AN IMPROVED ENERGY DEMODULATION ALGORITHM USING SPLINES

Dimitrios Dimitriadis and *Petros Maragos*

Department of Electrical & Computer Engineering,
National Technical University of Athens,
Zografou, 15773 Athens, Greece
ddim@cs.ntua.gr, maragos@cs.ntua.gr

ABSTRACT

A new algorithm is proposed for demodulating discrete-time AM-FM signals, which first interpolates the signals with smooth splines and then uses the continuous-time energy separation algorithm (ESA) based on the Teager-Kaiser energy operator. This spline-based ESA retains the excellent time resolution of the ESAs based on discrete energy operators but performs better in the presence of noise. Further, its dependence on smooth splines allows some optimal trade-off between data fitting versus smoothing.

1. INTRODUCTION

Demodulating AM-FM signals, i.e., nonstationary sines

$$x(t) = a(t)\cos(\int_0^t \omega(\tau)d\tau) \qquad (1)$$

that have a combined amplitude modulation (AM) and frequency modulation (FM), has been a significant research problem with many applications in communications systems, speech processing, and general nonstationary signal analysis. To solve it, a new approach was developed in the 1990's based on nonlinear differential operators that can track the instantaneous energy or its derivatives of a source producing an oscillation [2, 4]. The main such representative is the continuous-time Teager-Kaiser energy operator $\Psi[x(t)] \equiv [\dot{x}(t)]^2 - x(t)\ddot{x}(t)$, where $\dot{x}(t) = dx(t)/dt$. Applied to the AM-FM signal (1), Ψ yields the instantaneous source energy, i.e. $\Psi[x(t)] \approx a^2(t)\omega^2(t)$, where the approximation error becomes negligible [3] if the instantaneous amplitude $a(t)$ and instantaneous frequency $\omega(t)$ do not vary too fast or too much with respect to the average value of $\omega(t)$. Thus, AM-FM demodulation can be achieved by separating the instantaneous energy into its amplitude and frequency components. Ψ is the main ingredient of the first *energy separation algorithm (ESA)*

$$\sqrt{\frac{\Psi[x(t)]}{\Psi[\dot{x}(t)]}} \approx \omega(t) \quad , \quad \frac{\Psi[x(t)]}{\sqrt{\Psi[\dot{x}(t)]}} \approx |a(t)| \qquad (2)$$

developed by Maragos et al. [3] and used for signal and speech AM–FM demodulation.

The instantaneous energy separation methodology has led to several classes of algorithms for demodulating discrete-time AM-FM signals

$$x[n] = x(nT) = A[n]\cos(\int_0^n \Omega[k]dk) \qquad (3)$$

where $A[n] = a(nT)$ and $\Omega[n] = T\omega(nT)$. A direct approach is to apply the discrete-time Teager-Kaiser operator $\Psi_d[x_n] \equiv x_n^2 - x_{n-1}x_{n+1}$, where $x_n = x[n]$, to the discrete AM-FM signal (3) and derive discrete energy equations of the form

$$\Psi_d[x_n] \approx A^2[n]\sin^2(\Omega[n]) \qquad (4)$$

This yields the following algorithm, called *Discrete ESA* [3]:

$$\arccos\left(1 - \frac{\Psi_d[x_n - x_{n-1}] + \Psi_d[x_{n+1} - x_n]}{4\Psi_d[x_n]}\right) \approx \Omega[n] \qquad (5)$$

$$\sqrt{\frac{\Psi_d[x_n]}{\sin^2(\Omega[n])}} \approx |A[n]| \qquad (6)$$

Another approach involves estimating the instantaneous frequency by modeling the discrete-time signal x_n via the exact Prony, as shown in [1, 5]. This yields algorithms that also contain the discrete energy operator as their main ingredient. The best such algorithm is [1]

$$\arccos\left(\frac{\Upsilon_3[x_n] + \Upsilon_3[x_{n-1}]}{4\Psi_d[x_n]}\right) \approx \Omega[n] \qquad (7)$$

where $\Upsilon_3[x_n] \equiv x_n x_{n+1} - x_{n-1}x_{n+2}$ is a third-order energy operator measuring the energy derivative [4]. The instantaneous amplitude is obtained from Eq. (4). We call the combination of Eq. (6),(7) a *Prony ESA*.

The advantages of the ESAs are efficiency, low computational complexity and excellent time resolution (5-sample window) [3]. The main disadvantage is a moderate sensitivity to noise. A more systematic approach is, hereby, developed where we first interpolate the discrete-time signal using smooth splines [6, 7], and then apply the continuous-time ESA of Eq. (2). We begin with a brief background on splines and then develop and test the new algorithm.

2. SPLINE SIGNAL INTERPOLATION

2.1. Exact Splines

Spline functions are piecewise polynomials constructed as a linear combination of B-Splines. A spline function of order n has continuous (smooth) derivatives up to order $n - 1$, which is very important when using Ψ.

This research work was supported by the Greek Secretariat for Research and Technology and by the European Union under the program ΕΠΕΤ-98 with Grant # 98ΓΤ26 and the program ΠΕΝΕΔ-99 with Grant # 99ΕΔ164. It was also supported by the basic research program ARCHIMEDES of the NTUA Institute of Communication and Computer Systems.

Given the initial signal samples $x[n],\ n = 1, \ldots, N$, the interpolating spline function is given by

$$g_\nu(t) = \sum_{n=-\infty}^{+\infty} c[n]\beta_\nu(t-n) \tag{8}$$

where $\beta_\nu(t)$ is the B-spline of order ν, and the coefficients $c[n]$ depend only on the data $x[n]$ and the analytic expression of the B-spline. The B-spline can be formed as the the $(\nu+1)\, th$–fold convolution of the zeroth–order B-spline with itself:

$$\beta_\nu(t) \equiv \underbrace{\beta_0(t) * \beta_0(t) * \cdots * \beta_0(t)}_{(\nu+1)\ times},$$

where the zeroth–order B-spline is defined by

$$\beta_0(t) = \begin{cases} 1 & \text{if } -1/2 < t < 1/2 \\ 1/2 & \text{if } |t| = 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Using the discrete B-spline $b_\nu(n) \equiv \beta_\nu(n)$, Eq. (8) becomes

$$g_\nu(n) = (c * b_\nu)(n) \tag{9}$$

Note that for the exact interpolation problem is $g_\nu(n) = x[n]$. By transforming Eq. (9) in the Z domain,

$$C(z) = \frac{X(z)}{B_\nu(z)} \tag{10}$$

Thus, the spline coefficients $c[n]$ can be determined recursively from the above equation. Note that each original sample $g_\nu(n) = x[n]$ is resynthesized by the contributions of $\nu+1$ neighbor spline coefficients.

2.2. Smooth Splines

We have used splines to improve the performance of the ESA algorithm. Several experiments were conducted concerning different kinds of splines. In these experiments we used noisy AM-FM signals with different levels of SNR. The results were disappointing as the exact fitting of the curve, due to the presence of noise, was creating large estimation errors. This led to the search for a solution dealing with the problem of noise. The next step was to find a way to optimally interpolate signal samples passing closely but not exactly through them. Smooth splines are ideal for this purpose. The main advantage of smooth splines is that the interpolating polynomial does not pass through the signal samples but close enough. This concept is similar to the idea of polynomial fitting with least squares.

The smooth spline interpolating function is the function s_ν of order $\nu = 2r - 1$ that minimizes the mean square error criterion

$$\epsilon = \underbrace{\sum_{n=-\infty}^{+\infty} (x[n] - s_\nu(n))^2}_{\epsilon_d} + \lambda \underbrace{\int_{-\infty}^{+\infty} \left(\frac{\partial^r s_\nu(x)}{\partial x^r}\right)^2 dx}_{\epsilon_s}$$

where ϵ_d is the mean square error of the interpolation function and ϵ_s is the mean square error introduced by the need for a smoothed curve. This criterion is a compromise between the need for close-to-the-data points interpolation curve and the need for a smoothed curve. The positive parameter λ quantifies how smooth the interpolating curve will be and how close to the data points the interpolant will pass. For $\lambda = 0$ there is no smoothing and the interpolation curve fits exactly the signal samples. If $\lambda \neq 0$, the deviation from the data samples increases with the parameter λ.

As shown in [6, 7], the interpolating function $s_\nu(t)$ minimizing the mean square error ϵ is a linear combination of splines β_ν, as in Eq. (8), but the coefficients $c[n]$ are computed as the output of an IIR filter that is different from the filter in (10):

$$C(z) = H_\nu^\lambda(z)X(z) = \frac{X(z)}{P_\nu^\lambda(z)} \tag{11}$$

where $P_\nu^\lambda(z)$ is equal to

$$P_\nu^\lambda(z) = B_\nu(z) + \lambda(-z + 2 - z^{-1})^{\frac{\nu+1}{2}} \tag{12}$$

The IIR filter $H_\nu(z)$ has a symmetric impulse response and all its poles are inside the unit circle. Thus, the spline coefficients $c[n]$ can be determined stably via a few recursive equations [6, 7]. Henceforth, smooth splines with $\lambda \neq 0$ are applied unless stated otherwise.

3. SPLINE ESA

Our previous discussion has established that, by using the theory of smooth splines, we can interpolate the original discrete-time signal $x[n]$ using smooth splines of νth order and create a continuous-time signal

$$s_\nu(t) = \sum_{n=-\infty}^{+\infty} c[n]\beta_\nu(t-n) \tag{13}$$

Obviously, the signal $s_\nu(t)$ is a continuous-time expansion of the original discrete signal $x[n]$. Now, the basic idea of the new approach for ESA-based demodulation is to apply the continuous-time energy operator Ψ and the continuous ESA to the continuous-time signal $s_\nu(t)$, instead of using the discrete energy operator Ψ_d and the DESA on the discrete signal $x[n]$:

$$\Psi[s_\nu(t)] = \left[\frac{\partial s_\nu(t)}{\partial t}\right]^2 - s_\nu(t)\frac{\partial^2 s_\nu(t)}{\partial t^2} \tag{14}$$

In order to use the continuous ESA we have to compute the first-, second- and third-order derivatives of the signal. Given the coefficients $c[n]$ of the spline interpolation (13), we can derive the following closed-form expressions for these derivatives that involve only the coefficients $c[n]$ and the B-spline functions:

$$\frac{\partial s_\nu(t)}{\partial t} = \sum_n (c[n] - c[n-1])\,\beta_{\nu-1}(t - n + 1/2) \tag{15}$$

$$\frac{\partial^2 s_\nu(t)}{\partial t^2} = \sum_n (c[n+1] - 2c[n] + c[n-1])\,\beta_{\nu-2}(t-n) \tag{16}$$

$$\frac{\partial^3 s_\nu(t)}{\partial t^3} = \sum_n (c[n+1] - 3c[n] + 3c[n-1] - c[n-2]) \cdot\beta_{\nu-3}(t - n + 1/2) \tag{17}$$

By using these signal derivatives in the continuous ESA equations (2), we can estimate the instantaneous amplitude $a(t)$ and frequency

$\omega(t)$ of the continuous signal $s_\nu(t)$. Finally, by sampling these information-bearing signals, we obtain estimates of the instantaneous amplitude $A[n] = a(nT)$ and frequency $\Omega[n] = T\omega(nT)$ of the original discrete signal $x[n]$. This whole approach above is called the **Spline ESA**.

An important part of the Spline ESA is the computation of the spline coefficients $c[n]$. Next we discuss the details of this procedure. First, we find the zeros of the denominator polynomial $P_\nu^\lambda(z)$ in Eq. (11). Due to the symmetric form of this polynomial, the zeros come in pairs (z_i, z_i^{-1}), $i = 1, \ldots, r$. Thus, the transfer function in Eq. (11) can be written as

$$H_\nu(z) = c_0 \prod_{i=1}^{r} \frac{-z_i}{(1 - z_i z^{-1})(1 - z_i z)} \qquad (18)$$

From Eq. (11), (18) and [6, 7] the recursive equations are:

$$\begin{aligned} y_i^+[n] &= y_{i-1}[n] + z_i y^+[n-1] \quad , n = 2, \ldots, N \\ y_i[N] &= a_i(2y_i^+[N] - y_{i-1}[N]) \qquad (19) \\ y_i[n] &= z_i(y_i[n+1] - y_i^+[n]) \quad , n = N-1, \ldots, 1 \end{aligned}$$

where $a_i = -z_i/(1 - z_i^2)$, $y_{i-1}[n]$ is the input and $y_i[n]$ is the output of a digital filter with transfer function

$$T_i(z) = \frac{-z_i}{(1 - z_i z^{-1})(1 - z_i z)},$$

and $y_0[n] = x[n]$. If the above step is repeated as many times as the number of pole pairs (z_i, z_i^{-1}), the final output sequence $y_r[n]$ will equal $c[n]$. As boundary condition we set,

$$y_i^+[1] = \sum_{i=1}^{k_0} z_i^{|k-1|} y_{i-1}[k]$$

where k_0 is an integer that ensures a certain level of precision.

We present an example to clarify some of the steps in determining $c[n]$ for splines of order $\nu = 5 (r = 3)$ and for two different values of λ, $\lambda = 0$ and $\lambda = 0.5$. First, if $\lambda = 0$, we have interpolation with exact splines of order $\nu = 5$. The denominator of the transfer function $H_5(z)$ will be $P_5(z) \equiv B_5(z)$ and the poles will be,

$$\begin{aligned} z_1 &= -0.04309, & z_2 &= 0.43057, \\ z_3 &= z_1^{-1}, & z_4 &= z_2^{-1} \end{aligned}$$

Now, for $\lambda = 0.5$, by setting $r = 3$ in Eq. (12) we find the poles of $H_5(z) = 1/P_5(z)$:

$$\begin{aligned} &z_1 = 0.32548, \\ &z_2 = 0.32154 - 0.47128i, \; z_3 = 0.32154 + 0.47128i, \\ &z_4 = z_1^{-1}, \;\; z_5 = z_2^{-1}, \;\; z_6 = z_3^{-1} \end{aligned}$$

In both cases we find $c[n]$ by using the algorithm of Eq. (19). The only difference between the case with $\lambda = 0$ and the case with $\lambda \neq 0$ is the number and the values of the poles. Finally, having computed $c[n]$, the coefficient signal is convolved with the B-spline β_ν to yield the inerpolating signal s_ν.

In general, the evaluation of the spline coefficients by the filtering approach presented above is less computationally complex than the standard numerical analysis approach using sparse Toeplitz matrices.

4. EXPERIMENTS AND DISCUSSION

For the experiments conducted we used parametric AM-FM signals, [1, 3] and different levels of white Gaussian noise:

$$x[n] = (1 + 0.05k\cos(\frac{\pi n}{100}))\cos(\frac{\pi n}{5} + m\sin(\frac{\pi n}{100})) + e[n],$$

where $e[n]$ is the added noise and $m = 1, \ldots, 10$, $k = 1, \ldots, 10$. The first experiments concerned Spline ESA using exact splines, from fifth to seventh order. The experimental results were not so promising as the frequency mean estimation error of the spline ESA was always worse than the corresponding one of the Prony ESA and the DESA as shown in Fig. 1.

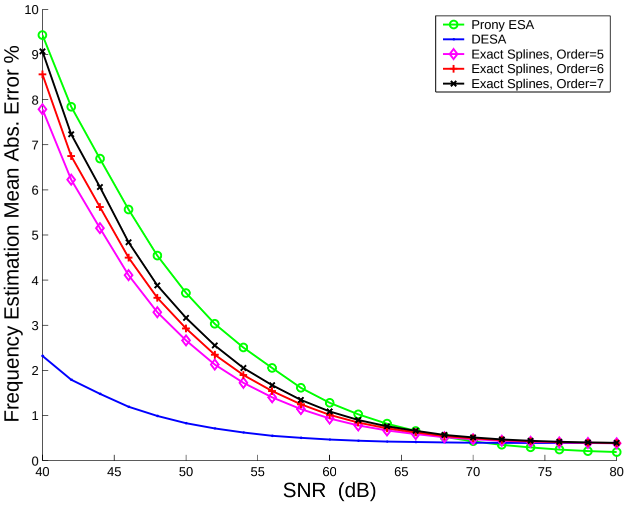

Fig. 1. Comparison of Spline ESA with exact splines with DESA and Prony ESA.

The exact spline fitting is responsible for the large mean square error of the algorithm, as the noisy samples insert a significant error. Since noisy signals have large variations and sharp edges, the need of a smoothing factor is apparent. So, the next series of experiments was conducted using smooth splines with different values of λ through a wide range of SNR values, which are shown in Fig. 2. Spline ESA using smooth splines performs better than the DESA and the Prony ESA in the presence of noise. The choise of value of λ is not completely arbitrary. We have attempted to find experimentally a good value for λ for different SNRs. Fig. 3 shows the mean error of the cubic and fifth–order smooth splines for various values of λ when SNR=35 dB. In these experiments the corresponding mean square error curves had minima for particular values of λ. More specifically, the minima occurred when $\lambda \in [0.1, 1]$ independently of the SNR's values. Note that, the mean square error of the fifth–order smooth spline is always smaller that the corresponding one of the cubic smooth spline independently of the value of λ or SNR. The optimal value of λ is not known and can be determined only through experimentation. This happens because the errors depend on the SNR, the signal, and the application.

Now, we compare the Spline ESA with DESA and Prony ESA, fixing λ equal to a constant value. For SNR$= \infty$, λ could be set to zero. Otherwise for SNR$\neq \infty$, λ takes values in the order of 0.25. The order of the smooth spline will be $\nu = 5$ $(r = 3)$ because the mean square error of the Spline ESA with fifth-order

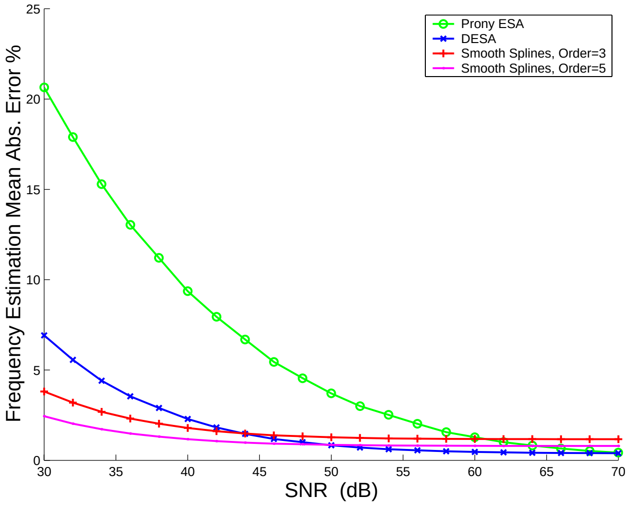

Fig. 2. Comparison of Spline ESA with smooth splines ($\lambda = 0.25$) with DESA and Prony ESA.

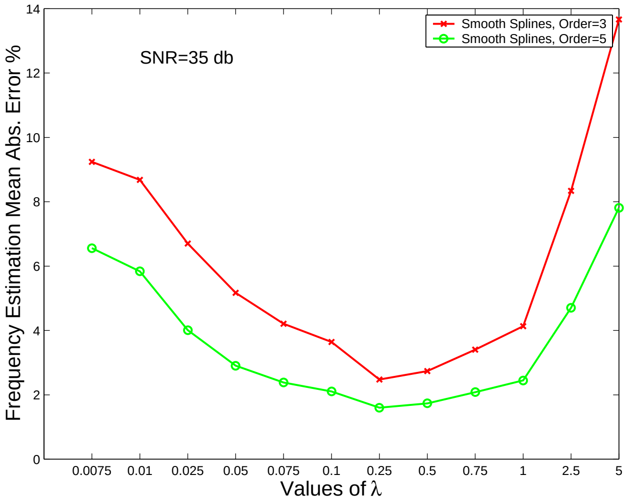

Fig. 3. Mean Square Frequency Error for Spline ESA with cubic and fifth-order smooth splines when SNR=35 dB.

smooth splines is always smaller than the error of the Spline ESA with cubic splines, as it is shown in Fig. 3. In addition, by setting $\nu = 5$, the time–window (i.e., the number of coefficients required to produce one output sample) of Spline ESA is equal to that of the DESA and the Prony ESA.

We tested all three algorithms for SNR$= +\infty$ (no noise is added to the signal) and for SNR=45 dB. In Table 1 we compare the DESA, Prony ESA and Spline ESA with different values of λ. When the SNR is small, smooth splines have better performance and the estimation error is much smaller than the corresponding one for the exact splines. This is the main reason for using smooth splines instead of exact ones. The Prony ESA algorithm, even though it yields very good results (mean square error 0.14%) when there is no noise, is not robust for noisy signals and the corresponding error increases dramatically as the signal's SNR is decreasing. On the contrary, the DESA and especially, the Spline ESA, when $\lambda \neq 0$, are more robust for signals with low SNRs. This fact is clearly represented in Fig. 2 and Table 1. Also, it must be noticed that in Fig. 2 the value of λ does not change with the SNR values (its value is constant). This is why for SNR values bigger than 70 dB, DESA seems to perform better than Spline ESA. Changing λ in inverse proportion to the SNR value, the Spline ESA yields a mean square error that is smaller than the corresponding one of DESA.

Table 1. Comparison of ESA Demodulation Algorithms.

Frequency Mean Absolute Error (%)			
SNR= $+\infty$ dB			
DESA	Prony ESA	Spline ESA $\nu = 5,\ \lambda = 0$	Spline ESA $\nu = 5,\ \lambda = 0.25$
0.38767	0.14774	0.37205	0.79749
SNR=45 dB			
DESA	Prony ESA	Spline ESA $\nu = 5,\ \lambda = 0$	Spline ESA $\nu = 5,\ \lambda = 0.25$
1.32737	6.02938	4.56928	0.95701

In conclusion the proposed new algorithm provides an efficient way to represent a discrete signal in continuous-time domain, which is very important for applying the Teager-Kaiser energy operator where the signal's derivatives must be determined with high precision. Further, using smooth splines gives the new algorithm some additional robustness in the presence of noise.

5. REFERENCES

[1] L. B. Fertig and J. H. McClellan, "Instantaneous Frequency Estimation Using Linear Prediction With Comparisons to the DESAs", *IEEE Signal Process. Lett.*, vol. 3, pp. 54-56, Feb. 1996.

[2] J. F. Kaiser, "On a simple algorithm to calculate the ènergy' of a signal", in *Proc. IEEE ICASSP-90*, Albuquerque, NM, Apr. 1990, pp. 381–384.

[3] P. Maragos, J. F. Kaiser, and T. F. Quatieri, "Energy Separation in Signal Modulations with Application to Speech Analysis", *IEEE Trans. Signal Processing*, vol. 41, pp. 3024–3051, Oct. 1993.

[4] P. Maragos and A. Potamianos, "Higher Order Differential Energy Operators", *IEEE Signal Process. Lett.*, vol. 2, p. 152-154, Aug. 1995.

[5] C. S. Ramalingam, "On the Equivalence of DESA-1a and Prony's Method When the Signal is a Sinusoid", *IEEE Signal Process. Lett.*, vol. 3, pp. 141-143, May 1996.

[6] M. Unser, "Splines A Perfect Fit for Signal and Image Processing", *IEEE Signal Process. Magaz.*, pp. 22-38, Nov. 1999.

[7] M.Unser, A. Aldroubi and M. Eden "B-Spline Signal Processing: Part I–Theory. Part II–Efficient Design and Applications", *IEEE Trans. Signal Processing*, vol. 41, pp. 821–848, Feb. 1993.