Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 51(65), Fascicola 2, 2006

A New Algorithm for Determining the Coefficients in B-spline Interpolation

Liliana Stoica[1]

Abstract – This algorithm is one of the methods that use spline functions for interpolation. In the context of general interpolation the coefficients are calculated using the values of the function and function's derivatives in the knots. Compared with another known algorithm, in this case is not necessary to perform the signal extension. But appear another problem: how to calculate the values for the derived function. Three methods are presented to resolve this. All the methods were applied for several input signals. From the practical results were made some conclusions.
Keywords: interpolation, B-spline functions, divided differences

I. INTRODUCTION

In this world of speed and high performances the interpolation problem remains on actuality. The traditional methods are improved and always are searched new ways to obtain better results with minimum costs. In this paper is presented a new algorithm for determine the B-spline coefficients in the generalized interpolation approach. All started with an algorithm presented in the specialty literature that has some disadvantages.

In Section II are presented the concepts of general and traditional interpolation and an algorithm for B-spline interpolation that use a modern technique. This algorithm was implemented and several observations were made [4]. To eliminate some disadvantages were searched another improved algorithms. In Section III are calculated the initial coefficients for interpolation using the properties of the spline functions: polynomial on short intervals, continuous and differentiable. The coefficients are determined from the input samples and from the derived function values in the knots. This method eliminates the signal extension necessary in the other algorithm. Section IV presents the new algorithm for calculating all the coefficients. This algorithm is based also on the derived function in the knots. The problem is to calculate those values. For that are presented three methods. The practical results of implementing all tree methods are discussed in Section V. There is made also a comparison with the Unser's algorithm results.

II. INTERPOLATION

A. Traditional Interpolation

Consider $y=\{y(\mathrm{k})\}$, k = 0,N-1 a set of discrete data, regularly sampled. To find the interpolated value $f(x)$ it is necessary to calculate:

$$f(x)=\sum_{k\in Z} y(k)\varphi_i(x-k) \tag{1}$$

This is the traditional method to perform the interpolation: using the input data and the basis function values $\varphi_i(x-k)$ that give the sample weights.

B. Generalized Interpolation

Another way to perform the interpolation is to use the generalized formulation [5]:

$$f(x)=\sum_{k\in Z} c(k)\varphi(x-k) \tag{2}$$

In this case the interpolated values are obtained from the coefficients $c(k)$ and not directly from the sample values $y(k)$. This method requires two different steps: determining the coefficients from the input data and calculate the interpolated values with those coefficients. It can be considered that the traditional interpolation is a particular case for $c(k) = y(k)$.

C. Unser's B-spline Interpolation Algorithm

The spline functions were used from a long time in problems of traditional interpolation. These are polynomial functions of degree n on adjacent intervals connected in the knots. The function and his derived up to n-1 order are continuous. These properties make the spline functions easy and convenient to use. For performing high-quality interpolation are often used the cubic spline function (n=3). In the traditional

[1] "Politehnica" University of Timisoara, Faculty of Electronics and Telecomunication, Bd. V. Pârvan No. 2, 300223 Timişoara, e-mail: liliana.stoica@etc.upt.ro

manner the spline interpolation is performed by matrix algebra methods and there is necessary a great amount of operations.

Another approach is to use digital filtering techniques. Michael Unser and his team developed an algorithm that uses digital filters for interpolation [6], [7], [8]. For the cubic B-spline function β^3 (x) in (3) it is defined the discrete B-spline function b_1^3(k) and the direct B-spline filter (4).

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

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

Applying this filter to the input signal are obtained the spline coefficients $c(k)$. The operation is called "direct B-spline transform". The interpolated function $f^n(x/m)$ by a factor m, denoted $f_m^n(x)$ will be:

$$f_m^n(x) = \sum_{k \in Z} c(k) b_m^n(x - km) \tag{5}$$

This operation is called "indirect B-spline transform" and it is implemented also by digital filtering [6], [8].

For calculating the coefficients the direct B-spline filter is implemented by 2 filters: first a causal filter and the second anti-causal. The recursive algorithm demands some initial conditions. Is performed the signal extension by mirroring and they are taken a finite number of samples. The initial conditions introduce some side errors for the coefficients [4]. Those errors are transmitted in the interpolated signal and could have great importance especially if the input signal contains a small number of samples.

III. NEW INITIAL B-SPLINE COEFFICIENTS

To perform the spline interpolation in the traditional manner are used the known input samples and some values of the derived function. From this idea, to determine the new initial coefficients there were evaluated also the derivatives for the input function.

Consider f(x) an approximation for the cubic spline function that pass trough all the input values: f(k) = y(k), k = 0, N-1. In the knots f(k) represent the convolution between the coefficients' string and the cubic B-spline function (2). The relation involving the function and the coefficients c(k) can be write:

$$6f(k) = 4\,c(k) + c(k-1) + c(k+1) \tag{6}$$

The cubic B-spline function derivatives of first and second order are analyzed. From these ones are determined the relations between the f(k) derivatives and the coefficients:

$$f'(k) = 0\,c\,(k) - ½\,c\,(k-1) + ½\,c\,(k+1) \tag{7}$$

$$f''(k) = -2\,c\,(k) + c\,(k-1) + c\,(k+1) \tag{8}$$

The formulas (6), (7) and (8) are evaluated for k=2 to determine the initial values. The first 3 coefficients can be obtained by:

$$c\,(2) = f\,(2) - f''(2)/6 \tag{9}$$

$$c\,(0) = c\,(2) - 2f'(1) \tag{10}$$

$$c(1) = \frac{6f(1) - c(0) - c(2)}{4} \tag{11}$$

Compared with the Unser's algorithm, in this new approach is not necessary to perform the signal extension. But it has to establish a way to determine the values for the function derivatives of order one and two. These values must be obtained by numerical methods only from the input samples. The problem is to calculate $f'(1)$ and $f''(2)$ from the known signal values. The interpolation function is a B-spline (piecewise polynomial), so we can approximate f (k) by a polynomial function on short intervals. With this polynomial and his derivatives we calculate the values for the first 3 coefficients.

IV. A NEW ALGORITM BASED ON NUMERICAL DIFERENTIATION

With 3 initial values calculated it can be established an algorithm to determine the other coefficients. From the relation (7) it can be established a general formulation in every knot:

$$c(k+1) - c(k-1) = 2\,f'\,(k) \tag{12}$$

The algorithm supposes to use the function derivatives and to impose their values. This type of interpolation is called Hermite interpolation.

In the knots the values of the function f (k) must be equal to the input data samples:

$$f\,(k) = y\,(k) \text{ for } k = 0, 1, \ldots, N-1 \tag{13}$$

Dealing with discrete dates, now the problem it is to perform the numerical differentiation. There are discussed 3 methods for calculating those.

A. The First Method

It is used the classical definition for the divided differences [1],[2]:

$$f'(k) = \frac{f(k+1) - f(k)}{(k+1) - k} \tag{14}$$

The divided differences of order 2:

$$f''(k) = \frac{f(k+2) - 2f(k+1) + f(k)}{(k+2) - k} \tag{15}$$

In this case the algorithm for calculating the coefficients for the input signal y(k) became:

$$c(k+1) - c(k-1) = 2[\, y(k+1) - y(k)] \tag{16}$$

B. The Second Method

Stanasila [3] defines the next divided differences:
- the divided differences at left:

$$f'(k) \cong \frac{f(k) - f(k-h)}{h} \tag{17}$$

- the divided differences at right:

$$f'(k) \cong \frac{f(k+h) - f(k)}{h} \tag{18}$$

- by averaging it is obtained:

$$f'(k) \cong \frac{f(k+h) - f(k-h)}{2h} \tag{19}$$

$$f''(k) \cong \frac{f(k+h) - 2f(k) + f(k-h)}{h^2} \tag{20}$$

The same relation can be found by calculating the central derivative for a polynomial function that goes through 3 points.
In this case the initial values are:

$$c(2) = y(2) - (y(3) - 2y(2) + y(1))/6$$
$$c(0) = c(2) - y(2) + y(0)$$
$$c(1) = \frac{6y(1) - c(0) - c(2)}{4}$$

For any k value the iterative relation for calculating the coefficients became:

$$c(k+1) - c(k-1) = y(k+1) - y(k-1) \tag{21}$$

As it can be observed any differences between 2 coefficients $c(k+h)$ and $c(k)$ depends of the samples values in k+h and k points only.

C. The Third Method

The convergence properties can be improved by stronger conditions of continuity. It means that the interpolation function is continuous and his derivatives up to the fourth order are continuous $f(x) \in C^4$. This is demonstrated by a theorem in [1]. So we consider $f(x)$ a polynomial function of 4 degree:

$$f(x) = a + b\,x + d\,x^2 + e\,x^3 + g\,x^4 \tag{22}$$

The function is piecewise polynomial, so it can be analyzed on short intervals. The function and the function derivatives of order 1 and 2 have been evaluated on the interval [0;4] and are obtained the next relations:

$$f'(1) = \frac{-3f(0) - 10f(1) + 18f(2) + f(4)}{12}$$
$$f''(2) = \frac{-(f(0) + f(4)) + 16(f(1) + f(3)) - 30f(2)}{12}$$

The general formulation for the firs derivative is:

$$f'(k) = \frac{f(k-2) - 8f(k-1) + 8f(k+1) - f(k+2)}{12} \tag{23}$$

For $f(k) = y(k)$, the algorithm for calculating the coefficients became:

$$c(k+1) - c(k-1) = \frac{y(k-2) - 8y(k-1) + 8y(k+1) - y(k+2)}{6} \tag{24}$$

It has to demonstrate the algorithm convergence. For that are take into consideration a finite number of successive iterations:

$$c(3) - c(1) = [y(0) - 8\,y(1) + 8\,y(3) - y(4)]/6$$
$$c(5) - c(3) = [y(2) - 8\,y(3) + 8\,y(5) - y(6)]/6$$
$$c(7) - c(5) = [y(4) - 8\,y(5) + 8\,y(7) - y(8)]/6$$
......
$$c(k) - c(k-2) = [y(k-3) - 8\,y(k-2) + 8\,y(k) - y(k+1)]/6$$
$$\Rightarrow \quad c(k) - c(1) = [\,8y(k) - y(k-1) - y(k+1) + 8\,y(1) + y(0) + y(2)]/6$$
$$\Rightarrow \quad c(k) - c(1) = \{y(k) - [y(k+1) - 2y(k) + y(k-1)]/6\} - \{y(1) - [y(2) - 2y(1) + y(0)]/6\} \tag{25}$$

Any differences $c(k) - c(1)$ does not depending on intermediary values, but only the ones related to $y(k)$ and $y(1)$. It can be observed that the expressions in square brackets in (25) represent the divided differences of second order by Stanasila's definition determined in the k and 1 points [3]. The values for $c(k)$ and $c(1)$ are not bounded by intermediary samples of the function, so this function can be arbitrary between k and 1. In this case the method could be generalized and used also for discontinuous signals.

V. COMPARATIVE RESULTS

All tree methods were implemented to determine the coefficients and the algorithms were applied for several known signals. Some significant results are gone be presented along. The input signal were $y(k)=\sin(2\pi k/M)$ or $y(k)=\cos(2\pi k/M)$, k = 0, N-1 for different values of M and N. Were analyzed situations for diverse sampling frequencies (different values for M). For the periodic signals the input string has a small number of samples (M=12 and N=13) corresponding to one period or an increased number of samples equivalent to more than two periods.
For the same input string were calculated the coefficients $c(k)$ using each of the three methods and it was performed the interpolation in every case. The interpolated values are obtained by the same method

like in the Unser's algorithm using the equation (5). It was performed the interpolation by factor m=2.

The results are comparative for the sine and cosine signals. If M=12 the input signal has a small number of samples. Applying the *A* method for calculating the coefficients the interpolation errors are of 10^{-1} order. Almost all the interpolated values are influenced by errors of this range. For the same input string the interpolation errors are 10^{-2} in case of *B* and 10^{-3} for *C* methods.

For a signal with a greater number of samples per period (M=120) the differences between the tree methods are significant. The interpolation errors are 10^{-3} with the classical definition for the divided differences. Using the *B* method these errors became 10^{-4}. If the derived function values are calculated by the polynomial of degree 4 then the interpolation errors are decreasing to 10^{-7}. By increasing the sampling frequency for the input signal are reduced the interpolation errors.

For y(k)=cos(2πk/M) being the input signal, some results are presented in Table 1. In two cases: M=12 and M=120 are given the interpolation errors for some distinctive points α on the function characteristic.

Table1. Interpolation errors for y(k)=cos(2πk/M).

α	M	*A* method	*B* method	*C* method
π/3	12	-0.05502116	-0.00817301	0.00024929
	120	-0.00068398	-0.00022684	-0.00000005
π/2	12	-0.03867513	-0.03867513	-0.00099717
	120	-0.00091239	-0.00045525	-0.00000015
π	12	-0.12200846	-0.06630823	-0.00274223
	120	-0.00136921	-0.00091207	-0.00000040

As it can be seen the *A* method has results that are not too good. The improved method *B* can offer acceptable errors for some applications. It has the advantage of simplicity and requires a relative small number of operations. The algorithm has better results by using the polynomial function of degree 4. But in this case are necessary additional operations. Decreasing errors is possible by increasing the computational costs. The operations are not complicated and they don't take much time for calculating in applications that require better results.

The results can be compared with the ones obtained with the Unser's algorithm where for determining the coefficients it is applied the direct B-spline transform. The errors in this case are greater at the beginning and the end of the data string compared with the data in the middle [4]. This is due to the finite number of samples used at the initialization procedure for determining the coefficients. For M=12 the side values present errors of 10^{-1} order and the others have interpolation errors of 10^{-3}. For M=120 the interpolation errors are 10^{-8} up to 10^{-6} at the beginning and at the end.

The new algorithm has the advantage that the errors introduced by the method of determining the coefficients are the same for all the interpolated samples.

In all studied cases the firsts and lasts 2 interpolated values are influenced by greater errors. These are introduced by the interpolation method. Every interpolated value is obtained from the coefficient corresponding to the current point and some anterior and posterior coefficients (convolution in (5)). Some of these (c(-1) and c(N) for example) are not known and considered zero when calculate the interpolated values on the sides of the string. This problem appears also at the Unser's algorithm. It can be resolved and it will be discussed in to a further paper.

VI. CONCLUSIONS

The algorithm use known techniques combined in new manner. The main advantage of this one compared with the Unser's algorithm is that the input signal don't have to be extended to establishes the initial conditions. The coefficients are calculated using the input samples and the values for the first derivative of the input function in the sampling points. The problem was to determine these values only from the input data. One of the presented methods (the *C* method) offers very good results for the interpolation. The function is approximated by a polynomial of forth order. From this is established the recursion formula for calculate the coefficients.

The algorithm offers simplicity of implementation. It was applied on input signals that are continuous for different sampling frequencies. The presented methods will be tested on other types of signals to observe if the results are as good as the presented ones.

ACKNOWLEDGEMENTS

The author would like to address special thanks to Professor Eugen Pop for his patience, guidance and support.

REFERENCES

[1] Gh. Micula, *Functii Spline si aplicatii*, Editura Tehnica Publishing House, Bucuresti, 1978

[2] I. Gh. Sabac, *Matematici Speciale*, vol. 2, Editura Didactica si Pedagogica Publishing House, Bucuresti, 1965

[3] O. Stanasila, *Analiza Matematica*, Editura Didactica si Pedagogica Publishing House, Bucuresti, 1981

[4] L. Stoica., "Contributions in Recursive Filtering for B-Splines Interpolation in Signal Processing", *Proceedings of the International Symposium on Electronics and Telecommunications ETC 2006*, Timisoara, September 21-23, 2006

[5] P. Thevenaz, T. Blu, M. Unser, "Interpolation Revisited", *IEEE Transactions on Medical Imaging*, Vol. 19, No. 7, pp. 739-758, July 2000.

[6] M. Unser, "Splines: A Perfect Fit for Signal and Image Processing", *IEEE Signal Processing Magazine*, Vol. 16, No. 6, pp. 22-38, Nov. 1999.

[7] M. Unser, A. Aldroubi, M. Eden, "B-Spline Signal Processing: Part I - Theory", *IEEE Transactions on Signal Processing*, Vol. 41, No. 2, pp. 821-833, Feb. 1993.

[8] M. Unser, A. Aldroubi, M. Eden, "B-Spline Signal Processing: Part II – Efficient Design and Applications", *IEEE Transactions on Signal Processing*, Vol. 41, No. 2, pp. 834-848, Feb. 1993.