EXPONENTIAL-PLUS-CONSTANT FITTING BASED ON FOURIER ANALYSIS

Matthieu Hodgkinson
National University of Ireland, Maynooth
matthew.hodgkinson@nuim.ie

ABSTRACT

A closed form method for Exponential Plus Constant (EPC) curve fitting is introduced. The method, referred to as Fourier-based Exponential Plus Constant Fitting (FEPCF), returns estimates of the EPC coefficients for user-input sets of equidistant time values and corresponding data points. An analytical expression for the exponent's coefficient is found through algebraic manipulation of the Fourier series of the EPC, taken over a finite time interval. The unknown is therefore approximated through the Discrete Fourier Transform (DFT) of the input sequence, itself used as an approximation of Fourier series. The discrepancy between Fourier series and DFT, byproduct of aliasing, is reduced with sine windowing. This is the essential contribution of the paper – the exponent coefficient found, what remains of the problem can be approached with standard linear least squares. FEPCF was motivated by the need for a measuring tool for the time-modeling of fundamental frequency and inharmonicity coefficient of tension-modulated strings. Care is however taken in the mathematical formulation of the method to facilitate its transportation to other fields of science. MATLAB function implementation and a demonstration of FEPCF can be downloaded from the address `http://www.cs.nuim.ie/~matthewh/`.

1. INTRODUCTION

On the occasion of the 13th International Conference on Digital Audio Effects (DAFx - 10), the author proposed mathematical time-dependent models for the Fundamental Frequency (FF) and Inharmonicity Coefficient (IC) of freely vibrating string tones undergoing tension modulation [4]. These models account for the phenomenon of pitch-glide, taking place when the transverse vibrational amplitude is such that changes in length (and thus tension) of the string cause an audible and measurable downwards glide in the fundamental frequency. It was found that the inharmonicity coefficient too underwent change of a definite trend, affecting significantly the trajectory of higher partials.

On the basis of the model for tension modulation formulated by Legge and Fletcher in [7] and the physical expression of the IC found in Fletcher *et al.* [1], it was argued in [4] that both the FF and the inverse of the IC could be generalised to Exponential-Plus-Constant (EPC) functions, of the form

$$x(t) = x_\Delta e^{\sigma t} + x_\infty, \tag{1}$$

where : σ is the growth rate ; x_∞ is the value $x(t)$ converges to as t approaches $-\infty$ (positive σ) or ∞ (negative σ) ; and x_Δ is the difference between $x(0)$ and x_∞, or $x(0) = x_\Delta + x_\infty$. Positive values of σ correspond to exponential growth, and negative values of σ, to exponential decay.

This paper deals with the problem of estimating the coefficients σ, x_Δ and x_∞, which is non-linear essentially because of the presence of an unknown in the exponent. Linearisation is possible through differentiation and logarithmic transformation, but this approach is sensitive to noise in rather ordinary cases. When such EPC models appear in other areas of science (e.g. molecular physics [2], visual sciences [8], applied physiology [6]), generic non-linear least-squares techniques are generally used. A algorithmic approach specifically designed for EPC fitting was attempted instead in [4]. However, the proposed method suffered several inconveniences, mainly complexity of implementation and substantial computational cost.

In contrast, the method presented here can be implemented in about a page of MATLAB code, and is computed almost instantly. A rigorous and knowing frequency-domain approach, using the similarities of and minimising the differences between Discrete Fourier Transform (DFT) and Fourier series, is presented in section 2. This frequency-domain detour provides an estimate of the exponent coefficient, which reduces the finding of the other two unknowns to a least-squares problem, formulated in section 3. Finally, in section 4, the performance of the method is tested and compared with a standard Non-Linear Regression algorithm, in terms of fit quality and computation time. The paper closes on a demonstration of the method with the fitting of FF and IC in string tones of contrasting characteristics.

2. FOURIER-BASED EXPONENT COEFFICIENT ESTIMATE

This section encapsulates the essence of the method, resorting to Fourier analysis to approximate the exponent coefficient σ, and thus reduce the non-linear problem to linearity.

2.1. Rectangle-windowed EPC spectrum

Consider a sequence $\mathbf{x}$ of N samples of (1)

$$\mathbf{x} = [x_0, x_1, x_2, ..., x_{N-1}]^T. \tag{2}$$

The following development requires the samples to be time-equidistant, i.e. $x_n = x(nT_s + t_0)$, where $n \in [0, N-1] \cap \mathbb{Z}$, where T_s is the time interval (in seconds per sample) between each sample taken, and t_0 is the time corresponding to the first sample x_0. The Fast Fourier Transform (FFT) of sequence (2) is

$$\bar{X}[k] = \sum_{n=0}^{N-1} x_n e^{-j\omega_k n}, \tag{3}$$

where $\omega_k = 2\pi k/N$. In the symbol $\bar{X}$, the bar above X symbolises the equivalence with the Z transform of the product of the sampled function with a rectangle window, i.e.

$$X(\omega_k) = \bar{X}[k] = \sum_{n=-\infty}^{\infty} x_n r(n) e^{-j\omega_k n}, \tag{4}$$

with

$$r(n) = \begin{cases} 1, & n \in [0, N-1] \\ 0, & n \notin [0, N-1] \end{cases}. \tag{5}$$

This added specification shall be found relevant when the issue of aliasing is approached.

The Fourier series of the continuous-time function (1) needs now be expressed. In computing environments such as MATLAB, the FFT is implemented as in (3). To allow the use of this efficient algorithm in the implementation of FEPCF, it is necessary to adapt the Fourier series' expression to match that of the FFT. Hence, in contrast with the conventional formulation of the Fourier series found in such textbooks as [3] and [9], a complex exponential form is used here, and the integration interval is set to $[0, N]$. Thus,

$$\bar{\mathcal{X}}[k] = \int_0^N x(nT_s + t_0) e^{-j\omega_k n} dn. \tag{6}$$

Substitution of (1) into (6) and development yields

$$\bar{\mathcal{X}}[k] = x_\Delta e^{\sigma t_0} \frac{e^{\sigma T_s N} - 1}{\sigma^2 T_s^2 + \omega_k^2} (\sigma T_s + j\omega_k), \tag{7}$$

for any $k \neq 0$. From (7), σ can easily be obtained as

$$\sigma = \frac{2\pi k}{N T_s} \frac{\bar{\mathcal{X}}_{\mathrm{R}}[k]}{\bar{\mathcal{X}}_{\mathrm{I}}[k]}, \tag{8}$$

where X_{R} denotes the real part of the complex number X, and X_{I}, its imaginary part.

In practice, only samples of $x(t)$ are given, and therefore the DFT – or its efficient FFT implementation – is the only frequency-domain expression available for use. $\bar{X}$ only being an approximation of $\bar{\mathcal{X}}$,

$$\bar{\sigma}_k = \frac{2\pi k}{N T_s} \frac{\bar{X}_{\mathrm{R}}[k]}{\bar{X}_{\mathrm{I}}[k]} \tag{9}$$

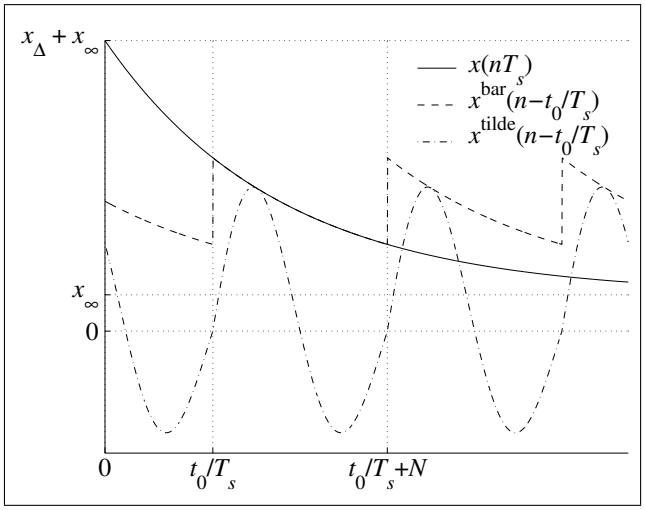

Figure 1. EPC function (solid line) and inverse transform of the rectangle- (dash-dotted line) and sine- (dashed line) windowed EPC function's Fourier series.

results, in turn, in being an approximation of σ. Notice that, while the equality in (8) holds for any k other than 0, the approximation $\bar{\sigma}_k$ is different for each k, hence the use of the subscript. This index-dependent approximation error can be looked upon as the effect of aliasing, which is considered and handled in the following section.

2.2. Aliasing bias

It must be realised that the output of the Inverse Fourier Series (IFS),

$$\begin{aligned} \bar{x}(n) &= \sum_{k=-\infty}^{\infty} \bar{\mathcal{X}}[k] e^{j\omega_k n} \\ &= x(\mathrm{mod}(n, N) T_s + t_0), \end{aligned} \tag{10}$$

is periodic in N. This is basically the function $x(t)$ in interval $t \in [t_0, t_0 + NT_s]$, repeated over and over. The phenomenon is illustrated in Figure 1, where the output of the IFS is denoted by the dashed line. The sharp corners every N samples are indicative of strong partials in higher frequency regions. In fact, $\bar{x}(n)$ visibly lies in between a sawtooth wave, whose spectral peaks are inversely proportional to their partial number, and a pulse train, whose spectrum does not decay but remains constant in magnitude.

On the other hand, the properties of DFT spectra of real signals are such that harmonics of frequency $2\pi l/N$, for any $l \in \mathbb{Z}$, are folded over in the interval $[0, N-1]$ and found in the spectrum at indices $k = \mathrm{mod}(\pm l, N)$, a phenomenon known as *aliasing*. Thus, continuous-waveform harmonics of index outside $[0, N/2]$, when discretised, are susceptible to interfering with harmonics originally within this interval. For the purpose of this paper, this effect is problematic, as it is responsible for the bias between $\bar{\mathcal{X}}$ and $\bar{X}$, and therefore of the approximation in $\bar{\sigma}_k$.

2.3. Reduction of aliasing with sinusoidal windowing

In a nutshell, the strength of aliasing depends on both the "brightness" of the analysed waveform and the low number of data points N. Increasing the number of data points might be an option, even for data sets of fixed size, wjere cubic interpolation can be used to create more points. However, this type of interpolation is sensitive to noise, when present in the measurements.

Instead, reducing the brightness of the waveform is chosen here. The Fourier series is fed with the product of x and a smoothing window, yielding the spectrum

$$\tilde{\mathcal{X}}[k] = \int_0^N x(nT_s + t_0)w(n)e^{-j\omega_k n}dn, \qquad (11)$$

approximated with the FFT

$$\tilde{X}[k] = \sum_{n=0}^{N-1} x_n w(n)e^{-j\omega_k n}. \qquad (12)$$

A window w was needed that made the IFS output $\tilde{x}$ smoother than $\bar{x}$, while at the same time exhibiting spectral properties simple enough not to make $\tilde{\mathcal{X}}$ much more complicate than $\bar{\mathcal{X}}$. The Hanning window was first thought of, but then it was realised that a yet simpler sine window,

$$s_N(n) = \sin\frac{2\pi n}{N}, \qquad (13)$$

satisfied both requirements. The IFS output $\tilde{x}$ thus obtained is represented with the dash-dotted line in Figure 1. The reader can appreciate the contrast in smoothness between $\tilde{x}$ and $\bar{x}$, promising of a dramatic reduction of aliasing. Supportingly of this statement, Figure 2 shows the real (left) and imaginary (right) parts of the Fourier series (circles) and DFT (crosses), without (upper plots) and with (lower plots) sinusoidal windowing. The number of points was purposely made low ($N = 10$) to emphasize the effect of aliasing. The discrepancy between the real parts of the Fourier series and DFT in the case of rectangular windowing is considerable for data sets of so few entries, but many times smaller, if not absent, when sinusoidal windowing is used.

Also, the simplistic, dual impulse spectrum of the sine window,

$$\mathcal{S}_N[k] = j\frac{N}{2}(\delta(k+1) - \delta(k-1)), \qquad (14)$$

keeps the Fourier series $\tilde{\mathcal{X}}[k]$ simple enough to yield, after algebraic manipulation,

$$\sigma = \frac{2\pi}{NT_s}\frac{\tilde{\mathcal{X}}_{\mathrm{R}}[k]}{\tilde{\mathcal{X}}_{\mathrm{I}}[k]}\left(k \pm \sqrt{k^2 + \frac{\tilde{\mathcal{X}}_{\mathrm{I}}[k]^2}{\tilde{\mathcal{X}}_{\mathrm{R}}[k]^2}(k^2-1)}\right) \qquad (15)$$

for any $|k| > 1$.

2.4. On the benefit of an extra sample

It should be noted from substitution of (13) into (12) that the sine window brings to zero the first sample of the sequence, x_0, which might be seen as a waste of a sample. For lengthy $\mathbf{x}$ this has mostly negligible effect, but for sequences of very small length, one sample more or less does make a difference. In this regard the theory should be modified slightly. The FFT size can be augmented by one sample, i.e. $N' = N + 1$, and an extra zero sample be added to the head of $\mathbf{x}$ to produce

$$\begin{aligned}\mathbf{x}' &= [0, x_0, x_1, ..., x_{N-1}]^T \\ &= [x'_0, x'_1, x'_2, ..., x'_{N'-1}]^T. \end{aligned} \qquad (16)$$

The continuous-time transform is accordingly modified, and becomes

$$\tilde{\mathcal{X}}'[k] = \int_0^{N'} x((n-1)T_s + t_0)s_{N'}(n)e^{-j\omega_k n}dn, \qquad (17)$$

which, in discrete time, translates to

$$\tilde{X}'[k] = \sum_{n=0}^{N'-1} x'_n s_{N'}(n)e^{-j\omega_k n}. \qquad (18)$$

Use of $\tilde{X}'$ in place of $\tilde{\mathcal{X}}$ to derive σ estimates yields

$$\tilde{\sigma}_k = \frac{2\pi}{N'T_s}\frac{\tilde{X}'_{\mathrm{R}}[k]}{\tilde{X}'_{\mathrm{I}}[k]}\left(k \pm \sqrt{k^2 + \frac{\tilde{X}'_{\mathrm{I}}[k]^2}{\tilde{X}'_{\mathrm{R}}[k]^2}(k^2-1)}\right), \qquad (19)$$

which is the final result of this section's development.

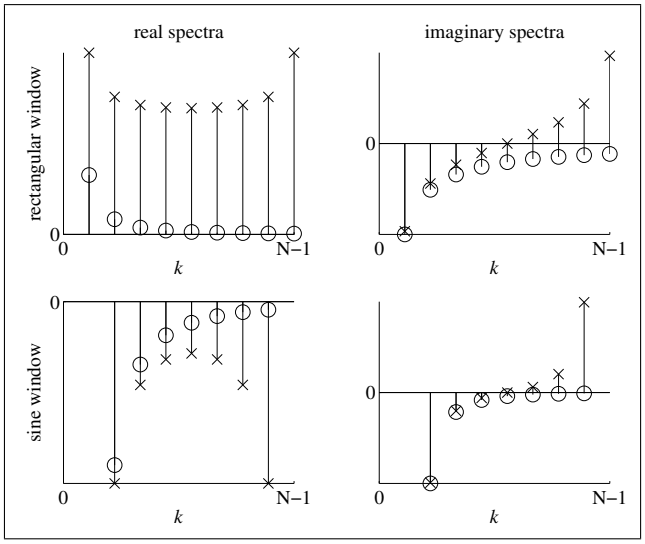

Figure 2. EPC Fourier series (circles) and DFT approximation (crosses) in the rectangle- and sine-windowed cases.

3. ESTIMATION OF THE REMAINING TWO UNKNOWNS

The obtention of $\tilde{\sigma}_k$ greatly facilitates the estimation of x_Δ and x_∞. In fact, (1) becomes a first-order polynomial in $e^{\tilde{\sigma}_k t}$, and thus linear least-squares can be used.

3.1. Standard least-squares approach

The system

$$\begin{aligned} e^{-\tilde{\sigma}_k t_0} x_\Delta + x_\infty &= x_0 \\ e^{\tilde{\sigma}_k T_s} e^{-\tilde{\sigma}_k t_0} x_\Delta + x_\infty &= x_1 \\ e^{\tilde{\sigma}_k 2T_s} e^{-\tilde{\sigma}_k t_0} x_\Delta + x_\infty &= x_2 \\ &\vdots \\ e^{\tilde{\sigma}_k (N-1)T_s} e^{-\tilde{\sigma}_k t_0} x_\Delta + x_\infty &= x_{N-1} \end{aligned}$$

can be written in matrix form,

$$\tilde{\mathbf{\Sigma}}_k \tilde{\mathbf{u}}_k = \mathbf{x}, \tag{20}$$

where

$$\tilde{\mathbf{\Sigma}}_k = \begin{bmatrix} e^{-\tilde{\sigma}_k t_0} & 1 \\ e^{\tilde{\sigma}_k T_s} e^{-\tilde{\sigma}_k t_0} & 1 \\ e^{\tilde{\sigma}_k 2T_s} e^{-\tilde{\sigma}_k t_0} & 1 \\ \vdots & \vdots \\ e^{\tilde{\sigma}_k (N-1)T_s} e^{-\tilde{\sigma}_k t_0} & 1 \end{bmatrix},$$

and

$$\tilde{\mathbf{u}}_k = \begin{bmatrix} \tilde{x}_{\Delta,k} \\ \tilde{x}_{\infty,k} \end{bmatrix}$$

is the vector of unknowns. $\tilde{\mathbf{u}}_k$ can be found in the least-squares sense as

$$\tilde{\mathbf{u}}_k = \left(\tilde{\mathbf{\Sigma}}_k^T \tilde{\mathbf{\Sigma}}_k \right)^{-1} \tilde{\mathbf{\Sigma}}_k^T \mathbf{x}, \tag{21}$$

and from there,

$$\tilde{x}_k(t) = \tilde{x}_{\Delta,k} e^{\tilde{\sigma}_k t} + \tilde{x}_{\infty,k}. \tag{22}$$

Notice the pervasiveness of the subscript k, reminiscent of the fact that this whole system and its solution are dependent on the k-index estimate $\tilde{\sigma}_k$. The next question to approach is what estimate index to use.

3.2. Selection of the estimate index

Figure 2 illustrates the fact that the discrepancy between Fourier series and DFT coefficients increases along with index number k. Hence, without the presence of noise, i.e. when the entries in $\mathbf{x}$ are samples of a pure mathematical function the type of (1), the best available σ estimate is $\tilde{\sigma}_2$, which corresponds to the lowest positive index for which (15), and by extension, (19), hold.

However, noise inherent to the data affects the spectrum. For instance, consider a sequence $\mathbf{x}^\epsilon$, sum of $\mathbf{x}$ and a corrupting error signal $\epsilon = [\epsilon_0, \epsilon_1, \epsilon_2, ..., \epsilon_{N-1}]^T$, i.e. $\mathbf{x}^\epsilon = \mathbf{x} + \epsilon$. Distributivity of the operation of windowing allows us writing

$$\begin{aligned} \tilde{\mathbf{x}}^\epsilon &= \mathbf{x}^\epsilon \cdot \mathbf{s}_N \\ &= \mathbf{x} \cdot \mathbf{s}_N + \epsilon \cdot \mathbf{s}_N \\ &= \tilde{\mathbf{x}} + \tilde{\epsilon}, \end{aligned} \tag{23}$$

where $\mathbf{s}_N = [s_N(0), s_N(1), ..., s_N(N-1)]^T$. This, combined with distributivity of the DFT over addition, leads to the conclusion that

$$\tilde{X}^\epsilon[k] = \tilde{X}[k] + \tilde{E}[k], \tag{24}$$

where $\tilde{E}$ is the transform of $\tilde{\epsilon}$. It follows from (24) that $\tilde{X}^\epsilon_{\mathrm{R}}[k] = \tilde{X}_{\mathrm{R}}[k] + \tilde{E}_{\mathrm{R}}[k]$ and that $\tilde{X}^\epsilon_{\mathrm{I}}[k] = \tilde{X}_{\mathrm{I}}[k] + \tilde{E}_{\mathrm{I}}[k]$. To summarise, we say that the noise corrupts both time- and frequency-domain data in addition.

This fact suggests that there may be an index k for which $\tilde{E}[k]$ is the least, and hence the estimate $\tilde{\sigma}_k$, the best. However, the frequency distribution of $\tilde{E}$ seems hardly predictable. To minimise the problem of noise corruption, frequency-domain fitting of a noise-free Fourier series $\tilde{\mathcal{X}}[k]$ into $\tilde{X}^\epsilon[k]$ was thought of. This, however, might not be easier than the time-domain fitting problem itself. Also, because of aliasing, k-estimates where $\tilde{E}[k]$ is the least may not be the best candidates, especially for values of k nearing $N/2$, where aliasing is strongest.

All those reasons led to the simplistic idea of estimating $\tilde{\sigma}_k$ for all indices up to $K = \lfloor (N-1)/2 \rfloor$, for each k evaluating

$$\tilde{\varepsilon}_k = \sum_{n=0}^{N-1} |x_n - \tilde{x}_k(nT_s + t_0)|, \tag{25}$$

and keeping the final estimate $\tilde{\sigma} = \tilde{\sigma}_k$, for whose index $\tilde{\varepsilon}_k$ is the least.

The drawback of this idea is its relative computational cost, especially as it requires (19) and (21) to be computed $K - 2$ times instead of once. Also, fitting real data using this algorithm showed to bring no significant improvement in comparison with using $\tilde{\sigma}_2$ as a reference estimate. The extra computational cost is measured and expressed in section 4.

4. PERFORMANCE AND RESULTS

This section shows tests of the Fourier-based method introduced in this paper, both in terms of computation time and quality of fit, onto synthetic and real signals. Also, the performance of the `fepcf` and `fepcfk` functions was compared with that of the MATLAB non-linear regression fitting function `nlinfit`. It is understood that, when confronted to the problem of fitting non-linear models into data, researchers may consider using Non-Linear Regression (NLR) before anything else, the method having the major advantage of being applicable to the ensemble of non-linear functions, and hence of sparing the search for a model-specific solution. However, NLR has drawbacks and limitations of its own, that will arise during the course of this section. But first, for clarity, a brief recapitulation of the FEPCF scheme is deemed desirable.

4.1. The FEPCF method in its essential steps

The method presented in this paper was implemented in the MATLAB functions `fepcf(t,x)` (taking $\tilde{\sigma}_2$ as

a final estimate) and `fepcfk(t,x)` (using the index to the least absolute value error $\tilde{\varepsilon}_k$, as explained previously). Both return the three estimates $\tilde{\sigma}$, $\tilde{x}_\Delta$ and $\tilde{x}_\infty$. The `fepcf` scheme, grossly, is as follows :

1. Generate the window $s_{N'}$
2. Evaluate $\tilde{X}'[2]$
3. Evaluate $\tilde{\sigma}_2$ after (19)
4. Construct the matrix $\tilde{\mathbf{\Sigma}}$ and find the vector of unknowns $\tilde{\mathbf{u}}$ after (21).

This list reveals the simplicity of the method and suggests its computational efficiency.

4.2. Tests onto synthetic signals

The synthetic signals used for the purpose of this section were noise-free EPC functions the type of (1).

4.2.1. Quality of the fits

In relation to aliasing, the quality of the fits depends on the number of samples N. Yet it was also realised that, for functions sampled over a period of time T, high ratios σ/T, translative of more angulated curves (Figure 3), impaired the faithfulness of the estimates. The fit quality was therefore tested in terms of both N and σ/T ; the result is shown in the surface plot in Figure 3. The surface matrix of values was obtained by integrating, for each fit, the absolute value of the difference between the function $x(t)$ (solid line in the upper plots) and its estimate $\tilde{x}(t)$ (dashed line) over the interval $[0, T]$. To standardise these measurements, normalisation was sought both horizontally, by scaling the result by $1/T$, and vertically, adjusting x_Δ and x_∞ so that $x(0) = 0$ and $x(T) = 1$. Mathematically formulated, the error surface in Figure 3 reflects

$$\varepsilon = \frac{1}{T}\int_0^T |x(t) - \tilde{x}(t)|dt, \tag{26}$$

approximated numerically with a hundredth-order Riemann sum.

So as to help the reader to associate the error ε to the quality of fit, four points on the surface were chosen for which the corresponding fits are shown in the upper subplots. These points are located on the surface with the diamond, triangle, square and circle shapes. The square and, especially, the circle fits, are visibly poor. This is due to the extremely small DFT sizes (ten or less), as well as to great ratios σ/T. Consequently, the discretised signals are similar to impulse shapes, immune to the anti-aliasing effect of windowing. It must be realised that, in the case of the square fit, where $\sigma = 40/T$ and the sampling period is $T/10$, the time-varying part increases by a factor of $e^4 \approx 55$ from one sample to the next, and in the cases of the circle fit, where $\sigma/T = 100$ and the sampling period is $T/5$, by a factor of e^{20}, i.e. about half a billion.

It was tested whether NLR performed better in such extreme cases. This iterative method needs be given initial, rough estimates for the unknowns. In a first attempt, σ, x_Δ and x_∞ were all initialised to 0. In a second attempt, the estimates returned by the Fourier-based method were used instead. In neither case did the recursion converge.

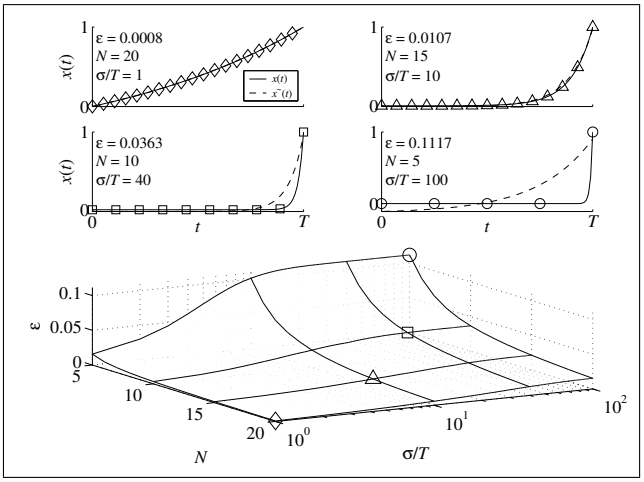

Figure 3. Pushing the method to its limits. Upper plots : $x(t)$ (solid line) and fits (dashed line) across various DFT lengths and growth rates. Lower plot : fit error in function of DFT length and growth rate.

Algorithm	Computation time (s)
`fepcf`	0.0008
`fepcfk`	0.0099
`nlinfit`	0.0113
`nlinfit` (`fepcf`-aided)	0.0029

Table 1. FEPCF and NLR computation time, averaged over 1,600 iterations. Only converging cases in NLR were considered.

4.2.2. Computation time

In this section the computation time of the Fourier-based fitting method is measured, and compared with that of NLR as implemented in MATLAB's `nlinfit` function. The fitting was done on sampled EPC functions – as per section 4.2.1, for combinations of growth rate $\sigma/T = 1, 2, 3, ..., 10$, and sequence length $N = 4, 5, 6, ..., 196$ (for an average length of 100). 1,930 sequences were thus fitted.The test was run on a MacBook Pro portable computer, equipped with a 2.16 GHz Intel Core 2 Duo processor. The computation time results, averaged, and are shown in (Table 1).

In this test FEPCF proves to run over 10 times faster than NLR. It is also seen that the performance of the latter is substantially improved when the Fourier-based estimates are used as initial estimates in the recursion, as the added performance times of `fepcf` and `fecpf`-aided `nlinfit` is only 33% of that of `nlinfit` alone. Also, it must be noted that the computation time of the Fourier-based method is more predictable, due to its closed form, while the non-linear regression computation time depends essentially on the number of iterations. In this regard, not

to give an easy advantage to the Fourier-based method, care was taken to exclude from the measurements signals for which the `nlinfit` function had been found not to converge.

On another matter, the question was raised in section 3.2 as to how much more computation it required to estimate $\tilde{\sigma}_k$ for all indices $k \in [2, \lfloor (N-1)/2 \rfloor$ (approach used in `fepcfk`), instead of $k = 2$ alone (`fepcf`). In terms of computation time, and for a sequence length average of 100 samples, this a tenfold increase.

4.3. Examples of real-signal fits

The motivation for the design of the FEPCF method stems in the finding of a discernible trend in the time-sampled measurements of the inharmonicity coefficient of tension-modulated, steel string guitar tones [4]. An EPC model for the tension-modulated fundamental frequency, simplification of the expression found in [7], proves to enable satisfactory fits, as exemplified shortly. The fundamental frequency model may be formulated as

$$f_0(t) = (f_{0,0} - f_{0,\infty})e^{-t/\tau_{\text{FF}}} + f_{0,\infty}, \qquad (27)$$

where $f_0(t)$ is the time-varying, down-gliding fundamental frequency, $f_{0,0}$, its value at time 0, $f_{0,\infty}$, its value as $t \to \infty$, and τ_{FF}, its *decay time*, i.e. the time interval over which the time-varying part of the signal decays by a factor of e.

The stiffness of certain strings brings an additional restoring force in the transverse vibrations. This has the effect of "stretching" the upper partials in the harmonic series, such that the frequency of the k^{th} harmonic obeys

$$f_k = k f_0(t)\sqrt{1 + \beta(t)k^2}, \qquad (28)$$

where β is the well-known *inharmonicity coefficient* [1], here made time-dependent. The model suggested for $\beta(t)$ in [4] is the inverse of an EPC function, i.e.

$$\beta(t) = \frac{\beta_0\beta_\infty}{(\beta_\infty - \beta_0)e^{-t/\tau_{\text{IC}}} + \beta_\infty}, \qquad (29)$$

where, in analogy to the fundamental frequency model, β_0 is the inharmonicity coefficient valuevalue at time 0, β_∞, its value as $t \to \infty$, and τ_{IC}, the decay time of the time-varying part of the denominator.

When relating these models to EPC functions, it is found that

$$\begin{aligned} x_\Delta &= f_{0,0} - f_{0,\infty}, \\ x_\infty &= f_\infty, \\ \sigma &= -1/\tau_{\text{FF}}, \end{aligned}$$

and

$$\begin{aligned} x_\Delta &= 1/\beta_0 - 1/\beta_\infty, \\ x_\infty &= 1/\beta_\infty, \\ \sigma &= -1/\tau_{\text{IC}}. \end{aligned}$$

Figure 4 shows examples of fundamental frequency and inharmonicity coefficient fits for three contrasting notes, instruments and recording quality. Fundamental frequency and inharmonicity coefficient estimates were obtained with Median-Adjustive Trajectories (MAT), method introduced in [5]. In the upper left-hand side plot, an acoustic guitar E3 (open bass string) sample was produced using Yellow Tools' *Independence Free* sampler [1] . Here the recording quality is optimal, and noise interfering in the measurements can be assumed to be negligible. Also, the lower, open string of the acoustic guitar is most susceptible among instrumental tones of undergoing tension modulation. As a result, both the fundamental frequency and inharmonicity coefficient show well-behaved trends, very successfully fit by the above-described models.

In the upper right-hand side plot of Figure 4, the treble E of a Martin acoustic guitar was recorded by the author in the Music Technology Laboratory in the National University of Ireland, Maynooth. Care was taken to produce the best recording possible, yet the recording featured a background, broadband noise situated approximately at -60 decibels full-scale, which might have impaired the quality of the fundamental frequency and inharmonicity coefficient measurements. Also, the treble E string in an acoustic guitar is considerably more flexible than the bass E, and as a result is less subject to inharmonicity coefficient modulation. The data seen in in this plot still shows an EPC trend in the fundamental frequency, if more noisy than in the Ovation open bass E case, but the inharmonicity coefficient samples are extremely noisy and show little trend. Logically, the parameters returned by the Fourier-based fit may hence not be taken as meaningful.

Finally, the bottom left-hand side plot shows the analysis of a Steinway grand piano middle C tone, downloaded from the University of Iowa Electronic Music Studios website [2] . The fundamental frequency trend is here very noisy, with a burst of noise around $t = 0.5$ caused by an impulse originally in the file, whose broadband, short-term energy is visible in the tone's spectrogram (bottom right-hand side plot). To the credit of FECPF, this unexpected impulse and the disruption it causes in the data did not loose the general trend to the fit. Yet, in the inharmonicity coefficient's case, the method returns a trend in contradiction with the expected stabilising trend. The reader can appreciate, however, that this trend is not obvious from the data points either. In the context of inharmonicity coefficient parameter fitting, it can in this regard be considered to force the growth rate estimate $\tilde{\sigma}$ to a maximum of 0, which corresponds constant inharmonicity.

5. CONCLUSION

The Fourier-based Exponential Plus Constant Fitting method (FEPCF) was exposed largely in its conception and mathematical grounds. A brief recapitulation in section 4.1 of its implementation yet reveals that the method essentially reduces to four steps, and its writing in MATLAB holds in less than one page of code. The method

1 . `http://www.yellowtools.com/` (latest access : February 13th, 2011)

2 . `http://theremin.music.uiowa.edu/MIS.piano.html` (latest access : March 14th, 2011)

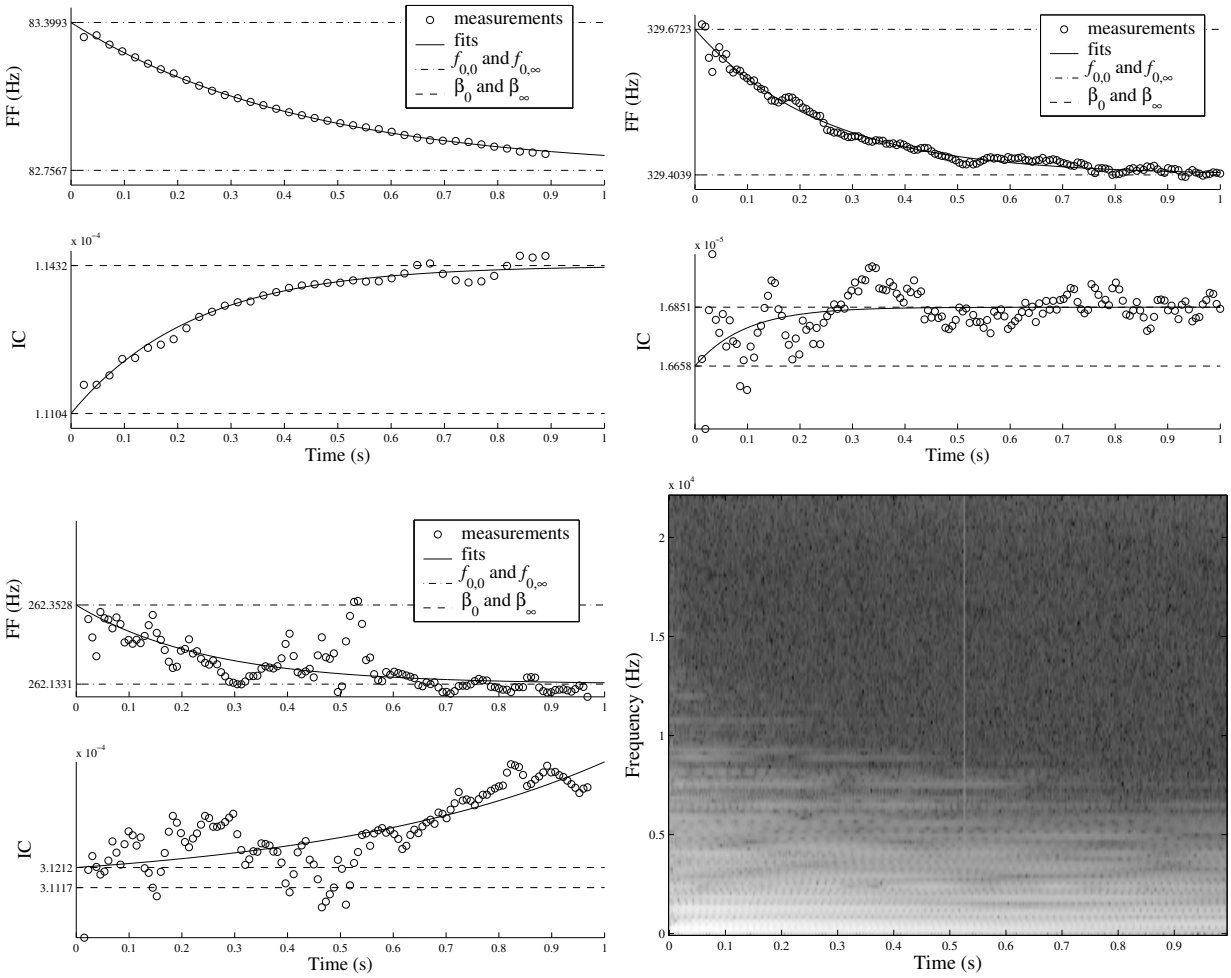

Figure 4. Fundamental Frequency (FF) and Inharmoncity Coefficient (IC) fits for : (top left) an Ovation acoustic guitar bass E ; (top right) a Martin acoustic guitar treble E ; (bottom left) a Steinway grand piano middle C. Spectrogram of the Steinway middle C (bottom right).

has several advantages over generic Non-Linear Regression (NLR) algorithms, such as : being faster by a factor of ten (or more for cases where NLR has difficulties converging) ; featuring a runtime dependent on the length of the analysed sequence only, otherwise constant ; and not requiring initial coefficient estimates. On noise-free, synthetic signals, NLR fitting either returns perfect estimates, in which cases it beats FEPCF in terms of accuracy, or does not converge at all, in which case FEPCF is preferable, as at least it tends towards the ideal solution. This however applies for sequences of very short length and extreme exponent coefficient values only ; elsewhere, FEPCF proves entirely satisfactory. In regard of its performance aspect, the FEPCF method seems especially interesting in the context of real-time applications. Where minimising the square of the error is a priority, results from Table 1 show that FEPCF offers a computationally efficient solution to initialising NLR. To the author, FEPCF stands as a reliable measurement tool for the modeling of modulated fundamental frequency and inharmonicity coefficient.

6. ACKNOWLEDGMENTS

I would like to express my gratitude to Victor Lazzarini for his support, to my Ph.D. supervisor Joseph Timoney for his advice on the writing of this contribution, and to Sylvain Marchand for his constructive feedback.

7. REFERENCES

[1] Fletcher, H., Blackham, E. D. and Stratton, R. "Quality of Piano Tones", *Journal of the Acoustical Society of America*, *34* (6), 749-761, 1962.

[2] Hagen, S. J. "Exponential Decay Kinetics in "Downhill" Protein Folding", *Proteins : Structure, Function and Bioinformatics*, *50*, 1-4, 2003.

[3] Hartman, W. M. *Signals, Sound, and Sensation*. Springer Science+Business Media, New-York, 1998.

[4] Hodgkinson, M., Timoney, J. and Lazzarini, V. "A Model of Partial Tracks for Tension-Modulated Steel-String Guitar Tones", *Proc. of the 13th Int. Conference*

on Digital Audio Effects (DAFx-10), Graz, Austria, September 6-10, 2010.

[5] Hodgkinson, M., Wang, J., Timoney, J. and Lazzarini, V. "Handling Inharmonic Series with Median-Adjustive Trajectories", *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-0)*, Como, Italy, September 1-4, 2009.

[6] Hughson, R. L., Weisiger, K. H. and Swanson, G.D. "Blood lactate concentration increases as a continuous function in progressive exercise", *Journal of Applied Physiology*, *62* (5), 1975-1981, 1987.

[7] Legge, K.A. and Fletcher, N. H. "Nonlinear generation of missing modes on a vibrating string", *Journal of the Acoustical Society of America*, *76* (1), 5-12, 1984.

[8] Nichols, J. J. and King-Smith, P. E. "The Impact of Hydrogel Lens Settling on the Thickness of the Tears and Contact Lens", *Investigative Ophthalmology and Visual Science*, *45* (8), 2549-2554, 2004.

[9] Raichel, D. R. *The Science and Applications of Acoustics*. Springer-Verlag, New-York, 2000.