The Reconstruction of the Compressed Digital Signal using Lagrange Polynomial Interpolation

Vladan Vučković [1]

Abstract - **This paper presents an efficient method for real-time reconstruction of the digitalized signal. The method is based on Lagrange polynomial interpolation. The implementation is suitable for transmission or archiving of the compressed digital signals, with special appliance to audio signals.**

Keywords - **Digital signal processing, Lagrange interpolation**.

I. Introduction

This paper describes an efficient method for reconstruction of the digitalized signal from samples using Lagrange polynomial interpolation. In Digital Signal Processing (DSP) there is necessity to convert a digital signal from one sample rate to another. In that way, the use of a digital computer becomes a key point in digital signal processing. After the processing or transmitting of the signal through digital networks, a digital to analog converter (DAC) is used to convert the digital signal back to analog representation. The input analogue signal is first converted to digital using analogue to digital (ADC) converter running at some rate. The Nyquist-Shannon sampling theorem must be satisfied in order for a sampled analog signal to be exactly reconstructed from its digital representation. This theorem states that the sampling frequency must be greater than twice the bandwidth of the signal. For instance, if a speech audio signal is concerned, 11 kHz of the sampling rate will be enough to reconstruct digitalized voice signal. The sampling frequency is often significantly more than twice the required bandwidth in practice. For instance, the CD digital music recording sample rate is 44 kHz.

For different applications, e.g. digital audio-video signal processing, digital television, digital image processing and software, different sampling rates are specified but data must be transmitted among these applications. Nevertheless, for the digital telecommunication, Internet communication or a long-time archiving of digital recordings the lower sampling rate is preferable.

The signal converted to the minimum sampling rate, given by the Nyquist sampling theorem is also reducing the computational burden.

[1] dr Vladan V. Vučković is with Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia. e-mail: vld@elfak.ni.ac.yu

There are different approaches to solve the problem of reconstruction of the samples between two neighbor points. One of the proposed methods, which is in the main scope of this paper, is based on Lagrange interpolation polynomials, which proved to be an efficient and fast method for real-time applications [1],[2],[3],[4],[5].

II. Lagrange Interpolation

The Lagrange interpolating polynomial is the polynomial that passes through all pre-defined points. The formula was first published by Waring (1779), rediscovered by Euler in 1783, and published by Lagrange in 1795 (Jeffreys and Jeffreys 1988) [1]. For example, in mathematics, it is used in the construction of Newton-Cotes formulas [3].

The constant problem in Lagrange interpolation is a tradeoff between having a better fit and having a smooth well-behaved fitting function. Also the number of data points must be optimal. It is well known that when constructing interpolating polynomials, the more data points that are used in the interpolation cause the higher degree of the resulting polynomial and the greater oscillation in interpolation function between the data points. In that way, a high-degree Lagrange interpolation may predict the function between points with greater error, although the accuracy at the data points will be perfect.

A. Definition

First we will define mathematical fundamentals of the Lagrange interpolation. Let us suppose that A is a field and B is a vector space over C. Elements of B is vectors and elements of C are scalars. If $a_1,...,a_n$ are scalars and $b_1,...,b_n$ are vectors, then the *linear combination* of those vectors with those scalars as coefficients is:

$$a_1b_1 + a_2b_2 + a_3b_3 + \ldots + a_nb_n \quad (1)$$

A and B are specified explicitly. A linear combination of the vectors $b_1,...,b_n$, with the unspecified coefficients (that must be in space C) are often in practice [2]. If S is a subset of space C, we may have a linear combination of vectors in S, where both the coefficients and the vectors are unspecified, except that the vectors must belong to the subset S.

By definition, a *linear combination contains only finite number of vectors.* The subset S that the vectors are taken from can still be infinite. Each individual linear combination will only involve finite number of vectors. Also, number n could be zero. In that case, we declare by convention that the result of the linear combination is the zero vector in B. If we have a set of $k + 1$ data points:

$$(x_0, y_0),\ldots\ldots,(x_k, y_k) \tag{2}$$

where no two x_j are the same, the interpolation polynomial in the Lagrange form is a linear combination:

$$L(x) = \sum_{j=0}^{k} y_j l_j(x) \tag{3}$$

where l_j represents Lagrange basis polynomials:

$$l_j(x) = \prod_{i=0, i\neq j}^{k} \frac{x - x_i}{x_j - x_i} \tag{4}$$

A Lagrange method of interpolation using the polynomial fits to the available values to interpolate between those values could be very effective in digital signal processing. If there are N data values, a polynomial of degree N-1 can be found that will pass through all the points. The Lagrange polynomials provide a convenient alternative to solving the simultaneous equations that result from requiring the polynomials to pass through the data values.

If a real digital signal is concerned, the Lagrange interpolation could reconstruct the continuous approximation signal from the samples. The other advantage of Lagrange interpolation is that the method does not need evenly spaced values in x. However, it is usually preferable to search for the nearest value in the table and then use the lowest-order interpolation consistent with the functional form of the data. High-order polynomials that match many entries in the table simultaneously can introduce undesirable rapid fluctuations between tabulated values. If used for extrapolation with a high-order polynomial, this method may give serious errors.

III. Methods of Signal Reconstruction

There are many methods for digital signal reconstruction mentioned completely developed in theory and proved in practice [6]. In this Section, our intention is to briefly induce some of them concentrating on modern and advanced approaches. We hope that future investigation of the method we proposed will prove the value of Lagrange approach.

The most popular approach in signal reconstructing today represents *Whittaker-Shannon* interpolation formula that is intensively developed and could be found ate many appliances. Many approximation algorithms are based on this formula. If the new methods are concerned, multi-stage filters are in dominance today.

A. Approximation of Signals by Gaussian Functions

The Gaussian approximations are relatively old methods of signal reconstruction. Modern investigations of the approximation property of *Gaussian* functions [7] is a direct corollary to the work of Wiener (1933) on the closure of translations in L_1 and L_2. This observation not only simplifies the proof of the approximation property, but also renders the result applicable, in a more general setting, to other functions (not necessarily Gaussian, see Section III, B.)

B. Chebyshev approximation methods

Low crest-factor of excitation and response signals is desirable in transfer function measurements, since this allows the maximization of the signal-to-noise ratio (SNR's) for given allowable amplitude ranges of the signals [8]. There is a new crest-factor minimization algorithm for periodic signals with prescribed power spectrum. The algorithm is based on approximation of the no differentiable Chebyshev (minimax) norm by l_p-norms with increasing values of p, and the calculations are accelerated by using FFT's. Several signals related by linear systems can also be compressed simultaneously. The resulting crest-factors are significantly better than those provided by earlier methods.

C. Multi-stage Wiener Filter

Subspace-based methods for estimating the direction of arrival are very important due to their high-resolution abilities. A fast and efficient method based on the multi-stage wiener filter technique to estimate subspace is developed, which doesn't require the training or reference signal [9]. The fact that the match filters of multi-stage wiener decomposition can span the signal subspace is simply proved. A more efficient subspace approximation method is obtained. The proposed method has less computation complexity with the approximation performance compared with the classical *eigen-vector decomposition* (EVD) based method, The simulation results, presented in the paper, demonstrate its effectiveness.

D. Lagrange Interpolation in DSP

As we mentioned, Lagrange's interpolation formula is a simple method for finding the unique polynomial of order N that exactly passes through N+1 distinct samples of a signal. Once the Lagrange polynomial is constructed, its value can easily be interpolated at any point using the polynomial equation. This implies, that a signal in interval [a,b] does need a large amount of data to interpolate, but only a constant number of points between them. The rest of the signal data will be reconstructed using Lagrange polynomial. Lagrange interpolation is useful in many applications, including Parks-McClellan FIR Filter Design,IIR Filter Design, Adaptive Filters [10]. The Discrete Fourier Transformation also with Fast Fourier Transformation and Practical Spectral Analysis Lagrange's interpolation method is a simple and efficient way of finding the unique LLth-order polynomial that exactly passes through L+1 distinct samples of a signal. Lagrange interpolation polynomials could be used to obtain good gradient estimations. As an error criterion we take the mean squared error. It could be pointed that squared error converges to N-1 if the number of evaluation points increases to infinity. In real application we will use finite number of points to define good signal approximation.

IV. Discretization of Signals

As has been mentioned, in order to use an analog signal on a computer it must be digitized with an analog to digital converter. This process is named sampling. Sampling has two stages: *discretization* and *quantization*. In a discretization phase, the signals are partitioned into equivalence classes. In this process, the original signal is replacing with representative signal of the corresponding equivalence class. In the quantization phase the signal values are approximated by values from a finite set. The resolution depends on the number of classes or bits defined for one sample. For instance, if we have n-bits there will be 2^n equivalence classes. The Nyquist-Shannon sampling theorem must be satisfied if we need that a sampled analog signal to be exactly reconstructed.

Let us suppose that analogue signal is discretizated with sampling rate f_q. If one sample contains n-bit, the total information needed to represent the whole signal in digital domain, in one second interval, will be $n*f_q$. It is obvious that better definition of the signal needs greater n or fq. If the signal is digitalized and stored, it must be reconstructed to analogue domain. The simplest method is to use samples to directly reconstruct signals from data. It could be easily demonstrated that this method is too rough for quality output signal (Fig.1):

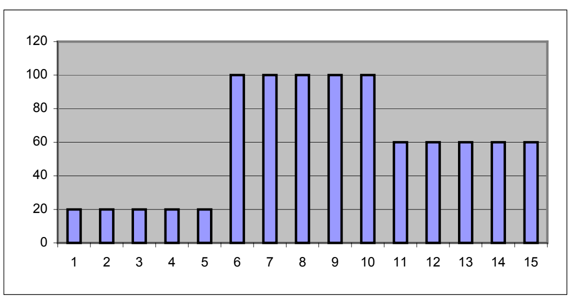

Fig 1. A sampled signal (example).

The next method is to use linear approximation between samples to converge signal (Fig. 2.):

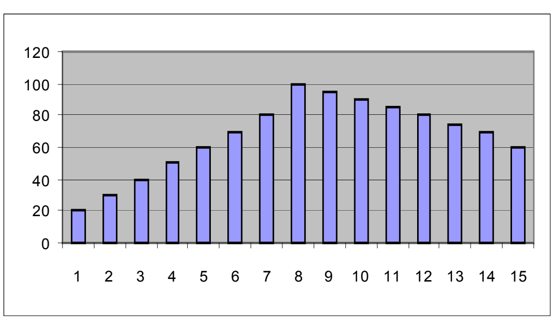

Fig 2. A linear approximation of the sampled signal (example).

But, according to our investigation the Lagrange approximation generates smoothly envelope and better output results. In the diagram (Fig. 4), a x^2 function is interpolated with Lagrange polynomial in 17 points, in interval [-8,8], with equidistant Δx. The original signal is very well fitted with the interpolation function in interval [-8,8]. So, there is no need to handle with all information about the function in interval [-8,8]. It is enough to memorize (or transfer) only *y* coordinates of these points. The original signal will be reconstructed using Lagrange interpolation with small error. Of course, this is an ideal example with perfect sinusoidal function, but the same approach is efficient in real signals.

The real *speech or music signals* are basically the sum of sinusoidal and noise parameters and could be compressed efficiently. If the sample rate is constant, there is no need to memorize *x* parameters, only *y* values have to be stored. If an adaptive sample rate is used, in higher frequencies the sample rate will rise and the amount of data needed to store the signal will also be greater.

V. Algorithm

In this section we will briefly present a key algorithm, coded in *Pascal* (*Delphi for Windows*) that is the core of the Lagrange application [11]. The next listing generates digital representation of the signal based on samples:

```
function fx_lagrange(x:real):real;
var suma,pro:real; i,j,f:integer;
begin
  suma:=0;
  for i:=1 to lnum do
    begin
     pro:=1;
     for j:=1 to lnum do
      if (i<>j) then
       pro:=pro*   (x-l[j].x)   /   (l[i].x-
l[j].x);
      suma:=suma+ l[i].y*pro;
    end;
    fx_lagrange:=suma;
end;
```

This procedure is based on formulas (3) and (4). Variable *LNUM* is the number of interpolation points. The procedure gives the y value based on Lagrange calculation on the input x coordinate. The procedure is very fast because it contains only fast arithmetic operations, without trigonometry ones. This is very suitable for the real time application.

VI. Application

The author has developed the application *Lagrange for Windows* (Fig. 3) in order to perform experiments with different sets of Lagrange interpolation in mathematical or real domain (signals). This application is very useful for developing of the Lagrange interpolation based algorithms. Using the results from different experiments, the author developed a set of supporting interpolation algorithms that could efficiently reconstruct signals from its compressed forms.

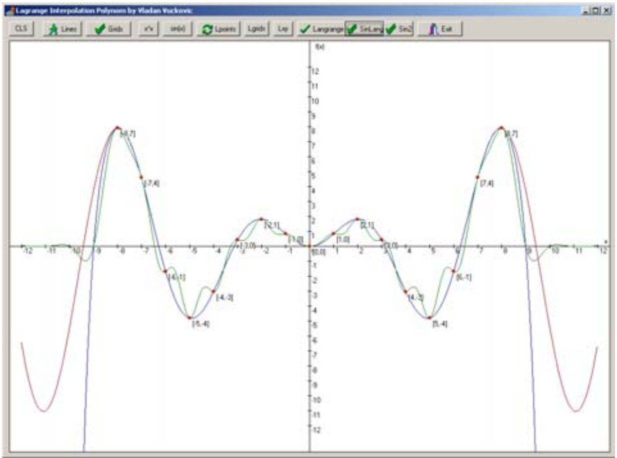

Fig 3. *Lagrange for Windows* application (periodical signal reconstruction).

Some basic functions of the applications are:

- Automatic generation of mathematical functions, loading real signals and approximations using Lagrange interpolation.
- Automatically calculation of middle-square error.
- Automatically generation of root points, their values, and approximation functions.
- Different types of visualization of function.

The application enables different kind of experiments with real or mathematically generated functions. The digitalized signals could be inserted, processed and compressed with this application. Also, it could help in researching of the signal reconstruction from the low sample rate digitalized signal.

VII. Conclusion

In this paper, we have presented the method and basic algorithm of digital signal reconstruction using Lagrange interpolation polynomial. The method is very useful for fast signal reconstruction from the finite number of samples. The mathematical and implementation fundamentals are also presented. As the practical part of this paper, the author has developed the *Lagrange for Windows* application. The application is core experimental platform for researching and implementation of the Lagrange polynomial approach in DSP domain.

The future work will be focused on implementation of the efficient digital signal compression/decompression real-time algorithms based on Lagrange interpolation.

References

[1] Jeffreys, H. and Jeffreys, B. S. *"Lagrange's Interpolation Formula."* §9.011 in Methods of Mathematical Physics, 3rd ed. Cambridge, England: Cambridge University Press, p. 260, 1988.

[2] Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; and Vetterling, W. T. "Polynomial Interpolation and Extrapolation" and *"Coefficients of the Interpolating Polynomial."* §3.1 and 3.5 in Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed. Cambridge, England: Cambridge University Press, pp. 102-104 and 113-116, 1992.

[3] Séroul, R. *"Lagrange Interpolation."* §10.9 in Programming for Mathematicians. Berlin: Springer-Verlag, pp. 269-273, 2000.

[4] Whittaker, E. T. and Robinson, G. *"Lagrange's Formula of Interpolation."* §17 in The Calculus of Observations: A Treatise on Numerical Mathematics, 4th ed. New York: Dover, pp. 28-30, 1967.

[5] Davis P.J. *Interpolation and Approximation*, Dover Publications, New York, 1975.

[6] Alan V. Oppenheim, Ronald W. Schafer, John R. Buck : *Discrete-Time Signal Processing*, Prentice Hall, ISBN 0-13-754920-2

[7] Ferreira, P.J.S.G. *"A comment on the approximation of signals by Gaussian functions"* Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on Volume 45, Issue 2, pp. 250 – 251, 1998.

[8] Guillaume, P. , Schoukens, J. , Pintelon, R. , Kollar, I. *"Crest-factor minimization using nonlinear Chebyshev approximation methods"* IEEE transactions on instrumentation and measurement, (ISSN 0018-9456) , vol. 40, n°6, pp. 982-989, 1991

[9] Bao, Z. , Wu, S. and Zhang, L. *"An efficient subspace approximation method based on multi-stage wiener filter"* , International Association Of Science And Technology For Development Proceedings of the 24th IASTED international conference on Signal processing, pattern recognition, and applications, Innsbruck, Austria, pp: 6 – 8, 2006.

[10] Artur Krukowski, Izzet Kale: *DSP System Design: Complexity Reduced Iir Filter Implementation for Practical Applications,* Kluwer Academic Publishers, ISBN 1-4020-7558-8

[11] Vučković, V. *Digitalna obrada i mašinsko prepoznavanje izolovanih govornih sekvenci*, magistarska teza, Elektronski fakultet u Nišu, Niš, 1997.

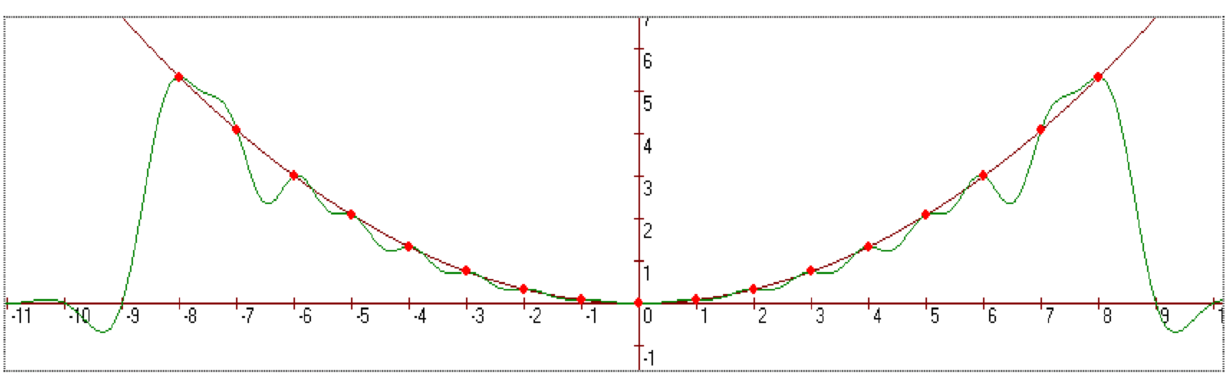

Fig 4. The Lagrange approximation of x^2 signal (red line is original, green line is approximation).