DIGITAL AUDIO EFFECTS IN THE WAVELET DOMAIN

David Darlington, Laurent Daudet and Mark Sandler

Department of Electronic Engineering, Queen Mary, University of London,
Mile End Road, London E1 4NS, United Kingdom.
Tel: +44 20 7882 7986; Fax: +44 20 7882 7997
{david.darlington, laurent.daudet, mark.sandler}@elec.qmul.ac.uk

ABSTRACT

Audio signals are often stored or transmitted in a compressed representation, which can pose a problem if there is a requirement to perform signal processing; it is likely it will be necessary to convert the signal back to a time domain representation, process, and then re-transform. This is time-consuming and computationally intensive; it is potentially more efficient to apply signal processing while the signal remains in the transform domain. We have implemented a scheme whereby linear processing of the traditional type often instinctively understood by those working in the audio field may be applied to signals stored in a wavelet domain representation. Results are presented which demonstrate that the method produces the same output – to within the limits of machine precision – as time-domain processing, for less computational effort than would be required for the full explicit process through the time domain and back again. The potential benefits for linear effects processing (for example, EQ and sample-level delays and echoes) and also for non-linear processing such as dynamics processing, will be introduced and discussed.

1. INTRODUCTION

In many audio applications, some form of processing (linear such as high pass filtering, or non-linear such as de-noising) may have to be performed on the signal, and it has been established [1] that there are *potential* efficiency gains if such processing can be performed in the transform domain. Wavelet methods have already proven popular for non-linear applications such as de-noising. Here we deploy them in a new paradigm for linear signal processing (such as, in the first instance, FIR filtering or simple delays) with the aim of being operable in real-time on signals which are already available *only* in the wavelet domain.

Other researchers [2][3][4] have also demonstrated the possibilities afforded to audio signal processing through consideration of wavelet domain analysis or processing. Our particular objective is the implementation of a signal processing system based on the wavelet model, where the processing applied is *the same* as would be applied to a time-domain audio signal. A major requirement is that a wavelet representation of the signal be available or derivable at every sample instant [5]. In a naive implementation, this would necessitate performing a wavelet transform on some segment of the signal for each sample; a crude and inefficient approach. What has been implemented is a means by which a wavelet representation of all potential shifted versions of any signal may be obtained without such excessive computation or storage requirements. We discuss two distinct formulations of our approach. in order to establish whether it is appropriate to process in the transform domain. Our first formulation is based on the work of Liang and Parks [6], which enables wavelet representations of shifted versions of a signal to be generated from a more extensive transform of the unshifted original. Our alternative approach recasts the problem as one of linear algebra.

2. LINEAR PROCESSING IN THE WAVELET DOMAIN

The proposed technique is applied to signal blocks of fixed length N, and is formulated as a block convolution. Assume a discretized signal x (of length N), convolved with FIR filter h (impulse response of length M) to produce output signal y (of length N+M-2). Thus:

$$x = [x(0), x(1), ..., x(N-1)]^T \text{ ;}$$
$$h = [h(0), h(1), ..., h(M-1)]^T \text{ ;}$$
$$y = [y(0), y(1), ..., y(N+M-2)]^T \text{ ;}$$

and $$y = x * h = h * x$$

$$y = X.h = \begin{bmatrix} x(0) & 0 & 0 \\ x(1) & x(0) & 0 \\ x(2) & x(1) & x(0) \\ x(3) & x(2) & x(1) \\ 0 & x(3) & x(2) \\ 0 & 0 & x(3) \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ h(2) \end{bmatrix}$$

$$= \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ 0 \\ 0 \end{bmatrix} h(0) + \begin{bmatrix} 0 \\ x(0) \\ x(1) \\ x(2) \\ x(3) \\ 0 \end{bmatrix} h(1) + \begin{bmatrix} 0 \\ 0 \\ x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} h(2)$$

$$= x(n).h(0) + x(n-1).h(1) + x(n-2).h(2) \qquad \textbf{(1)}$$

where $x(n-i)$ is a shifted version of $x(n)$ and X is the MxK matrix whose columns are shifted versions of the signal vector. The wavelet transform may be represented as a matrix operation. If the wavelet transform of x is

$$W_x = W^N.x \quad \textbf{(2)}$$

where W^N is an $N\times N$ matrix, and W_y is similarly defined, though of dimension $L\times L$ where $L=(N+M-2)$, then we may write

$$W_y = W^L.(x*h) \quad \textbf{(3)}$$

Because the wavelet transform is linear, from (3) we obtain

$$W_y = W^L.y = W^L.X.h \quad \textbf{(4)}$$

which can be rewritten as

$$\begin{aligned} W^L.y &= W^L\{x(n).h(0) + x(n-1).h(1) + ... + x(n-(M-1)).h(M-1) \\ &= h(0).\{W^L.x(n)\} + h(1).\{W^L.x(n-1)\} + ... \\ &\quad + h(M-1)\{W^L.x(n-(M-1))\} \end{aligned} \quad \textbf{(5)}$$

Thus the wavelet transform of the convolution of x and h can be computed by an appropriately weighted sum of the wavelet transforms of shifted versions of x. To implement this in practice requires a wavelet transform domain representation of the vector x at each time sample, and thus an $N\times N$ matrix of time-shifted wavelet domain data is required, compared with the length N vector of data needed for time domain processing. Increases in N and/or M exacerbate the impact of this problem. Thus it is necessary to obtain a wavelet transform domain representation of a shifted version of the input signal x without resort to explicit transform of signal blocks at each time sample.

3. WAVELET TABLE IMPLEMENTATION

3.1. Theory

Wavelet coefficients might be considered analogous to discrete samples of some periodic function within each decomposition level, each shift of the time-domain signal similarly shifting the sampling instants of this "wavelet coefficient function". Liang and Parks [6] took advantage of this to exploit redundancy in wavelet transformation. Based on mathematical work by Beylkin [7], the process has two steps: firstly, obtain the wavelet coefficients for all shifted versions of the signal, and secondly, select those wavelet coefficients which best characterise the signal according to some selected cost function. Liang and Parks report that it is possible to obtain the wavelet transforms of all circularly shifted versions of an N-point signal without having to explicitly transform a set of compound unit-shifted frames. The complexity is of order $N\log(N)$ compared to order N for the single frame wavelet transform and N^2 where the transform is explicitly determined for each shifted version of the frame.

At each decomposition stage, the input is passed through high- and low-pass filters, and then the output is downsampled by a factor of two. The outputs of the high-pass filter are called the differences d^j and the outputs of the low pass filter the averages s^j. The averages are input to the next stage of the filterbank. The operator of downsampling by two is a linear operator of period-two, which means that if the input is shifted by $2k$, $k \in Z$, the coefficients of the output will be shifted by k samples. Therefore the output at the first stage for even shifts is obtained simply by shifting the output for the original input, and the output for the odd shifts is obtained by shifting the output for the shift-by-one version of the original input. At each stage, therefore, it is only required to calculate the output for two shifts: the original input and its shift-by-one. The coefficients of the differences and averages are obtained by

$$s_k^j = \sum_{n=0}^{n=L-1} h_n s_{n+2k}^{j-1} \quad \textbf{(6)}$$

$$d_k^j = \sum_{n=0}^{n=L-1} g_n s_{n+2k}^{j-1} \quad \textbf{(7)}$$

We compute on each scale j, $(1 \le j \le L)$, 2^j vectors of differences and 2^j vectors of averages. If we denote the current scale as 0 and the next scale as 1, then

$$s_k^j(0) = \sum_{n=0}^{n=L-1} h_n s_{n+2k}^{j-1} \quad \textbf{(8)}$$

$$s_k^j(1) = \sum_{n=0}^{n=L-1} h_n s_{n+2k+1}^{j-1} \quad \textbf{(9)}$$

$$d_k^j(0) = \sum_{n=0}^{n=L-1} g_n s_{n+2k}^{j-1} \quad \textbf{(10)}$$

$$d_k^j(1) = \sum_{n=0}^{n=L-1} g_n s_{n+2k+1}^{j-1} \quad \textbf{(11)}$$

and stepping between the scales we double the number of vectors of averages and differences while at the same time halving the length of each of them. For all scales j, $(1 \le j \le L)$,and shifts i, $(1 \le i \le N-1)$, tables are computed of size jx$(i+1)$ which give direct access to the required coefficients.

N.B. The work of Cohen, Malah and Raz [8] is similar. Utilising the wavelet packet decomposition (WPD) approach, they used a cost function to estimate the best wavelet basis within each level and at each node of the decomposition. The resultant transformation has been proven to be shift-invariant; thus the process and decision are similar to those of Liang and Parks, arrived at via a different approach. Cohen *et al*'s SIWPD (shift-invariant wavelet packet decomposition) does not perform the automatic downsampling of the standard WPD, but selects those coefficients which minimise some chosen cost function. That decisions are taken within each decomposition level as to whether or not to apply a unit shift prior to transformation suggests, though, that the decision trees and wavelet bases selected are unique to each frame and highly dependent on the status of each signal component within the given frame.

3.2. Implementation

The wavelet transform of the signal is available in matrix form as a look-up table of order N $(j+1)$, in what might be termed a "wavelet table". The advantage of this approach is that the wavelet coefficients of *all* shifted versions of a frame are obtainable for an increase in required information - effectively, bandwidth - of order $j+1$ (in the examples presented in this paper, $j+1=7$).

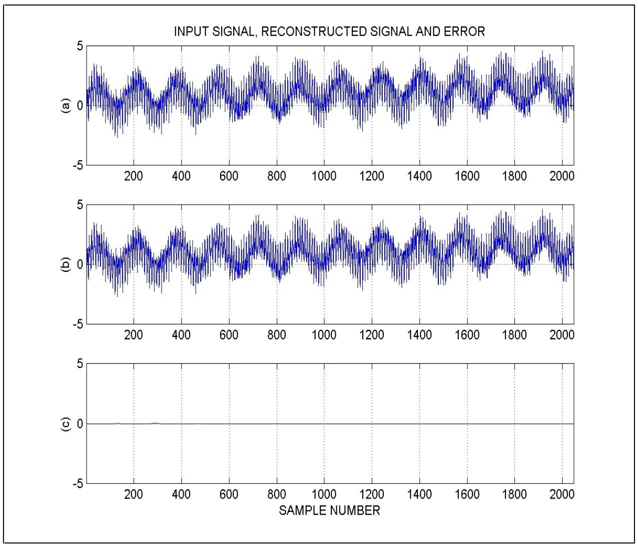

Figure 1: Reconstructed signal compared with original

Figure 1 shows a 2048-point signal, formed of a superposition of four sinusoids and white noise. The 'reconstructed signal' of this figure is that of a signal transformed to and from the wavelet table domain with signal blocks of size 256, using Daubechies-4 wavelet filters and decomposing through 6 levels. The resultant errors are of the order of machine precision.

In Figure 2 we see the results when a simple 3-tap low-pass FIR filter is applied to the signal while it is in the wavelet table domain, compared to the FIR filtering of the time domain signal. One of the potential problems of the implementation is illustrated – a discontinuity across boundaries between signal blocks is introduced, in that it is only after a block boundary that the 'shifted' wavelet coefficients are updated to utilize the data from the newer block. Such a discontinuity means that boundary (block-end) artefacts are anticipated.

Thus the wavelet-domain filtered signal matches the time-domain filtered signal closely, except near block boundaries where the "block-end" effects become significant. The weakness in this initial approach lay in the assumption that a linear shift along a signal could be modelled using a circular shift approach. This is a feature of the "wavelet table" transform itself - Liang and Parks' work is based upon a circulant shift of the signals, and the circulant shift in *itself* causes the algorithm to behave in an unpredictable manner. To compensate for this, we modified the approach using results from [9], in which an improved wavelet packet transformation of audio signals was effected by utilising samples from preceding blocks in order to eliminate block-end artifacts – a lapped wavelet transform. We doubled the length of the wavelet transform block, while maintaining the "reconstruction" block at the same size.

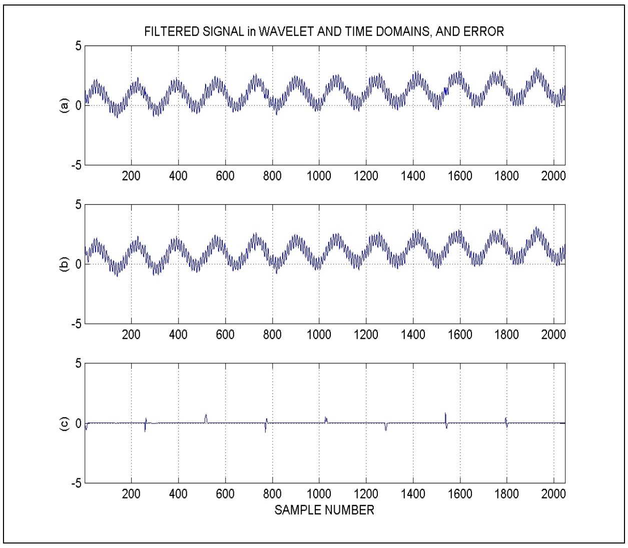

Figure 2: Signal filtered in time and wavelet domains

As can be seen in Figure 3, the block-end artifacts are eliminated, the only residual problem being a short glitch at the start of the signal when there is no data available for the filter to operate on; this can be easily eliminated in any practical implementation.

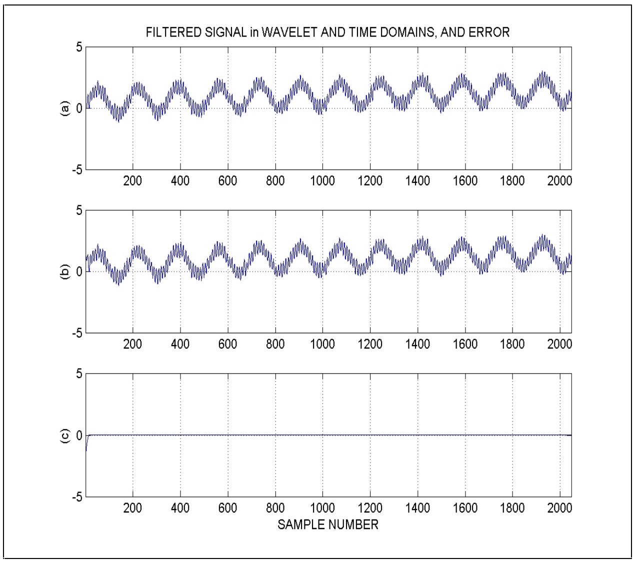

Figure 3: Filtered signals using "over-sampled" algorithm

4. LINEAR ALGEBRAIC IMPLEMENTATION

Another way to consider the problem is to see it from a purely algebraic point of view; in order to implement the filtering procedure, one must adjust the ordering of wavelet coefficients. The standard subband ordering is $\{s_J, d_J, d_{J-1}, \dots, d_1\}$, and within each subband the coefficients are ordered chronologically. It is more efficient to take into account the periodicity of the wavelet transform, and to group together coefficients that share a time localization at different scales. This is done by interleaving the wavelet coefficients recursively: starting from the d_J and s_J sequences, one constructs another sequence u_J with alternated coefficients $u_J = \{\dots\ d_J[k],\ s_J[k],\ d_J[k+1],\ s_J[k+1]\ \dots\}$. Recursively, u_{j-1} is obtained by alternating coefficients from d_j and from u_j. The interleaved sequence is then u_1, which has N coefficients.

In the general case, let us consider the expansion of the signal $x[n]$ in an (as yet unspecified) orthonormal basis $W = \{w_k\}_{k=1 \dots N}$:

$$x[n] = \sum_{k=1}^{N} \alpha_k w_k[n] \qquad (12)$$

Let us call $\tilde{x}$ the filtered version of x by the FIR filter h, and $\tilde{\alpha}$ the corresponding coefficients in W. If we call $\tilde{W} = \{\tilde{w}_k\}_{k=1 \dots N}$ the basis formed by the filtered set of vectors, we have the following relation:

$$\tilde{x}[n] = \sum_{k=1}^{N} \tilde{\alpha}_k w_k[n] = \sum_{k=1}^{N} \alpha_k \tilde{w}_k[n] \qquad (13)$$

Taking the left scalar product of both members of this last equality by a given vector w_{k_0} of W gives:

$$\tilde{\alpha}_{k_0} = \sum_{k=1}^{N} \alpha_k \left\langle w_{k_0}, \tilde{w}_k \right\rangle \qquad (14)$$

where $\langle .,. \rangle$ is the canonical scalar product over $\mathbf{R}^N$. This can be rewritten as a matrix-vector multiplication: if we denote by A(resp. $\tilde{A}$) the $N\times 1$ column vector formed by the α(resp. $\tilde{\alpha}$) coefficients, and M the $N\times N$ matrix which (i,j) coefficient is given by $M_{i,j} = \langle w_i, \tilde{w}_j \rangle$, then eq.(7) is equivalent to

$$\tilde{A} = MA \qquad (15)$$

From (15), it follows that filtering in the wavelet domain can be accomplished by a simple matrix multiplication (the matrix M can be pre-computed). As the *a priori* computational load is then of order N^2, one may wonder what the advantages are to doing so rather than doing the inverse transform, filtering the signal, and coming back in the wavelet domain, since all three operations can be done with fast (order N or $N \log(N)$) algorithms. In the specific case where W is a wavelet basis, the matrix M has few non-zero coefficients, and this considerably reduces the complexity of the matrix multiplication. The input signal is high- and low-pass filtered (perfect reconstruction is obtained with QMF filters), and sub-sampled by a factor 2, to obtain the two sub-signals d_1 and s_1 respectively. This procedure is repeated recursively J times on the low-pass sub-signals s_j to obtain the sub-signals d_{j+1} and s_{j+1}.

In order to implement the above filtering procedure, one has to choose the ordering of wavelet coefficients. The standard subband ordering is $\{s_J, d_J, d_{J-1}, \dots, d_1\}$, and within each subband the coefficients are ordered chronologically. It is more efficient to take into account the periodicity of the wavelet transform, and to group together coefficients that share a time localization at different scales. This is done by interleaving the wavelet coefficients as shown in Figure 4. This re-ordering is done recursively: starting from the d_J and s_J sequences, one constructs another sequence u_J with alternated coefficients $u_J = \{\dots\ d_J[k],\ s_J[k],\ d_J[k+1],\ s_J[k+1]\ \dots\}$. Recursively, u_{j-1} is obtained by alternating coefficients from d_j and from u_j. The interleaved sequence is then u_1, which has N coefficients.

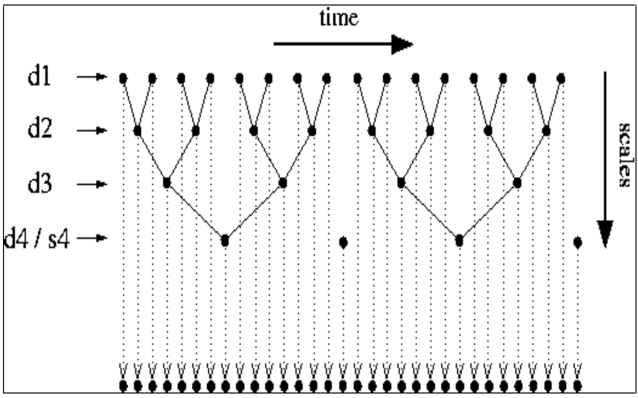

Figure 4: Interleaving the wavelet coefficients (J=4)

With this ordering of wavelet basis functions $\{w_i\}$, one can compute the matrix M with the scalar products between filtered and unfiltered wavelets. Filtering in the (interleaved) wavelet domain is made as follows (Figure 5): from the sequence of interleaved wavelet coefficients of the original signal x take a segment of length l_m (hop size of 2^J between segments), transpose it to a column vector and left-multiply by the matrix m: the result is the set of (interleaved) wavelet coefficients of the filtered signal $\tilde{x}$. Spurious boundary effects are easily suppressed by proper zero-padding of the initial sequence x. The total number of multiply operations for the filtering of the whole signal is reduced to $N \times l_m$.

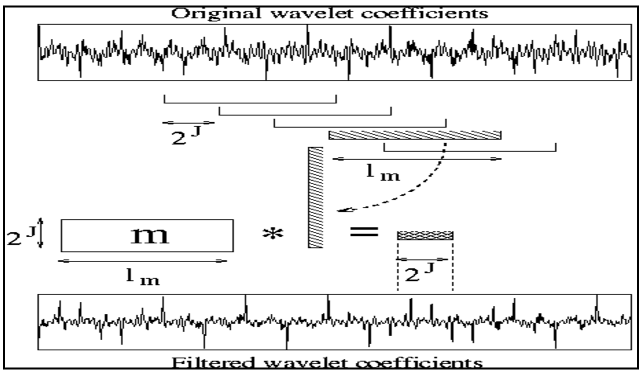

Figure 5: Filtering in the (interleaved) wavelet domain

5. DELAY IN THE WAVELET DOMAIN

To demonstrate the potential of our implementations in the field of digital audio effects, we first of all implemented one of the simplest effects to understand and implement in the time domain, that of delay. We sought to compare delay implemented using our wavelet processing with that implemented in the time domain and with an intelligent delay applied to the wavelet coefficients in each level of the decomposed (transformed) signal.

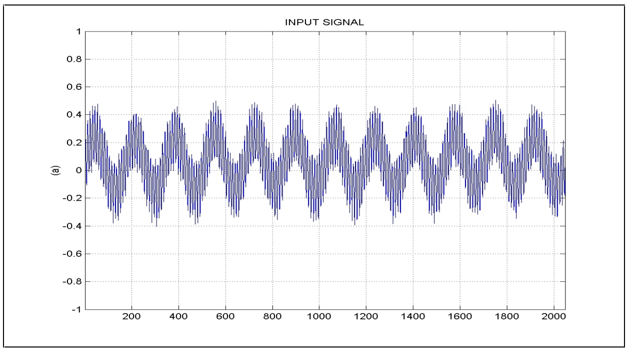

Figure 6: Input signal for digital delay

The signal of Figure 6 – designed with a high level of harmonic content – was fed into our wavelet table implementation and delayed by a variable number of samples. All subsequent figures show (a) signal delayed in the time domain, (b) signal delayed in the wavelet domain, by one or other of the methods mentioned, (c) error between the two.

Figure 7 illustrates the effect for a delay of 128 samples, and shows that while much of the harmonic content of the signal has been retained, a noticeable degree of residual error has been introduced into the signal. Figure 8, on the other hand, clearly demonstrates no such problems with the wavelet table implementation, which delays the signal accurately with residual errors negligible.

One might anticipate that the results of Figure 7 are a reasonable approximation because the rounded delays within each wavelet decomposition level are closest to accuracy when the delay is a power of two. Hence, for the results of Figure 9 we applied a delay of 127 samples in the wavelet domain, and the resulting errors are even more significant; this would correspond to considerable, noticeable unwanted distortion of an audio signal, whereas using the wavelet table method (Figure 10) shows no such problems.

The conclusion to be drawn from this is that there is potential for use of our methods to process wavelet domain audio signals; here we demonstrated the possibilities afforded by simple filtering and delays here, but it is possible to extrapolate these conclusions to any form of linear processing, and possibly to non-linear processing also.

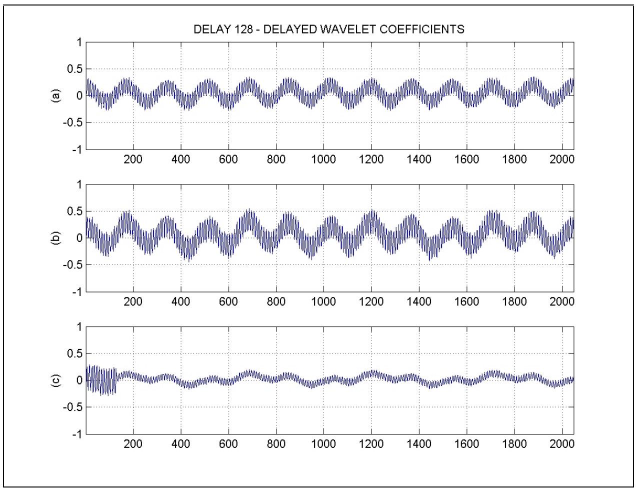

Figure 7: Signal produced by delay of wavelet coefficients

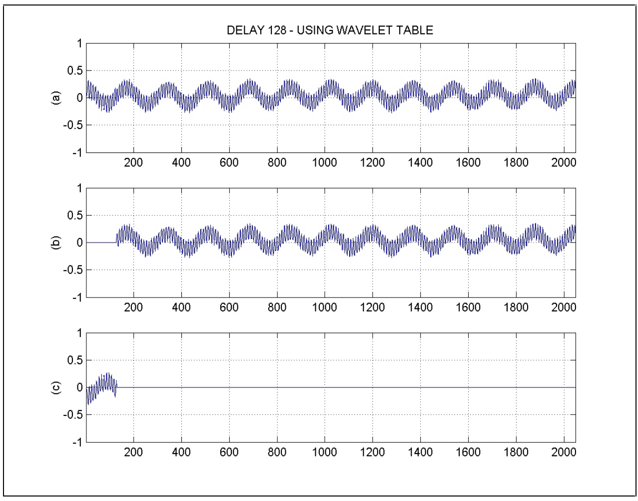

Figure 8: Signal produced by wavelet domain processing

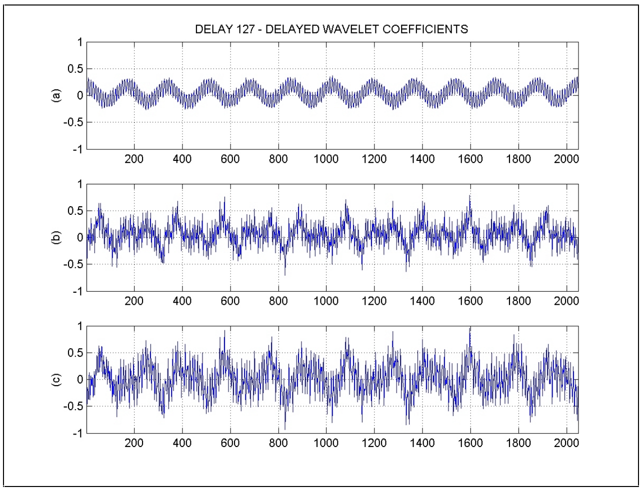

Figure 9: Signal produced by delay of wavelet coefficients

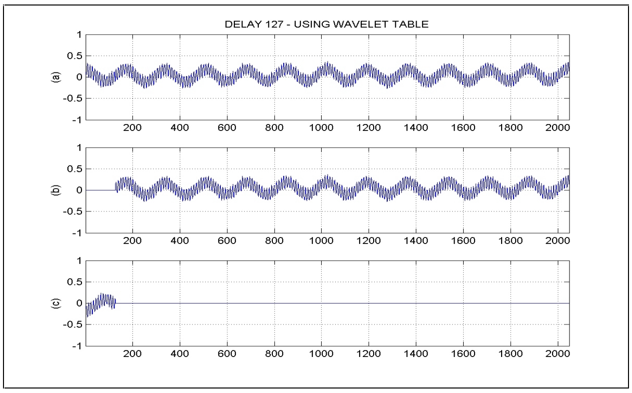

Figure 10: Signal produced using wavelet domain processing

6. CONCLUSIONS

The wavelet table approach has been successfully applied to simple digital audio processing tasks, and it operates effectively when incorporating the over-sampled ('extended block') modification. Similarly, the linear algebraic implementation has been demonstrated to successfully apply linear DSP tasks to 1-D signals such as audio signals. This implementation has the further potential advantage that it is possible to obtain an approximation of $\tilde{x}$ by restricting *m* to a sub-matrix containing the largest coefficients; in most cases the coefficients in the columns near the left and right sides of *m* have a very small amplitude. Such scalability in complexity can in itself justify the use of the wavelet domain for filtering signals.

In assessing success or applicability of this approach, it is necessary to consider whether this 2x*(j+1)*-fold increase in required information (bandwidth), and the resultant increase in complexity of the encoding and decoding, are not so excessive as to negate the advantages of being able to process the signals in the wavelet domain. The wavelet table implementation hinges upon an increase in required information (bandwidth), and the resultant increase in complexity of the encoding and decoding are evident drawbacks of the method. However, both this approach and the matrix algebraic formulation are more efficient implementations of the core idea than the basic requirement to transform signals back into the time domain prior to linear signal processing. Investigation is also ongoing into whether it is possible to extrapolate the wavelet table domain from the standard, non-time-shifted wavelet coefficients [11].

It is possible to extrapolate further potential uses for our implementations – the potential exists for echo, reverb and chorusing effects, for instance, to be deployed on wavelet domain signals. Similarly, non-linear processing is possible; consider dynamics processing (compression and expansion) as an example. At the moment multiband dynamics processing is a widely understood and utilised technique; multiscale dynamics processing is a potentially similar concept. However, in normal circumstances with the signals presented within a regularized "block" of data, the temporal resolution on the dynamic range adjustment is limited; the new schemes presented in this paper could incorporate finer temporal resolution.

7. ACKNOWLEDGEMENTS

David Darlington is supported by EPSRC ROPA grant M86491. Laurent Daudet is supported by the European Community Programme *Human Potential* under contract number HPMF-CT-2000-00917.

8. REFERENCES

[1] Mark Sandler, "Linear Time-Invariant Systems in the Wavelet Domain", *IEE Seminar: Time-scale and Time-Frequency Analysis and Applications*, London, 29 February 2000, pp. 2/1 - 2/6

[2] Pablo-Fernandez-Cid, Javier Casajus Quiros, "Distortion of Musical Signals by Means of Multiband Waveshaping", *Journal of New Music Research* Vol. 30 No. 3, pp/ 279-287.

[3] Philippe Guillemain, Richard Kronland-Martinet, "Characterisation of Acoustic Signals Through Continuous Linear Time-Frequency Representations", *Proceedings of the IEEE* Vol. 84, No. 4, April 1996, pp. 561-585.

[4] Gianpaolo Evangelista, Sergio Cavaliere, "Discrete Frequency Warped Wavelets: Theory and Applications", *IEEE Trans. Signal Processing*, Vol. 46 No. 4 April 1998, pp. 874-885.

[5] David Darlington, Laurent Daudet, Mark Sandler, "Linear Signal Processing in the Wavelet Domain", *SMMSP '02 Spectral Methods and Multirate Signal Processing*, September 7-8, 2002, Toulouse, France.

[6] J. Liang and T. W. Parks, "A Translation-Invariant Wavelet Representation Algorithm with Applications", *IEEE Trans. Signal Processing* Vol. 44 No. 2, February 1996 pp.225-232.

[7] G. Beylkin, "On the Representation of Operators in Bases of Compactly Supported Wavelets", *SIAM J on Numerical Analysis* Vol.6 No.6, pp. 1716-1740.

[8] Israel Cohen, Shalom Raz and David Malah, "Shift Invariant Wavelet Packet Bases", *Proceedings of the 20th IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP-95*, Detroit, Michigan, 1995. pp. 1081-1084.

[9] Ben Leslie, Mark Sandler, "A Wavelet Packet algorithm for 1-D data with no block end effects", *IEEE ISCAS 1999*, Orlando, USA.

[10] C. A. Lanciani and R. W. Schafer, "Sub-Band Domain Filtering of MPEG audio signals", *IEEE ICASSP 1999*.

[11] Qiao Wang and Lenan Qu, "Translation invariance and sampling theorem of wavelet", *IEEE Trans. Signal Processing* Vol. 48 No. 5, May 2000, pp 1471-1474.