On-Line Wavelet Denoising with Application to the Control of a Reaction Wheel System

François Chaplais*
Ecole des Mines de Paris
35 rue Saint Honore 77305, Fontainebleau, France

Panagiotis Tsiotras† and Dongwon Jung‡
Georgia Institute of Technology, Atlanta, GA 30332-0150, USA

A wavelet transform on the negative half real axis is developed using an average-interpolation scheme. This transform can be used to perform causal wavelet processing, such as signal denoising, with a small delay. The delay required to obtain acceptable denoising levels is decreased by using a redundant transform instead of a non-redundant one. 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 that drives a reaction wheel system are presented.

I. Introduction

Wavelets have recently become very popular in signal processing. They allow compact representation of a signal with a small number of wavelet coefficients. Given a sequence of data $a_0 = \{a_0[k]\}_{k\in\mathbb{Z}}$, the wavelet transform generates for each scale $j \geq 1$ a sequence of wavelet coefficients $a_j = \{a_j[k]\}_{k\in\mathbb{Z}}$ and $d_j = \{d_j[k]\}_{k\in\mathbb{Z}}$. The data a_j contain the "salient" (low-resolution) features of the signal and d_j are the residual features of the signal at this scale. The coefficients a_j and $d_j, d_{j+1}, \ldots$ are thus often referred to as the coarse and fine decomposition of the signal at scale j.

The simplest and probably most common method for signal compression via wavelet decomposition is thresholding. In thresholding one keeps only the coefficients d_j which are larger than a prespecified tolerance.[1] Thresholding applied on the coefficients of a wavelet transform is also known to be an efficient method for denoising signals with sharp transients.[2,3] Standard thresholding is typically performed using wavelets on the whole real line. This often causes significant delays in the processing, because some of the filters involved in the composition/decomposition phases of are not causal. 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 representing past, known values of the signal). In this context, any delays arising from the process of denoising will be detrimental to the performance or stability of the feedback loop. In order to minimize any delays arising from standard wavelet processing herein we propose a wavelet transform method that operates only on past data.

It should be mentioned at this point that although the word "wavelet" will be used throughout, the basic point of view taken in this paper is that of FIR filter banks.[4,5] More to the point, the various signal decomposition and reconstruction schemes described herein were inspired by the average-interpolation schemes of Donoho[6] and Sweldens;[7] see also Ref. 8. As noted in Ref. 7, this scheme can be modified to process signals on the half-axis. We explicitly (albeit briefly) show how this can be done. More details on the proposed algorithms can be found elsewhere.[9]

*Professor, Centre Automatique et Systemes, Email: francois.chaplais@ensmp.fr.
†Associate Professor, School of Aerospace Engineering. Tel: (404) 894-9526. Fax: (404) 894-2760. Email: p.tsiotras@ae.gatech.edu. Associate Fellow AIAA. Corresponding author.
‡Ph.D. candidate, School of Aerospace Engineering, Tel: (404) 894-6299. Email: dongwon_jung@ae.gatech.edu. Student member AIAA.

II. The Average-Interpolation Scheme

A very simple–yet classic–scheme for building a discrete wavelet transform with discrete vanishing moments is to use average-interpolation. The starting point of average-interpolation is the interpretation of the data samples as averages. Specifically, given a sequence a_j, average-interpolation assigns to each data sample $a_j[k]$ the dyadic interval $[2^j(k-1), 2^jk]$ of length 2^j. The discrete wavelet transform using average-interpolation then proceeds as usual and involves two steps: the *decomposition* phase, which yields the transformed signal in wavelet space, and the *reconstruction* phase, which retrieves a signal from its transform (that is, its wavelet coefficients). The decomposition and reconstruction of the signal can both be obtained using a pair of FIR filters. If there is no data processing after the decomposition phase, then this process results in perfect reconstruction: the original data sequence is recovered without error.

The whole process for a one-step decomposition/reconstruction is shown in Fig. 1. The filters $\bar{h}$ and $\bar{g}$ are the low- and high-pass (decomposition) filters and $\tilde{h}$ and $\tilde{g}$ are the low- and high-pass (reconstruction) filters. The details of the classical construction of these filters can be found in Refs. 6 and 7. Nonetheless, here we will use an alternative derivation of the transform starting from the average-interpolation framework, together with the usual FIR filter presentation, as illustrated in Fig. 1. We follow this approach because the average-interpolation presentation can be easily used to derive a similar scheme for processing signals over the half-axis which is the main objective of this paper. In addition, we later use this approach to obtain a *redundant* transform on the half-axis.

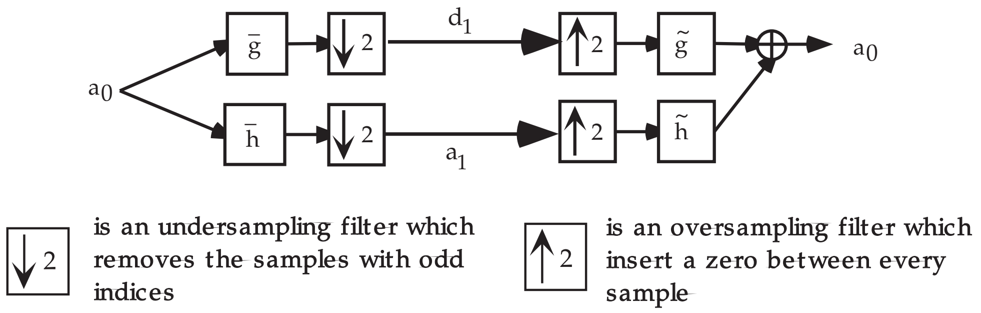

a_0
$\bar{g}$
$\downarrow 2$
d_1
$\uparrow 2$
$\tilde{g}$
a_0
$\bar{h}$
$\downarrow 2$
a_1
$\uparrow 2$
$\tilde{h}$
$\downarrow 2$ is an undersampling filter which removes the samples with odd indices
$\uparrow 2$ is an oversampling filter which insert a zero between every sample

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

A. Decomposition

Similarly to standard wavelet processing, the data is first filtered by a low-pass filter $\bar{h}$ and a high-pass filter $\bar{g}$. This separates the signal to low-frequency and high-frequency components. The construction of these filters is given next.

1. Low-Pass Filter

Up to the normalizing scalar $\sqrt{2}$, the proposed low-pass filter is the Haar decomposition filter, and computes the average of two successive signal values as follows:

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

Therefore, the low-pass decomposition filter is given by

$$\bar{h} = \left\{ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\} \tag{2}$$

which has support $[0, 1]$.

2. High-Pass Filter

We wish the high-pass decomposition filter $\bar{g}$ to have vanishing discrete moments. This means that there exists an integer N_p such that the output through the filter of all discrete polynomial sequences of the form $p[n] = \sum_{k=0}^{N_p} p_k n^k$ of degree less than or equal to N_p is zero, that is,

$$\sum_{n\in\mathbb{Z}} \bar{g}[\ell - n]\, p[n] = 0. \tag{3}$$

This vanishing moments property ensures that, at every scale j, the low-resolution part of the signal after the decomposition will retain the same number of discrete vanishing moments as the original sequence of data points. The vanishing moments property (3) of the high-pass decomposition filter also ensures that the corresponding low-pass reconstruction filter ($\tilde{h}$ in Fig. 1) will efficiently approximate sufficiently "regular" signals with an order determined by N_p. In the following we will assume that $N_p = 2$, that is, we impose that the proposed filters have three vanishing moments.

Recall first that, given a regularly spaced grid on $\mathbb{R}$ with step 2^j, say $x_{j,k} = 2^j k$ ($k \in \mathbb{Z}$), a discrete signal s_j is a polynomial if and only if it is a sequence of the averages of a continuous polynomial $p(x)$ over the intervals determined by the grid, that is, if and only if

$$s_j[k] = 2^{-j} \int_{2^j(k-1)}^{2^j k} p(x)\,\mathrm{d}x. \tag{4}$$

The proposed high-pass filter then considers the averages from (1) and identifies the unique polynomial $p(x)$ of degree less or equal to $N_p = 2$ whose averages of the coarser grid with step 2^{j+1} coincides with $N_p + 1 = 3$ 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 $\sqrt{2}$. The output of the high-pass filter is therefore the difference

$$d_{j+1}[k] = \sqrt{2}\,\bigl(a_j[2k] - p_{j+1}[k]\bigr). \tag{5}$$

Up to $\sqrt{2}$, this is the difference between the actual value of the signal and its prediction from the average-interpolating polynomial $p(x)$. This sequence of operations is illustrated in Fig. 2. Observe that the difference between the value $\tilde{p}_{j+1}[-1]$ on the left subinterval and the original value $a_j[-3]$ is the opposite of the similar difference computed on the right interval, i.e., $a_j[-3] - \tilde{p}_{j+1}[-1] = p_{j+1}[-1] - a_j[-2]$. In general,

$$a_j[2k-1] - \tilde{p}_{j+1}[k] = -d_{j+1}[k]/\sqrt{2}. \tag{6}$$

Hence, only one of these differences needs to be remembered during the algorithm.

As for the low-pass case, we can construct a filter to be used before the subsampling to perform these operations. This filter can be computed from (5) after writing down explicitly the expression for $p_{j+1}[n]$. The high-pass decomposition filter is given by

$$\bar{g} = \left\{ -\frac{\sqrt{2}}{16}, -\frac{\sqrt{2}}{16}, \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{16}, \frac{\sqrt{2}}{16} \right\}, \tag{7}$$

which has support $[-2, 3]$. It can be shown[9] that the detail coefficients are all zero when the input signal is a discrete polynomial of degree two. Therefore, this filter has three vanishing moments.

B. Reconstruction

After processing, the signal is reconstructed from the coarse approximation and the detail coefficients. The reconstruction stage is based of the same ideas as those presented in the section devoted to the high-pass decomposition filter. The reconstruction basically involves reversing the decomposition steps. During reconstruction we use the coarse scale data a_{j+1} and detail d_{j+1} to obtain the data a_j at the finer scale. Specifically, first a polynomial $p(x)$ of degree less than or equal to two is identified as in the construction of the high-pass decomposition filter, using the outputs of the decomposition low-pass filter a_{j+1}. The part of the signal a_j which has even indices is then recovered from the prediction and details using (5), whereas the part with odd indices uses equation (6). If after the decomposition the signal has not been modified, the reconstruction restores the signal exactly.

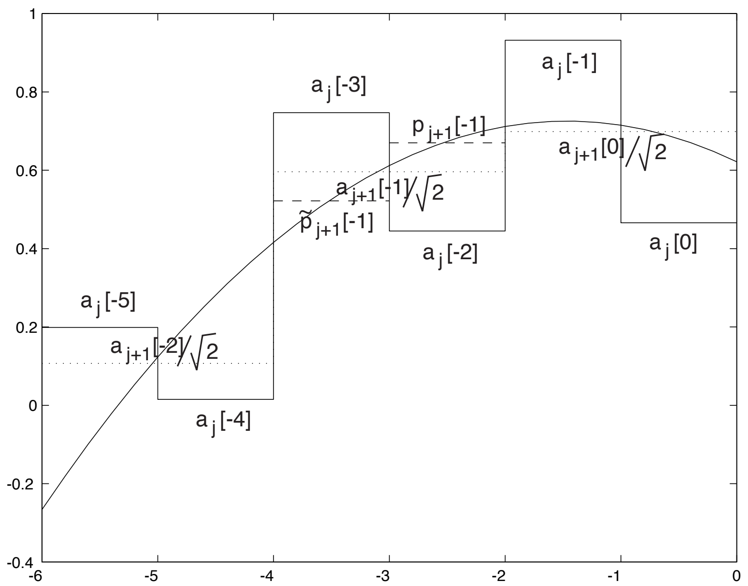


Figure 2. Average-interpolation using a polynomial of degree two.

1. Low-pass filter

The low-pass filter is constructed by writing down explicitly the two predictions $p_{j+1}[n]$ and $\tilde{p}_{j+1}[n]$ as the outputs of the low-pass reconstruction filter after a zero insertion on a_{j+1}. Using the expressions of the predictions $p_{j+1}[n]$ and $\tilde{p}_{j+1}[n]$ of the interpolating polynomial on the sub-intervals $[2^j(2k-1), 2^{j+1}k]$ and $[2^{j+1}(k-1), 2^j(2k-1)]$ yields the following value for the low-pass reconstruction filter $\tilde{h}$

$$\tilde{h} = \frac{1}{\sqrt{2}}\left\{-\frac{1}{8}, \frac{1}{8}, 1, 1, \frac{1}{8}, -\frac{1}{8}\right\},$$

which has support $[-3, 2]$.

2. High-pass filter

Up to $\sqrt{2}$, the high-pass filter outputs the detail for the even indices and the opposite of the detail for the odd indices. This leads to the Haar reconstruction filter,[4]

$$\tilde{g} = \left\{-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}, \tag{8}$$

with support $[-1, 0]$. We can finally reconstruct the signal at the fine scale as follows:

$$a_j[2n] = \sum_{p=-1}^{1} \tilde{h}[2p]\, a_{j+1}[n-p] + \sum_{p=0}^{0} \tilde{g}[2p]\, d_{j+1}[n-p] \tag{9a}$$

$$a_j[2n-1] = \sum_{p=-1}^{1} \tilde{h}[2p-1]\, a_{j+1}[n-p] + \sum_{p=0}^{0} \tilde{g}[2p-1]\, d_{j+1}[n-p] \tag{9b}$$

III. Adaptation to the Half-Axis

The discussion thus far has assumed an infinite amount of data available in both directions. Therefore, at each sample point there are enough samples to its left and to its right in order to construct the average-interpolating polynomial. This may not always be possible. For instance, for on-line applications we have only access to past data. In this section we modify the previous scheme in order to take into account the case when only past values are available for processing. We achieve this by designing a wavelet-like scheme that only processes the data $a_j[n]$ with $-\infty \leq n \leq N$. Without loss of generality, we may assume that $N = 0$. The idea here is to decenter the prediction scheme close to the right boundary so that it uses only the available values of the signal. Away from the boundary, the algorithm remains the same as before.

A. Decomposition

The low-pass averaging filter does not require any modification since it uses only past values; see equation (1). For $n < 0$ the output of the high-pass filter $d_{j+1}[n]$ uses only data with negative or zero input indices. The high-pass filter needs to be modified for the prediction at output index $n = 0$, since input data is not available at the indices $n = 1, 2$. In order to do so, the interpolating-polynomial is identified based on the three most recent averages, i.e., $a_{j+1}[-2], a_{j+1}[-1]$ and $a_{j+1}[0]$ and the prediction is now computed at the index $n = 0$ instead of index $n = -1$. This is illustrated in Fig. 3. One can compute the prediction $p_{j+1}[0]$ explicitly as

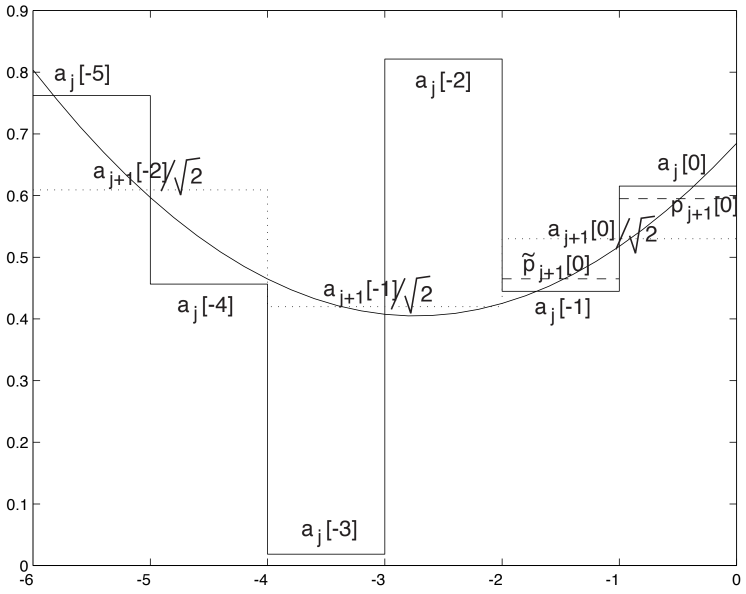


Figure 3. The average interpolation scheme is decentered to produce a prediction at the boundary. Here we assume that $j = 0$.

follows

$$p_{j+1}[0] = \sqrt{2}\left(\frac{1}{16}a_{j+1}[-2] - \frac{1}{4}a_{j+1}[-1] + \frac{11}{16}a_{j+1}[0]\right). \tag{10}$$

It follows that the high-pass output $d_{j+1}[0]$ at the boundary is given by

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

The prediction has now been decentered. Nonetheless, it is still based on average interpolation. In particular, if the input data represent a sequence of averages of a polynomial $q(x)$ on the intervals $[k2^j, (k+1)2^j]$, $k = -6, \ldots, -1$, the identified polynomial $p(x)$ will be equal to $q(x)$ and the detail $d_{j+1}[0]$ will be zero. Overall, the high-pass filter still has three discrete vanishing moments. One can show that the details are also oscillating at $n = 0$, that is,

$$\frac{1}{\sqrt{2}}d_{j+1}[0] = a_j[0] - p_{j+1}[0] = \tilde{p}_{j+1}[0] - a_j[-1] \tag{12}$$

where $\tilde{p}_{j+1}[0]$ is the prediction on the interval $[-2^{j+1}, -2^j]$ given by

$$\tilde{p}_{j+1}[0] = \sqrt{2}\left(-\frac{1}{16}a_{j+1}[-2] + \frac{1}{4}a_{j+1}[-1] + \frac{5}{16}a_{j+1}[0]\right). \qquad (13)$$

B. Reconstruction

Except for the indices $n = -1$ and $n = 0$, the reconstruction is performed exactly as before. The value $a_j[0]$ at the edge is recovered using equations (11) and (10), while $a_j[-1]$ is recovered from the formula $a_j[-1] = \tilde{p}_{j+1}[0] - d_{j+1}[0]/\sqrt{2}$, where $\tilde{p}_{j+1}[0]$ from (13).

Numerically, the reconstruction is performed at the edge by using the following equations

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

IV. Redundancy on the Half-Axis

The transforms presented in the previous sections are nonredundant, that is, any pair of sequences a_{j+1} and d_{j+1} can be interpreted as the (unique) decomposition of a signal a_j. Redundant transforms which provide more data per time step than their nonredundant counterpart can also be designed. These redundant transforms provide reconstructions which are more robust to errors and noise. The price to pay is that not all pairs of sequences a_{j+1} and d_{j+1} can be interpreted as a decomposition of the signal a_j. One must therefore be careful how to choose the pairs a_{j+1} and d_{j+1} to recover the original signal.

In this section we develop a redundant transform on the negative half-axis. This will be achieved in two steps: first, we will extend the transform of the previous section into a redundant transform at the original (fine) scale $j = 0$; second, by interpreting the zero insertion on filters (as opposed to signals) as filtering of multiplexed signals, we extend the method to higher scales.

A. Redundancy at the first scale

In order to obtain a redundant transform on the half-axis, we first keep the values with even indices of the previous transform (without subsampling). Note that keeping the odd indices also provides perfect reconstruction, provided one also knows the last decomposition output $a_1[0]$.

1. Decomposition

The filters 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 filter, and $n = -3$ for the high-pass filter. That is,

$$a_1[n] = \frac{a_0[n] + a_0[n-1]}{\sqrt{2}}, \quad n = \ldots, -3, -1 \qquad (14)$$

and $(n = \ldots, -5, -3)$

$$d_1[n] = \sqrt{2}\big(\frac{1}{16}a_0[n-3] + \frac{1}{16}a_0[n-2] - \frac{1}{2}a_0[n-1] + \frac{1}{2}a_0[n] - \frac{1}{16}a_0[n+1] - \frac{1}{16}a_0[n+2]\big). \qquad (15)$$

For the index $n = -1$, the detail $d_1[-1]$ is computed by identifying the unique polynomial $p(x)$ 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. The main difference with the boundary scheme of the previous section is that the average interpolating polynomial is not selected from disjoint intervals. This prediction scheme is illustrated in Fig. 4. The detail is now defined by

$$d_1[-1] = \sqrt{2}(a_0[-1] - p_1[-1]) \qquad (16)$$

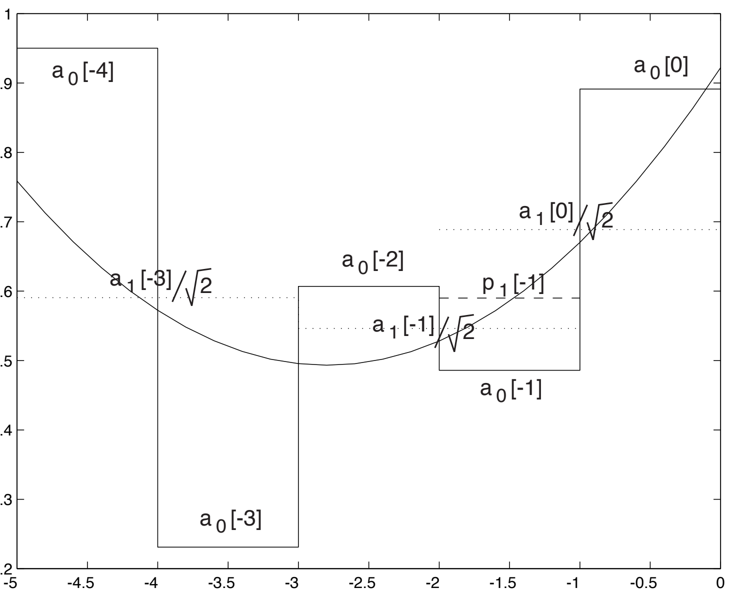


Figure 4. Computation of the prediction for the index $n = -1$. The polynomial $p(x)$ is fitted on overlapping intervals, using the values $a_1[-3]$, $a_1[-1]$ and $a_1[0]$.

The detail $d_1[-1]$ can be computed by writing down the value of the prediction $p_1[-1]$

$$p_1[-1] = \sqrt{2}\left(-\frac{1}{24}a_1[-3] + \frac{3}{8}a_1[-1] + \frac{1}{6}a_1[0]\right) \tag{17}$$

which yields, via (14),

$$d_1[-1] = \sqrt{2}\big(\frac{1}{24}a_0[-4] + \frac{1}{24}a_0[-3] - \frac{3}{8}a_0[-2] + \frac{11}{24}a_0[-1] - \frac{1}{6}a_0[0]\big). \tag{18}$$

We have just defined the outputs of the decomposition for all odd indices $n = \dots, -5, -3, -1$. Along with the output of the decomposition at the even indices $a_1[2k]$, $d_1[2k]$ obtained from the original transform, we have designed a redundant transform with values for all negative or zero integer indices.

2. 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. For indices smaller than $n = -2$, the reconstruction is performed as for the regular shift-invariant redundant transforms on the whole real line. For the indices $n = -2$, $n = -1$ and $n = 0$, the boundary filters are computed by averaging the reconstruction formulas from the even and odd subsamples.

B. Redundancy at coarser scales

The idea is to work with time-varying filters. This point of view allows us to define the boundary filters at all scales $j > 0$. First, when using wavelets on the whole real axis, the redundant transforms at the coarse scales are obtained via the standard method, that is, by inserting zeros into the original filters. Let us recall that, at the scale j, the zero insertion operation on the filter h results in the filter defined by

$$h_j[n] = \begin{cases} h[p], & \text{if } n = 2^j p, \\ h_j[n] = 0, & \text{otherwise.} \end{cases} \tag{19}$$

At the decomposition from scale $j \geq 1$ 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 $x_\ell[n] = x[2^j n - \ell]$. 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 of the signal x with the filter h_j can be split into the series of 2^j convolutions

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

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

$$y_\ell[n] = \sum_{p\in\mathbb{Z}} h_n[p]\, x_\ell[p]. \tag{20}$$

In practice, zero insertion is used away from the boundary. When close to the boundary, the multiplexed formula (20) is used on the explicit expressions of the boundary filters. A similar approach is used during reconstruction. Finally, the reconstructed signal is obtained by adding the outputs of the low-pass and high-pass filters and dividing the sum by two at each scale.

V. Experimental Results

The greatest benefit of an *on-line* denoising algorithm, as the one proposed in this paper, is the potential of its use for denoising signals in a feedback loop. In this section we report experimental results from the implementation of the previous algorithm to denoise the rotational velocity signal of a three-phase permanent magnet synchronous DC motor, which is used to drive a reaction wheel. Specifically, the DC motor used in the experiments is part of a variable-speed control moment gyro (VSCMG) actuator module[*], 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. In reaction wheel mode four wheels are available for complete control about all three axes (with one redundant wheel). Figure 5 shows the VSCMG assembly with the wheel spin motor and Fig. 6 shows the complete spacecraft simulator. Details for the design, construction and other specifications of this spacecraft simulator can be found in Ref. 10.

In reaction wheel mode control is provided by accelerating or decelerating each of the four wheels. For accurate attitude control it is necessary for the motors to be able to promptly deliver the commanded angular accelerations. Figures 7 and 8 show 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 with the IACS. It was therefore deemed necessary to implement a PI controller to achieve tight angular velocity control. The PI controller uses the angular velocity error of the wheel as an input in order to provide tight closed-loop torque (equivalently, angular acceleration) control. Before used by the PI controller, any measurement noise in the angular velocity should be removed. The wavelet algorithm developed in the previous sections was used to perform the task of on-line denoising.

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.

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

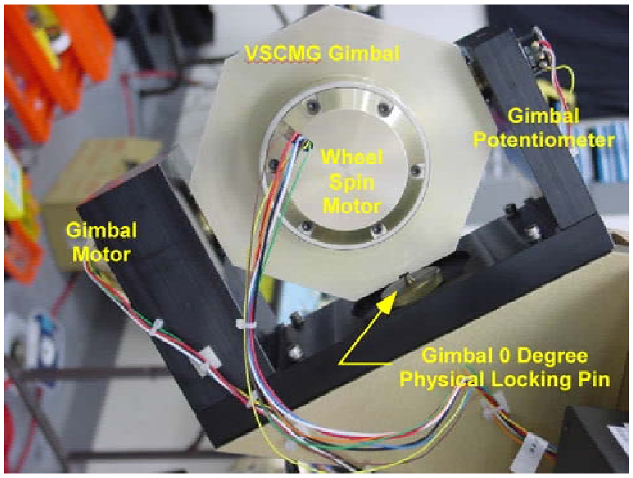


Figure 5. Main components of the VSCMG assembly. In reaction wheel mode only the wheel DC motor is active while the gimbal remains fixed.

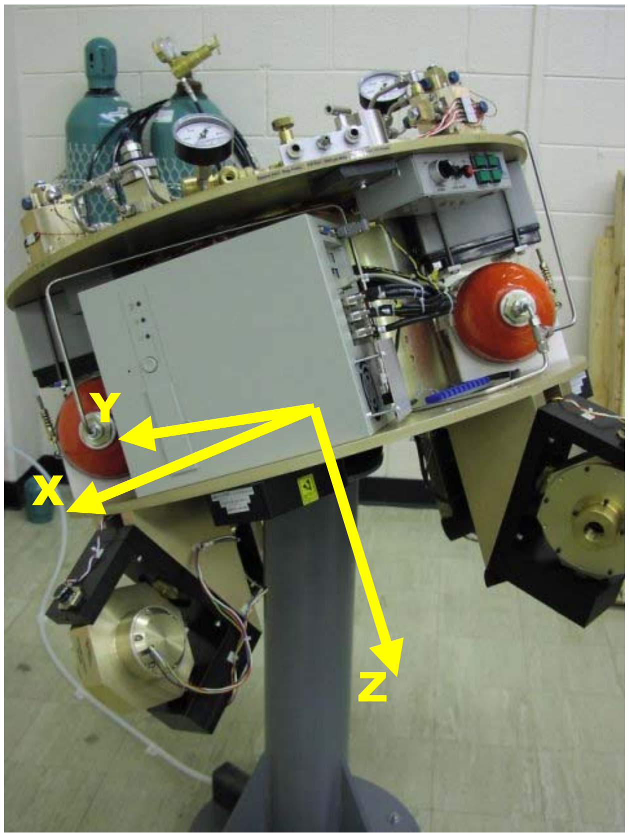
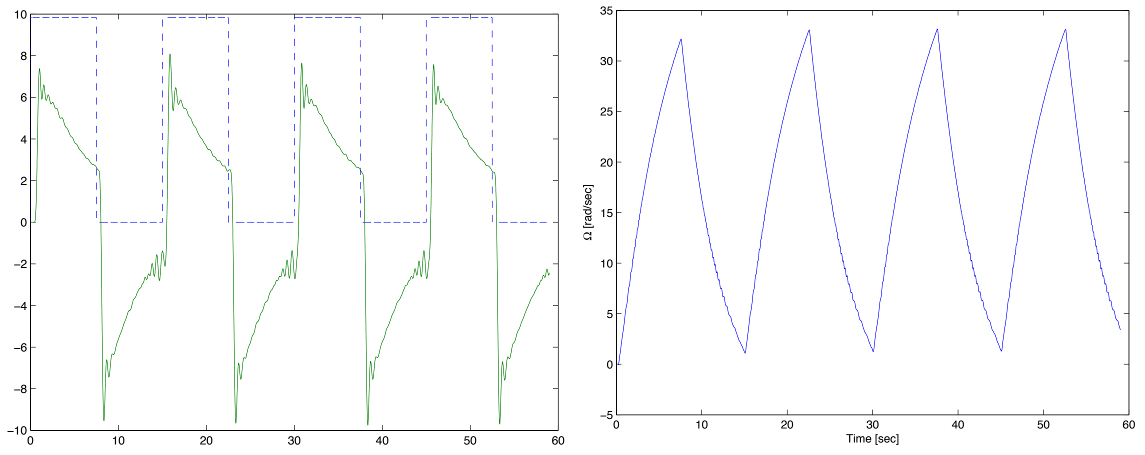


Figure 6. The body-axes definition for the IACS. The x-axis is designated as the roll axis, the y-axis is designated as the pitch axis and the z-axis is designated as the yaw axis.

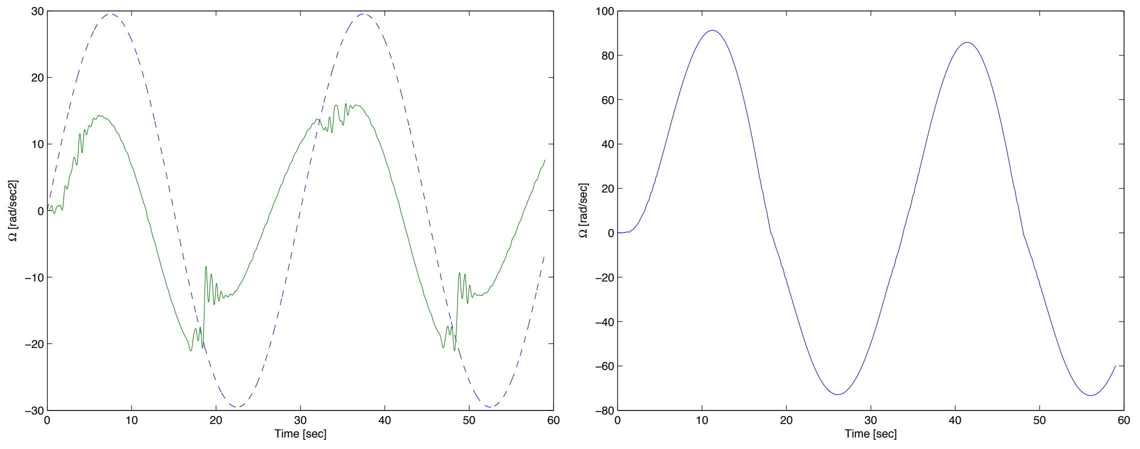
The results from two separate experiments are shown below. During the first experiment, measurements of the angular velocity signal where processed off-line using the wavelet denoising algorithm. The purpose of those experiments was to estimate appropriate values for the number of scales, threshold and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

The angular velocity measurements were initially obtained from a Hall sensor embedded in the DC motor. Figure 9 shows typical angular velocity data from the Hall sensor. The results of denoising with a



(a) Acceleration (b) Velocity

Figure 7. Open-loop response of angular acceleration and velocity to square-wave angular acceleration commands.



(a) Acceleration (b) Velocity

Figure 8. Open-loop response of angular acceleration and velocity to sinusoidal angular acceleration commands.

hard threshold of 50, delay 15 samples and for 4 and 6 scales are shown in Fig. 10.

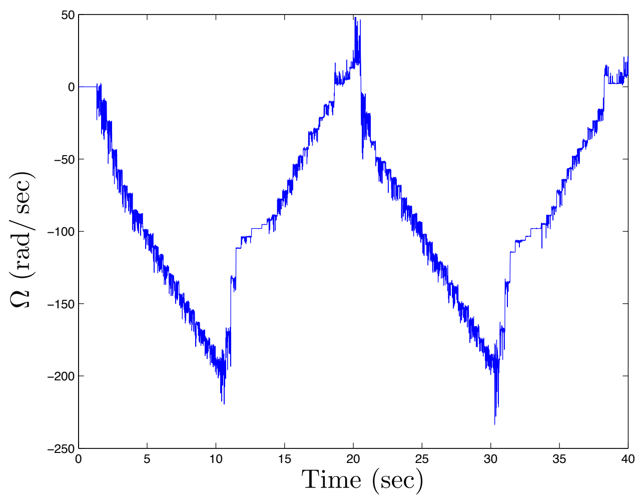
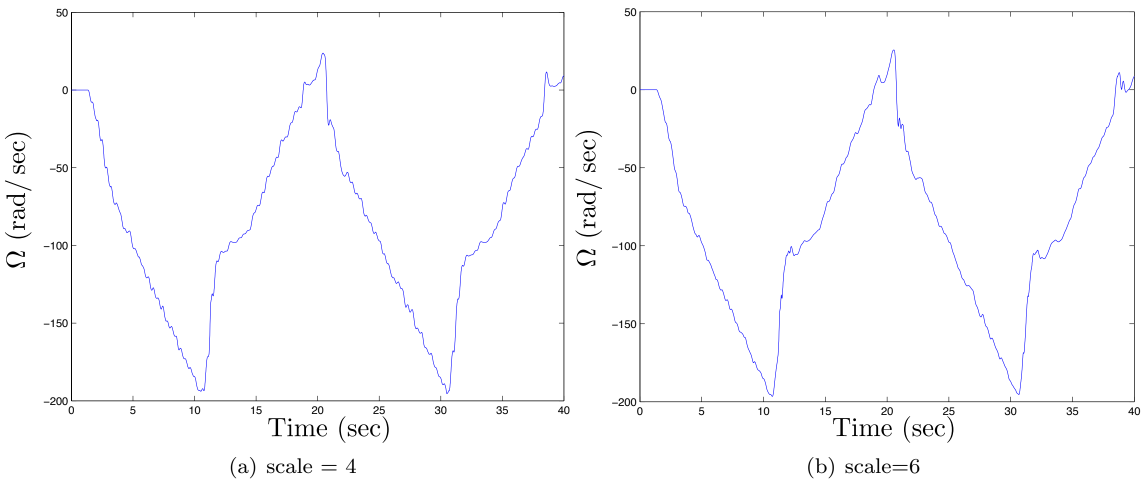


Figure 9. Angular velocity measurements from Hall sensor.



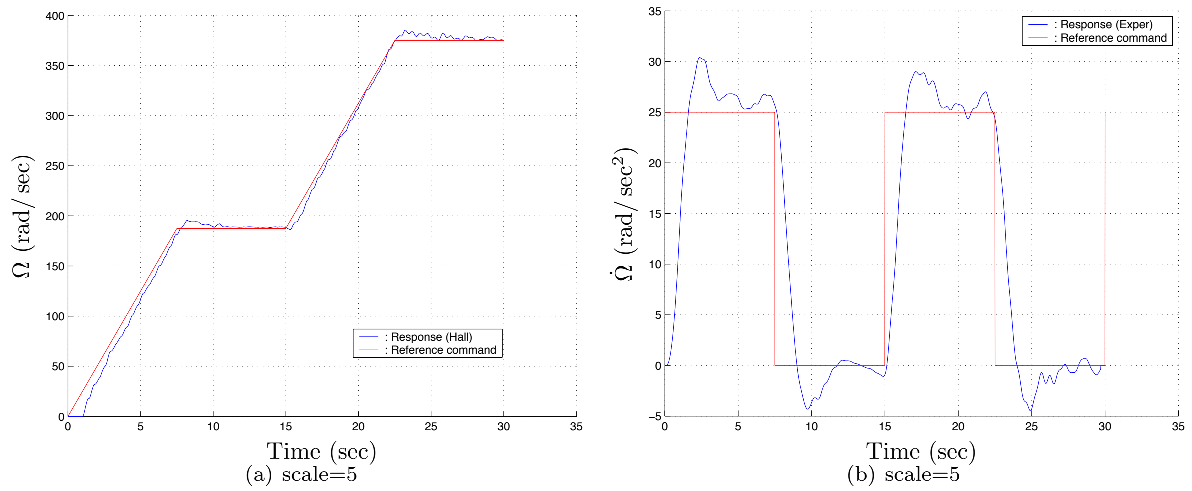
(a) scale = 4

(b) scale=6

Figure 10. Wavelet denoising of angular velocity signal from Hall sensor using a threshold of 50 and a delay of 15 samples.

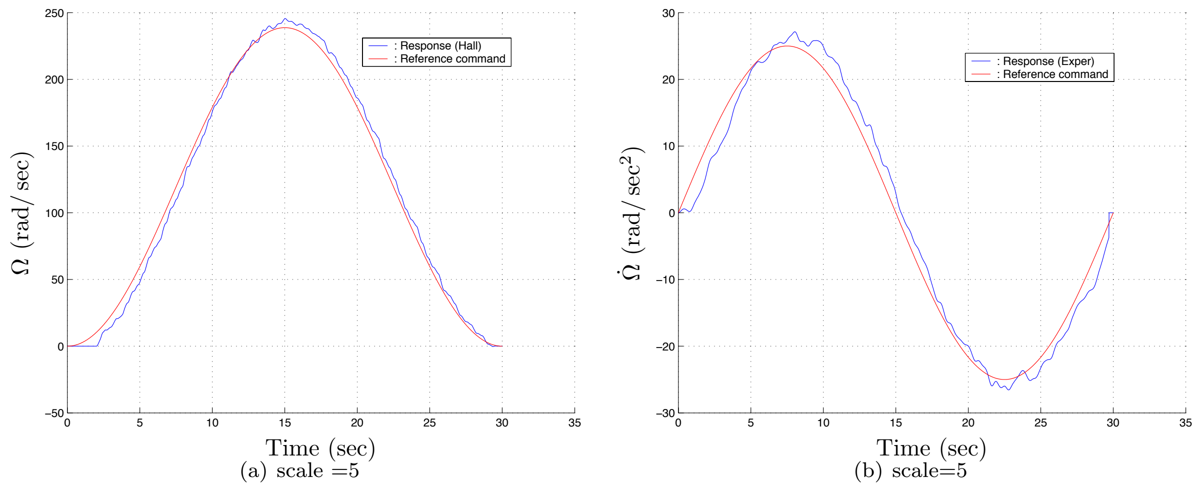
The results from closed-loop control are shown in Figs. 11 and 12. A PI control with a Smith predictor, to handle the delay, was used in these experiments. The performance is much better than the open-loop response, however, it is still not entirely satisfactory. The main reason for the unsatisfactory closed-loop performance shown in Figs. 11 and 11 is the very small update rate (about 3 Hz) of the Hall sensor measurements. This refresh rate induces severe limitations on the closed-loop bandwidth. Therefore, for better closed-loop control a set of angular encoders (US Digital E4) were installed on each wheel. Angular velocity was obtained from numerical differentiation of angular position data of the wheel spin axis provided via the encoders at a sampling rate of 100 Hz. Although having a much better performance than the Hall sensors, numerical differentiation still introduced some noise in the angular velocity signal; the purpose of denoising was therefore to remove the noise from this signal before it is used by the PI controller.

The results of this experiment for a sawtooth angular velocity input are shown in Fig. 13. The actual and filtered signal are very close to each other at this scale. Zooming in around $t = 15$ sec allows a more detailed examination of the results. Figure 14(a) shows the results of denoising over a 3 sec interval (300 samples). The dotted line shows the angular velocity data and the solid line is the result after wavelet denoising using 4 scales, a threshold of 20 and a delay of 5 samples. 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



(a) scale=5 (b) scale=5

Figure 11. Closed-loop response of angular acceleration and velocity to square-wave angular acceleration command; on-line denoising of angular velocity signal using a wavelet filter with 5 scales, a threshold of 50 and a delay of 15 samples.



(a) scale =5 (b) scale=5

Figure 12. Closed-loop response of angular acceleration and velocity to sinusoidal angular acceleration command; on-line denoising of angular velocity signal using a wavelet filter with 5 scales, a threshold of 50 and a delay of 15 samples.

has remained in the signal. This noise can be further removed at the expense of more delay. For instance, Fig. 14(b) shows the results of denoising using a delay of 15 samples or 0.15 sec. It is reminded that in contrast to other signal processing applications, delays due to signal processing in a feedback loop may lead to instability and should be avoided. A compromise must be reached between the level of acceptable delay and the requirement for noise removal.

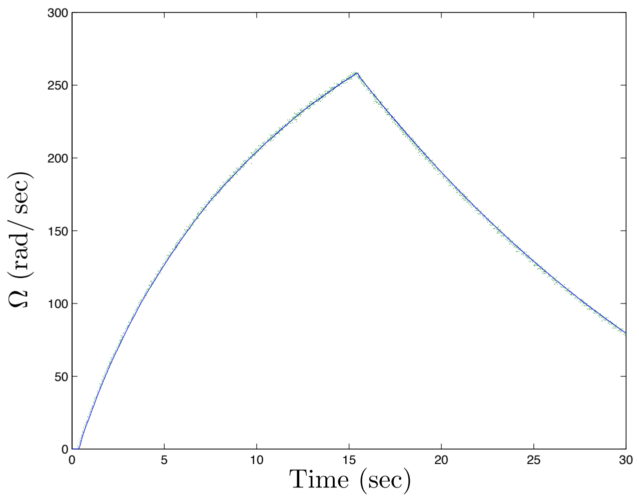


Figure 13. Wavelet denoising of angular velocity signal using 4 scales and a threshold of 20.

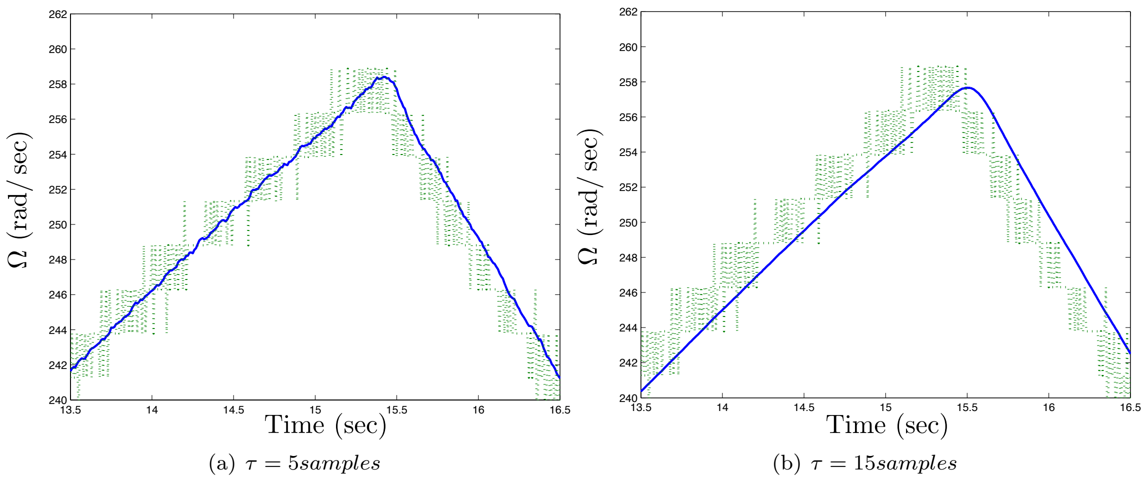


Figure 14. Wavelet denoising of angular velocity signal using 4 scales and a threshold of 20 (detailed view).

For the second experiment, the denoised angular velocity signal was 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 was achieved by a tight loop on the measured angular velocity. The block diagram schematic of the PI/Motor interconnection is shown in Fig. 15. The results from two separate angular acceleration commands are presented below. First, the results from a square-wave angular acceleration command of magnitude 10 rad/ sec^2 are shown in Fig. 16. In Fig. 16(a) the 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. 16(b). The results for a sinusoidal torque command of amplitude 15 rad/ sec^2 and frequency of 0.03 Hz are shown in Fig. 17. As before, in Fig. 17 the dotted line is the reference angular velocity, whereas the solid line is the actual response of the DC motor. Both Figs. 16 and 17 show very good tracking of the commanded signals (compare with the open-loop responses in Figs. 7 and 8.)

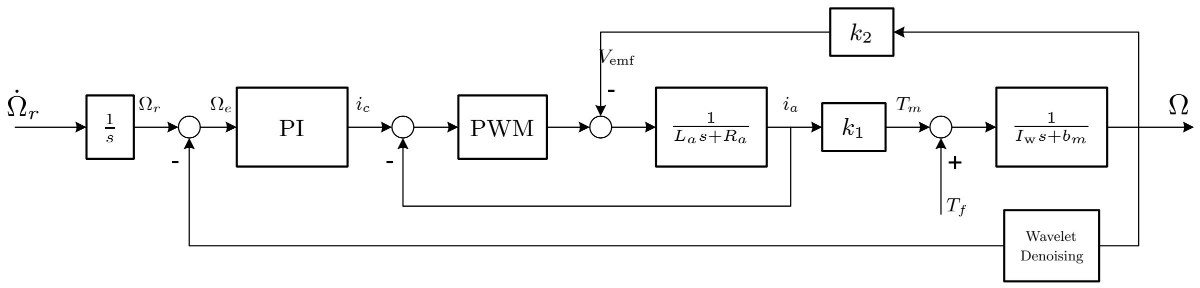
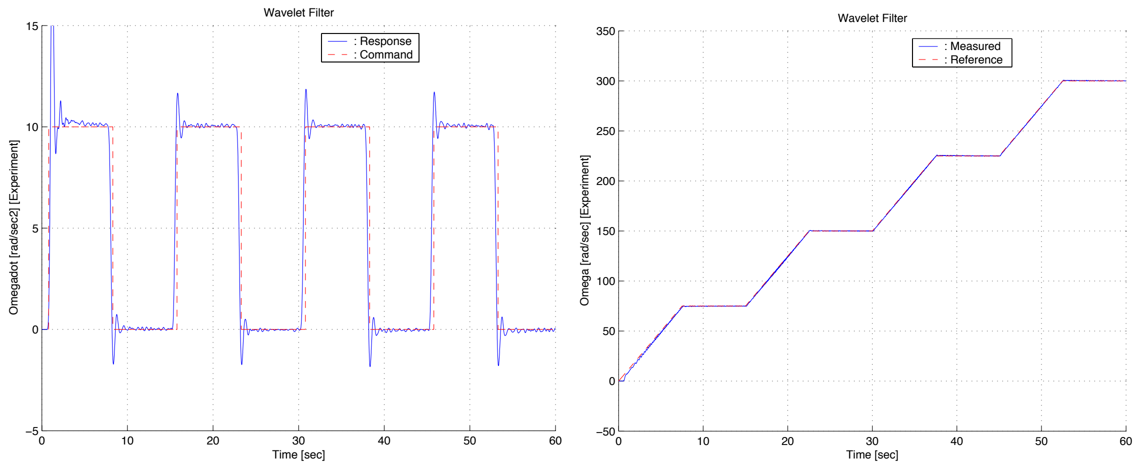


Figure 15. Closed loop control for $\dot{\Omega}$ using velocity feedback



(a) Acceleration (b) Velocity

Figure 16. 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.

VI. Conclusion

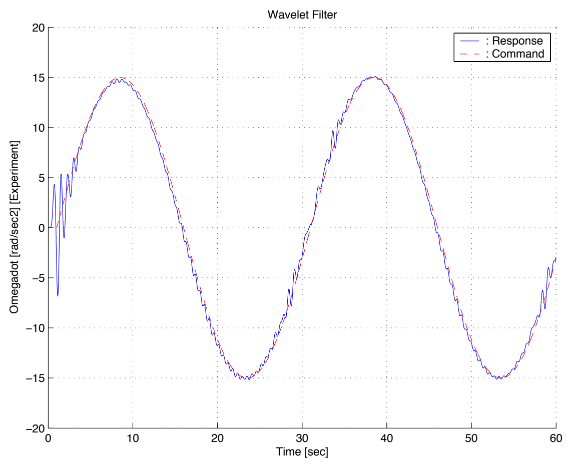
In this paper we have proposed a method for wavelet signal processing on the half-axis. The starting point for our developments is the method of average-interpolating polynomials of Donoho and the lifting scheme of Sweldens. Using this method, boundary effects arising from working in a semi-infinite 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 remove the noise from a feedback signal used for angular velocity and angular acceleration control of a brushless DC motor of a reaction wheel assembly.

Acknowledgements

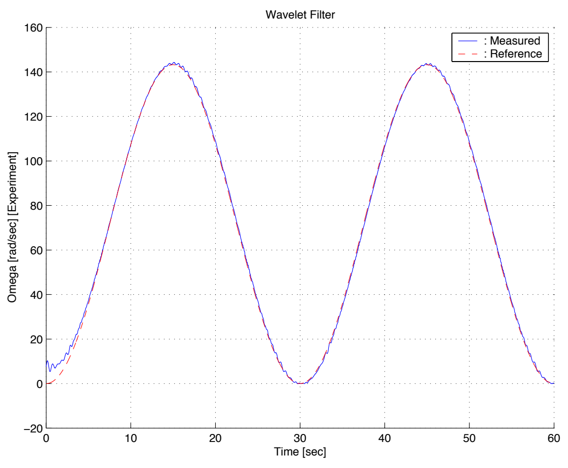
The first author wishes to thank Albert Cohen for suggesting the method used in this paper. The work of the second and third authors was partially supported by the NSF via award no. CMS-0084954. This work was performed while the second author was on a leave to CAS, ENSMP in the Spring of 2003.

References

[1]Donoho, D. and Johnstone, I., "Ideal Denoising in Orthonormal Basis Chosen from a Library of Bases," *C. R. Academy of Science*, Vol. 39, 1994, pp. 1317–1322, Serie I, Paris.



(a) Acceleration



(b) Velocity

Figure 17. 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.

[2]Donoho, D. and Johnstone, I., "Ideal Spatial Adaptation via Wavelet Shrinkage," *Biometrika*, Vol. 81, December 1994, pp. 425–455.

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

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

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

[6]Donoho, D., "Smooth Wavelet Decompositions with Blocky Coefficient Kernels," *Recent Advances in Wavelet Analysis*, edited by L. Schumaker and F. Ward, Academic Press, 1993, pp. 259–308.

[7]Sweldens, W. and Schroder, P., "Building your own wavelets at home," *Wavelets in Computer Graphics, ACM SIGGRAPH Course notes*, 1996, pp. 15–87.

[8]Harten, A., "Multiresolution Representation of Cell-Averaged Data: A Promotional Review," *Signal and Image Representation in Combined Spaces*, edited by Zeevi and Coifman, Academic Press, 1997.

[9]Chaplais, F., Tsiotras, P., and Jung, D., "Redundant Wavelet Filter Banks on the Half-Axis with Applications to Signal Denoising with Small Delays," *Proceedings, 43st IEEE Conference on Decision and Control*, 2004 (submitted), Paradise Island, Bahamas.

[10]Jung, D. and Tsiotras, P., "A 3-DoF Experimental Test-Bed for Integrated Attitude Dynamics and Control Research," *AIAA Guidance, Navigation and Control Conference*, Austin, TX, 2003, AIAA Paper 03-5331.