

Denoising via Recursive Wavelet Thresholding

by

Alyson Kerry Fletcher

A thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Electrical Engineering

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Kannan Ramchandran, Chair
Professor Laurent El Ghaoui

Spring 2002

The thesis of Alyson Kerry Fletcher is approved.

Chair

Date

Date

Date

University of California, Berkeley

Spring 2002

Denoising via Recursive Wavelet Thresholding

Copyright © 2002

by

Alyson Kerry Fletcher

Contents

1	Introduction	3
1.1	Overview	3
1.2	Previous Work	5
2	Background	7
2.1	The Discrete Wavelet Transform	7
2.2	Properties of the Wavelet Transform	10
2.2.1	Periodic Time-Invariance	10
2.2.2	Orthogonality	10
2.2.3	Vanishing Moments and Daubechies Filters	11
2.3	Denoising using Wavelet Thresholding	12
2.3.1	Basic Algorithm	12
2.3.2	Denoising Piecewise Polynomials	15
2.3.3	Notation	15
2.3.4	Threshold Selection	17
2.4	Cycle spinning	19
3	Improving Cycle Spinning through Projections	22
3.1	Problems with Cycle Spinning: An Illustrative Example	22
3.2	Wavelet Thresholding as a Projection	25
3.3	Cycle Spinning as Projection Averaging	26
3.4	Projecting to the Denoise Space Intersection	27
3.5	Characterization of the Denoise Space Intersection	31
3.5.1	Dimension Bounds	31
3.5.2	MSE: Averaging versus Projecting	34
3.5.3	Polynomial Approximations and Zero Moment Filters	34
4	Recursive Cycle Spinning	37
4.1	Basic Algorithm	37
4.2	Windowed Thresholding	39
4.3	Soft vs. Hard Thresholding	40
5	Convergence Analysis	42
5.1	Recursive Projections	42
5.2	Global Convergence of Recursive Cycle Spinning	46
5.3	Local Convergence of Iterative Cycle Spinning	49

6	Numerical Example	51
6.1	Simulation Description	51
6.2	Comparison of Methods	53
6.3	Comparison to Fourier Methods	56
6.4	Effect of Threshold Selection	57

Chapter 1

Introduction

1.1 Overview

A fundamental problem in statistical signal processing is the estimation of a signal from an observation of the signal corrupted by additive noise. The noise inherent in data-acquisition devices and analog communication systems motivated substantial body of work in this area.

In classical signal processing, it is typical to assume the signal is low-pass and the noise is not. Oversampling can ensure most of the signal frequency content is low-pass, and noise is typically broadband. The noise can then be reduced simply with a low-pass linear filter. Low-pass filtering is typically performed with a linear time-invariant (LTI) filter. Since linear time-invariant (LTI) filtering is equivalent to multiplication in the Fourier domain, all LTI techniques can be called “Fourier-based”. Low pass filtering is an extensively studied subject and covered in many classical signal processing textbooks, e.g. [25, 26, 14].

Unfortunately, many signals of interest have useful high-pass features, and simple low-pass filtering diminishes or removes these features. For example, photographic images typically have sharp discontinuities at the edges of objects. These sharp discontinuities appear as high frequency components in the Fourier-domain and would be attenuated with low-pass filtering. This attenuation would result in an undesirable blurring of the edges [19, 15].

Wavelet-based techniques offer an alternative to simple Fourier filtering that can address this problem. Wavelet denoising can provide low-pass filtering to reduce noise, while selectively preserving useful high-pass features of the signal, such as sharp discontinuities and abrupt transitions. Unlike the Fourier transform coefficients, each wavelet transform coefficient depends on the signal

only in a limited interval of time [29, 20, 27, 7].

Wavelet denoising is performed by taking the wavelet transform of the noisy signal, and then zeroing out the detail (typically high-pass) coefficients that fall below a certain threshold. An inverse wavelet transform is applied to the thresholded signal to yield the final estimate [9, 10]. As in classical low-pass filtering, zeroing out detail coefficients removes high-pass noise. However, in wavelet denoising, if the signal itself has a localized high-pass feature, such as a sharp discontinuity, the corresponding detail coefficients will have significant energy and not be zeroed out in the thresholding. In this way, wavelet denoising can low-pass filter the signal while preserving the high-pass components in selected time intervals.

Recently, a method known as *cycle spinning* has been proposed by [6] as an improvement on wavelet denoising. The wavelet transform is not time-invariant. Consequently, if the noisy signal is shifted in time, denoised and then shifted back, the result will, in general, be different from the estimate obtained from denoising without shifting. It can be shown that if the wavelet transform used in the denoising has J stages, up to 2^J different estimates can be obtained with different shifts. The cycle spinning estimate is obtained by simply linearly averaging these 2^J estimates. Heuristically, the errors in the 2^J individual estimates will not be wholly statistically dependent, and therefore, the averaging will reduce the noise power.

A natural question to ask is whether simple linear averaging is the best way to combine the information from the estimates at different shifts. The basic thesis of this work is that linear averaging is not the best way, and in fact, can be dramatically improved upon.

We reach this conclusion by considering the wavelet denoising and cycle spinning as types of projections. In wavelet denoising, selecting which wavelet coefficients are to be set to zero, in essence, identifies a subspace in which the signal of interest has most of its energy. The subspace is the set of signals having certain wavelet coefficients as zero. In zeroing out these coefficients on the noisy signal, the denoising, in effect, projects the noisy signal onto this subspace. Different time shifts identify different subspaces in which the unknown signal has most of its energy, and cycle spinning simply averages the projections onto these subspaces.

However, instead of averaging the 2^J projections, we argue it is better to project directly onto the *intersection* of the subspaces. The reason is simple. If the unknown signal has most of its energy in each of the subspaces, it will have most of its energy in their intersection. Consequently, projecting the noisy signal onto the intersection of the subspaces will remove little of the signal

energy. Now consider the noise. If the original additive noise is white, the amount of noise left after projecting is proportional to the dimension of the range space of the projection. Since the intersection of subspaces typically has a much smaller dimension than the subspaces themselves, projecting onto the intersection of the subspaces will remove much more noise than the projection onto the individual subspaces.

To perform the projection onto the intersection of the subspaces, we propose an algorithm, which we call *iterative cycle spinning*. The algorithm produces a sequence of estimates by recursively denoising with different time shifts, repeatedly cycling through the 2^J shift values.

Our main result will show that the estimates from the proposed method are guaranteed to converge to the intersection of the range spaces of the 2^J denoising projections. Our global analysis does not require any assumption on the initial condition of the algorithm. The global convergence result is non-trivial since the algorithm is highly non-linear due to the thresholding. Local convergence results are also proven. Convergence to the projection of the original noisy signal onto the above-stated intersection can be achieved when the initial signal has sufficiently small, yet typical, additive noise.

Imposing certain initial conditions on the signal yields not only convergence, but convergence to the projection of the original noisy signal onto the above-stated intersection. The necessary conditions are satisfied when the initial additive noise is small, yet typical, in magnitude. This projection is the closest point to the noisy signal in the space.

In addition to the theoretical analysis, numerical simulations of the algorithm denoising piecewise polynomial signals are presented. For these signals, it is shown that in a small number of iterations iterative cycle spinning offers significantly better noise reduction than standard cycle spinning. A comparison with Fourier techniques is also shown, along with studies of threshold values, filter bank stage numbers, and filter order choices.

1.2 Previous Work

Denoising by wavelet thresholding was introduced by Donoho and Johnstone in [10]. Their method of using “soft thresholding” was referred to as “wavelet shrinkage”. The basic idea of setting fine-scale wavelet coefficients to zero as a noise reduction method was not revolutionary. However, earlier and contemporaneous works on this method, *e.g.* [31], emphasized the need to separate “edge” and “non-edge” locations and threshold only the fine-scale coefficients in non-edge locations. Techniques

for this separation were effective but decidedly ad hoc. Donoho and Johnstone’s wavelet shrinkage was remarkable for its simplicity, requiring no tracking or correlation of maxima and minima across scales.

Wavelet shrinkage drew attention not only from engineers for its simplicity and effectiveness, but also from statisticians and mathematicians for its elegance and the precise asymptotic statements that could be made. Donoho [9] provided a clear and precise interpretation of wavelet shrinkage as the minimization of a mean-squared error (MSE) subject to a smoothness condition. Further theoretical interpretation of wavelet shrinkage and another threshold calculation method are given in [2].

Cycle spinning is central to this thesis and thus is explained in greater detail in Section 2.4. An erroneous motivation for cycle spinning is that averaging N noisy observations generally reduces the variance of the noise by a factor of N . The suboptimality of cycle spinning is related to the fact that the different phases are not corrupted by *independent* noise. Another wavelet thresholding work by Chang *et al.* [5] looks at multiple copies of a single image each corrupted by independent noise. There too simple averaging is suboptimal.

Other bases and estimation techniques The suitability of a basis for denoising by thresholding depends on the sparseness of the signal in the basis. Krim *et al.* [17] report improved denoising by applying thresholds in an adaptive best basis instead of in a fixed wavelet basis or set of wavelet bases, and Strela *et al.* [28] obtain improvements by replacing the wavelet transform with a multi-wavelet transform. A translation-invariant multiwavelet-based method is given in [1]. Thresholding of pyramid transform coefficients is considered in [22]. A novel way to combine features across scales is given in [11].

Finally, it should be noted that there is growing literature inspired by wavelet thresholding but using minimum MSE estimation of wavelet coefficients. This estimation depends on developing a statistical model for the coefficients. Amongst the best results are those reported in [23, 18, 12]

Chapter 2

Background

2.1 The Discrete Wavelet Transform

The simplest way to define the discrete wavelet transform (DWT) is to describe the computational structure used to calculate it. The DWT is most easily computed by an iterated filter bank; imposing specific conditions on the filters yields a wavelet transform, imposing further conditions determines the properties of that transform, such as orthogonality and frequency selectivity [29, 21, 27, 8].

As a precursor to the discrete wavelet transform, consider the two-channel filter bank depicted in Figure 2.1. The left half is denoted the *analysis filter bank*, and correspondingly, the right the *synthesis filter bank*. In the analysis filter bank, the input x is passed through two filters h_0 and h_1 and both filter outputs are downsampled by a factor of 2 producing the two outputs X_0 and X_1 . The synthesis filter bank recombines X_0 and X_1 by upsampling each of these by 2, filtering respectively by g_0 and g_1 , and then summing. This final output is designated \hat{x} .

For this structure to be of any interest, the filters h_0 and h_1 must be of different frequency selectivity. The upper branch h_0 is arbitrarily selected as having the lower passband of the two.

Under certain conditions on the filters, the output \hat{x} equals the input x for every input. The transform from x to (X_0, X_1) can then be inverted. Specifically, the synthesis filter bank inverts analysis filter bank. The analysis-synthesis pair is then called a *perfect reconstruction* filter bank.

These filter restrictions can be expressed in many forms, but time- and z -domain representations are the most transparent and useful for this work. The following conditions are equivalent to the

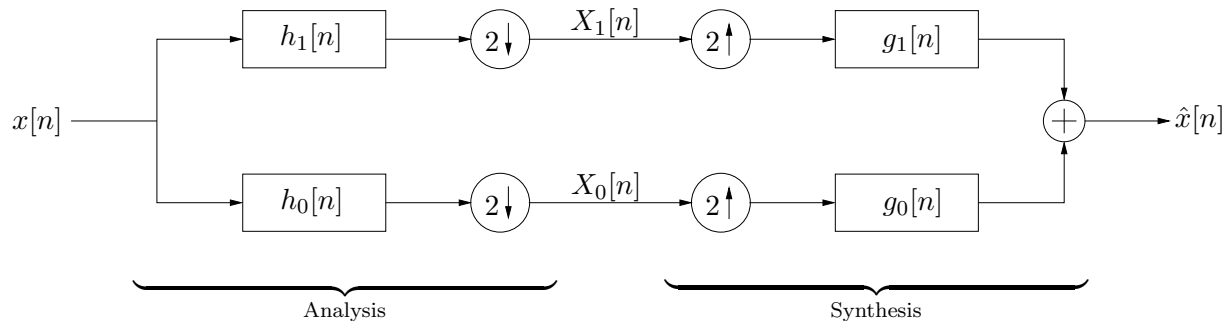


Figure 2.1: Block diagram of a two-channel filter bank.

perfect reconstruction property.¹

a) Frequency domain:

$$\begin{aligned} G_0(z)H_0(z) + G_1(z)H_1(z) &= 2 \\ G_0(z)H_0(-z) + G_1(z)H_1(-z) &= 0. \end{aligned}$$

b) Time-domain: For $i, j = 0, 1$, the convolution of the impulse responses of the filters satisfy

$$(h_i \star g_j)[2n] = \delta[i - j]\delta[n],$$

where $\delta[m]$ is the delta function: $\delta[m] = 1$ for $m = 0$ and $\delta[m] = 0$ for all $m \neq 0$.

With the convention above of h_0 being a low-pass filter, in a perfect reconstruction filter bank, g_0 must be a low-pass filter also, and h_1 and g_1 are forced to be high-pass.

Discrete wavelet transforms are generated via cascaded applications of the analysis filter bank part of a perfect reconstruction filter bank. The computational structure is shown in Figure 2.2. The output of the analysis low-pass filter is the input into another two-channel analysis bank. A J -stage wavelet transform can be produced with J such analysis banks iterating on the low-pass branches as shown. The final output consists of the outputs of the final stage low-pass filter and each of the J high-pass filter outputs.

The outputs of the branches are called *subbands*. For a J -stage filter, there are $J + 1$ subband outputs, which will be denoted X_0, \dots, X_J . The coefficients of the subband, X_0 , resulting from the J successive low-pass filter stages are called the *scaling coefficients*. The coefficients in the other J subbands, X_1, \dots, X_J , are called *detail coefficients*.

¹The convention of replacing a lower-case letter by its capital version to indicate the transformation of a time-domain sequence to z -domain function is followed throughout this document.

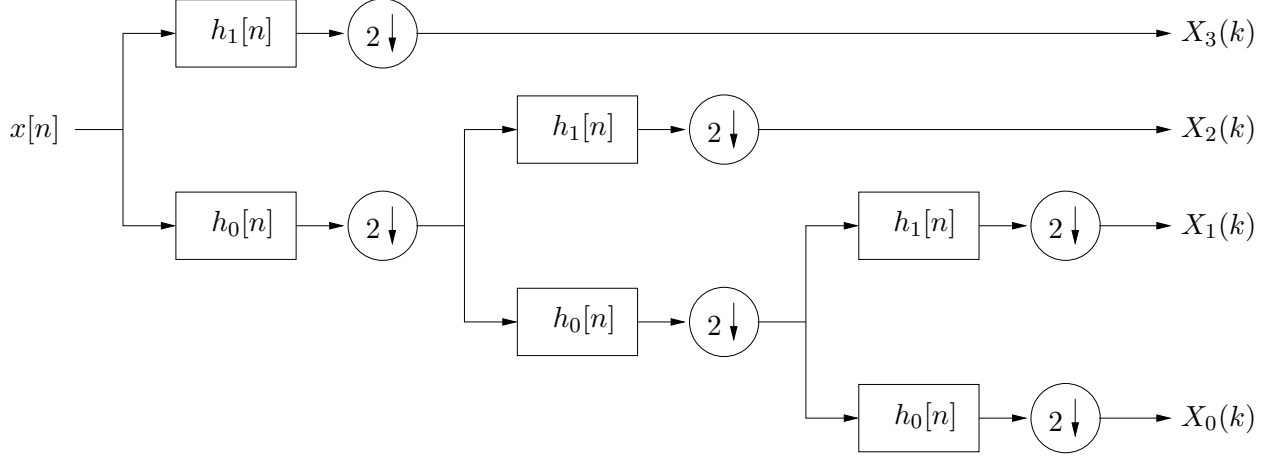


Figure 2.2: Block diagram of a filter bank implementation of a three-stage discrete wavelet transform.

Due to the downsampling, the subbands have different data rates and contain the information representing different frequency bands. The scaling subband, X_0 is $1/2^J$ times the original data rate of x . For $i > 0$, the data rate of the i -th detail subband, X_i , has a data rate of $1/2^{J+1-i}$ times the original data rate. The total data rate of the subband outputs is the same as the original rate. Also, assuming for convenience that h_0 is an ideal half-band low-pass filter, the signal X_0 represents the frequency band $[0, \pi/2^J]$, and for $i > 0$, X_i represents the frequency band, $[\pi/2^{J+1-i}, \pi/2^{J-i}]$. Hence, the label of subband.

In the remainder of this work, we will restrict our attention to finite length, discrete-time signals. Specifically, for a J -stage wavelet, we will assume the input x is an N -length discrete-time signal, where N is divisible by 2^J . In this case, the scaling coefficient output X_0 will have $N/2^J$ coefficients, denoted $X_0(k)$, $k = 0, \dots, N/2^J - 1$. For $i > 0$, X_i will have $N/2^{J+1-i}$ coefficients. We will use the notation, X , to denote the stacked vector of the $J + 1$ subband coefficients,

$$X = [X_0 \ X_1 \ \dots \ X_J].$$

The wavelet transform X can thus be considered a N -length vector, and we will let \mathcal{W} denote the wavelet transform operator from x to X : $X = \mathcal{W}(x)$.

Wavelet transforms can also be performed on continuous-time signals, although we will not consider the continuous-time transform here. The interested reader is referred to [8, 21, 29]. This work utilizes only the discrete wavelet transform.

2.2 Properties of the Wavelet Transform

2.2.1 Periodic Time-Invariance

Although it is implemented with linear time-invariant filters, the wavelet transform is not time-invariant. This is due to the downsampling in the analysis bank. However, the wavelet transform does retain the property of periodic time-invariance.

An operator \mathcal{O} is *periodically time invariant with period T* if it satisfies the property:

$$\mathcal{O}(x[n]) = y[n] \quad \Rightarrow \quad \mathcal{O}(x[n - kT]) = y[n - kT] \quad \text{for every } k \in \mathbb{Z}.$$

An operator is *time invariant* if it is periodically time invariant with period 1.

It is easy to verify that downsampling by a factor of K followed by upsampling by a factor of K is a periodically time-invariant operation with period of K . This operation is not periodically time-invariant with period T for any T that is not a multiple of K ; using the notation above, the sample $x[T]$ is discarded when the down- and upsampling are done without shifting but is retained when the input is first shifted by T . Thus downsampling by $K > 1$ followed by upsampling by K is not a time-invariant operation.

In a J -stage DWT, the lowest rate subbands have been decimated by a factor of $M = 2^J$. The transform is thus periodically time invariant with period $M = 2^J$.

2.2.2 Orthogonality

Orthogonal transforms are an important class of linear transforms. In general, a linear transform is *orthogonal* if it has a matrix representation T with $T^{-1} = T'$, where T' denotes the transpose of T .

Two properties of orthogonal transforms are used repeatedly in this work:

- a) Orthogonal transforms are Euclidean-distance preserving. If T is orthogonal, $\|T(x - y)\| = \|x - y\|$ for any two vectors x and y .

This property is useful for denoising, since it implies that the distance between two vectors is the same in transform domain as the original space. Consequently, a quantity such as an error between a true signal and an estimate can be evaluated equally in either space.

- b) Orthogonal transformations of a white noise are white. A zero-mean random vector $\eta \in \mathbb{R}^N$ is a white noise if its autocorrelation matrix $E[\eta\eta'] = \sigma_\eta^2 I_N$ for a scalar σ_η . If T is orthogonal,

$\nu = T\eta$ is also a white noise because

$$E[\nu\nu'] = E[T\eta\eta'T'] = T E[\eta\eta'] T' = \sigma_\eta^2 I_N.$$

This property is important for removing additive white noise because it allows one to consider the noise as added and removed in the transform domain.

Another important property of orthogonal transforms that is not used explicitly in this work but is important in any numerical experimentation is that orthogonal transforms are numerically well conditioned. Not only is the inverse of an orthogonal transform T known to be T' without requiring any computation, but solving $Tx = y$ for x is always well conditioned. Discrete wavelet transforms are, by the iterated filter bank construction, always invertible without explicit computation of a matrix inverse.

A discrete wavelet transform can be made orthogonal by imposing certain restrictions on the filters. Specifically, it can be shown that a wavelet transform is orthogonal if and only if the filter transfer functions satisfy

$$H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1}),$$

and

$$H_1(z) = -z^{2^k+1}H_0(-z^{-1})$$

for some integer k . In the remainder of this work, we will restrict our attention to orthogonal wavelet transforms.

2.2.3 Vanishing Moments and Daubechies Filters

A filter $h[n]$ is said to have p *vanishing moments* if $h[n] \star x[n] = 0$ for any polynomial x of degree less than p . In frequency domain, this property corresponds to the filter $H(z)$ having p zeros at $z = 1$. A wavelet transform is said to have the p vanishing moments if its high-pass filter $H_1(z)$ has p vanishing moments.

Wavelet transforms with vanishing moments have the useful property that the detail coefficients of polynomials are zero. Specifically, suppose that \mathcal{W} is a wavelet transform with p vanishing moments with any number of stages J , and x is a polynomial input to the transform with degree less than p . It can be verified that downsampling and filtering any polynomial will remain a polynomial with the same degree. Therefore, if x is the input to the transform \mathcal{W} , the input to the high-pass filter $H_1(z)$ in each detail coefficient subband will be a polynomial of degree less than p .

Since $H_1(z)$ has p vanishing moments, the output of the filter will be zero. Consequently, all the detail coefficients of $\mathcal{W}(x)$ will be zero for any polynomial of degree less than p . This vanishing moment property will be useful later in denoising piecewise polynomials.

Orthogonal wavelet transforms can be constructed with an arbitrary number of vanishing moments. The most common construction is through the *Daubechies filters*. For any p , the Daubechies filter pair, denoted D_p , is a pair of filters $(H_0(z), H_1(z))$ such that the resulting wavelet transform has exactly p vanishing moments. The D_p filters are FIR with length $2p$ and have a cutoff frequency of $\pi/2$: $H_0(z)$ is low-pass with passband $[0, \pi/2]$ and $H_1(z)$ is a high-pass filter with passband $[\pi/2, \pi]$. The Daubechies filters have the property that the filter pair D_p are the minimum length orthogonal FIR filters with p vanishing moments. The details of the construction of the filters are given in [8, 21]. For $p = 1$, the Daubechies filter pair D_1 is given by

$$H_0(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}), \quad H_1(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}).$$

This filter pair, D_1 , is also often called the *Haar wavelet*. For $p = 2$, the Daubechies pair, D_2 , is

$$\begin{aligned} H_0(z) &= \frac{1}{4\sqrt{2}}(a_0 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3}) \\ H_1(z) &= \frac{1}{4\sqrt{2}}(-a_3 + a_2z^{-1} - a_1z^{-2} + a_0z^{-3}) \end{aligned}$$

where

$$a_0 = 1 + \sqrt{3}, \quad a_1 = 3 + \sqrt{3}, \quad a_2 = 3 - \sqrt{3}, \quad a_3 = 1 - \sqrt{3}.$$

These filters, and the filters for higher values of p , are given in [8, 21].

2.3 Denoising using Wavelet Thresholding

2.3.1 Basic Algorithm

Denoising refers to the general problem of estimating an unknown signal from observations of the signal corrupted by additive noise. In this work, x will denote an unknown signal to be estimated, and y will denote the signal x with noise

$$y[n] = x[n] + d[n], \quad n = 0, \dots, N - 1,$$

where d is the additive noise. The goal of denoising is to find an estimate \hat{x} of x from the noisy signal y .

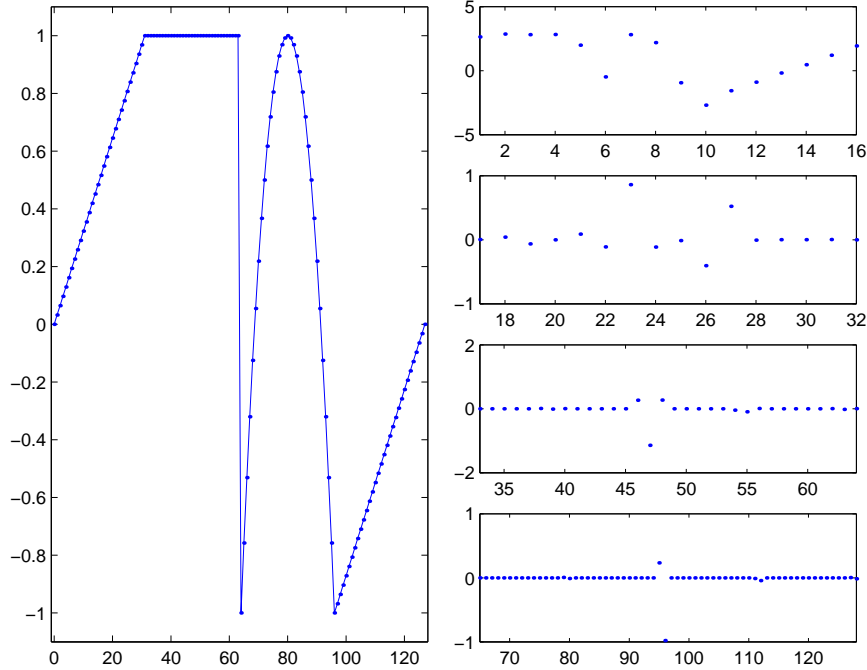


Figure 2.3: Wavelet transform of a piecewise polynomial signal with the transform coefficients arranged in subbands.

As discussed earlier, the most common method for denoising is low-pass filtering of the noisy signal y . The idea is that the true signal, x , is assumed to be mostly low-pass. Therefore, low-pass filtering preserves the signal energy while eliminating any noise in the high frequencies. Unfortunately, the true signal x may also have certain high-pass features, such as discontinuities, and low-pass filtering will attenuate these features. For example, images have discontinuities at the boundaries of objects, and low-pass filtering will have the undesirable effect of blurring the edges at these locations.

Wavelet thresholding is a method for denoising signals that can selectively locate and preserve important high-pass features of the signal. The wavelet thresholding estimate for x given y can be obtained in three steps:

1. The noisy signal, y , is first wavelet transformed to yield wavelet coefficients, $Y = \mathcal{W}(y)$, where \mathcal{W} denotes the wavelet transform operator.
2. All the detail wavelet coefficients of Y are zeroed out, except coefficients that are sufficiently large. The wavelet coefficient vector after zeroing out is denoted \hat{X} and can be described by

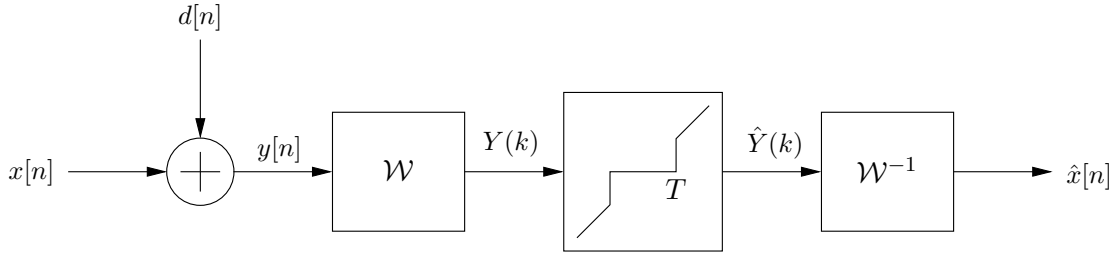


Figure 2.4: Block diagram depicting denoising via hard thresholding of wavelet coefficients.

the thresholding operation,

$$\hat{X}(k) = \begin{cases} Y(k) & \text{if } |Y(k)| > T(k) \\ 0 & \text{if } |Y(k)| \leq T(k) \end{cases}$$

where the values $T(k)$ are the *threshold levels*. In general, the threshold levels $T(k) = \infty$ for all the scaling coefficients k , and $T(k)$ is finite, but non-zero when k is a detail coefficients. In this way, all the scaling coefficients are kept along with any detail coefficients that are sufficiently large. The threshold level values are adjustable parameters of the denoising algorithm and will be discussed momentarily.

3. The final estimate, \hat{x} , is the inverse wavelet transform of the thresholded coefficients: $\hat{x} = \mathcal{W}^{-1}(\hat{X})$.

The basic assumption in the algorithm is that the energy of the true signal, x , in the detail subbands is concentrated in a few large coefficients. The remaining detail coefficients are assumed to be small. Under this assumption, the threshold levels could, in principle, be set such that the small detail coefficients fall below the threshold and are set to zero, while the large coefficients lie above the threshold and are preserved. Since the true signal, x , is assumed to have little energy on the small coefficients, the small coefficients of y should contain mostly noise. Therefore, setting these coefficients to zero removes mostly noise while removing little energy of the true signal. On the other hand, the large detail coefficients of y are likely to contain a large component due to the true signal x and will be kept. In this way, the estimate \hat{x} is obtained from the original signal y by removing the noise in the small detail coefficients, while selectively preserving the components in the large detail coefficients. Since the detail coefficients generally contain the high-pass content of the signal, we could say that wavelet thresholding low-pass filters the noisy signal, except at selected points where large high-pass features are detected.

One situation that wavelet thresholding is particularly well-suited to is the estimation of a “piecewise low-pass” signal. That is, the signal to be estimated, x , is low-pass signal except for some occasional discontinuities.

Since the detail coefficients of a signal generally contain the signal’s high-pass content and x is piecewise low-pass, the detail coefficients of x should be small, except possibly for the coefficients effected by the discontinuities. In general, the wavelet transform filters are FIR with short length. Therefore the effect of each discontinuity will appear in only a small number of detail coefficients. Consequently, all the detail coefficients of x should be small except for a few points due to the signal discontinuities. Thus, if the threshold levels are set correctly, the small coefficients will be set to zero, and the large coefficients due to the discontinuities will be preserved. In this way, the thresholding will ideally low-pass filter the noisy signal, except at the discontinuities, where the high-pass content will be preserved. The thresholding is thus able to low-pass filter the signal to remove noise, while maintaining the high-pass content at the discontinuities.

2.3.2 Denoising Piecewise Polynomials

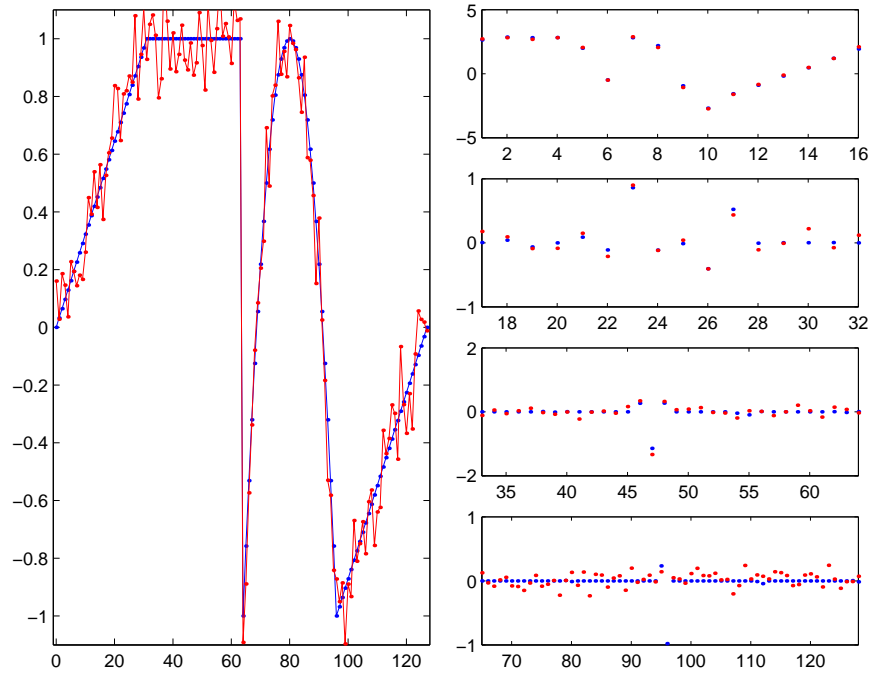
One important application of wavelet thresholding is the estimation of piecewise polynomials. In this case, the unknown, true signal, x , is assumed to be a piecewise polynomial of degree less than d , for some known value d . The wavelet transform used in the denoising is selected to have at least d vanishing moments. The construction of such wavelet transforms was discussed above in Section 2.2.3.

With the vanishing moment property, the detail coefficients of the true signal x will be exactly zero, except at the discontinuities. Therefore, the corresponding coefficients of the noisy signal y will contain only noise. Thus, if the coefficients are set to zero in the thresholding, the thresholding will remove only noise energy and no signal energy.

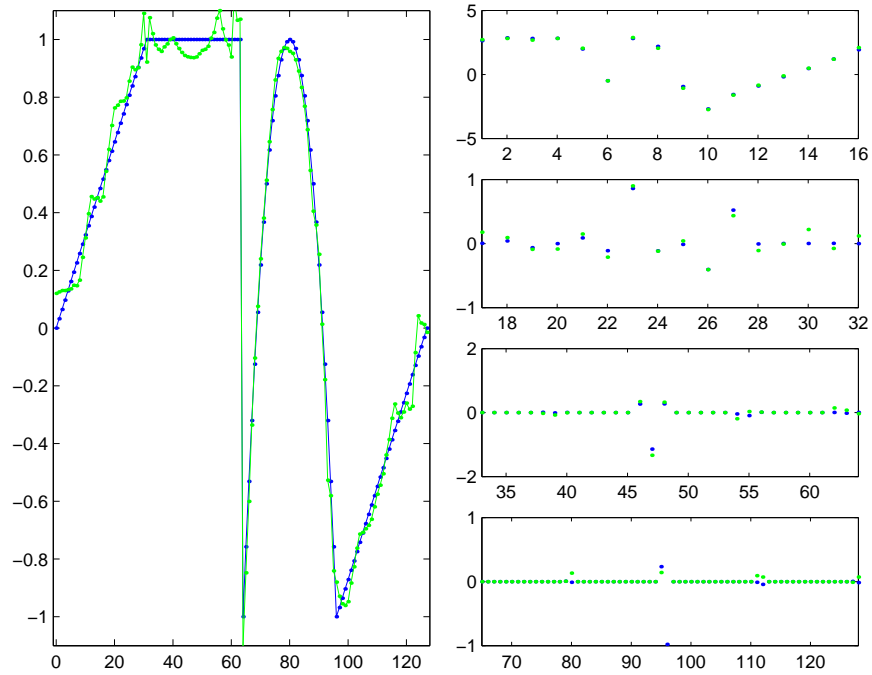
Thus, wavelet denoising with piecewise polynomials with vanishing moment wavelet transform represents an ideal case where the thresholding can be performed without removing any signal energy.

2.3.3 Notation

At this point, we introduce some notation that will be useful for the remainder of this work. Referring to Section 2.3.1, given a noisy signal y , we will let $\alpha(y)$ denote the set of coefficients that



(a)



(b)

Figure 2.5: Example of denoising a piecewise polynomial signal with wavelet thresholding. (a) Original signal and noisy signal with 17.1 dB SNR. (b) Hard thresholding increases the SNR to 22.9 dB.

are set to zero in the thresholding. That is,

$$\alpha(y) = \{ k : |Y(k)| < T(k) \},$$

where $Y = \mathcal{W}(y)$ are the wavelet transform coefficients of y . We let Λ_α denote a *hard thresholding operator* with zero set α ,

$$(\Lambda_\alpha Y)(k) = \begin{cases} Y(k) & \text{if } k \notin \alpha \\ 0 & \text{if } k \in \alpha \end{cases}$$

With this notation, the thresholding step of denoising can be written,

$$\hat{X} = \Lambda_{\alpha(y)} Y.$$

We will let D the wavelet denoising operator, $\hat{x} = D(y)$. Since denoising is performed by taking the wavelet transform, thresholding, and then taking the inverse transform, the denoising operator is given by

$$\hat{x} = D(y) = \mathcal{W}^{-1} \Lambda_{\alpha(y)} \mathcal{W}(y).$$

2.3.4 Threshold Selection

Since the wavelet transform is linear, the wavelet transform of the signal $y[n] = x[n] + d[n]$ is given by $Y(k) = X(k) + D(k)$, where $X = \mathcal{W}(x)$ and $D = \mathcal{W}(d)$. That is, each individual wavelet coefficient $Y(k)$ has a signal component $X(k)$ and a noise component $D(k)$. In wavelet thresholding, $X(k)$ is estimated either as $\hat{X}(k) = Y(k)$ or as $\hat{X}(k) = 0$. The squared error for this coefficient is

$$|X(k) - \hat{X}(k)|^2 = \begin{cases} |D(k)|^2, & \text{if } \hat{X}(k) = Y(k); \\ |X(k)|^2, & \text{if } \hat{X}(k) = 0. \end{cases}$$

The best choice between the two possible estimates thus depends on whether the noise component is larger than the signal component.

In the denoising problem considered in this work, there is obviously no direct access to $X(k)$ or $D(k)$ —if there was, there would be no challenge in estimating $x[n]$. However, purely for sake of comparison, one can imagine the existence of an oracle with knowledge of $|X(k)|$ and $\sigma^2 = E[|D(k)|^2]$ that tells you whether to use $\hat{X}(k) = Y(k)$ or $\hat{X}(k) = 0$.² The oracle would tell you to use

$$\hat{X}_{\text{oracle}}(k) = \begin{cases} Y(k), & \text{if } |X(k)| \geq \sigma; \\ 0, & \text{otherwise.} \end{cases}$$

²Notice that the oracle knows the signal magnitude and the variance of the noise but does not know the *realization* of the noise $D(k)$.

The expected squared error for this coefficient is thus $E[|X(k) - \hat{X}(k)|^2] = \min\{|X(k)|^2, \sigma^2\}$, and the expected squared error over the whole signal is then

$$E[\|X - \hat{X}\|^2] = \sum_k \min\{|X(k)|^2, \sigma^2\}. \quad (2.1)$$

Incidentally, this equation shows the value of a sparse representation of the signal: if the wavelet domain representation of the signal has only $K \ll N$ nonzero coefficients, the expected squared error after denoising is at most $K\sigma^2$, so the noise has been reduced by a factor of at least $N/K \gg 1$.

The performance of the above oracle-based technique may seem irrelevant because no such oracle is available in practice. However, it is useful as a reference against which to compare the performance of other techniques. Donoho and Johnstone [10] showed that with an appropriate choice of the threshold T , replacing the oracle by the simple and implementable rule

$$\hat{X}_{\text{thresh}}(k) = \begin{cases} Y(k), & \text{if } |Y(k)| \geq T; \\ 0, & \text{otherwise} \end{cases} \quad (2.2)$$

degrades the performance by a factor logarithmic in the signal length:

$$E[\|X - \hat{X}_{\text{thresh}}\|^2] \leq (2 \ln N + 1) \left(\sigma^2 + E[\|X - \hat{X}_{\text{oracle}}\|^2] \right). \quad (2.3)$$

Furthermore, no diagonal estimator can improve the $2 \ln N$ factor, so in asymptotic terms thresholding is as good as any denoising technique that acts independently on each of the wavelet coefficients. While (2.2) is the hard threshold operator, the same bound (2.3) applies for soft thresholding.

Thresholding is a simple way to detect signal energy. The increased error of hard thresholding as compared to the error obtained with an oracle is due to two types of errors: coefficients that should have been set to zero but were retained (“false positives” in the detection of signal energy) and those that should have been retained but were set to zero (“false negatives”). A small threshold will result in more false positives and a large threshold will result in more false negatives. The optimal threshold value balances these types of error.

The threshold value used to prove (2.3) is

$$T = \sigma\sqrt{2 \ln N}. \quad (2.4)$$

The intuition for this value is that with very high probability, the maximum amplitude of N independent samples of $\mathcal{N}(0, \sigma^2)$ noise is just below $\sigma\sqrt{2 \ln N}$. So this threshold value will insure that when the noise-free coefficient $X(k)$ is approximately zero, the thresholded value of the coefficient

is zero with high probability. For large enough values of N —when the maximum amplitude of the noise becomes predictable—threshold with this value of T removes all of the small coefficients that are dominated by noise but is also removes some of the smaller signal components that are not dominated by noise. In other words, it avoids false positives on small coefficients. The optimal threshold value is smaller than $\sigma\sqrt{2\ln N}$ but, in agreement with the earlier-stated results, does not greatly reduce the error below the bound (2.3).

Much of the engineering work that followed the seminal work of Donoho and Johnstone focused on the selection of threshold values for denoising images. Many researchers noted that the universal and asymptotically near-optimal thresholds of [9] could be improved by ad hoc means. Some proposed means for computing a better single threshold to apply to all the wavelet coefficients while others suggested using different thresholds for different subbands.

The best thresholding in a wavelet basis is achieved when the threshold is adapted spatially and with scale. In [4], threshold adaptation is done with context modeling and generalized Gaussian models for the wavelet coefficients. This adaptation is motivated by Rissanen’s minimum description length (MDL) principle in [13].³ The concepts of denoising by compression articulated in [24] and MDL inspired the threshold selections in [3]. Other methods of threshold selection are developed, for example, in [16, 30].

In this work, the threshold $T = 3\sigma$, where σ is the standard deviation of the noise or an estimate of the standard deviation of the noise, is generally used. This value is empirically supported for image denoising [21, p. 462] and has a similar heuristic justification as (2.4): the threshold $\sigma\sqrt{2\ln n}$ is intended to be larger than the noise with vanishing probability, while 3σ is smaller than the noise with small, fixed probability. Allowing a small, fixed probability of false positives better balances the two types of errors and hence leads to lower average error.

2.4 Cycle spinning

One way of improving the basic denoising algorithm is through a method called *cycle spinning*. In cycle spinning, the signal to be denoised is translated by various time shifts. Each time shift is separately transformed and denoised, and the estimate is then taken to be the average of these results.

³While denoising performance is not greatly improved, only a few nonzero coefficients are retained so the method of [13] is good for simultaneous compression and denoising.

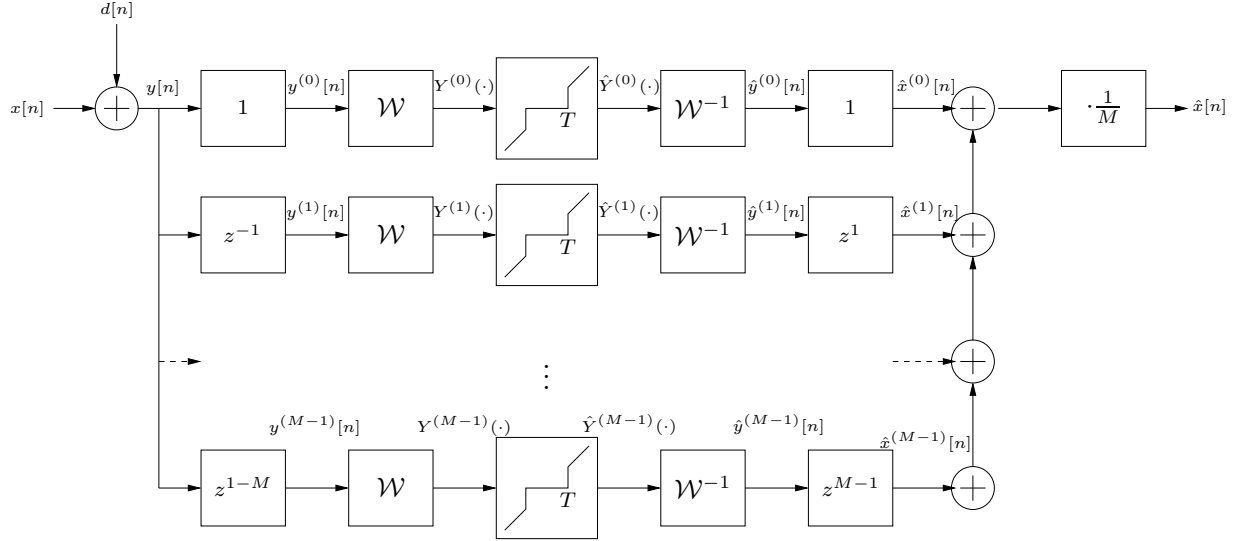


Figure 2.6: A depiction of cycle spinning. The number of shifts is $M = 2^J$, where the wavelet transform \mathcal{W} has J stages. The M wavelet transforms will have some coefficients in common, so this is not an efficient implementation.

For a sequence $x[n]$, let $x^{(i)}[n] = x[n - i]$, i.e. the right shift of $x[n]$ by i . Thus for various shifts i , we take the noisy signal $y[n]$, shift it by i , then perform a wavelet denoising on $y^{(i)}[n]$. Each of these denoised outputs are left-shifted back by i , and these estimates are denoted $\hat{x}^{(i)}[n]$.

In general, the estimates $\hat{x}^{(i)}[n]$ are different for different values of i because the wavelet transform is *periodically translation-invariant*. Because of the decimation in each branch, the wavelet transforms of different shifts differ from one another for shifts up to $N = 2^J$, where J is the number of stages in the filter bank. Thus, the estimates $\hat{x}^i[n]$ will also differ. For a J -stage filter bank, $\hat{x}^{(i)}$ is equal to $\hat{x}^{(k)}$ when $k = i \bmod 2^J$; therefore, the estimates $\hat{x}^{(0)}, \dots, \hat{x}^{(M-1)}$, where $M = 2^J$, are in general different estimates.

The cycle spinning estimate is obtained by simply linearly averaging the 2^J estimates, $\hat{x}^{(i)}[n]$, $i = 0, \dots, 2^J - 1$. The idea is that the error in the estimates are not completely dependent. Consequently averaging these estimates should yield a reduction in noise. For example, if the additive noise were white and if these estimate errors were completely independent, then the noise would be reduced by a factor of $1/M$. In general however, the errors are not independent but they do have independent components, and some reduction in noise is obtained.

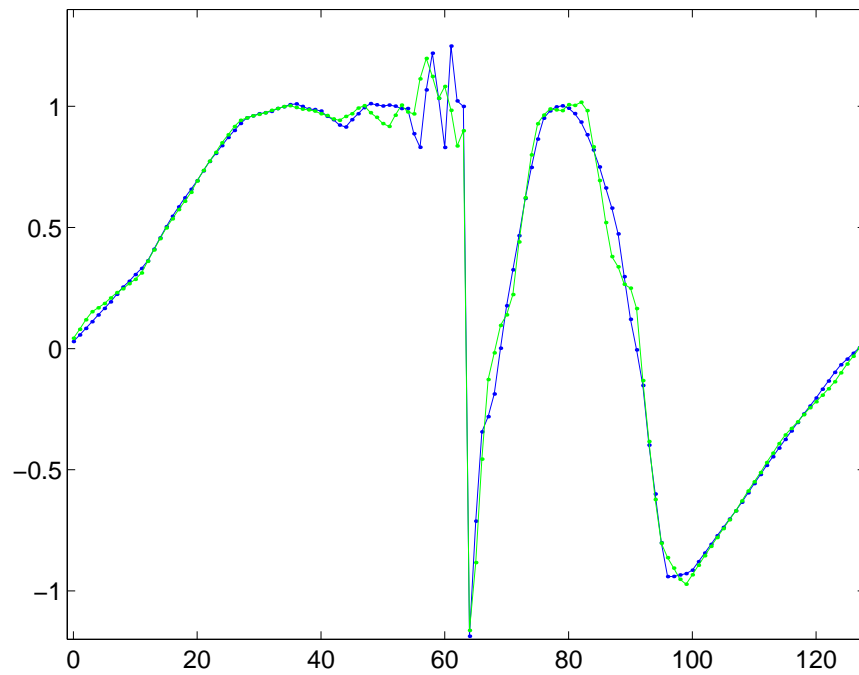


Figure 2.7: Denoising with a shift gives a different estimate than denoising without a shift. However, the errors are not uncorrelated.

Chapter 3

Improving Cycle Spinning through Projections

3.1 Problems with Cycle Spinning: An Illustrative Example

Cycle spinning reduces the estimation error by simply averaging the estimates obtained from denoising various shifts of the signal. It is natural to wonder if this is the best way to combine the information from these 2^J estimates. The basic thesis of this work is that linear averaging is, in fact, far from optimal. In this section, a simple example is presented illustrating how improvement may be possible.

Consider estimating a piecewise constant signal $x[n]$ from a noise corrupted signal $y[n]$,

$$y[n] = x[n] + d[n], \quad n = 0, \dots, N - 1, \quad (3.1)$$

where, as usual, N is the signal length and $d[n]$ represents additive noise.

To denoise a piecewise constant function, it is natural to use a wavelet transform that has one vanishing moment. For higher order polynomial signals, we could of course use a transform with more vanishing moments. To implement the wavelet transform in this example, we use a 1-stage filter bank with filters

$$H_0(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}), \quad H_1(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}).$$

This actually corresponds to the filter coefficients for the Haar, or the first order Daubechies (D_1) wavelet transform. As discussed in Section 2.2.3, the D_1 filter produces an orthogonal wavelet transform with the zero-moment property for constant functions.

Since this is a one-stage filter bank, cycle spinning would entail averaging estimates obtained from shifts of zero and one. Let us first consider the wavelet coefficients of the true signal, $x[n]$, under the two shifts. The detail coefficients are the output of the high-pass filter $H_1(z)$ subsampled by two. Applying this transform to the piecewise constant signal $x[n]$ yields detail coefficients

$$X_1^{(0)}(k) = \begin{cases} 2^{-1/2}(x[2k] - x[2k - 1]) & k = 1, \dots, N/2 - 1, \\ 2^{-1/2}x[0] & k = 0. \end{cases} \quad (3.2)$$

Here, we have assumed, for simplicity, that N is even. Similarly, the transform of the signal shifted by one has detail coefficients:

$$X_1^{(1)}(k) = 2^{-1/2}(x[2k + 1] - x[2k]), \quad k = 0, \dots, N/2 - 1. \quad (3.3)$$

Now suppose the function $x[n]$ happened to be constant on the entire interval $[0, N - 1]$, not just piecewise constant. Then (3.2) and (3.3) show that the detail coefficients of the unknown signal would satisfy:

$$\begin{aligned} X_1^{(0)}(k) &= 0, & \text{for } k = 1, \dots, N/2 - 1, \text{ and} \\ X_1^{(1)}(k) &= 0, & \text{for } k = 0, \dots, N/2 - 1. \end{aligned}$$

That is, all but one of the detail coefficients of the true signal would be zero.

Now consider the detail wavelet coefficients of shifts of the noisy signal $y[n]$. The wavelet transform of $y[n]$ is the sum of the wavelet transforms of the true signal, $x[n]$, and noise, $d[n]$. Therefore, for values of i and k with $X_1^{(i)}(k) = 0$, $Y_1^{(i)}(k)$ would contain only noise energy and no component of the signal. If the noise is sufficiently small, these coefficients will fall below the threshold level and would be set to zero in the thresholding. Therefore, the estimates, $\hat{x}^{(i)}[n]$, $i = 0, 1$, obtained from taking the inverse wavelet transform of the thresholded shifts of the signal would satisfy,

$$\begin{aligned} \hat{x}^{(0)}[2k] &= \hat{x}^{(0)}[2k - 1], & \text{for } k = 1, \dots, N/2 - 1, \\ \hat{x}^{(1)}[2k + 1] &= \hat{x}^{(1)}[2k], & \text{for } k = 0, \dots, N/2 - 1. \end{aligned} \quad (3.4)$$

Thus, the estimates on each shift will be constant over pairs of samples: $\hat{x}^{(0)}[n]$ would be constant on even pairs, and $\hat{x}^{(1)}[n]$ on odd pairs.

So with sufficiently small noise, denoising with a shift of zero revealed that the signal is constant over even pairs of samples. The second denoising, with the shift of one, revealed that the signal is constant over odd pairs. Therefore, simultaneously utilizing both sets of information, the final estimate should be constant on both even *and* odd pairs' samples. Such an estimate, $\hat{x}[n]$, would

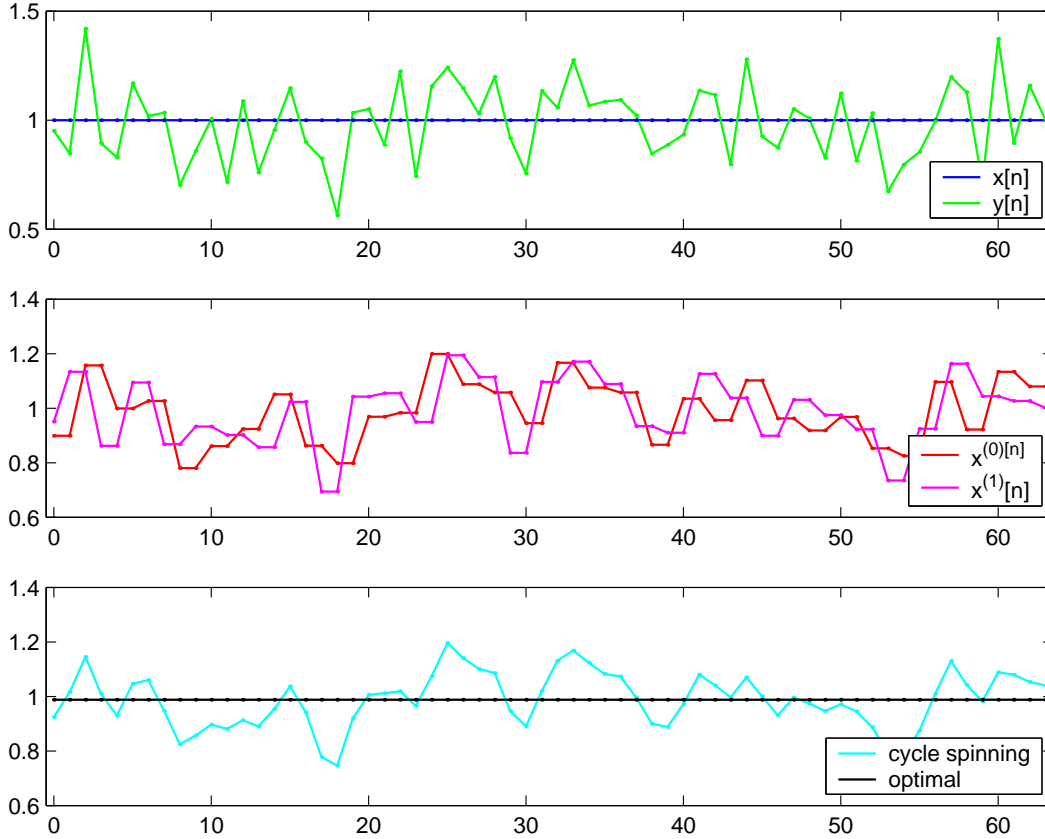


Figure 3.1: Example to illustrate the deficiency of cycle spinning.

satisfy,

$$\hat{x}[n + 1] = \hat{x}[n], \text{ for all } n = 0, \dots, N - 2.$$

That is, the estimate should be constant on the entire interval $[0, N - 1]$, just like the true signal.

But, simply linearly averaging the pairwise constant estimates, $\hat{x}^{(0)}[n]$ and $\hat{x}^{(1)}[n]$, does not result in a constant signal. In fact, the estimate obtained from averaging the two signals, would be constant on *neither* even nor odd samples. Thus, we see that although the information from the two shifts showed that the true signal is constant on the entire interval, cycle spinning does not produce an estimate with this property. This motivates seeking an alternative method for combining the estimates and the potential to better utilize the information from the two shifts.

A simple example is given in Figure 3.1. The top panel shows a constant signal $x[n] = 1$ and a noisy version of $x[n]$ denoted as $y[n]$. The second panel shows the two pairwise constant estimates $\hat{x}^{(0)}[n]$ and $\hat{x}^{(1)}[n]$. These are each less noisy than $y[n]$ but also are far from being constant sequences. The bottom panel shows the average of $\hat{x}^{(0)}[n]$ and $\hat{x}^{(1)}[n]$ as would be computed with

cycle spinning. This is compared to the optimal estimate computed from $y[n]$ given that the signal is constant.

3.2 Wavelet Thresholding as a Projection

To understand the problem with cycle spinning algorithm better and to motivate an alternative algorithm, we need to introduce the idea of wavelet thresholding as a projection.

Projections play a fundamental role in denoising. A classical method to denoise a signal is to find a subspace in which the true signal belongs, and to project the noisy signal onto that subspace. Noise reduction occurs since projecting preserves the component of the true signal, while removing any noise in the null space of the projection. Least squares estimation is a classic example of denoising by projecting.

Wavelet thresholding can also be seen as an projection. Selecting which wavelet coefficients to be set to zero, in effect, identifies a subspace in which most of the true signal energy is likely to belong. Setting those coefficients of the noisy signal to zero projects the noisy signal onto that subspace. Unlike least squares estimation, however, wavelet thresholding can be seen as an *adaptive* projection, since the denoising procedure both identifies the subspace containing the true signal and performs the projection to that subspace.

To describe the wavelet thresholding projection mathematically, recall that a projection is a matrix, P , satisfying the properties that $P = P^*$ and $P^2 = P$ (here, and in the remainder of this work, we will restrict our attention to *orthogonal* projections). A well-known property of projections is that a matrix P is a projection if and only if it can be expressed as the product $P = U\Lambda U'$ where U is an orthogonal matrix and Λ is a diagonal matrix with only ones or zeros on the diagonal. From Section 2.3, the output of wavelet thresholding a noisy signal y can be written,

$$D(y) = \mathcal{W}^{-1}\Lambda_{\alpha(y)}\mathcal{W}(y)$$

where $\mathcal{W}(y)$ is the wavelet transform of y , $\alpha(y)$ is the set of coefficients zeroed out in the thresholding and $\Lambda_{\alpha(y)}$ is the hard thresholding operator. Now, if the wavelet transform filters are orthogonal, \mathcal{W} , will have an orthogonal matrix representation. Also, the hard thresholding operator $\Lambda_{\alpha(y)}$ can be represented as a diagonal matrix with only ones or zeros. Therefore, taking $U = \mathcal{W}$ and $\Lambda = \Lambda_{\alpha(y)}$, we see that $D(y)$ is a projection of y .

The noise reduction from wavelet thresholding can be understood easily in terms of the range

and null space of the projection. Recall that the wavelet coefficients, $\alpha(y)$, that are set to zero are assumed to contain mostly noise. That is, the true signal should have little or no energy in these wavelet coefficients. The range space of the projection $D(y)$ is precisely the set of signals whose wavelet coefficients, $\alpha(y)$, are zero. Therefore, most of the signal energy should be contained in the range space of the projection, while the null space contains mostly noisy. Consequently, projecting the noisy signal y onto the range space will remove little of the energy of the true signal, while removing mostly noise.

3.3 Cycle Spinning as Projection Averaging

Now consider denoising a signal y using cycle spinning with a J -stage wavelet transform. As described in Section 2.4, cycle spinning will wavelet threshold the signal y with shifts of $i = 0, \dots, M - 1$ where $M = 2^J$. The signal y denoised with a shift i can be written as

$$\hat{x}^{(i)} = D_i(y) = S^{(-i)}\mathcal{W}^{-1}\Lambda_{\alpha(y,i)}\mathcal{W}S^{(i)}(y)$$

where $S^{(i)}$ is the left shift by i , and $\alpha(y, i)$ is the set of wavelet coefficients zeroed out in the denoising by a shift of i . The left shift operator $S^{(i)}$ is orthogonal with inverse $S^{(-i)}$. Therefore, as in the previous subsection, $D_i(y)$ is a projection of y .

For each shift i , we will call the range space of the projection $D_i(y)$ the *i -th denoise space of y* . The i -th denoise space of y is the set of signals x with $X^{(i)}(k) = 0$ for all $k \in \alpha(y, i)$. Wavelet thresholding the signal y with a shift i , projects y onto its i -th denoise space.

Assuming the thresholding works properly, for all shifts i , the i -th denoise space of y should contain most of the energy of the true signal. Each estimate $\hat{x}^{(i)}$ is the projection onto the i -th denoise space. Cycle spinning thus locates M subspaces containing most of the energy of the true signal, and finds M corresponding estimates in the subspaces.

The final cycle spinning estimate is the average of the denoised signals, $\hat{x}^{(i)}$,

$$\hat{x} = \frac{1}{M} \sum_{i=0}^{M-1} \hat{x}^{(i)}.$$

Now, although each estimate $\hat{x}^{(i)}$ lies in the i -th denoise space of y , the average of the estimates, \hat{x} , does not, in general, lie in any of the subspaces.

This fact is, in essence, the problem with cycle spinning: wavelet thresholding at different shifts locates a number of subspaces in which the true signal is likely to belong. However, the final cycle

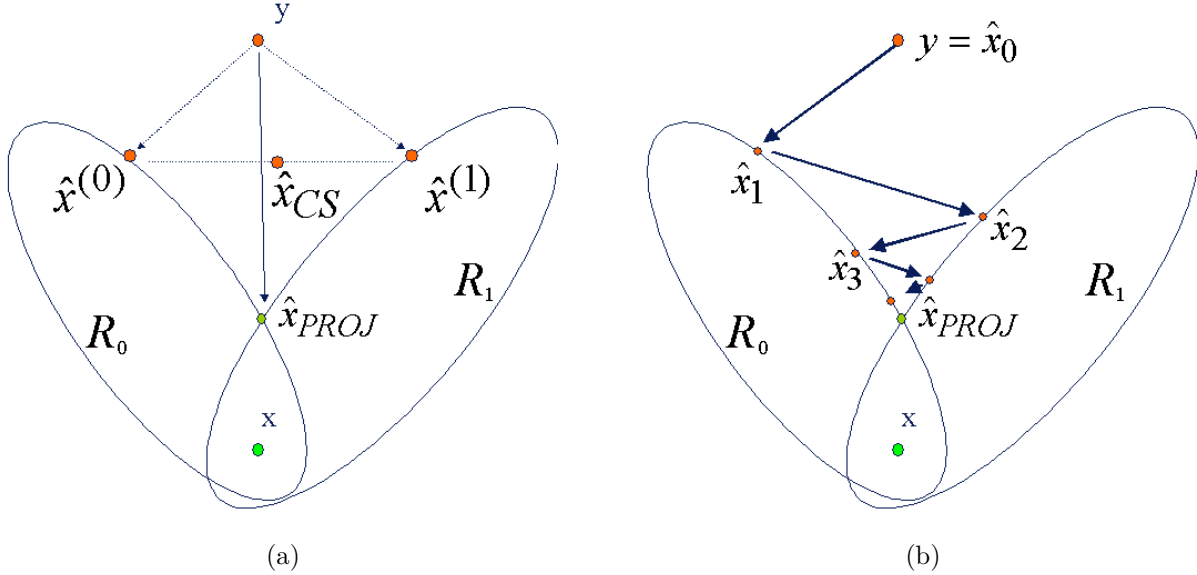


Figure 3.2: Comparison of denoising methods by considering projections onto subspaces. Denoising by hard thresholding in the original wavelet basis is a projection to R_0 ; including a shift makes it a projection to R_1 . (a) Basic wavelet thresholding results in estimate $\hat{x}^{(0)}$, which is the projection of noisy observation y onto R_0 . In cycle spinning the projection onto R_1 , denoted $\hat{x}^{(1)}$, is also computed. The estimate from cycle spinning is the average of $\hat{x}^{(0)}$ and $\hat{x}^{(1)}$ and does not belong to either R_0 or R_1 . (b) To fully utilize the postulated sparseness of the original signal x in either of the wavelet bases (the original basis or the basis used when there is a shift) the estimate should be in the intersection of R_0 and R_1 . The point in the intersection nearest to y is the projection of y onto $R_0 \cap R_1$. It can be found by alternating projections onto R_0 and R_1 .

spinning estimate does not belong to any of them. It is natural to think that cycle spinning can be improved by finding some estimate in all of the denoise subspaces. The relationships of these subspaces is illustrated in Figure 3.2(a).

3.4 Projecting to the Denoise Space Intersection

Call the intersection of the M denoise spaces of y , the *denoise space intersection*. If the thresholding of the noisy signal y works properly, the true signal should have most of its energy in each of M denoise spaces. Therefore the true signal should have most of its energy in the denoise space intersection as well.

This fact motivates a conceptually simple wavelet denoising procedure: use wavelet thresholding

at different shifts to identify the M denoise subspaces of the noisy signal y , and then project y onto the intersection of the subspaces, namely the denoise space intersection.

To quantitatively compare the performance of projecting the denoise space intersection to averaging consider the following. As usual, let x be an unknown signal, and let y be a noise-corrupted version of x given by,

$$y[n] = x[n] + d[n], \quad (3.5)$$

where $d[n]$ is additive noise.

Let R_i , $i = 0, \dots, M - 1$ be a set of M subspaces of signals and let R be the intersection of the subspaces R_i . Suppose that x is known *a priori* to belong to each subspace R_i . Consequently $x \in R$. Let P_i be the projection operator onto R_i and P be the projection onto R . Consider two estimates of x from the noisy signal y ,

$$\hat{x}_{avg} = \frac{1}{M} \sum_{i=0}^{M-1} P_i y, \quad (3.6)$$

$$\hat{x}_{proj} = P y. \quad (3.7)$$

The first estimate, \hat{x}_{avg} , is the average of the projections of the noisy signal y onto the subspaces R_i . If R_i were the i -th denoise space of y , then \hat{x}_{avg} would be the estimate obtained from the cycle spinning algorithm. The space R is the intersection of the subspaces R_i , and is therefore the denoise space intersection. Consequently, \hat{x}_{proj} , is the proposed estimate obtained by projecting the noisy signal y onto R . These estimates are illustrated in Figure 3.2(b).

To compare the estimation error in the two estimates, we begin with the following simple lemma.

Lemma 1 *Consider the estimates \hat{x}_{avg} and \hat{x}_{proj} above and suppose that the true signal $x \in R$. Then for all noise sequences d ,*

$$\|\hat{x}_{proj} - x\| \leq \|\hat{x}_{avg} - x\|.$$

Proof: Since $x \in R$ and P is the projection onto R , $Px = x$. Also, since each subspace R_i contains R , $PP_i = P$. Therefore,

$$\begin{aligned} P(\hat{x}_{avg} - x) &= P \left[\frac{1}{M} \sum_{i=0}^{M-1} (P_i y - x) \right] = \frac{1}{M} \sum_{i=0}^{M-1} PP_i y - Px \\ &= \frac{1}{M} \sum_{i=0}^{M-1} (Py - x) = Py - x = \hat{x}_{proj} - x. \end{aligned}$$

Since P is a projection, $\|P\| \leq 1$ and therefore,

$$\|\hat{x}_{proj} - x\| = \|P(\hat{x}_{avg} - x)\| \leq \|\hat{x}_{avg} - x\|.$$

□

Lemma 1 shows that for all noise sequences, d , the error in the projection estimate \hat{x}_{proj} is never higher than the averaging estimate \hat{x}_{avg} . Consequently, projecting onto the intersection of the subspaces is only better than averaging over the projections.

Under certain circumstances the projection estimate can be significantly better than the averaged estimate. One important case, is illustrated in the following lemma which quantifies the difference in the mean square errors when $d[n]$ is white.

Lemma 2 *Consider the estimates \hat{x}_{avg} and \hat{x}_{proj} above and suppose that the true signal $x \in R$. Suppose, in addition, that the noise $d[n]$ is white, zero-mean, with variance $\mathbf{E}|d[n]|^2 = \sigma^2$. Then, the mean square error (MSE) of the estimates satisfy,*

$$\begin{aligned} \mathbf{E}\|x - \hat{x}_{proj}\|^2 &= \sigma^2 \dim R \\ \mathbf{E}\|x - \hat{x}_{avg}\|^2 &\geq \frac{\sigma^2}{M^2} \sum_{i=0}^{M-1} \dim R_i. \end{aligned}$$

In the second equation, equality is obtained when the errors $x - P_i y$ are uncorrelated for different values of i .

The proof of the lemma is given at the end of the section.

The lemma shows that, in the case of white noise, the averaging estimate, \hat{x}_{avg} , has an error that is greater than or equal to σ^2/M times the average of the dimensions of the range spaces of the projections. On the other hand, the projection estimate, \hat{x}_{proj} , results in a error of σ^2 times the dimension of the intersection of the range spaces. Now, in general, the dimension of the intersection of the subspaces is typically much smaller than the dimensions of the subspaces themselves. Therefore, in the case of additive white noise, the projection estimate should result in a dramatically smaller error.

As an example, consider again the example in Section 3.1. In this case, the cycle spinning estimate is the average of $M = 2$ estimates: $\hat{x}^{(0)}$ and $\hat{x}^{(1)}$. The estimate $\hat{x}^{(0)}$ can be seen as the projection of the noisy signal y onto a space R_0 defined by the set of signals satisfying the first equation of (3.4). The estimate $\hat{x}^{(1)}$ is the projection onto a space R_1 defined by the set of signals

satisfying the second equation of (3.4). Since R_0 and R_1 are subspaces of an N -dimensional space defined by $N/2 - 1$ and $N/2$ constraints, respectively, they have dimensions $N/2 + 1$ and $N/2$. Therefore, Lemma 2 shows that the cycle spinning estimate error is lower bounded by,

$$\mathbf{E}\|\hat{x}_{avg} - x\|^2 \geq \frac{\sigma^2}{2^2}(N/2 + N/2 + 1) = \frac{\sigma^2(N + 1)}{4}.$$

Now, the intersection of the spaces $R = R_0 \cap R_1$ is the set of constant functions on $[0, N - 1]$. This space has dimension one. Therefore, from Lemma 2 an estimate obtained from the projection of y to the denoise space intersection, R , would have an MSE,

$$\mathbf{E}\|\hat{x}_{proj} - x\|^2 = \sigma^2.$$

Comparing the two estimations errors, we see that for large N , the projection estimate error can be dramatically smaller. For example, if $N = 100$ (ie. the signal is constant for 100 samples), the projection estimate error would be at least 25 times, or 14 dB, lower than obtained from cycle spinning.

Proof of Lemma 2: Since $x \in R$, $Px = x$, so

$$\hat{x}_{proj} - x = Py - Px = Pd.$$

Also, since d is white, zero mean and has variance $\mathbf{E}|d[n]|^2 = \sigma^2$, the expectation of the matrix dd' is given

$$\mathbf{E}(dd') = \sigma^2 I,$$

where I is the identity matrix. Therefore,

$$\begin{aligned} \mathbf{E}\|\hat{x}_{proj} - x\|^2 &= \mathbf{E}\|Pd\|^2 = \mathbf{E}(d'P^2d) = \mathbf{E}(d'Pd) \\ &= \mathbf{E}\mathbf{Tr}(Pdd') = \mathbf{Tr}[P\mathbf{E}(dd')] = \sigma^2 \mathbf{Tr}P = \sigma^2 \dim R. \end{aligned}$$

Here, we used the properties of the projection P that $P^2 = P$ and the trace of the projection, $\mathbf{Tr}P$, is equal to $\dim R$, the dimension of its range space. This proves the first part of the lemma.

For the second equation, observe that since $x \in R$, $x \in R_i$ for all i and therefore $P_i x = x$. Therefore

$$\hat{x}_{avg} - x = \frac{1}{M} \sum_{i=0}^{M-1} (P_i y - P_i x) = \frac{1}{M} \sum_{i=0}^{M-1} P_i d.$$

Consequently,

$$\mathbf{E}\|\hat{x}_{avg} - x\|^2 = \mathbf{E}(\hat{x}_{avg} - x)'(\hat{x}_{avg} - x) = \frac{1}{M^2} \sum_{i,j=0}^{M-1} \mathbf{E}(d' P_i P_j d). \quad (3.8)$$

To lower bound this sum, first observe that for all i and j ,

$$\mathbf{E}d'P_iP_jd = \sigma^2\mathbf{Tr}(P_iP_j) = \sigma^2\mathbf{Tr}(P_iP_j^2) = \sigma^2\mathbf{Tr}(P_jP_iP_j) \geq 0,$$

where in the last step, we have used the fact that, since the matrices P_i and $P_j \geq 0$, the matrix $P_jP_iP_j \geq 0$. Therefore, the sum can be lower bounded by removing all the terms with $i \neq j$,

$$\mathbf{E}\|\hat{x}_{avg} - x\|^2 \geq \frac{1}{M^2} \sum_{i=0}^{M-1} \mathbf{E}(d'P_id).$$

Now, similar to the previous computation,

$$\mathbf{E}d'P_id = \sigma^2 \dim R_i,$$

so,

$$\mathbf{E}\|\hat{x}_{avg} - x\|^2 \geq \frac{\sigma^2}{M^2} \sum_{i=0}^{M-1} \dim R_i.$$

This proves the second equation of the lemma.

Finally, suppose that the estimation errors $P_iy - x$ are uncorrelated for different values of i . Since $P_iy - x = P_id$,

$$0 = \mathbf{E}((P_iy - x)'(P_jy - x)) = \mathbf{E}(d'P_jP_id).$$

for all $i \neq j$. When $\mathbf{E}(d'P_jP_id) = 0$ for all $i \neq j$, the sum in (3.8) reduces to

$$\mathbf{E}\|\hat{x}_{avg} - x\|^2 = \frac{1}{M^2} \sum_{i=0}^{M-1} \mathbf{E}(d'P_id) = \frac{\sigma^2}{M^2} \sum_{i=0}^{M-1} \dim R_i,$$

showing that equality is obtained in the second equation of the lemma when the errors are uncorrelated. \square

3.5 Characterization of the Denoise Space Intersection

3.5.1 Dimension Bounds

The previous section claimed, without proof, that, in general, the denoise space intersection has a much smaller dimension than the individual denoise space. This property, along with Lemma 2, was used to argue that projecting to the denoise space intersection should result in a much lower MSE than obtained from cycle spinning. The dimension of the denoise space intersection and denoise subspaces were worked out for the simple constant signal case of Section 3.1. For this example,

it was shown that the denoise space intersection does indeed have a much smaller dimension than the individual denoise spaces. In this section, we extend this argument to more general wavelet transforms.

The denoise spaces are, in general, complex functions of the wavelet transforms and thresholding. Throughout the section, we make two simplifying assumptions. First, we will restrict our analysis to $J = 1$ stage wavelet transforms, where the computations are much simpler.

Assumption 1 *The wavelet transform \mathcal{W} is an orthogonal one-stage filter bank with filters $H_0(z)$ and $H_1(z)$. The high-pass filter, $H_1(z)$, has an impulse response $h_1[n]$, $n \geq 0$ with $h_1[0] \neq 0$.*

With a one-stage wavelet transform, denoising is performed with two shifts of the signal: shifts of $i = 0$ and 1 . As usual, the detail coefficients of the wavelet transform of a signal $x[n]$ shifted by i will be denoted $X_1^{(i)}(k)$. We will assume signals have N points, $x[n]$, $n = 0, \dots, N - 1$. Consequently, for each i , the detail coefficients will have $N/2$ points, $X_1^{(i)}(k)$, $k = 0, \dots, N/2 - 1$.

As a second assumption, we assume that the thresholding zeros out all but some initial number of detail coefficients. To analyze this situation, we fix a $K > 0$, and, for $i = 0, 1$, define R_i to be the set of signals

$$R_i = \left\{ \hat{x} : \hat{X}^{(i)}(k) = 0 \text{ for all } k \text{ with } 2k + i \geq K \right\}. \quad (3.9)$$

We let $R = R_0 \cap R_1$.

Later, we will assume the denoise spaces have the form R_i . Of course, in general, wavelet thresholding may not zero out all but some initial set of coefficients. However, the analysis in this section can be applied to an interval of the signal all of whose detail coefficients are set to zero.

The first lemma in this section computes the dimensions of the spaces R_i and R .

Lemma 3 *Consider the spaces R_0 , R_1 and R above.*

a) *The dimensions of R_0 and R_1 satisfy*

$$\dim R_0 + \dim R_1 = N + K.$$

b) *The space $R = R_0 \cap R_1$ is precisely the set of signals $\hat{x}[n]$ satisfying*

$$(h_1 \star \hat{x})[k] = 0 \text{ for all } k \geq K.$$

Also, $\dim R = K$.

Proof: Given a signal, $\hat{x}[n]$, the output of the filter $H_1(z)$ can be represented as a matrix product, $T\hat{x}$, where T is the $N \times N$ Toeplitz matrix,

$$T = \begin{bmatrix} h_1[0] & 0 & \cdots & 0 \\ h_1[1] & h_1[0] & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_1[N-1] & h_1[N-2] & \cdots & h_1[0] \end{bmatrix}.$$

Since the wavelet coefficients $\hat{X}^{(i)}(k)$ are obtained from the left shifting \hat{x} by i samples, filtering the shifted output with $H_1(z)$, and then downsampling by two, it can be verified that

$$\hat{X}^{(i)}(k) = (T\hat{x})[2k+i]. \quad (3.10)$$

for $k = 0, \dots, N/2 - 1$ and $i = 0, 1$.

Now, let K_i be the number of values of k with $k = 0, \dots, N/2 - 1$ and $2k+i < K$. Observe that $K_0 + K_1 = K$. Combining (3.9) with (3.10), we see that the space R_i is defined by the set of signals \hat{x} with $(T\hat{x})[j] = 0$ for $N/2 - K_i$ values of j . Since $h_1[0]$ is non-zero, the matrix T is invertible. Therefore,

$$\dim R_i = N - (N/2 - K_i) = N/2 + K_i.$$

Consequently,

$$\dim R_0 + \dim R_1 = N/2 + K_0 + N/2 + K_1 = N + K.$$

This proves part a).

For part b), observe that if $\hat{x} \in R$, then $\hat{x} \in R_0$ and R_1 and therefore,

$$\hat{X}^{(i)}(k) = 0 \text{ for } i = 0, 1 \text{ and } k \text{ with } 2k+i \geq K.$$

Therefore, (3.10) shows that $\hat{x} \in R$ is equivalent to

$$(h_1 \star \hat{x})[k] = (T\hat{x})[k] = 0 \text{ for all } k \geq K.$$

Consequently, R is a set of signals \hat{x} with $(T\hat{x})[k] = 0$ for $N - K$ values of k . Consequently, the dimension of R is given by

$$\dim R = N - (N - K) = K.$$

□

3.5.2 MSE: Averaging versus Projecting

Lemma 3 can be easily combined with Lemma 2 to compare the MSE of the cycle spinning estimate with that obtained by projecting to the denoise space intersection. Let $x[n]$ be the unknown signal, and $y[n]$ be a noisy version given by (3.5). Suppose $d[n]$ is zero-mean, white noise with $\mathbf{E}|d[n]|^2 = \sigma^2$.

Also suppose that wavelet thresholding the noisy signal $y[n]$ zeros out all but some initial set of detail coefficients. Specifically, suppose that, for some $K > 0$, the i -th denoise space of y is of the form R_i in the previous subsection. The cycle spinning estimate is the average of the projections of y onto R_i . Therefore, the cycle spinning estimate is given by \hat{x}_{avg} in (3.6).

Now suppose that the true signal x is in R , the intersection of R_0 and R_1 . This assumption is equivalent to assuming that the wavelet thresholding at each shift does not zero out any wavelet coefficients in which the true signal has energy. Under this assumption, Lemmas 2 and 3, along with the fact that $M = 2^J = 2$ show that the MSE of the cycle spinning estimate is bounded below by

$$\mathbf{E}\|x - \hat{x}_{avg}\|^2 \geq \frac{\sigma^2(N + K)}{4}. \quad (3.11)$$

Now consider the MSE of projecting to the denoise space intersection. Since R_0 and R_1 are the denoise spaces of y , $R = R_0 \cap R_1$ is the denoise space intersection. Therefore, the estimate \hat{x}_{proj} in (3.7) is the projection of the noisy signal to the denoise space intersection. Lemmas 2 and 3 show that the MSE of this estimate is

$$\mathbf{E}\|x - \hat{x}_{proj}\|^2 = \sigma^2 K. \quad (3.12)$$

To compare the MSE's in (3.11) and (3.12), note that K is the total number of detail coefficients not set to zero in the thresholding. Now, typically, thresholding should zero out detail coefficients unless they depend on samples at a discontinuity or other high pass feature. Usually, wavelet thresholding is applied only when these features of the signal are relatively sparse. Consequently, K is typically much smaller than the signal length N . Therefore, the MSE in (3.12) obtained from projecting to the denoise space intersection should be much smaller than the MSE from cycle spinning in (3.11).

3.5.3 Polynomial Approximations and Zero Moment Filters

Maintaining the notation of the previous subsection, suppose that the true signal $x[n]$ is an unknown polynomial of degree p . As discussed earlier, for denoising piecewise polynomial signals, it is natural to use a wavelet transform with the zero moment property. Suppose the high-pass filter in the

wavelet transform, $H_1(z)$, has the zero moment property for polynomials of degree p . Also suppose that $H_1(z)$ is implemented with a K -th order FIR filter with coefficients $h_1[n]$.

Since $x[n]$ is a polynomial of degree p and $H_1(z)$ has the zero moment property for degree p polynomials, we have

$$(h_1 \star x)[k] = 0 \text{ for all } k \geq K. \quad (3.13)$$

The initial K coefficients of the convolution may be non-zero due to the transient response of the filter.

Now, for a shift of $i = 0$ or 1 , the detail coefficients of x are given by shifting and decimating the filter output $h_1 \star x$,

$$X_1^{(i)}(k) = (h_1 \star x)[2k + i]. \quad (3.14)$$

Combining (3.13) and (3.14), we see that the detail coefficients of the true signal x satisfy (3.9). Therefore, the above analysis applies and the MSE's of the cycle spinning estimate and the estimate obtained by projecting onto the denoise space intersection are given by (3.11) and (3.12) respectively. These equations show that as long as the filter order K is much smaller than the length N on which the signal is a polynomial, the MSE of the projection estimate will be much better than that of cycle spinning.

It is instructive to compare the denoise space intersection with the space of polynomials. Suppose for a zero moment filter, $H_1(z)$, we use a D_{p+1} Daubechies filter. As discussed earlier, the D_{p+1} filter has the zero moment property for polynomials of degree p . The filter has order $K = 2p + 1$.

According to Lemma 3b), the denoise space intersection has dimension $K = 2p + 1$. However, the space of polynomials of degree p only has dimension $p + 1$. Consequently, the denoise space intersection is a larger space, having approximately twice the dimension for large p . Therefore, projecting to the denoise space intersection will not exactly produce a polynomial estimate. Instead it will project the noisy signal to a space containing, but larger than, the set of polynomials.

This property is a shortcoming of projecting to the denoise space intersection in comparison to direct least squares estimation of the polynomial coefficients. Least squares estimation would, in effect, project the noisy signal onto the space of polynomials. Since the space of polynomials has approximately half the dimension as the denoise space intersection, least squares would result in an approximately 3dB lower MSE than the denoise space intersection projection, assuming the additive noise were white.

There are however other advantages of projection estimate to least squares. Most significantly,

least squares does not identify the boundaries of the piecewise polynomial estimate, which is difficult, especially at higher dimensions. We will compare least squares and the proposed projection estimate further later.

Chapter 4

Recursive Cycle Spinning

4.1 Basic Algorithm

The previous chapter motivated the following simple conceptual algorithm: given a noisy signal y ,

1. Use wavelet thresholding of shifted versions of the noisy signal, y , to identify M subspaces likely to contain the most of the true signal energy. These spaces were called the *denoise spaces* of y .
2. Compute the *denoise space intersection*, the intersection of the M denoise spaces.
3. Project y onto the denoise space intersection to obtain an estimate for the true signal.

The previous chapter argued that, if the thresholding correctly identified M subspaces containing the true signal, the estimate obtained from projecting to the denoise space intersection would have a much lower MSE than that obtained from averaging the projections to the M denoise spaces.

Unfortunately, directly performing the above algorithm is computationally difficult. For example, the “brute force” method would be to write each of the M projections as matrices, numerically compute the intersection of their range spaces (say, with a singular value decomposition), and then project to the intersection. Unfortunately, all the matrices in the computation would have a size of $N \times N$, where N is the signal length. For large signal lengths, the matrix computations would be prohibitive.

Recursive cycle spinning is an approximate, but computationally simple, method for projecting to the denoise space intersection. The algorithm is simple to describe.

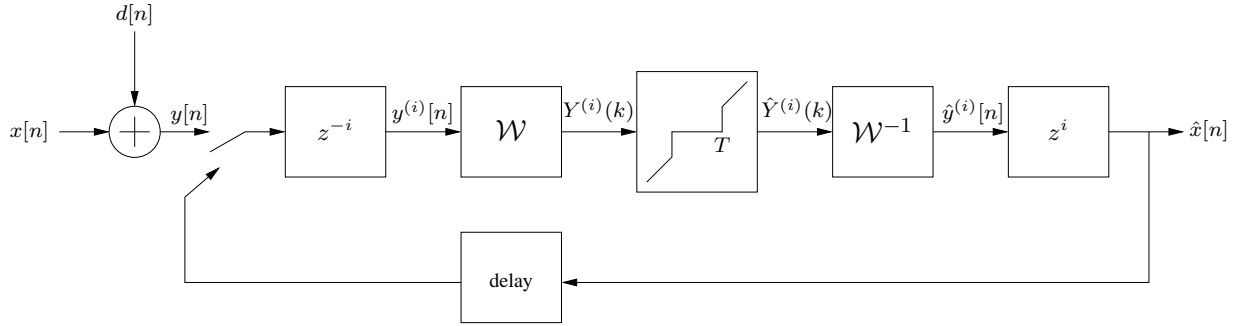


Figure 4.1: A depiction of recursive cycle spinning. The switch is initially in the upper position to draw the noising signal $y[n]$ into the system. Thereafter, the switch is in the lower position to recursively denoise the output of the previous iteration. The shift i is changing from one iteration to the next.

As before, let $D_i(x)$ denote the signal x wavelet denoised using a shift of i . That is, $D_i(x)$ is obtained by shifting x to the left by i samples, wavelet denoising the shifted signal, and then shifting the denoised signal back. Recursive cycle spinning generates a sequences of estimates, \hat{x}_ℓ , as follows. The initial estimate, \hat{x}_0 , is set equal to the original noisy signal: $\hat{x}_0 = y$. The subsequent estimates are generated from the recursive rule:

$$\hat{x}_{\ell+1} = D_i(\hat{x}_\ell), \quad i = \ell \bmod M.$$

That is, the $\ell + 1$ -st estimate is obtained by denoising the ℓ -th estimate with a shift of i , cycling through the shifts $i = 0, \dots, M - 1$ in a round-robin manner.

The idea of the algorithm is that each iteration projects the estimate to one of the M denoise spaces. Ideally, the algorithm would be performed for an infinite number of iterations, and the estimate sequence \hat{x}_ℓ would converge to some limit point, say \hat{x}_∞ . This limit point should be a fixed point of the recursion updates, so that

$$\hat{x}_\infty = D_i(\hat{x}_\infty), \quad \forall i = 0, \dots, M - 1.$$

This fixed point condition means that \hat{x}_∞ would be in the range space of projections D_i for all i . That is, \hat{x}_∞ would be in the denoise space intersection. As argued in the previous chapter, the denoise space intersection should be a space of small dimension containing the true signal, and consequently, the final limit point should be close to the true signal.

Of course, the convergence of the algorithm is far from obvious. It is possible, for example, that the estimate sequence could “bounce” between the M subspaces without converging. Moreover,

even if the estimates do converge, the algorithm must be terminated after some finite number of iterations. Consequently, it is important that the recursive cycle spinning converges quickly.

We will address these convergence issues in detail in Chapter 5. There, we will show that, under suitable conditions, the recursive algorithm does indeed converge to the projection of the original noisy signal to the denoise space intersection. Also, in a numerical example presented later, we will see that the algorithm can converge quickly.

However, before discussing the convergence, we will need to address some implementation issues that are particular to recursive cycle spinning.

4.2 Windowed Thresholding

In recursive cycle spinning, it is critical that not too many wavelet coefficients are set to zero in the thresholding step. Otherwise, the denoise space intersection, the space the algorithm attempts to project the noisy signal to, may be too small, and the estimate could be poor.

To illustrate this potential problem, consider using recursive cycle spinning with a $J = 1$ stage wavelet transform. For a one-stage transform, recursive cycle spinning is implemented cycling through shifts of $i = 0$ and 1.

Now, suppose that the unknown, true signal, $x[n]$, has some discontinuity, say at $n = n_0$. The discontinuity will result in non-zero detail coefficients, the number depending on the filter length. Using an analysis similar to Lemma 3, it can be shown that the total number of non-zero detail coefficients from the shifts of zero and one will equal K , where K is the filter length of the high pass filter $H_1(z)$.

However, even for a large discontinuity, some of these coefficients may be small and could be set to zero in the thresholding. In this case, recursive cycle spinning will try to project the noisy signal onto a space with less than a total of K non-zero wavelet coefficients. As a result, the range space of the projection, the denoise space intersection, would have less than K degrees of freedom from the discontinuity. Since the filter has length K , this lack of degrees of freedom will necessitate some linear dependence between the portion of the signal before and after the discontinuity.

This restriction could result in a poor estimate. For example, in denoising piecewise polynomials with filters with vanishing moments, the denoise space intersection may not contain all polynomials before and after the discontinuity.

To avoid this problem, recursive cycle spinning can be implemented with “windowed” thresh-

olding. As discussed earlier, in basic denoising, the set of wavelet coefficients that are set to zero for a noisy signal y left shifted by i , is of the form,

$$\alpha(y, i) = \{ k : |Y^{(i)}(k)| \leq T(k) \}, \quad (4.1)$$

where $Y^{(i)}(k)$ is the k -th wavelet coefficient of y shifted by i and $T(k)$ is the threshold level for the coefficient. For recursive cycle spinning, we propose to zero out the set of wavelet coefficients given by

$$\alpha(y, i) = \{ k : |Y^{(i)}(j)| \leq T(j) \text{ for all } j \in [k, k + \Delta(k)] \}, \quad (4.2)$$

where $\Delta(k)$ is a windowing parameter. The idea in the windowing is that a wavelet coefficient k will not be zeroed out if any coefficient within $\Delta(k)$ of the coefficients exceeds the threshold. The window length $\Delta(k)$ can be selected separately for each subband, and should be set equal to the filter response length for that branch. With this setting, the windowing will guarantee that a sufficient number of wavelet coefficients are left as non-zero.

4.3 Soft vs. Hard Thresholding

Up to now, we have described wavelet thresholding with hard thresholding. However, in wavelet denoising, it is common practice to use a soft thresholding function, described by

$$(\Lambda_{SOFT}Y)(k) = \begin{cases} 0 & \text{if } |Y(k)| < T(k), \\ Y(k) - T(k) & \text{if } Y(k) \geq T(k), \\ Y(k) + T(k) & \text{if } Y(k) \leq -T(k), \end{cases} \quad (4.3)$$

where $Y(k)$ are the wavelet coefficients of the noisy signal and $T(k)$ are the threshold levels. Unlike hard thresholding, soft thresholding is a continuous function and decreases the value of signal by the threshold level. It has been argued that under certain statistical assumptions, soft thresholding can result in slightly greater noise reduction.

In basic wavelet thresholding, one has the choice of using either soft or hard thresholding depending on the problem. However, recursive cycle spinning can be implemented only with hard thresholding.

The reason for this restriction is simple. In soft thresholding, all wavelet transform coefficients are reduced somewhat, even ones that are not set to zero. If soft thresholding were used in recursive cycle spinning, the estimate would decrease in norm in each iteration, eventually converging to

zero. Therefore, to obtain a meaningful result, soft thresholding cannot be used with recursive cycle spinning.

Chapter 5

Convergence Analysis

5.1 Recursive Projections

Before analyzing the convergence of the proposed recursive cycle spinning algorithm, a general result about recursive projections is proven. The following theorem shows, in essence, that if a set of projections are repeatedly applied to a vector, then the result will converge to a vector in the intersection of the range spaces of the projections. This result is the most technically complicated, but will be the basis of the subsequent analysis.

Theorem 1 *Let $\mathcal{P} = \{P_0, \dots, P_{K-1}\}$ be a finite set of orthogonal projections. Suppose \hat{x}_ℓ is a sequence of vectors satisfying*

$$\hat{x}_{\ell+1} = Q_\ell \hat{x}_\ell, \tag{5.1}$$

where $Q_\ell \in \mathcal{P}$ for all ℓ . Suppose that each P_i occurs infinitely often. That is, for every i , $Q_\ell = P_i$ for infinitely many values of ℓ . Then

$$\hat{x}_\ell \rightarrow P \hat{x}_0 \tag{5.2}$$

where P is the projection onto the intersection of the range spaces of the P_i 's.

Proof: The proof of this theorem is long and occupies the remainder of this section. We will begin with some notation. For any subset $I \subseteq \{0, \dots, K-1\}$, let \mathcal{S}_I denote the set of matrices of the form

$$Q = Q_1 \cdots Q_L$$

where

a) For every $\ell = 1, \dots, L$, $Q_\ell = P_i$ for some $i \in I$.

b) For every $i \in I$, $P_i = Q_\ell$ for some ℓ .

That is, matrices $Q \in \mathcal{S}_I$ are products of matrices P_i with $i \in I$, with each P_i occurring at least once in the product. For example, if $I = \{1, 3, 4\}$, \mathcal{S}_I contains matrices such as $P_1 P_4 P_3$, $P_1 P_4 P_3 P_4 P_1$, etc. Let R_i denote the range space of P_i and let

$$R_I = \bigcap_{i \in I} R_i.$$

The first step in the proof of Theorem 1 is to prove a somewhat technical bound. Define a sequence γ_n recursively as follows. Let γ_0 be any number with $0 < \gamma_0 < 1$, say $\gamma_0 = 1/2$. Given γ_n , define

$$\gamma_{n+1} = \max \|P_j x\| \tag{5.3}$$

where the maximum is over all x of the form

$$x = x_1 + x_2 \in R_J^\perp \cap R_I \oplus R_I^\perp, \tag{5.4}$$

where I is some subset with $|I| \leq n$, $J = I \cup j$ for some $j \notin I$, and

$$\|x_1\|^2 + \gamma_n^{-1} \|x_2\|^2 \leq 1. \tag{5.5}$$

We now prove the following.

Lemma 4 *For all n , $0 < \gamma_n < 1$.*

Proof: The lemma is proven by induction on n . The case for $n = 0$ follows from the definition of γ_0 . So, suppose that $\gamma_n \in (0, 1)$ for some n . We want to show that $\gamma_{n+1} \in (0, 1)$. Since the set of x satisfying (5.5) is compact, and there are only finitely many subsets I and finitely many values $j \notin I$, the maximum in (5.3) is achieved. Therefore, the lemma is proven if the following can be shown: for any subset I with $|I| \leq n$, $j \notin J$,

$$\|P_j x\| < 1$$

for any x satisfying (5.5).

First suppose $x_2 \neq 0$ where x_2 is the component of x in R_I^\perp in (5.4). Therefore, $\|x_2\| > 0$, and since $\gamma_n < 1$, $\|x_2\|^2 < \gamma_n^{-1} \|x_2\|^2$. Also, since P_j is a projection, $\|P_j\| \leq 1$. Using these two facts

along with equation (5.5), we have

$$\begin{aligned}\|P_j x\|^2 &\leq \|x\|^2 = \|x_1\|^2 + \|x_2\|^2 \\ &< \|x_1\|^2 + \gamma_n^{-1} \|x_2\|^2 \leq 1.\end{aligned}$$

Now suppose that $x_2 = 0$, but $\|P_j x\| \geq 1$. By (5.5), $\|x\| = \|x_1\| \leq 1$. Therefore,

$$\begin{aligned}1 &\leq \|P_j x\|^2 = \|P_j x_1\|^2 \\ &= \|x_1\|^2 - \|(I - P_j)x_1\|^2 \leq 1 - \|(I - P_j)x_1\|^2.\end{aligned}$$

Hence $(I - P_j)x_1 = 0$, or equivalently, $P_j x_1 = x_1$. Hence, $x_1 \in R_j$. But $x_1 \in R_j^\perp \cap R_I$ and therefore,

$$x_1 \in R_j^\perp \cap R_I \cap R_j = R_j^\perp \cap R_j.$$

Thus, $x_1 = 0$. This is impossible. □

The next lemma is, again, a technical bound, in terms of the variables γ_n .

Lemma 5 *Let $I \subseteq \{0, \dots, K - 1\}$, and $Q \in \mathcal{S}_I$. Then,*

- a) *If $x \in R_I$, then $Qx = x$.*
- b) *If $x \in R_I^\perp$, then $Qx \in R_I^\perp$.*
- c) *If $x \in R_I^\perp$, then $\|Qx\| \leq \gamma_n \|x\|$.*

Proof: The definition of R_I implies that if $x \in R_I$, $x \in R_i$ for all $i \in I$. Therefore, $P_i x = x$ for all $i \in I$. Since $Q \in \mathcal{S}_I$ is a product of matrices P_i with $i \in I$, $Qx = x$. This proves part a).

To prove part b), write Q as an operator on $R_I \oplus R_I^\perp$,

$$Q = \begin{bmatrix} I & Q_{12} \\ 0 & Q_{22} \end{bmatrix}. \tag{5.6}$$

Here the first column is $[I \ 0]'$, since $QR_I = R_I$. Now since Q is a product of projections, $\|Q\| \leq 1$. Therefore, the norm of the first row in (5.6) must be less than one. This is only possible with $Q_{12} = 0$. Consequently, $QR_I^\perp \subseteq R_I^\perp$ and this proves part b).

Part c) will be proven by induction on n . First consider the case with $n = 1$. If $|I| = 1$, then I has only one element. That is, $I = \{i\}$ for some i . Consequently, if $Q \in \mathcal{S}_I$, $Q = P_i$. Now if $x \in R_I^\perp = R_i^\perp$, $Qx = P_i x = 0$. Hence, $\|Qx\| = 0$. In particular, $\|Qx\| \leq \gamma_1 \|x\|$.

Now suppose that part c) is true for some n . To prove it is true for $n + 1$, consider a $Q \in \mathcal{S}_J$ with $|J| = n + 1$ and $x \in R_J^\perp$. WLOG we can assume $\|x\| = 1$. We must show that $\|Qx\| \leq \gamma_{n+1}$.

Since Q is a product of projections P_j , $j \in J$, with each projection occurring at least once, we can write Q as

$$Q = AP_jQ_0 \tag{5.7}$$

where $Q_0 \in \mathcal{S}_I$, $|I| = n$, $j \notin I$, $J = \{j\} \cup I$ and A some product of projections, P_j , $j \in J$. Write $x = x_1 + x_2$ with $x_1 \in R_J^\perp \cap R_I$ and $x_2 \in R_I^\perp$. Let $y = Q_0x$ and write $y = y_1 + y_2$ where $y_1 = Q_0x_1$ and $y_2 = Q_0x_2$. Part a) shows that $y_1 = x_1$ and therefore, $y_1 \in R_J^\perp \cap R_I$. Part b) shows that $y_2 \in R_I^\perp$, and therefore

$$y = y_1 + y_2 \in R_J^\perp \cap R_I \oplus R_I^\perp.$$

Also, by the induction hypothesis, $\|y_2\| = \|Q_0x_2\| \leq \gamma_n\|x_2\|$. Since $\|x\| = 1$,

$$\begin{aligned} \|y_1\|^2 + \gamma_n^{-1}\|y_2\|^2 &\leq \|x_1\|^2 + \|x_2\|^2 \\ &= \|x\|^2 = 1. \end{aligned}$$

By the definition of γ_{n+1} , we have that $\|P_jy\| \leq \gamma_{n+1}$.

Now, in the decomposition (5.7), the matrix A is a product of projections, so $\|A\| \leq 1$. Using this fact along with the fact that $y = Q_0x$ and $\|P_jy\| \leq \gamma_{n+1}$ we have that

$$\|Qx\| = \|AP_jQ_0x\| = \|AP_jy\| \leq \|P_jy\| \leq \gamma_{n+1}.$$

This completes the proof of the Lemma. □

We introduce some more notation. For $I = \{0, \dots, K - 1\}$, let R and \mathcal{S} denote R_I and \mathcal{S}_I respectively. The space R is the intersection of all the spaces R_i , and therefore P is the projection onto R .

Define $e_\ell = \hat{x}_\ell - P\hat{x}_0$. We need to show that $e_\ell \rightarrow 0$. Since P is the projection onto R , and $R_i \subseteq R$ for all i , $P_iP = P$ for all i . Now for every ℓ , $Q_\ell = P_i$ for some i . Therefore, $Q_\ell P = P$ for all ℓ . Consequently,

$$e_{\ell+1} = \hat{x}_{\ell+1} - P\hat{x}_0 = Q_\ell\hat{x}_\ell - Q_\ell P\hat{x}_0 = Q_\ell e_\ell.$$

Therefore, e_ℓ satisfies the recursion

$$e_{\ell+1} = Q_\ell e_\ell. \tag{5.8}$$

Since each Q_ℓ is a projection, $\|Q_\ell\| \leq 1$. Therefore, $\|e_\ell\|$ is non-increasing. Therefore, to show that $e_\ell \rightarrow 0$, it suffices to show that $e_\ell \rightarrow 0$ on some subsequence of values of ℓ .

We define this subsequence as follows. Let $T_0 = 0$. Given T_k , let T_{k+1} be chosen such that the product of matrices

$$\prod_{\ell=T_k}^{T_{k+1}-1} Q_\ell \in \mathcal{S}. \quad (5.9)$$

Now, for every i , $Q_\ell = P_i$ for infinitely many values of ℓ . Therefore, for any T_k , the product in (5.9) will contain the projections P_i for all i for sufficiently large T_{k+1} . Therefore, T_{k+1} can be chosen such that the product is in \mathcal{S} , and consequently, the sequence T_k is well-defined. Let A_k be the product,

$$A_k = \prod_{\ell=T_k}^{T_{k+1}-1} Q_\ell \in \mathcal{S},$$

and let $v_k = e_{T_k}$. From (5.8) it follows that

$$v_{k+1} = A_k v_k. \quad (5.10)$$

To show that a subsequence of e_k converges to zero, we will show that $v_k \rightarrow 0$.

To prove this, first note that since $T_0 = 0$,

$$v_0 = e_0 = \hat{x}_0 - P\hat{x}_0.$$

Since P is the projection onto R , $v_0 \in R^\perp$. Now, since $A_k \in \mathcal{S}$, Lemma 5b) along with (5.10) shows that if $v_k \in R^\perp$, $v_{k+1} \in R^\perp$. Since $v_0 \in R^\perp$, it follows from induction on k that $v_k \in R^\perp$ for all k .

Since $v_k \in R^\perp$, Lemma 5c) and (5.10) show that

$$\|v_{k+1}\| \leq \gamma_K \|v_k\|.$$

From Lemma 4, $0 < \gamma_K < 1$, and therefore, $v_k \rightarrow 0$. This shows that a subsequence of e_k converges to zero and completes the proof of the theorem. \square

5.2 Global Convergence of Recursive Cycle Spinning

We can now address the convergence of the recursive cycle spinning algorithm. We will use the notation of Chapter 4 where $D_i(x)$ denotes the output of wavelet thresholding a signal x with a shift of i . The recursive cycle spinning algorithm applied to a noisy signal y is given by $\hat{x}_0 = y$ and

$$\hat{x}_{\ell+1} = D_i(\hat{x}_\ell) \quad i = \ell \bmod M \quad (5.11)$$

where $M = 2^J$ and J is the number of stages in the wavelet transform.

The convergence of the algorithm depends on the thresholding decision rules. At least two thresholding methods have been discussed: the basic thresholding described in equation (4.1) and the windowed thresholding in (4.2). Within these methods, there is considerable flexibility in the choice of the threshold levels $T(k)$.

To analyze these methods in one common framework, we will assume that the set of coefficients to be zeroed out when thresholding a signal x with a shift of i are given by

$$\alpha(x, i) = \{ k : \phi(x, i, k) < 0 \}, \quad (5.12)$$

where $\phi(x, i, k)$ are a set of functions, continuous in y . This definition of $\alpha(x, i)$ incorporates a number of thresholding schemes. For example, the basic thresholding rule in (4.1) can be described by

$$\phi(x, i, k) = |X^{(i)}(k)| - T(k),$$

where $T(k)$ is the threshold level for the k -th wavelet coefficient. The definition (5.12) also applies to the case when the threshold levels are continuous functions of the input signal x , and to the case of windowed thresholding.

The following is the main result of this section.

Theorem 2 *Let \hat{x}_ℓ be a sequence of signals produced by the recursive cycle spinning algorithm (5.11). Suppose that the denoising operations are performed with an orthogonal wavelet transform, and assume that the thresholding sets are given by (5.12) for continuous functions $\phi(x, i, k)$. Then \hat{x}_ℓ converges to a signal \hat{x}_∞ with the property that*

$$\hat{x}_\infty = D_i(\hat{x}_\infty)$$

for all shifts i .

The theorem shows that the recursive cycle spinning algorithm is guaranteed to converge to some signal \hat{x}_∞ . Since $\hat{x}_\infty = D_i(\hat{x}_\infty)$ for all i , \hat{x}_∞ is in its i -th denoise space for all i . Therefore, the recursive cycle spinning algorithm is guaranteed to converge to a denoise space intersection as desired.

We call Theorem 2 a global convergence result, since the convergence does not require any restrictions on the initial condition of the algorithm \hat{x}_0 .

It is important to recognize that Theorem 2 does not relate the convergence point of the algorithm to any "true" signal. It doesn't, for example, state that the algorithm converges to the projection onto a desired denoise space intersection. Instead, it only states that the algorithm converges to some denoise space intersection. To relate the convergence point to a true signal, we will need further conditions, which will be discussed in the next section.

Proof of Theorem 2: Using the notation of Sect. 2.4, wavelet thresholding operator with a shift i can be described by,

$$D_i(x) = S^{(-i)}\mathcal{W}^{-1}\Lambda_\alpha(x, i)\mathcal{W}S^{(i)}(x).$$

where $S^{(i)}$ is the shift operator, $\Lambda_\alpha(x, i)$ is the hard thresholding operator and \mathcal{W} is the wavelet transform. Define the operator,

$$P_{i,\alpha}(x) = S^{(-i)}\mathcal{W}^{-1}\Lambda_\alpha\mathcal{W}S^{(i)}(x).$$

The difference between the operators $D_i(x)$ and $P_{i,\alpha}(x)$ is that $P_{i,\alpha}$ performs the wavelet thresholding with a fixed set of coefficients α . As discussed in Sect. 2.4, the operator $P_{i,\alpha}$ is an orthogonal projection.

Recursive cycle spinning can be re-written as:

$$\hat{x}_{\ell+1} = Q_\ell \hat{x}_\ell$$

where $Q_\ell = P_{i,\alpha_\ell}$ and $\alpha_\ell = \alpha(\hat{x}_\ell, i)$ where $i = \ell \bmod M$. Since there are only a finite number of subsets α and finitely many shifts i , there are a finite number of projections $P_{i,\alpha}$. By Theorem 1, $\hat{x}_\ell \rightarrow \hat{x}_\infty$ for some signal \hat{x}_∞ . The signal \hat{x}_∞ is in the range of all projections $P_{i,\alpha}$ if $Q_\ell = P_{i,\alpha}$ for infinitely many values of ℓ .

To show that $\hat{x}_\infty = D_i(\hat{x}_\infty)$, we must show that the wavelet transform of \hat{x}_∞ satisfies

$$\hat{X}_\infty^{(i)}(k) = 0 \text{ for all } k \in \alpha(\hat{x}_\infty, i).$$

Using (5.12), we must show that, for every k and i , either $\hat{X}_\infty^{(i)}(k) = 0$ or $\phi(\hat{x}_\infty, i, k) \geq 0$. Fix i and k and suppose $\phi(\hat{x}_\infty, i, k) < 0$. We must show that $\hat{X}_\infty^{(i)}(k) = 0$.

Consider the sequence of sets α_ℓ with $\ell \bmod M = i$. There must exist a set α such that $\alpha_\ell = \alpha$ for infinitely many values of ℓ with $\ell \bmod M = i$. Since $\phi(x, i, k)$ is continuous, $\phi(\hat{x}_\infty, i, k) < 0$ and $\hat{x}_\ell \rightarrow \hat{x}_\infty$, $\phi(\hat{x}_\ell, i, k) < 0$ for ℓ sufficiently large. Hence, $k \in \alpha_\ell$ for all ℓ sufficiently large with $\ell \bmod M = i$. Consequently, $k \in \alpha$.

Now, for any ℓ with $\ell \bmod M = i$, $Q_\ell = P_{i,\alpha_\ell}$. Since $\alpha_\ell = \alpha$ infinitely often, $Q_\ell = P_{i,\alpha}$ infinitely often. Therefore, \hat{x}_∞ must be in the range space of $P_{i,\alpha}$. This implies that $\hat{X}_\infty^{(i)}(j) = 0$ for all $j \in \alpha$. But $k \in \alpha$, and therefore, $\hat{X}_\infty^{(i)}(k) = 0$. \square

5.3 Local Convergence of Iterative Cycle Spinning

We will now address the convergence of the recursive cycle spinning algorithm to an unknown signal starting with a noisy estimate. Let x denote the true signal and y the noise-corrupted version,

$$y[n] = x[n] + d[n], \quad (5.13)$$

where $d[n]$ is additive noise. We consider the recursive cycle spinning algorithm estimates (5.11) initialized with $\hat{x}_0 = y$. As before, we assume that the wavelet transform in the denoising is orthogonal and the thresholding sets are given by (5.12).

We make two key assumptions. The first assumption is given as follows.

Assumption 2 *For all shifts i and wavelet coefficients k , either*

- a) $X^{(i)}(k) = 0$ and $\phi(x, i, k) < 0$, or
- b) $\phi(x, i, k) > 0$.

A consequence of the assumption is that $X^{(i)}(k) = 0$ whenever $k \in \alpha(x, i, k)$. Therefore,

$$x = D_i(x)$$

for all shifts i and x is contained in its own denoise space intersection. In other words, we assume that the wavelet thresholding of the true signal under any shift would not remove any of the signal energy. The second assumption is that the noise d is small. Under these assumptions, we have the following result.

Theorem 3 *Consider the sequence of signals, \hat{x}_ℓ , generated by the recursive cycle spinning algorithm (5.11) with the initial condition $\hat{x}_0 = y$. Suppose that the wavelet transform used in the denoising is orthogonal, and the thresholding sets are given by (5.12) for some continuous functions $\phi(x, i, k)$. Suppose, in addition, that the noisy signal y is given by (5.13) for a signal x satisfying Assumption 2. Then, there exists an $\epsilon > 0$ such that $\|d\| < \epsilon$ implies \hat{x}_ℓ converges to the projection of y onto the denoise space intersection of the true signal x_0 .*

The theorem states that, if the noise is sufficiently small, then the recursive cycle spinning converges to the projection of the noisy signal onto the denoise space intersection containing the true signal. The result is called a local result since it requires that the initial estimate is sufficiently close to the true signal.

It is important to recognize that Theorem 3 does not state the recursive cycle spinning estimates converge to the true signal x itself. Rather, they converge simply to the projection of the noisy signal onto the denoise space intersection containing x . We have argued earlier, that, in general, the denoise space intersection is a relatively small space containing the signal and therefore, the limiting point of the recursive cycle spinning algorithm should be a good estimate. However, the exact quality of the estimate depends on the wavelet transform and how well the thresholding is at removing coefficients.

Proof of Theorem 3: Using the notation from the previous section, define the projection operator,

$$P_i(\hat{x}) = P_{i,\alpha(x_0,i)}(\hat{x}).$$

A signal \hat{x} is in the range space of P_i if and only if $\hat{X}^{(i)}(k) = 0$ for all $k \in \alpha(x, i)$. That is, \hat{x} is in the range space of P_i if and only if $\hat{X}^{(i)}(k) = 0$ when $X^{(i)}(k) = 0$.

Choose an $\epsilon > 0$ such that $\|\hat{x} - x_0\| < \epsilon$ implies that $\alpha(\hat{x}, i) = \alpha(x_0, i)$ for all i .

Suppose that $\|d\| < \epsilon$. We claim by induction that for all ℓ , $\|\hat{x}_\ell - x_0\| < \epsilon$. For $\ell = 0$,

$$\|\hat{x}_0 - x_0\| = \|y - x_0\| = \|d\| < \epsilon.$$

Now suppose that $\|\hat{x}_\ell - x_0\| < \epsilon$. By the definition of ϵ , $\alpha(\hat{x}_\ell, i) = \alpha(x_0, i)$ for all i . Thus,

$$\hat{x}_{\ell+1} = D_i(\hat{x}_\ell) = P_{i,\alpha(\hat{x}_\ell,i)}(\hat{x}_\ell) = P_i(\hat{x}_\ell),$$

where $i = \ell \bmod M$. Theorem 1 now shows that $\hat{x}_\ell \rightarrow \hat{x}_\infty$ where \hat{x}_∞ is the projection of $\hat{x}_0 = y$ to the intersection of the range spaces of the operators P_i , $i = 0, \dots, M - 1$. But, the range of P_i is precisely the i -th denoise space of the true signal x , and therefore, the intersection of the spaces is the denoise space intersection. Therefore, the sequence converges to the projection of y to the denoise space intersection of the true signal. \square

Chapter 6

Numerical Example

6.1 Simulation Description

The recursive cycle spinning algorithm is illustrated with a simple example. The unknown signal, $x[n]$, to be estimated is a piecewise polynomial with $N = 1024$ points, given by

$$x[n] = \begin{cases} n + 0.08 & 1 \leq n \leq 512, \\ 0.27n^2 + 0.08n + 3 & 513 \leq n \leq 768, \\ 0.01n^4 - 0.07n^3 - 0.01n^2 - 0.03n & 769 \leq n \leq 1024. \end{cases}$$

This piecewise polynomial signal has a maximum degree of four. The signal is estimated from a noise-corrupted signal, $y[n]$, given by

$$y[n] = x[n] + d[n],$$

where $d[n]$ is additive white Gaussian noise with zero mean. The variance of $d[n]$ is set so that the original SNR is 20 dB, where the SNR is defined as,

$$\text{SNR} = 10 \log_{10} \left(\frac{\|x\|^2}{\mathbf{E}(\|d\|^2)} \right).$$

That is, the SNR is the signal energy over the average noise energy expressed in dB. The true signal, $x[n]$, and a typical noise-corrupted version, $y[n]$, are shown in the top panel of Figure 6.1.

To implement the recursive cycle spinning algorithm, we use a Daubechies D_4 wavelet filter bank with $J = 3$ stages. As discussed earlier, wavelet transforms with D_4 filters are orthogonal with 4 vanishing moments, which are adequate for polynomials up to degree three.

As discussed in Chapter 4, recursive cycle spinning should be implemented with windowed thresholding to avoid projecting to a space that is too small. In the example, the window length,

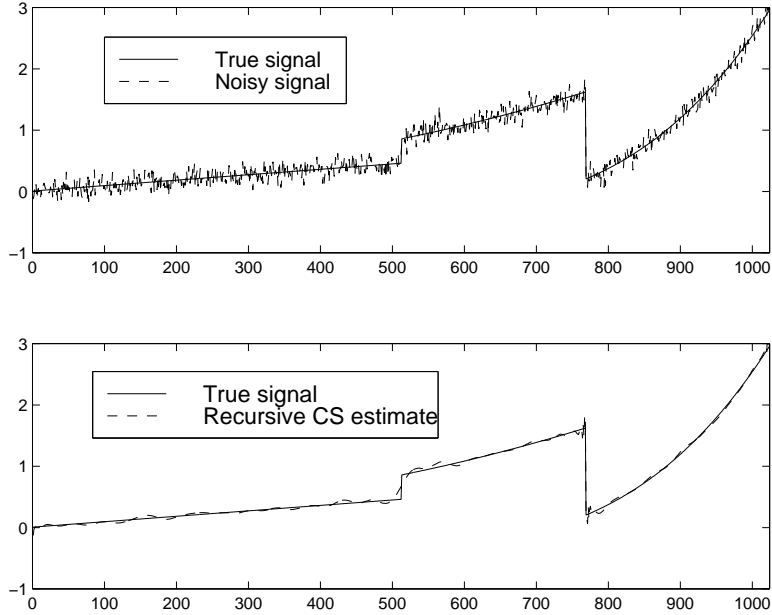


Figure 6.1: Typical result of recursive cycle spinning estimation. Top panel: True piecewise polynomial signal and the original noise-corrupted signal from which the estimate is made. Original SNR = 20 dB. Bottom panel: Final estimate from recursive cycle spinning.

defined by $\Delta(k)$ in (4.2), was set to be constant on each subband at a value equal to the filter impulse response in the band.

Hard-thresholding was used and the threshold levels, $T(k)$, were set on a per subband basis using the simple heuristic,

$$T(k) = 3\sigma,$$

where σ is the root mean square value of the wavelet coefficients in the subband. The idea in this rule is that, assuming the signal contributes little energy to the detail coefficients, σ will approximately be the noise power per wavelet coefficient in each subband. Assuming the noise is Gaussian, setting $T(k) = 3\sigma$ will insure that any coefficient that has only noise will be set to zero with a probability of 0.99.

The recursive cycle spinning algorithm was performed for 400 iterations – 50 iterations of denoising at each shift ranging from 0 to 7. The estimate after the 400 iterations is shown in the bottom panel of Figure 6.1. The final estimate can be seen to be a close estimate of the true signal and has an MSE of -29.3 dB.

Figure 6.1 shows the MSE as a function of the iteration number. As expected, the mean square

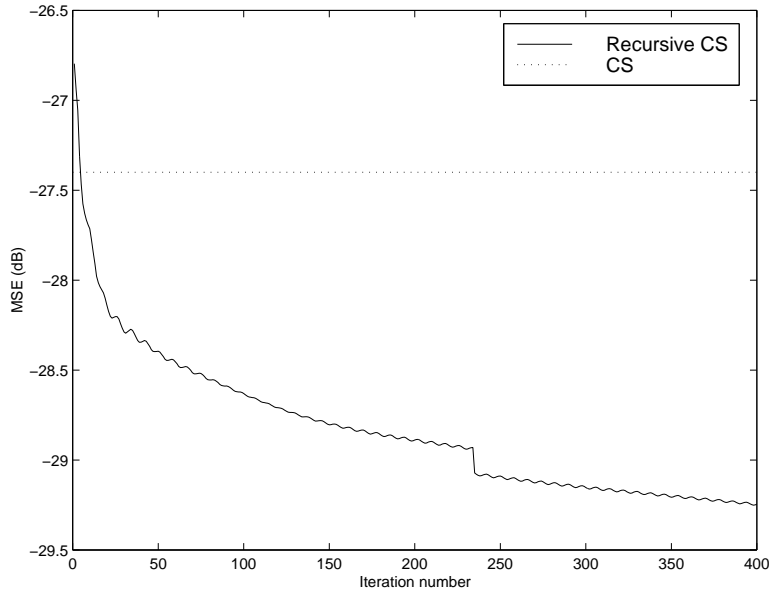


Figure 6.2: Progress of recursive cycle spinning.

error (MSE) decreases monotonically. The MSE drops fastest at the beginning and the estimate is within 1 dB of the final error after 100 iterations.

Also shown in Figure 6.1 for comparison is the MSE from the standard cycle spinning. Since the wavelet transform has $J = 3$ stages, the standard cycle spinning estimate is obtained by averaging the estimates from wavelet denoising at shifts from 0 to $2^J - 1 = 7$. In order that the comparison be equal, the standard cycle spinning was implemented with the same wavelet transform and thresholding rule. In this example, the standard cycle spinning algorithm obtains an MSE of -27.4 dB. The recursive cycle spinning estimate thus outperformed the standard cycle spinning algorithm after only a few iterations, and the final estimate is about 1.9 dB better.

6.2 Comparison of Methods

Figure 6.3 shows the results of the same experiment using a D_3 , or Daubechies third order filter, with various number of stages, J . For each stage, the recursive wavelet thresholding was performed 20 times. For each run, the algorithm was terminated after 100 iterations. The figure shows the mean and standard deviation of the noise reduction, where the noise reduction is defined as the difference between the final and original SNR. For comparison, the results of single-basis wavelet thresholding (thresholding once with no shifts) and standard cycle spinning are also shown.

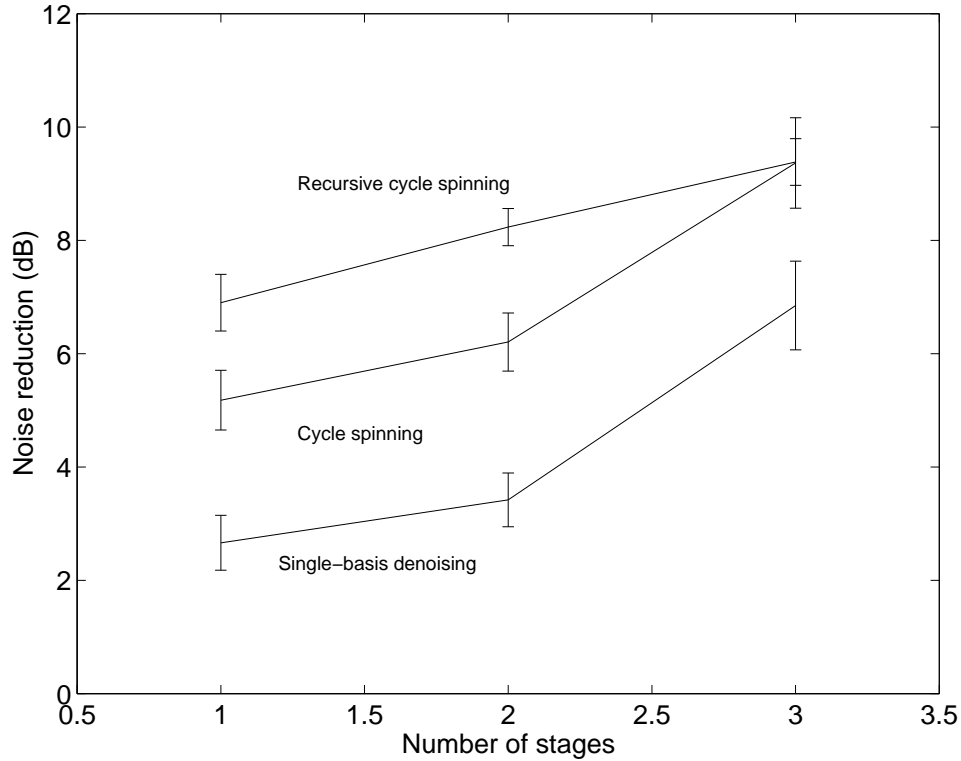


Figure 6.3: Comparison of (single-basis) wavelet thresholding, cycle spinning, and recursive cycle spinning for denoising a piecewise quartic signal. The D_3 filter pair is used and means and standard deviations of 20 experiments are shown.

It can be seen that the recursive cycle spinning method consistently outperforms both cycle spinning and single basis denoising. The improvement is most pronounced with $J = 1$ and $J = 2$ stages. For a $J = 1$ stage transform, recursive cycle spinning shows about a 2 dB improvement over standard cycle spinning, and for $J = 2$ stages, the improvement is about 2.5 dB. Standard cycle spinning itself performs approximately 3 dB better than single-basis denoising, due to the benefits obtained from averaging.

Increasing the number of stages from 1 to 3, results in better noise reduction in all three methods. This improvement is due to the fact that increasing the number of stages increases the low-pass filtering in the scaling coefficients, thereby filtering out more high pass noise.

However, increasing the number of stages beyond 3 results in a degradation in the noise reduction for cycle spinning and single-basis denoising. The degradation is the result of edge effects of the filter: for high filter stages, the transient response of the filter at the edges and discontinuities

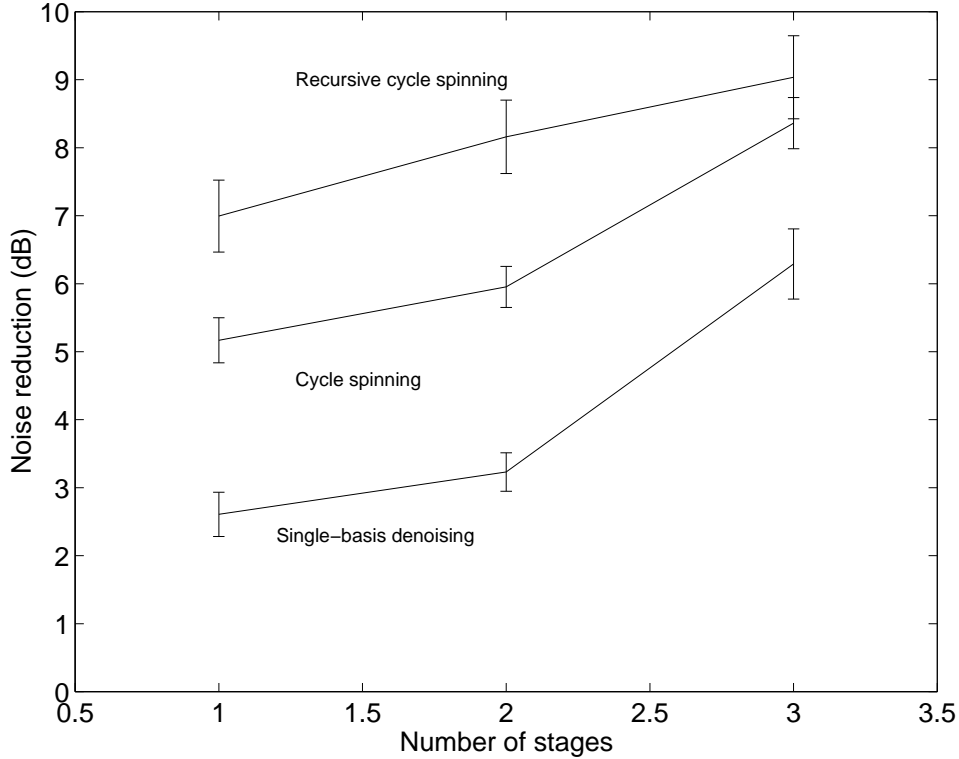


Figure 6.4: Comparison of (single-basis) wavelet thresholding, cycle spinning, and recursive cycle spinning for denoising a piecewise quartic signal. The D_4 filter pair is used and means and standard deviations of 20 experiments are shown.

occupy a significant number of the coefficients in the subband and less coefficients can be set to zero. While these filter edge effects also present a problem for recursive cycle spinning, the effect is outweighed by the fact that using a higher number of stages, results in larger number of subbands with smaller numbers of coefficients. The recursive cycle spinning algorithm tends to converges faster on smaller subbands. Consequently, the estimate error with recursive cycle spinning after a final number of iterations may be lower with a higher number of stages, despite the edge effects.

Figure 6.4 shows the results of repeating the experiment with a Daubechies fourth-order, D_4 , filter pair. The qualitative results are similar to the D_3 results, with recursive cycle spinning outperforming standard cycle spinning and single-basis denoising. The performance improvement is greatest at a lower number of stages. Figure 6.5 shows the results obtained with the D_5 filter pair.

For all three methods the D_4 filter pair performs better than the D_3 filter pair, since the

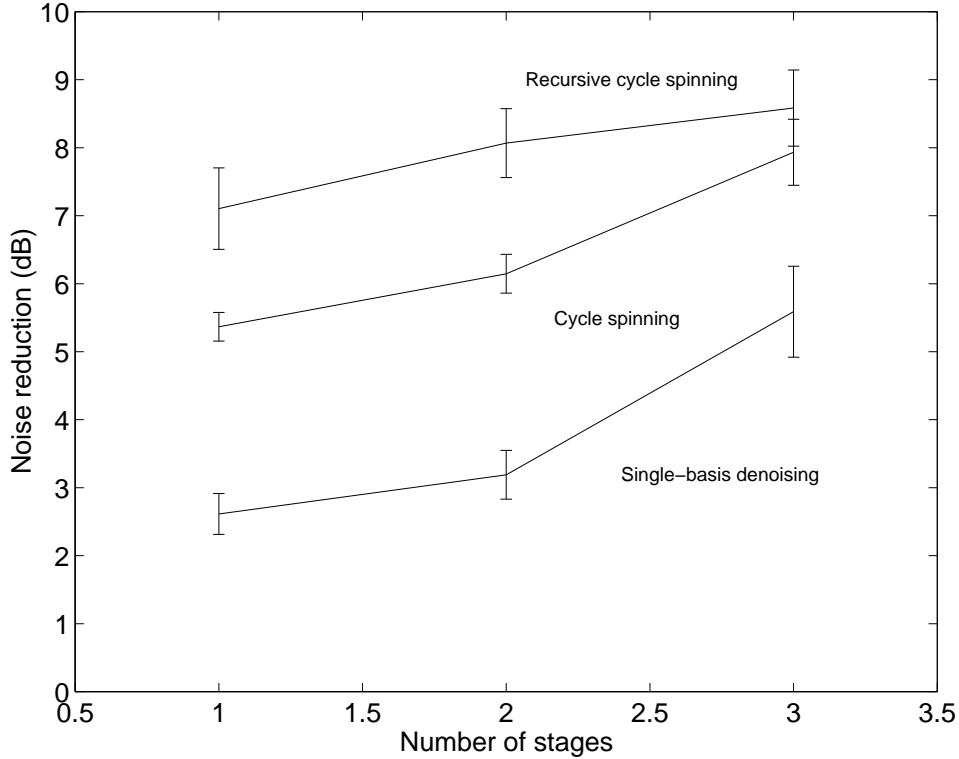


Figure 6.5: Comparison of (single-basis) wavelet thresholding, cycle spinning, and recursive cycle spinning for denoising a piecewise quartic signal. The D_5 filter pair is used and means and standard deviations of 20 experiments are shown.

D_4 has an extra vanishing moment which is desirable for denoising a fourth-order polynomial. Unfortunately, even though it has what would seem to be the optimal number of vanishing moments, the performance with the D_5 filter pair is not as good because of the increased edge effects from having longer filters.

6.3 Comparison to Fourier Methods

The recursive cycle spinning method was also compared against standard low-pass filtering using a DFT or a DCT. The low-pass filtering is implemented by simply zeroing out all but r of the low-pass terms in either the DCT or the DFT.

In general, the ideal number of terms to retain depends on the signal bandwidth and noise. In our comparison against an ideal Fourier low-pass filter, the number r is selected to maximize the *a posteriori* noise reduction. Of course, this cannot be implemented in practice because the true

Method	Noise reduction
RCS w/ D_3 , 1 stage	6.3 dB
RCS w/ D_3 , 2 stages	8.3 dB
RCS w/ D_3 , 3 stages	9.0 dB
RCS w/ D_4 , 1 stage	6.8 dB
RCS w/ D_4 , 2 stages	9.4 dB
RCS w/ D_4 , 3 stages	9.4 dB
RCS w/ D_5 , 1 stage	7.1 dB
RCS w/ D_5 , 2 stages	8.0 dB
RCS w/ D_5 , 3 stages	8.5 dB
Low-pass filtering with DCT	5.4 dB
Low-pass filtering with DFT	2.0 dB

Table 6.1: Performance of recursive cycle spinning with various Daubechies filter pairs compared to two idealized low-pass filtering techniques.

signal is not known. Therefore, we overestimate the performance of the Fourier low-pass filtering. The results are summarized in Table 6.1 Notice that cycle spinning and recursive cycle spinning both outperform the Fourier techniques. The DCT-based technique outperforms the DFT-based technique by about 3 dB because the original signal is real.

6.4 Effect of Threshold Selection

This section presents a set of simulation results to show the effect of the threshold selection. As discussed in Section 2.3.4, empirical evidence has shown that using a threshold size of 3σ , where σ is the standard deviation of the noise, leads to good denoising performance.

To study the sensitivity of the performance on the threshold value, a threshold of $\alpha \cdot \sigma$ was used with α ranging from 1 to 5. In addition, the true value of σ is generally not known. So experiments were performed with the thresholds $\alpha \cdot \hat{\sigma}$, where $\hat{\sigma}$ is an estimate of the standard deviation σ computed from the noisy data.

Figure 6.6 shows the effect of threshold selection. It confirms that good performance is obtained with a scale factor of $\alpha = 3$. Also, computing the threshold with respect to the estimated noise

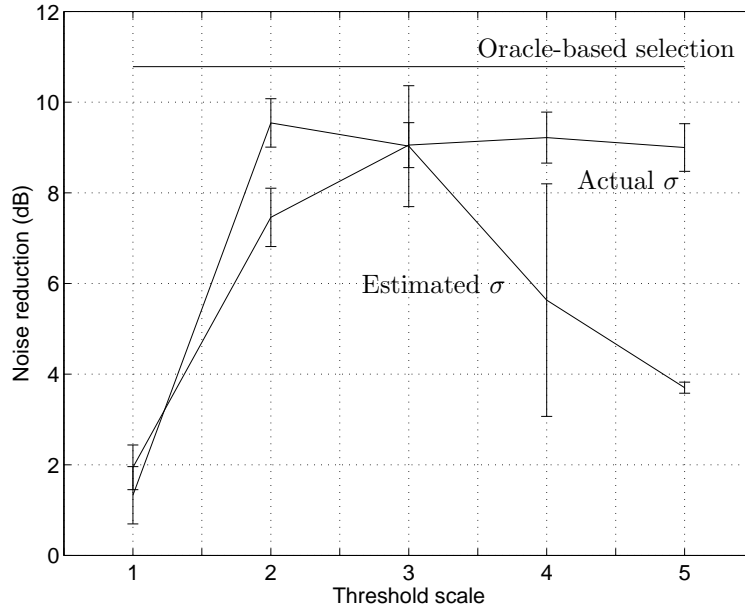


Figure 6.6: Comparison of threshold selection methods.

variance rather than the actual noise variance does not cause performance degradation; however, the sensitivity to α increases.

For comparison, Figure 6.6 also shows the performance with an oracle that sets to zero all of the coefficients for which the signal contribution is zero. The thresholding methods perform within 2 dB of this unimplementable oracle technique.

Bibliography

- [1] T. D. Bui and G. Chen. Translation-invariant denoising using multiwavelets. *IEEE Trans. Signal Proc.*, 46(12):3414–3420, December 1998.
- [2] A. Chambolle, R. A. DeVore, N.-y. Lee, and B. J. Lucier. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans. Image Proc.*, 7(3):319–335, March 1998.
- [3] S. G. Chang, B. Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Proc.*, 9(9):1532–1546, September 2000.
- [4] S. G. Chang, B. Yu, and M. Vetterli. Spatially adaptive wavelet thresholding with context modeling for image denoising. *IEEE Trans. Image Proc.*, 9(9):1522–1531, September 2000.
- [5] S. G. Chang, B. Yu, and M. Vetterli. Wavelet thresholding for multiple noisy image copies. *IEEE Trans. Image Proc.*, 9(9):1631–1635, September 2000.
- [6] R. R. Coifman and D. L. Donoho. Translation-invariant de-noising. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, volume 103 of *Springer Lecture Notes in Statistics*, pages 125–150, New York, 1995. Springer-Verlag.
- [7] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. Inform. Th.*, 36:961–1005, September 1990.
- [8] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.
- [9] D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Th.*, 41(3):613–627, May 1995.

- [10] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, 1994.
- [11] P. L. Dragotti and M. Vetterli. Wavelet transform footprints: Catching singularities for compression and denoising. In *Proc. IEEE Int. Conf. Image Proc.*, volume 2, pages 363–366, Vancouver, September 2000.
- [12] G. Fan and X.-G. Xia. Image denoising using a local contextual hidden markov model in the wavelet domain. *IEEE Sig. Proc. Let.*, 8(5):125–128, May 2001.
- [13] M. Hansen and B. Yu. Wavelet thresholding via MDL for natural images. *IEEE Trans. Inform. Th.*, 46(5):1778–1788, August 2000.
- [14] S. S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, Upper Saddle River, NJ, third edition, 1996.
- [15] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [16] M. Jansen and A. Bultheel. Multiple wavelet threshold estimation by generalized cross validation for images with correlated noise. *IEEE Trans. Image Proc.*, 8(7):947–953, July 1999.
- [17] H. Krim, D. Tucker, S. Mallat, and D. Donoho. On denoising and best signal representation. *IEEE Trans. Inform. Th.*, 45(7):2225–2238, November 1999.
- [18] X. Li and M. T. Orchard. Spatially adaptive image denoising under overcomplete expansion. In *Proc. IEEE Int. Conf. Image Proc.*, volume 3, pages 300–303, Vancouver, September 2000.
- [19] J.S. Lim. *Two-Dimensional Signal and Image Processing*. Prentice Hall, Englewood Cliffs, NJ, 1989.
- [20] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. and Mach. Int.*, 11(7):674–693, July 1989.
- [21] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, second edition, 1999.
- [22] V. P. Melnik, I. Shmulevich, K. Egiazarian, and J. Astola. Block-median pyramidal transform: Analysis and denoising applications. *IEEE Trans. Signal Proc.*, 49(2):364–372, February 2001.

- [23] M. K. Mihçak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Sig. Proc. Let.*, 6(12):300–303, December 1999.
- [24] B. K. Natarajan. Filtering random noise from deterministic signals via data compression. *IEEE Trans. Signal Proc.*, 43(11):2595–2605, November 1995.
- [25] A. V. Oppenheim and R. W. Schaffer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [26] A. V. Oppenheim, A. S. Willsky, and I. T. Young. *Signals and Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [27] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, MA, 1996.
- [28] V. Strela, P. N. Heller, G. Strang, P. Topiwala, and C. Heil. The application of multiwavelet filterbanks to image processing. *IEEE Trans. Image Proc.*, 8(4):548–563, April 1999.
- [29] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [30] N. Weyrich and G. T. Warhola. Wavelet shrinkage and generalized cross validation for image denoising. *IEEE Trans. Image Proc.*, 7(1):82–90, January 1998.
- [31] Y. Xu, J. B. Weaver, D. M. Healy, Jr., and J. Lu. Wavelet transform domain filters: A spatially selective noise filtration technique. *IEEE Trans. Image Proc.*, 3(6):747–758, November 1994.