

C^2 continuous time-dependent feedrate scheduling with configurable kinematic constraints

Carlotta Giannelli, Duccio Mugnaini, Alessandra Sestini

Abstract

We present a configurable trajectory planning strategy on planar paths for offline definition of time-dependent C^2 piecewise quintic feedrates. The more conservative formulation ensures chord tolerance, as well as prescribed bounds on velocity, acceleration and jerk Cartesian components. Since the less restrictive formulations of our strategy can usually still ensure all the desired bounds while simultaneously producing faster motions, the configurability feature is useful not only when reduced motion control is desired but also when full kinematic control has to be guaranteed. Our approach can be applied to any planar path with a piecewise sufficiently smooth parametric representation. When Pythagorean-hodograph spline curves are considered, the corresponding accurate and efficient CNC interpolator algorithms can be exploited.

Keywords. Trajectory planning, feedrate, acceleration, jerk, splines, CNC interpolators, Pythagorean-hodograph curves.

1 Introduction

Trajectory planning is a fundamental issue in several fields related to automation, such as robotics, digital animation and computer aided manufacturing. It requires the identification of a feasible path and the definition of an admissible time law. Even if conventional controllers were restricted to piecewise linear or circular paths, modern software systems enable the design of smooth curvilinear paths where zero velocity values can be avoided. Once the path is identified, the design of a suitable time law

Carlotta Giannelli, Alessandra Sestini, Dipartimento di Matematica e Informatica “U. Dini,” Università di Firenze, Viale Morgagni 67/A, I-50134 Firenze, Italy, e-mail: carlotta.giannelli@unifi.it, alessandra.sestini@unifi.it

Duccio Mugnaini, Dipartimento di Scienze e Alta Tecnologia, Università degli Studi dell’Insubria, Via Valleggio 11, I-22100 Como, Italy, e-mail: dmugnaini@uninsubria.it

is crucial to achieve high-speed control. This task corresponds to specify the object speed along the path, generally called *feedrate* function.

The final output of a trajectory planning scheme is a set of *reference points* for the control system. They indicate the prescribed machine positions with respect to uniform time spacing and constitute the input values for the controller to guide the movement of the machine. The feedrate scheduler provides the velocity profile for the planned path by taking into account the given geometric path and kinematic constraints. A characterizing feature of any feedrate scheduler relies on the level of kinematic control it ensures (or tries to ensure). This may strongly vary between different approaches. For example, full jerk control is usually not guaranteed, since the contribution of the jerk centripetal component is often ignored. The continuity properties of the motion represent another important aspect of the control strategy. In particular, since the time law should avoid unnecessary vibrations (that may also cause mechanical damages), smooth motions are strongly preferable. Only schedulers providing C^2 feedrate functions allow the possibility of ensuring acceleration and jerk continuity on sufficiently smooth path segments. This is an important feature of the scheduler since sudden changes or reversal in the kinematic profiles are avoided in the motions it produces, see e.g. [8].

In view of their compatibility with standard spline representation of Computer Aided Design systems, different kind of Non-Uniform Rational B-splines (NURBS) interpolators were recently proposed. The scheduling computation can be executed before or during the interpolation phase, leading to *offline* or *online* approaches, respectively. While the offline interpolator avoids real-time interaction at the price of using additional storage of computed data, the online solution requires more computational power at running time. Examples of offline and online NURBS interpolators with chord error, acceleration and jerk limitations were presented in [11, 16] and [1], respectively.

The feedrate profile is often expressed as a time-dependent function, but representations with respect to the curve parameter or the curvature are also considered. For example, when its representation with respect to the curvature profile is considered, the chord error control is highly facilitated. This is an important feature in CNC machining and processing, see, e.g., [7, 17]. Real-time interpolator algorithms for different feedrate variations (parameter-dependent, curvature-dependent and hybrid formulations) were also recently considered in connection with a corner rounding approach for high-speed machining [6]. A time-dependent feedrate instead simplifies the development of controllers that take into account different kinematic constraints. Among the wide set of references in this setting, trigonometric feedrate profiles have been considered by several authors in connection with different continuity requirements, see e.g., [11] for the C^1 case and [16] for the case of higher continuity.

In this paper we present a configurable offline trajectory planning strategy that

provides a time-dependent C^2 quintic spline feedrate profile. The more conservative formulation of the scheduler ensures chord tolerance control, as well as certified velocity, acceleration and jerk bounds. Since the less restrictive formulations of our strategy can usually still ensure all the desired bounds while simultaneously producing faster motions, the configurability feature is useful not only when reduced motion control is desired, but also when full kinematic control has to be guaranteed. In all formulations the scheduler first scans the path to detect a set of special points that subdivide the curve into successive curve blocks. The feedrate function is defined with a typical trapezoidal profile in any curve block. The maximal feedrate value of each curve block and the feedrate values at certain special points are properly initialized by taking into account the kinematic constraints and the block length. The feedrate is then defined as a time-dependent C^2 piecewise quintic polynomial function. Specific bounds on the first and second derivatives of the feedrate are suitably considered in the minimization of the block traversal time. The configurable feature of the scheduler allows us to specify if only acceleration control is required, or if either tangential or full jerk limitations should be also taken into account. Another possibility offered by our strategy enables the choice between maximal and mean values of curvature in each curve block, a key geometric aspect for the design of versatile kinematic control.

In order to exploit accurate and reliable real-time interpolators, the class of Pythagorean-hodograph (PH) curves is here considered [3] for specifying the considered planar paths. The polynomial nature of the parametric speed of a PH curve enables an exact computation of its arc-length and leads to robust processing algorithms. By avoiding repeated numerical approximations, the precision of the method is strongly enforced. The results obtained with Pythagorean-hodograph spline curves in CNC applications have confirmed the potential of this approach, see e.g., [13, 14, 15], as well as [3] and references therein. The complete trajectory planning scheme should properly integrate the feedrate scheduling with the geometric part of the motion. Planar algorithms for approximating and interpolating arbitrary input data with PH spline schemes with different order of continuity are available, as for example [4, 5]. More recently, tangent and curvature continuous path planning with obstacle avoidance techniques based on planar PH quintic spline curves were also proposed in [9] and [2], respectively. These path planning schemes can be suitably combined with the feedrate scheduler here proposed to properly define an optimal trajectory for a given path [12].

The structure of the paper is as follows. Section 2 introduces some preliminary concepts and the kinematic constraints here considered. The configurable feedrate scheduler is presented in Section 3, while the CNC interpolator algorithm for PH curves is reviewed in Section 4. Finally, Section 5 illustrates the performance of our approach on a selection of significative examples, while Section 6 concludes the paper with some final remarks.

2 Preliminaries and kinematic constraints

We assume that a planar path is given in parametric form, $\mathbf{r}(\xi), \xi \in [\xi_L, \xi_R]$ with \mathbf{r} sufficiently smooth. We are interested in determining a suitable time law for traversing the given path. More precisely, if t denotes the time and s denotes the *arc length*, we define the *feedrate* $v(t)$ as

$$v(t) := \frac{ds}{dt}(t).$$

The corresponding global traversal time T is then implicitly given by

$$\int_0^T v(t) dt = S, \quad (1)$$

where S is the total length of the path. Note that, when the feedrate is given as a function of the parameter ξ , the global traversal time T can be explicitly obtained from the following formula:

$$T = \int_{\xi_L}^{\xi_R} \frac{\sigma}{v} d\xi,$$

where $\sigma := ds/d\xi = |\mathbf{r}'(\xi)|$ denotes the curve *parametric speed*. Since we want to prescribe a time-dependent feedrate profile, we will use the relation in (1).

Let $\mathbf{v} := \dot{\mathbf{r}}$ be the velocity vector along the trajectory (with the dot symbol denoting a derivative with respect to time). Being defined at each regular point as

$$\mathbf{v} = v \mathbf{t}, \quad (2)$$

where $\mathbf{t} := \mathbf{r}'/|\mathbf{r}'|$ is the unit tangent vector to the curve (with the prime symbol denoting derivative with respect to the parameter ξ), the absolute value $|v|$ of the feedrate is simply the velocity module. Consequently, the absolute value of both the Cartesian components of the velocity can be bounded from above by V_m by requiring

$$0 \leq v \leq V_m, \quad (3)$$

where V_m is called the *commanded feedrate*. The required positivity for v means that we are interested only in motions that avoid any inversion of direction.

The acceleration vector $\mathbf{a} := \ddot{\mathbf{r}}$ is well defined at any regular point and its expression is given by

$$\mathbf{a} = \dot{v} \mathbf{t} - v^2 \kappa \mathbf{n}, \quad (4)$$

since $d\mathbf{t}/ds = -\kappa \mathbf{n}$, where

$$\mathbf{n} := \mathbf{t} \times \mathbf{z} \quad (\text{unit normal}) \quad \text{and} \quad \kappa := \frac{(\mathbf{r}' \times \mathbf{r}'') \cdot \mathbf{z}}{\sigma^3} \quad (\text{curvature})$$

with \mathbf{z} denoting a unit vector orthogonal to the plane of the path.

By differentiating the expression of \mathbf{a} given in (4), we obtain the expression of the *jerk* vector $\mathbf{j} := \ddot{\mathbf{r}}$ at any regular point

$$\mathbf{j} = (\ddot{v} - v^3 \kappa^2) \mathbf{t} - (3v\dot{v}\kappa + v^3 \kappa' / \sigma) \mathbf{n}, \quad (5)$$

in terms of the feedrate, its first and second derivative, and other geometric quantities connected to the given path.

Since in computer numerical control the motion is separately prescribed on each axis, the general interest is in bounding the absolute value of the Cartesian components of the acceleration vector \mathbf{a} and, possibly, also those of the jerk vector \mathbf{j} . In view of (4), we can guarantee that both $|a_x| \leq A_m$ and $|a_y| \leq A_m$ only by requiring

$$|\dot{v}| \leq A_t := p_a A_m, \quad v^2 |\kappa| \leq A_c := \sqrt{1 - p_a^2} A_m, \quad (6)$$

with $p_a \in (0, 1)$. A typical choice is $p_a = 1/\sqrt{2}$. Analogously, from the jerk expression in (5) we obtain that $|j_x| \leq J_m$ and $|j_y| \leq J_m$ is guaranteed only if

$$|\ddot{v}| \leq J_{t_1} := q_j p_j J_m, \quad v^3 \kappa^2 \leq J_{t_2} := (1 - q_j) p_j J_m, \quad (7)$$

and

$$3|\kappa|A_t v + |\kappa'|/\sigma v^3 \leq J_c := \sqrt{1 - p_j^2} J_m, \quad (8)$$

with $p_j, q_j \in (0, 1)$. Typical choices in this case are $p_j = 1/\sqrt{2}$ and $q_j = 1/2$. Note that the inequality in (8) related to the jerk centripetal component control, is often not considered in the literature, see [1] for one example. As noticed in the appendix of [1], its left hand side is an increasing polynomial in v . Thus there exists exactly one positive value v_{jc} for v such that (8) holds if and only if $v \leq v_{jc}$. When $\kappa' \neq 0$, the expression of v_{jc} is given by Cardano's rule,

$$v_{jc} = v_{jc}(\kappa, w) := \sqrt[3]{\lambda_+} + \sqrt[3]{\lambda_-}, \quad \text{with } \lambda_{\pm} = \frac{J_c}{|w|} \pm \sqrt{\left(\frac{J_c}{2|w|}\right)^2 + \left(\frac{\kappa A_t}{|w|}\right)^3}, \quad (9)$$

where $w := \kappa'/\sigma$ is the curvature derivative with respect to the arc length.

In addition to the kinematic control, it is also natural to ensure a suitable control on the *chord error* E_c associated with a given *sampling time* Δt during the overall traversal of the path. Since the chord error is a discrete function whose j -th value $E_c(j)$ is associated with $t_j = j\Delta t$, it necessarily depends on the sampling time Δt . As shown in Figure 1, when a point with non vanishing curvature is considered, $E_c(j)$ is defined as the distance between the osculating circumference through the corresponding

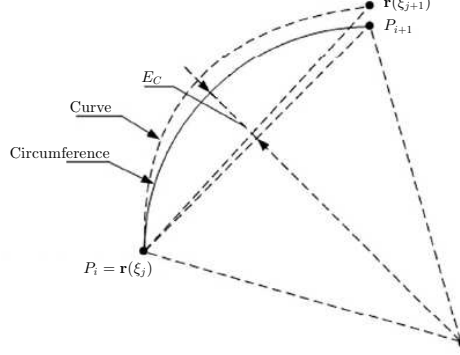


Figure 1: Chordal error $E_c(j)$ at the interpolated point $\mathbf{r}(\xi_j)$.

interpolated point $\mathbf{P}_j = \mathbf{r}(\xi_j)$ and a related circumference chord,

$$E_c(j) := \begin{cases} \frac{1}{|\kappa_j|} - \sqrt{\frac{1}{\kappa_j^2} - \frac{L_j^2}{4}}, & \text{if } \kappa_j \neq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

with $L_j := v(t_j)\Delta t$. This means that $L_j = \|\mathbf{P}_{j+1} - \mathbf{P}_j\|_2$, where \mathbf{P}_{j+1} is the next interpolated point along the circumference, by assuming a constant feedrate equals to $v(t_j)$ and considering that the tool moves on a straight line between any pair of successively interpolated points. Observe that, if $\kappa_j \neq 0$, $E_c(j)$ is well defined only if $L_j \leq 2/|\kappa_j|$, that is only if $v(t_j)|\kappa_j|\Delta t \leq 2$.

A suitable requirement is

$$E_c(j) \leq D, \quad (11)$$

where D is an assigned positive threshold. Clearly, if $D \geq 1/|\kappa_j|$ then (11) holds. On the other hand, when $D < 1/|\kappa_j|$, it can be verified with some computations that the inequality in (11) corresponds to require the following condition on the feedrate

$$v(t_j) \leq \frac{2}{\Delta t} \sqrt{\frac{1}{\kappa_j^2} - \left(\frac{1}{|\kappa_j|} - D\right)^2}. \quad (12)$$

It is interesting to note also that, if $v(t_j) < 2/\Delta t|\kappa_j|$, it is reasonable to use a first order Taylor expansion for the square root in (10) with respect to Δt^2 , leading to

$$E_c(j) \approx \frac{1}{|\kappa_j|} \left[1 - \frac{1}{2} \left(2 - \frac{\kappa_j^2 v_j^2 \Delta t^2}{4} \right) \right] = \frac{\Delta t^2}{8} |\kappa_j| v_j^2.$$

In this case the chord error is then approximately proportional to the absolute value of the normal component of the acceleration.

As previously mentioned in the introduction, we present a configurable strategy that allows six alternative formulations with different level of kinematic constraints. These alternatives are labelled as R_0, R_1, R_2 and S_0, S_1, S_2 and can be collected into two groups: the *relaxed* (R_0, R_1, R_2) and the *strict* (S_0, S_1, S_2) formulations. In any case, the path is preliminarily divided into segments jointly together at tangent or curvature discontinuous points, where the motion is halted. Each segment is then further subdivided into blocks by detecting on each segment a set of special points. In all the relaxed formulations, the mean value of the absolute curvature κ is determined on each block. In the R_2 formulation the mean value of w is also considered. In order to obtain small block traversal times, these values are used to imposed the selected kinematic constraints in the block only with respect to a mean value. On the other hand, in all strict formulations the maximal values of the same geometric quantities are determined in each block, in order to ensure *a priori* the desired kinematic constraints. These always include the bound in (12) for chord error control. In addition, the R_0 and S_0 formulations take into account only the bounds in (6) which concern acceleration, while R_2 and S_2 adds to (12) and (6) also the bounds in (7) and (8) for full jerk control. R_1 and S_1 are hybrid formulations since they add to (12) and (6) only the bounds related to the tangential component of the jerk given by (7).

3 C^2 time-dependent feedrate scheduling

In any possible configuration and for all curve segments, the scheduler defines a feedrate function v such that

$$0 \leq v \leq V_m, \quad |\dot{v}| \leq A_t, \quad |\ddot{v}| \leq J_{t_1}. \quad (13)$$

These inequalities, however, are not sufficient to ensure the desired chord, acceleration, and jerk control. In addition to them, by considering the expressions in (4), (5), and (10), we can easily derive that, to ensure full kinematic control, the feedrate v should satisfy the upper bound $v \leq V_b(\kappa, w)$, where $w = \kappa'/\sigma$ and

$$V_b(\kappa, w) := \min \left\{ \frac{2}{\Delta t} \sqrt{\frac{1}{\kappa^2} - \left(\frac{1}{|\kappa|} - D \right)^2}, \sqrt{\frac{A_c}{|\kappa|}}, \sqrt[3]{\frac{J_{t_2}}{\kappa^2}}, v_{jc}(\kappa, w) \right\}, \quad (14)$$

with $v_{jc}(\kappa, w)$ defined as in (9). Note that κ' is assumed continuous in the segment of reference. Clearly, if we are not interested in guaranteeing jerk bounds, the minimum on the right hand side of (14) is just between the first two terms.

Let us now present the strategy of our scheduler. The curve is preliminarily decomposed into *segments* whose extremes $\mathbf{B}_j, j = 1, \dots, N - 1$, usually called *breakpoints*, are assigned points where the curve is just G^0 , i.e. corner points, augmented by the initial and final positions $\mathbf{B}_0 = \mathbf{r}(\xi_L)$ and $\mathbf{B}_N = \mathbf{r}(\xi_R)$. At each breakpoint we always set to zero v, \dot{v} , and also \ddot{v} . For brevity, in what follows, we will focus on the generic j -th curve segment by removing the subscript (j) . We will also assume that the time is reported to zero whenever a new segment is encountered.

On each curve segment the scheduler first detects a set of special points. These points allow us to subdivide the segment into blocks where a feedrate trapezoidal profile is defined. The special points are of different type for the relaxed and strict formulations and they are called *critical* and *crossing* points, respectively. In the first case, they correspond to points where the absolute curvature has a local maximum exceeding a critical curvature value κ_{cr} . In the second case instead, they correspond to points where $|\kappa| - \kappa_{cr}$ has a sign change. The critical curvature is defined as

$$\kappa_{cr} := \min \left\{ \frac{8D}{V_m^2 \Delta t^2 + 4D^2}, \frac{A_c}{V_m^2}, \sqrt{\frac{J_{t_2}}{V_m^3}} \right\}, \quad (15)$$

for R_1, S_1, R_2 and S_2 . By assuming (13), this is the curvature value such that, if a point of the path is encountered with a feedrate greater than V_m , one among the chord error, the normal component of acceleration, and the tangential component of the jerk can exceed the related bounds. On the other hand, for the formulations R_0 and S_0 , κ_{cr} reduces to the minimum between the first two terms on the right hand side of (15), since they do not care of jerk bounds. The special points are denoted in the following as $\mathbf{P}_1, \dots, \mathbf{P}_{M-1}$, while \mathbf{P}_0 and \mathbf{P}_M will denote the endpoints of the considered segment.

3.1 The *relaxed* formulations of the scheduler

The first configurations of the scheduler are called *relaxed* (R) formulations because they consider only mean values for κ and w to try to ensure the desired kinematic control on each curve block. As mentioned in the previous section, R_0 takes into account only acceleration bounds, while R_1 adds also the two bounds in (7) and R_2 requires both (7) and (8).

3.1.1 Feedrate initialization at critical points

We start by initializing the feedrate values at any critical point $\mathbf{P}_i, i = 0, \dots, M$. At \mathbf{P}_0 and \mathbf{P}_M we prescribe zero values for v, \dot{v} , and \ddot{v} . At any $\mathbf{P}_i, i = 1, \dots, M - 1$, instead, a sufficiently small positive value $V_i < V_m$ of the feedrate is assigned for v , while \dot{v} and \ddot{v} are still set to zero. In particular, in order to ensure the desired kinematic control

at these points, the value $\hat{V}_i < V_m$ is chosen with R_1 and R_2 to initialize V_i

$$\hat{V}_i := V_{r_1}(\kappa_i), \quad (16)$$

where κ_i is the value of κ at \mathbf{P}_i and

$$V_{r_1}(\kappa) := \min \left\{ \frac{2}{\Delta t} \sqrt{\frac{1}{\kappa^2} - \left(\frac{1}{|\kappa|} - D \right)^2}, \sqrt{\frac{A_c}{|\kappa|}}, \sqrt[3]{\frac{J_{t_2}}{\kappa^2}}, \right\}. \quad (17)$$

Not being interested in jerk control, the R_0 formulation fixes $\hat{V}_i = V_{r_0}(\kappa_i)$, where

$$V_{r_0}(\kappa) := \min \left\{ \frac{2}{\Delta t} \sqrt{\frac{1}{\kappa_i^2} - \left(\frac{1}{|\kappa|} - D \right)^2}, \sqrt{\frac{A_c}{|\kappa|}} \right\}. \quad (18)$$

Since κ' vanishes at critical points and \dot{v} is set to zero, the centripetal jerk vanishes at these points. Furthermore, since also \ddot{v} is set to zero, the tangential acceleration vanishes. This implies that in (17) we could replace A_c and J_{t_2} with A_m and J_m , respectively, see for example [17]. We did not consider this substitution in our scheduler since no advantages were observed in the experiments by its implementation.²

Note, however, that the V_i values can be further reduced for compatibility reasons connected to the lengths $S^{(i)}, i = 1, \dots, M$, see Remark 1.

3.1.2 Initialization of maximal feedrate value on a curve block

Once the needed information at the critical points are prescribed, we need to specify the feedrate function during the motion between \mathbf{P}_{i-1} and \mathbf{P}_i accordingly. By following [11], we require a *trapezoidal* feedrate profile, i.e. $\dot{v} > 0$ (acceleration phase), then $\dot{v} = 0$ (constant phase), and finally $\dot{v} < 0$ (deceleration phase). However, we use a different analytical definition of the feedrate which produces a C^2 profile. In order to reduce the traversal time, we plan the feedrate so that its central constant phase at maximal feedrate $v = V_c^{(i)}$ is as long as possible. The maximal block feedrate value $V_c^{(i)}$ is initialized with $\hat{V}_c^{(i)}$ defined as follows,

$$\hat{V}_c^{(i)} := \begin{cases} V_b(\kappa_c^{(i)}, w_c^{(i)}) & \text{in case of } R_2, \\ V_{r_1}(\kappa_c^{(i)}) & \text{in case of } R_1, \\ V_{r_0}(\kappa_c^{(i)}) & \text{in case of } R_0, \end{cases} \quad (19)$$

²In our tests we considered paths that admit a C^2 parametric spline representation. Some critical points were often fixed at points where w is discontinuous and can remain high in the left and/or right neighborhood of these points.

where $\kappa_c^{(i)}, w_c^{(i)}$ are the mean absolute values of κ and w between \mathbf{P}_{i-1} and \mathbf{P}_i . Observe that with the R_2 formulation it can happen that $V_c^{(i)} < \hat{V}_{i-1}$ or $V_c^{(i)} < \hat{V}_i$. Thus, after initializing all the block maximal feedrates, we modify the initialization of the feedrate values at critical points as follows,

$$\hat{V}_i = \min\{\hat{V}_c^{(i)}, \hat{V}_c^{(i+1)}, \hat{V}_i\}.$$

Note also that, as in the case of the feedrate values at the critical points, also $V_c^{(i)}$ can be further reduced by the algorithm for compatibility reasons connected to the length $S^{(i)}$ (see the details below). In any case, the feedrate in the i -th block, $i = 1, \dots, M$, will have a central constant profile $v = V_c^{(i)}$ as long as possible, with $V_c^{(i)}$ as near as possible to $\hat{V}_c^{(i)}$. In some cases, depending on the values of $S^{(i)}$, V_{i-1} , and V_i , it can also happen that the three phases collapse to just two (or even one) phases.

3.1.3 Time-dependent feedrate scheduling on a curve block

In order to design a trapezoidal feedrate profile capable to ensure the continuity of the velocity, the acceleration, and the jerk vectors at all points of \mathbf{r} where κ' is continuous, we require v to be a C^2 quintic spline function, that can easily be represented in piecewise Bézier form. We then introduce our time-dependent feedrate definition by firstly associating with the i -th block the time interval

$$[t_s^{(i)}, t_s^{(i)} + T^{(i)}],$$

where $t_s^{(i+1)} := t_s^{(i)} + T^{(i)}$, with $t_s^{(1)} := 0$, and

$$T^{(i)} := T_{acc}^{(i)} + T_{con}^{(i)} + T_{dec}^{(i)},$$

with $T_{acc}^{(i)}$, $T_{con}^{(i)}$, and $T_{dec}^{(i)}$ denoting the time of the acceleration phase, the time of the constant maximal feedrate, and the time of the deceleration phase, respectively.

Let us focus on the general i -th block. Clearly, the corresponding traversal time $T^{(i)}$ cannot be a priori computed, since it depends on the feedrate function $v^{(i)}(t)$, defined as

$$v^{(i)}(t) := \begin{cases} v_{acc}^{(i)}(t) & \text{if } t_s^{(i)} \leq t < t_s^{(i)} + T_{acc}^{(i)}, \\ v_{con}^{(i)}(t) & \text{if } t_s^{(i)} + T_{acc}^{(i)} \leq t < t_s^{(i)} + T_{acc}^{(i)} + T_{con}^{(i)}, \\ v_{dec}^{(i)}(t) & \text{if } t_s^{(i)} + T_{acc}^{(i)} + T_{con}^{(i)} \leq t < t_s^{(i)} + T^{(i)}, \end{cases}$$

where in the constant phase we fix $v_{con}^{(i)}(t) \equiv V_c^{(i)}$. In particular, we assume the following definitions for the feedrate profile in the acceleration phase,

$$v_{acc}^{(i)}(\tau) := V_{i-1} \sum_{j=0}^2 b_j^5(\tau) + V_c^{(i)} \sum_{j=3}^5 b_j^5(\tau), \quad (20)$$

with

$$\tau := \frac{t - t_s^{(i)}}{T_{acc}^{(i)}},$$

and

$$b_j^n(\tau) := \binom{n}{j} \tau^j (1 - \tau)^{n-j}, \quad j = 0, \dots, n,$$

denoting the j -th Bernstein polynomial of degree n . Analogously, in the deceleration phase, we set

$$v_{dec}^{(i)}(\tau) := V_c^{(i)} \sum_{j=0}^2 b_j^5(\tau) + V_i \sum_{j=3}^5 b_j^5(\tau), \quad (21)$$

with

$$\tau := \frac{t - t_s^{(i)} - T_{acc}^{(i)} - T_{con}^{(i)}}{T_{dec}^{(i)}}.$$

Let us consider the constraints we need to impose in order to ensure the fulfillment of the inequalities in (3) and (6), as well as of the following compatibility condition,

$$S_{acc}^{(i)} + S_{dec}^{(i)} \leq S^{(i)}, \quad (22)$$

where $S_{acc}^{(i)}$ ($S_{dec}^{(i)}$) denotes the length of the portion of the i -th block of \mathbf{r} done in the acceleration (deceleration) phase. Note that, since $0 \leq V_j < V_c^{(i)} \leq V_m, j = i-1, i$, considering the properties of the Bernstein basis we can deduce that the inequality in (3) is surely satisfied on both sides. Since

$$\dot{v}_{acc}^{(i)}(t) = \frac{1}{T_{acc}^{(i)}} \frac{dv_{acc}^{(i)}}{d\tau} = 5 \frac{V_c^{(i)} - V_{i-1}}{T_{acc}^{(i)}} b_2^4(\tau),$$

we have $\dot{v}_{acc}^{(i)} \geq 0$. By considering the two inequalities in (6), we obtain

$$5 \frac{V_c^{(i)} - V_{i-1}}{T_{acc}^{(i)}} \max_{\tau \in [0, 1]} b_2^4(\tau) \leq p_a A_m, \quad 5 \frac{V_c^{(i)} - V_{i-1}}{(T_{acc}^{(i)})^2} \max_{\tau \in [0, 1]} |(b_2^4)'(\tau)| \leq J_m,$$

which in turn, in view of the expression of $b_2^4(\tau)$, reduce to

$$\frac{V_c^{(i)} - V_{i-1}}{T_{acc}^{(i)}} \leq \frac{8}{15} p_a A_m, \quad \frac{V_c^{(i)} - V_{i-1}}{(T_{acc}^{(i)})^2} \leq \frac{\sqrt{3}}{10} J_m.$$

If $V_c^{(i)}$ is set equal to $x^{(i)}$, with $\hat{V}_c^{(i)} \geq x^{(i)} \geq \max\{V_{i-1}, V_i\}$, the value of $T_{acc}^{(i)}$ has to be fixed as

$$T_{acc}^{(i)} = T_{acc}^{(i)}(x^{(i)}) := \max \left\{ \frac{15}{8p_a A_m} (x^{(i)} - V_{i-1}), \sqrt{\frac{10}{\sqrt{3}J_m}} (x^{(i)} - V_{i-1}) \right\}, \quad (23)$$

in order to minimize the time needed to arrive at the constant phase. With analogous computations we can verify that $\dot{v}_{dec}^{(i)} \leq 0$ and $T_{dec}^{(i)}$ has to be fixed as

$$T_{dec}^{(i)} = T_{dec}^{(i)}(x^{(i)}) := \max \left\{ \frac{15}{8p_a A_m} (x^{(i)} - V_i), \sqrt{\frac{10}{\sqrt{3}J_m}} (x^{(i)} - V_i) \right\}, \quad (24)$$

in order to ensure that the inequalities in (6) hold also in the deceleration phase with minimal time for the deceleration. In equations (23) and (24) the dependency of $T_{acc}^{(i)}$ and $T_{dec}^{(i)}$ on the value $x^{(i)} \in [\max\{V_{i-1}, V_i\}, \hat{V}_c^{(i)}]$ to be selected for $V_c^{(i)}$ is emphasized. In particular, we will fix $V_c^{(i)}$ as the maximum value of $x^{(i)}$ in this interval which guarantees the fulfillment of (22). Now, since (1) implies

$$S_{acc}^{(i)} = T_{acc}^{(i)} \int_0^1 v_{acc}^{(i)}(\tau) d\tau, \quad S_{dec}^{(i)} = T_{dec}^{(i)} \int_0^1 v_{dec}^{(i)}(\tau) d\tau,$$

we need to impose that

$$T_{acc}^{(i)} \int_0^1 v_{acc}^{(i)}(\tau) d\tau + T_{dec}^{(i)} \int_0^1 v_{dec}^{(i)}(\tau) d\tau \leq S^{(i)},$$

which, considering (20) and (21), in view of the integration formulas for Bernstein polynomials is equivalent to

$$g^{(i)}(x^{(i)}) \leq 2S^{(i)}, \quad (25)$$

with

$$g^{(i)}(x^{(i)}) := T_{acc}^{(i)}(x^{(i)})[V_{i-1} + x^{(i)}] + T_{dec}^{(i)}(x^{(i)})[V_i + x^{(i)}]. \quad (26)$$

If $x^{(i)} = \hat{V}_c^{(i)}$ fulfills (25), in order to keep the maximal admissible feedrate in the constant phase, we choose $V_c^{(i)} = \hat{V}_c^{(i)}$ with $\hat{V}_c^{(i)}$ defined in (19). Otherwise, considering that $g^{(i)}$ is an increasing function with respect to $x^{(i)}$, we preliminarily check if the following inequality holds,

$$g^{(i)}(\max\{V_{i-1}, V_i\}) \leq 2S^{(i)}. \quad (27)$$

If this is the case, we fix $V_c^{(i)}$ as the unique solution of the equation $g^{(i)}(x^{(i)}) = 2S^{(i)}$ in the interval $[\max\{V_{i-1}, V_i\}, \hat{V}_c^{(i)}]$.

Remark 1. The fulfillment of the inequality in (27) for any i -th curve block, $i = 1, \dots, M$, has to be preliminarily checked. If the inequality does not hold, we modify the initial feedrate values $\hat{V}_i, i = 1, \dots, M$, to lower values $V_i, i = 1, \dots, M$, determined by solving the following constrained minimization problem,

$$\min \sum_{i=1}^{M-1} (V_i - \hat{V}_i)^2 \text{ with } \hat{V}_i \geq V_i > 0, \quad g^{(i)}(\max\{V_{i-1}, V_i\}) - 2S^{(i)} \leq 0.$$

This problem can be numerically solved by an iterative minimization procedure by initializing each V_i with $\min_{i=1, \dots, M} \hat{V}_i$.

3.2 The *strict* formulations of the scheduler

The formulations in the second set of configurations are called *strict* (S) because they impose the selected bounds everywhere. Obviously, this is obtained at the price of defining slower motions.

On each segment we determine a set of inner *crossing points* $\mathbf{P}_1, \dots, \mathbf{P}_{M-1}$, which are points where the quantity $|\kappa| - \kappa_{cr}$ has a sign change. We also denote as \mathbf{P}_0 and \mathbf{P}_M the end points of the considered segment. The curve portion between any pair of successive crossing points of the considered segment is again referred to as a *curve block* and $S^{(i)}$ denotes the length of the i -th curve block, $i = 1, \dots, M$. The block length has to be preliminarily computed for any segment. The philosophy for the feedrate determination is somehow reversed with respect to the relaxed strategy. In this case we first determine a safe maximal feedrate for any curve block, for then selecting the feedrate values at the crossing points.

The maximal feedrate $V_c^{(i)}$ chosen on the i -th block, $i = 1, \dots, M$, is initialized as

$$\hat{V}_c^{(i)} := \begin{cases} V_b(\kappa_{max}^{(i)}, w_{max}^{(i)}) & \text{in case of } S_2, \\ V_{r1}(\kappa_{max}^{(i)}) & \text{in case of } S_1, \\ V_{r0}(\kappa_{max}^{(i)}) & \text{in case of } S_0, \end{cases}$$

where $\kappa_{max}^{(i)}$ and $w_{max}^{(i)}$ are the maximum values assumed by $|\kappa|$ and $|w|$ in the considered block. At the crossing point $\mathbf{P}_i, i = 1, \dots, M - 1$, we initialize the V_i with \hat{V}_i fixed as follows,

$$\hat{V}_i := \min\{\hat{V}_c^{(i)}, \hat{V}_c^{(i+1)}\}.$$

Since the inequality in (27) still needs to be fulfilled, Remark 1 applies again.

The feedrate inside each curve block is then selected with a trapezoidal profile by using the same approach used for the relaxed formulations. With this strategy at least one between the acceleration and deceleration phases disappears in many blocks.

4 Application to interpolators for PH curves

In this paper we are interested in applying the introduced feedrate scheduler on paths described by Pythagorean-hodograph curves since, among the different parametric curves commonly used in computer aided geometric design, PH curves strongly facilitates the design of efficient CNC interpolator algorithms. To avoid loss of generality, we assume that a given planar curve is preliminarily approximated by C^2 quintic PH-splines. The interpolation algorithm used for this aim is the one introduced in [4]. The corresponding trajectory planning scheme is summarized in Figure 2. For brevity, only the relaxed strategy is shown in the diagram.

In the following subsections we briefly introduce planar PH curves and the related interpolator, see [3] for the details.

4.1 Planar Pythagorean-hodograph curves

Planar polynomial curves $\mathbf{r}(\xi) = (x(\xi), y(\xi))$, $\xi \in [0, 1]$ with polynomial parametric speed $\sigma(\xi) = |\mathbf{r}'(\xi)|$ so that

$$|\mathbf{r}'(\xi)|^2 = x'^2(\xi) + y'^2(\xi) = \sigma^2(\xi), \quad (28)$$

are known as Pythagorean-hodograph curves [3]. Although PH curves of degree n have just $n + 3$ degrees of freedom (compared to $2n + 2$ for general degree- n polynomial curves), the Pythagorean condition (28) ensures distinctive properties such as polynomial arc-length functions, rational curvature and unit tangent and normal vectors, closed-form expression for the bending energy, rational offset curves. For a deeper and detailed discussion see [3].

In order to satisfy condition (28), the three polynomials $x'(\xi)$, $y'(\xi)$ and $\sigma(\xi)$ must comprise a Pythagorean triple. Hence, according to the result of Kubota [10], these polynomials must be expressible in terms of other real polynomials $u(\xi)$, $v(\xi)$ and $h(\xi)$ in the form

$$x'(\xi) = h(\xi) [u^2(\xi) - v^2(\xi)], \quad y'(\xi) = 2h(\xi) u(\xi) v(\xi), \quad (29)$$

with corresponding polynomial parametric speed

$$\sigma(\xi) = h(\xi) [u^2(\xi) + v^2(\xi)]. \quad (30)$$

Considering that $\sigma(\xi) = |\mathbf{r}'(\xi)|$, the arc-length function is defined as

$$s(\hat{\xi}) = \int_0^{\hat{\xi}} \sigma(\xi) d\xi$$

which is a polynomial function of the parameter, and, consequently, it can be *exactly* computed.

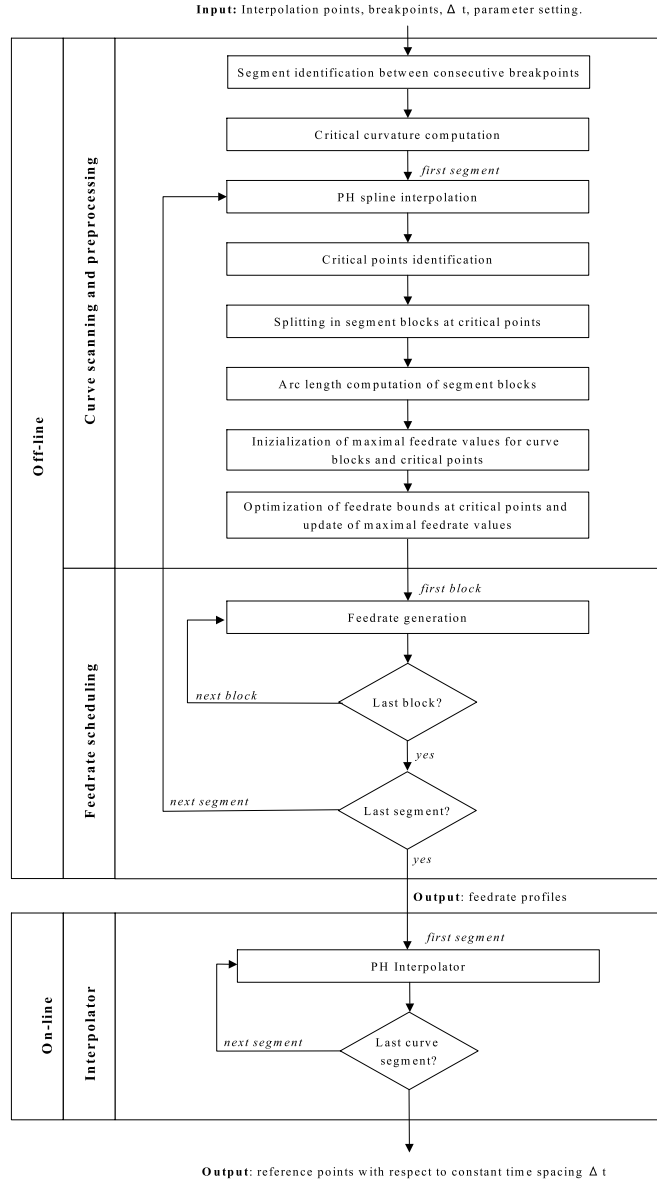


Figure 2: Diagram of the relaxed strategy for feedrate definition.

4.2 PH curve interpolators

PH interpolators based on time-dependent feedrate function $v(t)$ are exploited to generate smooth motion trajectories for a PH spline curve $\mathbf{r}(\xi)$. In order to be efficient, they require $v(t)$ for $t \in [0, T]$ to be non-negative and to have a simple expression $F(t)$ for its indefinite integral such that $v(t) = \dot{F}(t)$, as in the case of our scheduler. The PH curve interpolator implementation is based on the relation:

$$s(\xi) = F(t). \quad (31)$$

By considering a sampling time Δt and a finite sequence of time values $t_0 = \Delta t$, $t_1 = 2\Delta t$, ..., $t_N = N\Delta t$, we can then obtain from (31) the related sequence of parameter values, $0 = \xi_0 < \xi_1 < \dots < \xi_N = 1$ as the root of

$$s(\xi_k) - F(k\Delta t) = 0, \quad k = 1, \dots, N.$$

Each ξ_k corresponds to a reference point $\mathbf{r}(\xi_k)$ on the PH curve. It will be traversed at the corresponding t_k . The iterative Newton-Raphson method can be exploited to find the unique real root of the above equation, namely

$$\xi_k^{(r+1)} = \xi_k^{(r)} - \frac{s(\xi_k^{(r)}) - F(k\Delta t)}{\sigma(\xi_k^{(r)})} \quad (32)$$

with initial guess $\xi_k^{(0)} = \xi_{k-1}$.

For the experiments presented in the next section, we implemented a PH interpolator for PH quintic splines, in line with our choice to assign the planar paths. Particular care is needed whenever the r -th approximation of ξ_k in (32) results greater than 1. This corresponds to a change of polynomial segment in the PH spline at ξ_k . Few iteration of the The Newton-Raphson method was always sufficient to obtain approximations of the reference points of the desired accuracy.

5 Numerical results

In this section three NURBS paths are considered to test the performance of the proposed feedrate scheduling algorithm. They are indicated as *hat*, *starfish*, and *butterfly* and were previously considered in [11, 16, 1], respectively. The three corresponding C^2 PH quintic spline approximations are shown in Figure 3 together with the original NURBS paths. In particular, the hat curve is obtained with three PH spline segments interpolating 13, 6, and 13 points, respectively. For the other two curves instead, we used a single C^2 PH spline interpolating 20 (starfish) or 27 (butterfly) points on the

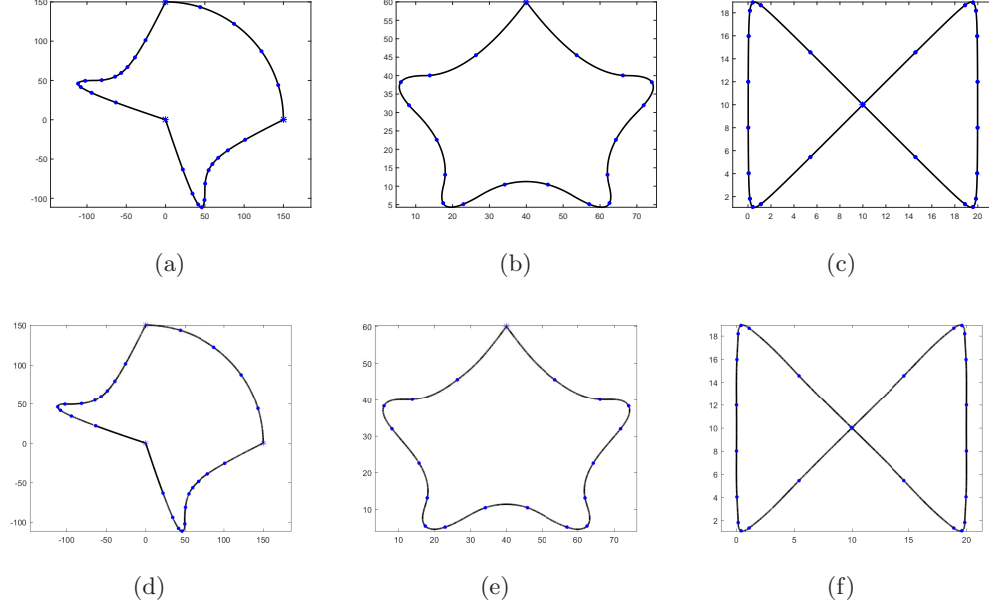


Figure 3: The three curved paths considered in the examples: the NURBS hat (a), the starfish (b), and the butterfly (c). Their PH spline approximations are shown in (d), (e), and (f), respectively. The interpolation points are also shown (blue dots).

original curve. These examples also confirm that PH curves are suitable for smooth approximation of any path.

The feedrate scheduler and the PH interpolator are implemented in Matlab and executed on a personal computer with Intel(R) Core(TM) i5-4300U 1.90GHz CPU. The feedrate scheduling parameters for the three examples are listed in Table 1, while the parameters p_a and p_j are always both set to $1/\sqrt{2}$ and q_j to $1/2$. The additional main parameters used in the experiments — the critical curvature κ_{cr} , the number of special points, and the simulation time — are reported in Table 2 together with the results obtained by the scheduler with all possible configurations of the relaxed and strict strategy. As expected, the strict strategy produces slightly lower motions. Note also that the motion produced by this strategy is more segmented, since the number of crossing points is higher than the number of critical points detected with the relaxed approach. When the simulation times reported in Table 2 are compared with those obtained with other methods, we may observe that our strategy can produce slightly slower motions. This is due to the fact that we always a priori ensures the desired acceleration control.

parameter	symbol	hat	starfish	butterfly
commanded feedrate (mm/s)	V_m	250	200	100
maximum acceleration (mm/s ²)	A_m	800	3000	1000
maximum jerk (mm/s ³)	J_m	26400	60000	50000
maximum chord error (mm)	D	0.0010	0.0010	0.0002
sampling time (s)	Δt	0.002	0.001	0.0005

Table 1: Parameter setting.

	hat	starfish	butterfly
κ_{cr} (mm ⁻¹)	$9.9561 \cdot 10^{-3}$	$5.3033 \cdot 10^{-2}$	$9.051 \cdot 10^{-3}$
<i>relaxed strategy</i>			
# critical points	4	10	4
R_0 simulation time (s)	7.016 s	1.396	1.854
R_1 simulation time (s)	7.045	1.453	1.899
R_2 simulation time (s)	7.159	1.903	2.474
<i>strict strategy</i>			
# crossing points	8	20	12
S_0 simulation time (s)	7.820	1.749	2.237
S_1 simulation time (s)	7.884	2.947	2.405
S_2 simulation time (s)	9.141	3.230	4.810

Table 2: Results of the scheduler obtained with the relaxed (R_0, R_1, R_2) and strict strategies (S_0, S_1, S_2).

All the following figures corresponds to the results obtained with the relaxed formulation of the scheduler, since there was no need to consider the strict formulation to satisfy the prescribed bounds in these cases. In particular we show the results obtained with R_0 for the *hat curve*, R_1 for the *starfish* example, and R_1 for the *butterfly* path.

Unlike [11], we are able to prescribe a C^2 feedrate function. Note that the feedrate defined in [16] is also C^2 but with a less smooth profile which can produce a major vibration during the machining. In the presented results we obtain acceleration continuous motions since we apply our feedrate scheduler to C^2 PH splines whose curvature first derivative has generally a finite jump at the spline knots. Jerk continuity could be obtained by applying the scheduler to curves with continuous curvature derivative.

Figures 4, 7, and 10 show the PH spline approximation, the plot of the absolute value of the curvature, and the reference points obtained for the three test cases. The critical points and the corresponding peaks on the curvature plots are marked in red. The reference points associated to the acceleration, constant, and the deceleration phases are depicted in green, black, and magenta, respectively. Figures 5, 8, and 11

show the chord error and the resulting profiles of $v(t)$, $\dot{v}(t)$ and $\ddot{v}(t)$. The Cartesian components of acceleration and jerk are illustrated in Figures 6, 9 and 12. Again, acceleration, constant, and deceleration phases are indicated in green, black, and magenta, respectively.

For each example, the maximal feedrate $V_c^{(i)}$ is less than V_m in several curve blocks, due to the adaptive behaviour of our strategy described in Section 3.1.2, see Figures 5, 8 and 11. We can appreciate that the greater is the curvature on the i -th block, more restrictive is the adjusted commanded feedrate. This behaviour is highlighted for example on the second block of the hat curve, where the curve presents an high curvature peak and, consequently, the commanded feedrate drops to 115.

Concerning the acceleration control, for all considered examples the absolute value of the x-y acceleration components are always below A_m , as shown in the first row of Figures 6, 9, and 12. In particular, we can observe that on the hat curve the acceleration components values are rather below the prescribed threshold. In [11] instead there exists one point where the threshold is not respected. Furthermore the maximum chord error $D = 10^{-3}$ mm is always fulfilled in our examples since the maximal chordal error is $O(10^{-4})$, see Figures 5(a), 8(a), and 11(a). The reference points produced by the PH interpolator with the sampling time Δt reported in Table 1 are show on the right of Figures 4, 7 and 10.

Finally, for all the three considered tests, Figures 5, 8 and 11(d) and Figures 6, 9 and 12 show the obtained profiles of \ddot{v} and the x-y jerk components, respectively. The desired control on \ddot{v} is ensured by our motion in all the cases. The x-y jerk components are also everywhere suitably controlled for each test.

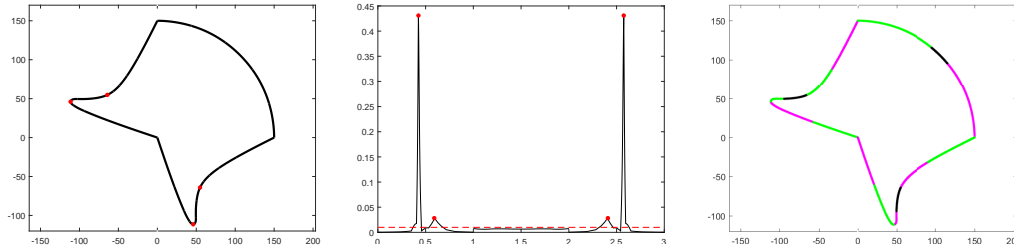


Figure 4: Hat curve example: PH spline approximation (left), absolute curvature plot and critical curvature value (center, black and dashed red lines), and reference points (right). The critical points on the curve and the corresponding peaks on the curvature plot are also shown (red dots).

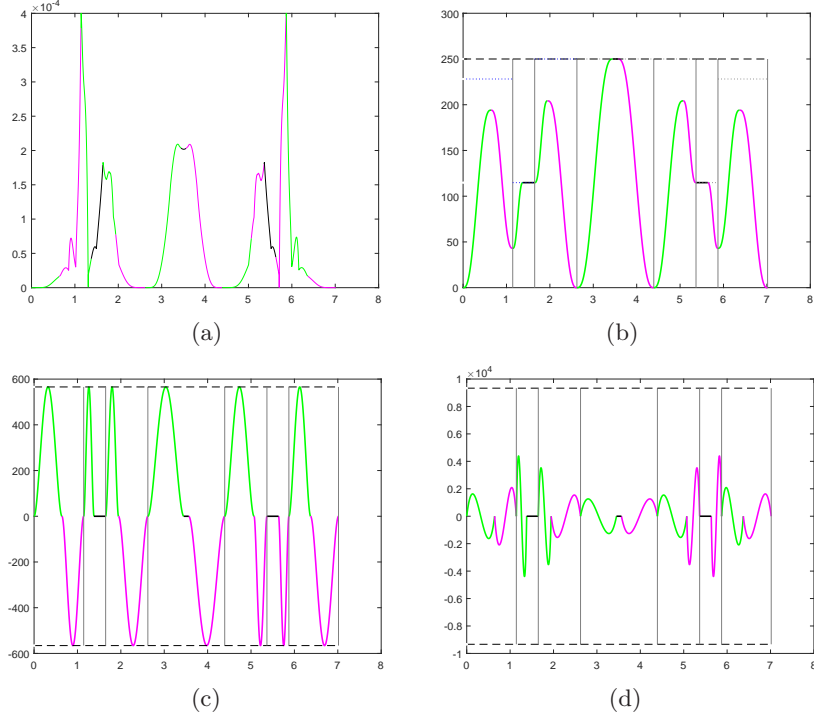


Figure 5: Feedrate scheduling results for the hat curve obtained using the R_0 configuration. The chord error (a) and the feedrate profile (b) are shown together with the first (c) and second (d) derivatives of the feedrate function.

6 Closure

We introduced a configurable C^2 feedrate scheduler whose more restrictive formulation ensures full kinematic control. Faster motions are obtained by using relaxed configurations, which often still preserve the desired motion control. The developed scheduler has been applied to C^2 PH spline curves to produce acceleration continuous motions. The selection of paths described in terms of PH spline curves is of great interest since it avoids numerical approximations of fundamental geometrical quantities and fosters the accuracy and robustness of the overall scheme.

The focus of this paper has been on the feedrate profile for planar (PH) paths. Although the spatial case requires a separate study, the procedure should readily extend to the three-dimensional setting. In order to obtain a jerk continuous motion on a PH spline path, a C^3 Hermite interpolation scheme with PH curves (of higher degree) should be investigated.

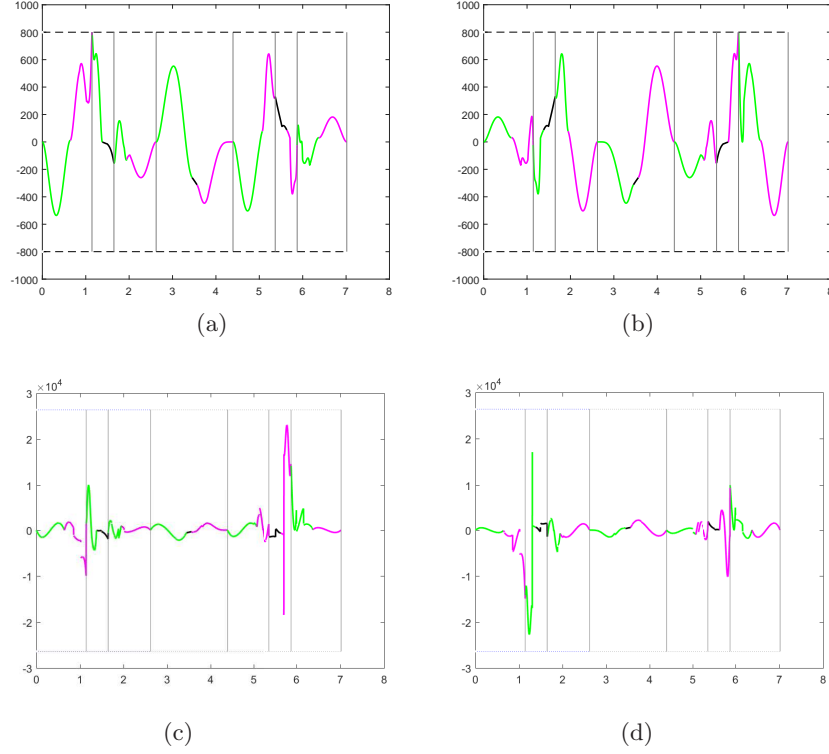


Figure 6: Cartesian components of acceleration (a-b) and jerk (c-d) obtained for the hat curve with the R_0 configuration.

Acknowledgments

This work has been partially supported by the MIUR “Futuro in Ricerca” project DREAMS (RBFR13FBI3), and by Istituto Nazionale di Alta Matematica (INdAM) through Finanziamenti Premiali SUNRISE.

References

- [1] M. Annoni, A. Bardine, S. Campanelli, P. Foglia, and C. A. Prete. A real-time configurable NURBS interpolator with bounded acceleration, jerk and chord error. *Comput. Aided Design*, 44:509–521, 2012.
- [2] M. Donatelli, C. Giannelli, D. Mugnaini, and A. Sestini. Curvature continuous path planning and path finding based on PH splines with tension. *Comput. Aided*

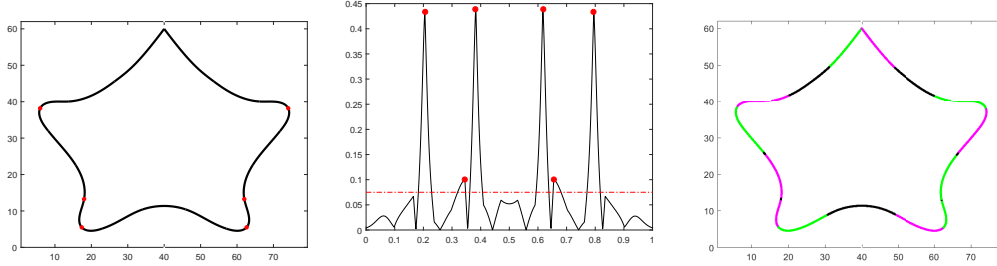


Figure 7: Starfish curve example: PH spline approximation (left), absolute curvature plot and critical curvature value (center, black and dashed red lines), and reference points (right). The critical points on the curve and the corresponding peaks on the curvature plot are also shown (red dots).

Design, 88:14–30, 2017.

- [3] R. T. Farouki. *Pythagorean-Hodograph curves: algebra and geometry inseparable*. Springer, Berlin, 2008.
- [4] R. T. Farouki, B. K. Kuspa, C. Manni, and A. Sestini. Efficient solution of the complex quadratic tridiagonal system for C^2 PH quintic splines. *Numer. Algor.*, 27:35–60, 2001.
- [5] R. T. Farouki, C. Manni, and A. Sestini. Shape-preserving interpolation by G^1 and G^2 PH quintic splines. *IMA J. Numer. Anal.*, 23:175–195, 2003.
- [6] R. T. Farouki and K. M. Nittler. Efficient high-speed cornering motions based on continuously-variable feedrates. I. Real-time interpolator algorithms. *Int. J. of Adv. Manuf. Technol.*, 87:3557–3568, 2016.
- [7] R. T. Farouki and S. Shah. Real-time CNC interpolators for Pythagorean-hodograph curves. *Comput. Aided Geom. Design*, 13:583–600, 1996.
- [8] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni. Trajectory planning in robotics. *Math. in Comp. Sciences*, 6:269–279, 2012.
- [9] C. Giannelli, D. Mugnaini, and A. Sestini. Path planning with obstacle avoidance by G^1 PH quintic splines. *Comput. Aided Design*, 75–76:47–60, 2016.
- [10] K. K. Kubota. Pythagorean triples in unique factorization domains. *Amer. Math. Monthly*, 79:503–505, 1972.

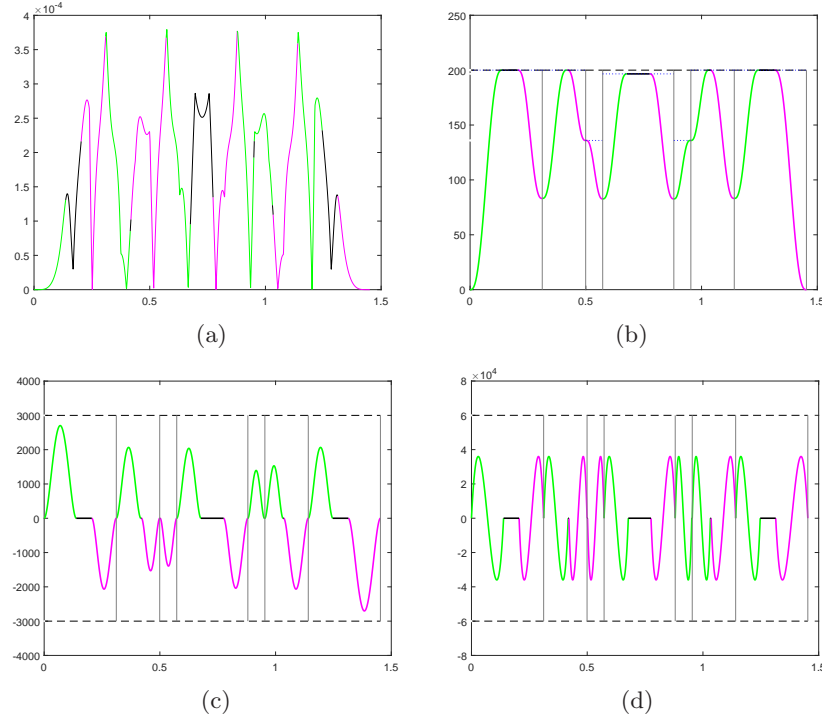


Figure 8: Feedrate scheduling results for the starfish curve obtained using the R_1 configuration. The chord error (a) and the feedrate profile (b) are shown together with the first (c) and second (d) derivatives of the feedrate function.

- [11] A. C. Lee, M. T. Lin, Y. R. Pan, and W. Y. Lin. The feedrate scheduling of NURBS interpolator for CNC machine tools. *Comput. Aided Design*, 43:612–628, 2011.
- [12] D. Mugnaini. Smooth path planning with Pythagorean-hodograph spline curves: geometric design and motion control. *Università degli Studi dell'Insubria*, PhD Thesis, 2017.
- [13] Z. Sír and B. Jüttler. Constructing acceleration continuous tool paths using Pythagorean Hodograph curves. *Mech. and Mach. Theory*, 40:1258–1272, 2005.
- [14] Z. Sír, E. Wings, and B. Jüttler. Rounding spatial G-code tool paths using Pythagorean hodograph curves. *Trans. of the ASME, Journal of Computing and Information Science in Engineering*, 7:186–191, 2007.

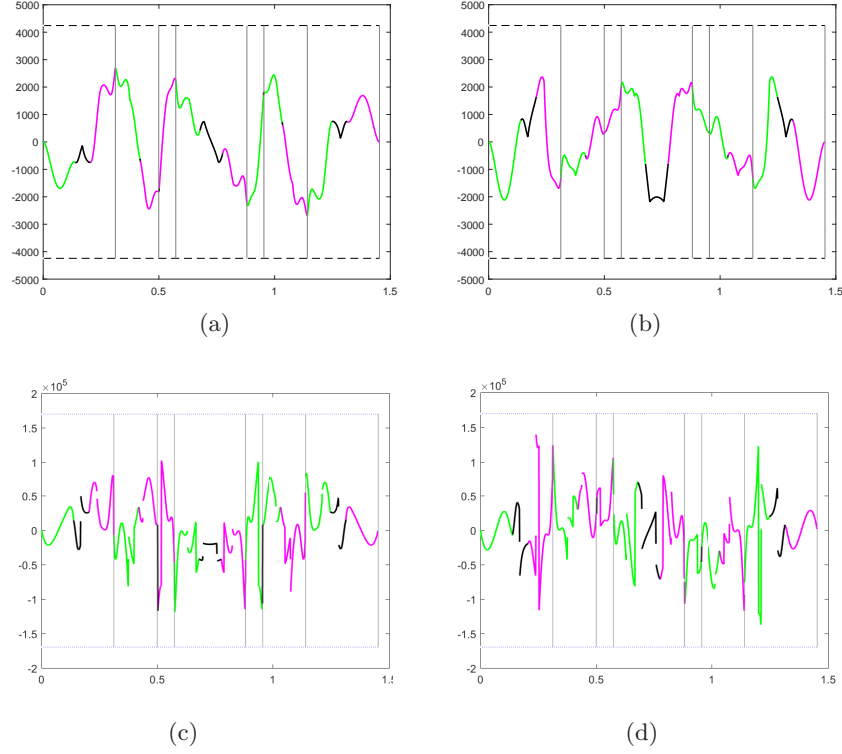


Figure 9: Cartesian components of acceleration (a-b) and jerk (c-d) obtained for the starfish curve with the R_1 configuration.

- [15] Y.-F. Tsai, R. T. Farouki, and B. Feldman. Performance analysis of CNC interpolators for time-dependent feedrates along PH curves. *Comput. Aided Geom. Design*, 18:245–265, 2001.
- [16] L. Xinhua, P. Junquan, S. Lei, and W. Zhongbin. A novel approach for NURBS interpolation through the integration of acc-jerk-continuous-based control method and look-ahead algorithm. *Int. J. of Adv. Manuf. Technol.*, 88:961–969, 2017.
- [17] S. S. Yeh and P. L. Hsu. Adaptive feedrate interpolation for parametric curves with a confined chord error. *Comput. Aided Design*, 34:229–237, 2002.

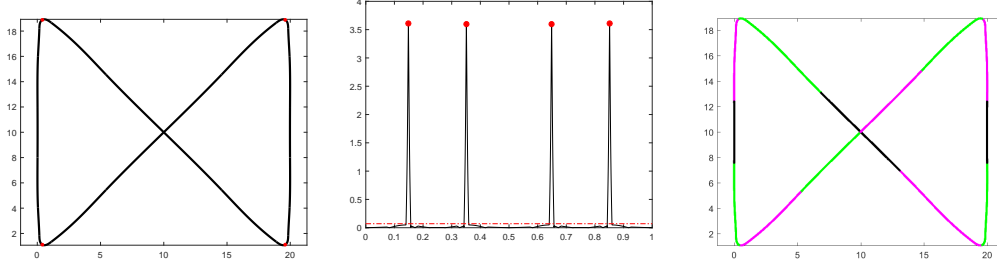


Figure 10: Butterfly curve example: PH spline approximation (left), absolute curvature plot and critical curvature value (center, black and dashed red lines), and reference points (right). The critical points on the curve and the corresponding peaks on the curvature plot are also shown (red dots).

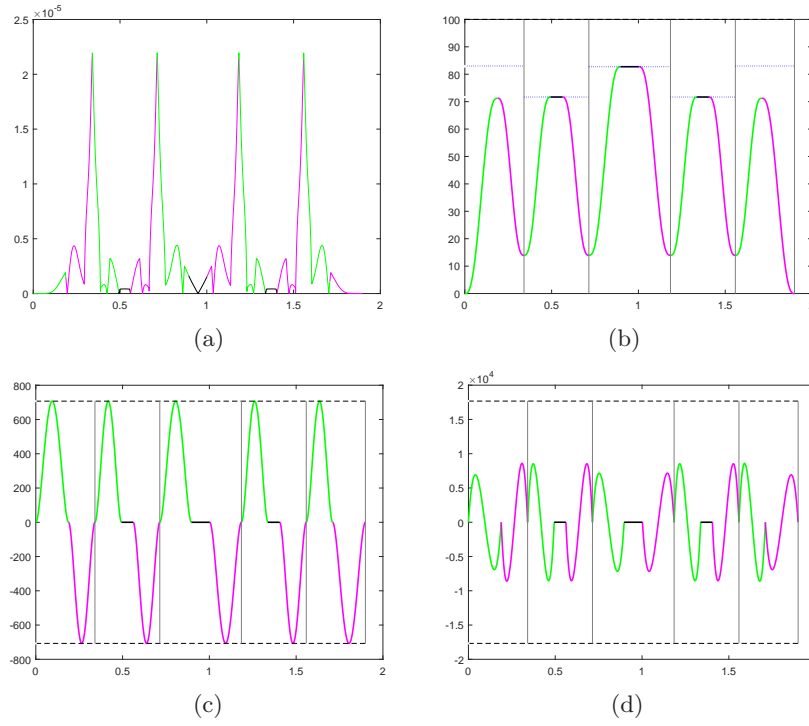


Figure 11: Feedrate scheduling results for the butterfly curve obtained using the R_1 configuration. The chord error (a) and the feedrate profile (b) are shown together with the first (c) and second (d) derivatives of the feedrate function.

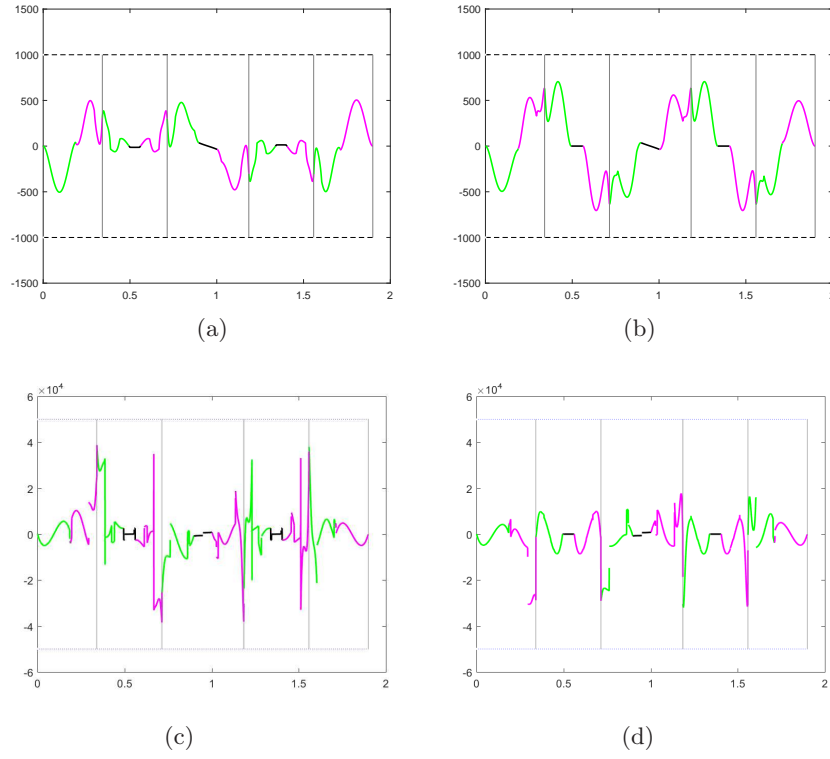


Figure 12: Cartesian components of acceleration (a-b) and jerk (c-d) obtained for the butterfly curve with the R_1 configuration.