A METHOD TO DETERMINE THE COEFFICIENTS IN B-SPLINE INTERPOLATION

Liliana Mâțiu-Iovan[1] Flaviu Mihai Frigură-Iliasa[2] Doru Vătău[2]
e-mail: liliana.matiu@etc.upt.ro e-mail: flaviu.frigura@et.upt.ro e-mail: doru.vatau@et.upt.ro
[1]*Faculty of Electronics and Telecommunications* [2]*Faculty of Electrical Engineering*
"Politehnica" University of Timisoara, Bv. Parvan No.2, 300223, Timisoara, Romania

Key words: Signal processing, interpolation, B-spline

ABSTRACT

In this paper it is presented another method to determine the coefficients in B-spline interpolation. The problem is resolved in the context of generalized spline interpolation. To determine the coefficients from the input samples is first step for performing the interpolation. The operation is done by using also the values of the second derivative for the input signal. The coefficients are used to perform the signal reconstruction and interpolation by a factor *m*=2. The algorithms are tested on several known signals. The practical results are discussed at the end.

I. INTRODUCTION

A great part of today industrial processes tends to be computer programmed and digital controlled. To perform numerical processing it is needed an acquisition and analog to numeric conversion. Some of the procedures require operations like approximation, reconstruction, or interpolation. In many cases the given data are uniformly spaced. The presented methods and algorithms can be applied on samples of this type.

The spline functions are successfully used in problems of function approximation, signal and image processing. From 1970 these functions were intensely developed on numerous directions. Functions like polynomial cubic spline or B-spline are combined with modern digital techniques in a new manner to provide faster algorithms and better results [1, 2].

Here we searched other methods to use spline functions in signal reconstruction and interpolation. We search the possibility to implement these methods on systems with digital signal processors. The new algorithms might be utilized in on-line applications and real-time signal processing.

A known algorithm from the literature was studied previously [3]. The Unser's algorithm is applied in image processing [2]. This was utilized in signal interpolation and some practical results were obtained [3]. The conclusions of the study were utilized to develop new methods for determining the B-spline coefficients [3, 4].

Here we use the values for the second derivative of the input function in the knots to calculate the coefficients. The problem is to determine these values from the known input samples. The methods are presented in this paper. We exemplify also some practical results.

II. METHODS OF INTERPOLATION

The input samples represent a set of discrete data $y=\{y(k)\}$, $k = 0, N\text{-}1$, regularly sampled. The interpolation problem is to determine a function $f(x)$ that pass through all the input data. For any $k \in \{0, N\text{-}1\}$ the values of the function are $f(k) = y(k)$.

The polynomial functions are frequently used for interpolation. Lagrange and Bernstein polynomials are well known methods used for resolving the problem. The polynomial spline functions were applied in the process of interpolation due to their properties. These are polynomial functions of degree n on adjacent intervals connected in the knots. The spline functions are continuous. Also, the function derivatives up to n-1 order are continuous [5].

To determine the interpolation function there were utilized systems of equations and matrix computational methods 5. This is the traditional approach of the problem. In these procedures are necessary many numerical calculations and is hard to make algorithms ready to implement on numerical systems.

The concept of generalized interpolation represents a modern approach. This provides a larger view on the problem. This method needs two steps to perform the interpolation.

First step is to determine some coefficients $c(k)$ from the input data $y(k)$. The interpolated values are obtained from these coefficients and not directly from the input values.

In the generalized formulation the interpolated value $f(x)$ it is obtained by convolving the coefficients string with a basis function [2]:

$$f(x) = \sum_{k \in Z} c(k)\varphi(x-k) . \quad (1)$$

The advantage is to allow the use of a larger class of potential basis functions. In practice is necessary to determine the interpolated values of the function and not the exact structure of the interpolation function. Also the actual direction is to perform a good approximation with minimum effort instead to make an exact interpolation with greater computational costs.

It can be said that the traditional interpolation is a particular case for the generalized interpolation: the coefficients are equal to the input samples.

The process of generalized interpolation can be performed using digital filtering techniques. The solution is proposed in image processing in 1978 and developed later on multiple studies [1, 2].

Michael Unser develops the idea and elaborates an algorithm that uses digital filters for interpolation [2]. A digital filter is applied to the input samples to obtain the coefficients. Another filter is used to find the interpolated values from these coefficients. The filters are determined using the cubic B-spline function:

$$\beta^3(x) = \begin{cases} 2/3 - |x|^2 + |x|^3/2, & 0 \le |x| < 1 \\ (2-|x|)^3/6, & 1 \le |x| < 2 \\ 0, & 2 \le |x| \end{cases} . \quad (2)$$

For β^3 (x) it is defined the discrete B-spline function $b_1^3(k)$:

$$(b_1^3)^{-1}(k) \quad \leftrightarrow \quad [B_1^3(z)]^{-1} = \frac{6}{z+4+z^{-1}} . \quad (3)$$

The spline coefficients $c(k)$ are obtained by applying the direct B-spline filter (3) to the input signal. The operation is called “direct B-spline transform”. The second step is called “indirect B-spline transform” and use the indirect B-spline filter B_m^n. For m representing the factor of interpolation, the function $f_n(x/m)$, denoted $f_m^n(x)$ will be:

$$f_m^n(x) = \sum_{k \in Z} c(k) b_m^n(x-km) . \quad (4)$$

The coefficients are calculated by a recursive algorithm that implements the direct B-spline filter. This filter is divided in 2 filters: first a causal and the second anti-causal [2]. The method requires some initial conditions [2] and introduces some side errors for the coefficients [3]. Those errors are transmitted in the interpolated signal and could have great importance especially if the input signal contains a small number of samples.

We searched another ways to calculate the B-spline coefficients trying to reduce those errors. In a previous work [4] it was developed and implemented an algorithm that calculates the coefficients using the relation between a function that approximates the input signal, the function derivatives and the B-spline coefficients. There are presented several methods to calculate the values for the function derivatives. Because these values are imposed the process became Hermite interpolation. Every coefficient is determined using one of the previous coefficients and the first derivative of the function. It was studied the possibility to use also the second derivative for the function $f(x)$.

III. A METHOD BASED ON THE SECOND DERIVATIVE

The function values and the values for the first and second derivatives of the input function were used in the traditional spline interpolation [7]. We took the idea and applied here to determine the coefficients for spline interpolation.

Consider $f(x)$ the cubic spline function that approximates the input function. It is passing trough all the input values, in the knots $f(k) = y(k)$, $k = 0, N\text{-}1$.

The relation involving the function and the coefficients $c(k)$ can be write:

$$6f(k) = 4\,c(k) + c(k\text{-}1) + c(k\text{+}1). \quad (5)$$

We want to find a relation between the coefficients, the samples values and the derivatives for the input function. The cubic B-spline derived function of second order it can be determined like:

$$\beta^{3}{''}(x) = \begin{cases} -2+3x, & 0 \le |x| < 1 \\ 2-x, & 1 \le |x| < 2 \\ 0, & 2 \le |x| \end{cases} . \quad (6)$$

The relation between the coefficients and the function’s second derivative is:

$$f''(k) = -2\,c\,(k) + c\,(k\text{-}1) + c\,(k\text{+}1) . \quad (7)$$

From the relations (5) and (7) it can be deduced that every coefficient can be determined like:

$$c(k) = f(k) - f''(k)/6, \quad (8)$$

where $f(k) = y(k)$ for $k = 0, 1, \ldots, N\text{-}1$ (in the knots).

The problem now is to establish the values for the second derivatives in the knots. We deal with digital data and it is necessary to determine the values of the divided differences for the input string. There were studied and will be presented 2 methods.

The divided differences of second order can be defined like [6]:

$$f''(k) \cong \frac{f(k+h) - 2f(k) + f(k-h)}{h^2} . \quad (9)$$

From (8) and (9), the coefficients can be determined for any value of k by the next relation:

$$c(k) = y(k) - (y(k+1) - 2y(k) + y(k-1))/6. \qquad (10)$$

The coefficients can be calculated by applying the digital filter H_{2D} to the input signal. The transfer function for this filter is determined from (10) and is illustrated by:

$$H_{2D}(z) = \frac{8-(z+z^{-1})}{6} \quad . \qquad (11)$$

It was searched a second method to determine the divided differences.

It is demonstrated that stronger conditions of continuity can improve the convergence properties [5]. It means that the interpolation function is continuous and his derivatives up to the fourth order are continuous $f(x) \in C^4$. We consider $f(x)$ a polynomial function of 4 degree:

$$f(x) = a + b\,x + d\,x^2 + e\,x^3 + g\,x^4. \qquad (12)$$

The function is analyzed on short intervals because it is piecewise polynomial. The function and the function derivatives of order 1 and 2 have been evaluated. The general formulation for the second derivative is:

$$f''(k) = (-(f(k-2) + f(k+2)) + 16(f(k-1) + f(k+1)) - 30\,f(k))/12. \qquad (13)$$

In every knot k the function $f(k)$ pass thought the input samples $f(k) = y(k)$. The algorithm for calculating the coefficients became:

$$c(k) = y(k) - (-(y(k-2) + y(k+2)) + 16(y(k-1) + y(k+1)) - 30\,y(k))/72. \qquad (14)$$

Using this method, the transfer function for the digital filter became:

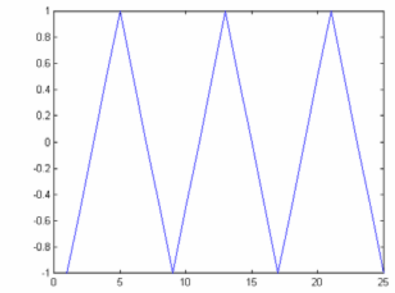
$$H_{2P}(z) = \frac{102 - 16(z+z^{-1}) + (z^2+z^{-2})}{72} \; . \qquad (15)$$

A problem appears at the beginning and the end of the string. We can not use the presented methods to determine the firs and last coefficients. There are need some initial conditions. The functions are evaluated on short intervals to determine the initial values. It is used the same method like for the algorithm that utilize the first derivative for the approximation function [3].

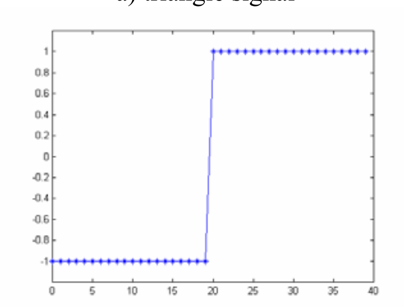
The main advantage of this algorithm is that previous values of other coefficients are not used for calculate the current coefficient. The algorithm is not recursive and the process of error propagation for determining the coefficients is not appearing. The errors appear only from the process of calculating the values for the function derivatives.

III. SOME COMPARATIVE RESULTS

The two presented methods were used to determine the coefficients in several cases. We present the results for two types of signals. The analogical signals that have some discontinuities present a special interest. The input samples were acquired from the signals presented in figure 1. In figure 1a) it is presented a triangle wave. In figure 1b) we took a part from a square signal.



a) triangle signal



b) square signal

Figure1. The input signals

For the same input strings were calculated the coefficients $c(k)$ using each of the two methods and it was performed the signal reconstruction and interpolation in every case. The filters with transfer function $H_{2D}(z)$ or $H_{2P}(z)$ (for each method) are applied on the input signal to obtain the coefficients $c(k)$. The reconstructed signal $yr(k)$ will be obtained at the system output.

The interpolated values are obtained from the coefficients by the same method like in the Unser's algorithm using (4). It was performed the interpolation by a factor m=2. The coefficients string is expanded by m=2 and brought as input for the filter $B_2^3(z)$. The output is the interpolated signal $yin(k)$.

It can be observed that the coefficients follow the signal variation and are close to the samples values. The same results were achieved also for the algorithm presented by Unser in [2] and for the methods used in [4].

For the triangle signal we took two cases, depending on the sampling frequencies. From the same analogical signal, it was acquired $y_1(k)$ with 9 samples per period and $y_2(k)$ with 17 samples per period. In every case we calculate the coefficients and performed the interpolation by a factor m=2. There were analyzed the interpolation errors. It is to observe that for both signals the maximal values of the errors are on the same order. Table I presents these values for the 2 input strings in all the studied cases.

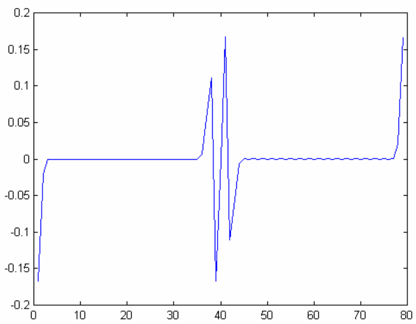
Table I. Maximal interpolation errors for the triangle wave

	$H_{2D}(z)$	$H_{2P}(z)$
$y_1(k)$	0.05902777	0.06539351
$y_2(k)$	0.02951388	0.03269675

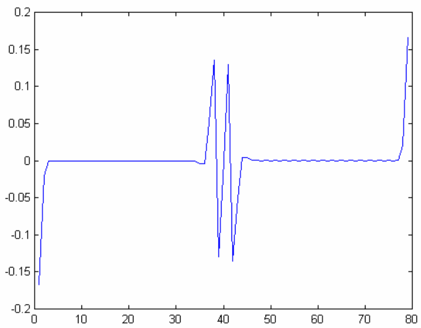
The bigger errors are obtained near by the discontinuity points. A small number of samples are influenced. Using the second method to determine the coefficients we have more samples influenced by significant errors. On the parts were the signal is continuous the errors are very small (considered zero).

We don't take in discussion the side errors.

For the square signal were obtained similar results. Major errors appear near by the discontinuity point. The extreme values are 0.16666666 in case of using $H_{2D}(z)$ for calculate the coefficients and 0.13599537 in the other case ($H_{2P}(z)$ being the transfer function for the filter). Also here the second method doesn't offer much better results. In figure 2 are presented the interpolation errors for both methods used to determine the coefficients.



a) for $H_{2D}(z)$



a) for $H_{2P}(z)$

Figure 2. Interpolation errors for the square signal.

In all presented cases are small differences between the interpolation errors. It can be said that stronger conditions of continuity for the considered function $f(x)$ don't offer better results in the end. In [4] the interpolation errors were smaller for a function $f(x) \in C^4$.

IV. CONCLUSIONS

There was presented an algorithm that use the second derivative values in the knots for calculating the coefficients in B-spline interpolation. We offered two methods for estimate the derivative values. These approaches have a big advantage: the error propagation is avoided. There are used only the input samples to determine every coefficient. It is not a recurrent technique. Any coefficients errors does not influence the others values.

A big advantage is that both methods can be easily implemented on a system with digital signal processor. But the second need supplementary operations.

The two methods for estimate the derivative values in the knots offer similar results for the considered cases. Other practical examples are in study. It is to verify if the extra amount of computation necessary for the second method it is justified.

REFERENCES

1. P. Thevenaz, T. Blu, M. Unser, "Interpolation Revisited", *IEEE Transactions on Medical Imaging*, Vol. 19, No. 7, pp. 739-758, July 2000.
2. M. Unser, "Splines: a Perfect Fit for Signal and Image Processing", *IEEE Signal Processing Magazine*, Vol. 16, No. 6, pp. 22-38, Nov. 1999.
3. L. Stoica., "Contributions in Recursive Filtering for B-splines Interpolation in Signal Processing", *Scientific Bulletin of the "Politehnica" University of Timisoara, Romania, Transactions on Electronics and Communications*, Vol. 51(65), No 2, "Etc2006", pp. 44-49, Sept. 2006.
4. L. Stoica., "A New Algorithm for Determining the Coefficients in B-spline Interpolation", *Scientific Bulletin of the "Politehnica" University of Timisoara*, *Romania, Transactions on Electronics and Communications*, Vol. 51(65), No 2, "Etc2006", pp. 5-8, Sept. 2006.
5. Gh. Micula, *Functii Spline si aplicatii*, Editura Tehnica, Bucuresti, 1978.
6. O. Stanasila, *Analiza Matematica*, Editura Didactica si Pedagogica, Bucuresti, 1981.
7. I. Gh. Sabac, *Matematici Speciale*, vol. 2, Editura Didactica si Pedagogica , Bucuresti, 1965.