B-Spline Based Filters for Multi-Point Trajectories Planning

Luigi Biagiotti and Claudio Melchiorri

***Abstract*— In this paper, the relation between B-splines and FIR (Finite Impulse Response) filters is demonstrated and exploited to design a digital filter for trajectory planning, combining the very simple structure and computational efficiency of FIR filters with the flexibility of splines. In particular, the trajectory generator consists of two main elements. The former is devoted to the solution of an optimization problem that, given a set of points to be interpolated (or approximated), provides the control points defining the spline. The latter, a cascade of moving average filters, gives the trajectory profile at each sampling time on the basis of such points. The proposed method has been applied to several robotic and industrial applications, and in this paper two case studies are reported as examples: an industrial robot performing a welding operation and a mobile robot moving in an environment with obstacles. With respect to these tasks, the main features of the trajectory generator are shown: the possibility of planning trajectories with high degree of smoothness (continuity of the derivatives), the possibility of easily changing the duration of the trajectory (and therefore the velocity, acceleration, jerk, etc. of the trajectory) maintaining the same geometric path, the possibility of locally modifying the pre-planned path.**

I. Introduction

Among many other applications, spline functions are extensively used in planning trajectories for robots because of their flexibility. As a matter of fact, the tasks demanded to robots (mobile robots, industrial robots, humanoid robots, etc.) often require position profiles with complex shapes which are usually defined by means of a number of via-points. These via-points are then interpolated or approximated with smooth functions to by optimized in order to comply with the constraints imposed by the specific robot application, i.e. kinematic constraints (such as limit values of velocity, acceleration, jerk, etc.) or dynamic constraints on the maximum torque available. In general, such interpolation tasks are performed by means of cubic splines since they assure the continuity of velocity and acceleration and prevent large oscillations of the trajectory that can result with high order polynomials, [1]. Therefore, cubic splines have been used to minimize the total traveling time of robot trajectories subject to constraints of velocity acceleration and jerk [2], or to globally minimize some quantities, such as acceleration [3] or jerk [4]. Some authors prefer the adoption of splines in the so-called B-form, i.e. B-splines, because they are much simpler from the computational point of view [5], [6] and because a local modification can be made quickly and easily without recomputing the entire trajectory [7]. Also in this case cubic B-splines are in general considered. Splines of order higher than three, able to guarantee the continuity of jerk and higher derivatives, have been proposed but they are based on the classical polynomial formulation [8]. Indeed, B-spline are very suitable to generate trajectories with continuous derivatives up to a generic order n, since the interpolation/approximation of a given set of points does not depends on the particular order of the B-spline [1]. On the other hand, despite the clear geometrical meaning of B-splines and their computational superiority with respect to the other (equivalent) spline formulations, their use in robotics is still limited (e.g. in robotics textbooks only polynomial splines are general considered, see [9], [10], [11] among many others) probably because their evaluation is based on a recursive procedure rather than on a closed form expression. Aim of this paper is to provide a more simple formulation of B-spline based trajectories, combining the advantages of these functions with the simplicity and the low computational complexity of FIR (Finite Impulse Response) filters.

L. Biagiotti is with the Department of Information Engineering, University of Modena and Reggio Emilia, 41100 Modena, Italy `luigi.biagiotti@unimore.it`

C. Melchiorri is with the Department of Electronics, Informatics and Systems, University of Bologna, 40136 Bologna,Italy `claudio.biagiotti@unibo.it`

II. B-splines and B-splines basis functions

Since the basic theory of B-splines is well known, we only give a brief summary of the concepts and notations. More details can be found e.g. in [1], [12], [13].

A B-spline of degree p is a parametric curve $s : [t_{min}, t_{max}] \to \mathbb{R}^d$ defined as linear combination of *B-spline basis functions* of degree p, $B_j^p(t)$:

$$\mathbf{s}(t) = \sum_{j=0}^{n} \boldsymbol{p}_j B_j^p(t), \qquad t_{min} \le t \le t_{max}. \tag{1}$$

The vectorial coefficients $\boldsymbol{p}_j$, $j = 0, \dots, m$, called *control points*, determine the shape of the curve and are computed by imposing approximation/interpolation conditions on a given set of data points. Let $\boldsymbol{t} = \{t_0, \dots, t_{m-1}\}$ be a vector of real numbers (called *knots*), with $t_j \le t_{j+1}$. The j-th *B-spline basis function* of degree p is defined, in a recursive manner, as

$$B_j^p(t) = \frac{t - t_j}{t_{j+p} - t_j} B_j^{p-1}(t) + \frac{t_{j+p+1} - t}{t_{j+p+1} - t_{j+1}} B_{j+1}^{p-1}(t) \tag{2}$$

with

$$B_j^0(t) = \begin{cases} 1, & \text{if} \quad t_j \le t < t_{j+1} \\ 0, & \text{otherwise.} \end{cases}$$

A particular case of B-splines is represented by *uniform* B-splines, that are defined for an equally-spaced distribution of the knots, i.e. $t_{j+1} - t_j = T, \; j = 0, \dots m-2$. In this case, the basis functions for a given degree p are consistent under shifts:

$$B^p_{j+1}(t) = B^p_j(t-T), \qquad j = 0, \ldots, m-2.$$

Therefore, for uniform B-splines it is possible to express the $(j+1)$-th basis function B^p_j in terms of the first basis function B^p_0, hereafter simply denoted by B^p:

$$B^p_j(t) = B^p(t - jT), \qquad j = 0, \ldots, m-1$$

and the B-spline can be rewritten as

$$\mathbf{s}_u(t) = \sum_{j=0}^{n} \boldsymbol{p}_j B^p(t - jT), \qquad 0 \le t \le (m-1)T. \quad (3)$$

Moreover, for uniform B-splines, the definition (2) of the basis function $B^p(t)$ of degree p is equivalent to

$$\begin{aligned} B^p(t) &= \frac{1}{T} B^{p-1} * B^0 \\ &= \underbrace{\frac{1}{T}B^0 * \frac{1}{T}B^0 * \ldots * \frac{1}{T}B^0}_{p \text{ times}} * B^0, \end{aligned} \quad (4)$$

where $*$ denotes the convolution product and

$$B^0(t) = \begin{cases} 1, & \text{if } \; 0 \le t < T \\ 0, & \text{otherwise.} \end{cases}$$

By Laplace transforming the general expression of the uniform B-spline (3) and substituting (4) one obtains

$$\boldsymbol{S}_u(s) = \sum_{j=0}^{n} \mathcal{L}\left\{ \boldsymbol{p}_j B^0 * \frac{1}{T}B^0 * \frac{1}{T}B^0 * \ldots * \frac{1}{T}B^0 \right\} e^{-jsT}.$$

Exploiting the linearity of the above expression and the fact that $\frac{1}{T}B^0$ is not a function of the index j, the B-spline expression becomes

$$\begin{aligned} \boldsymbol{S}_u(s) &= \sum_{j=0}^{n} \mathcal{L}\left\{ \boldsymbol{p}_j B^0 \right\} e^{-jsT} \cdot M(s) \cdot M(s) \cdot \ldots \cdot M(s) \\ &= \mathcal{L}\left\{ \sum_{j=0}^{n} \boldsymbol{p}_j B^0(t - jT) \right\} \cdot M(s) \cdot M(s) \cdot \ldots \cdot M(s). \end{aligned}$$

where $M(s) = \mathcal{L}\left\{ \frac{1}{T} B^0(t) \right\} = \frac{1}{T} \frac{1 - e^{-sT}}{s}$ performs the mean of the input function over an interval of duration T. This expression suggests that a uniform B-spline can be evaluated by feeding the cascade of p filters $M(s)$ with a piecewise constant function

$$\boldsymbol{p}(t) = \sum_{j=0}^{n} \boldsymbol{p}_j B^0(t - jT)$$

that in the generic interval $jT \le t < (j+1)T$ assumes the constant value $\boldsymbol{p}_j$. Note that, in multidimensional B-splines, the control points $\boldsymbol{p}_j$ are multi-dimensional and the function $\boldsymbol{p}(t)$ is multidimensional as well. In this case, it is necessary to consider the different components and filter each of them with a separate chain of mean filters.

III. Discrete B-splines

A unified transformation to convert analytic B-splines in the discrete domain does not exist. In particular, with reference to cardinal B-splines (uniform B-splines defined over the integers) which are mainly used for signal analysis, interpolation and image processing, it is possible to find in the literature different techniques to obtains discrete B-splines. In general, they are defined by directly sampling analytic B-splines with Z-transform, bilinear transform, etc. [14], [15], [16], [17], [18]. For this purpose, the first step consists in discretizing the basis functions. By exploiting the relationship

$$\begin{aligned} B^p(t) &= \frac{1}{T} B^{p-1}(t) * B^0(t) \\ &= \frac{1}{T} \int_{-\infty}^{\infty} B^{p-1}(t - \tau) \; B^0(\tau) d\tau \end{aligned}$$

the value of the B-spline basis function at the discrete time instants $t = kT_s$ (where T_s is the sampling time) results

$$\begin{aligned} B^p_k &= \frac{1}{T} \int_{-\infty}^{\infty} B^{p-1}(kT_s - \tau) \; B^0(\tau) d\tau \\ &\approx \frac{1}{T} \sum_{n=-\infty}^{n=\infty} B^{p-1}(kT_s - nT_s) \; B^0(nT_s) T_s \\ &= \frac{1}{N} \sum_{n=-\infty}^{n=\infty} B^{p-1}_{k-n} \; B^0_n \; = \; \frac{1}{N} B^{p-1}_k * B^0_k \end{aligned}$$

where $B^p_k = B^p(kT_s)$, $N = T/T_s$ is the number of samples in each knot span, and $*$ is the discrete convolution product. Finally, the discrete basis function B^p_k of degree p, can be written as

$$B^p_k = \underbrace{\frac{1}{N}B^0_k * \frac{1}{N}B^0_k * \ldots * \frac{1}{N}B^0_k}_{p \text{ times}} * B^0_k \quad (5)$$

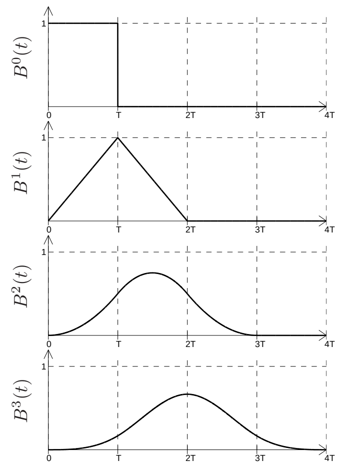


Fig. 1. B-spline basis functions $B^p(t)$ obtained for different values of the degree p.

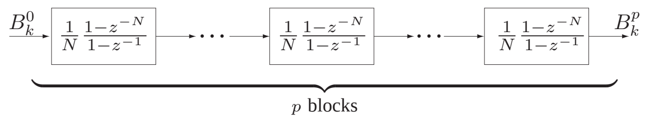

B_k^0
$\frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}}$
$\frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}}$
$\frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}}$
B_k^p
p blocks

Fig. 2. System composed by p filters for the computation of the discrete B-spline basis function B_k^p of degree p.

with

$$B_k^0 = \begin{cases} 1, & \text{if } \ 0 \le k < N-1 \\ 0, & \text{otherwise.} \end{cases}$$

Note that (5) is only an approximation of the analytic B-spline basis function and does not have the same values at discrete points. Nevertheless, it is possible to prove that the staircase curve obtained in this way tend to the B-spline basis function in the sense of root mean square (RMS) as T_s goes to zero [17]. The performed approximation leads to a very simple definition of discrete basis functions and discrete B-splines. As a matter of fact, the discrete function (5) can be seen as a cascade of p mean filters (moving average filters), whose transfer functions are

$$\begin{aligned} M(z) &= \frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}} \qquad (6) \\ &= \frac{1}{N}\left(1+z^{-1}+z^{-2}+\ldots+z^{-(N-1)}\right) \end{aligned}$$

with the input B_k^0, see Fig. 2. Like analytic splines, discrete uniform B-splines are defined as a linear combination of the discrete basis function properly time-shifted:

$$\mathbf{s}_k = \sum_{j=0}^{n} \boldsymbol{p}_j B_{k-jN}^p \qquad (7)$$

By applying the Z-transform to (7) one obtains

$$\mathbf{S}(z) = \sum_{j=0}^{n} \mathcal{Z}\left\{\boldsymbol{p}_j \frac{1}{N}B_k^0 * \frac{1}{N}B_k^0 * \ldots * \frac{1}{N}B_k^0 * B_k^0\right\} z^{-jN} \qquad (8)$$

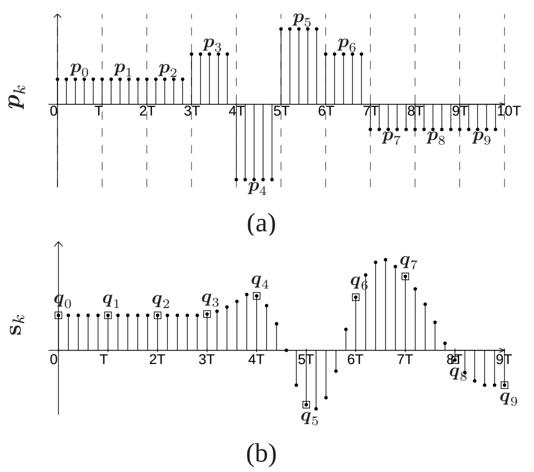


Fig. 3. Samples of the piecewise constant function $\boldsymbol{p}_k$ (a) generating the spline profile $\mathbf{s}_k$ that interpolates the given points $\boldsymbol{q}_j$ (b).

and then, because of the linearity of (8) and the fact that $\frac{1}{N}B^0$ is not a function of the index j,

$$\begin{aligned} \mathbf{S}(z) &= \sum_{j=0}^{n} \mathcal{Z}\left\{\boldsymbol{p}_j B_k^0\right\} z^{-jN} \cdot M(z) \cdot M(z) \cdot \ldots \cdot M(z) \\ &= \mathcal{Z}\left\{\sum_{j=0}^{n} \boldsymbol{p}_j B_{k-jN}^0\right\} \cdot M(z) \cdot M(z) \cdot \ldots \cdot M(z). \end{aligned}$$

Therefore, the discrete uniform B-spline of degree p is defined as the output of a chain of p mean filters feeded with the piecewise constant function

$$\boldsymbol{p}_k = \sum_{j=0}^{n} \boldsymbol{p}_j B_{k-jN}^0 \qquad (9)$$

shown in Fig. 3(a).

IV. Choice of the Control Points

The proposed trajectory planner, composed by a generator of sequences of constant values $\boldsymbol{p}_j$ and by a cascade of p mean filters, enjoys the same properties of analytic uniform B-splines. Therefore, in order to find the control points which define the piecewise constant function $\boldsymbol{p}_k$, one can exploit classical techniques derived by B-spline interpolation/approximation methods.

For example, if one considers the interpolation of a set of $l+1$ points $\{\boldsymbol{q}_0, \boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_{l-1}, \boldsymbol{q}_l\}$ it is necessary to impose the conditions

$$\mathbf{s}(t_i) = \boldsymbol{q}_i, \qquad i = 0, \ldots, l \qquad (10)$$

where t_i is the time instant at which the spline $\mathbf{s}(t)$ crosses the given point $\boldsymbol{q}_i$.

The first step consists in selecting the degree p of the spline according to the desired degree of smoothness. Strictly related to p is the choice of time instants t_i:

- if p is odd, the t_i are assumed coincident with the knots, $t_i = iT$;
- if p is even, the time instants t_i should be selected in the midpoint of each knot span, $t_i = \frac{2i+1}{2}T$.

Once the interpolation time instants t_i have been chosen it is possible to make the system of equations (10) explicit with the substitution of the values of basis functions at t_i in the spline definition (1). For uniform B-splines it is possible to find a closed form expression which provides the value of the basis function B^p at a generic time instant t [16]:

$$B^p(t) = \frac{1}{p!}\sum_{k=0}^{p+1}(-1)^k \binom{p+1}{k} \left(\frac{t}{T} - k\right)_+^p$$

where ! denote the factorial function, $\binom{a}{b} = \frac{a!}{b!(a-b)!}$ and $(x)_+^p$ is the truncated power function ($(x)_+^p = x^p$ if $x \ge 0$, $(x)_+^p = 0$ otherwise). The values of B^p for p odd and p even, computed at points $t_i = iT$ and $t_i = \frac{2i+1}{2}T$ respectively, are reported in Tab. I. Note that, because of the choice of the interpolation time instants, the values of B^p do not depend on T, but only on the index i and on the degree p.

In order to obtain a system of equations well conditioned from a mathematical point of view it is necessary to consider symmetrical B-splines $\mathbf{s}_{\mathbf{s}}(\mathbf{t})$, i.e. uniform B-splines whose basis function $\beta^p(t)$ is symmetric with respect the origin. The function $\beta^p(t)$ can be deduced from $B^p(t)$ with a simple time shift, $\beta^p(t) = B^p\left(t + \frac{p+1}{2}T\right)$, and as a consequence symmetrical B-splines are related to standard uniform B-splines by

$$\begin{aligned}\mathbf{s}_s(t) &= \sum_{j=0}^{n} \boldsymbol{p}_j \beta^p(t - jT)\\ &= \sum_{j=0}^{n} \boldsymbol{p}_j B^p(t + \tfrac{p+1}{2}T - jT) = \mathbf{s}_u(t + \tfrac{p+1}{2}T),\end{aligned}$$

that is, given the control points, uniform B-splines are equal to symmetrical B-splines delayed by $\frac{p+1}{2}T$. Obviously, the theory of Sec. II and Sec. III could be based on symmetrical B-splines but this would imply the presence of a temporal anticipation leading to noncausal filters for the evaluation of B-splines.

For each point to be interpolated, with the only exception of the first and last points, the equation (10) becomes

$$\mathbf{s}_s(t_i) = \sum_{j=0}^{n} \boldsymbol{p}_j B^p(t_i - jT + \tfrac{p+1}{2}T) = \boldsymbol{q}_i \qquad (11)$$

where the unknowns are the control point $\boldsymbol{p}_j$. The interpolation of the first and last points, with zero velocity and acceleration, is achieved by exploiting the characteristics of the dynamic system of Fig. 2 used to generate the spline. Since all the filters have unitary static gain, the output of the filters cascade will reach and maintain the desired value $\boldsymbol{q}_0$ or $\boldsymbol{q}_l$ iff the same value is applied to the input pT seconds before. In other words, in order to smoothly start from $\boldsymbol{q}_0$ and end to $\boldsymbol{q}_l$, the first/last p control points must be equal to $\boldsymbol{q}_0/\boldsymbol{q}_l$. The $l-1$ internal control points are then computed by solving the system of equations obtained by stacking (11)

	0	T	2T	3T	4T	5T	6T	7T	8T
$p=1$	0	1	0						
$p=3$	0	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$	0				
$p=5$	0	$\frac{1}{120}$	$\frac{26}{120}$	$\frac{66}{120}$	$\frac{26}{120}$	$\frac{1}{120}$	0		
$p=7$	0	$\frac{1}{5040}$	$\frac{120}{5040}$	$\frac{1191}{5040}$	$\frac{2416}{5040}$	$\frac{1191}{5040}$	$\frac{120}{5040}$	$\frac{1}{5040}$	0

(a)

	$\frac{1}{2}T$	$\frac{1}{2}T$	$\frac{3}{2}T$	$\frac{5}{2}T$	$\frac{7}{2}T$	$\frac{9}{2}T$	$\frac{11}{2}T$
$p=2$	$\frac{1}{8}$	$\frac{6}{8}$	$\frac{1}{8}$	0			
$p=4$	$\frac{1}{384}$	$\frac{76}{384}$	$\frac{230}{384}$	$\frac{76}{384}$	$\frac{1}{384}$	0	
$p=6$	$\frac{1}{46080}$	$\frac{722}{46080}$	$\frac{10543}{46080}$	$\frac{23548}{46080}$	$\frac{10543}{46080}$	$\frac{722}{46080}$	$\frac{1}{46080}$

(b)

TABLE I
B-SPLINE BASIS FUNCTION $B^p(t)$ FOR p ODD AT POINTS $t_i = iT$ (a), AND FOR p EVEN AT POINTS $t_i = \frac{2i+1}{2}T$ (b).

for $i = 1, \dots, l-1$ and the piecewise constant function (9) can be finally built.

A. Cubic B-splines

The adoption of the B-splines of degree $p = 3$, i.e. cubic B-splines, guarantees the continuity of velocity and acceleration. By substituting the values of Tab. I, the general equation (11) becomes

$$\mathbf{s}_s(iT) = \frac{1}{6}\boldsymbol{p}_{i-1} + \frac{4}{6}\boldsymbol{p}_i + \frac{1}{6}\boldsymbol{p}_{i+1} = \boldsymbol{q}_i, \quad i = 1, \dots, l-1 \qquad (12)$$

with the additional constraints $\boldsymbol{p}_0 = \boldsymbol{q}_0$ and $\boldsymbol{p}_l = \boldsymbol{q}_l$. After simple algebraic manipulations, (12) can be written in a matrix form as

$$\begin{bmatrix} 4 & 1 & 0 & & & & \cdots & 0\\ 1 & 4 & 1 & 0 & & & \cdots & 0\\ 0 & 1 & 4 & 1 & 0 & & \cdots & 0\\ \vdots & & & & \ddots & & & \vdots\\ 0 & \cdots & & 0 & 1 & 4 & 1 & 0\\ 0 & \cdots & & & 0 & 1 & 4 & 1\\ 0 & \cdots & & & & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1\\ \boldsymbol{p}_2\\ \boldsymbol{p}_3\\ \vdots\\ \boldsymbol{p}_{l-3}\\ \boldsymbol{p}_{l-2}\\ \boldsymbol{p}_{l-1} \end{bmatrix} = \begin{bmatrix} 6\boldsymbol{q}_1 - \boldsymbol{q}_0\\ 6\boldsymbol{q}_2\\ 6\boldsymbol{q}_3\\ \vdots\\ 6\boldsymbol{q}_{l-3}\\ 6\boldsymbol{q}_{l-2}\\ 6\boldsymbol{q}_{l-1} - \boldsymbol{q}_l \end{bmatrix} \qquad (13)$$

Due to the tridiagonal structure of the system matrix, the solution of (13) can be easily found. More generally, by adopting B-splines of generic degree p, the systems to be solved for obtaining the control points will be characterized by banded matrices, whose inversion can be carried out in a very efficient way.

Figure 4 shows the cubic B-spline trajectory obtained by interpolating the points

$$\{\boldsymbol{q}_j\} = \{5, 12, 3, 45, 23, 4, -3, 5, -3, 10, 10, 16, 19, 4, 23\}$$

with period $T = 1$s. For the sake of simplicity a one-dimensional case has been taken into account but the same considerations are valid in the multi-dimensional case as shown in the later Sec. VI. Note that the first and the last tracts are of length $2T$ (and in the case of B-splines of generic degree p of length $\frac{p+1}{2}T$). This is unavoidable but, on the other hand it allows the trajectory to smoothly start/end, i.e. with all the derivatives up to the order $p-1$ that are null

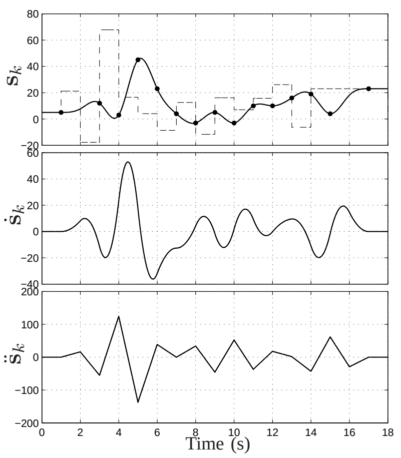


Fig. 4. One-dimensional cubic B-spline $\mathbf{s}_k$ interpolating a set of given points: profiles of position (superimposed to the piecewise constant function $\boldsymbol{p}_k$), velocity and acceleration.

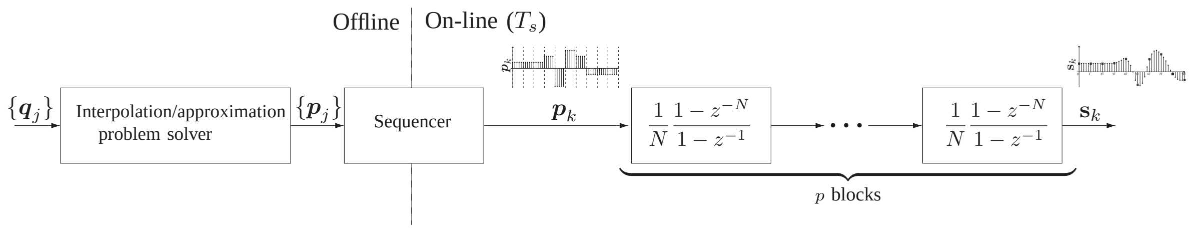


Fig. 5. Overall structure of the filter for B-spline trajectories planning.

at the start- and end- points. In the case of the cubic B-spline this means that initial and final velocities and accelerations are zero. By comparing the profiles of the function $\boldsymbol{p}_k$ and of the spline $\mathbf{s}_k$ it is clear that the latter roughly reproduces the shape of the former with a "delay" of $\frac{p+1}{2}T$ seconds, see Fig. 4. Moreover, when the last control point is applied, the output of the trajectory planner reaches such a value after pT seconds.

V. Structure of the Filter for Trajectory Planning

In Fig. 5 the structure of the trajectory generator is illustrated. It is composed by two main elements:

1) an algorithm (described previously) that transforms the desired points $\boldsymbol{q}_j$ in the set of control points $\boldsymbol{p}_j$;
2) a cascade of p moving average filters, where p is the desired degree of the B-spline (the resulting trajectory will be C^{p-1}).

Between them, the "Sequencer" produces the piecewise constant function $\boldsymbol{p}_k$ shown in Fig. 3(a) arranging in a line the control points $\boldsymbol{p}_j$, each one for a duration of T seconds, where T is the time distance between consecutive points. While the algorithm for the control points computation is performed offline, the system composed by the sequencer and by the filters runs on-line with a sampling time T_s, that in general is the same of the digital controller of the overall robotic system. Since T_s is fixed, the number N of samples considered in each FIR filter only depends on T, being $N = \text{round}(\frac{T}{T_s})$. Therefore, by changing N it is possible to directly modify T.

In order to smoothly starts from the first desired point $\boldsymbol{q}_0$ it is necessary that internal states of all the FIR filters are set to $\boldsymbol{q}_0$, while a smooth end at $\boldsymbol{q}_l$ is guaranteed by the application to the filters cascade input of the control point $\boldsymbol{q}_l$ for at least p intervals of duration T.

In robotic applications, the points $\boldsymbol{q}_j$ are generally defined in a d-dimensional space ($d = 2, 3$). In this case, the trajectory generator must be simply replicated d times, one for each component, and the desired B-spline is obtained if the related sequences are synchronized.

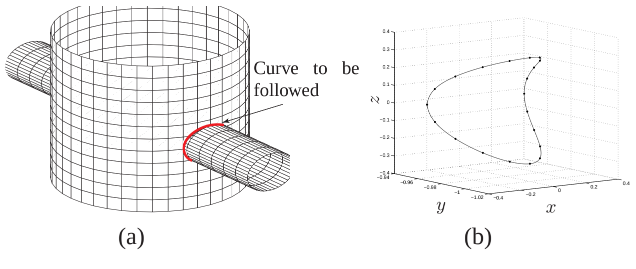


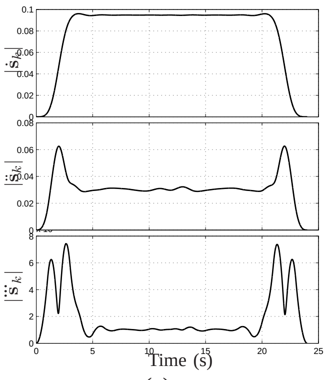
Fig. 6. Welding of two cylindrical objects (a) and related geometric path (b).

VI. Numerical examples

Two case studies are now reported as examples of the flexibility and simplicity offered by this technique for trajectory planning.

A. 3D welding

Consider the case of an industrial robot manipulator welding the two cylinders in Fig. 6(a). The path that the robot tool must track is reported in Fig. 6(b). Many CAD/CAM systems directly provide the control points $\boldsymbol{p}_j$ of the B-spline to be followed. Differently, it is possible to "sample" the analytic curve if given or, more generally, consider a certain number of points distributed along the geometric path to be followed. In the case of uniform B-splines, the distribution of the path points is crucial for imposing a desired feed-rate of the tool. As a matter of fact, it is not possible to freely chose the duration of each spline segment (being T equal for all the tracts), but on the contrary it is necessary to select the distance between points $\boldsymbol{q}_j$ with the purpose of "shaping" the velocity, the acceleration, etc. of the trajectory. For instance, in a welding application the velocity should be constant along the entire path. For this reason, points $\boldsymbol{q}_j$ have been selected at a distance which is approximatively constant (the mean value of the distance between consecutive points is $\delta = 0.0944$). From these via-points the control points $\boldsymbol{p}_j$ have been obtained as reported in Sec. IV. In order to prevent oscillation and reduce tracking errors a quintic B-spline has been considered ($p = 5$). In this way, velocity, acceleration, jerk and even snap (i.e. the derivative of jerk) are continuous, as shown in Fig. 7. The trajectory is evaluated on-line by a chain of five FIR filters with a sampling time $T_s = 0.001$s. Each filter calculates the average of the last $N = 1000$ samples, therefore the time distance between the points $\boldsymbol{q}_j$ is $T = NT_s = 1$s. As a consequence the speed, which is almost constant, is $|\dot{\mathbf{s}}_k| \approx \frac{\delta}{T} = 0.094$, see Fig. 7(a). By changing N it is possible to modify the duration of each tract and of the overall trajectory. Accordingly the derivatives of $\mathbf{s_k}$


(a)

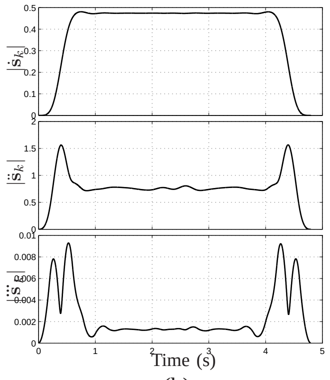

(b)

Fig. 7. Profiles of the norm of velocity, acceleration and jerk related to the B-spline $\mathbf{s}_k$ defining the curve of Fig. 6(b) for different values of T: $T = 1$s (a) and $T = 0.2$s (b).

(velocity, acceleration, jerk, etc.) are modified as follows

$$\mathbf{s}'^{(i)}_k = \alpha^i \, \mathbf{s}^{(i)}_k$$

where $\alpha = \frac{N}{N'}$, being N' the new number of samples considered in each FIR filter. In Fig. 7(b) the velocity, acceleration and jerk of the B-spline computed with $N = 200$ (thus $T = 0.2$s) is shown. Note that the velocity is scaled by 5 times, the acceleration by 25, the jerk by 125, while the shape of these profiles remains unchanged. Obviously also the geometric path defined by the trajectory generator does not change.

B. Mobile robots

Another interesting application of the proposed planner concerns the navigation of mobile robots in an environment with obstacles, as shown in Fig. 8(a) where a proper set of points (black spots) are interpolated by means of a second degree spline. The choice $p = 2$ guarantees only the continuity of the velocity, but on the other hand it reduces the delay between the application of the input $\boldsymbol{p}_k$ to the chain of two filters and the related output constituting the trajectory. In this way it is possible to change the trajectory pre-computed by simply changing the control points during the realtime generation and tracking of the trajectory itself.As a matter of fact, a B-spline trajectory has the property that a local modification can be made quickly and easily without recomputing the entire trajectory and leaves the remaining trajectory unchanged. For instance, in Fig. 8(b) in order to avoid the first obstacle, whose position is changed, some control points are moved (the small spots in the figure are the control points $\boldsymbol{p}_j$ and not the via-points $\boldsymbol{q}_j$).

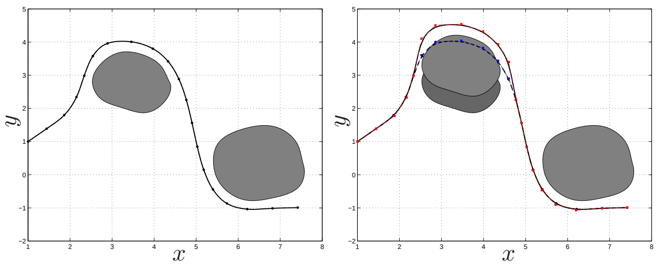


Fig. 8. Geometric path in a 2D environment with obstacles defined by means of a B-spline interpolating the given via-points (a) and local modification obtained by changing the function $\boldsymbol{p}_k$.

VII. Conclusions

In this paper a digital filter for trajectory planning is proposed. This trajectory generator is based on B-spline functions and therefore shares the same characteristics: multi-point trajectories composed by properly joined polynomial segments, degree of smoothness that can be freely selected, efficient methods for interpolation/approximation of given points, possibility of making local changes on the trajectory without recomputing it. On the other hand, the proposed structure composed by a chain of FIR filters allows a great simplification of the procedures for B-spline evaluation and on-line trajectory generation, making the proposed planner suitable for a number of robotic applications.

References

[1] L. Biagiotti and C. Melchiorri. *Trajectory Planning for Automatic Machines and Robots*. Springer, Heidelberg, Germany, first edition, 2008.

[2] C.-S. Lin, P.-R. Chang, and J.Y.S. Luh. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Transaction on Automatic Control*, 28(12):1066–1074, 1983.

[3] B. Cao, G.I. Dodds, and G.W. Irwin. Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots. *Proceedings of IEE Conference on Control Theory and Applications*, 144:467–475, 1997.

[4] A. Piazzi and A. Visioli. Global minimum-jerk trajectory planning of robot manipulator. *IEEE Transaction on Industrial Electronics*, 47(1):140–149, 2000.

[5] J. E. Bobrow. Optimal robot path planning using the minimum-time criterion. *IEEE Journal of Robotics and Automation*, 4 (5):443–450, 1988.

[6] C-H Wang and J-G Horng. Constrained minimum-time path planning for robot manipulators via virtual knots of the cubic b-spline functions. *IEEE transactions on automatic control*, 35 (5):573–577, 1990.

[7] S. E. Thompson and R. V. Patel. Formulation of joint trajectories for industrial robots using B-splines. *IEEE Transactions on Industrial Electronics*, 34 (2):192–199, 1987.

[8] A. Piazzi, C. G. Lo Bianco, and M. Romano. η^3-splines for the smooth path generation of wheeled mobile robots. *IEEE Transactions on Robotics*, 23 (5):1089–1095, 2007.

[9] M. W. Spong, S Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, 1997.

[10] J. Angeles. *Fundamentals of Robotic Mechanical Systems*. Springer, 2007.

[11] B. Siciliano, L. Sciavicco, L.Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 2009.

[12] L. Piegl and W. Tiller. *The Nurbs Book*. Springer-Veralg, second edition, 1997.

[13] C. de Boor. *A Practical Guide to Splines*. Springer, 2001.

[14] P.V. Sankar and L.A. Ferrari. Simple algorithms and architectures for B-spline interpolation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 (3):277 – 285, 1991.

[15] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part I and Part II. *IEEE Transaction on Signal Processing*, 41(2):821–848, 1993.

[16] H. Olkkonen. Discrete binomial splines. *Graphical Models and Image Processing*, 57 (2):101–106, 1995.

[17] K. Ichige and M. Kamada. An approximation for discrete B-splines in time domain. *IEEE Signal Process. letters*, 4:82–84, 1997.

[18] S. Samadi, M.O. Ahmad, and M.N.S. Swamy. Characterization of B-spline digital filters. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51 (4):808– 816, 2004.