EVALUATION OF STEP DETECTION OF PIECEWISE CONSTANT SIGNALS USING PHASE CONGRUENCY, REAL FOOTPRINTS AND COMPLEX FOOTPRINTS

by

SRIKANTESWARA SACHIDANANDA

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2011

Dedicated to The Universe

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor Prof. Soontorn Oraintara who has been a source of inspiration and guidance throughout my Masters studies. I have learnt immensely from the courses you taught in class, as well as from the advise you gave me at every stage of my thesis work. I thank you for giving me this opportunity to work in the Multirate Signal Processing lab, and have greatly benefited from this experience. I would also like to thank Prof. Michael Manry and Prof. Saibun Tjuatja for being on my thesis defense committee and for their valuable comments and suggestions.

I would like to thank my UTA lab mates Yothin Rakvongthai, Jixing Yao, Lakshmi Srinivasan and Nha Nguyen for their constant encouragement and suggestions for my work throughout the past year. I am grateful for all the interactions. I would like to acknowledge Yothin Rakvongthai in particular for the guidance and several discussions in my work. The environment in the lab always made one fully comfortable while working.

I thank my parents and family for supporting me during my MS studies, and encouraging me to do my best.

April 15, 2011

ABSTRACT

EVALUATION OF STEP DETECTION OF PIECEWISE CONSTANT SIGNALS USING PHASE CONGRUENCY, REAL FOOTPRINTS AND COMPLEX FOOTPRINTS

SRIKANTESWARA SACHIDANANDA, M.S.

The University of Texas at Arlington, 2011

Supervising Professor: Soontorn Oraintara

Several applications involve the use of signals that are piecewise constant. Often, their measurements or observations are contaminated with noise. Detecting the steps and recovering the true signal from its noisy measurements is an important problem.

There are a number of methods in the literature that address this issue. We use two different approaches. First, we use the idea of phase congruency in detecting singularities in a 1-dimensional piecewise constant signal. Phase congruency has been previously used in feature detection from images [1]. It is a dimensionless quantity that gives an absolute measure in finding the edges or discontinuity in a signal. We calculate phase congruency measure using Interscale, Relative Phase and Derotated Phase wavelet coefficients, and observe their performance in detecting discontinuities.

Second, we build on the work of Dragotti *et al*[2], where the concept of wavelet footprint transform is introduced. We extend this to a complex footprint transform. In image processing and analysis, coefficients from complex transforms are known

to possess better properties than those using a real transform, since they possess both magnitude and phase information. These can be utilized in improving edge and feature detection. We study the properties of the complex footprint representation for 1-dimensional piecewise constant signals. We follow an algorithm developed by Regi-Pique *et al*[3], to detects steps in a piecewise constant signal using it's footprint coefficients.

We studied the application of these methods to the field of bioinformatics. The first application involved step detection in order to find DNA copy number alterations. DNA copy numbers are piecewise constant, and discontinuities in the signal indicate possible genetic irregularities and are useful for cancer diagnosis. We attempt to extend the work of [3] by using complex wavelet footprints for detecting the discontinuities from its noisy measurements.

The second application is with molecular motors data. These are biological motors that operate on a molecular scale, and studying their dynamics is useful in building synthetic motors that replicate their action. The dynamics of these motor are piecewise constant in nature. We use footprint based technique in recovering the signal from its noisy measurements. We observe that this approach outperforms existing signal recovery or noise removal methods.

Thus, this report extends real footprint transform to a complex footprint transform, and uses it in detecting steps in a piecewise constant signal. We also use Phase congruency for detecting discontinuities in such signals. Two applications where these can be implemented are then studied, and the results are analyzed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Recovering a signal from it's noisy observations is a key problem in many applications. Detection theory is the study of how well the desired information can be extracted from the measured signal. This thesis focuses on the problem of detecting singularities from noisy measurements of 1-dimensional piecewise constant signals. For this, we focus on two approaches. Firstly, we use Phase Congruency measure to detect edges or discontinuities in a signal.

The second method of step detection uses the wavelet footprints transform, introduced by Dragotti and Vetterli[2]. An algorithm for step detection using these wavelet footprint coefficients was developed by Regi-Pique *et al*[7] , [3]. We extend the real footprint transform and develop a complex footprint representation. We apply the complex footprint transform on our 1-dimension signal, follow the same algorithm for step detection as [3]. For images, using a complex wavelet representation rather than a real wavelet transform provides some properties which can be used to achieve greater accuracy in edge detection and analysis, Ivan Selesnick *et al*[8]. We study whether the new representation gives better performance in detecting edges in the 1-dimensional signals that we use.

We apply these techniques to the field of bioinformatics. With new technologies such as micro-arrays, there is now a very high volume of raw information obtained at very high resolution, but is heavily corrupted by measurement and technical noise. We study the application of step detection using phase congruency, and wavelet footprints

to Molecular Machine Dynamics and to DNA copy number alterations, both of whose data have underlying piecewise constant structure.

Since the focus is on detecting edges from noisy signals, we first review the theory of signal detection and step detection.

1.1 Literature Review

1.1.1 Signal Detection

A measured signal is never an exact replica of the true underlying data. In general, we obtain data (also called measurements or observations) which consists of signal components embedded in additive noise. Based on the received data we try to determine whether or not a particular event has occurred. The process by which a decision is taken regarding the true state of a measured signal falls under the theory of signal detection.

There are several signal detection techniques in the literature. The simplest forms are Bayes Detection, Maximum A Posteriori Detection and Maximum Likelihood detection [9].

1.1.2 Step Detection Theory

We are concerned with the step detection of piecewise constant (PWC) signals. Piecewise constant signals are characterized by a finite number of constant levels and are commonly corrupted by unknown noise. Figure 1.1 shows some examples of signals that are piecewise constant, and are contaminated by noise. In many cases, the number of levels and their associated values are not known. The signals themselves also change levels randomly. These changes are instantaneous and are finite in number.

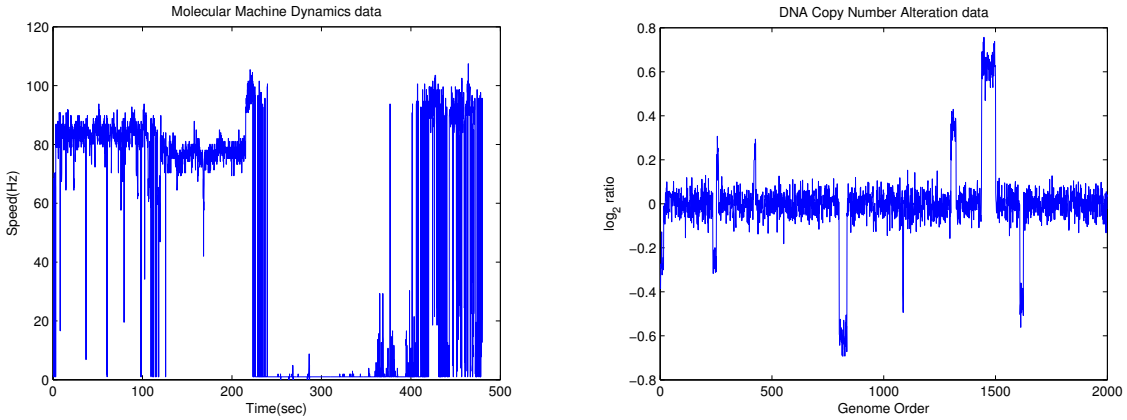


Figure 1.1: Examples of signals that could be modeled as piecewise constant obscured by noise

The main processing task is that of detecting the number of levels and reconstructing the noiseless signal. An abrupt change can also be called a *shift, edge, step, change point, singularity, level change, transition.* The filtering process itself (called *smoothing*) can also be called *detection* or *approximation* or sometimes *segmentation.*

Conventional linear methods include finite impulse response, infinite impulse response filters or fast Fourier transform-based filtering. Signal recovery involves removal of noise and conventional methods typically achieve this by *low-pass filtering*, that is, by removal of high-frequency detail in the signal. This method is effective if the signal to be recovered is sufficiently smooth. But PWC signals are not smooth and low pass filtering of PWC signals typically introduces large spurious oscillations near the jumps known as *Gibb's phenomenon.* The noise and the PWC signal *overlap substantially in the Fourier basis* and so cannot be separated by any basic approach that reduces the magnitude of some Fourier coefficients, which is the way conventional low-pass noise removal typically works.

The Running median filter [10] was a non-linear edge detection technique which was an improvement to the running mean filtering. Canny [11] developed an optimal

smoothing filter for edge detection in images, which is still popular today. He [11] evaluated the detectors by three criteria: good detection, good localization and low spurious response, and he showed that the optimal detector for an isolated step edge should be the first derivative of Gaussian. Hidden Markov Models for edge detection were developed by Godfrey in 1980. Phase Congruency based edge detection in images has also been used by Kovesi[1]. Since it was originally proposed by Marr [12], the Gaussian filter has been widely used smoothing filter in edge detection.

Wavelets have been extensively used for noise removal. Wavelets possess the existence of an algorithm with O(N) computational complexity in the forward and reverse wavelet transforms in the discrete-time setting [13]. Also, many signals in the wavelet basis are *sparse*, that is, a large proportion of the coefficients are effectively zero making the wavelet representation very compact [14]. Haar wavelet has been used for denoising, Cattani [15]. Edge detection by scale multiplication in wavelet domain was developed by Zhang [16].

Since we compare our step detection methods with a median filter and L_1-regularized fused LASSO global filtering, we describe them in more detail below.

- Median filter [10] is a non-linear filtering method used to remove noise from signals. Running Median Filters replaces the middle sample of a moving window that runs through the time series with the median of the samples in that window. The only parameter involved is the window length W. This filter performs different from Running Mean Filters in that it leaves the edge and impulse like root signals unchanged. Also, it is the maximum-likelihood estimate of the *location,* m of the distribution of samples in the window, if they are Laplace distributed [17]. The negative likelihood function of the window samples is:

$$-lnP(x_w|m) = -WlnA + a\sum_{(i\in w)}|x_i - m| \tag{1.1}$$

where w is the size W index set of samples in each window, A is an unimportant normalizing factor, and a is the spread of the Laplace distribution. Minimizing this function with respect to m is equivalent to minimizing $E = \sum_{i \in w} x_i - m$, which is solved when m is the median of the samples [17].

- *L_1-regularized fused LASSO global filtering* [18]: A different *global filtering* approach that finds an optimal solution to an entire time series at once, rather than considering a sliding window of samples. The lasso penalizes a least squares regression by the sum of the absolute values (L_1-norm) of the coefficients. The form of this penalty encourages sparse solutions (with many coefficients equal to 0). For a model defined by $y = f(x) = \beta x$, given the N-length training data $(x_1, y_1), (x_2, y_2)...(x_N, y_N)$, the LASSO (least square shrinkage and selection operator) model coefficients $\hat{\beta}^{lasso}$ are calculated as follows:

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij} \right)^2 \right\}$$

$$\text{subject to: } \sum_{j=1}^{d} |\beta_j| \leq s$$

$$s > 0$$

The model for L_1 fused lasso is as follows:

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij} \right)^2 \right\}$$

$$\text{subject to: } \sum_{j=1}^{d} |\beta_j| \leq s_1$$

$$\text{and } \sum_{j=2}^{p} |\beta_j - \beta_{j-1}| \leq s_2$$

As can be seen, the difference between lasso and fused-lasso is that in lasso, the L_1 regularization is applied to the coefficients β, whereas in fused-lasso, the L_1 regularization is applied to both the coefficients β and also the difference between coefficients $\beta_j - \beta_{j-1}$.

For all finite values of λ, the solution is a piecewise constant curve with finite number of steps, and is simultaneously the least squares fit to the data. Also, this is a convex *quadratic programming* problem guaranteeing that a globally optimal solution can be rapidly found using standard algorithms.

1.2 Footprint Representation: Motivation and Applications

The goal is to minimize the error in approximating the observed PWC signal. So far, we have reviewed some of the popular methods that are used. [3] recently introduced a technique that

- provides a maximally sparse representation of a PWC signal. (known as Wavelet Footprints [2])
- implements an efficient technique for recovering the true PWC signal from this sparse representation. (known as Sparse Bayesian Learning [3], [19])

In the remainder of this chapter, we introduce the idea of Wavelet Footprints, and the motivation for using it for the representation of piecewise constant signals.

Wavelet transforms are powerful tool because it manages to represent both transient and stationary behavior of a signal with a few transform coefficients. Discontinuities in a PWC signal carry relevant signal information and therefore, they represent a critical part to analyze. The footprints dictionary is built from the wavelet transform. Footprints are vectors that model discontinuities in (one-dimensional) piecewise constant signals exactly. They form a basis and allows for compact representation of piecewise constant signals.

Computationally, the representation can be stated as building a dictionary $D = \{f_i\}_{i \in I}$ of elementary functions that can well approximate any signal in the class of PWC signals, with the superposition of a few of its elements. Given a PWC signal g and D, a basis, there is a unique way to express g as a linear combination of the f_i's

The notion of footprints has been introduced by [2]. Given a signal of interest, we first perform the wavelet transform of this signal, and then the wavelet coefficients are expressed in terms of footprints. Together with the scaling coefficients, footprints can represent any piecewise constant signal. Wavelets alone are also efficient at representing discontinuities in a signal, however, the wavelet coefficients generated by a discontinuity are dependent across scales. By constructing the footprint expansion on the wavelet transform, this dependency is removed completely. By representing any PWC signal with the combination of a few footprints, a sparse representation of the signal under consideration is obtained. A PWC signal with K breakpoints can be represented using exactly $K+1$ footprint coefficients.

Our contribution is in the extension of the footprint representation. So far, the footprint representation has been used to generate only real coefficients. In this thesis, we develop a method to extend this dictionary so that complex footprint coefficients are generated, for a given signal of interest. These new coefficients gives us magnitude and phase information - since we now have both real and imaginary components. We study whether this can be used to obtain a more accurate signal reconstruction, than by using only real footprint coefficients.

There are several applications that require the recovery of true PWC signal from noisy measurements. [7], [3] uses wavelet footprints in it's algorithm for for analyzing DNA copy number alterations. We apply the complex footprint representation for the same application. We also, for the first time, use the real footprint representation in the analysis of molecular machine dynamics. Briefly, we introduce the two applications here, a more descriptive explanation is provided in later chapters.

1.2.1 DNA Copy Number Alterations analysis

DNA Copy Number Alterations, which are deletion or replication of chromosomal regions across the genome are known to be associated with the development and behaviour of tumors. Copy numbers correspond to physical losses/gains in genetic material and have an underlying piecewise constant structure. Recovering the true PWC signal from their noisy measurements is key to improving diagnostic and therapeutic strategies [7].

The development of a fast and accurate method to determine the underlying PWC signal's structure- breakpoints and magnitudes is a topic of interest. The footprint transform of such signals provides a sparse representation. A technique known as Sparse Bayesian Learning (SBL) uses these footprint coefficients to infer the copy number changes in the signal.

1.2.2 Molecular Machine Dynamics (MMD)

Molecular systems have evolved naturally within organisms, and perform certain specialized tasks such as pumps, motors, copiers etc - and are known as Molecular Machines. Understanding the functioning of these machines allows for the development of artificial molecular devices. However, this is a challenging task at this scale due to molecular measurement noise. [20]

Molecular Motors are one example of molecular machines. These motors operate in a series of step-like motion. The observation of these step motions are hampered by the presence of measurement noise. Removing the noise and recovering the underlying signal, which is a combination of steps and impulses, is the problem we try to solve. Footprint representation, followed by Sparse Bayesian Learning is applied to this data, and the true signal is recovered.

This report is organized as follows. Chapter II discusses the use of Phase Congruency using wavelets, for step detection. Chapter III discusses the theory of Wavelet Footprints. Chapter IV discusses about Real and Complex footprints representation, and their properties. Chapter V discusses the two applications that we have used Wavelet Footprints method in, along with theory, simulations, results and discussions. Chapter VI summarizes the work and provides concluding remarks.

CHAPTER 2

PHASE CONGRUENCY

2.1 Introduction

Features such as edges or lines in a signal (image) give rise to points where the Fourier components of a signal are maximally in phase. A model, based on this theory, for extracting features (such as edges) from a signal, called the Local Energy Model was developed by [4]. Phase varies linearly around a point of discontinuity. Phase congruency further develops on this - it is a dimensionless quantity that provides an absolute measure of the significance of feature points. It is a measure of how well phase is aligned at a discontinuity. Values of phase congruency (PC) vary from a maximum of 1 (indicating a very significant feature) to 0 (indicating no significance). Kovesi [1] has shown how phase congruency can be calculated from log Gabor wavelets. We study this model of phase congruency, applied to 1-dimensional signals and it's ability in detecting discontinuities in a piecewise constant signal.

The local energy model for feature (edge) detection states that features are perceived at points where the Fourier components are maximally in phase. For example, looking at the Fourier series that makes a square wave, all the Fourier components are sine waves which are exactly in phase at the point of the step. At all other points in the square wave, the PC is low. The Figure 2.1 explains this concept. In both diagrams, the solid line is the sum of the Fourier basis functions represented in dashed lines. [4] has developed a phase congruency measure using log Gabor wavelets.

(a) (b)

Figure 2.1: The Fourier components are all in phase at (a) the point of step in the square wave, and (b) at peaks and troughs in the triangular wave [4]

2.2 Computing Phase Congruency using wavelets

We are interested in calculating local frequency, and in particular, phase information in signals. To preserve phase information, linear-phase filters are used. [4] uses logarithmic Gabor functions suggested by [21]. These filters have the Gaussian transfer function when viewed on the frequency scale. On a linear scale, the log Gabor function has a transfer function of the form

$$G(w) = e^{\frac{-(log(w/w_0))^2}{2(log(\kappa/w_0))^2}}$$

where w_0 is the filter's center frequency.

In its basic form, the points of maximum phase congruency can be calculated by searching for peaks in the local energy function. [22]. Given 1-dimensional signal I(x), the local energy function is defined as

$$E(x) = \sqrt{F^2(x) + H^2(x)}$$

where $F(x)$ is the signal $I(x)$ with it's mean removed, and $H(x)$ is the Hilbert transform of $F(x)$.

$$PhaseCongruency = E(x)/\sum_n A_n$$

where A_n is the amplitude of the nth Fourier component. Thus, local energy function is directly proportional to the phase congruency function, so peaks in local energy will correspond to peaks in phase congruency.

A more involved Phase Congruency measure is as below.

$$PC = \frac{\sum_n \lfloor A_n(x)\Delta\phi_n(x) - T \rfloor}{\sum_n A_n(x) + \epsilon} \tag{2.1}$$

with $\Delta\phi_n(x) = cos(\phi_n(x) - \bar{\phi}(x) - |sin(\phi_n(x) - \bar{\phi}(x))|)$

ϵ is a small constant to avoid division by zero, T is an (empirically calculated) threshold based on the noise influence in the signal, $\Delta\phi_n(x)$ is a phase deviation function that is based on the phase angle $\phi_n(x)$ and the overall mean phase angle $\bar{\phi}_n(x)$ - as explained in the Figure 2.2. Further explanation about Phase congruency is available at [1]. Figure 2.2 shows the Fourier components at a location in the signal plotted head to tail. This arrangement illustrates Energy vector, the sum of the Fourier components, and the phase congruency of the signal. A more detailed explanation is available at [22].

2.3 Phase Congruency Measures used

Given a complex wavelet transform such as the log Gabor transform, we use the following phase relationships between the wavelet coefficients, while computing the Phase Congruency, Figure 2.3.

- Relative Phase coefficients: Relative phase is defined as the difference of phases between two adjacent complex wavelet coefficients at a given scale. Given a

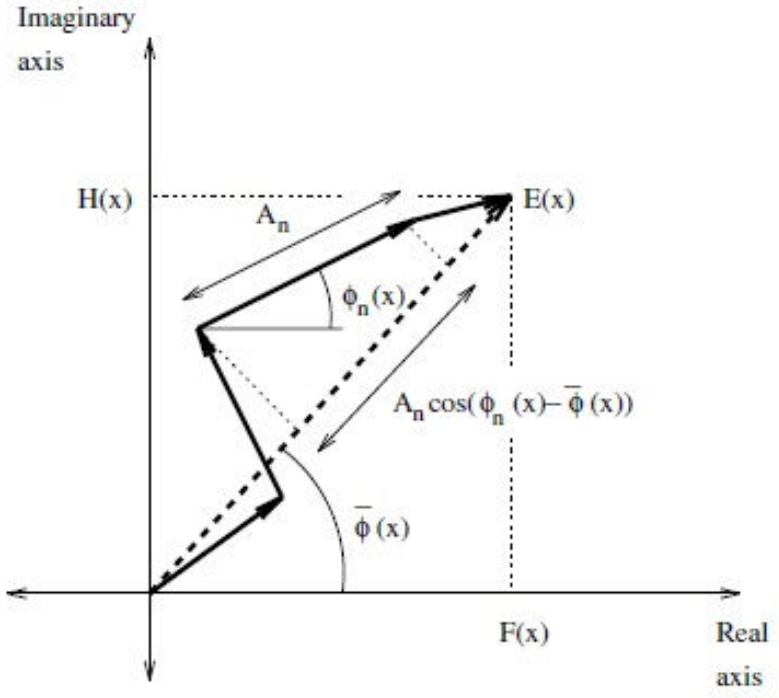


Figure 2.2: Polar diagram showing the Fourier components at a location in the signal plotted head to tail. [1]

wavelet coefficient C_a and it's adjacent coefficient C_b, $C_a - C_b$ is the relative phase. The relative phase coefficient C takes the magnitude of C_a.

$$C = |C_a| e^{j \angle C_a C_b^*} \tag{2.2}$$

Further details are available at [23].

- Derotated Phase coefficients: If C_1 is a wavelet coefficient at a given scale and translation, and C_2 is the corresponding coefficient at the next coarser scale.

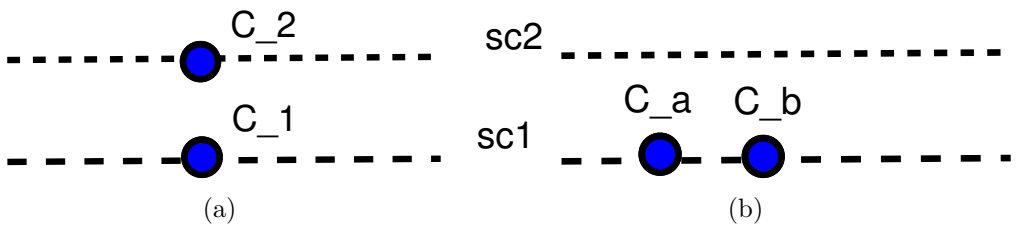


Figure 2.3: Phase relationships of complex coeffs: (a) Inter scale coeffs (b)Intra scale coeffs

The derotated phase coefficient has the magnitude of the C_1 coefficient, while the phase is $\angle C = \angle C_1 - 2\angle C_2$, or

$$C = |C_1| e^{j\angle C_1 (C_2 . C_2^*)} \tag{2.3}$$

Thus, this new element involves use of coefficients whose phase has been derotated by twice the phase of their interpolated coefficient at next coarser scale. Further details on derotated wavelet coefficients are available in [24].

2.4 Simulation and Results

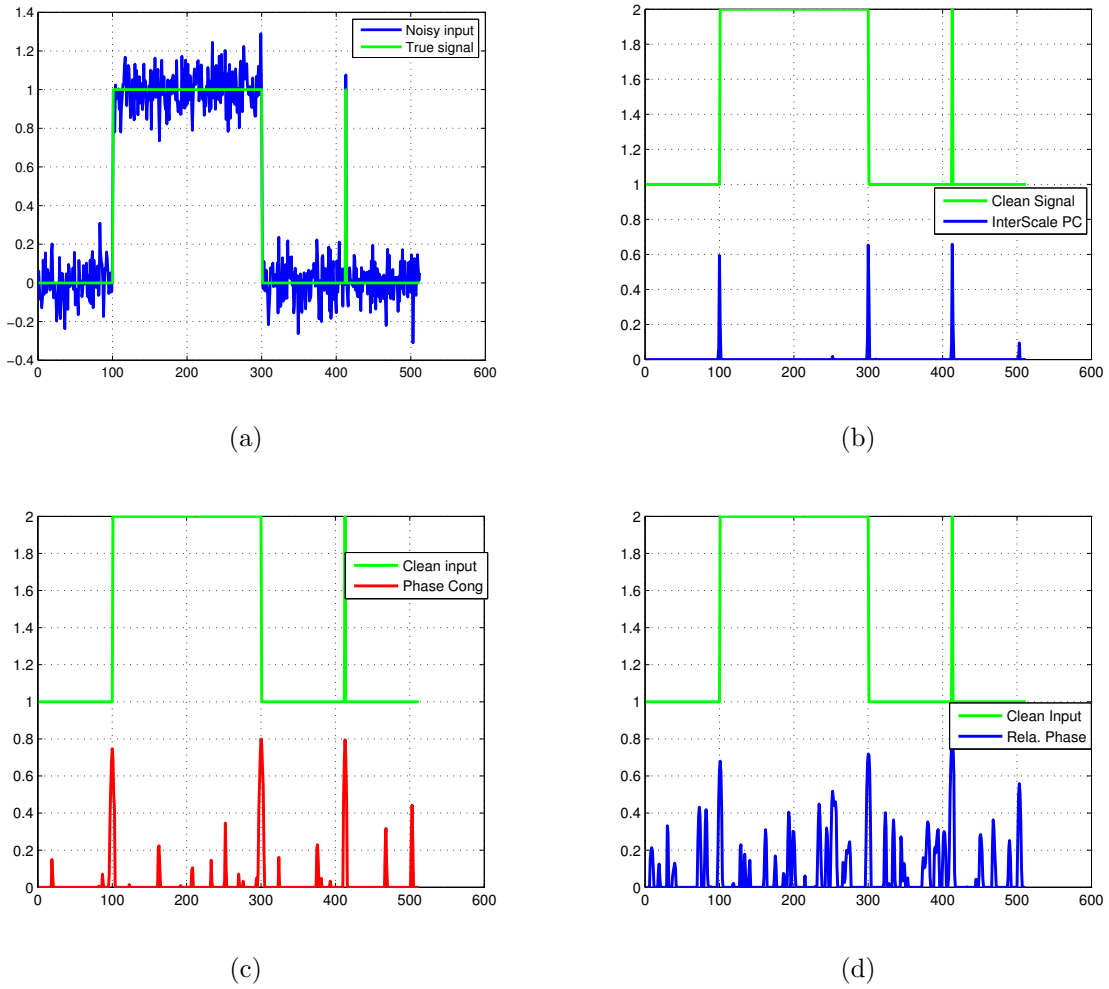
We simulate an input signal with a step and an impulse, and study the performances of the 3 methods of calculating Phase Congruency discussed above.

From Figure 2.4, we observe that for an input signal contaminated with noise having standard deviation 0.1, PC calculated using Interscale coefficients and Derotated phase are able to detect discontinuities in the underlying signal, whereas relative phase detects peaks due to the noise as well. We now proceed to analyze the PC performances using a numerical measure.

The measures we used to analyze the accuracy of PC are: Sensitivity and False Discovery Rate(FDR).

$$\text{Sensitivity} = \frac{\#\text{ discontinuities detected correctly}}{\text{Total }\#\text{ discontinuities present}}$$

Sensitivity close to 1 indicates a good detection performance.



(a) (b)

(c) (d)

Figure 2.4: (a) Simulated input Signal ($\sigma = 0.1$); Phase Congruency : (b) InterScale coefficients, (c) Derotated Phase, and (d) Relative Phase coefficients

$$\text{FDR} = \frac{\#\text{ discontinuities detected incorrectly}}{\text{Total }\#\text{ discontinuities detected}}$$

The FDR should ideally be close to 0.

We calculated the Sensitivity and FDR of a simulated PWC input signal of length 2000, and averaged the result over 100 iterations. The input signal is a standard synthetic input used for analysis of DNA copy number alterations. This is

based on the model proposed in [6]. The detailed construction of this input signal is discussed in chapter 5 Section 5.2.2. To describe in short, the true signal is PWC consisting of 6 discrete levels with amplitudes between 0 and 1. The input signal is then contaminated by noise having standard deviation ranging from $0 - 0.25$. We test the performance of Phase Congruency for these values of noise. For a favorable result, we require the Sensitivity to be close to 1 and FDR close to 0 for all noise standard variances.

Table 2.1: Performance metrics of Phase Congruence applied on a synthetic input signal

Noise Std Dev	Phase Congruence	Sensitivity	FDR
0.01 - 0.1	DP	0.9555	0.4589
	IS	0.9615	0.1739
	RP	0.9444	0.6222
0.05 - 0.15	DP	0.0.6667	0.1923
	IS	0.8	0.3422
	RP	0.5	0.7585
0.1 - 0.2	DP	0.3717	0.2857
	IS	0.5	0.3717
	RP	0.2941	0.845

As observed from the Table 2.1, the performance of Phase Congruency in terms of Sensitivity and FDR is good for noise standard devation in the range from 0 upto 0.1. Of these, PC calculated using InterScale coefficients performs better than that using Relative Phase or Derotated Phase coefficients. At levels of noise standard deviation in the range 0.1-0.2, none of the PC measures performs satisfactorily - the sensitivity is at most 0.5, which indicates a poor detection ability, while the FDR is also quite high. This suggests that Phase Congruency is a good measure for detecting

edges in signals that are contaminated with low levels of noise, but are not as useful when the signals are contaminated by higher levels of noise.

CHAPTER 3

THEORY OF FOOTPRINTS

3.1 Introduction

Wavelets are able to characterize the local regularity of a function. They are known to be efficient in representing piecewise smooth (including piecewise constant) functions. For piecewise smooth signals, the wavelet coefficients are non zero only around discontinuities. This is equivalent to saying that all the information about the signal is contained in the few significant wavelet coefficients around the discontinuities. Away from singularities, the inner product between a wavelet and a smooth function will be either zero or very small. At singular points, a finite number of wavelets concentrated around the discontinuity lead to non-zero inner products.

As explained in [2], these wavelet coefficients generated at the singularities are highly dependent across scales. In traditional wavelet based compression and denoising algorithm these coefficients are processed independently. However they should be gathered in a vector and jointly processed. Footprints are used for this purpose. A footprint is a vector containing all significant wavelet coefficients across scales around a discontinuity. For instance if our wavelet filter has length L and we have J wavelet decomposition levels then the footprint of a discontinuity is a vector of dimension $(J+1)\times(L-1)$ containing $L-1$ wavelet coefficients at each scale in the position corresponding to the discontinuity position plus the $L-1$ scaling coefficients at the same position.

Given a PWC signal of interest, the wavelet transform is performed on it to obtain the wavelet coefficients. These coefficients are then represented in terms of

footprints. Along with the scaling coefficients, the footprints can completely represent any piecewise polynomial (including PWC) signal. By representing any discontinuity with the combination of a few footprints, we can get a sparser representation of the signal under consideration.

In the following sections, we discuss the development of wavelet footprints and mathematical notations. First, we discuss the dependence of wavelet coefficients across scales, and then discuss about building a footprint dictionary using wavelet basis. We also provide a simple example at the end of the chapter to illustrate the concepts discussed.

3.2 Dependence of wavelet coefficients across scales - Motivation for developing footprints

In this section, we focus on the analysis of the dependency of the wavelet coefficients across scales, generated by discontinuities in a piecewise smooth signal. A piecewise constant signal is a subset of this class of signals.

Consider an orthonormal wavelet series with scale and shift parameters m and n respectively. (where small scales correspond to large m)

$$\psi_{m,n}(t) = \frac{1}{2^{m/2}}\psi(2^{-m}t - n) \quad m, n \in Z$$

and $\psi(t)$ is the wavelet basis function.

Also, assume that the wavelet has k vanishing moments, that is

$$\int_{-\infty}^{\infty} t^d \psi(t) dt = 0, \quad d = 0, 1, ..k-1$$

Then, as stated in [25], the wavelet coefficients of a piecewise smooth function decay rapidly around the discontinuities in the signal. Because of this decay property, larger wavelet coefficients tend to be around the singular parts of a PWC signal. These wavelet coefficients gather most of the energy of the original signal.We need

to analyze the dependency across scales of the wavelet coefficients generated by these piecewise constant discontinuities.

Consider the simple case of a piecewise constant function with only one discontinuity at location t_1 i.e. $p(t) = a_0^{(0)}\mathbf{1}_{[0,t_1)}(t) + a_1^{(0)}\mathbf{1}_{[t_1,T)}(t)$ and a wavelet series with one vanishing moment and compact support (eg: Haar Wavelet). The decomposition of this signal in the wavelet basis results in zero wavelet coefficients, except in the *cone of influence*[25] of t_1. The cone of influence of t_1 is the set of points (m, n) such that t_1 is included in the support of $\psi_{(m,n)}(t)$, as in Figure 3.1. In this case, the wavelet coefficients in this cone of influence are dependent - they have only one degree of freedom.

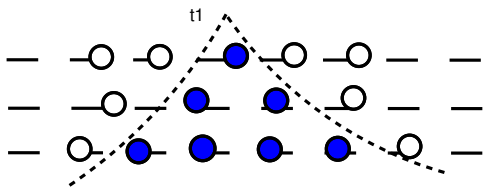


Figure 3.1: Cone of Influence

This can be shown since a wavelet with k vanishing moments can be written as a kth-order derivative of a function θ which also has a fast decay[25]. The following conditions are true: $\psi(t) = (-1)^k(d^k\theta(t)/dt^k)$. It can be shown that

$$\begin{aligned}\langle p(t), \psi_{m,n}(t)\rangle &= 2^m \int \frac{dp(t)}{dt}\theta_{m,n}(t)dt \\ &= 2^m \int (a_1^{(0)} - a_0^{(0)})\delta(t - t_1)\theta_{(m,n)}(t)dt\end{aligned}$$

where the property $\langle p(t), (d\theta(t)/dt)\rangle = -\langle (dp(t)/dt), \theta(t)\rangle$ has been used. Thus, if the wavelet has compact support, $\langle p(t), \psi_{m,n}(t)\rangle$ is equal to zero if $\psi_{m,n}(t)$ does not overlap t_1 and depends only on the difference $a_1^{(0)} - a_0^{(0)}$ otherwise.

This means that the wavelet behaviour across scales is deterministic. If one knows the value of a single nonzero wavelet coefficient in the cone of influence of t_1, one can derive from it all the other wavelet coefficients in that cone of influence.

Hence, we can see that although piecewise smooth signals are well represented by wavelets, it is possible to model them in a more efficient way. This leads to the development of footprint representation.

3.3 Wavelet Footprints

A footprint dictionary is constructed, for the class of discrete piecewise constant signals using Haar wavelet basis [2]. For this, we introduce the following discrete time wavelet operators:

- $\psi_{j\ l}[n]$ denotes the wavelet function at scale j and shift l
- $\phi_{Jl}[n]$ denotes the scaling function at shift l

Consider a discrete-time piecewise constant signal $x[n], n \in [0, N-1]$ with only one discontinuity at position k, and consider a J level wavelet decomposition of the signal using Haar wavelets:

$$x[n] = \sum_{l=0}^{N/2^J-1} c_l \phi_{jl}[n] + \sum_{j=1}^{J} \sum_{l=0}^{N/2^j-1} y_{jl} \psi_{jl}[n] \tag{3.1}$$

where $y_{jl} = \langle x, \psi_{jl} \rangle$, and $c_l = \langle x, \phi_{Jl} \rangle$ Since Haar wavelet has one vanishing moment and finite support, the nonzero wavelet coefficients of this decomposition are only in the cone of influence of k. Thus the above equation becomes

$$x[n] = \sum_{l=0}^{N/2^J-1} c_l \phi_{Jl}[n] + \sum_{j=1}^{J} y_{jk_j} \psi_{jk_j}[n]$$

where $k_j = \lfloor k/2^j \rfloor$. As can be seen, all these coefficients depend only on the amplitude of the discontinuity at k. Thus, if we define a vector that contains all of them, then

one can specify any step discontinuity at k by multiplying this vector by the right factor.

Hence, a footprint can be defined as follows [2]: *Given a piecewise constant signal x with only one discontinuity at position k, a footprint $f_k^{(0)}$ is a scale space vector obtained by gathering together all the wavelet coefficients in the cone of influence of k and then imposing $\|f_k^{(0)}\| = 1$.* Expressed in the wavelet basis, this footprint can be written as $f_k^{(0)}[n] = \sum_{j=1}^{J} d_{jk_j}\psi_{jk_j}[n]$ where $d_{jk_j} = y_{jk_j}/\sqrt{\sum_{j=1}^{J} y_{jk_j}^2}$.

Hence, any piecewise constant signal $x[n]$ with a step discontinuity at k can be represented in terms of the scaling functions $\phi_{Jl}[n]$ and $f_k^{(0)}$. The signal $x(n)$ from 3.1 becomes

$$x[n] = \sum_{l=0}^{2^J-1} c_l\phi_{Jl}[n] + \alpha f_k^{(0)}[n]$$

where $\alpha = \left\langle x, f_k^{(0)} \right\rangle = \sum_{j=1}^{J} y_{jk_j}d_{jk_j}$. The above discussion can be repeated for any other step discontinuity at different locations, and for each location l we have a different footprint $f_l^{(0)}$. Let $D = \{f_k^{(0)}, k = 0, 1, ..., N-1\}$ be the complete dictionary of footprints. Thus, we see that any piecewise constant function $x[n], n \in [0, N-1]$ can be represented in terms of scaling function and footprints. In particular, if x is a PWC function with K discontinuities, together with the scaling functions, K footprints are sufficient to represent it. A PWC signal with only one discontinuity can be expressed in terms of one footprint, and PWC signals with K discontinuities are given by the superposition of K piecewise constant signals with only one discontinuity. Therefore the footprint representation of a signal x with K discontinuities at positions $k_1, k_2, ..k_K$ is given by

$$x[n] = \sum_{l=0}^{N/2^J-1} c_l\phi_{Jl}[n] + \sum_{i=1}^{K} \alpha_i f_{k_i}^{(0)}[n] \tag{3.2}$$

For the case where $J = log_2(N)$, (N being a power of 2), a footprint vector is a **pure step function**. Hence the footprint basis then becomes a set of linearly independent step functions. Intuitively, this will allow sparse representation of a PWC signal.

3.4 An Example

We illustrate how to develop a footprint basis to represent any PWC vector with a small example. We choose $N = 8$, $J = log_2(N) = 3$, and use the Haar wavelet basis for taking wavelet transform.

The 8×8 Haar basis matrix is:

$$F = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

Denote each row of the above matrix by $\psi_n \quad n = 1, ..., 8$

- Consider an input $x = [\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ ]$. Looking at 3.2, we can re-write x as $x = \sum_{n=1}^{8} \langle x, \psi_n \rangle \psi_n$

  $= 0.87\psi_1 - 0.12\psi_2 - 0.18\psi_3 - 0.25\psi_5$

$$= 0.87 \overbrace{\psi_1}^{\text{scaling fn.}} + 1 \underbrace{[-0.88 \;\; 0.12 \;\; 0.12 \;\; 0.12 \;\; 0.12 \;\; 0.12 \;\; 0.12 \;\; 0.12]}_{\text{footprint vector}}$$

Ignoring the scaling coefficient $0.87\psi_1$, the resultant vector is a step function. This is the footprint vector $f_1^{(0)}$.

In a similar manner, we can develop footprint vectors for other x's, and thus obtain the footprint dictionary for representing PWC signals of length 8. Figure 3.2 shows the footprint basis developed in this manner. This is a set of basis vectors that can fully represent any PWC signal of length 8 in a maximally sparse manner. For the Haar basis that we have used, these basis vectors themselves are PWC in nature.

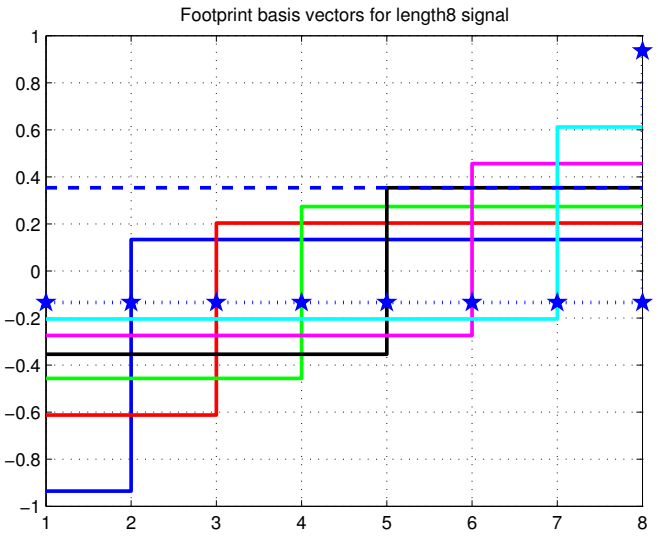


Figure 3.2: Complete footprint basis vectors for representing length-8 PWC sequences. Each vector is a step function. The set of basis vectors are linearly independent

CHAPTER 4

DEVELOPMENT OF REAL AND COMPLEX FOOTPRINTS

4.1 Introduction

In this chapter, we build on the theory from the previous chapter. We construct Real Footprints using Haar wavelet basis and form a sparse representation for piecewise constant functions. We then develop Complex Footprints for the same. With an example, we discuss the basis functions used and compare Real and Complex Footprints.

4.2 Real Footprints

In the previous chapter, wavelet footprints have been proposed in the design of complete dictionaries for representing piecewise constant signals. For piecewise constant signals and Haar wavelets (and $N = 2^J$), the wavelet footprint dictionary is formed by a set of vectors, where each vector f_k is a simple step function with one discontinuity between $k-1$ and k, $\sum_{m=1}^{M} f_k(m) = 0$ and $\|f_k\|^2 = 1$ for $k = 1, ..., M-1$. These properties can be used to formulate the footprint dictionary for signals of arbitrary length M (not just a power of 2) as:

$$f_k(m) = \begin{cases} \sqrt{\frac{M-k}{kN}} & \text{if } m \leq i \\ \sqrt{\frac{k}{M(M-k)}} & \text{if } m > i \end{cases} \tag{4.1}$$

where $k = 1, ..., (M-1)$, and $f_0(m) = \frac{1}{\sqrt{M}}$ is defined to be the DC component [7] . Consider a piecewise constant vector x with K breakpoints. We know that the signal can be completely represented by a linear combination of K step vectors f_i (each with a single breakpoint as discussed above) plus a constant vector f_0

Let us construct $F = [f_0, f_1, ..., f_{M-1}]$. With this notation, the footprint representation for the PWC vector x can be written compactly as:

$$\underset{[M\times 1]}{x} = \underset{[M\times M][M\times 1]}{F}\ \underset{}{w} \tag{4.2}$$

$$\underset{[M\times 1]}{w} = \underset{[M\times M][M\times 1]}{F^{-1}}\ \underset{}{x} \tag{4.3}$$

where $x = (x_0, x_1, ..., x_{M-1})$ is the input PWC signal and $w = (w_0, w_1, ..., w_{M-1})$ is the vector of footprint coefficients (or weights).

This representation has the following properties:

- The columns of F are a *basis* for the R^M i.e. for any $x \in R^M$ there exists a unique w such that $x = Fw$.
- Any arbitrary PWC vector with exactly K breakpoints can be represented with K+1 non-zero footprint coefficients.

4.3 Complex Footprints

We now look to extend the theory of Real Footprints to Complex Footprints - by computing a set of complex coefficients. All computation is done on real numbers - no complex arithmetic is required for the implementation of the complex footprint transform. Before doing so, we give a brief background on the Hilbert Transform and its interaction with the Fourier Transform.

4.3.1 Analytic signal and Hilbert Transform

Let $g(t)$ be a real valued finite energy signal defined over the time interval $-\infty < t < \infty$ with Fourier transform:

$$G(f) = \int_{-\infty}^{\infty} x(t)\ exp\ (-j\ 2\ \pi\ f\ t)\ dt$$

defined over the frequency interval $-\infty < f < \infty$. Because $g(t)$ is real, the FT is complex conjugate symmetric, i.e. $G(-f) = -G*(f)$. The magnitude spectrum is symmetric and infinite frequency extent for a non-bandlimited real-valued signal.

The continuous time analytic signal $g_a(t)$ corresponding to $g(t)$ is defined in the frequency domain as :

$$G_a(f) = \begin{cases} 2G(f) & \text{if } f > 0 \\ G(0) & \text{if } f = 0 \\ 0 & \text{if } f < 0 \end{cases} \tag{4.4}$$

which is inverse transformed to $g_a(t)$.

Due to it's one sided spectral definition, the analytic signal $g_a(t)$ will be necessarily complex-valued and can therefore be represented in terms of its real and imaginary components $g_a(t) = g_r(t) + j\ g_i(t)$, for which $g_r(t) = Re\{g_a(t)\}$ and $g_i(t) = Im\{g_a(t)\}$ are both real valued functions. It can be shown [26] that:

$$g_r(t) = g(t) \quad and \quad g_i(t) = HT\{g(t)\} \tag{4.5}$$

in which HT designates the Hilbert Transform operation.

Hilbert transform operation is a two sided ($-\infty < t < \infty$) time-domain convolution of g(t) with the function $\frac{1}{\pi t}$. The Fourier transform of the real component is $G_r(f) = G(f)$, which is a conjugate symmetric (even) function. The Fourier transform of the imaginary component is

$$G_i(f) = \begin{cases} G(f) & \text{if } f > 0 \\ 0 & \text{if } f = 0 \\ -G^*(-f) & \text{if } f < 0 \end{cases} \tag{4.6}$$

which is a conjugate anti-symmetric (odd) function. Combining $G_r(f)$ and $G_i(f)$ then yields the definition 4.4.

The analytic signal created has an important property, specifically, the orthogonality between the real and imaginary components of the analytic signal:

$$\int_{-\infty}^{\infty} g_r(t)\ g_i(t)\ dt = 0$$

Another interesting property of the Hilbert transform is $\mathcal{H}(\mathcal{H}(f)) = -\mathcal{I}$, for any signal f, where $\mathcal{I}$ is the identity matrix.

4.3.2 Properties of discrete time complex signal

An analytic signal is a complex-valued continuous time function with a Fourier transform that vanishes for negative frequencies. A discrete time complex sequence can only approximate an analytic signal. We now consider the properties of a discrete-time complex signal that is close to an analytic signal. There are 2 properties that must be satisfied in order that a length-N discrete signal $g_a[n] = g_r[n] + j\ g_i[n]$ to be an analytic-like discrete signal. First, the real part must exactly yield the original discrete time sequence

$$g_r[n] = g[n]$$

Second, the real and imaginary components must be orthogonal to each other:

$$\sum_{n=0}^{N-1} g_r[n]\ g_i[n] = 0 \tag{4.7}$$

However, due to finite numerical precision during computation, exact orthogonality may not be possible.

4.3.3 Discrete Hilbert transform matrix

We require the discrete Hilbert transform matrix in order to compute imaginary components of the footprint coefficients. The ideal Hilbert transform is infinite length, and hence needs to be approximated.

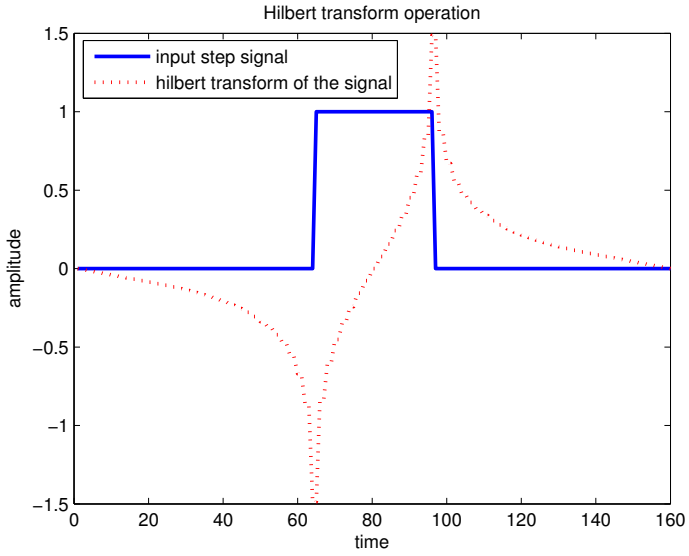


Figure 4.1: An illustration of a PWC Signal and it's Hilbert transform pair.

The procedure that we follow to compute the Discrete Hilbert Transform of an N-length signal $g[n]$ is as follows [26]:

- Compute the N-point DFT G[f] of g[n]
- Form the N-point transform of the imaginary component of the signal using the relation 4.6
- Compute the N-point inverse DFT to obtain a time domain signal.

The signal thus obtained is the imaginary component of the discrete time complex signal.

The standard N-point DFT of a signal $x[n]$ is

$$X(f) = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \qquad 0 \leq k \leq N-1$$

By setting $W_N = e^{-j2\pi/N}$ we re-write this by

$$X(f) = \sum_{n=0}^{N-1} x[n]W_N^{kn} \qquad 0 \leq k \leq N-1$$

In our computations, we require that the Discrete Hilbert transform matrix we develop be invertible i.e. have full rank. If we use the standard DFT matrix for the

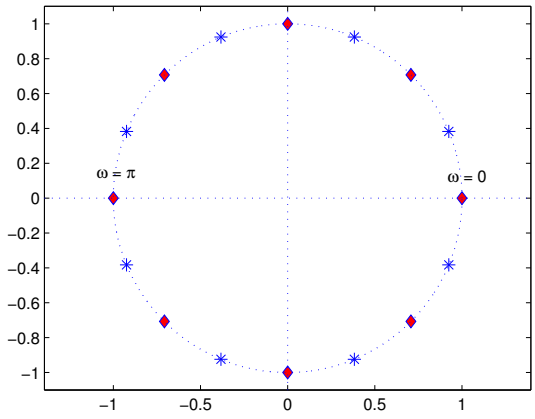


Figure 4.2: Modified DFT sampling points

Discrete Hilbert Transform matrix computation, we find that the matrix is not of full rank. We use a modified DFT, defined by:

$$X(f) = \sum_{n=0}^{N-1} x[n] W_{2N}^{(2k+1)n} \qquad 0 \leq k \leq N-1$$

where $W_N = e^{-j\pi/N}$

This is detailed in the Figure 4.2, where the blue marks denote the new frequency sampling points, compared with the normal sampling points marked in red . With this frequency shift, we obtain a discrete Hilbert transform matrix that is full rank and invertible.

4.4 Determining Complex Footprints coefficients

We find a vector of coefficients, that contains both the real and imaginary components. The real and imaginary parts are considered as separate elements within the vector. That is, if we have K complex coefficients, the vector we compute is of length $2K$. We need to develop equations for the complex case that are analogous to 4.2 and 4.3, i.e.: $x = Fw \qquad and \qquad w = F^{-1}x$

where F is the footprint basis matrix and w are the footprint coefficients.

That is, we need to develop equations of the form

$x = \tilde{F}\tilde{w} \qquad or \qquad \tilde{w} = \tilde{F}^{-1}x$

where $\tilde{F}$ denotes the new footprint basis matrix, and $\tilde{w}$ the new footprint coefficients with real and imaginary components.

$\tilde{w}$ is the vector of *real* and *imaginary* components, $\tilde{w} = \underset{[2M\times 1]}{\begin{bmatrix} w_R \\ w_I \end{bmatrix}} = \underset{[2M\times M]}{\tilde{F}^{-1}} \underset{[M\times 1]}{x}$.

From the above equation, we see that w_R can be computed using an [M × M] basis (say F_R) and w_I is computed using a different [M × M] basis (say F_I). In order to compute $\tilde{F}^{-1}$, we refer [8].

[8] states that if the two real wavelet basis matrices are represented by the square matrices F_1 and F_2, then the complex wavelet (footprint) basis matrix can be represented by the rectangular matrix

$$\underset{[2M\times M]}{F_C} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \tag{4.8}$$

and

$$\underset{[2M\times 1]}{\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}} = \underset{[2M\times M]}{\begin{bmatrix} F_1 \\ F_2 \end{bmatrix}} \underset{[M\times 1]}{x}$$

If the vector x represents a real signal, then $w_1 = F_1 x$ represents the real part and $w_2 = F_2 x$ represents the imaginary part of the complex footprint coefficients. The complex coefficients are given by $w_1 + jw_2$.

A (left) inverse of F_C is given by

$$\underset{[M\times 2M]}{F_C^{-1}} = \frac{1}{2}[F_1^{-1} \quad F_2^{-1}] \tag{4.9}$$

and we can verify that

$$F_C^{-1}.F_C = \frac{1}{2}\begin{bmatrix} F_1^{-1} & F_2^{-1} \end{bmatrix}\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} I + I \end{bmatrix} = I \tag{4.10}$$

Hence, $x = \tilde{F}^{-1}\tilde{w}$ can be re-written as

$$x = \frac{1}{2}\begin{bmatrix} F_R^{-1} & F_I^{-1} \end{bmatrix}\begin{bmatrix} w_R \\ w_I \end{bmatrix} \tag{4.11}$$

Note that the original signal x can be recovered from either the real part or the imaginary part alone; however, such inverse footprints do not capture all the advantages a complex transform offers.

- Computation of F_R^{-1}: Comparing 4.11,and 4.2 we can see that $F_R^{-1}w_R = Fw_R \implies F_R^{-1} = F$
- Computation of F_I^{-1}: We know that Hilbert Transform is used in forming the imaginary component of a discrete complex signal. We compute the imaginary coefficients as $w_I = F_I x$ where $F_I = F_R\mathcal{H}$ and $\mathcal{H}$ being the discrete hilbert transform matrix.

$$x = F_I^{-1}w_I \tag{4.12}$$

where

$$F_I^{-1} = (F_R\mathcal{H})^{-1} = \mathcal{H}^{-1}{F_R}^{-1} = -\mathcal{H}F \tag{4.13}$$

Thus, given a real signal x, we have its footprint representation as

$$x = \frac{1}{2}\begin{bmatrix} F & -\mathcal{H}F \end{bmatrix}\begin{bmatrix} w_R \\ w_I \end{bmatrix} \tag{4.14}$$

The objective is to compute w_R and w_I and form a set of complex coefficients $w_R + jw_I$. We use the magnitude of the complex coefficients for our calculations. In

image analysis, it is known that using the magnitude of a complex transform coefficient, one can represent edges and features more accurately than by using only real coefficients. With this motivation, we apply the same theory to the one-dimensional signals that we consider, and observe the results.

4.4.1 Example

We give a small example for comparing Real and Complex footprint coefficients.

Consider an arbitrary PWC input signal x of length 180 having an impulse at the $t = 100^{th}$ sample. We compute the real footprint coefficients using $w = Fx$ and the complex footprint coefficients using the relation $\begin{bmatrix} w_R \\ w_I \end{bmatrix} = \tilde{F}^{-1}x$. As stated above, we take the magnitude of the complex coefficients. This is shown in Figure

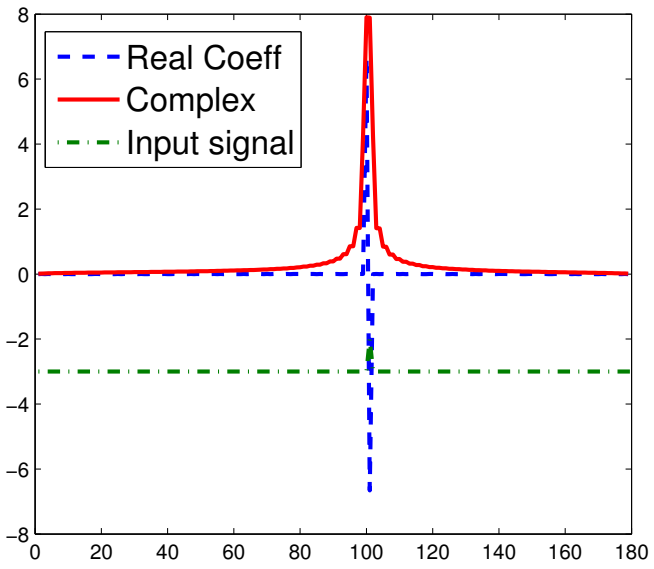


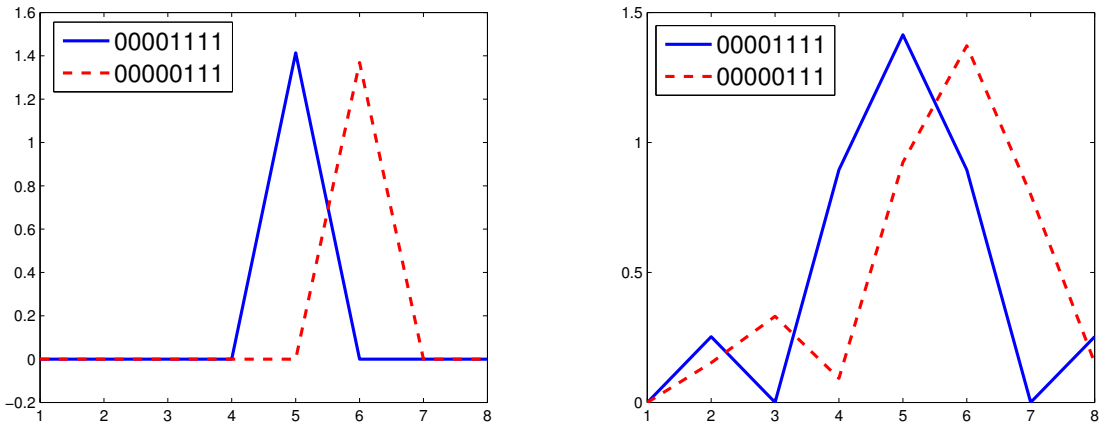
Figure 4.3: Real and complex footprint coefficients

4.3. We observe that the complex coefficient's magnitude is greater than the real

coefficient's alone. This could lead to better detection of a discontinuity. However, the width of the lobe of the complex coefficient magnitude is greater than that of the real coefficient. This indicates that the complex coefficients may not be able to give a more accurate location of a discontinuity than real coefficients.

4.4.2 Shift invariance Property

To study the shift invariance, we use two sample signals, one a shifted version of the other. ([0 0 0 0 0 1 1 1] and [0 0 0 0 1 1 1 1]). We find that real footprint transform is not shift invariant, but close to it. The development of complex footprints do no improve the shift invariance property of the transform. The figure 4.4 shows the shift variance properties of real and complex footprints. The numerical values associated with the two cases is listed in Table 4.1.



(a) (b)

Figure 4.4: (a) Coefficients computed by taking a real footprint transform of a signal and it's unit shifted version. (b) Coefficients computed using complex footprint transform. Visually, it suggests that the shift-invariance property does not change significantly in the two cases. Numerical values are in Table 4.1

Table 4.1: Shift variance property of Real and complex coefficients

	y = [0 0 0 0 0 1 1 1]	y = [0 0 0 0 1 1 1 1]
Real Footprint Coefficients	0	0
	0	0
	0	0
	0	1.4142
	1.3693	0
	0	0
	0	0
	0	0
Complex Footprint Coefficients	0	0
	0.1528	0.2531
	0.3314	0
	0.0926	0.8945
	0.9238	1.4142
	1.3724	0.8945
	0.8001	0
	0.1528	0.2531

CHAPTER 5

APPLICATIONS

5.1 Introduction

In this chapter, we illustrate the use of footprints in detecting singularities in noisy measurements of an underlying piecewise constant signal. We focus on two applications in bioinformatics - The first one is for detecting breakpoints in DNA copy number sequences, from array-CGH data [3]. Real Footprints have been used by [7] to develop a compact representation of the data, and Sparse Bayesian Learning [19] is then applied to infer the breakpoints in the signal. We use Complex footprints for the same application and analyze the results. The second application is in the area of molecular machine dynamics where we recover an underlying PWC signal from noisy observations of a molecular motor [20].

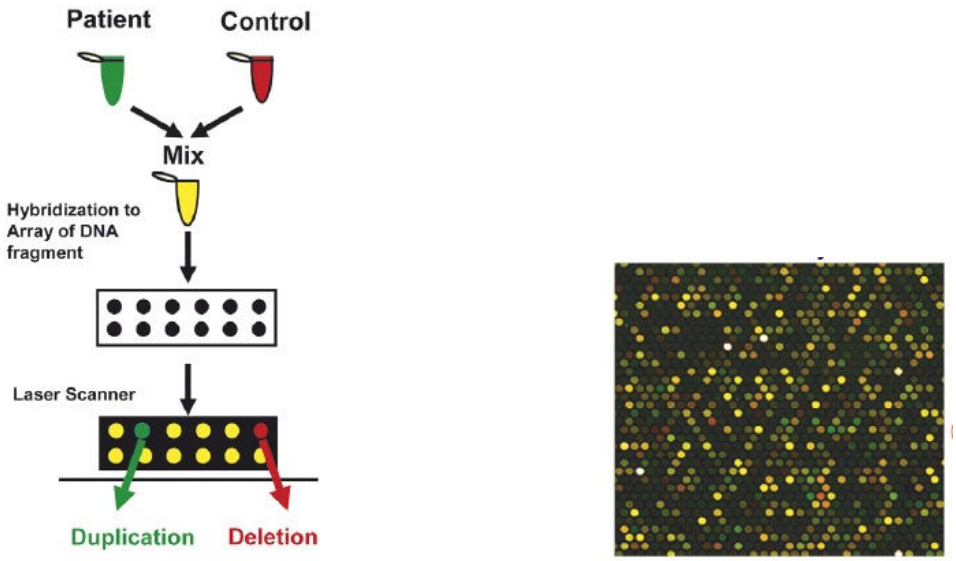
5.2 DNA copy number alterations analysis from array-CGH data

5.2.1 Introduction to aCGH

Copy Number Alterations (CNA) involves deletion or replication of chromosomal regions and are known to occur in numerous genetic disorders. Array-Comparative Genomic Hybridization (aCGH) is a technique that was developed for high resolution screening of CNA. CNA data is piecewise constant, with discontinuities at the locations of deletion or replication of chromosomes.

As explained in Figure 5.1, DNA from the sample to be tested (e.g. blood or amniotic fluid) is labeled with a green dye and an equal amount of reference DNA is labeled with red. The two samples are mixed and cohybridized to an array

containing genomic DNA targets that have been spotted on a glass slide. The resulting ratio of the fluorescence intensities is proportional to the ratio of the copy numbers of DNA sequences in the test and reference genomes. The areas on the slide that appear green indicate extra chromosomal material (duplication) in the test sample at that particular region. The slides are scanned into image files using a microarray scanner. The spot intensities are measured. Areas on the slide that appear red indicate relatively less test DNA (deletion) in the sample at that specific spot.[5] The log2 ratio of the red to the green intensities are computed for each spot, and converted into a 1-d signal. These are the log intensities that are measured, corresponding to the relative copy number in the genome, and have an underlying PWC structure.



(a) (b)

Figure 5.1: (a) Principle of acgh. (b) An output array of scanning hundreds of spots with different ratios of intensities [5]

.

Several algorithms have been proposed to detect CNA [27] [28] [29] etc. Some of them rely on a fundamental characteristic, that a genome is composed of relatively long segments, DNA sequences that have a constant number of copies present. The genomic segments can be represented by m probes mapping to a specific position on the genome having c_m copies. The copy numbers c_m can be ordered and arranged as vectors that have the following properties.

- They are *piecewise constant* with very small number of breakpoints relative to the number of probes
- They have *discrete values* i.e. copy numbers can only be 0, 1, 2, 3 ...

However, these properties cannot be directly observed in the log-intensities y_m measured with micro-arrays, due to contamination by biological and technical noise; thus it is modeled as:

$$y_m = x_m + \epsilon_m \tag{5.1}$$

where x_m represents the average log intensity, and ϵ_m is an additive zero-mean white random process. See Figure 5.2.

The fact that the copy number is piecewise constant along the genome is exploited to build a basis expansion using wavelet footprints. In [3], the PWC signal is represented using real footprint basis, and Sparse Bayesian Learning (SBL) followed by Backward Elimination (BE) are then applied to infer the discontinuities in the signal. We now extend this by representing the observed signal in a Complex Footprint basis.

The goal is to infer where the copy number alteration points are located, from noisy observed hybridization intensities. We seek to minimize the error in approximating the observed noisy signal using this new complex footprint representation.

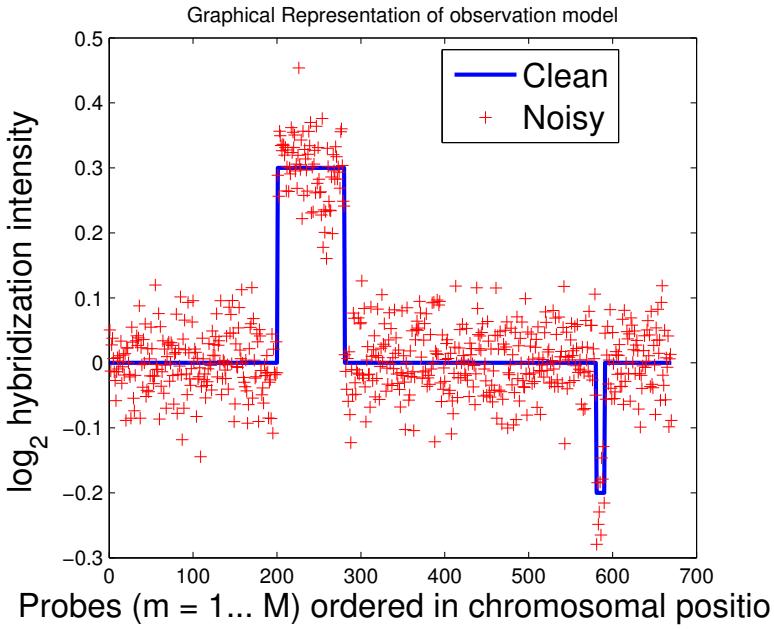


Figure 5.2: Observation model

Before we start off with the detection algorithm, we present the synthetic dataset used for this application.

5.2.2 Synthetic Data Set and Metrics Used

The datasets used to compare the algorithms rates of accuracy are those proposed by [6].

1. Determine copy number probability and the distribution of segment length. The chromosomal segments with DNA copy number $c = 0, 1, 2, 3, 4$ and 5 are generated with probability $0.01, 0.08, 0.81, 0.07, 0.02$ and 0.01. The lengths for segments are picked up randomly from the corresponding empirical length distribution given in [6].

2. Compute $log2ratio$. Each sample is a mixture of tumor cells and normal cells. A proportion of tumor cells is P_t , whose value is from a uniform distribution between 0.3 and 0.7. The $log2ratio$ is calculated by

$$log2ratio = log_2 \frac{cP_t + 2(1 - P_t)}{2} \tag{5.2}$$

where c is the assigned copy number. The expected $log2ratio$ value is then the latent true signal.

3. Add Gaussian noises. Gaussian noises with zero mean and variance σ_n^2 are added to the latent true signal. Now, we get the equally spaced CGH signal

In order to evaluate the signal reconstruction accuracy of the algorithms, the following metrics are employed:

1. Sensitivity: $\dfrac{\#\text{ discontinuities detected correctly}}{\text{Total }\#\text{ of discontinuities present}}$

2. False Discovery Rate: $\dfrac{\#\text{ discontinuities detected incorrectly}}{\text{Total }\#\text{ discontinuities detected}}$

The next section is organized as follows:

- Complete algorithm for step detection using Real Wavelet Footprints representation of data.
- Implementation of Sparse Bayesian Learning.
- Comparison of steps, equations and formulas for the implementation of the algorithm using Real and Complex representations.
- Simulation results and discussion of performance of the algorithm using Real and Complex footprints.

5.2.3 Algorithm for step detection using Real Wavelet Footprints

5.2.3.1 Formulation of the problem, and motivation for using SBL

The footprint representation can be used to facilitate estimating x from a degraded observation y generated as in the model 5.1

$$y = x + \epsilon = Fw + \epsilon \tag{5.3}$$

where x has been replaced by its footprint representation, Fw. Since the number of copy number changes (K) is very small compared to the number of probes (M), $x = Fw$ has a sparse representation in the footprint basis, while the noise ϵ is not sparse in this representation. Under this scenario, the problem is formulated as that of finding $\hat{x} = F\hat{w}$ that is closest to the observed y subject to having only K non-zero components of $\hat{w}$ [3].

$$\hat{w} : \min_{w} \; e\,(Fw, y) \;\; s.t. \;\; s(w) \;=\; K \tag{5.4}$$

Different measures of *closeness e(.)* and *sparseness s(.)* can be used. For closeness, we use the least squares error measure since it is the most widely used for approximation and will facilitate comparison among algorithms, although it may be sensitive to outliers.For measuring sparseness we are especially interested in the l_0 norm (i.e. the number of $w_m \neq 0$), which best models the biological property that $K << M$. Then, the cost function with these measures can be rewritten as follows

$$\hat{w} : \arg\min_{w} \|y - Fw\|_2 + \|w\|_0 \tag{5.5}$$

where the l_p norm and the l_0 pseudo-norm are defined as:

$$\|w\|_p = \sum_{m=1}^{M} |w_m|^2 \qquad \|w\|_{p\to 0} = \sum_{m=1}^{M} I(w_m \neq 0) \tag{5.6}$$

If we replace the l_0 by l_1 norm, then we can obtain the standard Basis Pursuit cost function, which finds the weights vector w satisfying $x = Fw$ with minimum l_1 norm. The l_1 norm is often used because convex optimization or linear programming can be used to solve the problem. In such methods, we can guarantee convergence to the global minimum of the cost function. However, the global minimum may not coincide with the sparsest solution. If the *coherence* - represented as $C = max \langle f_k, f_j \rangle \quad k \neq j$ is small, then minimizing for l_1 is equivalent to minimizing for l_0. However, the performance of these methods is severely limited because the coherence of the footprint basis vectors approaches 1. That is, footprints that correspond to two adjacent discontinuities are highly collinear. Other methods such as FOCUSS [30] are able to compute a solution with minimum l_0 norm. However, there are convergence errors that are associated with this solution. [19] Therefore, when the basis F is highly coherent, as in our case, these techniques lead to sub-optimal performance and a new approach is needed.

5.2.3.2 Sparse Bayesian Learning

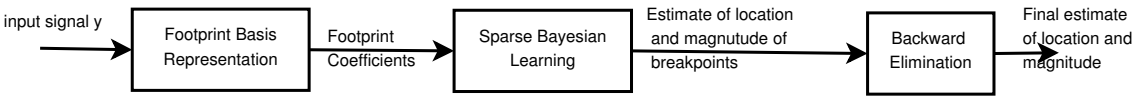


Figure 5.3: The main processing steps in SBL algorithm

Given a data set of input-output pairs $\{x_n, y_n\}_{n=1}^{N}$, we assume the data is in the presence of additive Gaussian noise and a Gaussian likelihood model is used [31]:

$$p(y|w, \sigma^2) = (2\pi\sigma^2)^{-N/2} exp\left\{-\frac{1}{2\sigma^2}\|y - Fw\|^2\right\} \tag{5.7}$$

where σ^2 is the noise variance of the input signal.

With as many parameters in the model as input data, we would expect the ML estimation of w to lead to severe over-fitting. To avoid this, a common approach is to impose some additional constraint on the paramters, for example, through the addition of a penalty term to the likelihood or error function. Here, an explicit *prior* probability distribution is defined over the parameters. A popular choice is a zero-mean Gaussian prior distribution over w.

$$p(w|\alpha) = \prod_{m=1}^{M-1} N(w_m|0, \alpha_m^{-1}) \tag{5.8}$$

with α a vector of N *hyperparameters*. There is an individual hyperparameter associated independently with each weight.

To complete the specification of the *hierarchial* prior, the hyperpriors α and noise variance σ^2 are defined. Gamma distributions are used for defining α.

$$p(\alpha) = \prod_{m=1}^{M-1} \Gamma(\alpha_m|a, b) \tag{5.9}$$

For fixed values of the hyperparameters governing the prior, the posterior density of the weights is Guassian [31].

$$p(w|y, \alpha, \sigma^2) = N(w|\mu, \Sigma) \tag{5.10}$$

with

$$\Sigma = (\sigma^{-2} F^{'} F + diag(\alpha))^{-1} \qquad \mu = \sigma^{-2} \Sigma F^{'} y \tag{5.11}$$

Once we have these values, we choose the weights satisfying $w = \mu$.

Thus, we need to estimate α and σ^2. σ^2 is estimated from the data as:

$$\hat{\sigma^2} = \frac{1}{2M} \sum_{m=1}^{M} (y_m - y_{m-1})^2 \tag{5.12}$$

To find α, the Expectation Maximization (EM) algorithm [32] is used. The EM algorithm proceeds by treating the weights w as *hidden variables* and then maximiz-

ing $E_{w|y,\alpha,\sigma}[p(y,w;\alpha)]$ where $p(y,w|\alpha) = p(y|w)p(w|\alpha)$ represents the likelihood of complete data {w,y}. We have for the l-th iteration, [19]

$$Estep : E_{w|y,\alpha^{(l)},\sigma^{-2}}(w_m^2) = \Sigma_{mm} + \mu_m^2 \tag{5.13}$$

$$Mstep : \hat{\alpha}_m^{(l+1)} = \frac{1+2a}{\Sigma_{mm} + \mu_m^2 + 2b} \tag{5.14}$$

Upon convergence, we find that several α's are 0, forcing the associated weights to zero while the non-zero weights are free to take any value - which matches well our underlying biological knowledge for copy number changes.

Thus, the model contains several hyperparameters, and α and σ parameters are estimated from the data while b is set to zero (uninformative prior). a is a tradeoff between speed of convergence and sparsity, and can be adjusted. Thus, sparseness is adjusted by the α parameter.

An overview of the main steps in the algorithm is presented in the Figure 5.4: The output of the SBL algorithm is a set of $(location, magnitude)$ pairs corresponding to the detected breakpoints, of the input signal. However, not all breakpoints found by SBL have the same statistical significance since noise may make areas without any underlying alteration appear similar to those areas corresponding to actual alterations. Some breakpoints mark the separation between two long segments (i.e. such that each segment includes many probes) and are such that the difference between the estimated amplitudes of the two segments is large. Such breakpoints are more likely to correspond to true underlying changes in copy number, and therefore will have a higher statistical score

A Backward Elimination (BE) strategy is used, in which we recursively eliminate the breakpoint with lowest statistical evidence. The BE procedure can be stopped when all the remaining breakpoints have a score higher than a specified

threshold, the BE critical value. More details on Backward Elimination are available at [3].

5.2.4 Implementation of SBL

Given the special structure of the footprint basis matrix F, (i.e. an invertible matrix who's columns are piecewise constant vectors) [3] has shown that SBL computation can be optimized for the PWC representation. In particular, for the real footprint basis matrix F, a closed form solution to a inverse of $(F'F)$, a (large) matrix, has been determined. This can make the processing very efficient. The matrix $H = G^{-1} = (F'F)^{-1}$ a symmetric tridiagonal matrix with main diagonal

$$h_0(j) = \frac{(M - i_j)i_j}{M} \frac{(i_{j+1} - i_{j-1})}{(i_{j+1} - i_j)(i_j - i_{j-1})} \tag{5.15}$$

and upper/lower diagonal elements are

$$h_1(j) = \frac{\sqrt{(M - i_j)i_j(M - i_{j+1})i_{j+1}}}{M(i_{j+1} - i_j)} \tag{5.16}$$

This structure can be used to efficiently compute μ_m and Σ_{mm} for each EM step in the algorithm.

Also, removing the mean from the signal allows us to remove the f_0 from F for all computation in the algorithm. Hence, the footprint basis matrix F is no longer an $M \times M$ square matrix but has dimensions of $M \times M - 1$.

Below, we present the steps followed for implementing the SBL algorithm.

Inputs: y, α, σ^2

1. $\bar{y} \leftarrow \frac{1}{M} \sum_{m=1}^{M} y_m$
2. $y \leftarrow y - \bar{y}$;
3. (h_0 and h_1) from $5.15 and 5.16$
4. $w_0 = F^{-1}y$

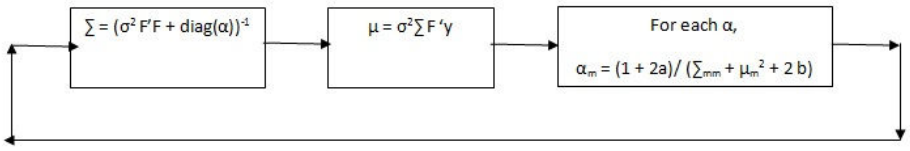


Figure 5.4: Main steps in SBL algorithm

5. loop: $T = \sigma^2(F^TF)^{-1}\Lambda + I$
6. $Tw = w_0$; Solve for w using LU decomposition method
7. Compute $diag(\Sigma)$ where $\Sigma = \sigma^2 T^{-1}(F^TF)^{-1}$
8. for $i = 1, ..., I$

   $\alpha_j = \frac{1+2a}{w_j^2+\Sigma_{jj}}$

   end for
9. until w has converged $\|w_{old} - w_{new}\| \leq \epsilon$

Outputs: w, I i.e. estimated breakpoint locations and magnitudes of the underlying PWC signal.

For computational efficiency, rather than compute the mean and covariance directly, as in 5.11 a variable is introduced $T = \sigma^2(F^TF)^{-1}\Lambda + I$. By this, the inverse of any matrix does not have to be computed during the processing of the algorithm - the only inverse that is required is $(F^TF)^{-1}$, which has a known closed form solution 5.15 and 5.16. Calculation of inverse of a matrix is a computationally heavy operation and by introducing this new variable T, it is possible to avoid this operation.

As discussed, the first column of the F matrix can be deleted if we remove the mean from the input signal. Hence, the basis matrix now has size $M \times M - 1$. Since it is not square, it is not invertible. However, the pseudo-inverse exists, and F^{-1} is bidiagonal. This makes calculation of the initial estimate w_0 quite straightforward.

Table 5.1: Shows the modifications involved while using complex footprints in the SBL algorithm

	Real footprint representation	Complex footprint representation
a)	$x = Fw$	$x = \frac{1}{2}\begin{bmatrix} F & -HF \end{bmatrix}\begin{bmatrix} w_R \\ w_I \end{bmatrix}$
b)	$w_0 = (F^TF)^{-1}F^Tx$	$w_{R_0} = (F^TF)^{-1}F^Tx$ $w_{I_0} = H(F^TF)^{-1}F^Tx$
c)	*loop*: $T = (\sigma^2(F^TF)^{-1}\Lambda + I)$	*loop*: $T_r = (\sigma^2(F^TF)^{-1}\Lambda + I) = T_i$
d)	$Tw = w_0; \quad w = T^{-1}w_0$	$T_r w_R = w_{R_0}; w_R = T_r^{-1}w_{R_0}$ $T_i w_I = w_{I_0} \quad w_I = T_i^{-1}w_{I_0}$
e)	$\Sigma = \sigma^2 T^{-1}(F^TF)^{-1}$	$\Sigma_r = \sigma^2 T_r^{-1}(F^TF)^{-1} \quad = \Sigma_i$
f)	$forj = 1, ..., I$ $\alpha_j = \frac{1+2a}{w_j^2+\Sigma_{jj}}$ *end*	$forj = 1, ..., I$ $\alpha_j = \frac{1+2a}{\frac{1}{2}(w_{R_j}^2+w_{I_j}^2)+\Sigma_{jj}}$ *end*
g)	until w converges	until w converges

5.3 Comparison of SBL algorithm using Real and Complex footprints

We now discuss the steps in the implementations, as presented in Table 5.1. We consider an input signal of length M, and a basis matrix of size $M \times M-1$ We compute the initial estimates of real and imaginary components. We use these initial estimates in updating the coefficients in each iteration of SBL algorithm. The EM algorithm calculates an updated vector of hyperparameters α in each iteration. While computing the hyperparameters, we now use both real and imaginary parts, as in step f) in the table. Hence at the output of the algorithm, we have M weights that have been calculated by using information from real and imaginary components.

At this point, we state the reason why we choose this method of computing real and imaginary coefficients individually, rather than the more straightforward method of replacing the real footprint basis F with the complex basis $[F - HF]$ in step a) in the table. Let $\tilde{F} = [F - HF]$. We observe that $(\tilde{F}^T\tilde{F})^{-1}$ is not invertible (the matrix is of half-rank), and nor does it have a pseudo-inverse. Hence, any solution which

uses $\tilde{F}$ explicitly will not be computationally feasible. This is why we are restricted to use methods that do not involve explicit use of the complex basis, but achieve the same computations as the complex basis.

5.4 Simulation Results and Discussion

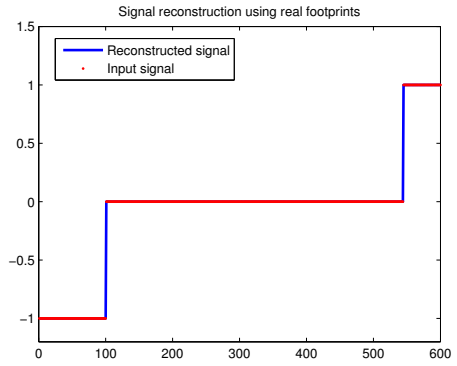
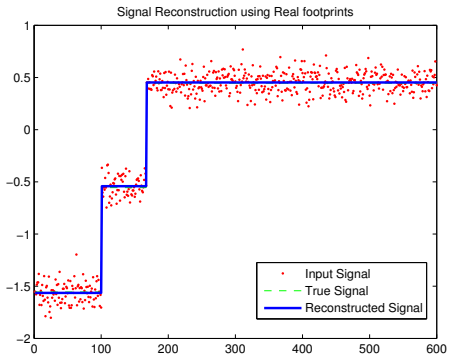
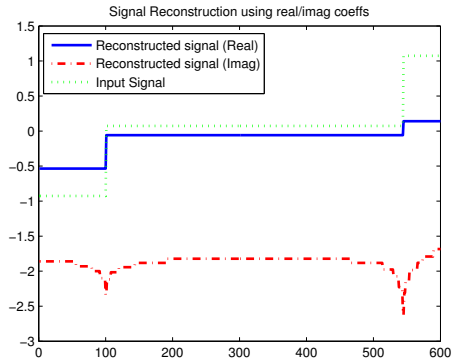
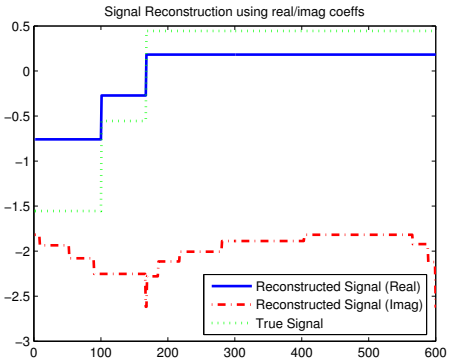
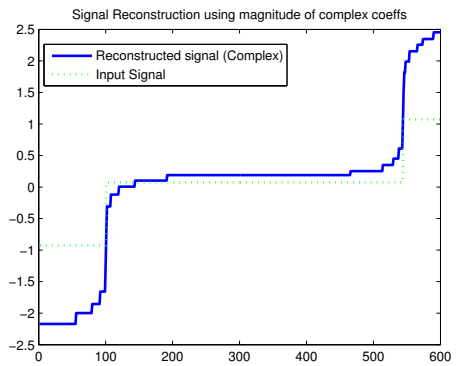
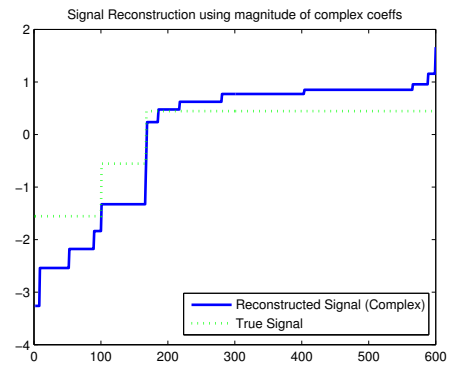
We first consider a simple test signal:

The input signal is a length-600 signal that has a few breakpoints at random. When the input signal is clean, as in the figures on the left column, we observe that reconstruction using real footprints gives a signal that is very close to the original signal. Upon using the imaginary coefficients alone, the discontinuities in the signal are located with good accuracy. However, using the magnitude of the complex footprint coefficients, the reconstruction does not improve from that obtained by using real footprints alone.

When we add some noise to the input signal (we add Gaussian noise with standard deviation 0.1), we observe that reconstruction using real footprints is still very close to the original signal. The imaginary coefficients do not locate the discontinuities well, and reconstructed signal using complex footprints is poor compared with using real footprints alone.

To obtain numerical values, we constructed a signal as shown above, and added noise of different variances to the signal. We performed signal reconstruction using both real and complex footprints. We then analyzed their performances using the metrics - Sensitivity and False Discovery Rate (which have been defined earlier in this chapter). The values were averaged over 100 iterations. Table 5.3 shows the

Table 5.2: Comparison of Real and Complex footprints - using Clean and Noisy input

Clean Input	Input with $\sigma = 0.1$
	
	
	

values that were obtained.

Table 5.3: Comparison of Sensitivity and FDR for signals reconstructed using real and complex footprints

σ	Sens. (real)	FDR (real)	Sens. (complex)	FDR (complex)
0.001	1	0	1	0.3333
0.01	1	0	1	0.3678
0.1	1	0.02	0.955	0.4
0.2	0.9533	0.4599	0.8717	0.5565
0.4	0.8117	0.7124	0.735	0.7542
0.6	0.7483	0.77	0.6683	0.8038
0.8	0.7333	0.7861	0.645	0.8177
0.9	0.7067	0.7987	0.6317	0.8244
1.0	0.7083	0.8	0.6317	0.8258

We also conducted the same experiment using the synthetic array-CGH data described in section 5.5.2. One such signal is shown in Figure 5.5. We used signal length of 256. The input signal is contaminated with noise of standard deviation in the range of $0.1 - 0.2$ The results are as shown in Table 5.4. These observations

Table 5.4: Performance metrics of footprint based method applied on DNA copy number measurements, proposed by [6]

Noise Std Dev	Real Footprints		Complex footprints	
	Sensitivity	FDR	Sensitivity	FDR
0.01 - 0.1	0.8533	0.4020	0.667	0.691
0.05 - 0.15	0.8420	0.3846	0.667	0.715
0.1 - 0.2	0.7871	0.395	0.402	0.825

tell us that using complex footprints for this application does not improve the results in terms of sensitivity and false detection rate.

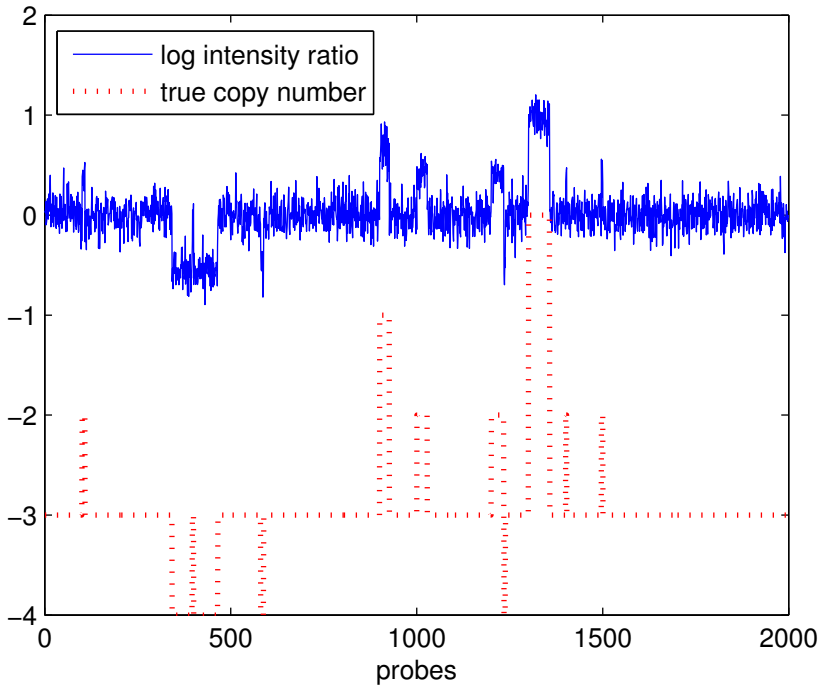


Figure 5.5: Synthetic data as suggested in [6]

5.4.1 Discussion

We believed that by using a complex footprint representation rather than real footprints, the complex coefficients would improve the edge detection performance. We used the magnitude of the complex coefficients for our analysis. We expected that the magnitudes would represent any discontinuity in the signal better than a real footprint coefficient could. However, we observed that detection using the complex coefficients is never better than detection by using only real components. With a noisy input, the algorithm using complex coefficients gives a poorer reconstruction than the algorithm using real coefficients only.

We believe this is due to the approximation of the Hilbert transform used while forming the complex footprint basis. The ideal Hilbert transformer is infinite length,

whereas we have used a finite length approximation, which could have introduced errors in the computations.

Also, we observed in the previous chapter that the magnitudes of the complex footprints coefficients did not improve the shift invariance property compared to using real footprint coefficients. We conclude that real footprints are more effective than complex footprints in signal reconstruction of such PWC signals. Hence, for our next application, we compare the performance of signal reconstruction using real footprints with other existing methods.

5.5 High-Throughput Analysis of Molecular Machine Dynamics

5.5.1 Molecular Machines

Nanotechnology allows one to construct useful machines on a molecular scale. Nature has evolved many robust molecular machines such as pumps, tugs, copiers and motors and understanding the function of these machines is key to the possibility of designing of interacting artificial molecular devices. These motors that convert electrochemical energy to linear or rotary kinetic energy, do so in a series of rapid, nano-scale step-like motions.This step like motion has been observed using advanced experimental procedures [20].

These experiments produce large volumes of time series data with sampling rates that often exceed 100kHz but the step-like motions of the molecular components is obscured by noise. This noise must be removed to extract the underlying molecular dynamics- in this case, a PWC signal.

This noise removal is a challenging signal processing problem because the signal to noise ratio is low ad the signal itself is a step-and-impulse like i.e. it is *highly discontinuous*

5.5.2 Step-like synthetic data

Typical noisy step-like time series from experiments on molecular machines is shown in the Figure 5.6. This is a time series of piecewise constant segments with superimposed, additive, i.i.d. Gaussian noise. The steps between the constant segments typically occur at time intervals that are exponentially distributed, and the steps may be upwards or downwards. This is very similar to a Poisson process, except that the event count can go down as well as up. The observed discrete time signal is defined here as $x_n = \mu_n + \epsilon_n$ where μ_n is a piecewise constant step signal with steps of the same height and ϵ_n is i.i.d. Gaussian noise of variance σ^2 [20].

5.5.3 Existing Methods

The signal that we are dealing with is highly discontinuous. The support in the Fourier domain of the signal and the noise overlap considerably. Separation in the Fourier domain using classical linear filtering is not feasible. Special techniques are thus required that can cope with both high noise levels and discontinuous signals. Previous methods use a step-filtering approach to this problem include Median Filtering [10] and Global Filtering using L_1-*regularized fused LASSO* [18], discussed earlier in this report.

5.5.4 Simulation and Results

We illustrate the use of footprints in detecting singularities in a Molecular Machine motor data. We used both real and complex footprints and observed the results. As we have seen in the last section, signal reconstruction using Real Footprints give a much better performance than using complex footprints for noisy inputs. Also, we find that using real footprints to reconstruct the signal outperforms existing methods for obtaining a clean signal from noisy Molecular Machines data.

Table 5.5: Accuracy of the three algorithms in recovering a PWC signal for a range of noise variances

Noise std dev	MAE median filter	MAE *fused*-LASSO	MAE using footprints
0.1	0.0231	0.013	**0.0028**
0.3	0.0692	0.0187	**0.0083**
0.5	0.1123	0.0263	**0.0138**
0.6	0.1358	0.0336	**0.0188**
0.7	0.1553	0.0393	**0.0215**
0.9	0.1988	0.0521	**0.0295**
1.1	0.24	0.071	**0.036**

Below we present a comparison of the *Running Median Filtering*, *Bayesian L_1-regularized LASSO* and *Real Footprint* methods in recovering underlying step dynamics μ_n, for a unit step height, for a range of noise variances. We create a signal of length 5000 as described earlier, and take the average over 100 iterations.

The metric used to measure the accuracy of the algorithms was the Mean Abolute Error (MAE).

$$MAE = \sum_{n=1}^{N} |\mu_n - m_n|/N \tag{5.17}$$

A low value of MAE indicates a good signal approximation.

Figure 5.6 shows the reconstructed (or filtered) signals for synthetic data obscured by noise of standard deviation 0.6. The median filter operates with a window size $W = 20$, the regularization paramter for the fused-LASSO method $\lambda = 10$. These values were chosen such that the MAE in recovering the known steps is minimized.

Numerical values obtained are shown in the Table 5.5.

A graphical comparison of the three methods is shown in Figure 5.7. The MAE achieved using footprint based method is significantly lower than other methods.

We also run the algorithms on real data from a molecular motor - We use WS8N wild-type *Rohdobacter sphaeroides* cells. The flagella (tails) are removed and 0.83 micron beads are attached to the flagellar hooks [20]. The beads are then laser

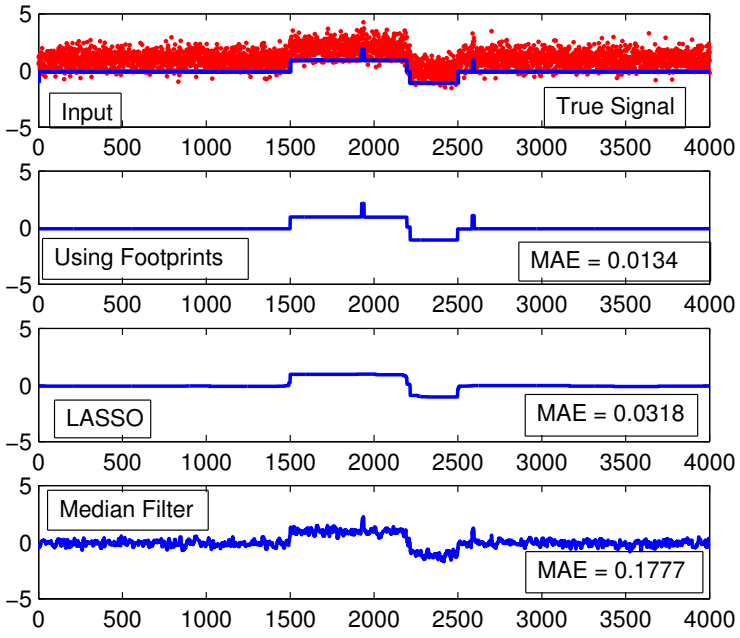


Figure 5.6: Mean absolute error calculated by using Footprints, *fused*-LASSO, and by using Median Filtering respectively

illuminated, and the speed of rotation of the flagellar motor against time is recorded. We display typical measurement time series, and those obtained after applying the various algorithms. As can be seen from Figure 5.8, the reconstruction of the signal using footprints method gives a much cleaner output compared to other algorithms. This leads us to conclude that the use of footprints for reconstructing molecular machine PWC data, outperforms existing methods.

5.5.5 Discussion

Although wavelet basis and especially Haar basis have been suggested in literature [15] for removing noise from piecewise constant signals, there is an argument against using wavelet based algorithms as discussed in [13]. Removing noise typically requires the removal of small-scale detail coefficients. The result of removing these

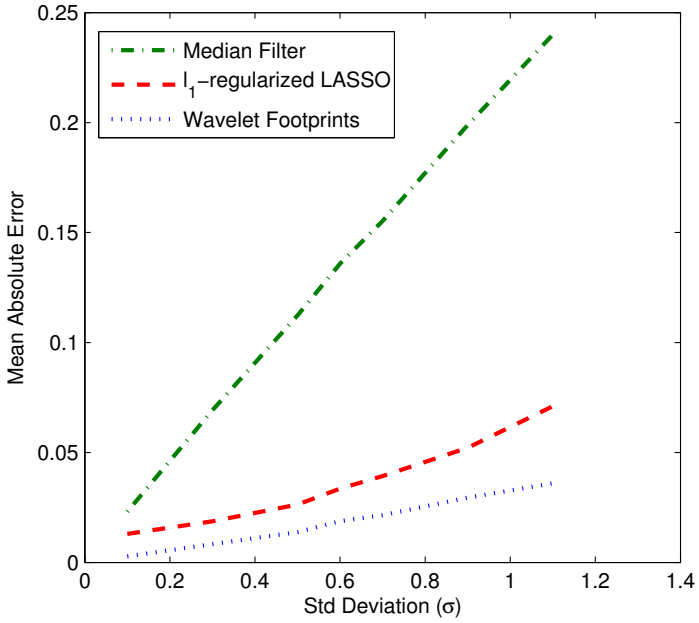


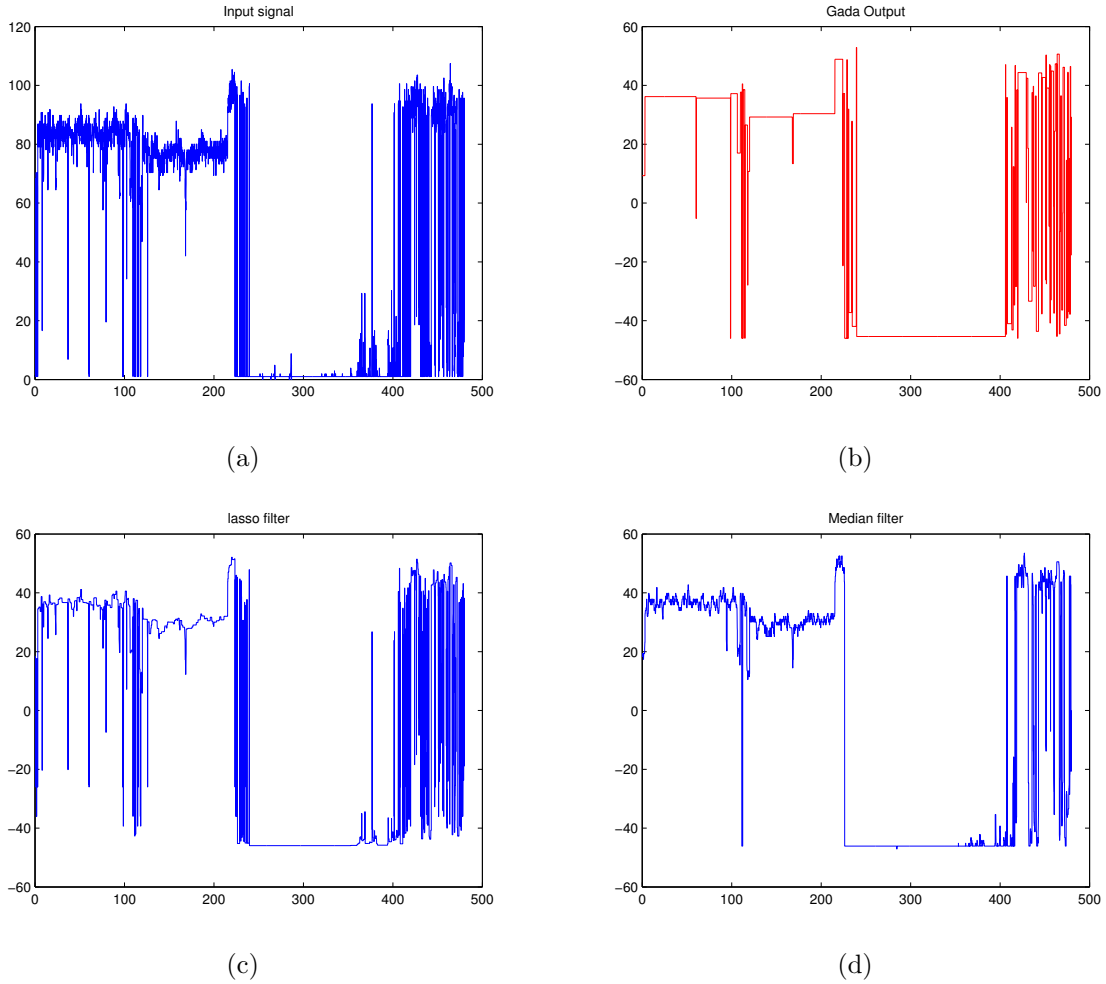
Figure 5.7: MAE comparison using the 3 different methods

coefficients is that the time-localization of the remaining large scale coefficients is poor, and jumps in the signal cannot be accurately located - they are not exactly aligned to the locations of the jumps in the reconstructed signal.

However, while using wavelet footprints, the issue of ignoring information from certain scales does not arise, because by definition, a wavelet footprint gathers information from wavelet coefficients at all scales. Hence, it is possible to accurately determine the locations of jumps in the highly discontinuous signal that is used in this application.

As can be seen from Figure 5.6, the footprints method outperforms the *Bayesian L_1-regularised LASSO* by a factor of nearly 2. That is, the MAE using footprints is nearly half that of the next-best algorithm at all the tested noise variances. This algorithm produces smooth results with sharp edges. However, median filter produces noisy results.

The performance of the algorithms on the synthetic time-series with unit step height across a range of noise variances is shown in the table. The median filter has the worst overall performance, as the error can reach as much as 20% of the step height. By contrast, the Bayesian filter can achieve errors of less than 10% of the step height, while the footprint-based method achieves errors that are even lesser, in the order of 5% of the step height.



(a) (b)

(c) (d)

Figure 5.8: Molecular motor data: (a) Input signal (b)Footprint based reconstruction, (c) Using LASSO filter with $\lambda = 10$, (d) Using median filter with window $W = 10$

CHAPTER 6

CONCLUDING REMARKS

In this work, we have studied different techniques for step detection in a piecewise constant signal. We used 3 methods, namely step detection using Phase Congruency, real footprints, and using complex footprints. We studied the performance of these methods using Sensitivity and False Discovery Rate. We used these methods in 2 different bioinformatics signal processing applications.

First we used the concept of Phase Congruency to detect discontinuities in PWC signals. For synthetically created signal having low noise levels (noise standard deviation ≤ 0.1), we found that this measure was able to detect edges accurately. In particular, Phase Congruency using Inter Scale wavelet coefficients was empirically found to be more effective than computing using Relative Phase or Derotated Phase coefficients. Using Phase Congruency for step detection would be beneficial in applications where the signal is not heavily contaminated by noise.

We then developed the concept of complex footprints. We referred to the work of [2] who has developed a transform called wavelet footprints, and extended it so that coefficients take complex values. However, all computation involved continues to be real. We represented PWC signals using complex footprints. We studied the shift invariance property of the two transforms. We found that the real footprint representation is nearly shift-invariant, but not exactly shift invariant. We determined that the complex representation does not improve the shift-invariance property from the real case.

We then applied these step detection methods to two applications which involved recovery of a PWC signal from its noisy observations - both in the area of bioinformatics. The first application involved step detection in array-CGH data. [3] has developed an algorithm based on real footprint transform, and achieves highly accurate results. Replacing the real footprint coefficients by the magnitude of complex coefficients did not improve on these results. For noisy inputs, the result was much poorer when using complex footprints. We believe this is due to the finite-length approximation of the infinite length ideal Hilbert transform, in our computation.

Since we found that step detection using real footprints work better then using complex footprints, we then applied the algorithm using real footprints in the application of molecular machines data. The accuracy metric used was Mean Absolute Error. We found that this method outperforms the existing algorithms that are used, by a factor of at least two.

Future work on this topic could focus on ways to improve the Hilbert transform calculation, and also in finding suitable applications for complex footprints.

We conclude that phase congruency can be used when detecting PWC signals from an input contaminated with low noise. For inputs corrupted with high levels of noise, the detection algorithm using real wavelet footprints performs better than that using complex footprints, or phase congruency.

REFERENCES

[1] P. Kovesi, "Image features from phase congruency," *Videre: Journal of Computer Vision Research*, vol. 1, no. 3, pp. 2–26, 1999.

[2] P. L. Dragotti and M. Vetterli, "Wavelet footprints - theory, algorithms, and applications," *IEEE Trans. Signal Processing*, vol. 51, no. 5, pp. 1306–1323, 2003.

[3] R. Pique-Regi *et al.*, "Sparse representation and bayesian detection of genome copy number alterations from microarray data," *Bioinformatics*, vol. 24, no. 3, pp. 309–318, 2008.

[4] M. Morrone and R. Owens, "Feature detection from local energy," *Pattern Recognition Letters*, vol. 6, pp. 303–313, 1987.

[5] M. Shinawi and S. W. Cheung, "The array cgh and its clinical applications," *Drug Discovery Today*, vol. 13, no. 17, pp. 760–770, 2008.

[6] H. Willenbrock and J. Fridlyand, "A comparison study: applying segmentation to array cgh data for downstream analysis," *Bioinformatics*, vol. 21, pp. 4084–4091, 2005.

[7] R. Pique-Regi *et al.*, "Wavelet footprints and sparse bayesian learning for dna copy number change analysis," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, Hawaii, USA, Apr. 2007, pp. 353–356.

[8] R. B. Ivan Selesnick and N. Kingsbury, "The dual tree complex wavelet transform," *IEEE Signal Processing Magazine*, pp. 123–151, 2005.

[9] M. Barkat, *Signal Detection and Estimation*. Boston, MA: Artech House, 1991.

[10] S. . SK Mitra, *Digital Signal Processing: A computer based approach.* New Delhi, India: Tata-McGraw Hill, 2008.

[11] J. Canny, "A computational approach to edge detection," *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[12] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Royal Society A, Math. Phys. Sci.*, vol. B 207, pp. 187–217, 1980.

[13] M. Little and N. Jones, "Generalized methods and solvers for noise removal from piecewise constant signals," *Proceedings of the Royal Society A*, in review.

[14] S. Mallat and W. Hwang, "Singularity detection and processing with wavelets," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 617–643, 1992.

[15] C. Cattani, "Haar wavelet-based technique for sharp jumps classification," *Mathematical and Computer Modelling*, vol. 39, pp. 255–278, 2004.

[16] L. Zhang and P. Bao, "Edge detection by scale multiplication in wavelet domain," *Pattern Recognition Letters*, vol. 23, pp. 1771–1784, 2002.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, UK: Cambridge University Press, 2004.

[18] R. Tibshirani *et al.*, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society Series B-Statistical Methodology*, vol. 67, pp. 91–108, 2005.

[19] D. Wipf and B. Rao, "Sparse bayesian learning for basis selection," *IEEE Trans. Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[20] M. A. Little and N. S. Jones, "Sparse bayesian step-filtering for high-throughput analysis of molecular machine dynamics," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, Dallas, USA, Mar. 2010, pp. 4162–4165.

[21] D. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of The Optical Society of America*, vol. 4, no. 12, pp. 2379–2394, 1987.

[22] S. Venkatesh and R. Owens, "On the classification of image features," *Pattern Recognition Letters*, vol. 11, pp. 339–349, 1990.

[23] S. O. A. P.N. Vo and T. T. Nguyen, "Using phase and magnitude information of the complex directional filter bank for texture image retrieval," in *Proc. IEEE ICIP*, San Antonio, TX, 2007, pp. IV–61–IV–64.

[24] M. Miller and N. Kingsbury, "Image modeling using interscale phase properties of complex wavelet coefficients," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 17, no. 9, pp. 1491–1499, 2008.

[25] S. Mallat, *A Wavelet Tour of Signal Processing - The Sparse Way.* Burlington, MA: Elsevier, 2009.

[26] J. Lawrence Marple, "Computing the discrete-time analytic signal via fft," *IEEE Trans. Signal Processing*, vol. 47, no. 9, pp. 2600–2603, 1999.

[27] L. Hsu *et al.*, "Denoising array-based comparative genomic hybridization data using wavelets," *Biostatistics*, vol. 6, pp. 211–226, 2005.

[28] S. Venkatraman and B. Olshen, "A faster circular binary segmentation algorithm for the analysis of array cgh data," *Bioinformatics*, vol. 23, pp. 657–663, 2007.

[29] N. Nguyen *et al.*, "A new smoothing model for analyzing array cgh data," in *Proc. IEEE International Conference on Bioinformatics and Bioengineering*, Boston, USA, Oct. 2007, pp. 1027–1034.

[30] Gorodnitsky and B. Rao, "Sparse signal reconstruction from limited data using focuss: a reweighted minimum norm algorithm," *IEEE Trans. Signal Processing*, vol. 45, pp. 600–616, Mar 1997.

[31] M. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, pp. 211–244, 2001.

[32] McLachlan and Krishnan, *The EM Algorithm and Applications*. New York, NY: Wiley, 2008.

BIOGRAPHICAL STATEMENT

Srikanteswara (Eshwar) Sachidananda was born in Bangalore, India in 1986. He completed his Bachelors in Electronics and Communication Engineering from Visveswaraya Technological University, Bangalore, India in 2008. He is currently pursuing a Masters degree at the Electrical Engineering department at UT Arlington. He has worked as a DSP Developer with Indsp Audio Tech, a start-up in the area of audio-DSP for music applications, in Bangalore between 2008-2009, and was an intern with the Wireless Connectivity Group at Qualcomm Inc during the Summer 2010.

His interests are in using signal processing concepts to new applications like bioinformatics.