Redundant Wavelet Filter Banks on the Half-Axis with Applications to Signal Denoising with Small Delays

François Chaplais, Panagiotis Tsiotras and Dongwon Jung

Abstract— A wavelet transform on the negative half real axis is developed using an average-interpolation scheme. This transform is redundant and can be used to perform causal wavelet processing, such as signal denoising, without delay. Nonetheless, in practice some boundary effects occur and thus a small amount of delay is required to reduce them. The theory is implemented on a challenging signal with large noise and sharp transients. Results from the experimental implementation of the proposed algorithm for the denoising of a feedback signal for controlling a three-phase permanent-magnet synchronous brushless DC motor are also presented.

I. INTRODUCTION

Given a sequence of data $\{a_0[k]\}_{k \in \mathbb{Z}}$ the wavelet transform generates for each scale depth $J \geq 1$ a sequence of scale coefficients $a_J = \{a_J[k]\}_{k \in \mathbb{Z}}$ which give an approximation of the signal at the low resolution scale J along with a family of detail coefficients $d_j = \{d_j[k]\}_{k \in \mathbb{Z}}$, $1 \leq j \leq J$, which contain the information lost when going from a finer resolution scale to a coarser resolution scale.

Thresholding applied on the coefficients of a wavelet transform is also known to be an efficient method for denoising signals with sharp transients [1], [2]. In thresholding one keeps only the coefficients d_j which are larger than a pre-specified tolerance. Standard thresholding typically is performed using wavelets on the whole real line. This causes significant delays in the processing, because some of the filters involved in the composition/decomposition phases of are not causal; see Fig. 1. When operating on on-line data (such as for denoising signals within a feedback control loop) it is imperative to use wavelets on the negative half real axis (the half axis in this context representing past, known values of the signal).

It should be mentioned at this point that although the word "wavelet" will be used throughout in the sequel, the basic point of view taken in this paper is that of FIR filter banks [3], [4]. The various signal decomposition and reconstruction schemes described in this paper were inspired by the average-interpolation scheme of Donoho [5] and Sweldens [6]; see also [7]. As noted in [6], this scheme can be modified to process signals on the half-axis. We explicitly show how this can be done. The proposed scheme is subsequently modified to provide a redundant transform on the negative half-axis. Redundant transforms have several attractive properties such as translation invariance [3]. Recall that the standard (non-redundant) wavelet transform on the other hand is not translation invariant: if the signal is shifted, the wavelet coefficients are not simply a shifted version of the coefficients of the original signal.

F. Chaplais is with Centre Automatique et Systèmes, École Nationale Supérieure des Mines de Paris, 35 rue Saint Honoré, 77305 Fontainebleau, France. Email: francois.chaplais@ensmp.fr. Corresponding author. The author wishes to thank Albert Cohen for suggesting the algorithms of Sections II and III.

P. Tsiotras and D. Jung are with School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150, USA. This work was performed while P. Tsiotras was on a leave to CAS, ENSMP in the Spring of 2003.

For simplicity of exposition, in this paper we will restrict ourselves to wavelets with three discrete vanishing moments. However, the results are easily extendable to wavelets with any number of moments. Recall that a filter h has N discrete vanishing moments if and only if $\sum_{n \in \mathbb{Z}} n^{\ell} h[n] = 0$, $\ell = 0, \ldots, N-1$.

II. THE AVERAGE-INTERPOLATION SCHEME

A very simple, yet classic, scheme for building a discrete wavelet transform with discrete vanishing moments is based on so called "cell averages" [7], also known as "blocky kernels" [5]. This discrete transform has two parts: the decomposition, which yields the wavelet coefficients, and the reconstruction, which retrieves a signal from these coefficients. The decomposition and reconstruction are both obtained using a pair of FIR filters. While these filters are actually identical [5] to the $(1, 3)$ spline biorthogonal filters of Daubechies [8], here we will use average-interpolation together with the usual FIR filter presentation, as illustrated in Fig. 1. Average interpolation can be easily adapted to derive a similar scheme for processing signals over the half axis, which is the main objective of this paper.
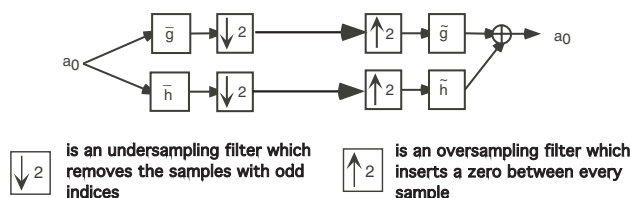

Fig. 1.   Perfect reconstruction filter banks, used for the implementation of the wavelet transform on the real axis.

A. Decomposition

As usual in wavelet processing, the data is filtered by a low pass and a high pass filter.

1) Low pass filter: Up to a normalizing scalar of $\sqrt{2}$, the low pass filter computes the average of two successive signal values. This is the Haar decomposition filter [3]

$$a_{j+1}[n] = \frac{a_j[2n] + a_j[2n-1]}{\sqrt{2}}. \tag{1}$$

2) High pass filter: First, we present the high pass filter using the average-interpolation scheme, giving its numerical value later on.

We wish the high pass decomposition filter to have three discrete vanishing moments. This means that the output of discrete polynomial sequences of the form $a[n] = \sum_{k=0}^{k=2} a_k n^k$ of degree less than or equal to two through the filter is zero. This vanishing moments property ensures that the corresponding low pass reconstruction filter will efficiently approximate sufficiently "regular" signals up to order two. This property is satisfied with the use of an average-interpolation scheme.

The high pass filter considers the averages $(a_j[2n] + a_j[2n-1])/2 = a_{j+1}[n]/\sqrt{2}$ and identifies the unique polynomial $P(x)$

of degree less than or equal to two whose averages of the coarser grid with step 2^{j+1} coincides with three successive values of $a_{j+1}[n]/\sqrt{2}$. It then computes the averages of $P(x)$ on the fine grid with step 2^j, subtracts the actual values of the signal from these fine averages, and normalizes the result by a factor $\sqrt{2}$. This sequence of operations is illustrated in Fig. 2.
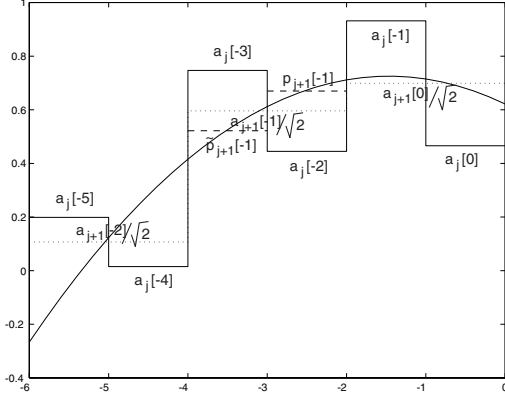

Fig. 2. Average interpolation using a polynomial of degree less than or equal to two.

The steps of the construction of the high pass decomposition filter can be summarized as follows:

(i) At scale j and time $2n$, the three low pass outputs $a_{j+1}[n-1]$, $a_{j+1}[n]$ and $a_{j+1}[n+1]$ are computed.

(ii) Let $P(x)$ be the unique polynomial of degree less than or equal to two such that the discrete averages $a_{j+1}[n-1]/\sqrt{2}$, $a_{j+1}[n]/\sqrt{2}$ and $a_{j+1}[n+1]/\sqrt{2}$ are the averages of P on the intervals $[-2^{j+1}, 0]$, $[0, 2^{j+1}]$ and $[2^{j+1}, 2^{j+2}]$ respectively. Denote by $p_{j+1}[n]$ the average of P on $[2^j, 2^{j+1}]$. $p_{j+1}[n]$ is the called the *prediction* of $P(x)$ obtained from $a_{j+1}[n-1]$, $a_{j+1}[n]$ and $a_{j+1}[n+1]$.

(iii) The output of the high pass filter (or detail) is then the difference

$$d_{j+1}[n] = \sqrt{2}(a_j[2n] - p_{j+1}[n]). \tag{2}$$

Up to $\sqrt{2}$, this is the difference between the actual value of the signal and its prediction of the average interpolating polynomial $P(x)$.

These operations can be conveniently represented as convolution with the filter (before subsampling)

$$\bar{g} = \left[-\frac{\sqrt{2}}{16}, -\frac{\sqrt{2}}{16}, \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{16}, \frac{\sqrt{2}}{16} \right] \tag{3}$$

with support $[-2, 3]$; see Fig. 1. These coefficients are computed from (2) after writing down explicitly the expression for $p_{j+1}[n]$ which results in the following expression for the detail coefficients

$$d_j[n] = \sqrt{2} \big(\frac{1}{16} a_{j-1}[2n-3] + \frac{1}{16} a_{j-1}[2n-2] - \frac{1}{2} a_{j-1}[2n-1]$$
$$+ \frac{1}{2} a_{j-1}[2n] - \frac{1}{16} a_{j-1}[2n+1] - \frac{1}{16} a_{j-1}[2n+2] \big)$$

a) Vanishing moments: We now show explicitly the vanishing moments property for polynomials of degree less than or equal to two. If $a_j[2n-3], \ldots, a_j[2n+2]$ is a polynomial sequence of degree less than or equal to two, then by definition it is a sequence of the averages of some polynomial $Q(x)$ of degree less than or equal to two on the intervals $[n2^j, (n+1)2^j]$ for

$n = -2, \ldots, 3$. Since the averages of averages on disjoint intervals are averages, $P(x)$ at step (ii) of the high pass filter necessarily satisfies $P(x) = Q(x)$, and its average on the interval $[2^j, 2^{j+1}]$ is equal to $a_j[2n]$. This proves that $d_{j+1}[n] = 0$.

b) Oscillation of the details: Denote by $\tilde{p}_{j+1}[n]$ the average of $P(x)$ on $[0, 2^j]$ at step (ii) of the high pass construction filter above. Recall that $a_{j+1}[n]/\sqrt{2}$ is both the average of $a_j[2n]$ and $a_j[2n-1]$ and the average of $P(x)$ on $[0, 2^{j+1}]$. Hence it is also the average of $p_{j+1}[n]$ and of $\tilde{p}_{j+1}[n]$, which are the averages of P on the half intervals. This yields,

$$\frac{a_j[2n] + a_j[2n-1]}{2} = \frac{a_{j+1}[n]}{\sqrt{2}} = \frac{p_{j+1}[n] + \tilde{p}_{j+1}[n]}{2}.$$

This proves that the details are oscillating, that is,

$$a_j[2n-1] - \tilde{p}_{j+1}[n] = -d_{j+1}[n]/\sqrt{2} \tag{4}$$

At the reconstruction stages, the low pass outputs a_{j+1} are available, and the average interpolated polynomial $P(x)$ can be computed. Hence $p_{j+1}[n]$ and $\tilde{p}_{j+1}[n]$ can be computed; since $d_{j+1}[n]$ is also available, $a_j[2n]$ and $a_j[2n-1]$ can be recovered using (2) and (4). In short, the sequence a_j can be reconstructed from the sequences a_{j+1} and d_{j+1}.

B. Reconstruction

The reconstruction stage is based on the same as before. First, a polynomial of degree less than or equal to two is identified as in step (ii) of the high pass decomposition filter, using the outputs of the decomposition low pass filter. The averages $p_{j+1}[n]$ and $\tilde{p}_{j+1}[n]$ of this polynomial on the intervals $[2^j, 2^{j+1}]$ and $[0, 2^j]$ are then computed. The part of the signal a_j which has even indices is recovered from the prediction and details using (2). The part with odd indices uses equation (4), as explained in the section about the oscillation of the details. If the decomposition has not been modified, the reconstruction restores the signal exactly.

It is common to express the reconstruction process as the sum of the outputs of a low pass and a high pass reconstruction filter. These filters are expressed as the succession of an oversampling of the signal by zero insertion followed by the filtering itself, as in Fig. 1.

1) Low pass filter: Computing explicitly the two predictions on the left and right subintervals of $[2^{j+1}(n-1), 2^{j+1}n]$ gives the following values for the coefficients of the low pass reconstruction filter:

$$\left[-\frac{1}{8}, \frac{1}{8}, 1, 1, \frac{1}{8}, -\frac{1}{8} \right] \frac{1}{\sqrt{2}},$$

with support $[-3, 2]$.

2) High pass filter: This is the Haar reconstruction high pass filter, e.g.,

$$\left[-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right],$$

with support $[-1, 0]$.

III. ADAPTATION TO THE HALF AXIS

In this section the scheme of Section II will be modified at the right boundary to use only past, available values of the signal. The modification essentially decenters the prediction scheme at the boundary to account for the absence of future values.

A. Decomposition

The low pass averaging filter does not require any modification since it uses only past values. For negative output indices, the high pass filter uses data with negative or zero indices (see Fig. 2). Hence the high pass filter needs only to be modified for the prediction at output index $n = 0$, when input data is not available at the indices $n = 1, 2$. To achieve this the polynomial is identified on the last three averages and the prediction is computed at the index $n = 0$, as illustrated in Fig. 3.
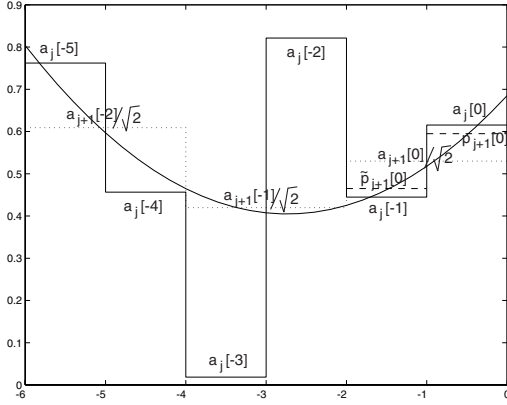
Fig. 3.   The average interpolation scheme is decentered to produce a prediction at the boundary. Here $j = 0$.

The details are given below:

(i) At scale j and index $n = 0$, the three low pass outputs $a_{j+1}[-2]$, $a_{j+1}[-1]$ and $a_{j+1}[0]$ are computed.
(ii) Let $P(x)$ be the unique polynomial of degree less than or equal to two such that $a_{j+1}[-2]/\sqrt{2}$, $a_{j+1}[-1]/\sqrt{2}$ and $a_{j+1}[0]/\sqrt{2}$ are the averages of $P(x)$ on $[-3 \cdot 2^{j+1}, -2 \cdot 2^{j+1}]$, $[-2 \cdot 2^{j+1}, -2^{j+1}]$ and $[-2^{j+1}, 0]$ respectively. Denote by $p_{j+1}[0]$ the average of $P(x)$ on $[-2^j, 0]$, that is, the rightmost half interval.
(iii) The output of the high pass filter (or detail) is the difference

$$d_{j+1}[0] = \sqrt{2}(a_j[0] - p_{j+1}[0]). \quad (5)$$

The prediction is now decentered. It can be verified using the same arguments as in the centered scheme that the high pass filter still has three discrete vanishing moments and the details are oscillating at $n = 0$.

One can compute the prediction explicitly and then use it to compute the high pass output $d_{j+1}[0]$ at the boundary. Its value is

$$d_{j+1}[0] = \sqrt{2}\big(\frac{5}{16}a_j[0] - \frac{11}{16}a_j[-1] + \frac{1}{4}a_j[-2]$$
$$+ \frac{1}{4}a_j[-3] - \frac{1}{16}a_j[-4] - \frac{1}{16}a_j[-5]\big).$$

B. Reconstruction

Except for the indices $n = -1$ and $n = 0$, the reconstruction is performed exactly as in Section II-B; the signal is recovered for negative indices. At the boundary, the prediction $p_{j+1}[0]$ is performed as in Section III-A, with an extra prediction $\tilde{p}_{j+1}[0]$, which computes the average of the interpolation polynomial $P(x)$ on the interval $[-2^{j+1}, -2^j]$. Up to a shift, this amounts to predicting the averages on the last two intervals of the fine grid from the identified polynomial.

The value $a_j[0]$ at the edge is recovered using (5), while $a_j[-1]$ is recovered with a similar equation which uses $\tilde{p}_{j+1}[0]$ and the oscillation of the detail, as in the centered scheme. The reconstruction is performed at the edge by using the following equations, which are obtained by explicitly writing down the predictions:

$$a_j[0] = \frac{1}{\sqrt{2}}\big(\frac{1}{8}a_{j+1}[-2] - \frac{1}{2}a_{j+1}[-1] + \frac{11}{8}a_{j+1}[0] + d_{j+1}[0]\big),$$
$$a_j[-1] = \frac{1}{\sqrt{2}}\big(-\frac{1}{8}a_{j+1}[-2] + \frac{1}{2}a_{j+1}[-1] + \frac{5}{8}a_{j+1}[0] - d_{j+1}[0]\big).$$

IV. REDUNDANCY ON THE HALF-AXIS

The transform presented in the previous sections is nonredundant. That is, any pair of sequences a_{j+1} and d_{j+1} can be interpreted as the decomposition of a signal a_j. Redundant transforms are known to improve signal processing with wavelets, especially for denoising [2]. Ideally, the proposed redundant transform should be shift invariant. When wavelets on the real line are used, this is achieved by removing the subsampling and zero insertions in the filter banks of Figure 1 and using filters which depend on the scale. More precisely, at the input scale j, $2^j - 1$ zeros are inserted between each coefficient of the original *filter*. The corresponding algorithm is known as the *algorithme à trous* [9]. This scheme cannot be used directly for transforms on the half-axis because boundary effects are not properly handled by the zero insertion trick. Here we develop a redundant transform on the half-axis in two steps: the first step extends the transform of Section III to a redundant transform at the base scale $j = 0$. The second one interprets the zero insertion on filters (as opposed to signals) as filtering on multiplexed signals. This interpretation is still valid for time-varying filters and it will be used later on to define the boundary filters for the scales $j > 0$.

A. Redundancy at the first scale

The task of this section is to extend the transform of Section III to a redundant transform such that:

• The transform of Section III is obtained by keeping the values with even indices in the redundant transform
• The transform provides perfect reconstruction using the odd indices, provided one also knows the last decomposition output $a_{j+1}[0]$.

The simplest idea for generating the values of the transform for the odd indices is to shift the signal one step to the right, drop the boundary value $a_0[0]$, use the previous transform, and shift back the result to the left. This approach has two small flaws that we wish to remedy: first, $a_0[0]$ cannot be reconstructed; second, the average interpolation scheme at the last index is still decentered, meaning that an extrapolation is performed in the prediction step instead of an interpolation. If we used the extra value $a_1[0]$, in addition to the low pass output of the shifted transform, the boundary high pass filter can be modified to use interpolation instead of extrapolation; moreover, $a_0[0]$ can be recovered at the reconstruction.

1) Decomposition: The shift invariant filters of Section II-A are first applied to the signal, and the output is now subsampled at the odd indices, the last output index being $n = -1$ for the low pass, and $n = -3$ for the high pass. For the index $n = -1$, the detail $d_1[-1]$ is computed as follows:

(i) Let $P(x)$ be the unique polynomial of degree less than or equal to two such that $a_1[-3]/\sqrt{2}$, $a_1[-1]/\sqrt{2}$ and

$a_1[0]/\sqrt{2}$ are the averages of $P(x)$ on the intervals $[-5,-3]$, $[-3,-1]$ and $[-2,0]$ respectively. Denote by $p_1[-1]$ the average of $P(x)$ on $[-2,-1]$. The difference with the boundary scheme of the previous section is that the average interpolating polynomial is not drawn from disjoint intervals, and the prediction is computed from an interpolation, thanks to the use of $a_1[0]$. This prediction scheme is illustrated in Fig. 4.

(ii) The detail is defined by

$$d_1[-1] = \sqrt{2}(a_0[-1] - p_1[-1]). \qquad (6)$$
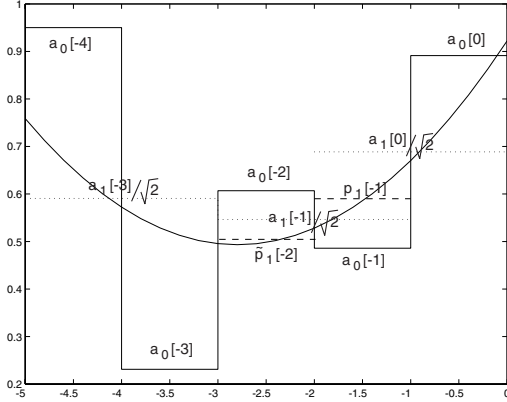
Fig. 4. Computation of the prediction for index $n = -1$. The polynomial is fit on overlapping intervals, using the values $a_1[-3]$, $a_1[-1]$ and $a_1[0]$.

The detail $d_1[-1]$ can be computed by writing down the value of the prediction

$$d_1[-1] = \sqrt{2}\Big(\frac{1}{24}a_0[-4] + \frac{1}{24}a_0[-3] - \frac{3}{8}a_0[-2]$$
$$+ \frac{11}{24}a_0[-1] - \frac{1}{6}a_0[0]\Big).$$

Since the outputs of the decomposition of Section II-A are to be now the even subsamples of a redundant transform, we shall denote by $a_1[2k]$ and $d_1[2k]$ their values, with $k \leq 0$. We therefore have just defined the outputs of the decomposition for all odd indices $n = \ldots, -5, -3, -1$. This complements the sequences $a_1[2k]$, $d_1[2k]$ into a redundant transform with values for all negative or zero integer indices.

a) Vanishing moments: If $a_0[-4], \ldots, a_0[0]$ are the averages of a polynomial $Q(x)$ of degree less than or equal to two on the intervals $[-5,-4] \ldots [-1,0]$, then $a_1[-3]/\sqrt{2}$, $a_1[-1]/\sqrt{2}$ and $a_1[0]/\sqrt{2}$ are the averages of $Q(x)$ on the intervals $[-2,0]$, $[0,2]$ and $[1,3]$ respectively. Hence $P(x) = Q(x)$ and the detail is zero at step (ii) above. This shows that the previous boundary management preserves the vanishing moments property.

b) Oscillation of the details: We leave it to the reader to verify that

$$-d_1[-1] = a_0[-2]\sqrt{2} - \tilde{p}_1[-2] \qquad (7)$$

and hence, the details are oscillating.

2) Perfect reconstruction from the odd samples only: The signal can be entirely recovered provided the last even low pass output $a_1[0]$ is added to the previous transform. For indices smaller or equal to $n = -3$, the signal is recovered as in Section II-B, since there was no boundary management at the decomposition. For the indices $n = -2$ and $n = -1$, the predictions $p_1[-1]$ and $\tilde{p}_1[-2]$

can be computed from the polynomial P, which is computed from $a_1[-3]$, $a_1[-1]$ and $a_1[0]$. Together with $d_1[-1]$, equations (6) and (7) provide the reconstruction of $a_0[-1]$ and $a_0[-2]$. Finally, $a_0[0]$ is restored trough the equation

$$a_0[0] = \sqrt{2}a_1[0] - a_0[-1].$$

Specifically, the reconstruction of $a_0[-2]$ and $a_0[-1]$ is given by

$$a_0[-2] = -\frac{1}{\sqrt{2}}\Big(\frac{1}{12}a_1[-3] + \frac{5}{4}a_1[-1] - \frac{1}{3}a_1[0] - d_1[-1]\Big),$$

$$a_0[-1] = \frac{1}{\sqrt{2}}\Big(-\frac{1}{12}a_1[-3] + \frac{3}{4}a_1[-1] + \frac{1}{3}a_1[0] + d_1[-1]\Big).$$

This proves that the signal can be recovered from the odd outputs of the redundant transform, to which $a_1[0]$ has been added.

3) Perfect reconstruction from the redundant transform: For shift-invariant transforms on the real line, the signal is reconstructed by computing the average of the two possible reconstructions, e.g., from the even- and odd-indexed subsamples of the transform. We do the same here and restore the signal by computing the average of the reconstructions of Sections III-B and IV-A.2. For indices smaller than $n = -2$, the reconstruction is performed as for shift-invariant redundant transforms. For the indices $n = -2$, $n = -1$ and $n = 0$, the boundary filters can be precomputed by averaging the reconstruction formulas from the even and odd subsamples.

B. Redundancy at coarser scales

When using wavelets on the real axis, the redundant transforms at the coarse scales are obtained, as usual, by inserting zeros into the original filters. We give here an interpretation of this zero insertion as an operation of the original filters on a multiplexed signal. This interpretation is then used to extend the zero insertion to time-varying filters, specifically, filters which take different values close to the boundary.

At the decomposition from scale j to scale $j + 1$, we consider a signal x to be a combination of 2^j signals $\{x_\ell\}_{0 \leq \ell < 2^j}$, with $x_\ell[n] = x[n2^j - \ell]$. Recall now that at scale j the zero insertion of the filter h is the filter defined by

$$h_j[n] = \begin{cases} h[p], & \text{if } n = 2^j p, \\ h_j[n] = 0, & \text{otherwise.} \end{cases} \qquad (8)$$

The redundant decomposition from scale j to scale $j + 1$ is obtained by convolving the input with the zero insertion of the filters at the scale j. Then the convolution y with h_j can be split into the series of 2^j convolutions

$$\begin{aligned} y_\ell[n] &= y[2^j n - \ell] = \sum_{k \in \mathbb{Z}} h_j[k]\, x[2^j n - \ell - k] \\ &= \sum_{p \in \mathbb{Z}} h_j[2^j p]\, x[2^j n - \ell - 2^j p] = \sum_{p \in \mathbb{Z}} h[p]\, x_\ell[n - p] \\ &= \sum_{p \in \mathbb{Z}} h[n - p]\, x_\ell[p] = h * x_\ell[n], \qquad 0 \leq \ell < 2^j \end{aligned}$$

Hence the convolution with zero inserted filters can be viewed as the merging of 2^j parallel convolutions y_ℓ of the filter h with the 2^j signals x_ℓ. This can be extended to time varying filters $h_n[p]$ by setting

$$y_\ell[n] = \sum_{p \in \mathbb{Z}} h_n[p]\, x_\ell[p]. \qquad (9)$$

In practice, zero insertion is used only when the filtering coincides with the time invariant processing of Section II. When

close to the boundary, the multiplexed formula (9) is used on the explicit expressions of the boundary filters. Similarly, during reconstruction, zero insertion is used when wavelets on the whole real axis are used. When boundary filters are used, then the multiplexed formula (9) is used as in the decomposition stage. Finally, the reconstructed signal is obtained, as usual, i.e., by adding the outputs of the low pass and high pass filters and dividing the sum by two at each scale [9].

V. NUMERICAL EXAMPLE

Consider the noisy reference signal shown in Fig. 5. The steep jump after sample number 2000 makes this signal particularly relevant to wavelet-based denoising methods. This is because it is known that wavelet thresholding can be used to efficiently denoise piecewise regular signals [1], [2].
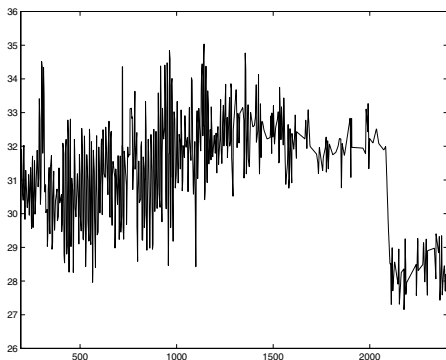
Fig. 5. Original signal. Horizontal axis indicates sample number.

The denoising procedure is performed as follows: at each time t, a time window $[t - T, t]$ of length T is used to select the data on which the transform is to be computed. The window length T is essentially determined by the number of scales used in the transform. It is equal to the length of the filter which is the longest at the fine scale, multiplied by 2^j. Hard thresholding is then performed on the wavelet coefficients[1], that is, all $d_j[n]$ which are below a given threshold are set to zero. A signal is then reconstructed using the techniques of Section IV-A.3. Ideally, sampling the reconstructed signal at time t provides an estimate of the denoised signal without any delay. In practice, it was observed that the estimate is degraded at instants close to the present time. For this reason, a user-defined delay τ is introduced and the value of the reconstructed signal at time $t - \tau$ is taken as an estimate of the denoised signal at time $t - \tau$.

Figure 6 shows the effect of thresholding using the nonredundant transform of Section III with six scales, a hard threshold of 8, and a delay $\tau = 20$ samples. As a comparison, the large jump of the signal after 2000 has a duration of 60 samples.

Figure 6(b) shows the results using the redundant transform with the same threshold value. One can see–at least visually–the gain brought about by the redundancy.

Finally, the Cohen-Daubechies-Vial-Jawerth wavelets were used to denoise the signal of Fig. 5. Recall that the Cohen-Daubechies-Vial-Jawerth wavelets [10], [11] are orthogonal wavelets on the interval and have vanishing moments. Nonetheless, the associated filters do not have vanishing moments, that is, polynomial sequences do not vanish through the high-pass filter. Figure 7 shows

---

[1]The number of scales of the transform and the threshold have been determined empirically. They are the same for all transforms.
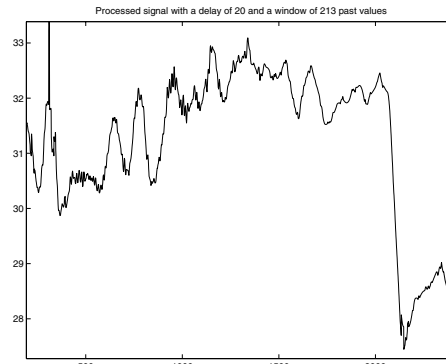
(a) Non-redundant transform
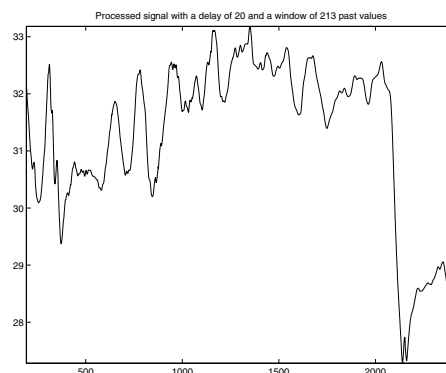
(b) Redundant transform

Fig. 6. Denoising with wavelet transform over 6 scales, a hard threshold of 8 and a delay of $\tau = 20$ samples.

the result of hard denoising using the Cohen-Daubechies-Vial-Jawerth wavelet filters with the same threshold and number of scales as before. The delay is again taken to be $\tau = 20$ samples. It is evident that the performance is much poorer than using the average interpolation scheme.
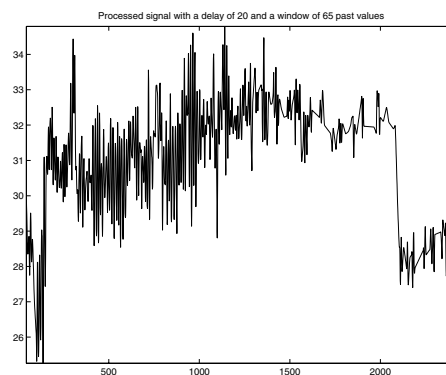

Fig. 7. Denoising with Cohen-Daubechies-Vial-Jawerth wavelets over 6 scales, a hard threshold of 8 and a delay of $\tau = 20$ samples.

VI. EXPERIMENTAL RESULTS

The greatest benefit of an *on-line* denoising algorithm, as the one proposed in this paper, is its potential use for denoising signals in a feedback loop. In this section we report experimental results from the implementation of the previous algorithm to denoise
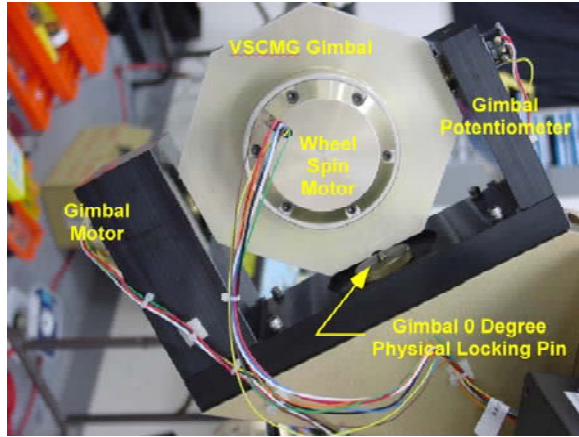
Fig. 8. Main components of the VSCMG assembly. In reaction wheel mode only the wheel DC motor is active while the gimbal remains fixed.

the angular velocity signal of a three-phase permanent magnet synchronous DC motor which is used to drive a reaction wheel. The DC motor used in the experiments is part of a variable-speed control moment gyro (VSCMG) actuator module[2] which provides attitude control for the Integrated Attitude Control Simulator (IACS) located at the School of Aerospace Engineering at the Georgia Institute of Technology. IACS is an experimental facility for simulating three-axis spacecraft attitude maneuvers. Figure 8 shows the VSCMG assembly with the wheel spin motor and Fig. 9 shows the complete spacecraft simulator. Details for the design, construction and other specifications of this spacecraft simulator can be found in Ref. [12].

In reaction wheel mode, control is provided by accelerating or decelerating each of the four wheels. Figure 10 shows the open-loop motor responses to a square and a sinusoidal angular acceleration command, respectively. These open-loop responses are not satisfactory owing to friction, nonlinear effects in the motor dynamics, etc. Such discrepancies will have deleterious effects in achieving tight pointing attitude requirements.

To improve performance, a PI controller was designed using velocity feedback. The rotational speed of the wheel was obtained by numerically differentiating angular position data of the wheel provided via digital encoders at a sampling rate of 100 Hz. Numerical differentiation introduces noise in the angular velocity signal which must be reduced before the signal is used by the PI controller. The purpose of denoising was therefore to remove the noise due to the differentiation of the encoder signal.

A C code of the proposed algorithm was written and implemented as an S-function in SIMULINK®. The Real-Time Workshop® toolbox was used to compile and generate the code from the complete SIMULINK® diagram, and the xPCTarget® (with Embedded Option) toolbox was used to run the executable module in real-time.

The results from two separate experiments are shown below. During the first experiment measurements of the angular velocity signal where processed off-line using the wavelet denoising algorithm. The purpose of those experiments was to estimate appropriate values for the number of scales, threshold and delay.

---

[2]Each VSCMG actuator has two motors. One motor controls the wheel speed (reaction wheel mode) and the other controls the gimbal rate (CMG mode).
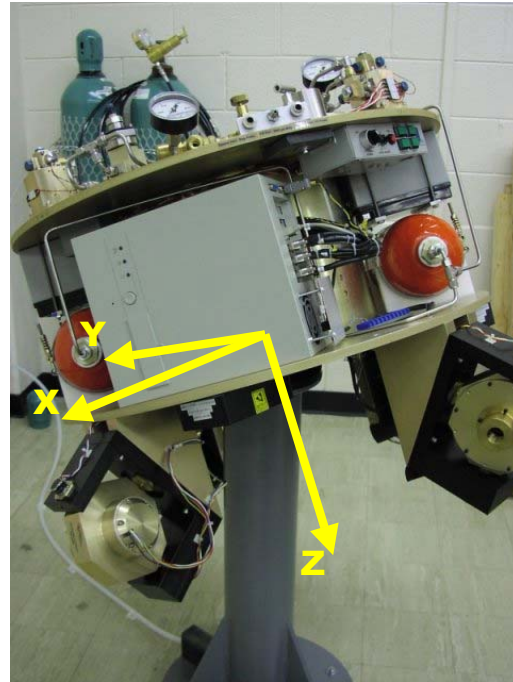

Fig. 9. The body-axes definition for the IACS. The x-axis is designated as the roll axis, the y-axis is designated as the pitch axis and the z-axis is designated as the yaw axis.
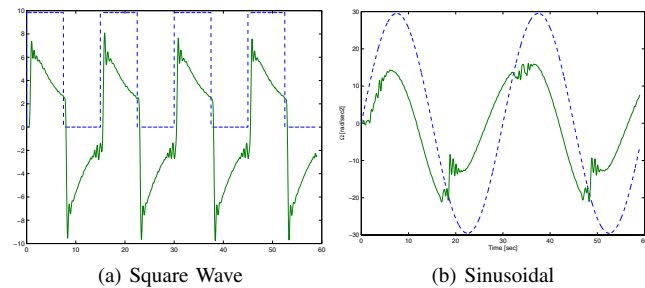

(a) Square Wave   (b) Sinusoidal

Fig. 10. Open-loop response of angular acceleration to square-wave and sinusoidal angular acceleration commands.

The results of this experiment for a sawtooth angular velocity input are shown in Fig. 11. Figure 11(a) shows the results of denoising over a 3 sec interval (300 samples) at around $t = 15$ sec. The dotted line shows the raw angular velocity data and the solid line shows the result after wavelet denoising using 4 scales, a threshold of 20 and a delay of 5. The measured data has been smoothed and most of the noise has been removed. A small delay is evident because of the processing, and a small amount of noise has remained in the signal. This noise can be further removed at the expense of more delay. For instance, Fig. 11(b) shows the results of denoising using a delay of 15 samples or 0.15 sec.

For the second experiment, the denoised angular velocity signal is used as the input to a PI controller. The purpose of the PI controller is to achieve good tracking to torque (i.e., angular acceleration) commands. This achieved by a tight loop on the measured angular velocity. The block diagram schematic of the PI/Motor interconnection is shown in Fig. 12. The results from two separate angular acceleration commands are presented below. First, the results from a square-wave angular acceleration command of
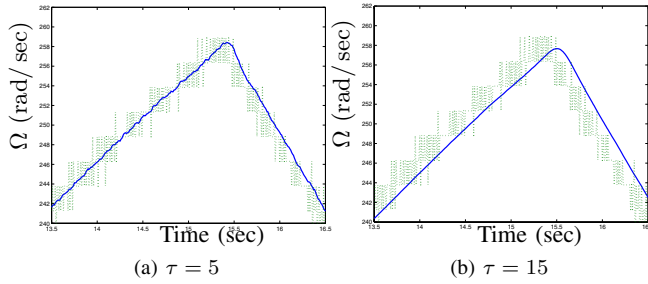
Fig. 11. Wavelet denoising of angular velocity signal using 4 scales and a threshold of 20 (detailed view).
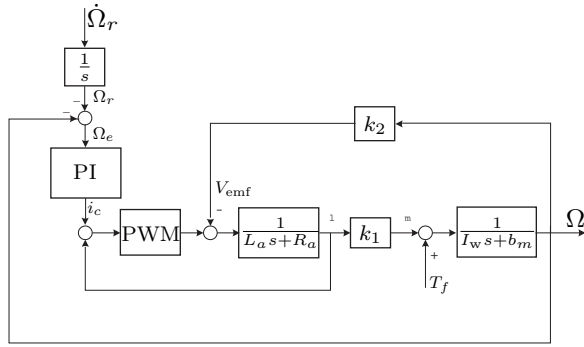

Fig. 12. Closed loop control for $\dot{\Omega}$ using velocity feedback

magnitude $10 \text{ rad}/\sec^2$ are shown in Fig. 13. In Fig. 13(a) the

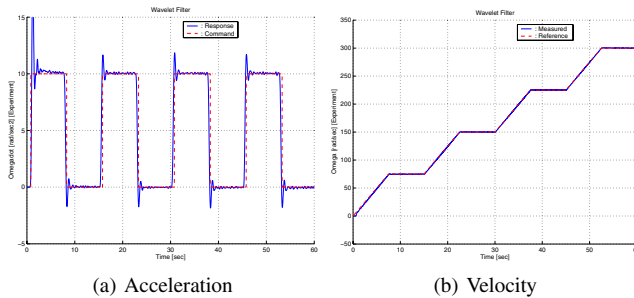(a) Acceleration   (b) Velocity

Fig. 13. Closed-loop response of angular acceleration and velocity to square-wave angular acceleration commands; on-line denoising of angular velocity signal using a wavelet filter with 4 scales, a threshold of 30 and a delay of 10 samples.

command is shown by a dashed line and the response is shown by a solid line. The corresponding response of the angular velocity is shown in Fig. 13(b). The results for a sinusoidal torque command of amplitude $15 \text{ rad}/\sec^2$ and frequency of 0.03 Hz are shown in Fig. 14. As before, in Fig. 14 the dotted line is the reference angular velocity whereas the solid line is the actual response of the DC motor. Both Figs. 13 and 14 show very good tracking of the commanded signals (compare with the open-loop responses in Fig. 10.)

VII. CONCLUSION

In this paper we propose a method for wavelet signal processing on the half-axis. The starting point for the development of the
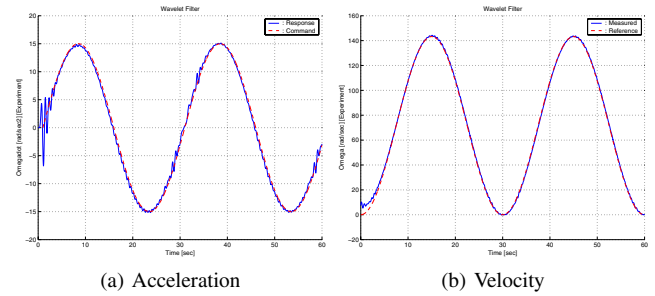

(a) Acceleration   (b) Velocity

Fig. 14. Closed-loop response of angular acceleration and velocity to sinusoidal angular acceleration commands; on-line denoising of angular velocity signal using a wavelet filter with 4 scales, a threshold of 30 and a delay of 10 samples.

results in this paper is the method of average-interpolating polynomials. Using this method boundary effects arising from working in a semi-infinite interval can be handled is a straightforward manner. The motivation behind denoising on the half-axis stems from the need for on-line denoising for certain applications (e.g., within a feedback loop) where future values of the data are not available. We provide experimental evidence for the potential of the proposed scheme for on-line denoising by applying it to denoise the feedback signal used for angular velocity and angular acceleration control of a brushless DC motor of a reaction wheel assembly.

REFERENCES

[1] D. Donoho and I. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, December 1994.

[2] R. Coifman and D. Donoho, "Translation invariant de-noising," in *Wavelets and Statistics*, ser. Lecture Notes in Statistics, A. Antoniadis and G. Oppenheim, Eds. Berlin: Springer-Verlag, 1995, pp. 125–150.

[3] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms*. New Jersey: Prentice-Hall, 1998.

[4] S. Mallat, *A Wavelet Tour of Signal Processing*. New York: Academic Press, 1999.

[5] D. Donoho, "Smooth wavelet decompositions with blocky coefficient kernels," in *Recent Advances in Wavelet Analysis*, L. Schumaker and F. Ward, Eds. Academic Press, 1993, pp. 259–308.

[6] W. Sweldens and P. Schröder, *Building Your Own Wavelets at Home*. ACM, 1996, pp. 15–87, ACM SIGGRAPH Course Notes.

[7] A. Harten, "Multiresolution representation of cell-averaged data: A promotional review," in *Signal and Image Representation in Combined Spaces*, Zeevi and Coifman, Eds. Academic Press, 1997.

[8] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.

[9] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, *Wavelets, Time-Frequency Methods and Phase Space*. Berlin: Springer-Verlag, 1989, ch. A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform, pp. 289–297.

[10] A. Cohen, I. Daubechies, and P. Vial, "Wavelet bases on the interval and fast algorithms," *J. of Appl. and Comput. Harmonic Analysis*, vol. 1, pp. 54–81, 1993.

[11] A. Cohen, I. Daubechies, B. Jawerth, and P. Vial, "Multiresolution analysis, wavelets and fast algorithms on an interval," *Compt. Rend. Acad. Sci. Paris, A*, vol. 316, pp. 417–421, 1992.

[12] D. Jung and P. Tsiotras, "A 3-dof experimental test-bed for integrated attitude dynamics and control research," in *AIAA Guidance, Navigation and Control Conference*, Austin, TX, 2003, AIAA Paper 03-5331.