Proceedings of the 40th IEEE
Conference on Decision and Control
Orlando, Florida USA, December 2001

ThM13-3

Recursive frame inverse computation using wavelets on the real line

Benoît Bourrel , Ecole Polytechnique, benoit.bourrel@polytechnique.org
François Chaplais, Centre Automatique et Systèmes, École Nationale Supérieure des Mines de Paris, 35 rue Saint Honoré, 77305 Fontainebleau Cedex FRANCE, chaplais@cas.ensmp.fr

Abstract

A scale and time recursive algorithm which computes the wavelet frame inverse of a signal is proposed.

1 Introduction

1.1 Problem motivation and statement

Wavelet processing provides tools for denoising [4] piecewise regular signals and estimating their regularity [4, 2]. In control, it can be used to process the controller inputs. For instance, measures can be denoised and their regularity may be computed before their are differentiated for observation purposes [3]. To do so, the wavelet transform must be computed online, as well as the signal reconstruction.

The simplest method operates on the coefficients of the signal in a wavelet basis, using a cascade of Finite Impulse Response (FIR) filters [4]. Unfortunately, the amplitude of the wavelet coefficients largely depends of the locations of the transient patterns. This is avoided by computing a shift invariant wavelet transform. It is a redundant representation; processing it may produce data which is not itself a wavelet transform. The *frame inverse* recovers a signal by solving a least squares problem. This paper shows how to to compute the frame inverse signal recursively in time and scale. It is applied to the online denoising a a real life signal.

1.2 Dyadic shift invariant wavelet transform

Notations: If x is a discrete sequence, $x_{j\uparrow}[k]$ is defined by $x_{j\uparrow}[k] = x[p]$ if $k = 2^j p$ and $x_{j\uparrow}[k] = 0$ in the other case.

Let h and g a pair of conjugate mirror FIR filters. If J is an integer, and a_0 a discrete signal, the transform of $(a_J, d_J, \dots d_1)$ of a_0 is defined recursively by

$$a_{j+1}[p] = \sum_{n\in\mathbb{Z}} h_{j\uparrow}[n-p]a_j[n] \qquad (1)$$

$$d_{j+1}[p] = \sum_{n\in\mathbb{Z}} g_{j\uparrow}[n-p]d_j[n] \qquad (2)$$

The signal can be reconstructed by the *algorithme à trous*[4]:

$$\tilde{a}_j[p] = \frac{1}{2}\sum_{n\in\mathbb{Z}} h_{j\uparrow}[p-n]\tilde{a}_{j+1}[n] + \sum_{n\in\mathbb{Z}} g_{j\uparrow}[p-n]\tilde{d}_{j+1}[n] \qquad (3)$$

2 Recursive frame inverse on the real line

Here a sequence α_J and a family of sequences δ_j, $j = 1, \dots, J$ are given for all sample times $n \in \mathbb{Z}$. The frame inverse problem is:

$$\min_{a_0} \frac{1}{2^J}\sum_{k\in\mathbb{Z}}(a_J[k]-\alpha_J[k])^2 + \sum_{j=1}^{j=J}\frac{1}{2^j}\sum_{k\in\mathbb{Z}}(d_j[k]-\delta_j[k])^2 \qquad (4)$$

Theorem 1 *The signal $\tilde{a}_0$ generated by the algorithme à trous (3) with $\tilde{a}_J = \alpha_J$ and $\tilde{d}_j = \delta_j$ is a solution to problem (4).*

Proof: A direct computational proof can be found in [1]. At the scale J, the signal is made of 2^J non redundent representations, which provide 2^J reconstructions. Since the transforms are orthonormal, the optimum reconstruction at scale J is the average of the reconstructions, and it is computed by the algorithme à trous. ∎

3 Recursive frame inverse on the interval

The data is now only given on a finite interval $[K_-, K_+]$. The lower scale (e.g., 1) in problem (4) is allowed to vary with value j in order to find a scale recursive solution. At scale j, the problem is to find a signal $a_{j,0}$ which satisfies

$$\min_{a_{j,0}} \left[\frac{1}{2^J}\sum_{k=K_-}^{k=K_+}(a_{j,J}[k]-\alpha_J[k])^2 + \sum_{i=j+1}^{i=J}\frac{1}{2^i}\sum_{k=K_-}^{k=K_+}(d_{j,i}[k]-\delta_i[k])^2\right] \qquad (5)$$

3.1 Notations

Boundary and inner sets: since we use wavelets on the real axis to estimate a signal on an interval, boundary effects are to be expected. Let $[p_-, p_+]$ (resp. $[q_-, q_+]$) the support of h (resp. g). We will use the following adjacent intervals at the scale j: for $j = J$, $\Gamma_J = [K^-, K^+], \Gamma_J^- = \emptyset, \Gamma_J^+ = \emptyset$; for $j < J$, $\Gamma_j^- = [\phi_j^-, \gamma_j^- - 1]$, $\Gamma_j = [\gamma_j^-, \gamma_j^+]$ and $\Gamma_j^+ = [\gamma_j^+ + 1, \phi_j^+]$ with $\gamma_j^- = K_- + (2^{J-1} - 2^j)p_+ + 2^{J-1}\max(q_+, p_+)$, $\gamma_j^+ = K_+ + (2^{J-1} - 2^j)p_- + 2^{J-1}\min(q_-, p_-)$, $\phi_j^- = K_- + (2^{J-1} - 2^j)p_- + 2^{J-1}\min(q_-, p_-)$ and $\phi_j^+ = K_+ + (2^{J-1} - 2^j)p_+ + 2^{J-1}\max(q_+, p_+)$.

Boundary matrices: let $\Phi_j^- = \Gamma_j^- - K_-$ and $\Phi_j^+ = \Gamma_j^+ - K_+$; they do not depend on K_- and K_+. The symmetric matrices which computes the cost on the boundary sets are computed recursively with $M_j^\sigma(n,p) = 2^{-j+1}\sum_{k\in[0,\sigma\infty]} g_j[l-k]g_j[J-k] + 2^{-j+1}\sum_{k\in[\beta_j^\sigma,\sigma\infty]} h_j[l-k]h_j[J-k] + \sum_{m,n\in\Phi_j^\sigma} M_{j+1}^\sigma(m,n)h_j[l-m]h_j[J-n]$, with $\sigma = +$ or $\sigma = -$ and $\beta_j^\sigma = (2^{J-1} - 2^j)p_\sigma + 2^{J-1}\max(q_\sigma, p_\sigma)$.

Inner estimate: an intermediate signal $\tilde{a}_j$ is defined by $\tilde{a}_j[p] = 1/2\sum_{k\in[K_-,K_+]} g_j[p-k]d_{j+1}[k]$ $+1/2\sum_{k\in\Gamma_{j+1}} h_j[p-k]a_{j+1}[k]$ $+2^{j-1}\sum_{m,k\in\Gamma_{j+1}^+} M_{j+1}^+(m-K_+, k-K_+)h_j[p-m]a_{j+1}[k]$ $+2^{j-1}\sum_{m,k\in\Gamma_{j+1}^-} M_{j+1}^-(m-K_-, k-K_-)h_j[p-m]a_{j+1}[k]$.

3.2 Optimal signals

Theorem 2 *At the scale j the optimal signals $\breve{a}_j$ satisfy*

$$\breve{a}_j[n] = \begin{cases} \frac{1}{2^j}(M_j^-)^{-1}\tilde{a}_j[n] & \textit{if } n \in \Gamma_j^- \\ \tilde{a}_j[n] & \textit{if } n \in \Gamma_j \\ \frac{1}{2^j}(M_j^+)^{-1}\tilde{a}_j[n] & \textit{if } n \in \Gamma_j^+ \end{cases} \tag{6}$$

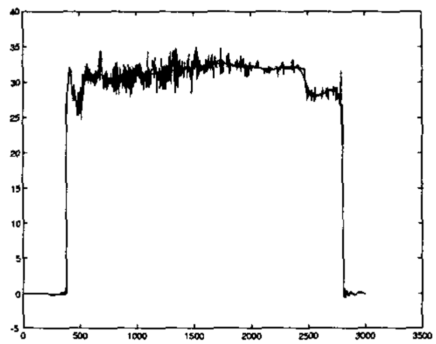
and their values for the other indices are indeterminate. If $n \in \Gamma_j$, $\breve{a}_j[n]$ is obtained by the algorithme à trous.

Proof: Details of the proof are in [1]. Theorem (1) is used when the data is far enough from the boundaries; when there are boundary effects, the *ad hoc* matrices M_j^σ are used. ∎

4 Application

At each time t, a dyadic shift invariant wavelet transform (1,2) is performed on the past noisy data. Wavelet coefficients whose magnitude is below a given threshold are set to zero to denoise the signal [4]. A signal estimation is recovered using the algorithm of theorem 2 with $K+ = t$. From this *signal* estimate at time t, a signal *value* estimate is extracted for the time $t - \tau$. If τ is small, boundary matrices are used; if τ is large enough, the algorithme à trous is used.

The figure below shows a measure from an actual plant which is controlled to produce piecewise constant outputs, and its denoising using a wavelet threshold and a reconstruction with the algorithme à trous[1]. The estimation delay due to the FIR filters is 381 time steps. Tests show that, if the delay is shortened using boundary matrices, the estimate becomes rapidly corrupted and the derivative is not reliable.



5 Conclusion

A recursive algorithm which estimates a signal in the least squares sense from pervaded wavelet data is detailed. It has been tested on real life signals; these tests indicate that the price to pay for a good estimate may be some important delays.

References

[1] B. Bourrel. Calcul récursif de frame inverse sur la transformée en ondelettes dyadique. Technical report, Ecole Polytechnique, available at `http://cas.ensmp.fr/~chaplais/FTP/Preprints/FrameInverse.pdf`, 2000.

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

[3] S. Diop, J. W. Grizzle, and F. Chaplais. On numerical differentiation algorithms for nonlinear estimation. In *Proceedings of the 39th IEEE Conference on Decision and Control,* 2000.

[4] S. Mallat. *A Wavelet Tour of Signal Processing, Second Edition.* Academic Press, 1999.

[1]The signal has a length of 2424. The transform is computed with Daubechies wavelets with two vanishing moments and over 7 scales. The threshold corresponds to an estimated variance of 9 ($\sigma = 3$). A Simulink Wavelet toolbox featuring this denoising is available at http://cas.ensmp.fr/~chaplais/FTP/Matlab_sources/SimuWave/