y=-\ddot{r}, \tag{9}$$

where $y = x - r$, $\dot{y} = \dot{x} - \dot{r}$ and $\ddot{y} = \ddot{x} - \ddot{r}$ are the tracking error and its first- and second-order time derivatives, respectively. The eigenvalues of Eq. (9) are real and coincident: $s_{1,2} = -\sqrt{U\lambda}$. The settling time of the system decreases when λ increases. The eigenvector $V: \dot{y} = -\sqrt{U}y$ corresponding to eigenvalues $s_{1,2}$ is shown in Fig. 4 as the dash–dotted line V. It crosses the boundary of region R_F in correspondence of the two points B and D where it is also "tangent" to the two parabolas p_2 and p_4. These two parabolas and the eigenvector V are real system trajectories and therefore they cannot be crossed by other trajectories. No overshoot is allowed in the system, and the final position r is reached in the minimum time compatible with the presence of the boundary layer. Moreover, from (9) it follows that:

(1) if the reference signal is a step $r = c_0$ or a ramp $r = r_0 t$, the acceleration and the steady-state tracking error are zero: $\ddot{r} = 0$ and $\lim_{t\to\infty} y(t) = 0$;
(2) if the reference signal is a parabola $r(t) = at^2$, the acceleration and the steady-state tracking error are constant: $\ddot{r} = a$, $\lim_{t\to\infty} y(t) = -a/(\lambda U)$. Since $|a| < U$, the tracking error is always smaller than the amplitude of the boundary layer $|\lim_{t\to\infty} y(t)| < 1/\lambda$. □

Simulation results obtained with controller $C2$ are shown in Fig. 5. The saw tooth reference signal $r(t) = 2.5 + 3[t - \mathrm{Int}(t)]$ is considered, where $\mathrm{Int}(\cdot)$ is the integer-part function. The simulation parameters are: $U = 50$, $\lambda = 200$, $\dot{x}_M = 10$ and zero initial conditions. To test performance with time-variable bounds, at time $t = 2$ the maximum allowed acceleration has been changed to $U = 100$. The reference $r(t)$, the position $x(t)$, the velocity $\dot{x}(t)$ and the acceleration $u(t)$ are reported in Fig. 5a–c. The trajectory $(x, \dot{x})$ in the phase plane is reported in Fig. 5d. The trajectory of the tracking error in the phase plane $(y, \dot{y})$ is reported in Fig. 5e. A detail in the vicinity of the origin is shown in Fig. 5f.

From these figures it is evident that perfect tracking is obtained without overshoot and without exceeding the velocity and acceleration constraints. The maximum acceleration is maintained during almost all the transients assuring the fastest possible dynamic response compatible with the imposed constraints and with the presence of the boundary layer. The global behaviour is quite satisfactory.

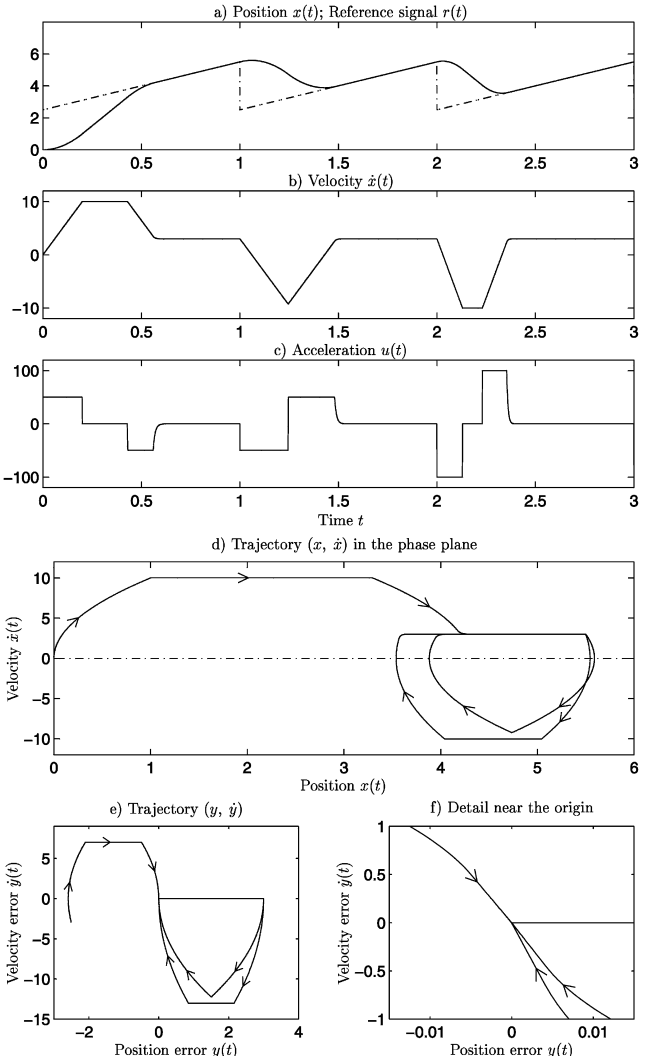


Fig. 5. Simulation results obtained when controller $C2$ is used: (a) position $x(t)$ and reference signal $r(t)$; (b) velocity $\dot{x}(t)$; (c) acceleration $u(t)$; (d) trajectory $(x, \dot{x})$ in the phase plane; (e) trajectory $(y, \dot{y})$ of the tracking error; (f) detail in the vicinity of the origin.

4. Discrete-time case

The trajectory smoother presented in the previous section guarantees the absence of overshoot only if controller C is implemented as a "continuous-time" system. Direct discretisation of controller $C2$ does not ensure that the system trajectories remain within the boundary layer during the transient, and overshoot may arise in the system. For example, if the previous simulation of Fig. 5 is repeated computing controller $C2$ only at the sampling instants $t = nT$ with $T = 0.005$ s, the results shown in Fig. 6 are obtained. The tracking error trajectory $(y,\dot{y})$ is reported in Fig. 6a; a detail in the vicinity of the origin is shown in Fig. 6b. These figures clearly show the presence of a large undesired overshoot on the output trajectory generated by the controller "discretisation". To overcome this problem the trajectory smoother must be "re-designed" as "discrete-time system". The proposed structure for the discrete-time trajectory smoother is shown in Fig. 7. This structure is a "modified" discrete-time version of the continuous-time scheme of Fig. 2. Sampled variables are denoted by the same name as that

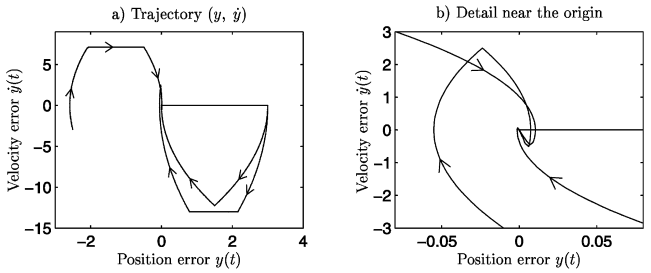


Fig. 6. Trajectories when a discretised version of controller $C2$ is used: (a) trajectory $(y, \dot{y})$ of the tracking error; (b) detail in the vicinity of the origin.

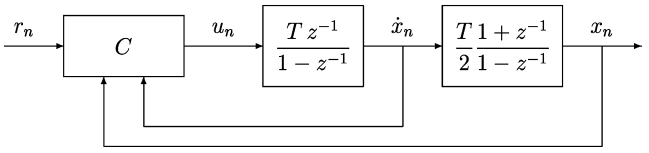


Fig. 7. Trajectory smoother in the discrete-time case.

of the correspondent analogue ones, i.e. $r(t) \to r_n$, $x(t) \to x_n$, etc.

Remark. The two discrete-time integrators of Fig. 7 have "different" structures. This choice has been made in order to guarantee the discrete-time smoother the same dynamic behaviour as that of the continuous-time one when the acceleration u_n is constant. For the discrete-time case, the following controller is proposed.

Controller 3. *The control signal u_n is chosen as follows*:

$$C3: \quad u_n = -U\,\mathrm{sat}(\sigma_n)\frac{1+\mathrm{sgn}(\dot{x}_n\,\mathrm{sgn}\,\sigma_n + \dot{x}_M - TU)}{2}, \tag{10}$$

$$\sigma_n = \dot{z}_n + \frac{z_n}{m} + \frac{(m-1)}{2}\,\mathrm{sgn}(z_n), \tag{11}$$

$$m = \mathrm{Int}\left[\frac{1+\sqrt{1+8|z_n|}}{2}\right], \tag{12}$$

$$z_n = \frac{1}{TU}\left(\frac{y_n}{T} + \frac{\dot{y}_n}{2}\right), \qquad \dot{z}_n = \frac{\dot{y}_n}{TU}, \tag{13}$$

where $y_n = x_n - r_n$ is the tracking error, $\dot{y}_n = \dot{x}_n - \dot{r}_n$ is the velocity error, $\dot{r}_n$ is the discrete-time derivative of signal r_n computed as follows:

$$\dot{r}_n = \frac{2}{T}(r_n - r_{n-1}) - \dot{r}_{n-1} \tag{14}$$

and T is the sampling period of the controller. Signal r_n is supposed to be known, and its discrete-time derivative $\dot{r}_n$ is supposed to be piece-wise constant.

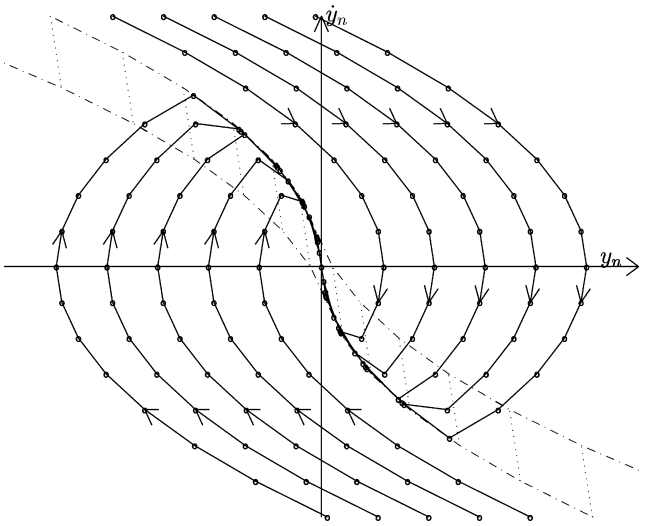


Fig. 8. System trajectories when controller $C3$ is used.

Proof. The state-space dynamic model of the trajectory smoother of Fig. 7 when Eq. (14) holds, is the following:

$$\begin{bmatrix} y_{n+1} \\ \dot{y}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} y_n \\ \dot{y}_n \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} u_n. \tag{15}$$

When controller $C3$ is used, the system trajectories in the phase plane $(y_n, \dot{y}_n)$ are shown in Fig. 8. When state-space transformation (13) is considered, system (15) transforms as follows:

$$\begin{bmatrix} z_{n+1} \\ \dot{z}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} z_n \\ \dot{z}_n \end{bmatrix} + \begin{bmatrix} \frac{1}{U} \\ \frac{1}{U} \end{bmatrix} u_n. \tag{16}$$

From Eqs. (10)–(13), it is evident that the control signal u_n is not saturated ($|u_n| < U$) only if the system state $\mathbf{z} = (z_n, \dot{z}_n)$ belongs to the boundary layer $|\sigma_n| < 1$. The shape of the boundary layer, when controller $C3$ is used, is shown in Fig. 9. Without loss of generality, in the following only the case $z_n < 0$ is considered. The boundary layer is delimited by the following points:

$$A_m = \left(-\frac{m(m-1)}{2}, m\right), \quad B_m = \left(-\frac{m(m-1)}{2}, m-2\right),$$

where $m \in \{1,2,\dots\}$. Let S_m denote the vertical stripe of the phase plane contained between points A_m and A_{m+1} ($m \in \{1,2,\dots\}$):

$$S_m = \left\{\mathbf{z}_n : \frac{m(m-1)}{2} \le |z_n| < \frac{m(m+1)}{2}\right\}. \tag{17}$$

Given any state $\mathbf{z}_n$, from (17) it follows that the index m of the corresponding stripe S_m satisfies the inequalities

$$\frac{-1+\sqrt{1+8|z_n|}}{2} < m \le \frac{1+\sqrt{1+8|z_n|}}{2}. \tag{18}$$

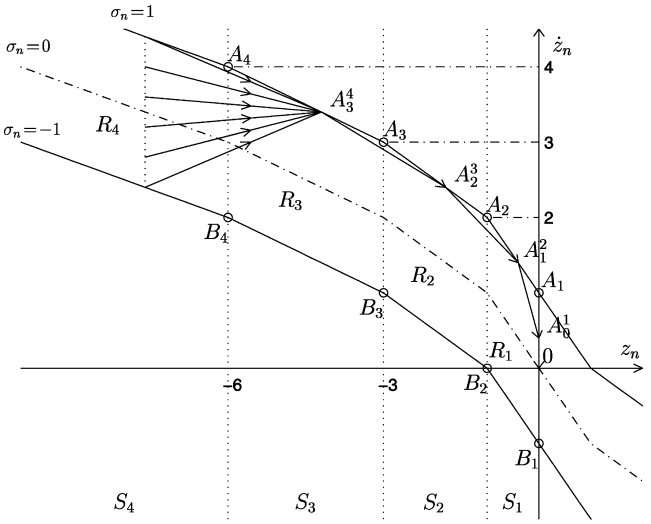


Fig. 9. Boundary layer when controller $C3$ is used: $\overline{A_1 0}$ is the eigenvector; when $\sigma_n = 1$ we have $A_3^4 \to A_2^3 \to A_1^2 \to A_0^1 \to 0$.

Only one integer value m is contained within the interval (18). This value m can be evaluated by means of relation (12). Let R_m denote the region of the phase plane contained within stripe S_m and delimited by points A_m, A_{m+1}, B_{m+1} and B_m, see Fig. 9. Region R_m is defined as follows:

$$R_m = \{\mathbf{z}_n\}: \begin{cases} z_n = m\left[\dfrac{2\alpha + m - 1}{2} - \dot{z}_n\right], \\ \dot{z}_n \in [m + \alpha - 1, m + \alpha[, \\ \alpha \in [-1, 1] \end{cases} \tag{19}$$

Let A_m^{m+1} denote the set of points $\mathbf{z}_n$ which belong to segment $\overline{A_m A_{m+1}}$. The definition of set A_m^{m+1} is obtained from (19) when $\alpha = 1$:

$$A_m^{m+1} = \{\mathbf{z}_n\}: \begin{cases} z_n = m\left[\dfrac{m+1}{2} - \dot{z}_n\right], \\ \dot{z}_n \in [m, m+1[. \end{cases} \tag{20}$$

When $\mathbf{z}_n \in R_m$, the control signal u_n is

$$u_n = -U\left[\dot{z}_n + \frac{z_n}{m} - \frac{m-1}{2}\right]. \tag{21}$$

Applying control (21) to (16), the system evolves as follows:

$$\begin{bmatrix} z_{n+1} \\ \dot{z}_{n+1} \end{bmatrix} = \begin{bmatrix} m-1 \\ -1 \end{bmatrix} \frac{z_n}{m} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \frac{m-1}{2}. \tag{22}$$

Remark. State $\mathbf{z}_{n+1}$ depends only on the value of z_n, whatever the value of $\dot{z}_n$. By substituting (19) in (22), one

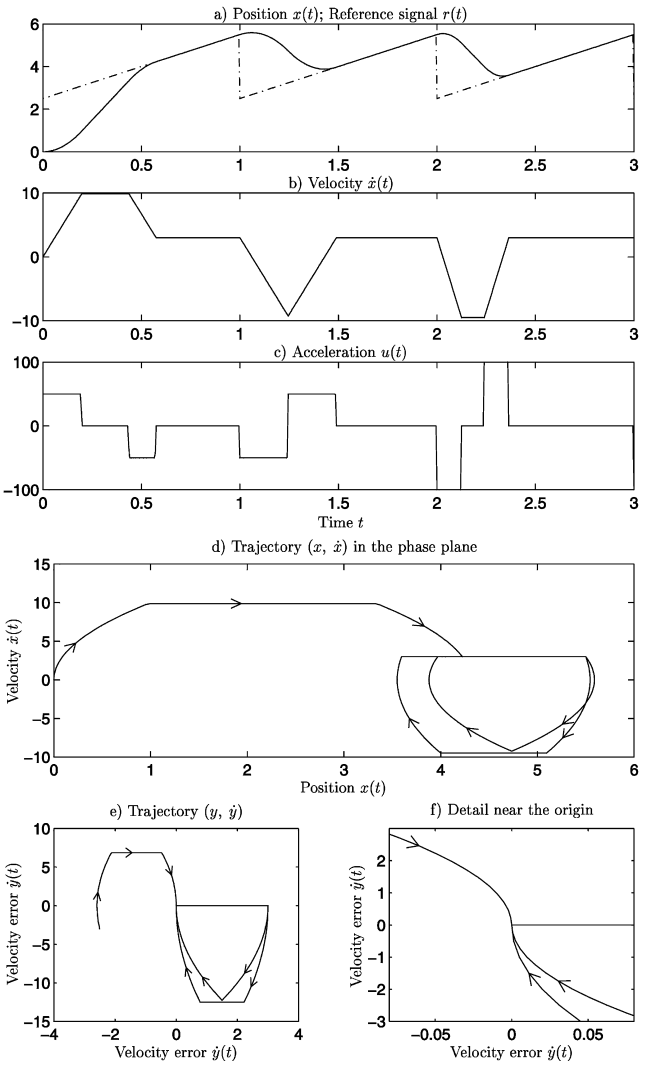


Fig. 10. Simulation results obtained when controller $C2$ is used: (a) position $x(t)$ and reference signal $r(t)$; (b) velocity $\dot{x}(t)$; (c) acceleration $u(t)$; (d) trajectory $(x, \dot{x})$ in the phase plane; (e) trajectory $(y, \dot{y})$ of the tracking error; (f) detail in the vicinity of the origin.

directly obtains that for any starting point $\mathbf{z}_n \in R_m$, the point $\mathbf{z}_{n+1}$ belongs to the segment

$$A_{m-1}^m = \{\mathbf{z}_{n+1}\}: \begin{cases} z_{n+1} = (m-1)\left[\dfrac{m}{2} - \dot{z}_{n+1}\right], \\ \dot{z}_{n+1} \in [m-1, m[. \end{cases} \tag{23}$$

Region R_m is mapped into segment A_{m-1}^m by means of the control law (21). During the next sampling period T, the control law (21) maps segment A_{m-1}^m (23) into segment A_{m-2}^{m-1}, and so on. It can be easily shown that the following sequence is obtained:

$$R_m \to A_{m-1}^m \to A_{m-2}^{m-1} \to \cdots \to A_0^1 \to 0.$$

It is interesting to observe that all the points which belong to segment A_0^1 (i.e. $m = 1$) are exactly mapped into the origin. The system state $\mathbf{z}_n$ reaches the origin in

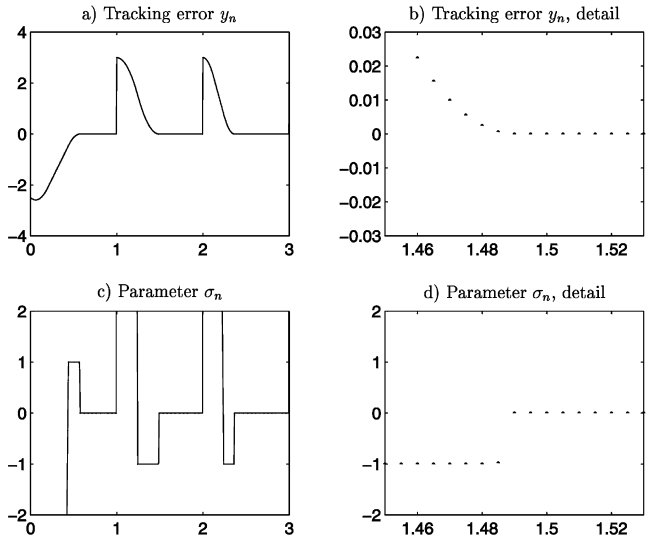


Fig. 11. Simulation results obtained when controller $C2$ is used: (a) tracking error y_n; (b) tracking error, detail; (c) parameter σ_n; (d) parameter σ_n, detail.

a finite number of sampling periods T. The final steady-state condition $\mathbf{z}_n = 0$ is reached in minimum time, without overshoot and with a "dead-beat" dynamics. Clearly, $C3$ is the "optimal" controller which solves the Control Problem. □

The simulation results obtained by using controller $C3$ in the same conditions as those of the continuous-time case (Fig. 5), are reported in Figs. 10 and 11. Owing to the "dead-beat" behaviour of the controlled system, the origin is reached in "minimum time". As a result, the drawback of a slow final approach, typical of the continuous-time case, is completely avoided.

The time behaviour of the tracking error y_n and of parameter σ_n is reported in Fig. 11a and c. Details of the same variables are shown in Fig. 11b and d. These details definitely show that the tracking dynamics of the system is of dead-beat type.

5. Simulation example

Let us consider the servo system of Fig. 12: an electric DC motor is used to control the horizontal position of a drill that moves parallel to an assembly line. The tool must track exactly the position r of the approaching object, perform the prescribed manipulation and go back to track the next object. Motor torque τ_n and motor angular velocity ω are bounded as: $|\tau_n| \le \tau_M$, $|\omega| \le \omega_M$.

The control scheme is shown in Fig. 13. It is composed by the "Motor", the trajectory smoother "TS", the feed-forward action "FF" and the linear controller "LR".

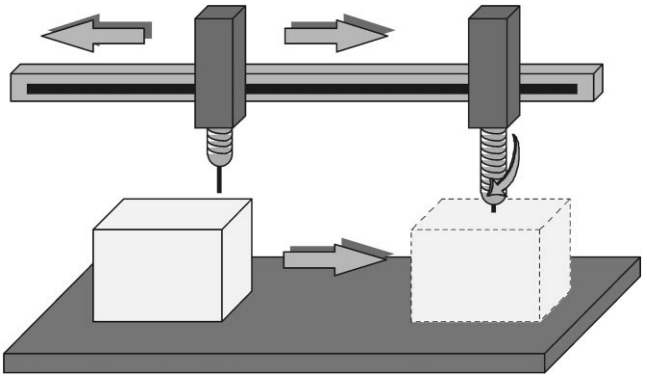


Fig. 12. A drill moving parallel to an assembly line.

Only the mechanical equations of the motor have been considered: J and b represent the inertia and the linear friction coefficients of the overall plant (motor + tool). Variables r_n, x_n and θ_n denote, respectively, the desired angular position, the generated trajectory and the sampled angular position of the motor; $e_n = x_n - \theta_n$ and $\dot{e}_n = \dot{x}_n - \omega_n$ are the position and velocity errors with respect to the smooth reference signals x_n and $\dot{x}_n$. Torque τ_n used to control the system is expressed as

$$\tau_n = \underbrace{K_p e_n + K_v \dot{e}_n}_{\text{LR}} + \underbrace{J\ddot{x}_n + b\dot{x}_n}_{\text{FF}}. \tag{24}$$

The controller is composed by the linear feed-back control LR and the feed-forward control FF. The FF control directly uses output derivatives from the trajectory smoother. It is easy to verify that with a proper choice of the design parameters K_p and K_v it is possible to impose arbitrarily the system dynamics: when $K_p = (b + K_v)^2/4J$ the two poles of the system are real and coincident.

The simulation results reported in Fig. 14 have been obtained by using the parameters (MKS units): $b = 0.02$, $J = 0.3$, $K_v = 40$, $U = \tau_M/J = 50$, $K_p = 1334$, $T = 0.005$, $\dot{x}_M = 100$, $r_n = 2.5 + 3[nT - \text{Int}(nT)]$, $\dot{r}_n = 3$ and zero initial conditions. The set point r_n and the discrete trajectory smoother used for this simulation are exactly the same as those used in the simulation of Fig. 10: the time behaviours of signals x_n, $\dot{x}_n$ and $\ddot{x}_n$ are shown in Fig. 10a–c. The maximum allowed torque τ_M has been changed on-line during the simulation: at instant $t = 2$ its value switches from $\tau_M = 12$ to 30. Position and velocity errors e_n, $\dot{e}_n$ and control torque τ_n are shown in Fig. 14. Position and velocity errors are very small. Control action τ_n is mainly held by the feed-forward action: $\tau_n \simeq J\ddot{x}_n$. The system transients are well damped.

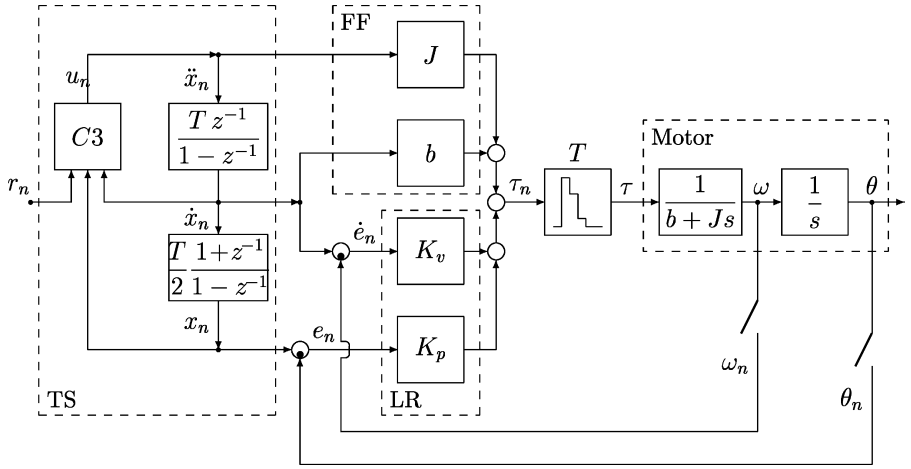


Fig. 13. Position control of a DC motor: block diagram used in simulation.

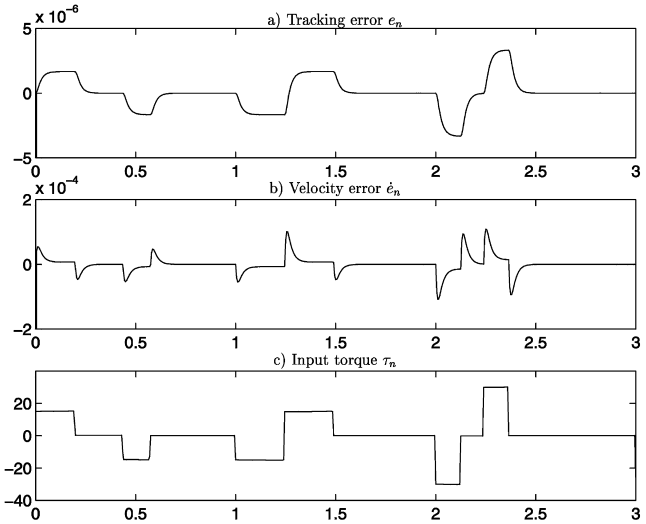


Fig. 14. Position control of a DC motor. Simulation results: (a) position error e_n; (b) velocity error $\dot{e}_n$; (c) input torque τ_n.

6. Conclusions

It has been shown that using VSC techniques it is possible to design a nonlinear trajectory smoother whose state variables can be freely changed even during system operation. Near "time-optimal" response is ensured without overshoot. Bounded state-variables are available to implement feed-forward compensations. An application example has been reported to show system performance. The presented solution is limited to second-order filter. The generalisation to the higher dimensional case, when constraints are given also on the third derivative, and to multiple synchronised filters, is under development.

References

Brady, M. (1982). Trajectory planning. In M. Brady et al., *Robot motion: planning and control* (pp. 221–243). Cambridge, MA: MIT Press.

Guarino Lo Bianco, C., Tonielli, A., & Zanasi, R. (1996). Nonlinear trajectory generator. *IECON'96 — 22nd annual international conference on the IEEE Industrial Electronics Society*, Taipei, Taiwan.

Leonhard, W. (1985). *Control of electrical drives*. Berlin: Springer.

Lewis, F. L. (1986). *Optimal control*. New York: Wiley.

Rossi, C., & Tonielli, A. (1992). A unifying approach to robust control of electrical motor drives. *Proceedings of international conference on industrial electronic control instruments and automatics*, *IECON'92*, San Diego (pp. 95–100).

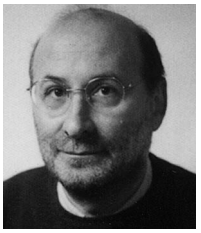
Rossi, C., & Tonielli, A. (1994). Robust control of permanent magnet motors: VSS techniques lead to simple hardware implementation. *IEEE Transactions on Industrial Electronics*, *41*, (4).

Utkin, V. I. (1977). Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, *22*, 212–222.

Zanasi, R., Guarino Lo Bianco, C., & Tonielli, A. (1998). Nonlinear filter for smooth trajectory generation. *NOLCOS'98*, University of Twente, Enschede, The Netherlands.

Roberto Zanasi was born in Bomporto (MO), Italy, in 1959. He graduated with honors in Electrical Engineering in 1986 at the University of Bologna. In 1992 he received from the same University the Ph.D. degree in System Engineering. From 1994 to 1998 he was a Researcher in Automatic Control in the Department of Electronics, Computer and System Science, University of Bologna. Since 1998 he has been working as Associate Professor of Automatic Control in the Dipartimento di Scienze dell'Ingegneria, University of Modena and Reggio Emilia. He held the position of Visiting Scientist at the IRIMS of Moscow (1991), at the MIT of Boston (1992) and at the Université Catholique de Louvain (1995). His research interests include linear and nonlinear control theory, variable-structure control, integral control, robotics, mathematical modelling and simulation.

Corrado Guarino Lo Bianco was born in Sassari (Italy) on May 3, 1964. He graduated with honors in electronic engineering in 1989 from the University of Bologna, Italy. In 1994 he received the Ph.D. degree in Control System Engineering from the same University. At the moment, he is with the Dipartimento di Ingegneria dell'Informazione of the University of Parma as Research Associate on System Engineering. He is involved in researches on Variable Reluctance motors, power devices thermal analysis, smooth profile generation for motion control, robust control design via semi-infinite optimization, genetic algorithms and interval analysis.



Alberto Tonielli was born in Tossignano, Bologna, Italy, on April 1, 1949. He received the Dr. Ing. degree in electronic engineering from the University of Bologna, Italy, in 1974. In 1975, he joined the Department of Electronics, Computer and System Sciences (DEIS) of the University of Bologna, with a grant from the Ministry of Public Instruction. In 1979 he started teaching as an Assistant Professor. In 1980, he became a Permanent Researcher. In 1981, he spent two quarters at the University of Florida, Gainesville, as a Visiting Associate Professor. Since 1985, he has been an Associate Professor of Control System Technologies at the University of Bologna. His current research interests are in the fields of nonlinear and sliding mode control for electric motors, nonlinear observers, robotics and DSP-based control architectures.