SHORT-TIME WAVELET ANALYSIS OF ANALYTIC RESIDUALS FOR REAL-TIME SPECTRAL MODELLING

Jeremy J. Wells and Damian T. Murphy

Audio Lab, Intelligent Systems Group, Department of Electronics
University of York, YO10 5DD, UK
{jjw100|dtm3@}ohm.york.ac.uk

ABSTRACT

This paper describes an approach to using compactly supported spline wavelets to model the residual signal in a real-time (frame-by-frame) spectral modelling system. The outputs of the model are time-varying parameters (gain, centre frequency and bandwidth) for filters which can be used in a subtractive resynthesis system.

1. INTRODUCTION

Extraction of the sinusoidal part of an audio signal, leaving a residual, is a common approach to spectral modelling [1]. Whereas the sinusoidal part of the signal consists of long-term, narrow-band components, the residual is comprised of both long- and short-term broad-band components. Systems have been proposed that separate these two types of residual component, classifying them as transients or noise, such as in [2]. A multiresolution approach to residual modelling, such as that offered by wavelets, can enable the good time localisation required for transients, along with the generality required for more stationary components. This paper introduces such an approach.

The analysis system described here was developed for use in a real-time spectral modelling system. In this context real-time means frame-by-frame; decisions about model parameters are made, and time-domain resynthesis is executed, within the current frame to minimise the delay between input and output. Because it produces complex analysis data it is possible to obtain estimates for the centre frequency of components. Also, through use of the wavelet splitting method which is used in wavelet packet decomposition, the bandwidth of components can be estimated. This allows the resynthesis filters to adapt their time-frequency localisation properties to the analysed signal. B-spline wavelets are used since they also offer control over the time-frequency localisation of the analysis filters.

Whilst critically sampled orthogonal wavelets are often useful in situations where sparseness in the analysis data is required, for analysis-modelling-transformation applications over-complete (redundant) wavelet representations are usually more desirable. However such representations come at greater computational cost and highly redundant analysis may well be prohibitively expensive in a real-time application where the analysis-modelling-transformation-resynthesis cycle for a single frame must take less time than for that frame to be played out and the next frame to be acquired. To offer some mediation between cost and redundancy a 'partially decimated' transform is used where the amount of decimation can be controlled by the user (and potentially by the system in response to other processing demands).

Section 2 of this paper provides an introduction to the existing literature that describes B-spline wavelets and their properties. Section 3 gives an overview of the context in which they are used here and the spectral subtraction method used to obtain the residual. Section 4 describes the complex wavelet system employed while Section 5 assesses its cost and proposes partial decimation as a way of offering flexibility in this regard. Section 6 presents a method for estimating the bandwidth of components of the residual. Section 7 briefly describes a context in which the system is employed.

2. B-SPLINE WAVELETS

This section summarises the existing literature on B-splines and their associated wavelets. For further information the reader is directed to the references cited, particularly [3], [4] and [5].

A B-spline ('basis' spline) curve through a set of points consists of the linear combination of shifted B-spline basis functions of a given order. A zeroth order spline curve is constructed from a series of constant functions at the height of each data point. A first order spline curve is constructed from a series of straight lines that join each data point. A second order spline curve is constructed from a series of quadratic functions that span three data points and so on, with each 'piece' of the curve having its own weighting coefficient, meaning that a function can be described by:

$$f(x) = \sum_{k \in Z} c(k) \beta^m (x - k) \tag{1}$$

where β^m is the B-spline curve of order m and $c(k)$ are the weighting coefficients. The zeroth order B-spline is given by

$$\beta^0(x) = \begin{cases} 1, & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 0, & \text{elsewhere} \end{cases} \tag{2}$$

Higher orders are obtained by repeated (m times) convolution of the zeroth order B-spline. For example the cubic (third order) B-spline is obtained by convolution of the zeroth order with itself three times. An order m B-spline can be found directly (without convolution) from:

$$\beta^m(x) = \frac{1}{m!} \sum_{k=0}^{m+1} \binom{m+1}{k} (-1)^k \left(x - k + \frac{m+1}{2} \right)_+^m \tag{3}$$

where

$$(x)_+^m = \begin{cases} x^m, x \geq 0 \\ 0, x < 0 \end{cases} \quad (4)$$

is the one-sided power function [3]. It should be noted that, for high order splines, there are stability problems when using (3) with finite precision. A version of (3) which exploits the symmetry inherent in the repeated convolution of a box function is proposed and used here. This is given by

$$\beta^m(x) = \frac{1}{m!}\sum_{k=0}^{m+1}\binom{m+1}{k}(-1)^k\left(-|x|-k+\frac{m+1}{2}\right)_+^m \quad (5).$$

The constant (zeroth order) B-spline basis leads to a model which is not continuous (since it is piecewise constant). The first order basis offers a continuous (piecewise linear) underlying model but it is not smooth since the first derivative is not continuous. The second order (quadratic) basis is continuous and smooth but its rate of change of curvature (second derivative) changes in a piecewise constant fashion. The third order (cubic) basis exhibits the 'minimum curvature property' since the second derivative is continuous and so for many applications the cubic B-spline is considered the most appropriate underlying continuous piecewise function. However, if better frequency localisation is required (at the cost of poorer time localisation) then the B-spline order can be increased.

For the *m* order B-spline wavelet transform the scaling and wavelet functions and their associated discrete filter sequences are given by

$$\beta^m(x/2) = \sum_{n=-\infty}^{\infty} u_2^m[n]\beta^m(x-n) \quad (6)$$

$$\psi(x/2) = \sum_{n=-\infty}^{\infty} g[n]\beta^m(x-n) \quad (7)$$

where $u_2^m[n]$, the interpolation (approximation) filter, is the binomial kernel of order *m* given by

$$u_2^m[n] = \begin{cases} \frac{1}{2^m}\binom{m+1}{n}, & 0 \leq m \leq n+1 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

and *g*[*n*], the wavelet (detail) filter, is given by

$$g[n] = u_2^m[n] * \left((-1)^m u_2^m[n]\right) * \left((-1)^m b^{2m+1}[n]\right) \quad (9)$$

where b^m is the *m*th order B-spline sampled at the integers [4], [6]. Figure 1 shows the wavelet functions associated with B-splines of order zero (the Haar wavelet), one, three and twenty, at scale one.

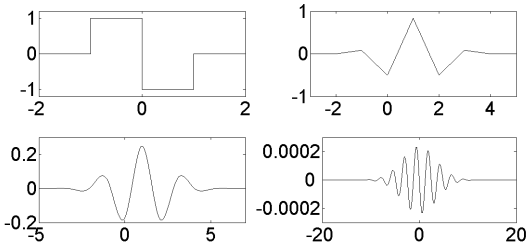

Figure 1: *Underlying wavelet functions for B-splines of different orders. The horizontal axis values are samples.*

As the order of the B-spline increases so the shape of the wavelet function tends to a modulated Gaussian (Gabor function) which has optimum time-frequency localisation properties. For a cubic B-spline it has been shown that the error in approximating a Gabor function is less than 3% and the localisation is within 2% of the optimum [5]. Unfortunately only the zeroth B-spline wavelet transform is energy preserving. At higher orders the wavelet and scaling filters are not the power complements of each other. Figure 2 shows the magnitude responses of these filters for orders zero, one and three. This lack of orthogonality can be overcome by over-sampling in both time (partial or no decimation) and scale domain (parallel transforms of the input at different sample rates) and by taking account of the approximate Gaussian shape of the filters in the Fourier domain. Knowledge of the filter shape, along with estimation of a component's width and centre frequency, allows for magnitude correction of estimates [8].

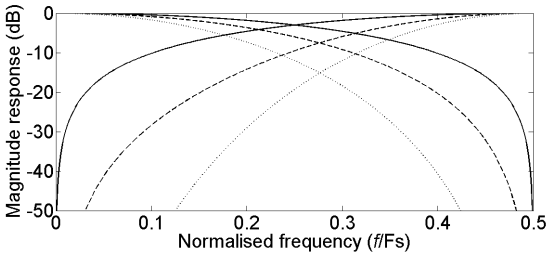

Figure 2: *Magnitude response of wavelet and scaling filters at orders zero (solid line), one (dashed) and three (dotted).*

3. DERIVATION OF THE RESIDUAL

The real-time spectral modelling system in which the analysis method described here is used is discussed in [7] and described in detail in [8]. The system models and synthesizes the sinusoidal part of monophonic audio signals as successive non-overlapping frames with piecewise quadratic phase and piecewise linear amplitude on a frame by frame basis. Only the phase is aligned between frames for continuing sinusoids; discontinuities in frequency and amplitude at synthesis frame boundaries are usually very small due to the high-accuracy analysis method employed.

The Spectral Modelling Synthesis system (SMS, see [1]) uses time domain subtraction to produce the residual; the entire sinusoidal

signal is synthesized and subtracted from the original input signal. An advantage of this approach is that having a time domain representation of the residual means that spectral analysis of it can be performed with optimised parameters, such as a shorter analysis frame, for what is assumed to be a stochastic signal. In a real-time system which produces output from input on a frame by frame basis it is not possible to employ this approach. Unless there is no overlap between frames (only possible with a rectangular window) the synthesis and analysis frames will be of a different length and so short-time time domain subtraction is not available either. For this reason spectral subtraction is employed here to calculate the residual signal. Once this has been performed the data is finally transformed back to the time domain in analytic form after Hilbert transformation in the Fourier domain, ready for the complex B-spline wavelet analysis described in this paper.

An assumption of SMS is "that the residual is fully described by its amplitude and its general frequency characteristics. It is unnecessary to keep either the instantaneous phase or the exact spectral shape information" [9]. Augmentations of the SMS model to include a third signal component type (transients) acknowledge that this assumption is not valid in some cases [2]. Whilst it is the case that for long term stationary noise the phase spectrum does not contain important information the situation for short duration broad band (i.e. impulsive) components is that both the phase and magnitude are needed to retain perceptually relevant fast changing temporal detail. The spectral modelling technique used here for the residual is intended to be capable of capturing the temporal detail of transient components and the spectral resolution of longer term stochastic components. Since both the phase and magnitude of non-sinusoidal components remain intact after spectral subtraction, the inherent timing information contained within these components is passed onto the complex wavelet analysis combining both transient and long term noise in the one model.

Time domain subtraction is a straightforward and, provided the instantaneous frequencies and amplitudes of the sinusoids are well predicted by the model, effective operation. Spectral subtraction is a more complex process since individual sinusoidal components are not represented by individual points in the Fourier domain. Finite length windowing smears components into multiple bins and non-stationarity exacerbates this: frequency change widens the main lobe and amplitude change narrows the main lobe but increases the level of side lobes, increasing the spread of energy to distant bins. A single sinusoid is represented by a single complex number in the Fourier domain only in a very specific situation: a rectangular analysis window is used, the analysed sinusoid has stationary amplitude and frequency and its frequency coincides exactly with the centre of an analysis bin (i.e. the length of the analysis window is an integer multiple of the sinusoidal period).

In [10] a spectral subtraction technique was described which was developed for use in a transform based thresholding process, *Wavethresh*. This technique used knowledge of the window power spectrum to predict the contribution made to adjacent bins made by a stationary sinusoid for a given deviation of the sinusoid's frequency from that of the bin centre. This was necessary since *Wavethresh* used a non zero-padded FFT. This produces large variations in energy localisation around a sinusoidal peak for different deviations of the mean frequency from that of the centre of the analysis bin. For the sinusoidal analysis employed here a zero-padding factor of 8 is used which significantly reduces the variation in energy localisation. In fact the number of bins that require zeroing in order to produce a desired level of attenuation does not change as a function of the distance of a component from the centre frequency of an analysis bin (for example 30 bins either side of, and including, the sinusoidal peak require zero-ing to achieve 48 dB of attenuation for an 8 times zero-padded 1025 sample frame, regardless of frequency).

Since the spectral data is available in zero-padded form there are two approaches that can be taken to obtain a time domain version of the residual: decimation in the frequency domain or in the time domain. Following inverse transformation decimation in time is performed by discarding samples beyond the time support of the analysis window. Since the spectral subtraction process can spread some of the remaining component energy outside the support of the analysis window this also helps to reduce the sinusoidal energy in the residual signal. The disadvantage of not decimating before transformation to the time domain is the increased cost of the IFFT. The time domain decimation method is used here since this greatly simplifies the spectral subtraction process and offers much greater consistency in the relationship between the number of bins that are zeroed and the attenuation of deterministic components.

Non-stationarity must also be accounted for in the spectral subtraction process. Frequency non-stationarity causes a widening of the main lobe but there is little change in the energy contained in distant bins. There is no analytic method for expressing a window's power spectrum where there is frequency non-stationarity. However, the number of bins that need to be zeroed for a given level of attenuation for a particular intra-frame frequency change can be reasonably well modelled by a second order polynomial as shown in Figure 3. This illustrates the number of bins, actual and predicted, that need to be zeroed to produce an attenuation of 48 dB for a given frequency change.

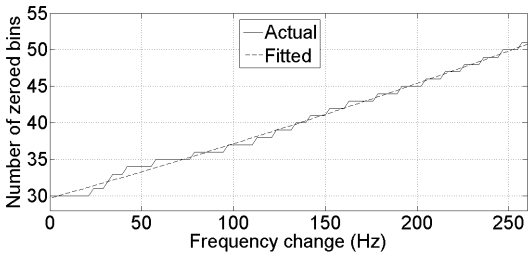

Figure 3: *Number of bins zeroed either side of peak to produce an attenuation of -48 dB for a non-stationary sinusoid versus amount of intra-frame frequency change.*

Amplitude non-stationarity can produce a significant de-localisation in the Fourier domain of a sinusoidal component. This is due to the localisation in the time domain that is produced by the amplitude change; the greater the amplitude change, the more impulse-like the component becomes. The more impulsive a component becomes the less energy it contains compared to a stationary sinusoid with the same peak amplitude. A positive amplitude change localises energy at the end of the frame and negative change localises energy at the beginning of a frame. These are the parts of the frame that experience the greatest attenuation when a window is applied.
The lower energy in a component with non-stationary amplitude combined with the attenuation introduced by the windowing process offsets the energy spreading in the Fourier domain: although zero-

ing a given number of bins produces less attenuation for a component with non stationary amplitude this loss of attenuation is compensated. This is illustrated in Figure 4 which shows the attenuation produced by spectral zeroing of 60 bins and the attenuation produced by the amplitude non-stationarity. It can be seen that the combined attenuation actually falls as the amplitude change increases. For this reason the intra-frame amplitude change for a sinusoidal component is not considered in the spectral subtraction process.

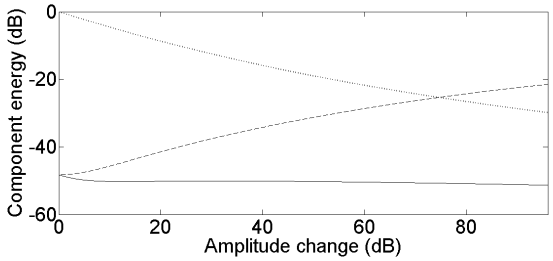

Figure 4: *Maximum component energy for a given amplitude due to intra-frame amplitude change (dotted line), energy reduction due to spectral subtraction (dashed) and combined attenuation (solid)*

4. COMPLEX B-SPLINE WAVELET ANALYSIS

Once the spectrum of the residual has been obtained via the subtraction process described in the previous section its analytic time domain version is computed. First the Hilbert transform is performed in the Fourier domain by

$$X_{\text{analytic}}(k) = \begin{cases} X(k), k = 0, \frac{N}{2} \\ 2X(k), 1 \le k \le \left(\frac{N}{2}\right) - 1 \\ 0, \left(\frac{N}{2}\right) + 1 \le k \le N - 1 \end{cases} \quad (10)$$

where N is the zero-padded transform size. The inverse transform is then computed and the output truncated so that it is the same length as the input frame. Then the B-spline wavelet transform is applied separately to the real and imaginary parts of the analytic signal. The reasons for this approach are twofold. Firstly, a major advantage of the B-spline wavelets is their compact support (both for computational speed and because only short-frames are being analysed). By having two parts of the analytic signal of the same length and analysing these separately with the B-spline filters a saving, in terms of the convolutional demands, is made over analysing the same real signal with two different transforms, one of whose filters would not be compactly supported. Secondly, since the data is already in the Fourier domain the Hilbert transform can be easily implemented at very little additional cost.

When the wavelet transform is considered as a multiresolution analysis (MRA) the sampled sequence which forms the input to the transform is considered to be the approximation of the underlying continuous signal at scale 0. However, this sequence is not the equivalent of projection of the continuous function (in this case band-limited by the anti-aliasing filter) on to the vector subspace that this scale represents. Projection is achieved by convolution of the input with a filter that is the inner product of the sinc function and the dual scaling function (the scaling function itself if the transform is orthogonal rather than biorthogonal) [11]. For the B-spline case this is achieved by convolution of the input with the sampled B-spline of the same order as that of the B-spline transform to be applied [4].

As discussed in Section 2 the B-spline wavelet approximates a Gabor function. The centre frequency of the wavelet is given by

$$f_{\text{centre},k} = \frac{f_0 F_s}{2^{k-1}} \quad (11)$$

where k is the analysis scale and f_0 = 0.4092 [5]. However it has been found here that (11) fails at scale 1 and that correct initialisation is only achieved by multiplication in the Fourier domain of the input with

$$F(\omega) = \text{sinc}^{m+1}\left(\frac{\omega}{2}\right) \quad (12)$$

which is the Fourier transform of the continuous, rather than sampled, B-spline of order m. If (12) is implemented in the Fourier domain then (11) holds at all scales including 1 and since the data is already in the Fourier domain there is no more expense in this filtering operation, despite the far fewer coefficients of the sampled B-spline filter in the time domain. Figure 5 shows the frequency response of the wavelet at scale 1 for both initialisation filters for a cubic B-spline. In this figure the shape given by the filter calculated from (12) is visually indistinguishable from the Gaussian function it approximates.

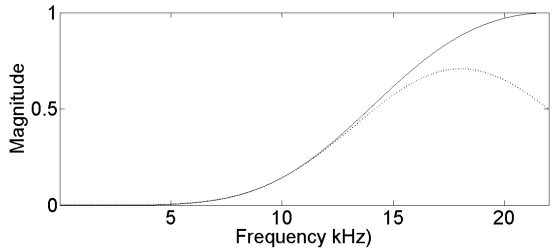

Figure 5: *Normalised magnitude response of the cubic B-spline wavelet at scale 1 for initialisation of the input sequence by the sampled (solid line) and continuous (dotted) cubic B-splines. The sample rate of the input sequence is 44.1 kHz.*

With this improved initialisation a multiresolution analysis is achieved which is akin to an atomic decomposition with Gabor functions which are successively dilated by a factor of 2. Since the critically sampled decomposition of Mallat is achieved by decimation of coefficients at each scale, aliasing is present and the transform is shift-variant [12]. A shift-invariant, non-aliased alternative to the decimated transform is the 'algorithme à trous' (algorithm with holes). This algorithm achieves dilation of the underlying wavelet and scaling functions by inserting a zero (placing a hole) in between each sample of the filters, rather than by decimating their outputs, at each successive scale [13]. Figure 6 shows the time domain, and Figure 7 the Fourier domain, shape of the wavelet filters' impulse responses at the first five analysis scales for a 1025 sample frame. Some spreading of the response in Figure 6 can be observed

at scale 1, this is an unavoidable artefact of the Hilbert transform due to its inherent band-limiting of the signal.

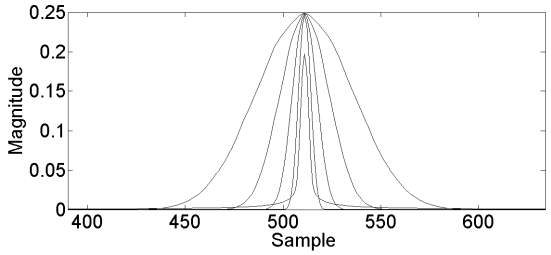

Figure 6: *The undecimated time-domain magnitude response of the complex cubic B-spline wavelet for an impulse at the centre of a 1025 sample frame. The responses widen with increasing scale.*

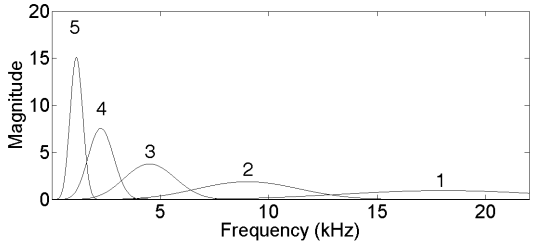

Figure 7: *Magnitude responses derived from Figure 6. The sample rate for this and subsequent plots is 44.1 kHz.*

As is the case for the DFT, the mean instantaneous frequency of a spectral component can be estimated using the complex wavelet transform, particularly since the wavelet used so closely approximates a windowed sinusoid. Reassignment is used for frequency estimation for the prior sinusoidal analysis in this system since an estimate can be obtained from a single analysis frame [10]. Unlike the STFT the wavelet transform provides more than one coefficient at each scale (apart from the highest scale of a critically sampled wavelet transform). This implies that the mean instantaneous frequency can be estimated from the first order difference of the phase between consecutive coefficients in a given scale within a single frame:

$$\overline{f}_{k,n,n+1} = \left(\phi_{\text{detail } k,n+1} - \phi_{\text{detail } k,n}\right)\frac{F_s}{2\pi} \qquad (13)$$

for an undecimated transform, where n is the index at the scale k, and ϕ is the phase, of the coefficients, and

$$\overline{f}_{k,n,n+1} = \left(\phi_{\text{detail } k,n+1} - \phi_{\text{detail } k,n}\right)\frac{F_s}{2^k\pi} \qquad (14)$$

for a decimated transform. The kth power of 2 in (14) is present since the temporal distance between indices is doubled for each increment in scale. Whilst (14) is effective for the lower half of the frequency band occupied by each scale, a correction must be applied to prevent negative frequency estimates in the upper half:

$$\overline{f}_{\text{corrected}} = \begin{cases} \dfrac{F_s}{2^k} + \overline{f}_{\text{estimated}}, & \overline{f}_{\text{estimated}} < 0 \\ \overline{f}_{\text{estimated}}, & \overline{f}_{\text{estimated}} \geq 0 \end{cases} \qquad (15)$$

An additional problem when using the decimated transform is aliasing caused by high energy in nearby out-of-scale components. A straightforward solution to this problem is to not decimate the output detail coefficients at each scale. Whilst this doubles the number of coefficients produced at each scale it does not increase the computational burden since the detail coefficients are not used in further iterations of the decimated algorithm, it is only the approximation coefficients that are used recursively. This prevents aliasing at scale 1 however aliasing still occurs at higher scales since the number of detail coefficients at each scale is reduced by decimation of the approximation coefficients at the previous scale. The ideal solution is to use the undecimated transform however this comes at a significantly increased cost than its decimated counterpart.

Circular convolution is not desirable for time-scale analysis since the purpose is to describe where events occur in (linear) time. In this analysis system the synthesis frame width is determined by the frame overlap. If frames overlap then encroachments due to circular convolution near frame boundaries can be ignored; the greater the overlap factor, the more samples that can be ignored near the boundaries. For example, with an overlap factor of 4 and a frame size of 1025 the wavelet coefficients of concern correspond to the middle 257 samples of the frame. However where circular convolution is employed a component is likely at higher scales, where the filter response is longer, to wrap into the region of interest. Therefore, for short-time wavelet analysis, circular time within frames, as opposed to linear time, makes matching of components at synthesis frame boundaries difficult. Although linear convolution is more expensive than its circular counterpart, since it increases the length of the output of each scale and, therefore, the input to the next scale, it is better suited to this application.

5. TRANSFORM COST AND PARTIAL DECIMATION

The costs of the decimated and undecimated transforms are now considered in terms of the number of multiply and add operations for the linear convolution case. For the mth order spline wavelet the length of the low and high pass filters in samples, at scale 1 are given by:

$$L_{\text{LPF}} = m + 2 \qquad (16)$$

$$L_{\text{HPF}} = 3m + 2 \qquad (17)$$

When a sequence of length S is convolved with a filter of length L the length of the output is $S + L - 1$. For the undecimated transform the input sequence at one scale is the approximation of the previous scale which is achieved by convolution with the dilated low pass filter. Therefore, for the undecimated transform, the sequence length before low pass filtering at scale k is given by:

$$N_k = \left(N + \left(L_{LPF} - 1\right)\sum_{n=1}^{k-1} 2^{n-1}\right) = N + \left(L_{LPF} - 1\right)\left(2^{k-1} - 1\right) \qquad (18)$$

where N is the analysis frame length. This gives a total cost for the transform of:

$$C = \left(L_{LPF} + L_{HPF}\right)\sum_{k=1}^{K} N_k \qquad (19)$$
$$= \left(L_{LPF} + L_{HPF}\right)\left(NK + \left((L_{LPF} - 1)(2^K - 1 - K)\right)\right)$$

where K is the total number of scales. For the decimated transform the filter output is decimated at each scale and so the sequence length at scale k, before filtering and decimation, is given by:

$$N_k = \left\lceil N2^{-(k-1)} + (L_{LPF} - 1)\left(1 - 2^{-(k-1)}\right) \right\rceil \\ = \left\lceil 2^{-(k-1)}(N - L_{LPF} + 1) + L_{LPF} - 1 \right\rceil \quad (20)$$

Allowing for rounding up of numbers of coefficients when an odd length sequence is decimated the approximate total cost is given by:

$$C = (L_{LPF} + L_{HPF}) \sum_{k=1}^{K} N_k \\ \approx (L_{LPF} + L_{HPF})\left(K(L_{LPF} - 1) + (N - L_{LPF} + 1)\left(2 - 2^{-(K-1)}\right)\right) \quad (21)$$

In order to offer some mediation between these two extremes the partially decimated wavelet transform is proposed here. The principle is straightforward: the algorithm begins by filtering the signal and inserting holes into the filter until a given decomposition level (scale) is reached, at which point the filter remains the same and the output is decimated for subsequent iterations. The only other wavelet analysis that combines decimated and undecimated transforms in this way is the over complete DWT (OCDWT) described in [14]. However, this system begins with decimation and then at higher scales switches to filter dilation. This order is reversed in the system proposed here since this reduces shift variance at all scales.

Equations (18) – (21) can be combined to calculate the cost of the partially decimated transform. The cost of calculating the undecimated scale coefficients can be calculated directly from (19) where N is the length of the input sequence and $K = U$ is the number of undecimated scales. The cost of calculating the subsequent decimated coefficients is given by a modified version of (21):

$$C = (L_{LPF} + L_{HPF}) \sum_{d=1}^{D} N_d \\ = (L_{LPF} + L_{HPF})\left(D(L_{dLPF} - 1) + (N_{undec} - L_{dLPF} + 1)\left(2 - 2^{-(D-1)}\right)\right) \quad (22)$$

where D is the number of decimated scales and N_{undec} is the length of the final approximation sequence output from the undecimated part of the transform, given by (18) where $k = U + 1$, which is then halved (since this sequence is decimated before the next filtering stage). L_{dLPF} is the length of the dilated LPF and is given by

$$L_{dLPF} = (L_{LPF} - 1)\, 2^{U-1} + 1 \quad (23)$$

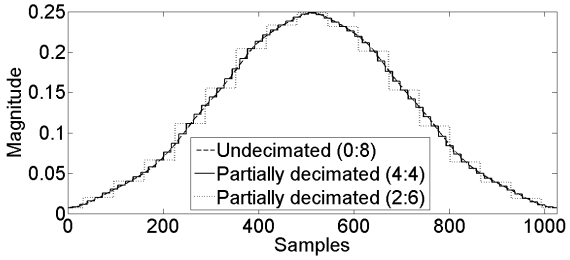

Figure 8: *Time-domain magnitude response at scale 8 of the complex cubic B-spline wavelet transform for differing amounts of decimation.*

Figure 8 shows the time domain magnitude response at scale 8 for an impulse in the centre of the analysis frame for different ratios of numbers of decimated to undecimated scales.

6. SPLIT WAVELETS FOR BANDWIDTH ESTIMATION

The frequency-splitting 'trick' described in [15], and used to produce the full binary tree decomposition used in wavelet packets, is used here to produce estimates of the bandwidth of components at each scale. At one extreme, the instantaneous mean centre frequencies of the scale filter and the two split filters will coincide for an impulse in the frequency domain and, at the other, their centre frequencies will be the same as those of the fixed filters for an impulse in the time domain. Therefore the proximity of the derived centre frequencies for the two complex split filters can be used to estimate the width of the underlying component.

For the undecimated transform the split at each scale is achieved by filtering of the detail coefficients at that scale. The filters are obtained by dilation by a factor of two of the high and low pass filters used to derive the approximation and detail coefficients. For the decimated transform the split can be achieved by convolution of the decimated detail coefficients with the existing filters. However this would produce fewer split than scale coefficients meaning that there could not be a one-to-one mapping of a scale coefficient to its lower and upper split coefficients. Therefore, in the split implementation described here, the filters are dilated and the scale coefficients left undecimated whether the split is occurring for a decimated or undecimated scale in the partially decimated transform. It should be noted that where splitting *is* performed then not decimating the detail coefficients at each scale (discussed earlier as a method of reducing aliasing) will increase the computational cost.

Splitting at a given scale is achieved by convolution of the detail signal with the low and high pass wavelet filters dilated by a factor of two from those used to generate the approximation and detail coefficients at that scale. Dilation of a filter's impulse response in the time domain is equivalent to an equal contraction of its response in the frequency domain. Therefore the frequency responses of the split wavelets' filters are given by:

$$\Psi_{\text{lower}}(\omega) = \Psi_{\text{scale}}(\omega)\, HPF_{\text{scale}}(2\omega) \quad (24)$$

$$\Psi_{\text{upper}}(\omega) = \Psi_{\text{scale}}(\omega)\, LPF_{\text{scale}}(2\omega) \quad (25)$$

where Ψ and XPF are the Fourier transforms of the various filters. The perhaps counter-intuitive result that the upper split wavelet is produced by convolution with the LPF and the lower split by convolution with the HPF is explained by the fact that it is the aliased (reflected) parts of the filters' frequency responses (which are contracted by a factor of 2 in the above equations) that coincide with the region where the response of the wavelet filter is greatest. The upper part of Figure 9 shows the magnitude frequency responses of the wavelet and its upper and lower splits at scale 1. The lower part of this figure shows the shape of the underlying continuous functions. As would be expected of the dilation and convolution operations of the splitting operations, the split wavelets have greater time support but are more localised in frequency than the parent wavelet.

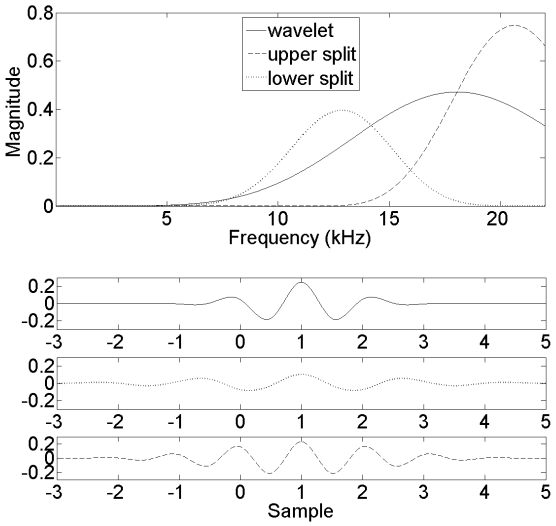

Figure 9: *Magnitude frequency response (top) and time domain shape of cubic B-spline wavelet and its splits at scale 1.*

The centre frequencies of the split wavelets at each scale are given by (11) where f_0 = 0.2919 for the lower and 0.4678 for the upper splits respectively [8]. Therefore the maximum difference (i.e. that due to an impulse) between split filters at scale *k* is given by:

$$\Delta f = \frac{0.1760 F_s}{2^{k-1}} \tag{26}$$

Figure 10 illustrates how differences between frequency estimates at a single scale occur where a component has spectral breadth. The frequency estimates at scale 1 for the wavelet and its splits are shown for a sinusoid and for a single impulse which occurs in the middle of the frame (sample 513). There is a clearly visible difference in estimates for the impulse whereas, at the same scaling of the vertical axis, there is no difference in estimates for a stationary sinusoid.

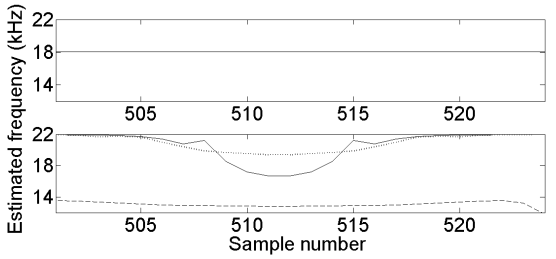

Figure 10: *Frequency estimates for a sinusoid (top) and an impulse at the centre of the frame (bottom).*

The cost of the split transform is the cost of the non-split transform, given by equations (18) to (23), plus the cost of filtering that produces the splits at each scale. The split at each scale is achieved by high pass filtering of the detail coefficients at that scale followed by high and low pass filtering with filters which are dilated by a factor of two from those used to produce the approximations and details at that scale. For the undecimated transform the sequence length, Ns_k (the s indicates 'split'), before the high and low pass split filtering is given by:

$$Ns_k = \left(N + (L_{LPF} - 1) \sum_{n=1}^{k-1} 2^{n-1} \right) = N_k + 2^k (L_{HPF} - 1) \tag{27}$$

and so the combined cost of the all the splitting stages is given by:

$$\begin{aligned} Cs &= (L_{LPF} + L_{HPF}) \sum_{k=1}^{K} Ns_j = (L_{LPF} + L_{HPF}) \sum_{k=1}^{K} N_j + 2^{k-1} (L_{HPF} - 1) \\ &= (L_{LPF} + L_{HPF}) \left(NK + ((L_{LPF} - 1)(2^k - 1 - K)) + (L_{HPF} - 1) \sum_{k=1}^{K} 2^{k-1} \right) \\ &= (L_{LPF} + L_{HPF}) \left(NK + ((L_{LPF} - 1)(2^K - 1 - K)) + (L_{HPF} - 1)(2^K - 1) \right) \end{aligned} \tag{28}$$

and the total cost of the transform is given by adding (19) and (28). For the decimated transform the sequence prior to splitting is the detail sequence at that scale. This is given by

$$Ns_k = \left\lceil 2^{-(k-1)} (N - L_{LPF} + 1) + L_{LPF} + L_{HPF} - 2 \right\rceil \tag{29}$$

and the total cost of the splitting stage is given by adapting (21):

$$C = (L_{LPF} + L_{HPF}) \left(K (L_{LPF} - 1) + (N - L_{LPF} + 1)(2 - 2^{-(K-1)}) + J (L_{HPF} - 1) \right) \tag{30}$$

The total cost of the split decimated transform is given by adding (21) and (30). The cost of the splitting stage of the partially decimated transform can be calculated for the undecimated levels by the same sum. The cost of splitting at the decimated levels is given by a modification of (30)

$$\begin{aligned} C &= (L_{LPF} + L_{HPF}) \\ &\times \left(K (L_{dLPF} - 1) + (N_{undec} - L_{dLPF} + 1)(2 - 2^{-(K-1)}) + K (L_{dHPF} - 1) \right) \end{aligned} \tag{31}$$

where L_{dHPF} is given by

$$L_{dHPF} = (L_{HPF} - 1) 2^{U-1} + 1 \tag{32}$$

Finally, the total cost of the partially decimated split wavelet transform is given by adding (22) and (32). Figure 11 shows the computational cost of the split and un-split, decimated and undecimated complex cubic spline wavelet transforms for linear convolution. For comparison the cost of a 1024 and 8192 point FFT are also shown.

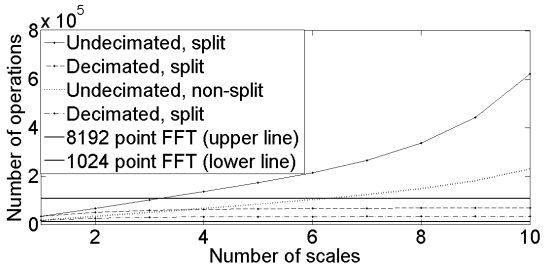

Figure 11: *Number of complex multiply and odd operations required for various complex cubic B-spline wavelet transforms of a 1025 sample frame.*

7. APPLICATION

The complex wavelet analysis system described is able to adapt to different types of input component. The frame-by-frame spectral modelling system in which it is used employs biquadratic parametric equalisers applied to a white noise source for resynthesis of the residual. Two examples are now given which demonstrate how this system performs on different types of input signal. The top part pf Figure 12 shows a resynthesized sequence of unity impulses. In this case the time localisation is good, with energy focussed in a small number of samples. At the other extreme the bottom part shows the time domain input, output and magnitude frequency response of a stationary sinusoid. Although such a component is unlikely to form part of the residual, it demonstrates the ability of the resynthesis filters to adapt their bandwidth to give good frequency localisation and to shift their centre frequency to that of the input component. This time-frequency adaptation is made possible by the bandwidth estimation described in this section.

Figure 13 demonstrates how the residual synthesis can adapt in a single frame. The time localisation at the onset is good but this changes to good frequency localisation later on in the frame (the analysis overlap factor is 2 so the synthesis frame is half the size of the analysis frame). During the last half of the frame the sinsusoidal oscillator ramps on exponentially, 'taking over' from the residual synthesis by the next frame.

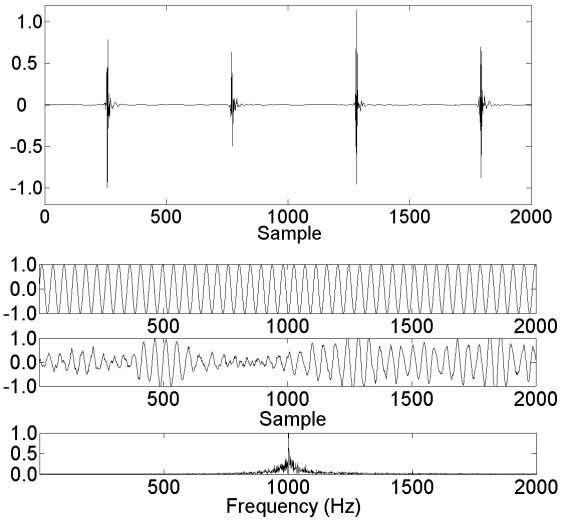

Figure 12: *Residual resynthesis of time domain (top) and frequency domain impulses (bottom).*

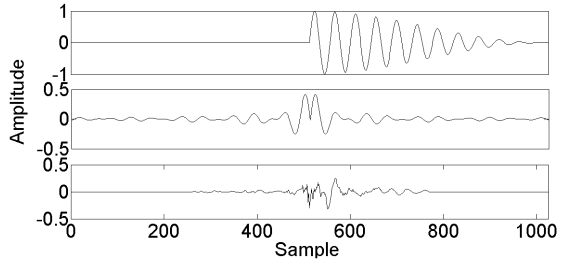

Figure 13: *Windowed sinusoid with sudden onset (top), residual after spectral subtraction (middle) and resynthesized residual (bottom).*

8. SUMMARY

A multiresolution analysis system which produces estimates of magnitude, mean instantaneous frequency and bandwidth of components, and is suited to a residual modelling system has been presented and placed in the context of a frame-by-frame spectral modelling system. The properties of the wavelet analysis, its cost, and partial decimation as a means of negotiating between computational cost and shift-variance\aliasing have been described. Further work will look at how the synthesis and analysis filters can be better matched whilst retaining the simplicity of the resynthesis method. A more detailed treatment and analysis of the work presented here (including measures of aliasing and shift-invariance for different levels of partial decimation) can be found in [8].

9. REFERENCES

[1] X.Serra, "A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic plus Stochastic Decomposition", PhD thesis, Stanford University, USA, 1989.

[2] T. Verma and T. Meng, "An Analysis\Synthesis Tool for Transient Signals", *Computer Music Journal*, vol. 24, pp. 47-59, Summer 2000.

[3] M. Unser, "Splines, A Perfect Fit for Signal and Image Processing", *IEEE Signal Processing Magazine*, pp. 22-38, November 1999.

[4] M. Unser et al, "A Family of Polynomial Spline Wavelet Transforms", *Signal Processing*, vol. 30, pp. 141-162, January 1993.

[5] M. Unser et al, "On the Asymptotic Convergence of B-Spline Wavelets to Gabor Functions", *IEEE Trans. on Information Theory*, vol. 38, pp. 864-872, March 1992.

[6] C. Chui and J. Wang, "On Compactly Supported Spline Wavelets and a Duality Principle", *Trans. of the American Mathematical Society*, vol. 330, pp. 903-915, April 1992.

[7] J. Wells and D. Murphy, "High-Accuracy Frame-by-Frame Non-Stationary Sinusoidal Modelling", *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, pp.253-258, 2006.

[8] J. Wells, "Real-Time Spectral Modelling of Audio for Creative Sound Transformation", PhD Thesis, University of York, January 2006, Available at http://www.jezwells.org

[9] X. Amatriain, "Spectral Processing", chapter in (ed. Zölzer), *DAFX – Digital Audio Effects*, John Wiley, Chichester, 2002.

[10] J. Wells and D. Murphy, "Real-Time Spectral Expansion for Creative and Remedial Sound Transformation", *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-03)*, pp. 61-64, 2003.

[11] P. Abry and P. Flandrin, "On the Initialization of the Discrete Wavelet Transform", *IEEE Signal Processing Letters*, vol. 1, pp. 32-34, February 1994.

[12] S. Mallat, "A Wavelet Tour of Signal Processing", Academic Press, San Diego, 1999.

[13] M. Shensa, "The Discrete Wavelet Transform: Wedding the À Trous and Mallat Algorithms", *IEEE Trans. on Sig. Proc.*, vol. 40, pp. 2464-2482, October 1992.

[14] A. Bradley, "Shift-Invariance in the Discrete Wavelet Transform", *Proc. of the 7th Conf. on Digital Image Computing: Techniques and Applications*, pp. 29-38, December 2003.

[15] I. Daubechies, "Ten Lectures on Wavelets", Society for Industrial and Applied Mathematics, Philadelphia, 1992.