Universitat Politècnica de Catalunya (UPC)
Departament d'Enginyeria Telemàtica (EnTel)
Escola Politècnica Superior de Castelldefels (EPSC)

PhD Program on Telematics Engineering

Contributions to the Wavelet-based Characterization of Network Traffic

David Rincón Rivera

Advisor: Prof. Sebastià Sallent Ribes

Barcelona, July 2007

Contents

List of Figures

List of Tables

Abstract

Network traffic exhibits fractal or scaling properties such as self-similarity, long-range dependence (LRD) and multifractality. This has implications for network performance, including higher buffer losses than those predicted by classical teletraffic models. Though real traffic is expected to change its behavior over time, few studies have addressed the case of time-varying fractal parameters, and almost all of them rely on constant-length windows. Identifying the boundaries of segments with homogeneous scaling parameters will be useful for the real-time application of algorithms that exploit the fractal properties of network traffic (i.e. connection admission control, or effective bandwidth estimators). This thesis deals with the estimation of the fractal parameters of network traffic when such parameters are not stationary and the development of algorithms capable of segmenting traffic in regions whose scaling parameters remain (approximately) constant.

The Discrete Wavelet Transform (DWT)-based LogScale Diagram (LD) is nowadays considered the best and most efficient estimator of fractal parameters due to its multiscale analysis of variance. On the other hand, DWT's resolution coarsens at the higher time scales, and it lacks of shift invariance, making the results dependent on the time shifts of the original signal. This is undesirable for our purposes, and that is why we turned our attention towards alternative, redundant wavelet-based transforms that avoid DWT's limitations. The Wavelet Packet (WPT) is a frequency-redundant transform which allows total flexibility when choosing the basis, either orthogonal, undercomplete or overcomplete. Our first contribution has been the development of a WPT-based estimator called the *Wavelet Packet variance map*, which is a first step towards a time-varying basis analysis in which the signal is studied with a subband decomposition scheme that adapts to the spectral properties of the time series under study. The (static) WPT-based analysis applied to real traffic traces has shown clear evidence of the non-stationarity of their scaling parameters.

We then turned our attention towards time-redundant transforms such as the Maximal Overlap DWT (MODWT), which can be considered the counterpart of the WPT in the time domain. The MODWT has the nice properties of scale-constant resolution and shift invariance, though at the price of introducing a non-negligible scale-dependent correlation. Given this drawback, we explored the signal processing literature looking for other transforms that could provide a trade-off between correlation and time resolution, finding the Dual Tree WT (DTWT), which provides near-time-shift invariance and a moderate correlation, while effectively doubling the sampling rate of the classical DWT.

With these tools we developed our second contribution, a trace segmenting algorithm composed of three blocks: a wavelet transform, a variance change detector, and a clustering and alignment detection method based on the Hough Transform. If the scaling parameters of traffic change at a given moment, so does the complete variance structure at the whole range of scales. Therefore, if a change point detection algorithm is applied at the output of the wavelet transform monitors the variance at every scale, a change of the scaling parameters can be detected as a simultaneous, time-aligned variance change across scales. A final step of clustering and alignment of the changes would provide the boundaries of the constant-scaling segment. Our approach is more accurate than the constant length window-based methods, since we can localize the variance-transition points to any position and scale.

The thesis presents the results obtained by combining the DWT, MODWT or DTWT wavelet transforms together with two variance change detection algorithms (the Iterated Cumulative Sum of Squares and the Schwarz Information Criterion) and the alignment method. We study the performance of the algorithms in presence of mean, variance and Hurst parameter changes, including assessments of the empirical power of the tests. A progressive (in the sense of near-real-time) version of our algorithm is also discussed. The methods are applied to both synthetic traces (for validation and statistical characterization) and real traffic traces, which have been found to have higher variations of their scaling properties than those usually reported in the literature.

Resumen

El tráfico transportado por las redes de ordenadores presenta características fractales como la autosimilitud, la dependencia a largo plazo o la multifractalidad. Este hecho tiene consecuencias en el rendimiento de las redes, provocando desbordamientos de colas y pérdidas mucho mayores que las previstas por los modelos clásicos de teletráfico. Pese a la alta variabilidad del tráfico, hay pocos estudios sobre la no estacionariedad de los parámetros fractales, y casi todos los estudios disponibles usan ventanas temporales de longitud fija. La correcta identificación de los instantes en que el tráfico cambia sus características sería de gran utilidad a los algoritmos que explotan las propiedades fractales del tráfico, como por ejemplo ciertos controles de admisión de conexiones o algunos estimadores de ancho de banda efectivo. Esta tesis se centra en la estimación de dichos parámetros y la segmentación del tráfico en fragmentos homogéneos cuya fractalidad es aproximadamente constante.

El LogScale Diagram (LD), basado en la Transformada Wavelet Discreta (DWT), es considerado como el mejor y más eficiente estimador de parámetros fractales gracias a su análisis multiescala de la varianza. Sin embargo la resolución de la DWT a las escalas temporales elevadas es muy baja, y además carece de invarianza frente a desplazamientos, lo que provoca que los resultados dependan de la posición de la señal original. Esto no es lo más adecuado para nuestros objetivos, y por ello hemos considerado el uso alternativo de transformadas wavelet redundantes que nos permitan superar las limitaciones de la DWT. La transformada Wavelet Packet (WPT) es redundante en frecuencia y permite una flexibilidad total en la elección de la base de la descomposición, pudiendo ser ésta ortogonal, incompleta o sobrecompleta. Nuestra primera contribución ha consistido en el desarrollo de un estimador basado en WPT y denominado *Wavelet Packet variance map*, que constituye un primer paso hacia un análisis temporal en el que la descomposición de la señal en subbandas se adapta a las propiedades espectrales de la serie temporal. El análisis (estático) WPT aplicado a trazas reales de tráfico ha dado como resultado evidencias de la no estacionariedad de los parámetros de escalado.

A continuación centramos nuestra atención en las transformadas redundantes en tiempo como la Maximal Overlap DWT (MODWT), que puede ser considerada la homóloga de la WPT en el dominio temporal. La MODWT presenta las propiedades de ser invariante al desplazamiento y de tener una resolución constante a todas las escalas; por contra introduce un alto nivel de correlación que se incrementa con la escala. Dado este inconveniente se ha realizado una búsqueda de otras

transformadas que proporcionen un compromiso entre la correlación y la resolución temporal. La Dual Tree WT (DTWT) proporciona quasi-invarianza a desplazamiento al coste de una correlación mínima, pero ofreciendo una resolución temporal que dobla la de la DWT clásica.

A partir de estas herramientas hemos desarrollado nuestra segunda contribución, un algoritmo de segmentación compuesto de tres elementos: una transformada wavelet, un algoritmo de detección de cambios de varianza, y un algoritmo de agrupamiento y alineación basado en la transformada de Hough. Cuando los parámetros fractales cambian en un instante dado, también lo hace la estructura completa de la varianza en todo el rango de escalas temporales. Por tanto, si se aplica un detector de cambio de varianza a cada escala, podemos detectar los cambios de fractalidad mediante la alineación de cambios a lo largo de todas las escalas. Un paso final de agrupación y alineamiento de los puntos de cambio de varianza proporciona la segmentación en fragmentos con características fractales homogéneas. Nuestro método es más preciso que los basados en ventanas de longitud constante, ya que nos permite localizar los cambios de varianza en cualquier posición temporal y a cualquier escala.

Esta tesis presenta los resultados obtenidos mediante la combinación de las transformadas DWT, MODWT y DTWT, de dos métodos de detección de cambios de varianza (Iterated Cumulative Sum of Squares y Schwarz Information Criterion) y el método de alineación. Hemos estudiado el comportamiento de los algoritmos en presencia de cambios de media, varianza y parámetro de Hurst, incluyendo evaluaciones empíricas de la potencia de los tests. También se presenta una versión progresiva del algoritmo que constituye un primer paso hacia el análisis en tiempo real. Los métodos se han aplicado sobre trazas sintéticas (para evaluación y caracterización estadística) y sobre trazas reales de tráfico, en las cuales se han encontrado variaciones de los parámetros fractales mucho más pronunciadas que los publicados en otros estudios.

Acknowledgements

This thesis is the end of a long (too long!) journey, during which I have met a lot of people who, in some degree, have contributed to the finalization of my PhD. Professor Sebastià Sallent, my advisor, is of course the person to whom I am most indebted for the work presented here. He has been very supportive for the times I could not see the end.

I would like to thank two colleagues (and now friends) who I met during the EuroNGI/FGI Network of Excellence and who contributed with some of the research results described here: Flaminio Minerva, from the University of Pisa (Italy), whose Master Thesis I had the pleasure to co-advise during the months he spent in Barcelona; and Piotr Zuraniewski, from Akademia-Górniczo Hutnicza (Cracow, Poland) with whom I have shared some periods of intense work during our mutual visits to Barcelona and Cracow.

I would also like to mention some UPC students whose BsC or MsC theses was related to the analysis of network traffic and I had the pleasure of advise: Xavier Hernández, Glòria Ferrer, Sergi Martínez, Cristina Cano (twice!), Aristides Rib, Josep Canales and Iván Pérez. Thanks to all, and I wish them good luck in their careers.

My gratitude goes to the creators of several pieces of code I have used in my PhD: P. Abry and D. Veitch for his LogScale Diagram, N. Kingsbury (for his DTWT Matlab toolbox), Minh N. Do (for his Generalized Gaussian distribution parameter estimator), I. Selesnick and his students (for their Matlab DTWT and DDWT toolboxes), and the programmers of the Matlab® Wavelet Toolbox, which has been my main development platform. I would also like to thank the anonymous reviewers of my papers for their comments.

My closest friends have had to bear a lot of desperate calls and e-mails; guys, it's over! My gratitude goes to Imma, Antonio, MJ, and Lluís. Not to mention some colleagues whose companionship, comments and jokes makes daily work a bit less painful: Anna, Rafa, Lluís and all the members of the Castelldefels section of the Telematics Department. They all know how hard it has been. Thanks also to Paquita, Ángel, Rocío, Carmina and Carmen for providing a comfortable atmosphere during the trips I spent writing. Thanks also to Robert for proofreading the thesis (remaining mistakes are mine), and to Maite and all the Norbaltic staff for all the fun.

Last but not least, this work is dedicated to my family and particularly to my parents, for their unconditional love and support, and also to my partner Rafael, with whom I begin now a much harder and longer task than a PhD: sharing a life.

Introduction

Network traffic modeling

The availability of good stochastic models of network traffic is key to developing protocols and services. A precise statistical characterization of packet interarrival time, packet size distribution, instantaneous bitrate and connection arrival rate helps engineers to design network equipment, ensure its correct operation and evaluate its performance under different load conditions. All these magnitudes are usually studied as unidimensional or multidimensional time series, and can be captured from passive network monitors[1] that record the arrival time and the size of every packet.

The classical traffic models developed for the telephone network more than a century ago, such as the Erlang and Engset models, are based on exponential distributions, Poisson arrival processes, and Markov chains [Kle76]. These models are *memoryless*, a property of the exponential distribution meaning that past events do not have influence on the future behavior of the process, i.e., the traffic process has no time correlation. This assumption is quite accurate for telephone traffic, which is usually assumed to be generated by infinite populations with independent customers, and has been successfully used by telephone engineers in the last decades.

In the early days of computer networks and packet switching other models were developed for the new traffic sources, giving birth to the discipline known as *queueing theory*. These models generally assumed uncorrelated and independent sources, and were used for network dimensioning and performance evaluation. But as networks and protocols grew in complexity the results obtained with the models and those found in reality diverged more and more. Modern network applications such as World Wide Web, FTP, Telnet, or video streaming present correlation and show differentiated traffic patterns at different time scales [PF95]; some protocols, such as TCP, have memory and are capable of maintaining the state of the connection and perform retransmissions [Ste94], thus introducing correlation in network traffic; and some traffic sources are even intrinsically correlated, such as certain variable bit rate (VBR) video codecs based on motion estimation (see for example [GW94]). The classical models failed to incorporate these correlations. More complex but still Poisson-based models were developed, such as the Correlated Interarrival Poisson Process (CIPP) [MDK99], but there was still a gap between reality and theory.

[1] such as Ethereal, an open source tool (`http://www.ethereal.com`)

An additional problem was the lack of accuracy of the traffic capturing devices, usually built around standard network interface cards (NICs) which were not designed for high-performance monitoring tasks but for normal operation. These devices were not able to provide accurate timestamps nor ensure the capture of all packets at wire speed, and therefore the traces produced by such monitoring systems were prone to errors and a hence were not a good source for stochastic models. The progress of technology made possible the capture of traffic traces with accurate timestamps around the end of the 1980s [LW91], thus allowing a really accurate characterization of network traffic.

Fractal properties of network traffic

The first studies

In a widely appraised paper, Leland, Taqqu, Willinger and Wilson analyzed the traffic at the Ethernet network of Bellcore Laboratories[2] and showed that it presented fractal properties [LTWW94]. Fractals were developed by Mandelbrot in the 1960s [Man83], and their main property is that they have the same "look" when observed at any scale, either fine or coarse; i.e., if we *zoom* a fractal image, it will always look the same. This property has been found in several natural phenomena (the crystal structure of snow flakes, or the shape of certain tree leafs, for example) and is well-known to be a typical property of time series studied in fields such as hydrology (river annual minimum and maximum levels), aerodynamics (air turbulences), or financial time series (stock prices).

What does it mean to *zoom* a time series? Consider the example of the instantaneous bitrate series. It consists of defining a certain time window δ over which we sum the sizes of the arrived packets. Let's say that we record the number of bits arrived over a window $\delta_1 = 10$ms. If we further *aggregate* the series with a window length of $\delta_1 = 100$ms, we are looking at the time series in a coarser time scale; i.e., we are "zooming out" the series.

Fractality was detected in the Bellcore traces: the intrinsic variability (burstiness) observed at the finer time scales is maintained even at the coarser time scales, contrary to what is predicted by classical, Poisson-based models, in which traffic is smoothed at the coarser scales by the effect of statistical multiplexing. Figure 1 illustrates this effect. Since the coarser time scales of a process correspond to the lower frequencies of the spectrum, fractal processes are usually related to the concept of Long-Range Dependence (LRD), which in turn is linked with a slow-decaying autocorrelation. LRD processes are said to have *memory*, contrary to the Short-Range Dependent, memoryless Poisson process.

After the seminal paper of Leland et al. which reported self-similarity in Ethernet, several authors examined other networks and traffic sources. Just to mention some of the most outstanding contributions, Garrett and Willinger [GW94] studied the long-range dependence in variable-bitrate (VBR) video, finding strong low-frequency components. Fractality has been found by Duffy

[2]The Bellcore dataset can be found at the Internet Traffic Archive: `http://ita.ee.lbl.gov/`

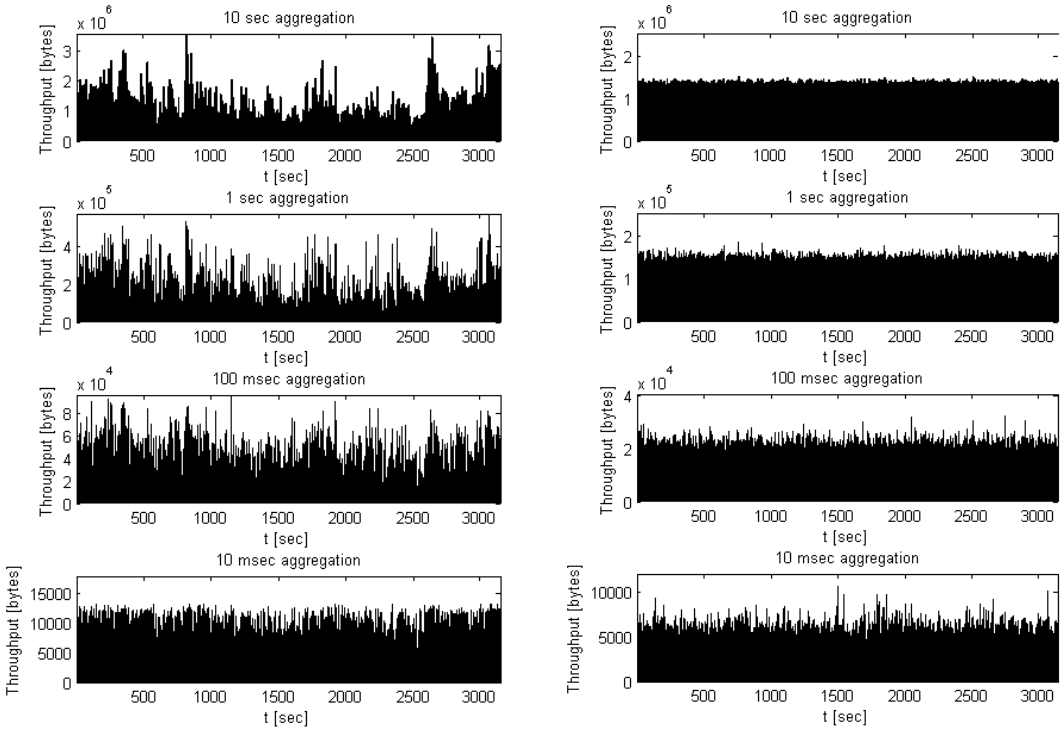

Figure 1: *Left*: the throughput (bytes arrived per time slot) of the Bellcore `BC-pAug89` trace, seen at different scales. Burstiness is clearly maintained. *Right*: a Poisson process on the same scales. The process is smoother on the coarser (higher) scales.

et al. [DMRW94] in the ITU-T SS#7 signalling system of telephone networks, which is actually a packet-switched network. Paxson and Floyd [PF95] found fractal properties in wide-area network traffic, specifically in TCP, FTP and Telnet connection arrivals. Crovella and Bestavros [CB97] detected self-similarity in World Wide Web (WWW) traffic. Finally, peer-to-peer file sharing systems such as Kazaa, Napster or eDonkey have recently been found to be much burstier than web traffic and to be well modeled by self-similar processes [JLC04][3].

Mathematical models

The Bellcore paper shook the discipline of teletraffic modeling, giving birth to a whole new world of alternative models based not on Markov chains and Poissonian processes but on a completely new class of models that were added to an already long list of traffic theoretical representations. Several mathematical models have been proposed for fractal traffic: Self-similarity, Long-range Dependence (LRD) and Multifractal processes, among them. All these processes are usually linked with heavy-

[3]The presence of self-similarity and LRD in P2P networks is still a controversial topic and needs further and deeper analysis. See for example [HMGBH06], where the authors state that P2P traffic is much smoother than classical applications in the Internet.

tailed distributions of some network-related parameters, such as file size or connection duration. Although Chapter 1 describes these models in detail, we now give a short description of the most usual forms of traffic's fractal properties:

- A *self-similar* traffic process has a similar statistical distribution regardless of the time scales at which is analyzed. In other words, the process is fractal and looks *statistically* the same at all scales.
- A *long-range dependent* process exhibits a hyperbolically, *slow-decaying* autocorrelation function, while in the case of short-range dependent input the tail decays exponentially. The presence of LRD means that the process is highly correlated and, therefore, the past evolution of the process has a strong influence in its future evolution.
- An LRD process exhibits a $\frac{1}{f^{\alpha}}$-like power spectral density, where α is the *scaling exponent* that characterizes the process. This is also known as the *scaling* or *power-law* property, since it imposes a certain variance structure of the process across time scales.
- The queue length distribution of an infinite buffer system with LRD input traffic exhibits a Weibull distribution. Therefore, the probability of finding the system in a state where the queue is very long is higher than that found for the case of Poisson traffic, which builds an exponentially distributed queue length.
- A *multifractal* process shows a highly variable short-time scale behavior, in which the fractal parameters of self-similarity or LRD are not constant nor (in some cases) measurable.
- A *general scaling* process presents non-strict scaling behavior (for example, a process with two or more scaling regimes[4] in certain scale ranges), and can be understood as a generalization of self-similar, LRD and multifractal processes.

A note about terminology

Though in the literature the terms *self-similar*, *long-range dependent*, and *fractal* are usually considered as equivalent, they are not (as we will see in Chapter 1. Besides there are *multifractal* processes, that have also been found to be of interest when modeling network traffic.

For historical reasons, researchers usually use the terms *self-similar* or *fractal traffic* in a general sense, referring to all the possible fractal characteristics of network traffic. Strictly speaking, the word "fractality" refers to a single scaling exponent describing a strict scaling. However, the research efforts carried out during the last years have shown that describing the statistical properties of the traffic with a single exponent does not match reality, and that scaling parameters vary over time, and even depend on the considered timescales. Hence, it seems that terms connected to statistical self-similarity are more appropriate, since in some sense they are more general. Other authors have used the term *scaling* as a generalization.

[4] i.e., power-laws with different shape parameters.

Our work is, strictly speaking, focused on time-varying LRD processes, although most results can also be applied to other process classes. That is why we will also use the terms *self-similar traffic* and *scaling traffic* indistinctly throughout this thesis.

Implications for network performance

Scaling phenomena can be explained in terms of the well-known variability or burstiness of network traffic, i.e., the fact that long packet bursts are followed by long periods of inactivity. The novelty is that this happens on a wide range of time scales, while classical models can only capture the burstiness on a certain *natural* scale of the system under study.

This has strong implications for network performance and results in an increase in buffer overflow probability of fractal-fed queues when compared to Markov-based models. Norros [Nor94] contributed an outstanding study of the performance of a single server queue under monofractal Fractional Brownian traffic, where it was shown analytically that self-similar traffic causes higher buffer overflow probability and/or an increase in packet delay when compared to classical Poisson-based modes, as shown in Figure 2. The subexponential distribution of the tail of the queue length distribution in queues fed with fractal traffic has important implications for network resource planning, since the smoothing effect of buffers no longer works with self-similar traffic, unless we incur prohibitive buffer sizes [ST99].

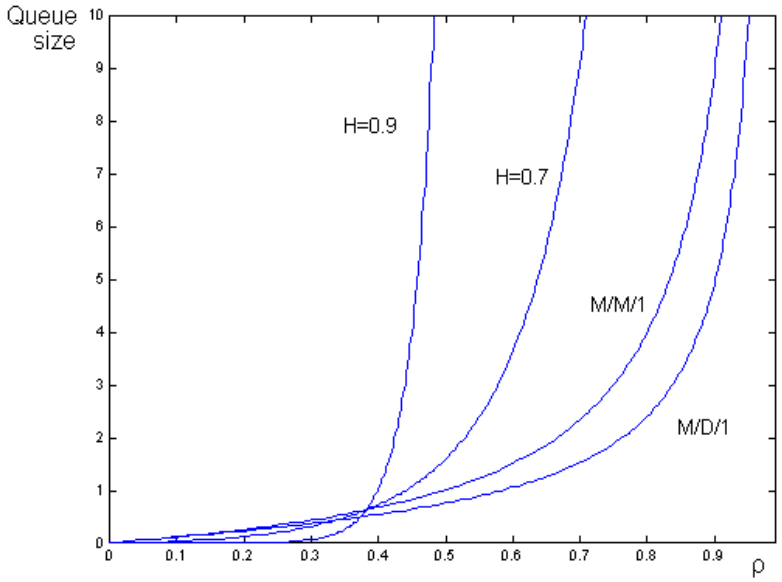

Figure 2: Mean queue size vs normalized offered traffic (utilization) for Poisson-based queue models M/M/1 and M/D/1, and FBM with H=0.7 and H=0.9 [Nor94]

Wavelet-based techniques for estimation of the scaling properties of traffic

Several estimators of the degree of self-similarity of traffic can be found in the literature (such as variance-time analysis, the rescaled adjusted range statistic (also known as R/S plot), the periodogram-based method, and Whittle's method [Mol04]). The recent (1998) application of *wavelet-based techniques* to the analysis of traffic traces represented a significant step toward the development of more accurate estimation procedures.

The Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) performs a subband decomposition of the signal spectrum in which the subband bandwidth decreases dyadically with the frequency, giving rise to a multiresolution analysis (MRA) in which the original signal is decomposed into a low-pass approximation at scale J, $a_x(J,k)$ and a set of high-pass details $d_x(j,k)$ for each scale $j = 1 \dots J$, being k the time variable. Each subband corresponds to a certain time scale, thus allowing the study of the signal simultaneously at different time resolutions, also known as scales or *octaves*. This feature is important for detecting the low frequency components of the signal, which are related to the burstiness phenomenon.

The resulting transform is orthogonal and can be efficiently implemented using a bank of quadrature-mirror low-pass and high-pass filters ($h(n)$ and $g(n)$, respectively) followed by a downsampling step [Mal89]. The downsampling step causes the number of DWT coefficients to be approximately halved at each stage. The procedure is iterated in the low pass subband as many times as the desired number of scales. In the frequency domain, the DWT provides a decomposition in subbands whose bandwidth are halved at each step. Figure 3 illustrates the algorithm and the associated subband decomposition.

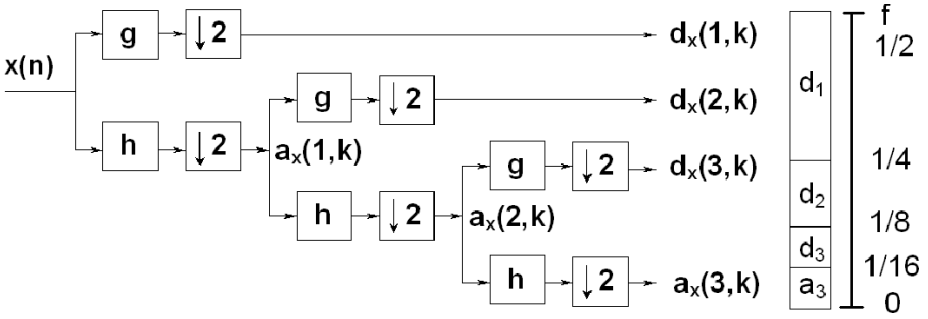

Figure 3: Left: the DWT as a filter bank (J=3). Right: the normalized frequency subband decomposition, with the approximation a_3 and details $d_{1,\dots,3}$.

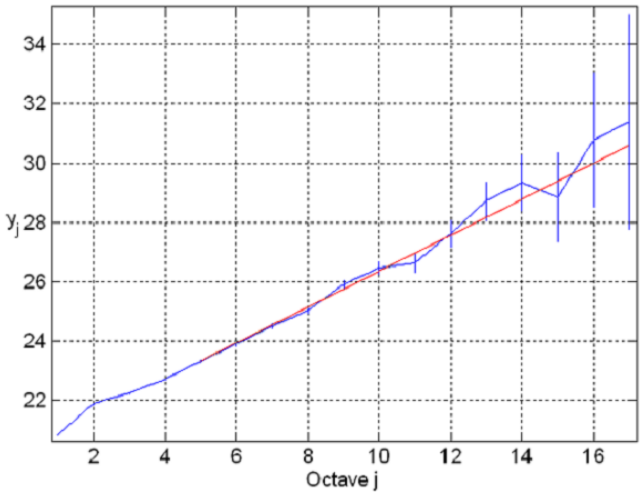

Figure 4: LogScale Diagram of the bytes-per-slot time series from the Bellcore `BC-pAug89` trace, aggregated at 10 msec, analyzed with a db3 wavelet.

Estimation of scaling parameters with the DWT

The DWT-based fractal parameter estimator developed by Abry and Veitch, also known as LogScale Diagram (LD) [VA99], is widely accepted as one of the best and most efficient estimators of scaling. It performs a joint estimation of α (the scaling parameter, a qualitative measure of the phenomenon) and c_f (the second order parameter, a quantitative measure). The scaling parameter is usually expressed in a rewritten form, the Hurst parameter $H = (1 + \alpha)/2$.

The algorithm is based on the distribution of the process' variance across scales, which is known to follow a $\frac{c_f}{|f|^\alpha}$-like power law for fractal processes. The LD estimates μ_j, the power of the signal at each scale (subband) j, and represents its logarithm versus the scale index j, turning the power law into a straight line with slope α and an offset related to c_f. Both parameters can be estimated by performing a weighted linear regression. Figure 4 illustrates the Logscale Diagram applied to the Bellcore `BC-pAug89` trace, with the characteristic variance alignment shown by scaling processes. More details of the mathematics involved are given in Chapter 3.

Alternative wavelet transforms

The choice of the DWT provides a certain trade-off between frequency and time localization whose main drawback is its lack of time resolution at the higher scales, where few wavelet coefficients are available due to the downsampling step and where LRD and self-similarity is actually to be detected, since those subbands correspond to the lower frequencies of the signal.

That is why other orthogonal and non-orthogonal wavelet transforms with different accuracy trade-offs have also been considered as tools for the analysis of scaling processes. The redundant descriptions, far from being useless, provide higher accuracy for parameter estimation and generation

of synthetic traces. Redundant wavelet transforms pave the way for an enhanced time-frequency characterization of network traffic.

Of the whole set of possibilities, we focused our attention on the following list:

- **Wavelet Packet Transform (WPT)**. The WPT is derived from the DWT by iterating the low-pass/high-pass filter decomposition not only in the approximation (or low-pass) branch at each scale, but on the whole tree. This generates, for a given depth level, a set of possible subband decompositions, being the DWT a particular case. This is a frequency-redundant decomposition that we thought could open the way towards real-time detection of the changes of the scaling parameters through an on-line automatic subband selection based on an on-line *best basis* selection algorithm. This approach was studied but we abandoned it and focused on a different family of techniques: those based on time-redundant representations.

- **Maximal Overlap Discrete Wavelet Transform (MODWT)**, also known as *Stationary (SWT)* or *Shift-Invariant Wavelet Transform*. The MODWT is derived from the DWT, by the suppression of the downsampling stage at the output of the filters. Therefore, MODWT is a time-redundant transform that produces at the output J+1 times the amount of samples at the input, being J the number of scales at which the analysis is performed (J details + approximation). This allows for a more accurate estimation of the variances at each scale, and what is more, a better characterization of the evolution of the signal in the time domain. The main drawback of the MODWT is the presence of correlation in the output of the higher scales.

- **Dual Tree (Complex) Wavelet Transform (DTWT)**. This transform can be interpreted as a double DWT-style tree in which the original samples to be analyzed in one tree are the odd positions, while the even-numbered samples are analyzed by the second tree. Its *complex* nature comes from the interpretation of every couple of coefficients from both trees (a and b) as the real and imaginary parts of a complex number $a + ib$ (i denotes imaginary unit). Given that the filters of the trees meet some conditions, the DTWT can provide an almost shift-invariant analysis with almost uncorrelated coefficients at the expense of a slight increase in redundancy (two times the input samples) and complexity when compared with DWT. This transform is a good trade-off between the DWT and the MODWT, keeping a good level of time resolution and computational load, while maintaining redundancy and correlation at an acceptable level for our purposes.

Our aim: detection of changes in the scaling parameters

Network traffic is expected to change its fractal parameters over time. An algorithm for segmenting traffic into regions with homogeneous characteristics would be useful for the development of fractal-aware network algorithms.

The existing literature on fractal traffic modeling is dominated by *fractal-stationary* models, in the sense of assuming constant values of α and c_f for the whole time series analyzed. Fractal-stationary models are not suited for real applications since measured traffic is expected to change its behavior as time goes on. Few studies ([DLO+94, RVA98, RV99a, RV99b, VA01, AAI05] among others) have explored the possibility of non-stationarity fractality or time-varying fractal parameters.

Detecting the *change points* (i.e., the instants that bound segments showing a homogeneous fractal behaviour) can thus be useful for some network mechanisms that exploit the *long-range dependence* of network traffic. Some examples of these mechanisms are the TCP congestion control presented in [HGHP02], a predictive bandwidth control for MPEG sources illustrated in [OY01], and the effective bandwidth estimator for fractal sources described in [YTJ01]. All these mechanisms rely on the estimation of the scaling parameters, and would benefit from the knowledge of a change in the fractal parameters (though they differ on the time scale at which they work; the bandwidth estimators react at the scale of seconds or tenths of second, while the TCP congestion control needs feedback at the millisecond scale). The aim of this thesis is the development of algorithms that are capable of tracking the evolution of the fractal characteristics of traffic and of signalling the changes to the aforementioned fractal-aware mechanisms, in order to make them adapt to the new situation.

Previous works

We now briefly mention some of the papers related to the study of the non-stationarity of the scaling parameters of traffic. A comprehensive, more detailed description of the work done by other authors can be found in Section 5.3.

There exists an on-line version of the Abry-Veitch estimator that performs a progressive (cumulative) computation of subband variance and returns updated estimates using all available samples [RVA98] from the beginning of the measure. The same authors also studied the robustness of the method when mean and variance shifts are present [RV99a]. Elsewhere, changes in the Hurst parameter were analyzed for a LAN trace in a 1-hour or 4-hour window [RV99b]. A statistical test of the constancy of the scaling parameter over constant-length windows has also been developed and is able to detect changes in H but is insensitive to changes in the variance structure [VA01].

Furthermore, a wavelet-based study of TCP flow arrivals revealed nonstationarities, found no strict scaling, and generalized the study to high-order scaling [Uhl04]. Finally, an accuracy-based approach has been reported for the real-time estimation of multifractality, where the window length is related to the confidence interval of the estimation [AAI05].

In another field of application, Whitcher [Whi98] has used the DWT, WPT and MODWT in procedures for testing the homogeneity of variance across scales, with applications to the study of the series of Nile river minimum water levels, ocean shear measurements, and the Madden-Julian atmospherical oscillation.

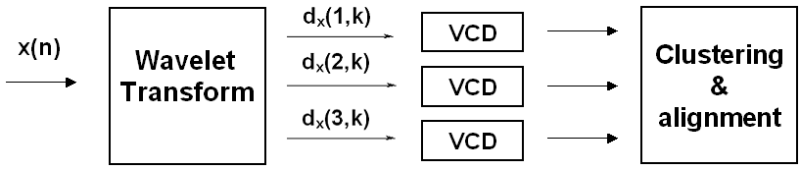

Figure 5: Scheme of our segmenting algorithm.

Our approach

We propose an algorithm capable of detecting the change points in which monofractal traffic changes its behavior; i.e., its scaling parameter α or Hurst parameter H and c_f. Our main idea follows: if the scaling parameter and/or the second order parameter of traffic change at a given moment, so does the complete variance structure at the whole range of scales. Therefore, if a change point detection algorithm is applied at the output of the wavelet transform monitors the variance at every scale, a change in α or/and c_f can be detected as a simultaneous, time-aligned variance change across scales. A final step of clustering and alignment of the changes provides the boundaries of the constant-scaling segment. Figure 5 illustrates the three steps of our method. We have experimented with the DWT, MODWT and DTWT transforms, together with the variance change point detection (VCD) methods known as Iterated Cumulative Sum of Squares (ICSS) [IT94] and the Schwarz Information Criterion (SIC) [CG97], and an alignment algorithm based on the Hough Transform [Rus99].

Network traffic shows a very complex behavior, mixing a highly non-stationary volume with changes in its fractal properties. Besides, the definition of the scaling parameters are usually asymptotical. Finally, recent research results have identified variance change phenomena, usually triggered by protocol dynamics, that take place at certain scales but not at the whole range of scales [JD05]. That is why we face a complex situation in which it is not easy to isolate each of the scaling-related phenomena present in traffic. Our algorithms are able to track the changes of the variance structure of network traffic, which can correspond to changes in the Hurst parameter, changes of variance, or simultaneous changes of both parameters.

Our approach is more general than the constant window-based methods mentioned in the previous section, since we can localize the variance-transition points to any position and scale. In addition, we can monitor the nonstationarities of the time series (both for the whole process and scale-by-scale). This is important, since LRD estimators can be fooled by variance changes or shifts in traffic volume [DLO$^+$94, MD00].

To sum up, the aim of the present dissertation is to study the *time evolution* of fractal traffic parameters via a *wavelet-based analysis*. The long-term goal of our research is a *real-time* characterization of fractal traffic which, in turn, can lead the way to the development of new fractal-aware network algorithms, exploiting the properties of self-similarity and long-range dependence.

Our contributions

The main contributions of this dissertation are:

- A comprehensive description of the field of network traffic modeling, the presence of fractal properties, and the previous works related to the study of the non-stationarity of the scaling parameters of traffic [RSed, Rin05].
- The introduction of alternative wavelet transforms (WPT, MODWT, DTWT and DDDWT) as possible tools for characterizing network traffic.
- The extension of the DWT-based LogScale Diagram to the WPT, as a first step towards an on-line joint method composed of the WPT estimator and a fast best-basis splitting algorithm capable of adapting itself to the time-frequency characteristics of the signal [RS02, RS04a, RS04b, Min05].
- The development of joint wavelet - VCD methods for detecting the changes in the variance structure of scaling processes. DWT [RS05a, RS05b, RMSP05b], MODWT [RS05c, RSdA05, Min05, RMS06] and DTWT [Rin05, ZR06, RZ06] have been combined with ICSS and SIC, including a progressive (in the sense of near-real-time) version of the DWT algorithm [RMCS04, RS05a, Min05, Per06].
- The development of an automatic clustering and alignment detection procedure based on the Hough transform, as the final step of the joint wavelet-VCD algorithms [Min05, RMSP05b, RMSP05a, RMS06].
- The performance evaluation of each of the proposed methods when applied to synthetic and real traffic traces, including empirical powers of the change detection tests.
- The analysis and segmentation of real traffic traces (from the Bellcore dataset) and the confirmation of its nonstationary scaling behavior [RS04b, Min05, RMS06, Rin05].

Outline of the thesis

We now give a brief outline of the structure of this book. Chapter 1 serves as an introduction that provides the necessary technical background on fractal traffic, including useful definitions for the rest of the work. Furthermore, it gives an overview of the various research activities surrounding self-similar traffic and outlines the principal issues in the areas of traffic and physical modeling, queuing analysis and traffic control. Finally, it will present the results related to tracking the evolution of the fractal parameters of traffic, and will emphasize the need for an algorithm capable of performing such an analysis in real time.

Chapter 2 discusses the mathematical basics of the wavelet theory and its relation to scaling phenomena in traffic series. Due to its ability to localize a given signal in both time and scale

(or frequency), the wavelet transforms provide a powerful and refined technique for detecting and quantifying scaling behavior (self-similarity, LRD, etc) in network traffic. On the other hand, the aforementioned trade-off between time localization precision and redundancy will be described, together with the alternative redundant wavelet transforms: WPT, MODWT and DTWT.

Chapter 3 illustrates the issues of detection and measurement of scaling behavior under the hypothesis of *stationary monofractality*; that is, of stationarity of the LRD and self-similar scaling parameters across time. In order to achieve this purpose, the following wavelet-based *static* estimators will be discussed:

- a DWT-based estimator (the *Abry-Veitch estimator*), built on a graphic tool called the *LogScale Diagram*;
- an extension of the *LogScale Diagram* for the WPT.

This chapter contains the first of our contributions: the generalization of the LogScale Diagram for the WPT. This opens the way for future developments of joint WPT-best basis algorithms, as was our first intention. The WPT-based estimator, though, stands by itself.

Chapter 4 illustrates the issues of detection of variance changes in unidimensional time series. This chapter begins with the statement of the variance change detection problem and follows with the description of two algorithms proposed in the literature: the Iterated Cumulative Sum of Squares (ICSS) and the Schwarz Information Criterion (SIC), formerly introduced in the contexts of financial series and stock price analysis, respectively. The performance of both methods is studied for both uncorrelated and correlated time series, in order to characterize their behavior.

Chapter 5 justifies the need for the study of the changes in the variance structure of the traffic processes and its *segmentation* in regions with homogenous behavior, that in some cases can be classified as *locally stationary monofractal* process. This chapter presents our time-segmenting algorithm. The primary matter is naturally related to the joint use of wavelet transforms and change points detection algorithms, as previously explained. The methodology used in the comparison of the different methods is exposed here. The automatic alignment method based on the Hough transform is also described and characterized. This chapter is the introduction to our main contribution: the study of the performance of the joint algorithms, tested with synthetical and real traffic traces.

Chapters 6, 7 and 8 are dedicated to the description and evaluation of the three families of methods we have developed, classified by their wavelet transforms: Chapter 6 describes the DWT-ICSS and DWT-SIC methods (including its progressive version), while Chapters 7 and 8 focus on the alternative wavelet-based methods: MODWT and DTWT-based algorithms, respectively. For each of these transforms, a set of synthetic and real traces will be analyzed, including an empirical statistical test of the power of the algorithms.

The dissertation ends with the conclusions, identification of open issues, and suggestions for future work. Appendix A describes the Double Density Discrete Wavelet Transform (DDDWT), another alternative wavelet transform with almost shift-invariant properties, whose use with the variance change detection algorithms is one of the future lines of development of the present work.

Chapter 1

Scaling properties of network traffic

1.1 Introduction

This chapter presents technical background on the scaling properties found in network traffic, together with an overview of the various research activities surrounding the topic, such as the development of the mathematical models, the origins of the scaling phenomenon, and its impact in network performance. Special emphasis will be put on the challenge of tracking the evolution of the variance structure and the scaling parameters of traffic. It is not our aim to provide an in-depth, systematical description of all the theory related to the topic. For the interested reader, we recommend the excellent tutorials provided by Park and Willinger [PW00], and by Hlavacs et al. [HKS99]. A shortened version of this chapter has been accepted for publication in [RSed].

1.1.1 Renewal models

The classical models used in teletraffic modeling belong to the class of *renewal* processes. In a renewal process, the observations $X(t)$ are independent and identically distributed, though their distribution is allowed to be general [HKS99]. The independency property means that the observations do not depend on the past, and therefore the autocorrelation function $\rho(k)$ for all lags $k \neq 0$ are equal to zero.

Poisson processes

A Poisson process describes the arrival of observations (for example, telephone calls or data packets) at certain instants in time.

Definition 1.1. *Given an exponentially distributed interarrival time random variable, with mean* $\frac{1}{\lambda}$*, the number of arrivals within an interval of length* T *is described by the Poisson process:*

$$Prob[k\ arrivals\ in\ T] = \frac{(\lambda T)^k e^{-\lambda T}}{k!}$$

Such a process has a mean arrival rate λ, measured in arrivals per time unit. Poisson processes are considered *memoryless*, i.e., the number of arrivals occurring in any bounded interval of time after time t is independent of the number of arrivals occurring before time t. The memorylessness comes from the exponentially distributed nature of its interarrival time: the expected time of the next arrival is independent of the time since the last observation. Therefore, the process has no memory and the arrivals are considered as independent. The Poisson-based traffic models usually assume an infinite population of independent customers.

Markov models

Markov models introduce some dependency in the observations. A Markov process with discrete state space $x_n, n \in \mathbb{N}$ is called a *Markov chain*, if the probability that the next observed value (or *state* of the Markov chain) $x_{n+1} = j$ depends only on the current state $x_n = i$. The dependency is, therefore, limited to one lag in discrete time. Therefore, the entire past history is summarized in the current state of the process [Kle76].

There are variations of the simple model, such as the Markov-modulated Processes, in which an auxiliary Markov chain is evolving in time, and its state controls the probability law of the model. In Markov-modulated Poisson Processes (MMPP) the current state n of a superimposed Markov chain defines the current arrival rate λ_n of the modulated Poisson process. The MMPP's have been used in traffic modeling in the recent past [HL86].

Application of renewal processes

Poisson and Markov-based arrival processes are very attractive due to their mathematical tractability and their connection to the queueing theory. Besides that, such kinds of models make the challenges of traffic control a lot easier, due to a simple correlation structure that decays rapidly and ensures that a *bad event* such as a packet burst will not affect the system excessively. When such models are aggregated in time (as in a multiresolution analysis), the (residual) dependence is rapidly lost and the nice properties of independent random variables apply.

1.1.2 Traffic traces and its aggregation

Traffic traces are composed of at least two columns: a timestamp (the time at which the packet was captured) and packet size. Further information such as the port or the origin and destination addresses can be useful for filtering different flows. In traffic studies, the series of interest are usually the packet interarrival time (the difference between consecutive timestamps) and the instantaneous

bit rate (the number of bytes that arrive during time slots of a constant length). In both cases, we get a one-dimensional time series over which scaling estimation methods can be applied.

The aggregation or rescaling of the packet arrival process plays an important role in performance analysis and control. As stated in [PW00], "*(...) buffering and, to some extent, bandwidth provisioning can be viewed as operating on the rescaled process*". Traffic traces are, therefore, studied on different time scales through aggregation.

Definition 1.2. *Given a continuous-time stochastic process $X(t)$ defined in $t >= 0$, we define its discrete-time aggregated version $X^m(k)$, $k >= 0$ at aggregation level m as the average of $X(t)$ in non-overlapping blocks of size m:*

$$X^{(m)}(k) = \frac{1}{m} \sum_{t=km}^{(k+1)m} X(t) \qquad k = 0, 1, 2 \dots$$

As the aggregation level increases, and given that the original signal has finite length (which is the case of real traffic traces), the amount of samples of the rescaled process decreases by a factor m.

1.1.3 An intuitive definition of self-similarity

Self-similarity (or *fractality*) describes the scale-invariance (in space or time) of an object, for example an image or a time series. When part of a fractal is magnified, a similar object is obtained.

The *1D Cantor set* depicted in Figure 1.1 (left) is an example of a fractal object. This set, defined in the $[0, 1]$ support, can be obtained by equally dividing a segment of length 1 in three parts, scaling its size by $1/3$, placing two copies of this scaled segment at the two edges of the support and then applying recursively the same process of scaling and translation *ad infinitum*. An alternative version of the 1D Cantor set can be obtained by relaxing the condition of equality of the *mass* distribution among segments, as long as the total mass is preserved at each stage of the construction. Figure 1.1 (middle) shows such a construction with weights $\alpha_L = 2/3$, $\alpha_R = 1/3$ for the left and right components, respectively.

In our context, we can interpret the result of both versions of the Cantor set as traffic intensity in a communications network. In the first case we obtain an on/off source that can model the transmission of constant-length packets (for example, ATM cells, 53 bytes long). In the second case we can interpret the values obtained as the amount of packets or bytes arrived per time slot. In both cases the obtained random variable is, by construction, exactly self-similar.

Consider a discrete-time stochastic process $X(k),\ k \in \mathbb{Z}$, that can be interpreted as the traffic volume measured in bytes at time slot k of length m, following the intuition established with the Cantor set example. The total traffic volume *up to* time t (from time 0) is also of interest. Let's consider the corresponding *cumulative* process $Y(t)$, whose differences (also called *increment process*) constitute the original $X(t)$ Cantor process. In our case, let us assume that time is discrete and, therefore, we can equivalently index the time steps by $t = 0, 1, 2, \dots$ and define the cumulative

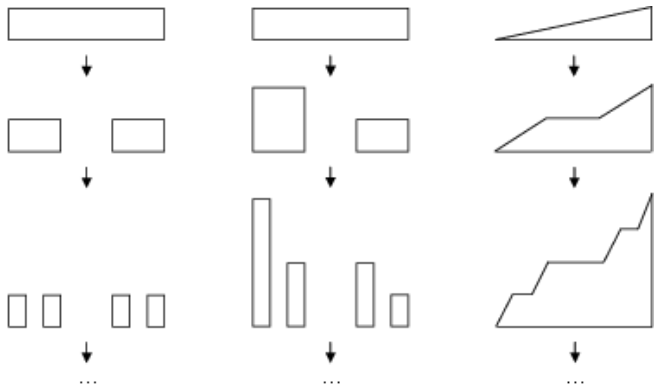

Figure 1.1: *Left*: 1D Cantor set interpreted as on/off traffic. *Middle*: One-dimensional nonuniform Cantor set with weights $\alpha_L = 2/3$, $\alpha_R = 1/3$. *Right*: Cumulative process corresponding to the 1D on/off Cantor traffic depicted on the left. Reproduced from [Min05].

process $Y(t)$ as:

$$Y(t) = \sum_{k=0}^{t} X(j), \quad t = 1, 2, \ldots \tag{1.1}$$

and equivalently,

$$X(t) = Y(t) - Y(t-1), \quad t = 1, 2, \ldots \tag{1.2}$$

The result is shown in Figure 1.1 (right). Now $Y(t)$ represents the total traffic volume up to time t, whereas $X(t)$ represents the traffic intensity during the i^{th} interval.

A further generalization can be achieved by abandoning the constraint of the *deterministic* distribution rule and substituting it with a *stochastic* rule in which a random variable decides the amount of mass to be assigned to each segment. This is more useful for modeling purposes, since teletraffic possesses an intrinsic stochastic variability component. Of course, the concept of fractality is somehow lost, since the shape of the process cannot be exactly the same, but its statistical distribution can. That is why teletraffic studies implicitly assume *stochastic self-similarity* of the input process. This is the case of the Bellcore trace `BC-pAug89`, shown in Figure 1.2 at different scales.

1.2 Mathematical models

This section provides a brief introduction to the necessary mathematical background about scaling processes. For a more exhaustive discussion regarding these concepts, see [Ber94] and [PW00].

1.2.1 Second-order self-similarity

In teletraffic, time series coming from traffic traces are implicitly considered stationary. For this reason we would like $X(t)$ (the traffic intensity or instantaneous bitrate) to be stationary in the

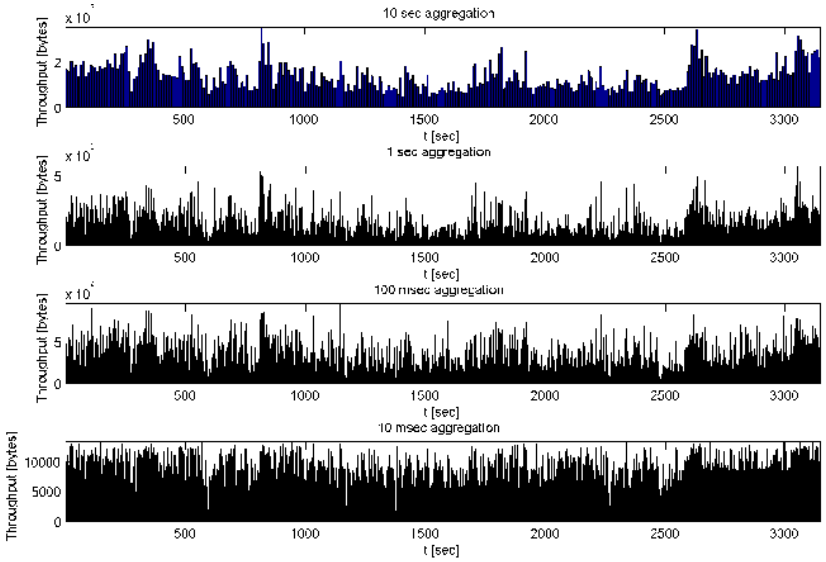

Figure 1.2: Throughput (bytes arrived per time slot) of the Bellcore `BC-pAug89` trace, at the 10 ms, 100 ms, 1 second and 10 seconds scales.

sense that its behavior or structure is invariant with respect to time shifts. What follows is a more formal definition of this concept.

Definition 1.3. *The process $X(t)$ is strictly stationary if the original and the shifted processes are statistically equivalent, regarding its finite-dimensional distributions:*

$$X(t) =^d X(t+k) \quad \forall\ t,\ k \in \mathbb{Z}$$

or, equivalently, if $[X(t_1), X(t_2), \ldots, X(t_n)]$ and $[X(t_1+k), X(t_2+k), \ldots, X(t_n+k)]$ possess the same joint distribution for all $n \in \mathbb{N}^+,\ t_1, \ldots, t_n,\ k \in \mathbb{Z}$.

Real traffic traces are not strictly stationary. Therefore, we have to define a weaker, more tolerant concept of stationarity, the *second-order stationarity.*

Definition 1.4. *A process $X(t)$ is said to be second-order stationary if its mean $\mu(t)$, variance $\sigma^2(t)$ and autocovariance $\gamma(t,s)$ (this is, its first and second-order statistics) satisfy:*

$$\begin{aligned}
\mu(t) &= E\{X(t)\} = \mu \quad \forall\ t \in \mathbb{Z} \\
\sigma^2(t) &= E\{[X(t)-\mu]^2\} = \sigma^2 \quad \forall\ t \in \mathbb{Z} \\
\gamma(t,s) &= E\{[X(t)-\mu][X(s)-\mu]\} = \gamma(t+k, s+k) \quad \forall\ t,\ s,\ k \in \mathbb{Z}
\end{aligned}$$

In the previous definition the first two moments of the process are assumed to exist and be finite. From now on, we will assume $\mu = 0$. Due to the stationarity property $\gamma(t,s) = \gamma(t-s, 0)$ and since time-shifts do not really affect the results, from now on we can denote the autocovariance function as $\gamma(k)$.

Under the assumption of second-order stationarity, we can derive the following definitions.

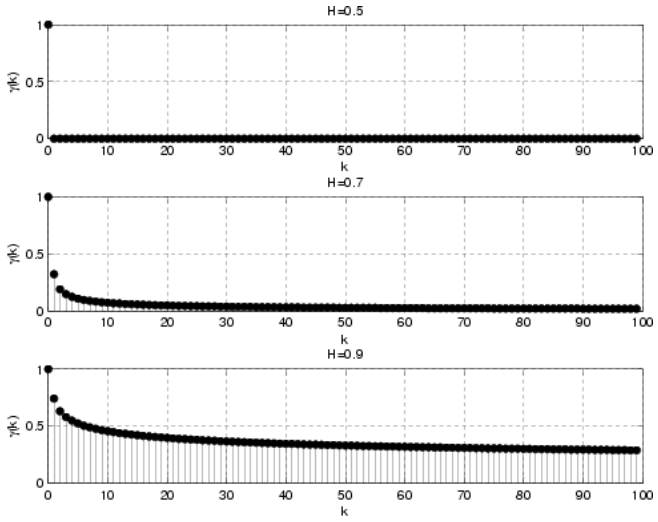

Figure 1.3: Autocovariance function of an exactly second-order self-similar process for different values of the Hurst parameter. *Top to bottom*: $H = 0.5, H = 0.7, H = 0.9$. Reproduced from [Min05].

Definition 1.5. *$X(t)$ is exactly second-order self-similar with Hurst parameter H, $1/2 < H < 1$, if its autocovariance function is given by the following expression:*

$$\gamma(k) = \frac{\sigma^2}{2}\left[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right] \quad \forall\ k \geq 1 \tag{1.3}$$

Definition 1.6. *Let $\gamma^{(m)}(k)$ denote the autocovariance function of $X^{(m)}(i)$; then $X(t)$ is asymptotically second-order self-similar with Hurst parameter H, $1/2 < H < 1$, if*

$$\lim_{m\to\infty} \gamma^{(m)}(k) = \gamma(k) = \frac{\sigma^2}{2}\left[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right] \quad \forall\ k \geq 1 \tag{1.4}$$

Equation (1.3) implies $\gamma(k) = \gamma^{(m)}(k)$ for all $m \geq 1$. Therefore, second-order self-similarity captures the property that the correlation structure is exactly (as shown in Equation (1.3)) or asymptotically (as shown in Equation (1.4)) preserved under time aggregation.

A plot of the $\gamma(k)$ function for different values of the Hurst parameter is given in Figure 1.3. From the figure it is clear that the correlation of the samples extracted from the process increases with the Hurst parameter. In particular, $H = 0.5$ corresponds to uncorrelated samples. The link between this correlation structure and the nature of self-similar traffic is often expressed in literature (for example, in [LTWW94]) by the following proposition.

Proposition 1.1. *The Hurst parameter is a measure of the burstiness of the traffic series $X(t)$: the higher the H, the higher the number of scales across which the burstiness is kept.*

1.2.2 Distributional self-similarity

In order to understand the expression for $\gamma(k)$ in the previous definition of second-order self-similarity, let's take a second look to self-similar processes.

Consider the cumulative, *continuous-time* process $Y(t)$, $t \in \mathbb{R}$. A definition of self-similarity (in the sense of finite-dimensional distributions) for continuous-time processes follows:

Definition 1.7. *$Y(t)$ is self-similar with Hurst parameter H, $0 < H < 1$, denoted as H-SS, if*

$$Y(t) =^d a^{-H} Y(at) \quad \forall\ a > 0,\ t \geq 0 \tag{1.5}$$

where $=^d$ means *equally distributed*. The definition states that $Y(t)$ and its time scaled version $Y(at)$ normalized by a^{-H} follow the same distribution. Observe that, unless degenerated ($Y(t) = 0$ for all $t \in \mathbb{R}$), the process $Y(t)$ *cannot* be stationary. Its increment process $X(t)$, however, can be stationary. Recall that in our context $Y(t)$ is the cumulated traffic process, and $X(t)$ is the instantaneous traffic. We can define $X(t)$ as *self-similar with stationary increments* and denote it by H-SSSI, if it verifies the following definition:

Definition 1.8. *$Y(t)$ is said to be H-SSSI if it is H-SS and its increment process $X(t) = Y(t) - Y(t-1)$ is stationary.*

Which is the connection between continuous-time distributional self-similarity and discrete-time second-order self-similarity? Recall that the latter requires exact or asymptotic invariance with respect to second-order statistical structure of the aggregated time series $X^{(m)}$. The answer follows:

Proposition 1.2. *If $Y(t)$ is a H-SSSI process, then its increment process $X(t)$ satisfies*

$$X(k) =^d m^{1-H} X^{(m)}(k) \quad \forall\ k \in \mathbb{Z}^* \tag{1.6}$$

Proof. Noting that $X^{(m)}$ can be viewed as a sample mean, we have

$$\begin{aligned} X^{(m)}(k) &= \frac{1}{m} \sum_{t=km}^{(k+1)m} X(t) = \frac{1}{m}[Y((k+1)m) - Y(km)] \\ &=^d \frac{1}{m} m^H [Y(k+1) - Y(k)] = m^{H-1} X(k) \\ \Rightarrow X^{(m)}(k) &=^d m^{H-1} X(k) \end{aligned}$$

Since $X^{(m)}(k) =^d m^{H-1} X(k)$, then $X(k) =^d m^{1-H} X^{(m)}(k)$ ■

Equation (1.6) shows how $X^{(m)}(k)$ is related to $X(k)$ via a simple scaling relationship involving H in the sense of finite-dimensional distributions. Therefore, Equations (1.3) and (1.4) express the fact that $X(k)$ and $m^{1-H} X^{(m)}(k)$ are required to have exactly or asymptotically the same second-order structure. As a result, depending on whether a discrete time process $X(k)$ satisfies Equation (1.6) for all $m \geq 0$ or only as $m \to \infty$, the process $X(k)$ is said to be *exactly self-similar* or *asymptotically self-similar*. In the Gaussian case this definition coincides with second-order self-similarity.

1.2.3 Long-range dependence (LRD)

In the previous pages we have focused on self-similarity in the second-order stationary and distributional senses, and paid little attention to the Hurst parameter H and its range of values. In order

to fill this gap, we have to go back to the definition of second-order self-similarity of a process $X(t)$ and to its autocovariance function $\gamma(k)$.

For $0 < H < 1$, $H \neq 1/2$, it holds [Ber94]:

$$\gamma(k) \sim \sigma^2 \; H(2H-1)k^{2H-2}, \quad k \to \infty \tag{1.7}$$

In particular, if $1/2 < H < 1$, with the settings

$$c_\gamma = \sigma^2 \; H(2H-1), \; c_\gamma > 0 \tag{1.8}$$

$$\alpha = 2H - 1, \; 0 < \alpha < 1 \tag{1.9}$$

the asymptotical behaviour of $\gamma(k)$ can be written as

$$\gamma(k) \sim c_\gamma \; k^{-(1-\alpha)}, \quad k \to \infty \tag{1.10}$$

Therefore, the autocovariance function decays hyperbolically, *slower* than a negative exponential. Such a function is not summable:

$$\sum_{k=-\infty}^{+\infty} \gamma(k) = \infty$$

and we can state the following definition.

Definition 1.9. *A stationary process $X(t)$ is said to be long-range dependent (LRD) if its autocovariance function is not summable; otherwise, we call it short-range dependent (SRD).*

An equivalent definition can be given in the frequency domain, studying the *power spectral density* (PSD) of $X(t)$. The PSD is defined as the Fourier transform of the autocorrelation function, which in our case coincides with the autocovariance function since we assumed $\mu = 0$):

$$S(f) = \sum_{k=-\infty}^{+\infty} \gamma(k) \; e^{-j2\pi kf} \tag{1.11}$$

The long-range dependence property then becomes:

$$S(f) \sim c_f \; |f|^{-\alpha}, \quad f \to 0 \tag{1.12}$$

where $S(f)$ satisfies, in the case of discrete-time processes

$$\gamma(0) = \sigma^2 = \int_{-1/2}^{1/2} S(f)df \tag{1.13}$$

As stated, $S(f)$ diverges around the origin, implying large contributions by low-frequency components of the process $X(t)$. This is shown in Figure 1.4 for different values of the Hurst parameter.

Each of the two definitions of long-range dependence includes two parameters: (α, c_γ) and (α, c_f), respectively. These are equivalent as [Ber94]:

$$c_f = 2(2\pi)^{-\alpha} \; c_\gamma \Gamma(\alpha) \; \sin\frac{(1-\alpha)\pi}{2} \tag{1.14}$$

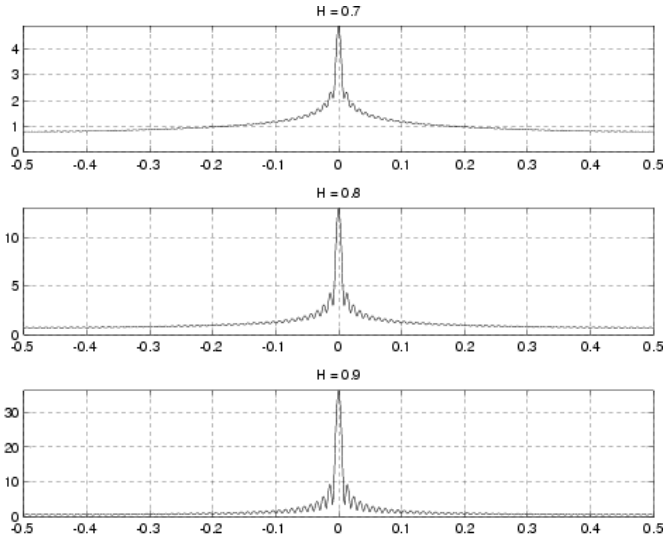

Figure 1.4: Power spectral density of a LRD process for different values of the Hurst parameter. *Top to bottom*: $H = 0.7, H = 0.8, H = 0.9$. Reproduced from [Min05].

where Γ is the Euler function. In each pair, α is the dimensionless *scaling parameter*: it is the most important, as it defines the existence of the phenomenon itself and describes the *qualitative* nature of scaling. Consequently, the emphasis has traditionally been on the estimation of α or, equivalently, of the Hurst parameter, related to α by Equation (1.9). The second parameter, c_γ or c_f, is an independent, *quantitative* parameter with dimensions of variance which has received far less attention, and the importance of its estimation has been largely neglected. This is unfortunate, because in applications the second parameter plays a major role in fixing the absolute *size* of LRD-generated effects, the general character of which is determined by α. The importance of c_f has been highlighted in [VA99], leading to a joint estimator $(\widehat{\alpha}, \widehat{c_f})$. We will discuss it again in Chapter 3.

We conclude this paragraph by pointing out the impact of the H value on the autocovariance function $\gamma(k)$:

- if $H = 1/2$, then $\gamma(k) = \delta(k)$, and $X(t)$ is completely SRD, that is, its samples are completely uncorrelated;
- if $0 < H < 1/2$, we have $\sum_{k=-\infty}^{+\infty} \gamma(k) = 0$, a condition rarely found in practical applications;
- $H = 1$ is uninteresting, since it leads to the degenerate situation $\gamma(k) = 1$ for all k;
- finally, values $H > 1$ are prohibited, due to the stationary condition on $X(t)$.

1.2.4 Self-similarity versus long-range dependence

From the previous discussion one can conclude that there are self-similar processes whose increment processes are not long-range dependent, and vice-versa. For example, Brownian motion is 1/2-SSSI

with white Gaussian noise as its increment process, but the latter is not long-range dependent. Our analysis, however, will not need this distinction: the reason can be found in the following proposition (proof will be omitted).

Proposition 1.3. *In the case of asymptotic second-order self-similarity and by the restriction* $1/2 < H < 1$*, self-similarity implies long-range dependence, and vice-versa.*

For this reason and for the fact that asymptotic second-order self-similar processes are employed as *canonical* traffic models, that the authors use *self-similarity* and *long-range dependence* interchangeably. Besides, the expression *scaling traffic* is also found in the literature.

1.2.5 Heavy tails

Heavy tails are intimately related with LRD and self-similarity. Heavy tails have been found in network-related variables (HTTP or FTP file sizes, TCP and HTTP connection durations) and have been identified as being responsible for the associated traffic's long-range dependence.

Definition 1.10. *The distribution of a random variable* X *is said to be heavy-tailed if its tail decays hyperbolically:*

$$Prob(X > x) \sim cx^{-\alpha}, \quad x \to \infty \tag{1.15}$$

where c is a positive constant and $0 < \alpha < 2$ is the tail index or shape parameter.

Heavy-tailed distributions have infinite variance and can have infinite mean if $0 < \alpha < 1$, though the latter case is not of much use in the teletraffic modeling field.

The Pareto distribution is probably the best example of a heavy-tailed distribution, and is frequently used in teletraffic research:

$$Prob(X \leq x) = 1 - \left(\frac{b}{x}\right)^{\alpha}, \quad b \leq x \tag{1.16}$$

where $0 < \alpha < 2$ is the shape parameter and b is the location parameter. The mean of the Pareto distribution is given by

$$E[X] = \frac{\alpha b}{\alpha - 1} \tag{1.17}$$

and its variance in infinite. There are other distributions, such as Weibull and log-normal, that although having *subexponential* tails, possess finite variance. Exponential and Gaussian distributions are light-tailed distributions.

Why do heavy tails play such a prominent role in scaling traffic research? Heavy tails are present in extremely variable random variables, with nonnegligible probabilities of generating large values. If these variables are used for modeling packet interarrival times or instantaneous bitrate series, we get the behavior found in computer networks regarding the burstiness of traffic: large sizes of packet bursts do appear frequently, followed by long periods of inactivity (*silence*).

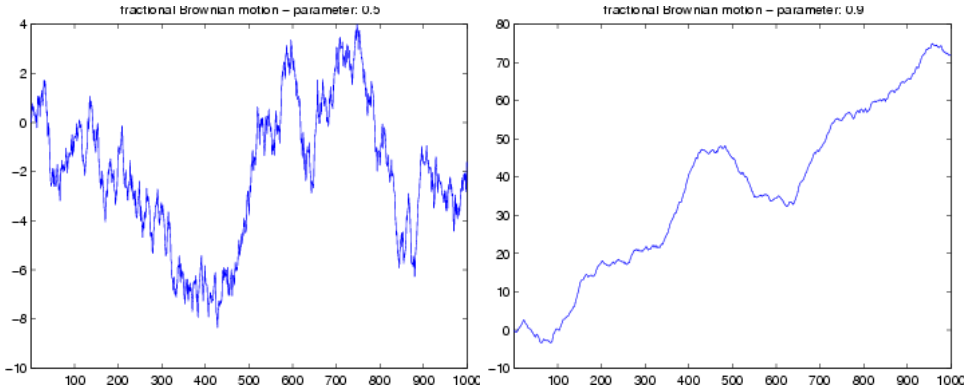

Figure 1.5: Two sample paths of FBM processes. *Left*: H=0.5, reducing to standard Brownian Motion. *Right*: H=0.9, with a clear non-stationary trend.

1.2.6 Fractional Brownian Motion (FBM) and Fractional Gaussian Noise (FGN)

We introduce now two processes that will be especially useful as *traffic models* in our simulations: Fractional Brownian Motion (FBM) and its increment process, Fractional Gaussian Noise (FGN). These are Gaussian, self-similar processes with long-range dependence [MN68]. Due to its Gaussianity, they are especially useful as *aggregate* traffic models, since (under certain regularity conditions) the aggregation of independent and identically distributed traffic sources leads to Gaussian distribution by the central limit theorem.

Definition 1.11. *$Y(t)$, $t \in \mathbb{R}$, is called Fractional Brownian Motion (FBM) with Hurst parameter H, $0 < H < 1$, if it is Gaussian and H-SSSI.*

Fractional Brownian Motion was defined by Mandelbrot and Van Ness [MN68] as:

$$B_H(t) = \frac{1}{\Gamma(H+\frac{1}{2})} \int_{-\infty}^{0} [(t-s)^{H-\frac{1}{2}} - (-s)^{H-\frac{1}{2}}] dW(s) + \int_{0}^{t} (t-s)^{H-\frac{1}{2}} dW(s) \tag{1.18}$$

where W(s) denotes a Wiener process defined on $(-\infty, +\infty)$ [PV95].

The main properties of FBM are [KGD03, HKS99]:

- $B_H(0) = 0$.
- $B_H(t)$ has independent increments.
- $E[B_H(t) - B_H(s)] = 0$.
- $B_H(t+T) - B_H(t)$ is normally distributed, $N(0, \sigma|\delta|^H)$.
- $E[(B_H(t+T) - B_H(t))^2] = V_H T^{2H}$, where V_H is a constant.
- $E[B_H(t)B_H(s)] = \frac{\sigma^2}{2}(|t|^{2H} + |s|^{2H} - |t-s|^{2H})$

In [Nor94], FBM is used to characterize the number of packet arrivals in the interval (0,t), $N_t = mt + \sqrt{am}Z_t$, where m denotes the mean of the process, a is the coefficient of variation, and Z_t is the normalized FBM with Hurst parameter H.

Figure 1.5 illustrates the behavior of two sample paths of FBM with different values of H, where it can be seen that H modulates the roughness of the time series.

Definition 1.12. *$X(t)$, $t \in \mathbb{Z}$, is called Fractional Gaussian Noise (FGN) with Hurst parameter H if it is the increment process of a FBM process with parameter H.*

The main properties of a FGN process $G_H(t)$ are [HKS99]:

- $G_H(t) = \frac{1}{\delta}(B_H(t+\delta) - B_H(t))$.
- $G_H(t)$ is normally distributed, $N(0, \sigma|\delta|^{H-1}$.
- $E[G_H(t+\tau)G_H(t)] = \sigma^2 H(2H-1)|\tau|^{2H-2}) \quad , \tau \gg \delta$.

Besides, discrete time FGN has the following autocorrelation function:

$$\rho_X(k) = \frac{1}{2}(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}), \quad k \geq 1. \tag{1.19}$$

which is not summable and denotes LRD.

The values of the scaling parameter α of the FGN LRD process and the Hurst parameter H of the associated FBM self-similar process are related by equation 1.9. That is why researchers usually mention the "H parameter of LRD traffic", though it is not strictly correct, as the rewritten version of α.

By the definition of H-SSSI, when $H = 1/2$:

- FBM reduces to Brownian motion;
- FGN reduces to white Gaussian noise.

Since Gaussian processes are characterized by their second-order structure, for each H, $0 < H < 1$, there is a unique Gaussian process that is the stationary increment of a H-SSSI process. For the same reason, for Gaussian processes, distributional self-similarity and second-order self-similarity yield equivalent definitions. Figure 1.6 shows two examples of FGN processes.

1.2.7 Multifractals

LRD and self-similar processes fall into the class of *monofractal* processes; this is, its scaling exponent α or Hurst parameter H remains constant with time. If this constraint is relaxed, we get *multifractal* processes, whose Hurst parameter can change with time.

Of particular interest is the Multifractional Brownian Motion (mFBM), defined as [PV95]:

$$B_{H_t}(t) = \frac{1}{\Gamma(H_t + \frac{1}{2})} \int_{-\infty}^{0} [(t-s)^{H_t - \frac{1}{2}} - (-s)^{H_t - \frac{1}{2}}] dB(s) + \int_0^t (t-s)^{H_t - \frac{1}{2}} dB(s) \tag{1.20}$$

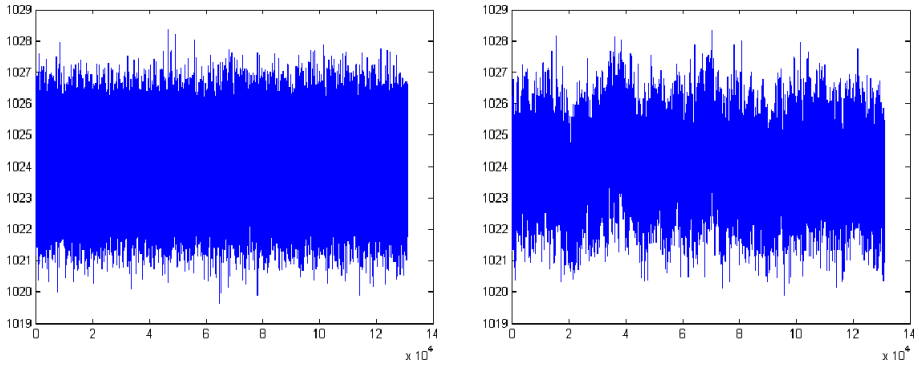

Figure 1.6: Two sample paths of FGN processes generated with a mean value of 1024 and variance 1. *Left*: FGN with H=0.5: short-range dependent, resembling white noise. *Right*: FGN with H=0.9: long-range dependent, with remarkable low-frequency components.

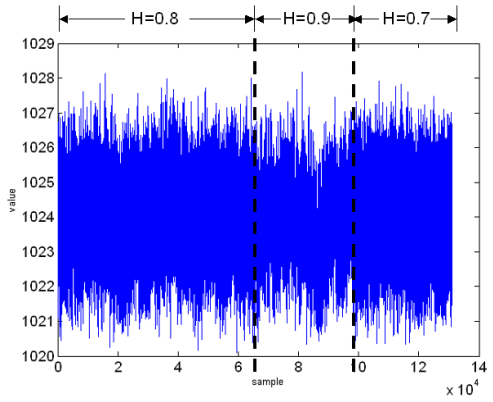

Figure 1.7: Example of mFGN trace composed by three FGN segments with mean 1024, variance 1 and different values of the Hurst parameter: $H = 0.8$, $H = 0.9$ and $H = 0.7$.

where $B(s)$ is the ordinary Brownian motion process and the integration is taken in the mean square sense. The Hurst parameter of such a process, $H_t = H(t)$ is a function of time and can be estimated[1]. In our work we have used synthetic FGN traces with piecewise constant Hurst parameter, such as the one shown in Figure 1.7. We can consider these signals as Multifractional Gaussian Noise (mFGN), the increment process of mFBM.

In general, multifractals are not as easy to define as the particular case of mFBM and mFGN. Multifractals focus on the *local* scaling behavior of the process, and can be characterized by the Hölder exponent $\alpha(t)$, defined later in Equation 1.21[2]. Unfortunately, the Hölder exponent is difficult to work with, and usually one can only determine its *multifractal spectrum* $f(\alpha)$; that is, its distribution.

An alternative definition can be obtained by considering that the local strength of singularity

[1] In this case, $H(t)$ is usually called a *Hölder function*.

[2] Note that this parameter does not necessarily coincide with an LRD time-dependent scaling parameter.

of a function $X(t)$ is characterized by the Hölder exponent. Then, the real number $\alpha(t)$ given by the following expression [KGD03]:

$$\alpha(t) = \sup\{\beta \geq 0 : |X(t, t+\delta t) - X(t)| = O(|\delta t|^{\beta}) \text{ as } \delta t \to 0\} \tag{1.21}$$

is called the Hölder exponent of $X(t)$ at t. This local scaling behavior can be interpreted as the instantaneous traffic rate, i.e., the number of bytes or packets which have arrived in the interval $[t, t+\delta]$, as $\delta t \to 0$. In this case,

$$|X(t, t+\delta t) - X(t)| \sim |\delta t|^{\alpha(t)} \tag{1.22}$$

where $\alpha(t)$ is the order of the polynomial that approximates $|X(t, t+\delta t) - X(t)|$ as $|\delta t| \to 0$.

The value of the Hölder exponent $\alpha(t)$ of a process can be interpreted as follows: the intervals where $\alpha(t) < 1$ correspond to bursts of the process, while those where $\alpha(t) > 1$ show low intensity fluctuations.

The rest of the multifractal formalism will not be presented in detail here, since it is out of the scope of our work. More details can be found in [KGD03, RCRB99]. One multifractal process, the Multiplicative Generated Multifractal (also known as cascades), breaks a given set into smaller pieces according to a geometric or random rule [RW00, GWF99]. This process has been proposed as a traffic model because it mimics the fragmentation performed by the protocol stack: each trace is composed of sessions, each session consists of object requests, each request triggers TCP connections, each TCP connection is fragmented in IP packets, and finally IP packets are transported (and possibly further fragmented) by Layer 2 protocols.

1.3 Topics related to scaling traffic

This section discusses briefly several aspects of the scaling properties of traffic, such as its origins, the implications for network performance, and a short discussion of the proposed models.

1.3.1 The origins of traffic scaling

The teletraffic research community has long since advocated for physically understandable models; that is, models which not just provide a mathematical description but also try to explain what is happening in the network and what are the physical causes of self-similarity. Three main scenarios have been identified, depending whether the causes are the behavior of users (or the intrinsically fractal traffic sources), the heavy-tailedness of certain network-related variables, or the dynamics of network protocols.

Users and traffic sources

Users can generate fractal usage patterns that, when replicated in the session characteristics and in the packet generation, can induce scaling properties at the packet level. The same happens when

a traffic source is intrinsically fractal, such as VBR video; in this case, we have what Park and Willinger call *single source causality* [PW00]. Some of the most relevant works whose results belong to this category are:

- **FTP and Telnet**: It has been shown that wide-area TCP traffic packet interarrivals are not always exponentially distributed [PF95]. This study explored the characteristics of user-initiated TCP session arrivals of several network services. Remote-login and file transfer session arrivals are the only ones that are well modeled by Poisson processes (with time-dependent arrival rates), while the rest of services exhibit severe deviations from the exponential case: Telnet arrivals are much burstier than the exponential model and preserves burstiness across scales; and FTP data connections within FTP data sessions come in *connection bursts*, the largest of them being the dominant of the whole FTP traffic.

- **HTTP, World Wide Web**: Web users seem to follow heavy-tailed usage patterns [CB97]; actually, the way a WWW user behaves is intrinsically bursty at various scales, due to the transitions between page download and inactivity (reading time), that translate into connections and packet bursts that inherit fractality.

- **Variable Bit Rate (VBR) video**: MPEG video, for example, exhibits variability at multiple time scales related to the variability found in the time duration between successive scene changes [BSTW95]. The authors of [GW94] hypothesize that digital compressed TV and cinema possess an intrinsical scaling property due to the distribution of the scene durations.

- **Peer-to-peer (P2P) file sharing**: peer-to-peer file sharing systems (such as Kazaa, Napster or eDonkey) have recently been found to be much burstier than web traffic and to be well modeled by self-similar processes. In [JLC04] the authors study the P2P traffic generated by the residential users of an ADSL access operator, and quantify the scaling properties, finding that the Hurst parameter of P2P is above 0.9, while WWW traffic is only slightly under that number; and propose FGN and the aggregation of Pareto-distributed on/off sources as models. Another study [Mar02] shows that the Gnutella file sharing system exhibits high burstiness across scales. For small intervals ranging from one to ten seconds, the traffic is very bursty and varies in 2 and even 3 orders of magnitude, and in one and five minute intervals burstiness is maintained, though not as intensively. In both studies, the scaling properties of traffic seem to be linked with the behavior of individual users, that can be modeled as on/off Pareto sources.

Heavy-tails

The second cause of fractality, also called *structural causality* by Park et al. [PKC96a], is related to an empirical property inherent in distributed systems: the heavy-tailedness of files stored in disk systems. Heavy-tailed distributions have been reported in UNIX disk systems [PKC96a] and in web objects [CB97], among others. The transfer of these objects through networks generates heavy-tailed bursts and connections, and its multiplexing at the network core generates self-similarity. The

latter is one of the fundamental results of modern traffic modeling, and was shown by Willinger et al. [WTSW97], who proved that the superposition of a large number of independent on/off sources with heavy-tailed on or off periods leads to self-similarity and long-range dependence in the aggregated process. We remark that this model is able to induce both *Fractional Gaussian Noise* (see Section 1.2.6), as a result of the central limit theorem, and a form of self-similarity called *asymptotic second-order self-similarity* (see Section 1.2.1), which constitute two of the most commonly used self-similar traffic models in performance analysis.

The connection between higher-layer (source heavy-tailedness) and lower-layer descriptions of scaling resides in the fact that the application-layer property of heavy-tailed file sizes is preserved by the protocol stack and mapped to approximate heavy-tailed busy periods at the network layer. This phenomenon was shown to hold true for a variety of transport layer protocols, such as TCP in its different flavors (Tahoe, Reno and Vegas, to mention some of them) [PKC96a].

Protocols

Finally, protocol dynamics have been identified as sources or propitiators of fractality. Peha [Peh97b, Peh97a] pioneered the study of the influence of protocols in the statistical characteristics of traffic; i.e., protocols *shape* traffic and can induce scaling behavior in it.

Network traffic shows long bursts of packets followed by long silences, and this property is found at several time scales (packets can be aggregated in bursts, bursts in connections, connections in user sessions). The fragmentation induced by the protocol stack may be the cause of traffic multifractality, due to its similitude to cascades and multiplicative processes. The congestion control algorithms implemented in TCP also seem to affect the short time structure of traffic and induce multifractality [FGHW99] or at least some degree of scaling [JD05].

1.3.2 Implications for network performance

Queueing analysis

The analysis of queuing behavior with long-range dependent input exhibits a completely different behavior from corresponding systems with Markovian input. In particular, the queue length distribution of an infinite buffer system with scaling traffic decays much more slowly than the exponential, which is the rate at which it decreases with short-range dependent input, as shown by Norros [Nor94] (see Figure 2 in the Introduction). LRD input can give rise to *Weibullian* or *polynomial* tail behavior of the queue length distribution. In these cases, the long-used resource provisioning rule of increasing buffer length incurs a disproportionate penalty in queuing delay.

Some authors advocate the truncating effect of finite (and short) buffers [HL96] that destroy the memory of the process and therefore its long-range dependence; but this is only possible if we assume a certain degree of losses, and may not work if retransmission mechanisms such as those of

TCP repeat the lost packets, incurring in higher delays. Other studies propose a simple resource provisioning approach based on the combination of small buffers and large bandwidth [PM96].

One of the major weaknesses of the queuing analysis results available is that they are *asymptotic* in the sense of buffer capacity: either the queue is assumed to be infinite and asymptotic bounds of the tail of the queue length distribution are derived, or the queue is assumed to be finite but its overflow probability is computed for the buffer capacity approaching infinity. Few empirical studies have tried to bridge the gap between asymptotic analysis and real testbeds. A further drawback is that most queuing results obtained for long-range dependent input are for open-loop systems that ignore feedback control issues presented in actual networking environments (*e.g.*, TCP). Feedback can influence and shape notably the traffic that arrives at the queue, and TCP has been identified as a strong traffic shaper [JD05].

Regarding multifractal traffic, Dang et al. [DMM03] provided analytical results for the case of a single server infinite capacity queue with a constant service rate that are coherent with those found by Norros for monofractal traffic. The authors concluded that multifractal processes behave even worse (in the sense of queue length) than monofractal input processes.

Resource provisioning

The works on queuing analysis with self-similar inputs have a direct bearing on the resource dimensioning problem. The statistical characterization of the multiplexing of self-similar flows is an important question, since it can open the way to a new set of *rules of thumb* for network operators.

The questions answered with the help of these techniques are:

- quantitatively estimating the marginal utility of a unit of additional resource such as bandwidth or buffer capacity;
- pointing towards how resources can be utilized efficiently when shared across multiple flows (statistical multiplexing).

Several contributions in the scientific literature have attacked this question, but we will provide only some pointers. Gibbens [Gib96] provided an excellent overview of the topic and redefined the concept of *effective bandwidth* in the scaling context. Fan and Georganas [FG95] proved that when two comparable flows with different Hurst parameters are merged, the one with the higher H dominates the mix, though the effect is modulated depending on the traffic volume balance. Finally, open-loop resource provisioning relies strongly on the queueing analysis results, which state the uselessness of the strategy of allocating a buffer capacity proportional to bandwidth, and the strong influence of correlation when finite buffer capacity is provisioned [GB99].

Dimensioning

Regarding systems with feedback, also known as *closed-loop control*, some work has been done in the field of *multiple time scale congestion control*, exploiting the correlation structure that exists

across multiple time scales in self-similar traffic for congestion control purposes. Though the scaling properties of traffic impact negatively the performance, a positive aspect is that long-range dependence and its inherent predictability paves the way for utilizing correlation at large time scales for guiding congestion control actions. The problem of designing such a control mechanism is a nontrivial technical challenge for two reasons:

- the correlation structure in question exists at time scales typically in an order of magnitude above that of the feedback loop;
- the information extracted is necessarily imprecise, due to its probabilistic nature.

Some contributions have explored these topics. Papagiannaki et al. [PTZD05] successfully applied LRD properties in order to predict the long-term evolution of Internet backbone traffic measured in an operator. Other works have focused on connection duration prediction, where it has been found that the duration of flows, sessions or connections follow heavy-tailed distributions. Actually, most connections are short-lived but the bulk of traffic is contributed by few long-lived flows [PKC96a]. This heavy tailedness implies *predictability*: a connection whose measured time duration exceeds a certain threshold is more likely to persist in the future.

1.3.3 Discussion on the models

Though it is clear that Poisson-based models fail to incorporate the intrinsic burstiness of traffic sources at several scales, it is still under discussion whether (and which of) the models proposed in the literature provide a good characterization of the complex processes found in real traces.

This is an ill-posed problem, since Internet traffic is evolving constantly, not only in volume but also in structure; the kind of applications used in the first years of the 1990s, when self-similarity was first described, had nothing to do with the predominant position of web traffic seen later, and in the 2000s it seems that peer-to-peer and file sharing applications are responsible for the biggest share of bytes transferred. What is more, Internet infrastructure is growing constantly due to the demand from existing and new users, which makes it even more difficult to compare past results and new traces.

Just to give an overview of the latest tendencies in traffic modeling, we will comment on some of the newest contributions, in which it seems that network researchers are advocating for simpler, more physical-rooted models. It has recently been shown by Karagiannis et al. [KMFB04] that modern aggregated wide-area traffic can be well modeled by Poisson processes at sub-second scales, while piecewise-linear non-stationarity and long-range dependence seem to fit well at higher scales. Veitch et al. [VHA05] consider that the physical evidence for multifractality in TCP traffic is weak, and provide a case against multifractality in which multifractal behavior can be seen as *"a misinterpretation due to a lack of power in statistical methodology"*. Another interesting topic is the multiple scaling regimes found in some traces; in these cases, different scaling parameters can be identified for certain scale ranges. For example, Jiang and Dovrolis [JD05] recently found that

some TCP mechanisms can induce scaling, but only in a certain time scale range, usually under the round-trip time (RTT). Finally, there is a growing interest in the paper of non-stationarity in the statistical properties of traffic. Uhlig [Uhl04] studied the high-order scaling in TCP flow arrivals by performing non-stationarity checks and using the wavelet transform.

The scaling paradigm is firmly established as the foundation of modern teletraffic modeling, though it is still unclear which specific model is the most appropriate. Given the complexity of Internet traffic, probably no model –either known or yet to be devised– will ever be found to be definitive. However, more research is needed on this topic. Clearly there are important network performance issues, and protocols and algorithms should be aware of the scaling properties of traffic. Some steps have been taken in this direction, with the development of a fractal-aware version of TCP congestion control [HGHP02], the effective bandwidth estimators for LRD traffic described in [YTJ01] and [ST01] and the novel resource-control algorithm for IP over WDM networks described in [EC05]. A promising field for research is the development of estimation algorithms that can adapt to the non-stationarity of traffic, which is a challenging task.

1.4 Summary of the chapter

Several studies have proven that network traffic exhibits scaling properties that cannot be captured by traffic models based on Poissonian or Markovian stochastic processes. Traffic fractality has been detected in almost every network service, including FTP, P2P, HTTP and video-over-IP. The implications for network performance are serious: higher delays and loss ratios at router queues than have been predicted by Poissonian models. However, the correlation induced by scaling opens the opportunity for new fractal-aware mechanisms such as traffic predictors, congestion control and resource-provisioning algorithms.

The inherent non-stationary nature of Internet traffic makes traffic characterization more difficult and highlights the need for algorithms capable of detecting and adapting to the evolution of traffic. Starting from the assumption of LRD with piecewise-stationary Hurst parameter (the mFGN model described in Section 1.2.7) as a good model for network traffic, we have developed wavelet-based algorithms capable of segmenting the traffic into regions with a homogeneous variance structure. In the ideal case, we will be able to find the evolution of the scaling parameters of traffic across time.

Chapter 2

Wavelet transforms

2.1 Introduction

This chapter provides an overview of the wavelet analysis, starting with a historical perspective and following with the mathematical foundations involved. Two *classical* wavelet transforms will be presented: the Continuous Wavelet Transform (CWT) and its scale- and location-discretized version, the Discrete Wavelet Transform (DWT), with special emphasis on the latter and its associated Multiresolution Analysis (MRA) capabilities.

Apart from the CWT and DWT other redundant, non-orthogonal variants will also be analyzed: the Wavelet Packet Transform (WPT), the Maximal Overlap Discrete Wavelet Transform (MODWT), the Maximal Overlap Discrete Wavelet Packet Transform (MODWPT), the Dual Tree (Complex) Wavelet Transform (DTWT). As we will see in Chapter 3 this redundancy will allow us to enhance the methods for the analysis of the scaling parameters of network traffic.

2.2 A short history of wavelets

From a historical point of view, the *wavelet analysis* can be considered a *new* method, since it was developed mainly in the 1980s, though its mathematical foundations rely on the work of Joseph Fourier in the 19^{th} century. Fourier developed the concept of frequency or spectrum analysis, the foundation on which signal processing was built.

2.2.1 Fourier Transform

The Fourier Transform is one of the most important tools used in signal processing. It consists of the product of the signal to be studied, $x(t)$, and a set of complex exponentials of different frequencies $e^{-i2\pi ft} = \cos(2\pi ft) - i\sin(2\pi ft)$, where $i = \sqrt{-1}$ and $-\infty \leq f \leq +\infty$ is the frequency at which

the analysis is performed. The expressions for the continuous Fourier Transform and its inverse are given by equations 2.1 and 2.2:

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-i2\pi f t} dt \tag{2.1}$$

$$x(t) = \int_{-\infty}^{+\infty} X(f) e^{i2\pi f t} df \tag{2.2}$$

The complex exponential is the *eigenvector* of the frequency response of a filter; i.e., is the function that passes through a linear system without being modified (except a gain factor, the *eigenvalue*). Therefore the Fourier transform expresses the original signal $x(t)$ in terms of the weights of the different frequencies present at the signal. That is why $x(t)$ is said to be *mapped* from the *time domain* to the *frequency domain*.

There is an implicit limitation in the Fourier Transform that affects the practical application of the spectral analysis: the fact that the complex exponential basis of the Fourier Transform is defined in the entire $-\infty < t < +\infty$ support. This prevents the Fourier Transform from accurately determining *when* a particular event takes place in the original signal. For example, the position of a single peak added to a sinusoid can be completely masked in the spectrum, since it will appear as a high-frequency noise, but nothing can be said about its position in time. Another example is a signal composed of a single tone that changes its frequency abruptly from f_0 to f_1 at time t_0; the resulting spectrum will be a mix of both tones, but we will not be able to determine when the change took place. Figure 2.1 illustrates the limitations of Fourier analysis in the aforementioned cases. On the top it shows the time representation of a single tone that changes its frequency from 50 to 100 Hz, together with its Power Spectral Density (PSD), from which no information of *when* the transition happens can be extracted. In the middle row, there is the plot of a signal composed of two simultaneous tones of 50 and 120 Hz and its PSD, with two clear peaks at the corresponding frequencies. On the bottom, the bitonal signal includes a peak with value 5 in the middle of the time series. The PSD of the new time series is practically indistinguishable from the PSD of the pure bitonal. The DWT does not *see* the high-frequency peak.

2.2.2 The Short-time Fourier Transform (STFT)

Gabor tried to adapt the Fourier Transform and make it able to analyze just a small part of the signal at a time, through the use of a *windowing* technique. This involves isolating sections of the signal, as if a window was applied over the original support. Gabor's adaptation is known as *Short-Time Fourier Transform (STFT)*, and its outcome is a mapping of the original signal into a two-dimensional function of time and frequency, which is called *time-frequency plane*.

The STFT relies on a compromise between the accuracy at which the analysis can be done in both dimensions: if the window is short, time precision is enhanced at the cost of losing spectral

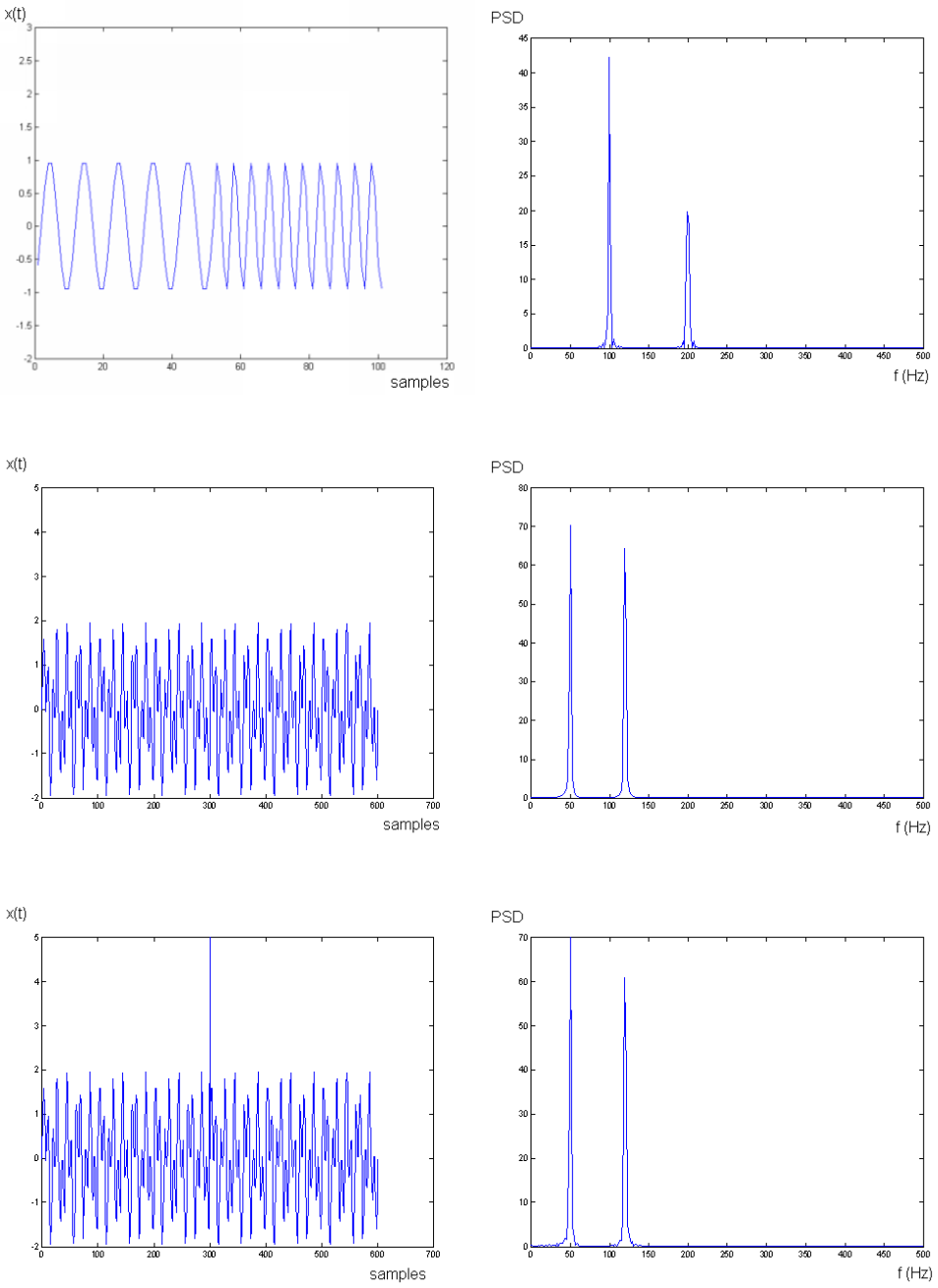

Figure 2.1: *Top left*: A tone that changes its frequency from 50 to 100 Hz. *Top right*: Power spectral density (PSD) of the frequency-changing tone. *Middle left*: Two simultaneous tones of 50 and 120 Hz. *Middle right*: PSD of the bitonal signal. *Bottom left*: Addition of a peak to the bitonal signal. *Bottom right*: PSD of the bitonal + peak signal.

accuracy (i.e., we *know* better when an event occurs but cannot say precisely which are the frequencies involved), and vice versa. This causes the time-frequency plane to be *tiled* in regions whose dimensions are defined by the accuracy or dispersion in time and frequency.

The trade-off between time resolution and frequency accuracy is similar to that of Heisenberg's Uncertainty Principle in quantum mechanics, which imposes a constraint on the precision with which we can measure the position of a particle and its momentum. This is a well-known natural law that cannot be avoided by technological (or algorithmical) means. In its signal processing equivalent, the principle states that, given a time-frequency tiling and an uncertainty or *dispersion* in time Δt and its counterpart in frequency Δf, the product of both dispersions cannot be less than a certain constant K:

$$\Delta t \Delta f \geq K \tag{2.3}$$

Therefore, if we want to measure precisely the frequency components of a signal, we will not be able to detect events in time, and the contrary. Figure 2.2 (bottom left) illustrates the time-frequency trade-off of the Gabor Transform and its associated tiling.

Though the STFT has been used for years, it was clear that a better tool was needed, capable of flexibly adapting the accuracy trade-off to the characteristics of the signal.

2.2.3 The Wavelet Transform

In the first years of the 1980s the attention of signal processing researchers gradually turned from *frequency-based* analysis to *scale-based* analysis when it started to become clear that an approach measuring average fluctuations at different scales might prove less sensitive to noise, boundary artifacts, and other effects. On the other hand, an alternative to the windowed Fourier Transform was needed in order to provide a more flexible time-frequency plane tiling.

Wavelets provide an answer to the shortcomings of the STFT. The concept of frequency is abandoned in favor of the concept of *scale*. Signals can be analyzed at different time scales, as if a zoom was performed on them. The higher the scale, the lower the frequency components of the signal analyzed at that scale (*zoom out*), and vice versa. Essentially, the wavelet transform is a windowing technique with variable sizes: it uses longer time intervals where more precise low-frequency information is required, and shorter lengths where the high-frequency components of the signal are of interest. This gives rise to a non-uniform time-frequency tiling, where the time and frequency dispersions Δt and Δf change with the scale (frequency), though the uncertainty principle still holds: improving the resolution in time implies a loss of accuracy in frequency, and vice versa.

Figure 2.2 compares the time-frequency tiling of the Wavelet Transform with those produced by Gabor and Fourier transforms. It can be seen that while Gabor Transform tiles the plane in a regular form, with constant dispersion in both axis, the Wavelet Transform chooses a different trade-off between Δt and Δf at each scale.

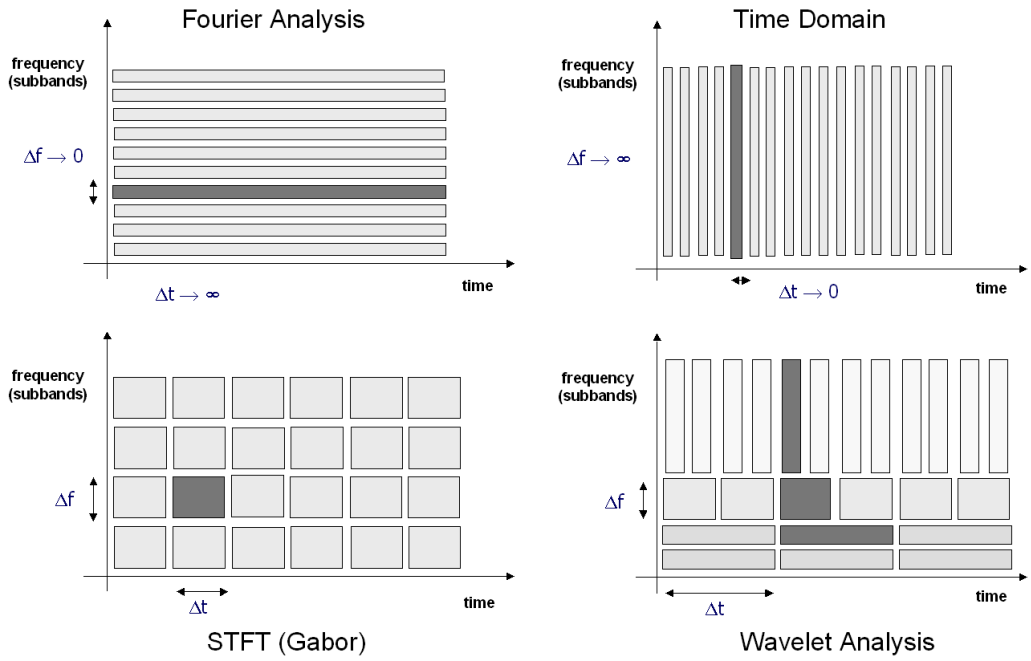

Figure 2.2: Time-frequency tilings, with the trade-off between time dispersion Δt and frequency location dispersion Δf, for the Fourier Transform, the Shannon analysis (the usual time-based representation of signals), the Gabor Transform and the Discrete Wavelet Transform.

Alfred Haar was the first mathematician to use what we now call a *wavelet*. In 1909 Haar used for the first time a well-localized oscillating function in order to perform time-frequency analysis. Though their techniques were known in the signal processing community, it was not until 1984 that Jean Morlet and Alex Grossmann proposed the concept of wavelets in its present theoretical form. Mathematicians such as Y. Meyer, Ingrid Daubechies [Dau92], Stephane Mallat [Mal89], Ronald Coifman, and Victor Wickerhauser [CW92] have developed the methods of wavelet analysis and have ensured their dissemination. Nick Kingsbury [Kin01, SBK05] and Ivan Selesznick [Sel01, Sel04] have contributed recently to the development of alternative wavelet transforms. Wavelet analysis is probably the most promising research area in the field of signal processing.

The following sections will focus on the mathematical underpinnings of the main wavelet-based tools developed in recent years. Namely, we will discuss the following transforms:

- Continuous Wavelet Transform (CWT);
- Discrete Wavelet Transform (DWT);
- Wavelet Packet Transform (WPT);
- Maximal Overlap Discrete Wavelet Transform (MODWT);
- Maximal Overlap Discrete Wavelet Packet Transform (MODWPT);
- Dual Tree (complex) Wavelet Transform (DTWT).

Yet another wavelet transform, the Double Density Wavelet Transform (DDDWT), will not be discussed in this chapter but is described in Appendix A.

2.3 Continuous Wavelet Transform (CWT)

The Continuous Wavelet Transform (CWT) is defined as the scalar product of the original signal $x(t)$ and a set of translated and dilated versions of the so-called *mother wavelet* $\psi(t)$.

$$C_x(a,b) = \int_{\mathbb{R}} x(t)\frac{1}{\sqrt{a}}\psi\Big(\frac{t-b}{a}\Big)dt, \quad a \in \mathbb{R}^+ - 0, \quad b \in \mathbb{R} \tag{2.4}$$

Therefore, the CWT coefficients measure how the original signal resembles *locally* (in terms of time and frequency) the mother wavelet. The CWT performs a multiresolution analysis: we can see the original signal at different time scales without losing its time evolution, since the scalar product is done at all the possible locations. As aforementioned, the dispersion with which we can measure the signal in both axis is constrained.

The key to understanding the properties of the CWT is the mother wavelet $\psi(t)$. The mother wavelet is chosen so that both its spread in time and frequency are relatively limited, thus ensuring the locality of the analysis. It consists of a small wave (hence the name *wavelet*) defined on a support, which is *almost* limited in time and having most of its energy within a limited frequency band. While the time support and frequency band cannot both be finite, there is an interval in which they are *effectively* limited. This causes the mother wavelet to be a *bandpass* function. If $\psi(t)$ has a compact support (i.e., is zero outside a certain range of values of t), the wavelet analysis is well localized in time, at the cost of being not so well localized in frequency. There are mother wavelets without compact support (though their values are close to 0 outside a certain range) that enhance their frequency localization.

In addition to being not too much dispersed both in time and frequency, the mother wavelet is required to satisfy the *admissibility condition*

$$\int \psi(t)dt = 0 \tag{2.5}$$

which asks for an oscillating function, and justifies the name "wavelet".

Figure 2.3 illustrates how the shortcomings of Fourier analysis are avoided by the wavelet transform. The two cases already presented in Figure 2.1 (and the bitonal signal contaminated with a peak) are now analyzed with the CWT. In this figure the convention for representing the CWT is the brighter the color, the higher the value of the coefficient. The first case is the tone with a frequency change. It can be seen that the frequency of the first tone is centered in scale 9, while the second tone is centered in scale 4. The transition point is clearly located in the middle of the series, though it is more precisely seen in the higher frequencies (lower scales) and appears blurred at the lower frequencies (higher scales). The second case consists of two tones of 50 and 120 Hz with a peak of value 5 in the middle of the time series. Its wavelet representation shows a mix

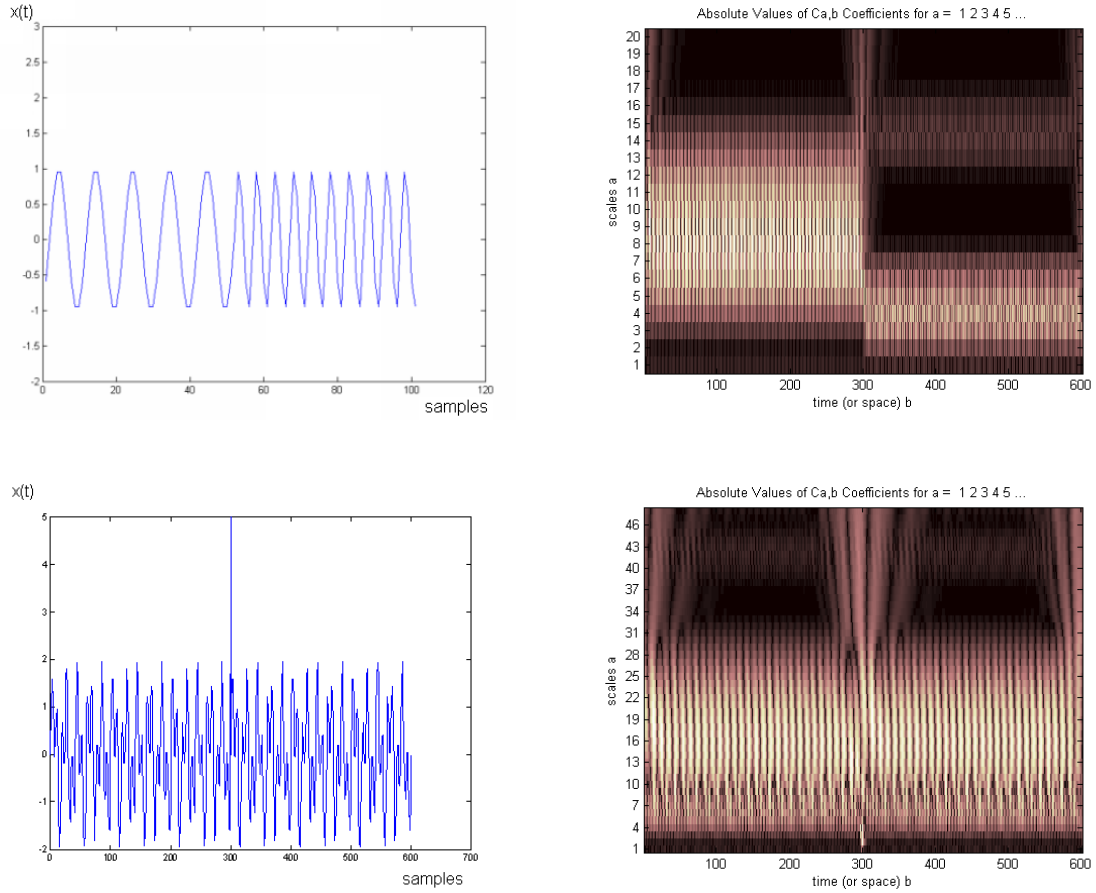

Figure 2.3: *Top left*: A single tone that changes its frequency from 50 to 120 Hz. *Top right*: The time-frequency wavelet representation of the signal, with a clear transition point in the middle. *Bottom left*: The two tones of 50 and 120 Hz with a peak of value 5 in the middle of the time series. *Bottom right*: The wavelet representation of the signal. The anomaly is still clearly located in the middle of the time series.

of both tones, whose frequencies are more difficult to resolve than in the Fourier case, due to the impairment in frequency dispersion, but the anomaly is still clearly located in the middle of the time series.

The CWT analyzes the signal at all the possible scales through the variation of the dilation parameter a, and at all time axis through the time-shifting parameter b. Its practical implementation, though, is limited to a certain range of discrete values for the scale (usually $a \in \mathbb{N}$) and the time axis (which, being the typical analyzed signal intrinsically discrete, is usually sampled at the same sample rate as the signal $x(t)$). These are the conventions used in figure 2.3. Therefore, the Continuous Wavelet Transform is usually not as continuous as one would desire. This is why a discretized, sampled version known as DWT was developed.

2.4 Discrete Wavelet Transform (DWT)

2.4.1 Definition

The Discrete Wavelet Transform (DWT) consists of the collection of coefficients

$$d_X(j,k) = \langle X, \psi_{j,k} \rangle, \quad j = 1, \ldots, J, \quad k \in \mathbb{Z} \tag{2.6}$$

that are obtained comparing (by means of scalar products) the continuous-time signal $X(t)$ to be studied with a set of analyzing functions

$$\psi_{j,k}(t) = 2^{-j/2}\psi_0(2^{-j}t - k), \; j = 1, \ldots, J, \quad k \in \mathbb{Z} \tag{2.7}$$

As the CWT, this *wavelet basis* is constructed from a reference function $\psi_0(t)$, called the *mother wavelet*, by the action of a *change of scale* operator dependent on the *scale* index j

$$\psi_{j,0}(t) = 2^{-j/2}\psi_0(2^{-j}t), \quad j = 1, \ldots, J \tag{2.8}$$

and a *time-shift* operator dependent on the *time* index k

$$\psi_{0,k}(t) = \psi_0(t - k), \quad k \in \mathbb{Z} \tag{2.9}$$

The change of scale (or dilation) operator defines the scale of time over which the original signal is observed, while the time-shift operator enables the selection of the time instant around which one wishes to analyze it.

In the case of discrete-time signals, in principle, the DWT can only be performed on signals whose length L is a power of 2, due to the dilation factor of 2 defined in Equation 2.8. There are methods for extending the length to 2^n through padding or circular shifts, for example (see [PW02]). Another constraint holds for the depth J of the analysis, that obviously has to verify $L \geq 2^J$. Actually the usual situation is $L = 2^J$, denoted as *full DWT*; that is, the analysis is performed at the deepest level on a signal whose length is that of a power of 2.

2.4.2 Relation between CWT and DWT

The DWT is actually a discretization of the CWT, in which only certain values of a and b are analyzed: $a = 2^j, \; b = k2^j, \; j \in \mathbb{N}, \; k \in \mathbb{Z}$. When these values are substituted in Equation 2.4, Equation 2.6 is obtained.

2.4.3 Properties of the wavelet basis

In the analysis of scaling phenomena, the following two features of the wavelet basis play a key role.

Proposition 2.1. *Since the wavelet basis is constructed from the change of scale operator given in Equation 2.8, the analyzing family exhibits itself, by construction, a scale-invariance feature.*

Figure 2.4: DWT subband decomposition ($J = 3$).

As we saw in Chapter 1, the scaling properties of traffic can be understood as the absence of any characteristic frequency (and therefore, scale) in the range of frequencies close to the origin. This property can thus be interpreted as a scale invariance characteristic which is efficiently analyzed by wavelets, as will be shown in Chapter 3

Proposition 2.2. *$\psi_0(t)$ has a number N of vanishing moments which can be freely chosen, provided $N \geq 1$. By definition, this means that*

$$\int_{-\infty}^{+\infty} t^k \psi_0(t) dt = 0, \quad k = 0, 1, 2, \ldots, N-1 \tag{2.10}$$

or equivalently, that the Fourier transform of $\psi_0(t)$ satisfies

$$|\Psi_0(f)| = O(|f|^N), \quad f \to 0 \tag{2.11}$$

This property can be used to control divergences arising with processes having power-law spectra at the origin. For $k = 0$, Equation (2.10) gives the aforementioned *admissibility condition*

$$\int_{-\infty}^{+\infty} \psi_0(u)\ du = 0 \tag{2.12}$$

which confirms that the mother wavelet is a bandpass (or oscillating) function. An important consequence of the admissibility condition is that the wavelet transform is "blind" to nonzero mean values of the original process $X(t)$, that is:

$$E\{d_X(j,k)\} = 0, \quad j = 1, \ldots, J \tag{2.13}$$

A natural generalization consists of considering wavelets with more than one vanishing moment, in the sense of Equation (2.10), since assuming that the number of vanishing moments is N allows the analysis to be blind to polynomials of order up to $N - 1$. In other words, the removal of a *polynomial trend* of order p is guaranteed by a wavelet with $N \geq p + 1$.

2.4.4 Multiresolution Analysis (MRA)

Subband decomposition

In the frequency domain the result of the DWT is a decomposition of the original signal $X(t)$ in subbands whose bandwidths are halved at each scale increase, due to the factor of 2 in the dilation procedure). Figure 2.4 shows such a subband decomposition in the $J = 3$ case.

This gives rise to a *Multiresolution Analysis* (MRA) [Dau92] in which the original signal $X(t)$ is decomposed in a low-frequency *approximation* at scale J, $A_J(t)$, and a set of *details* $D_j(t)$ for each scale $j = 1, \ldots, J$. The details $D_j(t)$ represent the changes of the original signal on a time scale $\tau_j = 2^{(j-1)}$, while the approximation $A_J(t)$ represents the average of the original signal at time scale $\lambda_J = 2^J$. Note that if the length L of the original signal $X(t)$ is such that $L = 2^J$, that is, the analysis is performed at the deepest level, then $A_J(t)$ is composed by just a coefficient that corresponds to the *mean* of the original signal.

In the time domain, the MRA consists of rewriting the information contained in the original signal as a sum of approximations $A_j(t)$ and details $D_j(t)$ of $X(t)$.

$$X(t) = A_J(t) + \sum_{j=1}^{J} D_j(t) \tag{2.14}$$

Note that the j^{th} detail of $X(t)$ corresponds to the information that is removed when going from one approximation level to the next, coarser one: $D_j(t) = A_{j-1}(t) - A_j(t)$. Therefore, equation 2.14 can be rewritten as

$$\begin{aligned} X(t) &= D_1(t) + A_1(t) \\ &= D_1(t) + D_2(t) + A_2(t) \\ &= D_1(t) + D_2(t) + D_3(t) + A_3(t) \\ &\ldots \\ &= \sum_{j=1}^{J} D_j(t) + A_J(t) \end{aligned}$$

Which is the relation between the components of the MRA and the wavelet coefficients? The details D_j are easy to understand from what we already know of wavelet analysis: they are the *reconstruction* of the analysis performed by the DWT, and can be obtained by the sum of the dilates-and-translated wavelet mother $\psi(t)$, weighted by the wavelet coefficients,

$$D_j(t) = \sum_k d_X(j,k)\psi_{j,k}(t) \tag{2.15}$$

Similarly, the approximations A_j are the weighted sum of dilated and translated versions of what is called the *scaling function*, denoted by $\phi(t)$. The scaling function is a *low-pass* function derived from the mother wavelet, and its role in the computation of the approximation coefficients is similar to that of the mother wavelet in the computation of the detail coefficients. Then, the j^{th} coarse or low-pass approximation of the original signal can be computed as

$$A_j(t) = \sum_k a_X(j,k)\phi_{j,k}(t) \tag{2.16}$$

where

$$\phi_{j,k}(t) = 2^{-j/2}\phi_0(2^{-j}t - k), \quad j = 1, \ldots, J, \quad k \in \mathbb{Z} \tag{2.17}$$

and the approximation coefficients $a_X(j,k)$ at scale j are defined by

$$a_X(j,k) = \langle X, \phi_{j,k} \rangle, \quad j = 1, \ldots, J, \quad k \in \mathbb{Z} \tag{2.18}$$

Equation 2.14 can then be rewritten as the *Inverse Discrete Wavelet Transform*

$$X(t) = \sum_k a_X(J,k)\phi_{J,k}(t) + \sum_{j=1}^{J} \sum_k d_X(j,k)\psi_{j,k}(t) \tag{2.19}$$

Therefore, the wavelet transform actually provides several approximations and details of the original signal through the comparison of $X(t)$ with the high-pass wavelet mother and the low-pass scaling function, respectively. The weights obtained in these scalar products are the detail coefficients $d_X(j,k)$ and approximation coefficients $a_X(J,k)$.

It is important to clearly differentiate the roles of the wavelet coefficients $d_X(j,k)$ and $a_X(j,k)$, and the reconstructed functions $D_j(t)$ and $A_j(t)$. The former are the representation of the signal in the wavelet domain, and depend on the scale j and the (discretized) time k, while the latter are time functions defined in the whole original support of $X(t)$ that represent the low- or high-frequency components of the original signal.

Figures 2.5 and 2.6 illustrate the multiresolution analysis performed by the DWT. The first figure shows wavelet and approximation coefficients of the bitonal + peak signal. Note that the number of coefficients diminishes with scale. At lower scales of the detail coefficients (higher frequency) the peak can be clearly identified, while at higher detail scales and the approximation (both corresponding to lower frequencies) the peak disappears. Figure 2.6 shows the details and approximations, which are time-aligned with the original signal. The higher-frequency details show the peak even more clearly than the correspondent wavelet detail coefficients, while the approximations smooth the peak, as expected.

Analysis of variance

Given the MRA equation for a DWT decomposition at level J , $X(t) = \sum_{j=1}^{J} D_j(t) + A_J(t)$, and thanks to the orthogonality of the transform, the energy preserving condition holds:

$$\|X\|^2 = \int X^2(t)dt = \|A_J\|^2 + \sum_{j=1}^{J} \|D_j\|^2 = \sum_k a_X(J,k)^2 + \sum_{j=1}^{J} \sum_k d_X(j,k)^2 \tag{2.20}$$

where $\|D_j\|^2$ represents the contribution to the energy of $X(t)$ due to fluctuations at scale j, and $\|A_j\|^2$ is the energy of the J^{th} coarse approximation of $X(t)$ [PW02]. Due to the orthogonality of the DWT, $\|D_j\|^2 = \sum_k d_X(j,k)^2$.

If the DWT analysis is performed at the deepest level $J = \log_2(L)$ where L is the length of the signal, and noting that in the case of Daubechies wavelets it can be shown [PW02] that $a_X(J,k) = a_X(J) = E[X]/\sqrt{L}$, an analysis of variance (ANOVA) can be done on a scale-by-scale basis:

$$\hat{\sigma}_X^2 = \frac{1}{L}\|X\|^2 - \hat{X}^2 = \frac{1}{L}\sum_{j=1}^{J} \|D_j\|^2 = \frac{1}{L}\sum_{j=1}^{J} \sum_k d_X(j,k)^2 \tag{2.21}$$

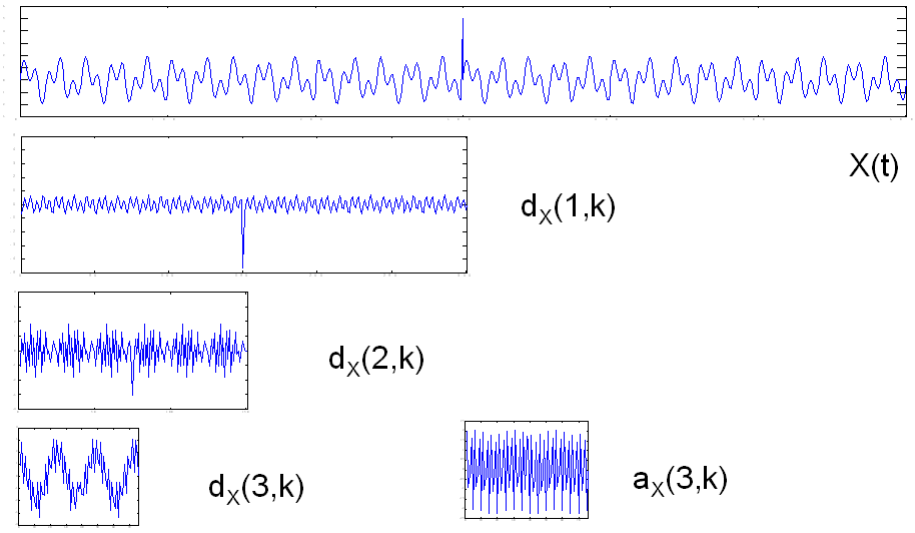

Figure 2.5: Wavelet coefficients of the 601-sample signal composed by two tones of 50 Hz and 120 Hz with a peak of value 5 at position 300. From top to bottom: the original signal, the wavelet detail coefficients $d_X(j,k)$ at levels $j = 1, 2, 3$, and the approximation coefficients $a_X(3,k)$.

where $\hat{\sigma}_X^2$ is the sample variance of $X(t)$. This result will allow us to perform ANOVAs directly on the DWT coefficients of scaling processes, instead of having to reconstruct the approximation $A_J(t)$ and details $D_j(t)$ of the signal.

The decomposition of sample variance defined in 2.21 can be used to define a *wavelet empirical spectrum* $P_W(\tau_J)$:

$$P_W(\tau_j) \equiv \frac{1}{L} \sum_k d_X(j,k)^2 \tag{2.22}$$

for which

$$\sum_{j=1}^{J} P_W(\tau_j) = \hat{\sigma}_X^2 \tag{2.23}$$

2.4.5 Wavelet families

There are different types of wavelet families, whose qualities vary according to several criteria. The most important are:

- the support of $\psi_0(t)$, that is, its speed of convergence to zero when time t goes to infinity;
- the number of vanishing moments;
- the regularity, which is useful for getting beneficial features, like smoothness of the reconstructed signal, and for the estimated function in nonlinear regression analysis;
- the orthogonality (or biorthogonality) of the resulting analysis;
- the existence of an explicit expression.

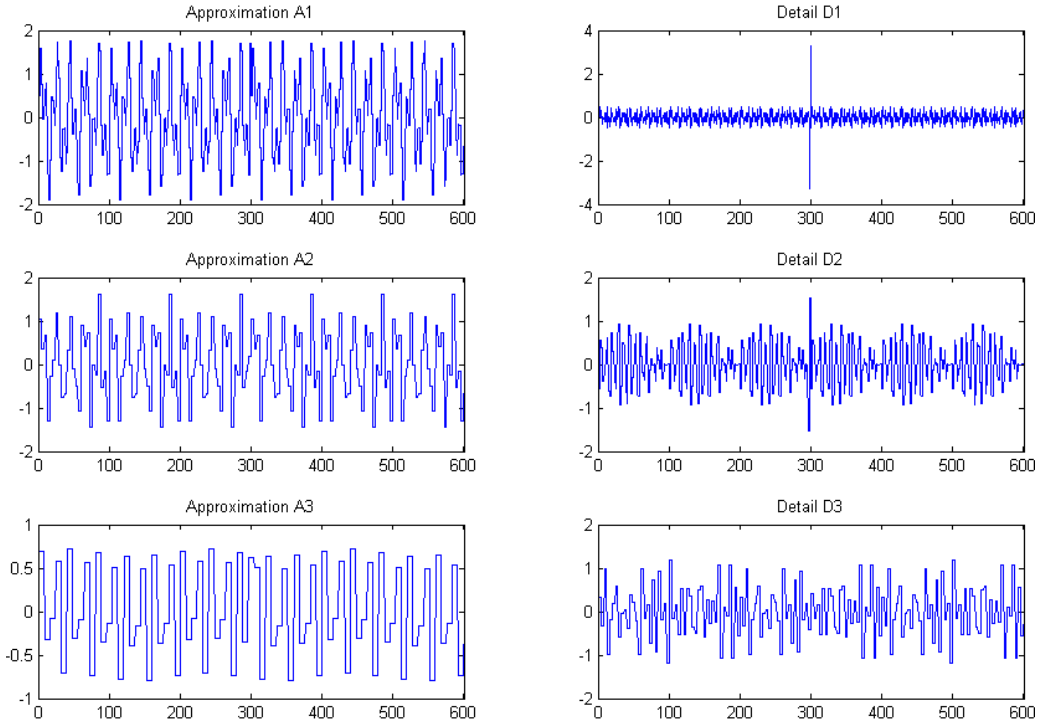

Figure 2.6: Multiresolution analysis of the bitonal + peak signal, with the details $D_j(t)$ and approximations $A_j(t)$ for $j = 1, 2, 3$.

Wavelets that are often used in practice include the Haar wavelet, the Daubechies wavelets, the Symlets, the biorthogonal wavelets, the Coiflets and the Meyer wavelets, among others. In the following, we will focus on the first four of these families.

The Haar wavelet

Any discussion about wavelets begins with the Haar wavelet (Figure 2.7, top left), the first and simplest. The Haar wavelet is discontinuous, and resembles a step function. Haar wavelet is also given the name Daubechies `db1`, whose expression is

$$\psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

while the corresponding scaling function $\phi(t)$ is given by

$$\phi(t) = 1 \quad 0 \leq t < 1 \tag{2.25}$$

Daubechies wavelets

Ingrid Daubechies [Dau92] developed the so-called *compactly supported orthonormal wavelets*, thus making discrete wavelet analysis practicable. Daubechies' wavelets are also known as *extremal phase* wavelets due to the properties of their transfer function, since they produce a *minimum delay* filter.

The wavelets of the Daubechies family are usually referred to as `dbN`, where N is the order, and db the "surname" of the wavelet. The `db1` wavelet, as mentioned above, is the same as the Haar wavelet. Figure 2.7 shows on the top the `db4` (middle) and `db8` (right) wavelet functions.

Symlets or Least Asymmetrical wavelets

Symlets are called after *symmetrical wavelets*, though they are nearly but not exactly symmetrical, and that is why they are also known as *least asymmetrical (LA) wavelets*. These are compactly supported wavelets with the highest number of vanishing moments for a given support width. In the usual notation `symN`, N is the order. When LA notation is used, `symN` corresponds to LA(2N), where $2N$ is the length of the associated wavelet filter. Figure 2.7 shows on the bottom the `sym1` (left) `sym4` (middle) and `sym8` (right) wavelet functions.

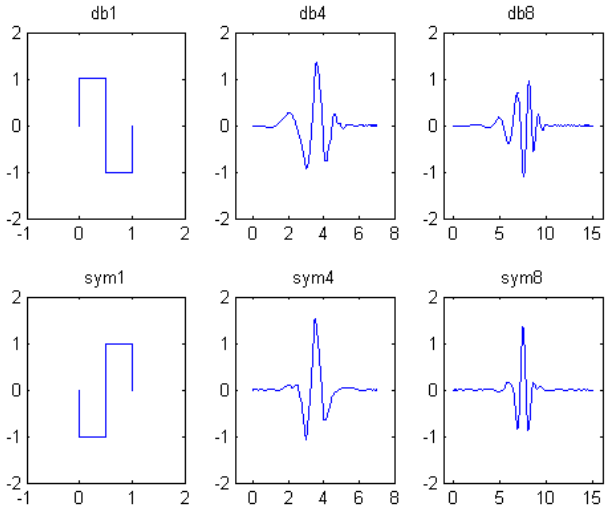

Figure 2.7: Example of Daubechies and Symlets wavelets.

The main property of this family is the near-linear phase of its filters, when studied in terms of the frequency response. The phase function of the least asymmetric filters generated by the LA wavelets have the smallest maximum deviation in frequency from the best fitting linear phase function [PW02]. This is a nice feature if we are interested in a precise localization of some event in time, since it allows us to meaningfully relate DWT coefficients to the original time axis. Daubechies `dbN` wavelets do not have this property, while the phase shift introduced by Symlets can be easily corrected.

The phase properties of the Symlets depend on their order N. We can obtain approximate zero phase wavelet filters by advancing the filter output by $|2N-1+\nu|$ samples, where

$$\nu = \begin{cases} -N+1, & \text{if } N = 4, 6, 8 \text{ or } 10 \text{ (i.e., N is even);} \\ -N & \text{if } N = 5 \text{ or } 9; \\ -N+2 & \text{if } N = 14. \end{cases} \tag{2.26}$$

In particular, if the original signal $X(n)$ of length L is associated with the actual time $t_0 + n\Delta t$ (where t_0 is an offset and Δt is the sampling period), then wavelet coefficient $d_X(j,k)$ from the symN or LA(2N) analysis is associated with actual time

$$t_0 + (2^j(n+1) - 1 - |\nu_j^H| \mod L)\Delta t, \quad n = 0, \ldots, \frac{L}{2^j} - 1 \tag{2.27}$$

where

$$|\nu_j^H| = \frac{(2^j - 1)(2N-1)}{2} + N + \nu - 1 \tag{2.28}$$

More details of the phase properties of the different wavelet families, together with the derivation of Equations 2.26 and 2.27 can be found in [PW02].

Biorthogonal wavelets

The properties of symmetry and exact reconstruction are impossible to achieve simultaneously with FIR filters. A solution is the use of two wavelets instead of just one, where the first one, $\tilde{\psi}(t)$, is used in the analysis

$$C_X(j,k) = \int X(t)\tilde{\psi}_{j,k}(t)dt \tag{2.29}$$

and the second one, $\psi(t)$, in the synthesis.

$$X(t) = \sum_{j,k} C_X(j,k)\psi_{j,k}(t) \tag{2.30}$$

This way *"the useful properties for analysis (e.g., oscillations, null moments) can be concentrated in the $\tilde{\psi}$ function; whereas the interesting properties for synthesis (regularity) are assigned to the ψ function"* [Coh92]. The two wavelets verify the biorthogonality conditions 2.31 and 2.32:

$$\int \tilde{\psi}_{j,k}(t)\psi_{j',k'}(t)dt = 0 \quad \text{for } j \neq j' \text{ or } k \neq k' \tag{2.31}$$

$$\int \tilde{\phi}_{0,k}(t)\phi_{0,k'}(t)dt = 0 \quad \text{for } k \neq k' \tag{2.32}$$

As we will see in Section 2.8 for the DTWT, some wavelet transforms use biorthogonal wavelets such as Antonini or LeGall families.

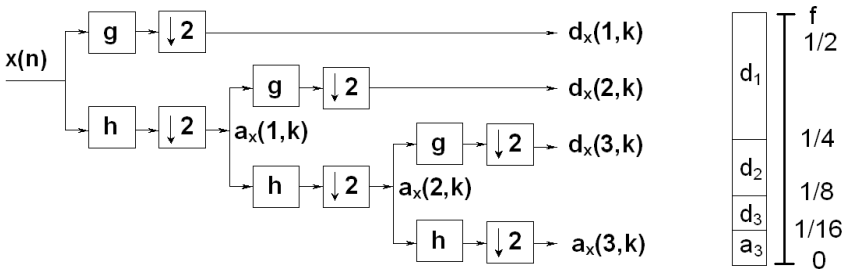

Figure 2.8: Filter bank implementation of the DWT. *Left*: fast filter-bank-based pyramidal algorithm. *Right*: the corresponding subband decomposition

2.4.6 Fast pyramidal algorithm

In 1988 Mallat [Mal89] developed a fast wavelet decomposition algorithm with a linear computational cost $O(N)$ for data of length N, making it very suitable for the analysis of very large data sets. The Mallat algorithm uses a recursive filter-bank-based *pyramidal algorithm*, sketched on Figure 2.8. This classical scheme in the signal processing community is also known as a *two channel subband coding using quadrature mirror filters (QMF).*

Quadrature Mirror Filters

Two filters $a(k)$ and $b(k)$ are said to be QMF if their frequency responses $A(f)$ and $B(f)$ are such that their squared gain functions are mirror images of each other:

$$|A(f)|^2 = |B(\frac{1}{2} - f)|^2 \tag{2.33}$$

The main property of QMF filters is that they cover completely and uniformly the normalized spectrum:

$$|A(f)|^2 + |B(f)|^2 = 2, \quad \forall f \tag{2.34}$$

Therefore, if one of the filters has a low-pass squared gain function, the other one will act as a high-pass. It can be shown that the impulse responses $a(k)$ and $b(k)$ of two QMF filters of length L satisfy

$$a(k) = (-1)^{k+1} b(L - 1 - k), \quad \forall k \tag{2.35}$$

and the inverse relationship

$$b(k) = (-1)^{k} a(L - 1 - k), \quad \forall k \tag{2.36}$$

Mallat's algorithm

The main idea of the algorithm, coming from the nested structure of MRA, consists of the fact that the $d_X(j,k)$ and the $a_X(j,k)$ can be computed through a discrete-time convolution involving the sequence $a_X(j-1,k)$ and two discrete-time QMF filters $h(k)$ (the low-pass *scaling* filter) and

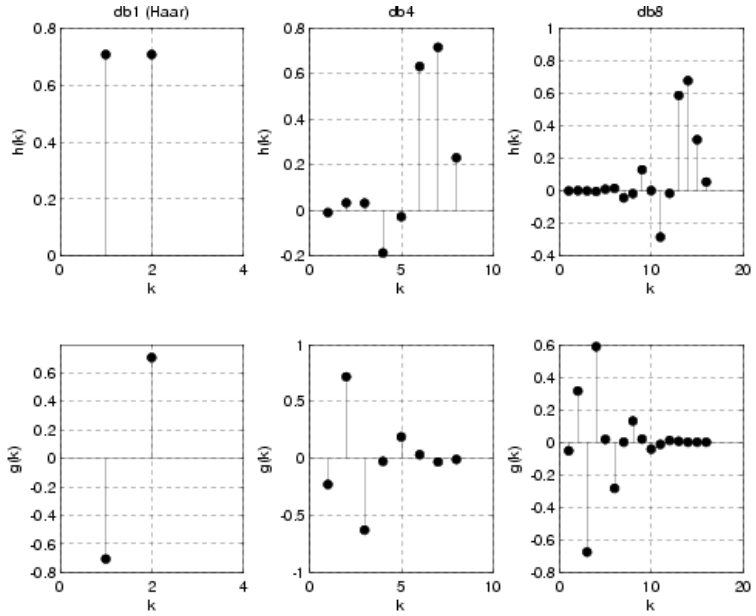

Figure 2.9: The two decomposition filters for Daubechies 1, 4 and 8 wavelets. *Left*: `db1` (Haar). *Middle*: `db4`. *Right*: `db8`.

$g(k)$ (the high-pass *wavelet* filter). The process is then iteratively repeated in the low-pass output J times. Figure 2.9 shows the wavelet and scaling filters for `db1`, `db4` and `db8`.

The output of the filters is downsampled by 2 in order to maintain orthogonality. Therefore, the number of coefficients *approximately* (due to border effects) halves with each increase in j:

$$n_{j+1} \approx \frac{n_j}{2} \tag{2.37}$$

Wavelet and scaling filters

The wavelet filter $g(k)$ is derived from $\psi(t)$, and must satisfy some properties: zero mean, unit energy, and must be orthogonal to even shifts. While the first condition is ensured by the wavelet admissibility function, the latter two conditions are called the *orthonormality property* of wavelet filters.

The scaling filter $h(k)$ is derived from the wavelet filter and the QMF property (Equations 2.35 and 2.36). Therefore, the scaling function $\phi(t)$ is derived from the wavelet mother $\psi(t)$ through the QMF property.

Let's compute the filters for the Haar wavelet. Its wavelet filter is given by $g(k) = [1/\sqrt{2}, -1/\sqrt{2}]$, while its scaling filter is given by $h(k) = [1/\sqrt{2}, 1/\sqrt{2}]$. The shape of both filters is actually the same as the mother wavelet and scaling function, with a normalization factor required by the unit energy condition. Figure 2.10 shows the squared gain functions of the Haar scaling and wavelet filters, where the QMF property can be clearly appreciated. The Haar filters are not specially good

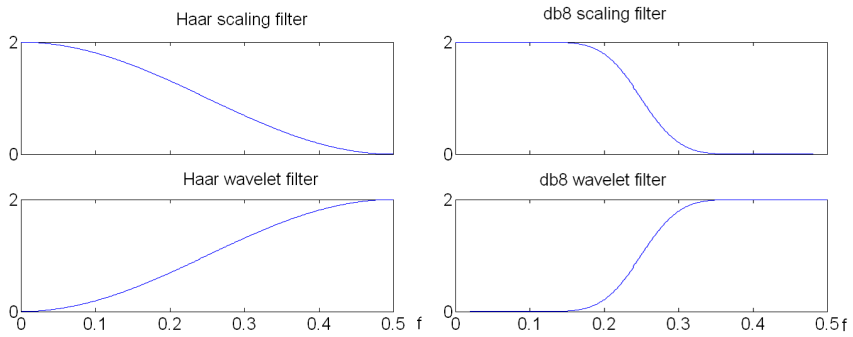

Figure 2.10: Squared gain functions of the `db1` (Haar) and `db8` scaling and wavelet filters.

in terms of its selectivity; the ideal QMF filters should be abrupt. Higher order wavelets, such as `db8` show better frequency response, as shown in the same figure.

The frequency response of each subband of the DWT decomposition is derived from the composition of the filter bank. The compound filters and the associated subbands are depicted in Figure 2.11 for the Haar and `db8` wavelets. The subbands generated by `db8` have more abrupt frequency responses.

What is the relationship between the filters and the mother wavelet and scaling function? The mother wavelet can be obtained by an iterative procedure:

1) Take the wavelet filter $g(k)$.
2) Upsample the filter by 2.
3) Convolve the upsampled filter with the scaling filter $h(k)$.
4) Iterate steps 2-3.

The scaling function can be computed following a similar procedure, iteratively upsampling and convolving the low-pass filter $h(k)$. Figure 2.12 illustrates the procedure for obtaining the wavelet mother and scaling function for `db2`.

2.4.7 MRA initialization for discrete-time data

Several authors [QF99, AF94, VTA00] point out the inaccuracy of the usual application of the DWT to discrete-time signals. The *discrete* character of the DWT comes from the sampling of certain values of the dilation and location parameters of the CWT, and does not involve a discrete-time analysis.

The use of the discrete-time algorithm for computing the continuous-time scalar product $d_X(j,k) = \langle X, \psi_{j,k} \rangle$ requires an *initialization procedure* if the data $X(n)$ is intrinsically time-discrete. Without this prefiltering, the DWT and the subsequent estimation of features of the signal (the scaling parameters, for example) results in errors. Veitch, Taqqu and Abry show in [VTA00] that the lack of initialization can bias the LogScale Diagram.

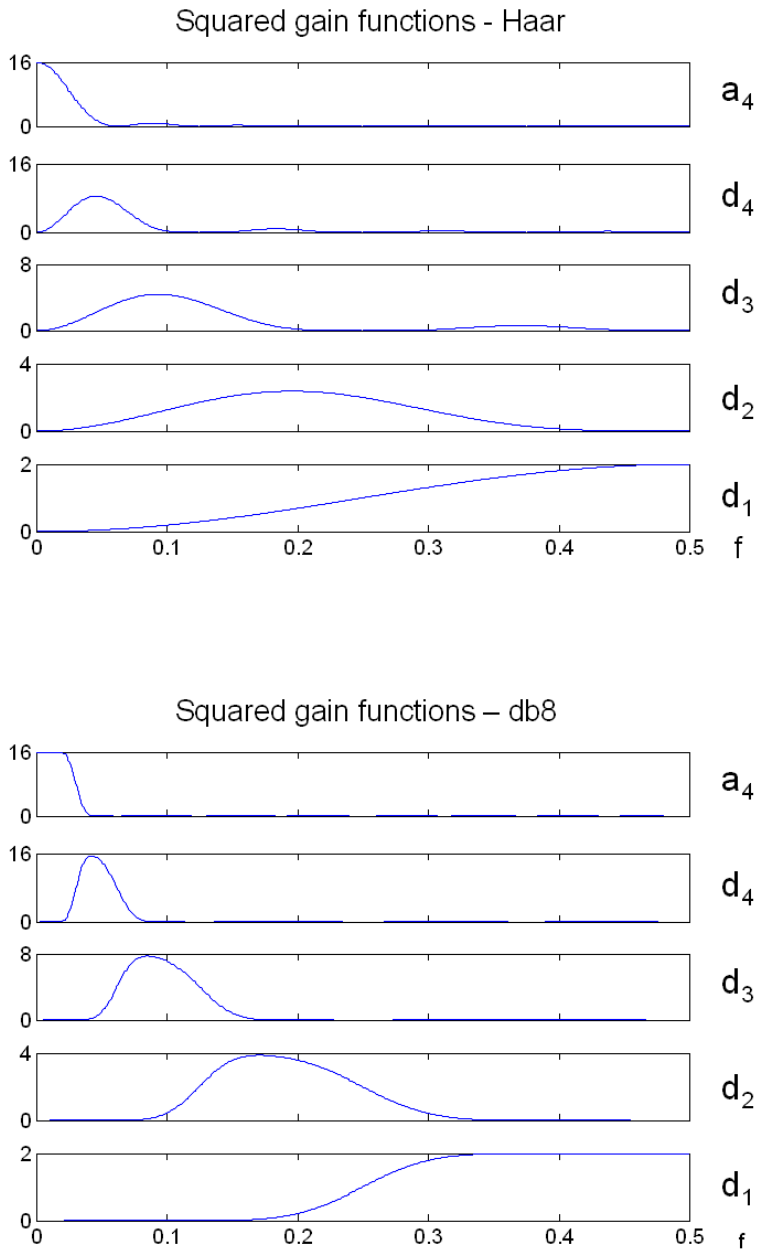

Figure 2.11: *Top*: The squared gain functions of the Haar 4-level filter bank. *Bottom*: The squared gain functions of the `db8` 4-level filter bank.

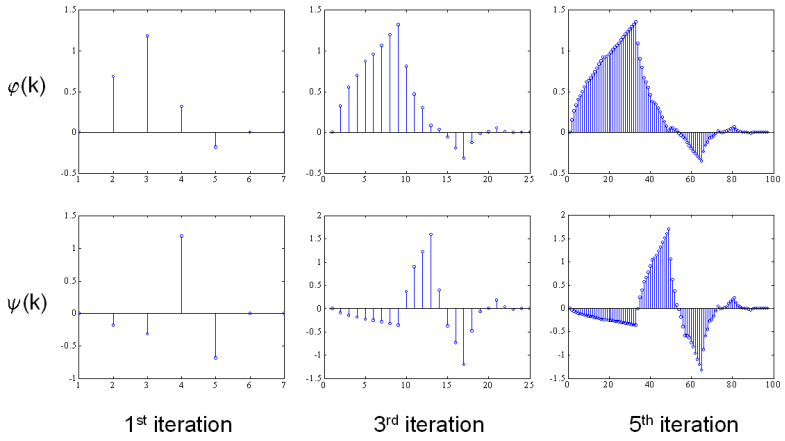

Figure 2.12: First, third and fifth iterations of the procedure for obtaining the scaling function (top) and wavelet mother (bottom) for the `db2` wavelet.

The idea of the initialization procedure is very simple. To the discrete-time signal $X(n)$ we associate a closely related continuous-time equivalent signal $\overline{X}(t)$, chosen such that the power spectral density of both signals are identical in the interval $-1/2 < f < 1/2$. Being the spectra identical, any conclusion on the spectrum of $\overline{X}(t)$ automatically holds for $X(n)$.

$\overline{X}(t)$ is chosen as a sum of sinc funtions by the Shannon Sampling Theorem,

$$\overline{X}(t) = \sum_{n=-\infty}^{\infty} X(n) sinc(t-n) = \sum_{n=-\infty}^{\infty} X(n) \frac{sin(\pi(t-n))}{(t-n)\pi} \tag{2.38}$$

since this ensures that $\overline{X}$ and X coincide at integer times:

$$\{\overline{X}(k), k \in \mathbb{Z}\} = \{X(k), k \in \mathbb{Z}\} \tag{2.39}$$

Then, the multiresolution analysis can be initialized with

$$\begin{aligned} a_{\overline{X}}(k) &= \int_{-\infty}^{\infty} \overline{X}(t)\phi(t-k)dt \\ &= \sum_{n=-\infty}^{\infty} X(n) \int_{-\infty}^{\infty} \phi(t-k) sinc(t-n) dt \\ &= \sum_{n=-\infty}^{\infty} X(n) I(k-n) \\ &= (X * I)(n) \end{aligned}$$

where $*$ denotes discrete-time convolution, and

$$I(m) = \int_{-\infty}^{\infty} \phi(t) sinc(t+m) dt \tag{2.40}$$

depends only on the scaling function $\phi(t)$ associated to the wavelet transform. This initialization technique has been used in our DWT-based analysis.

2.5 Wavelet Packet Transform (WPT)

2.5.1 Introduction

As discussed in Section 2.4.1, each scale of the DWT represents a frequency range half as wide as the scale below it. Conversely, the time resolution on each scale is half that of the scale below it, due to the decimation step. These characteristics pose some problems:

- at the lower scales, discrimination of higher frequencies is sacrificed for time localization in the transform;
- at the higher scales, vice versa, time localization in the transform is sacrificed for discrimination of lower frequencies.

We can overcome these problems by introducing the Wavelet Packet Transform (WPT) [CW90] (also cited as Discrete Wavelet Packet Transform (DWPT)), a generalization of the DWT decomposition that offers a more versatile signal analysis. The WPT provides an *overcomplete* expansion of the original signal, in the sense that it gives more information than is strictly necessary in order to reconstruct the original signal; i.e., is a *redundant* wavelet transform, also called a *frame* [Sel01]. If, as in the case of the WPT, the energy preserving condition (Parseval's theorem) holds, the frame is said to be *tight*.

2.5.2 Filter bank implementation

In the DWT algorithm the LP/HP filter decomposition is not applied to the details $d_X(j,k)$, which are not further processed. When we do elaborate upon them in the same way as the approximations $a_X(j,k)$, a tree of possible decompositions grows, as depicted in Figure 2.13. The DWT algorithm is just one branch of this tree.

Compared to the DWT algorithm, the WPT gives more flexibility. Instead of zooming in on lower and lower frequencies (thus making only one subband decomposition possible), for a given depth level J it generates a set of 2^J possible subband decompositions (DWT being a particular case), as shown in Figure 2.14 for the $J = 2$ case.

The logical consequence of the above is that, in addition to j for *scale* and k for *time*, the WPT coefficients possess an additional index m, corresponding to the *subband number*. Since the distinction between approximations and details no longer makes sense, we will refer to the WPT coefficients by the notation $W_X(j,k,m)$.

Figure 2.15 illustrates the projection of such a decomposition in the scale-frequency plane. As we can easily check from this figure, subbands with $m = 1$ constitute the DWT decomposition. Figure 2.17 shows the squared gain function of the filter bank for a level 3 decomposition, when using the `sym4` or LA(8) wavelet. Note that the "ideal" pass-bands, marked with vertical lines, have a length of $1/8^{th}$ of the spectrum.

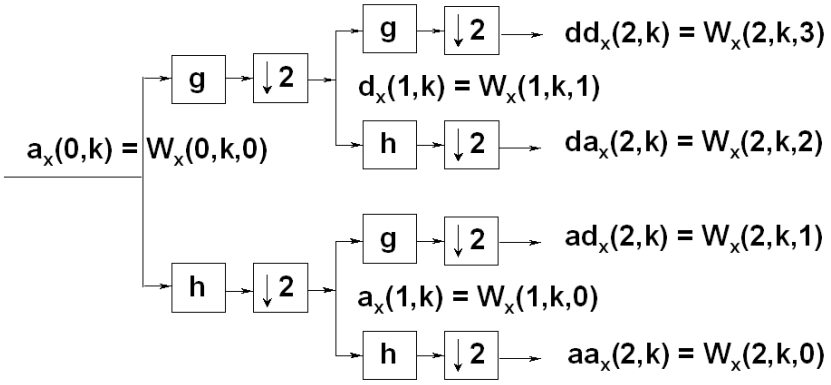

Figure 2.13: Filter bank implementation of the WPT decomposition at level $J = 2$.

2.5.3 Basis selection and best basis algorithm

From the point of view of providing a basis for the data, the full (m, j) tree contains redundancy and no longer represents an orthonormal basis as does the DWT algorithm. To obtain an orthonormal basis, one has to select boxes from the decomposition of Figure 2.15 in such a way that the frequency axis is covered horizontally without overlapping. As previously reported, choosing $m = 1$ is one of the possibilities that leads to this result. On the other hand, there are several m values that correspond to an incomplete decomposition (for example, $m = 2$), and some others to a redundant one (for example, $m = 0$).

There are some algorithms for finding the *best basis*, depending on the goodness criteria. If the metric is an additive measure, the optimum is given by Coifman and Wickerhauser's best-basis algorithm [CW92], whose total computational complexity is $O(N \log_2 N)$. The algorithm consists of associating with each (m, j) subband an additive cost function $M(m, j)$, and choosing the subband decomposition that minimizes the total cost of all possible decompositions, starting from the simplest one $j = 1, m = 0 \dots 1$ and expanding the tree branch only if the sum of the costs of both leafs/subbands is less than the cost of the father leaf/subband.

One of the most popular additive cost functions, widely used in compression applications, is the $-\ell^2 \log(\ell^2)$ norm, or *entropy* of the squared coefficients:

$$m(|\overline{W}_x(j,k,m)|) = \begin{cases} -\overline{W}^2_x(j,k,m) \log_2 \overline{W}^2_x(j,k,m) & \text{if } \overline{W}_x(j,k,m) \neq 0 \\ 0 & \text{if } \overline{W}_x(j,k,m) = 0 \end{cases} \tag{2.41}$$

where $\overline{W}^2_x(j,k,m) \equiv \frac{W_x(j,k,m)}{||X||}$

A contribution to this field was made by Sola and Sallent [SSR94], who developed a fast splitting algorithm (FSA) capable of performing a progressive orthogonal tiling, and thus allowing the real-time operation of the best-basis algorithm. This is the algorithm we would like to use jointly with the WPT-based estimator described in Chapter 3.

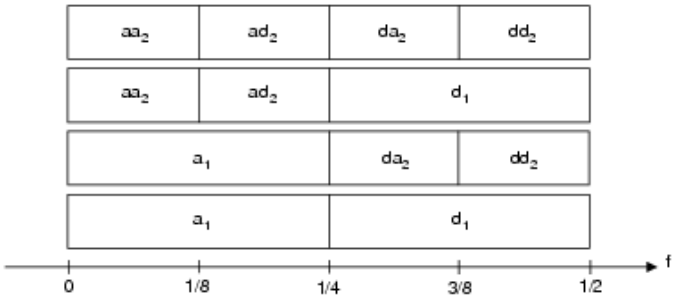

Figure 2.14: The four possible subband decompositions generated by the WPT tree ($J = 2$).

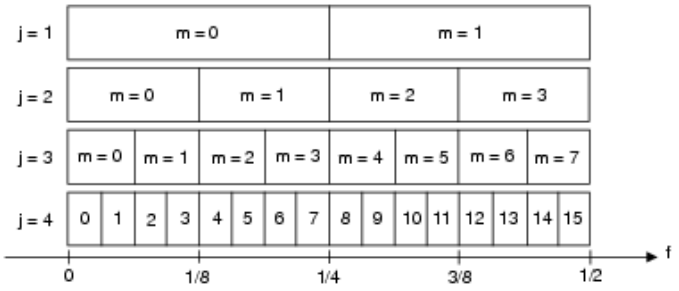

Figure 2.15: WPT subband decomposition in the (j, m) plane for $J = 3$.

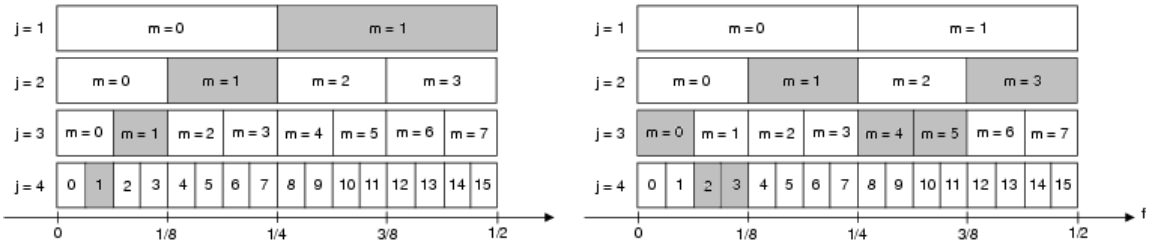

Figure 2.16: *Left*: the WPT subband decomposition for m=1, equivalent to the DWT. *Right*: a general orthogonal subband decomposition.

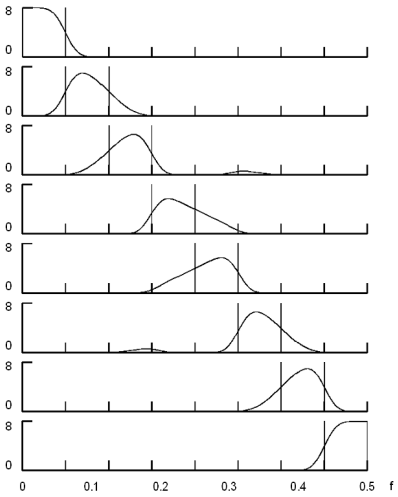

Figure 2.17: Squared gain functions for $J = 3$ and $m = 0 \dots 7$ for the `sym4` or LA(8) WPT. The ideal pass-bands are marked by vertical lines. Extracted from [PW02]. ©Cambridge University Press, reproduced with permission.

2.5.4 MRA and ANOVA

Given an orthonormal decomposition, the energy preserving condition holds. For example, for the $W_x(2, k, m), m = 0 \dots 3$ decomposition shown in Figure 2.13 we have

$$\begin{aligned}
\|X\|^2 = \sum_t X^2(t) &= \sum_{m=0}^{3} \|W_x(2,k,m)\|^2 \\
&= \|W_x(2,k,0)\|^2 + \|W_x(2,k,1)\|^2 + \|W_x(2,k,2)\|^2 + \|W_x(2,k,3)\|^2 \\
&= \sum_k |W_x(2,k,0)|^2 + \sum_k |W_x(2,k,1)|^2 + \sum_k |W_x(2,k,2)|^2 + \sum_k |W_x(2,k,3)|^2 \\
&= \sum_{m=0}^{3} \sum_k |W_x(2,k,m)|^2
\end{aligned}$$

but we could also have an alternative complete decomposition such as

$$W_x(2,k,0), \quad W_x(2,k,1), \quad W_x(1,k,1)$$

(corresponding to the DWT) and then

$$\begin{aligned}
\|X\|^2 &= \|W_x(2,k,0)\|^2 + \|W_x(2,k,1)\|^2 + \|W_x(1,k,1)\|^2 \\
&= \sum_k |W_x(2,k,0)|^2 + \sum_k |W_x(2,k,1)|^2 + \sum_k |W_x(1,k,1)|^2
\end{aligned}$$

Therefore, for orthogonal decompositions the expressions of the ANOVA are similar to those of the DWT. Non-orthogonal decompositions (either overcomplete or undercomplete) are a different case, since no energy preserving is ensured. However, in Chapter 3 we will describe an ANOVA-like analysis for the case of a general WPT decomposition, particularized for scaling traffic.

2.5.5 Alignment

In section 2.4.5 the phase properties of some wavelet families were studied. In particular, the time shifts that enable the correct alignment of events in the original signal and its counterpart in the wavelet domain were derived. The same can be done for the WPT, with the novelty of adding the new dimension m. Though it is relatively easy to compute the time shifts, the new dimension forces us to express them in a table which will not be included here. The details can be found in [PW02].

2.6 Maximal Overlap Discrete Wavelet Transform (MODWT)

2.6.1 Orthogonality of the wavelet transforms

As we discussed in the previous sections, wavelet transforms can be classified as either *orthogonal* (*nonredundant*) or *non-orthogonal redundant, overcomplete*). The DWT belongs to the first class, whereas the WPT may be defined in an orthogonal or redundant form, depending on the selected subband decomposition. The orthogonal transform is appealing for two main features:

- a better preservation of the compression ability, since only components relevant to reconstruction are included;
- the statistical decorrelation of the wavelet coefficients.

On the other hand, the major drawback of the DWT is its time- or shift-variance (i.e. the wavelet coefficients of a delayed signal are not a time shifted version of those of the original signal) [PKC96b]. The time invariance property is particularly important in statistical signal processing applications, such as parameter estimation of signals with unknown arrival time, or estimation of a parameter that changes across time (as is our case). In other words, the orthogonal DWT is not capable of tracking correctly the time evolution of the signal with the same precision at all scales, due to its particular time-frequency plane tiling scheme, as already discussed in Section 2.5.2: lower scales have more coefficients, giving more time resolution but less frequency resolution, and viceversa. The introduction of the redundant Maximal Overlap Discrete Wavelet Transform (MODWT) is a good way to overcome this problem.

Figure 2.18 gives an example of the time variance of the DWT. A 32-sample signal $x(n)$ with all samples equal to zero except $x(16) = 5$, and its shifted version $y(n)$ with $x(17) = 5$ are analyzed with the DWT, giving the MRA shown in the figure. It is obvious that the results of the analysis of the time-shifted signal are not identical to the time-shifted version of the MRA. Figure 2.19 shows the time invariance property of the MODWT. The same signals $x(n)$ and $y(n)$ are analyzed with the MODWT, whose MRA applied to the time-shifted signal is identical to the time-shifted version of the MRA (i.e., the approximations and details of $y(n)$ are shifted versions of the approximations and details of $x(n)$). All the analyses have been performed with the Haar wavelet, at level $J = 3$.

MODWT has been given a lot of different names in the literature: “Stationary Wavelet Transform (SWT)”, “Undecimated DWT”, “Shift-invariant DWT”, “Wavelet frames”, and “Time-invariant DWT”, among others.

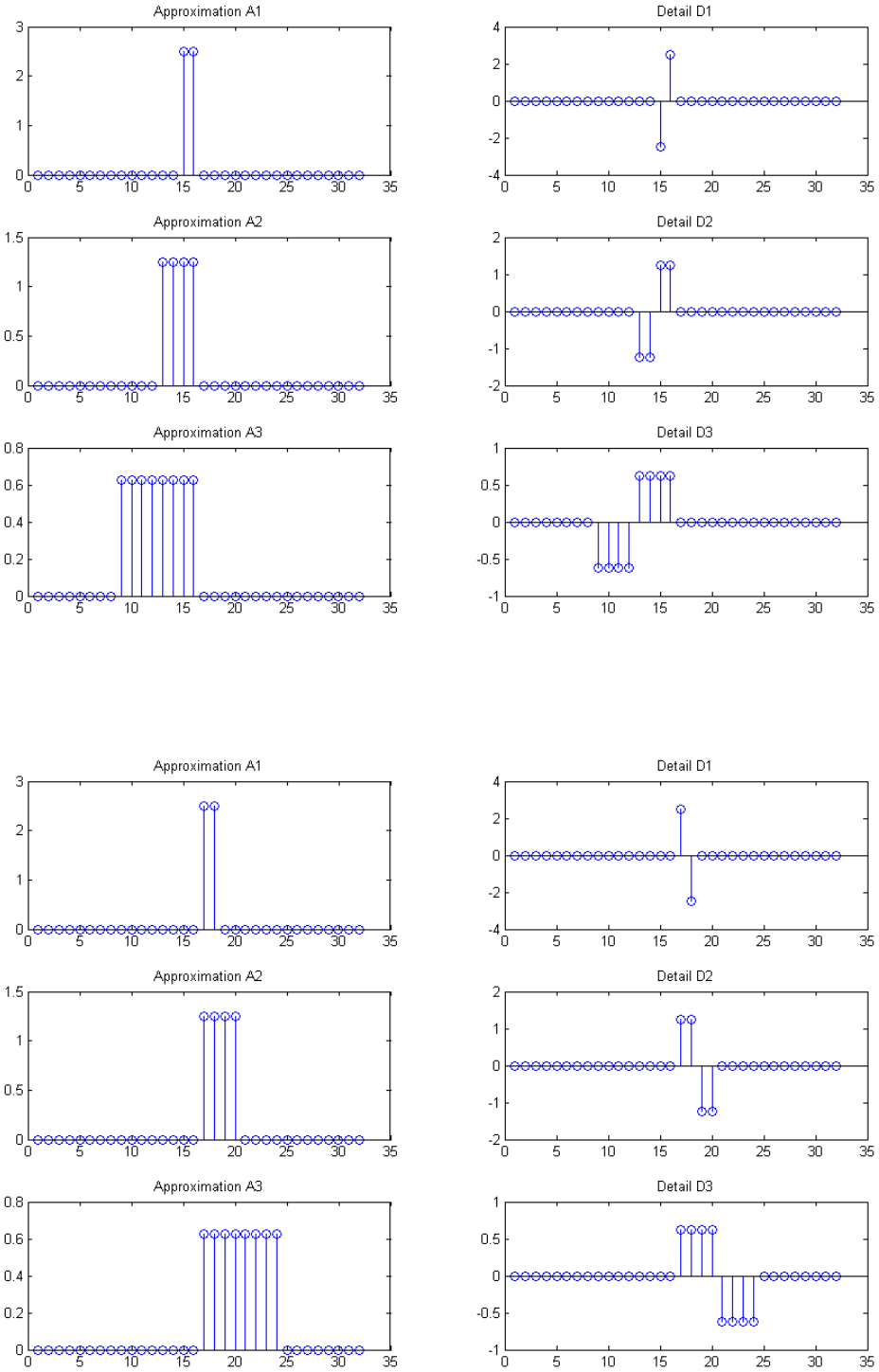

Figure 2.18: *Top*: MRA analysis of $x(n)$ with the DWT. *Bottom*: MRA analysis of $y(n)$ with the DWT.

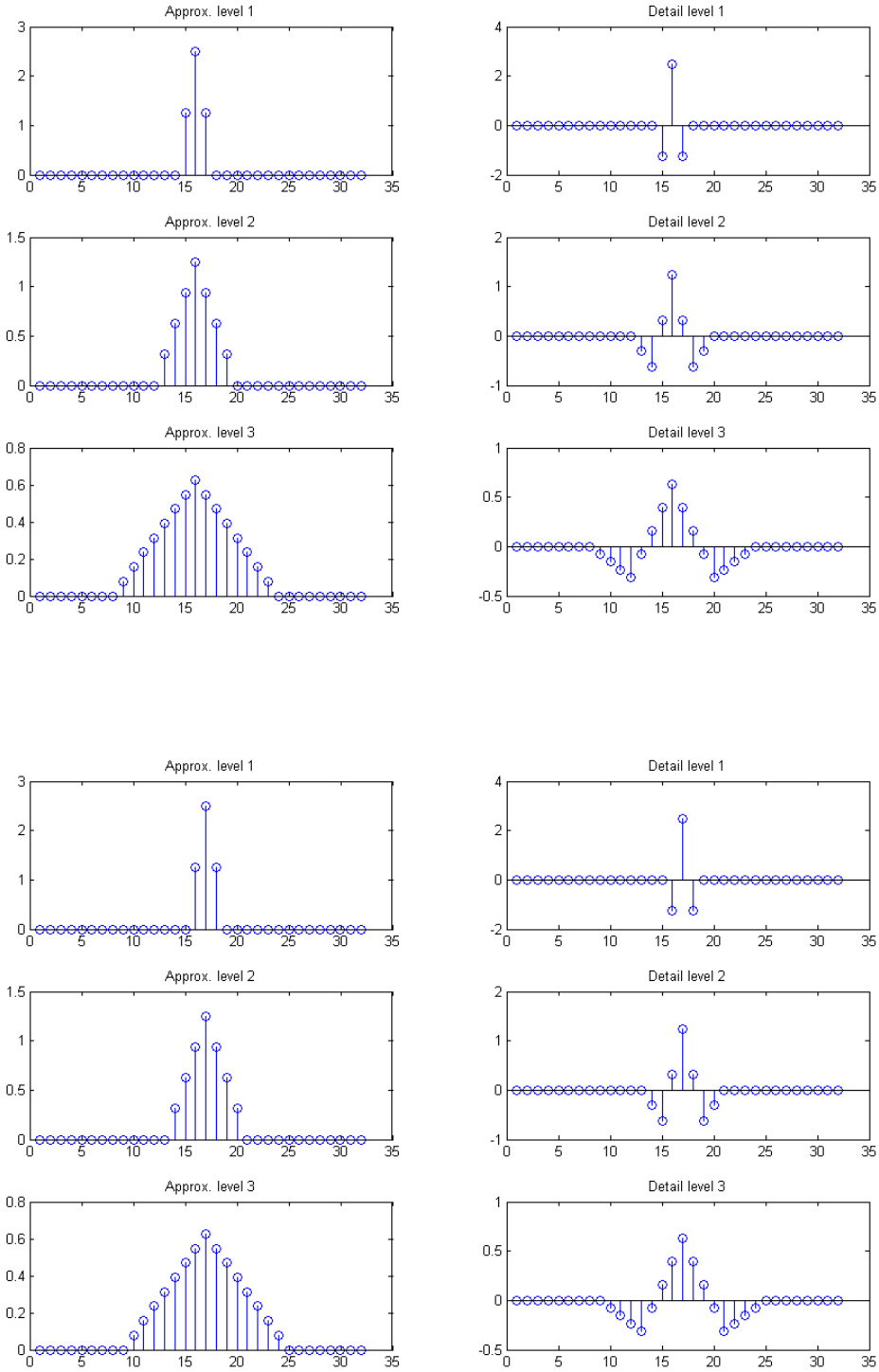

Figure 2.19: *Top*: MRA analysis of $x(n)$ with the MODWT. *Bottom*: MRA analysis of $y(n)$ with the MODWT.

2.6.2 Filter bank implementation

The MODWT [PKC96b] is obtained from the DWT by suppression of the downsampling stage, as sketched in Figure 2.20. This causes the sequences $\tilde{d}_X(j,k)$ to have the same length at all the branches of the filter bank.

The intuitive idea behind the MODWT is "to fill the gaps" caused by the downsampling step in the standard wavelet transform. This leads to an over-determined, over-complete, or redundant, representation of the original data, but one which has considerable statistical potential, as we will see in Chapter 7. Given an MODWT performed at a depth of J scales, the result is overcomplete by the factor $J+1$, i.e. we get J details and one approximation, each one with the same resolution as the original signal.

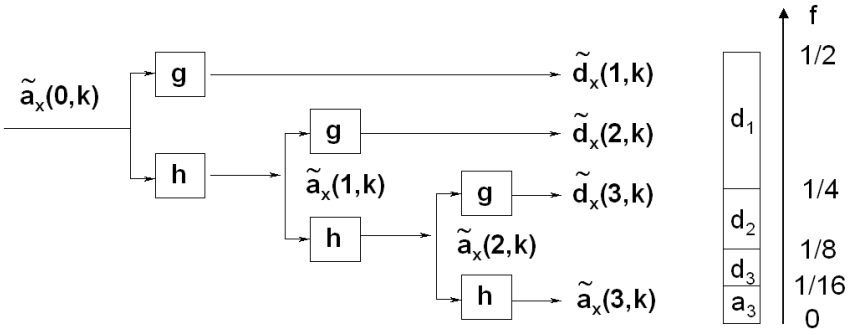

Figure 2.20: *Left*: Filter bank implementation of the MODWT decomposition at level $J = 3$. *Right*: The subband decomposition generated by the MODWT at level $J = 3$.

2.6.3 Properties

What follows is a quick review of the main properties of the MODWT:

Subband decomposition

Despite the modifications in the time domain, the frequency decomposition performed by MODWT (shown in Figure 2.20) is the same as that of the DWT.

Complexity

Regarding the computational complexity of the pyramid algorithm when applied to a signal $X(n)$ of length N, where N is a power of 2, the MODWT requires $O(N \log_2 N)$ multiplications, while the corresponding DWT can be computed in only $O(N)$ operations. The MODWT is clearly heavier to compute, but its complexity is comparable to that of the Fast Fourier Transform.

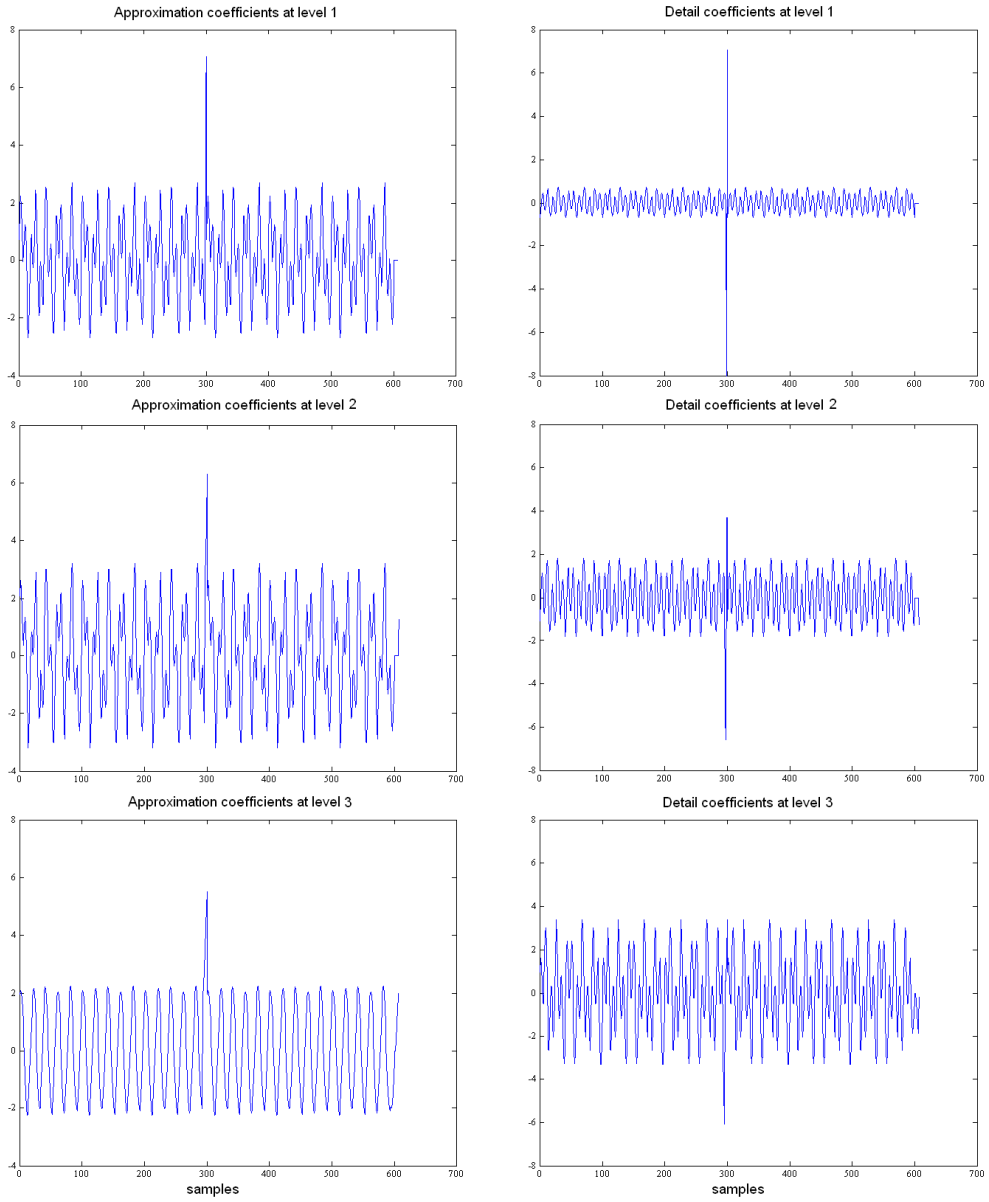

Figure 2.21: Example of the event alignment properties of the MODWT. The original signal is the 601-sample composed of two tones of 50 Hz and 120 Hz, with a peak in the middle of the sequence. The signal has been extended to 608 samples through zero-padding (this explains the border effects on the last samples). The analysis has been performed at level $J = 3$ with the Haar (`db1`) wavelet. *Left column*: approximation coefficients. *Right column*: detail coefficients.

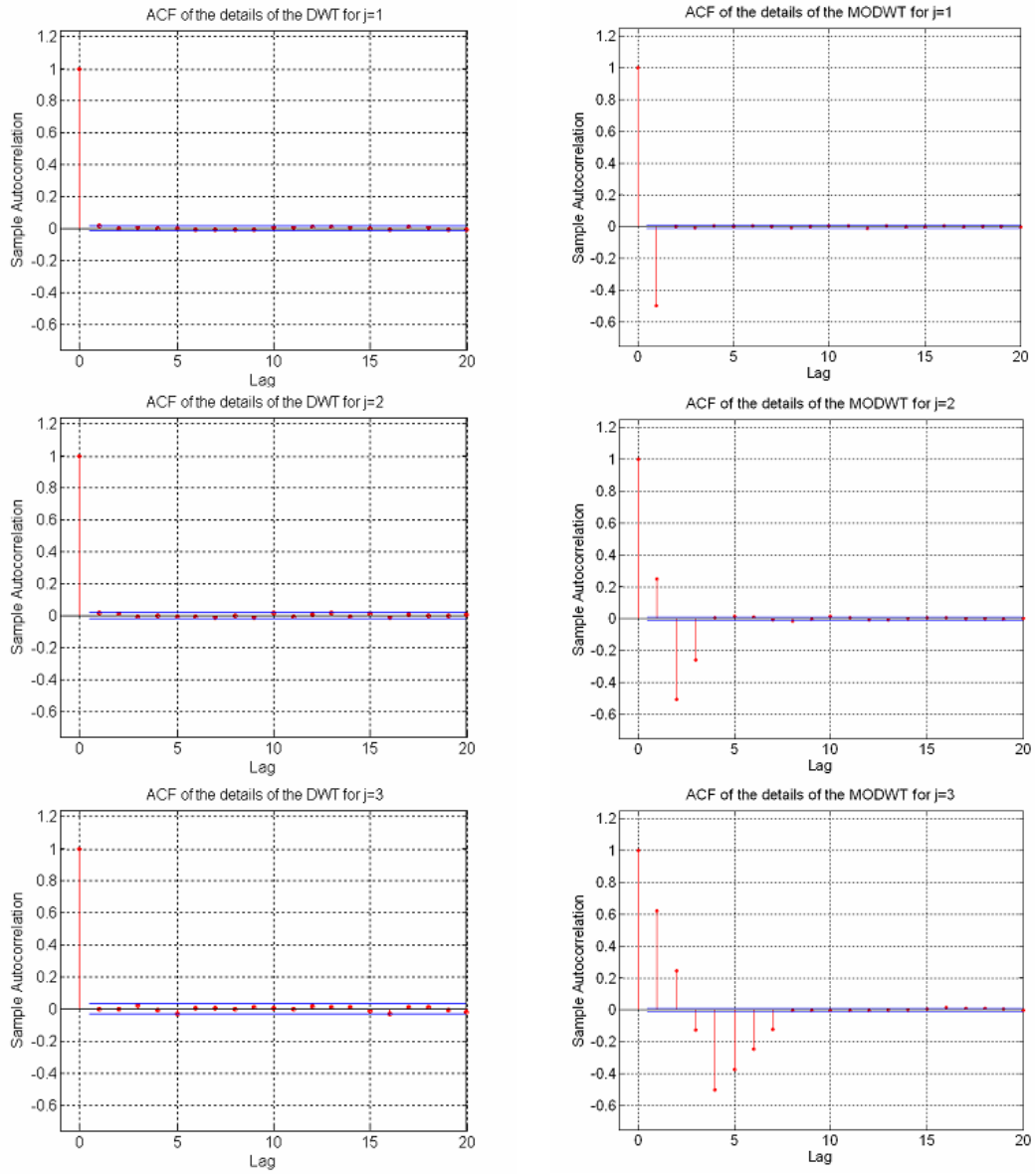

Figure 2.22: *Left column*: The autocorrelation functions for the detail coefficients of the DWT of a white noise signal. *Right column*: The autocorrelation functions for the detail coefficients of the MODWT of a white noise signal.

MRA and ANOVA

As in the case of the DWT, the MODWT can perform a multiresolution analysis. The analysis of variance can also be done with the wavelet coefficients but not with details $\tilde{D}_j$ and approximations $\tilde{A}_j$. The non-orthonormality of the MODWT is the cause of the inequality.

$$\|X\|^2 = \sum_k \tilde{a}_X(J,k)^2 + \sum_{j=1}^{J}\sum_k \tilde{d}_X(j,k)^2 \neq \|\tilde{A}_J\|^2 + \sum_{j=1}^{J}\|\tilde{D}_j\|^2 \tag{2.42}$$

Alignment

In contrast to the DWT, the MODWT output is associated with zero phase filters, making it easy to align events in the original time series with the wavelet coefficients. Figure 2.21 shows the alignment features of the MODWT.

2.6.4 The correlation issue

Being a time-redundant transform, we expect the MODWT to introduce some kind of time correlation in the analyzed signals. The decorrelation properties of the wavelet transforms will be presented in Chapter 3, but just to help the reader build some intuition, we want to emphasize that it is the orthogonality of a transform which ensures that a correlated signal in time domain is mapped into a decorrelated signal in the transformed domain. Therefore, the DWT provides uncorrelated $d_X(j,k)$ coefficients when analyzed in the "time" domain k, while the MODWT coefficients present a certain degree of correlation that increases with the scale j at which the analysis is performed. The explanation for this increase is the "averaging" of the wavelet filter bank analysis when the downsampling step is not done.

Figure 2.22 shows an example of the aforementioned properties of the transforms. An uncorrelated white noise signal of length $L = 2^{15} = 32768$ has been analyzed with both the DWT and the MODWT, using the Haar wavelet. The figure shows the autocorrelation function $\rho(m)$ for the first 25 lags, together with the 95% confidence interval (shown as horizontal lines). While the DWT coefficients are perfectly uncorrelated, MODWT coefficients are strongly correlated, and the phenomenon is intensified with scale.

As we will see in Chapters 4 and 7, the correlation properties of the wavelet transforms will be important when a change point detection algorithm is applied at their outputs, since these change detection methods expect an uncorrelated input. From this point of view the DWT is clearly better than the MODWT. On the other hand, the detection will be enhanced with the time-shift invariance properties of the MODWT, which allows us to accurately locate the position of changes in the time series. Therefore, we will face a trade-off between localization capabilities and correlation.

2.7 Maximal Overlap Discrete Wavelet Packet Transform (MODWPT)

Just as the MODWT was derived from the DWT by skipping the downsampling step of the pyramid algorithm, a Maximal Overlap version of the WPT can be defined in the same way. Figure 2.23 illustrates the filter bank implementation of the MODWPT.

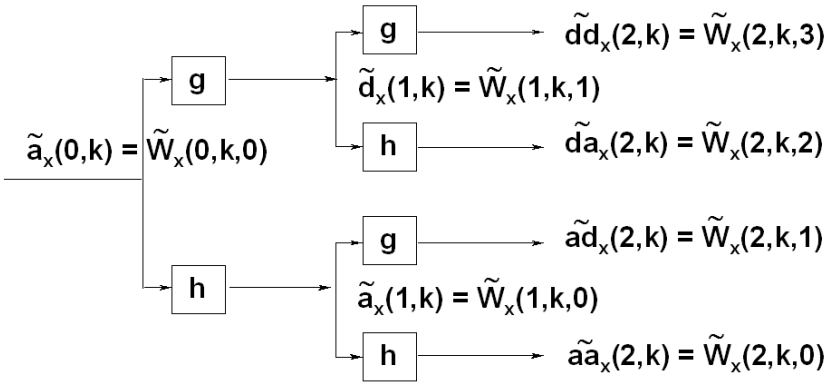

Figure 2.23: Filter bank implementation of the MODWPT decomposition at level $J = 2$.

The MODWPT is a highly redundant transform in both time and frequency domains; actually, it is the most redundant transform that can be defined. It includes the nice properties of time-shift invariance of the MODWT and the flexible subband tiling of the WPT. Its ANOVA is similar to that of the WPT, with the only difference that, as in the MODWT case, the ANOVA can be performed only on the wavelet coefficients and not on the "approximations" and "details" (though these terms do not have any sense in the context of a WPT-like transform).

2.8 Dual Tree (Complex) Wavelet Transform (DTWT)

2.8.1 A not-so-redundant transform

The main drawbacks of the MODWT are the computational complexity and the correlation induced in the output, as both phenomena are caused by the redundancy of the transform. On the other hand redundancy is good for our interests, since it allows us to accurately locate the position where a certain event takes place. Therefore, we face a trade-off where the DWT and the MODWT represent the extreme positions. But there are other, alternative wavelet transforms that can be considered as an intermediate point.

An interesting and recent development is Kingsbury's Dual Tree (Complex) Wavelet Transform (DTWT) [Kin01]. The transform can be interpreted as a dual DWT tree in which one of the trees analyzes the odd samples, while the even-numbered samples are analyzed by the second tree, thus

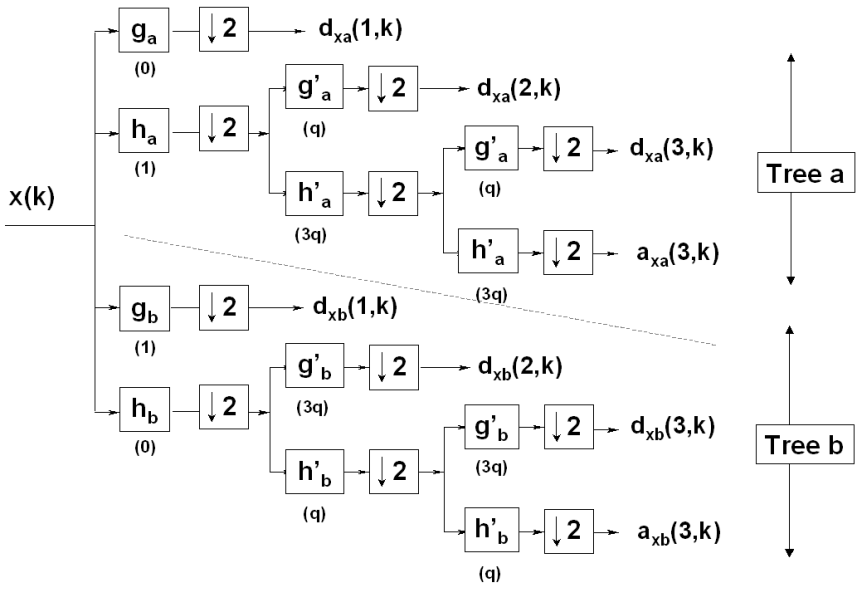

Figure 2.24: Filter bank for the DTWT decomposition at level $J = 3$. The parentheses mark the delays of the Q-shift filters, where (q) means 1/4 of a sample.

introducing a relative delay of a sample between them. This is somehow equivalent to skipping the downsampling step at the first level of the filter bank. For the second and subsequent levels, the filter bank scheme is identical to that of the DWT on both trees, as shown in Figure 2.24. The DTWT doubles the sampling rate of the original DWT; i.e., for an input $x(n)$ of length N samples, the DTWT generates $2N$ samples. Recall that the effective rate of a MODWT at level J is $J + 1$ times higher than the DWT, since it generates JN detail samples and N approximation samples at the output.

The main feature of the DTWT is that, when used with the appropriate filters, it achieves an *almost* shift-invariant analysis with *almost* uncorrelated coefficients at the expense of a slight increase in redundancy when compared with DWT. Therefore, the DTWT maintains almost all the beneficial features of the MODWT without paying such a high price in terms of redundancy, correlation and complexity.

The properties of the DTWT can be summarized as follows [Kin00]:

- Approximate shift invariance;
- Perfect reconstruction using short linear-phase filters;
- Limited redundancy, independent of the number of scales;
- Efficient order-N computation (only twice the computational complexity of the classical DWT);
- Good directional selectivity[1].

[1]This is a very valuable property for 2-D image coding but not so important in our 1-dimensional time analysis.

2.8.2 The complex nature of the DTWT

The complex term usually applied to the DTWT comes from the interpretation of every couple of coefficients from both trees (a and b) as the real and imaginary parts of a complex number $a + ib$. This is equivalent to using a DWT-style filter bank in which the filters are complex [MK98]; i.e., the filters have complex coefficients and generate complex output samples.

As a complex signal, the output of the DTWT can be expressed in terms of real and imaginary parts, or as modulus and phase. In Chapter 8 we will see how to interpret these magnitudes for our interests. Figure 2.25 shows a complex wavelet, with its real and imaginary parts (shifted), and its modulus (centered).

2.8.3 DTWT filters

Contrary to what we have learned about the "classical" wavelet transforms (DWT, MODWT, WPT) built from the couple of scaling/wavelet filters, eight (potentially) different filters are involved in the computation of the DTWT:

- $g_a(k)$ and $h_a(k)$: the high-pass and low-pass filters for the first level in tree a,
- $g_b(k)$ and $h_b(k)$: the high-pass and low-pass filters for the first level in tree b,
- $g'_a(k)$ and $h'_a(k)$: the high-pass and low-pass filters for the levels above the first, in tree a, and
- $g'_b(k)$ and $h'_b(k)$: the high-pass and low-pass filters for the levels above the first, in tree b.

As aforementioned, we want to double the sampling rate and simulate the elimination of the downsampling. This is equivalent to ensuring the delays of filters $g_b(k)$ and $h_b(k)$ are one sample offset from $g_a(k)$ and $h_a(k)$. In this way we ensure that the level 1 downsamplers pick up different samples in each tree (the even ones for one tree and the odd ones for the other). Furthermore, in order to obtain uniform intervals between samples from the two maximally decimated trees above level 1, the filters in one tree must delay the signal by half a sample (at each filter rate) from those in the other tree. If we want to keep the linear phase property, this requires odd length filters in one tree and even length filters in the other. Greater symmetry can be obtained if each tree uses odd and even filters alternatively from level to level.

There are different possibilities for designing the filters. The first level filters are much like the classical DWT filters, while the filters above the first level are less ordinary. For the first level, biorthogonal filters such as Antonini (9,7), LeGall (5,3), or Near-Symmetric (5,7) and (13,19) tap filters are the usual choice [Kin01]. The filters above the first level must ensure the half a sample delay, which can be obtained by 0 and $\frac{1}{2}$ delays (such as the so-called *near-orthogonal odd/even filter sets* [Kin04]).

The odd/even filter approach has some drawbacks: the frequency response of both tress are not exactly identical; and the sampling structure generated is not totally symmetrical (wavelet and scaling functions at a given scale are not well aligned). Better performance can be obtained with the quarter-shift or *Q-shift* orthogonal filters. These filters, of even lengths ((6,6), (10,10), (14,14), (16,16) and (18,18) taps), are interleaved on both trees with a delay of approximately $\frac{1}{4}$ and $\frac{3}{4}$ of a

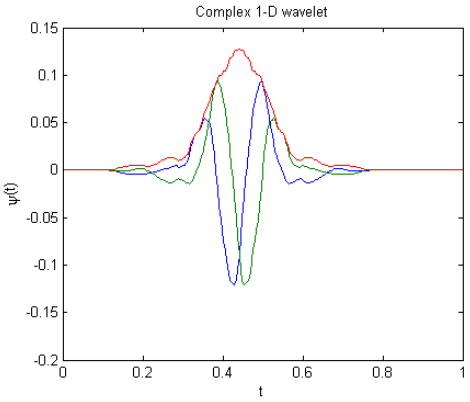

Figure 2.25: Example of a complex wavelet with real and imaginary parts (in blue and green, respectively) and its modulus (in red).

sample [Kin01], as shown in Figure 2.24. Since the Q-shift filter coefficients are no longer symmetric, orthonormal perfect-reconstruction filter sets can be constructed, just like Daubechies filters. In this case the reconstruction filters are the time reverse of the equivalent analysis filters in both trees, and all filters above level 1 are derived from the same orthonormal prototype set [Kin01].

2.8.4 Shift invariance

The almost time shift invariance properties of the DTWT are illustrated in Figure 2.26. A step function with 16 one-sample shifts is analyzed with the DTWT and the DWT. The former outperforms the latter, since the DTWT details and approximations are *almost* exact shifted versions, while the DWT signals are completely different.

2.8.5 Multiresolution analysis and ANOVA

A DTWT-based MRA can be performed in the same way as the DWT or MODWT cases, except that for each detail or approximation we get a complex number with the contributions from both trees. The reconstruction into a real signal is achieved by separately reconstructing the real and imaginary parts using biorthogonal filters designed for perfect reconstruction with the corresponding analysis filters, which results in two real signals. Averaging both signals we get the original signal. Given an original signal $X(t)$ and a DTWT decomposition with complex details at level j, $D_j(t)$ for $j = 1 \dots J$, and a complex approximation at level J, $A_J(t)$,

$$X(t) = \sum_{j=1}^{J} D_j(t) + A_J(t) \tag{2.43}$$

The whole DTWT system is a *wavelet frame* [Mal98]) with a redundancy factor 2. If the filters are designed in such a way that the analysis and reconstruction filters have very similar frequency responses, then it is an *almost tight frame*, which means that energy is approximately preserved when

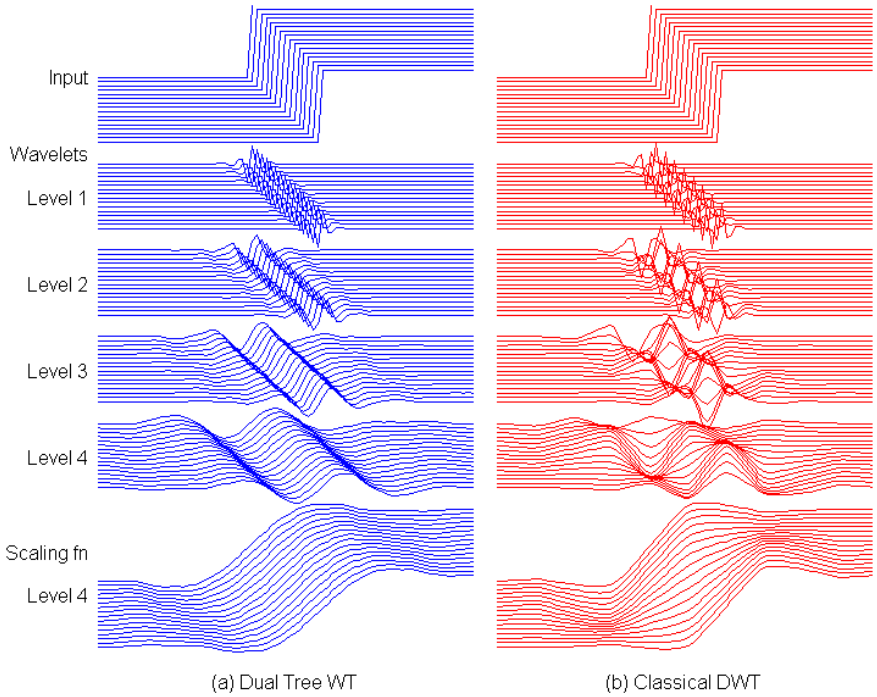

Figure 2.26: Example of the shift invariance properties of the DTWT. Sixteen shifted step functions are analyzed with the DTWT and the DWT at level $J = 4$. The approximations and the details are shown for both transforms.

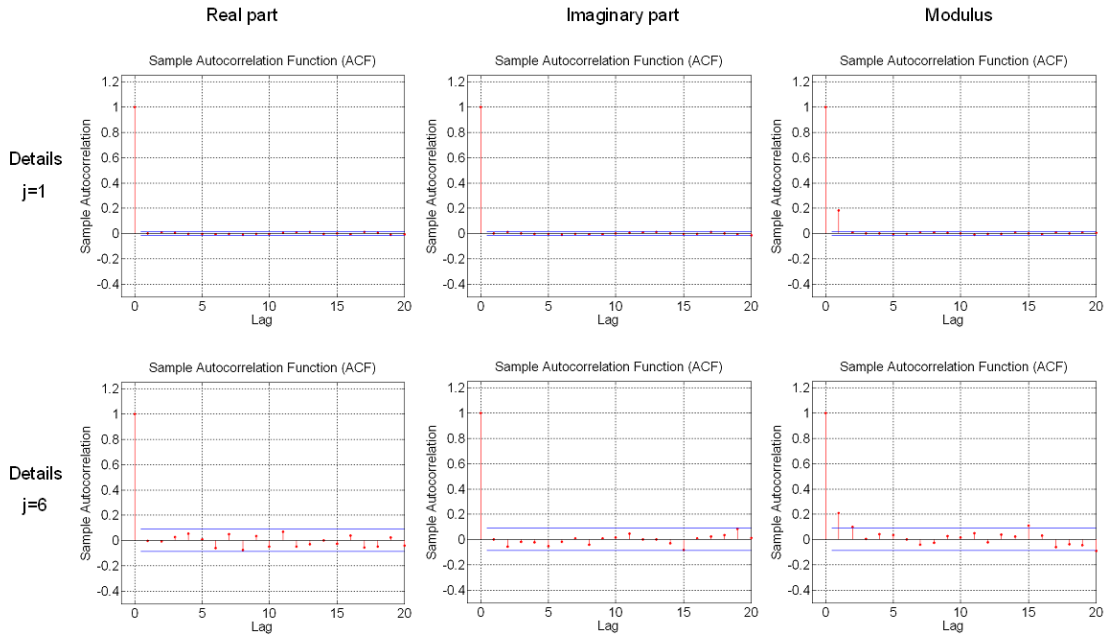

Figure 2.27: Autocorrelations of the real and imaginary parts and the modulus of the DTWT wavelet coefficients of a 32768-sample white noise signal.

signals are transformed into the wavelet domain [Kin01][2]. Therefore, no exact energy preserving condition holds for the DTWT and no exact ANOVA can be performed. This would be a drawback for our purposes, since as we will see in Chapter 3, the scaling parameter estimation methods rely on a scale-by-scale analysis of variance. However, if the two DWT-style trees are orthonormal (the usual situation), energy is preserved [SBK05]. Given an original signal $X(t)$ and its DTWT decomposition with complex wavelet coefficients at level j $d_X(j,k)$ for $j = 1 \dots J$ and a complex approximation coefficients at level J, $a_X(J,k)$,

$$\begin{aligned}\|X\|^2 &= \sum_t X^2(t) \\ &= (Re(a_X(J,k)))^2 + \sum_k (Im(a_X(J,k)))^2 + \sum_{j=1}^{J}\sum_k (Re(d_X(j,k)))^2 + \sum_{j=1}^{J}\sum_k (Im(d_X(j,k)))^2\end{aligned}$$

We provide now an example of the approximation of the analysis of variance. A white noise signal of 32768 samples, similar to the one analyzed in Section 2.6.4, is decomposed with the DTWT. The energy of the original signal, $\|X\|^2 = \sum_t X^2(t) = 52368$. The energy of the real and imaginary parts of the wavelet detail coefficients and the approximation coefficients is shown in Table 2.1. The relative difference between the energy of the original signal and the the sum of the energy of the subbands of the DTWT is as low as $1.8e-4$.

	$\|\|d_X(1,k)\|\|^2$	$\|\|d_X(2,k)\|\|^2$	$\|\|d_X(3,k)\|\|^2$	$\|\|d_X(4,k)\|\|^2$	$\|\|a_X(4,k)\|\|^2$	Total
Real	13121	6398	3329	1701	1634	26184
Imag	12988	6511	3351	1674	1660	26184
Total	26109	12909	6680	3375	3294	52368

Table 2.1: Energy decomposition of the white noise signal analyzed with the DTWT at level $J = 4$.

2.8.6 Correlation

The presence of high correlation was mentioned in Section 2.6.4 as one of the drawbacks of the MODWT, and this was one of the main reasons for considering the DTWT as an alternative. Thanks to the low redundancy factor (2) when compared with the DWT, the DTWT keeps correlation at a low level, though different from zero. We will not provide a mathematical treatment but just an example. Figure 2.27 shows the autocorrelation function of the real and imaginary parts and the modulus of the DTWT of a white noise signal of 32768 samples. The real and imaginary parts are uncorrelated, while the modulus show a short correlation, usually confined in the first lag, at most. The correlation level does not increase significantly with the scale.

[2] A tight frame is one where the signal reconstruction can be performed with the transpose of the forward transform.

2.9 Summary of the chapter

This chapter began with an intuitive introduction to wavelet analysis, starting with the continuous CWT and later with the orthogonal, fully downsampled DWT. The DWT is a powerful tool whose main property is the capability of performing a multiresolution analysis, in which the original signal is decomposed in a low-pass approximation and a set of high-frequency details. Another important feature is its ability to perform scale-based analysis of variance, either with the wavelet and approximation coefficients, or with the details and approximations as time functions. The MRA and ANOVA capabilities open the way towards developing a multi-scale time event detector, in which a certain event in time (a change of variance, a change in the scaling parameters of traffic) can be detected by the alignment of events seen at different scales. Finally, the DWT is easily implemented as a quadrature-mirror filter bank, iterated in the low-pass subband. The computational complexity of the scheme is very low.

However, the DWT presents some drawbacks. Its time resolution at higher scales can be very low, due to the trade-off between the dispersion of time and frequency analysis. Its lack of shift invariance is another problem; since the time events we are looking for will not necessarily be aligned with the boundaries of the wavelet analysis, the result of the MRA and ANOVA for the same signal would be different depending on the relative time shift between the signal and the wavelet transform.

An alternative to both the time-resolution and shift-invariance problems is the non-orthogonal, time-redundant MODWT. This transform is perfectly shift-invariant and provides the same time resolution at any scale. However, its computational complexity is also increased notably when compared with the DWT and, what is more, introduces correlation in the wavelet coefficients. Since the statistical methods for detecting changes in time series usually assume an uncorrelated input, this would prevent us from using the MODWT as planned.

If the redundancy introduced by the MODWT is in the time axis, another wavelet transform known as Wavelet Packet (WPT) is its counterpart in the frequency domain. The WPT generalizes the filter bank by allowing all the possible dyadic subband decompositions; i.e, by letting the quadrature-mirror filter bank be iterated in both the high-pass and the low-pass outputs. This results in a total flexibility for choosing the transform basis, either orthogonal, undercomplete or overcomplete. This transform paves the way for applying a time-varying basis analysis in which the signal is studied with a subband decomposition scheme that adapts to the spectral properties of the time series under study. An MRA and ANOVA can be performed with the WPT.

Two recently developed, time-redundant wavelet transforms known as Dual Tree (Complex) WT and Double Density WT (see Appendix A) are somehow between the DWT and the MODWT. They both provide near-time-shift invariance and a moderated correlation (being the DTWT better than the DDDWT), while effectively doubling the sampling rate of the classical DWT. MRA and ANOVA can be performed with these transforms, though the latter only works with the transform coefficients and not with the approximations and details. These two transforms provide an intermediate position in the trade-off between (undesired) correlation and (desired) redundancy.

Chapter 3

Static estimation of the scaling parameters of self-similar processes

3.1 Introduction

As pointed out in Section 1.3 scaling behavior has been detected in several measurements of the traffic carried by packet switching communication networks. This phenomenon is surprising because of its universality, but also because of the wide range of scales over which scaling holds.

This chapter deals with the issues of *identification and measurement of scaling behaviour*, and also reviews the results regarding the *static* situation in which scaling parameters are (ideally) constant. Though this is mathematically convenient, reality is not as nice. Nonstationarities are expected in both traffic volume and scaling behavior. A typical situation is when the Hurst parameter of a LRD process $X(t)$ changes over time. In this case, if $H(t)$ is a deterministic function with enough regularity, the process is said to be *locally monofractal*. Such a process is clearly not stationary, since its distribution depends by definition on the deterministically changing $H(t)$, but this nonstationarity derives from a true scaling behaviour. Therefore, the static estimation algorithms have to be modified in order to adapt them to changing situations; this will be the topic of Chapter 5.

We want to stress the fact that the development of such measurement techniques are not only important by themselves; they are also useful for other studies even if not directly related to scaling. The presence of scaling affects the estimation of other parameters, not merely those that describe scaling [AFTV00]. An example is the estimation of the mean of a data set of length N from a second-order stationary process $X(t)$, with mean η and variance σ^2. The sample mean estimator is:

$$\widehat{\eta} = \frac{1}{N} \sum_{t=1}^{N} X(t)$$

Skipping the details, we have that [VA99]:

- if $X(t)$ is SRD, the sample mean asymptotically follows a normal distribution with expectation equal to η and second moment

$$E\{\widehat{\eta}^{\,2}\} = \frac{\sigma^2}{N} \ , \quad N \gg 1$$

- if $X(t)$ is LRD, the sample mean is also asymptotically normally distributed with expectation equal to η and second moment

$$E\{\widehat{\eta}^{\,2}\} = \frac{2c_\gamma N^{\alpha-1}}{\alpha(1+\alpha)} \ , \quad N \gg 1$$

Therefore, the variance of the sample mean estimator of an LRD process decreases with N in a much slower way than the classical, Poisson-style case. This has strong implications for the computation of confidence intervals. It is also a powerful example of the importance of the second LRD parameter, as discussed in Section 1.2.3. Since the variance is proportional to c_γ, it follows immediately that the confidence intervals about the sample mean estimate are essentially proportional to $\sqrt{c_\gamma}$. Another example of the importance of the second order parameter is given in [VA99], where it is proved that for queues fed with LRD traffic queueing delays increase when c_γ increases. This is consistent with the role of the second order parameter as a quantitative or volume measure of the LRD phenomenon.

This chapter focuses on the wavelet-based techniques that allow the measurement of scaling under the hypothesis of *monofractality*, that is, of stationarity of the scaling parameters (and, in particular, of the Hurst parameter) across time. To obtain this goal, we first present the details of the Abry-Veitch estimator), a DWT-based estimator built on a graphic tool called the *LogScale Diagram* [VA99]. This estimator is widely accepted as the most accurate, flexible and computationally efficient estimator of (static) scaling. But as we discussed in Chapter 2, the DWT has some limitations. That is why we extend the estimator to other, alternative wavelet transforms with better properties.

The Wavelet Packet Transform (WPT) is one of these alternatives. Recalling the theory presented in Chapter 2, the WPT provides an overcomplete representation of the original signal, by iterating the low-pass / high-pass filter bank in a full binary tree. This generates 2^j subbands at each scale j, and a total of $2^{J+1} - 1$ subbands for a J scales decomposition. The identification of the subbands cannot be done solely by the scale j, and a new parameter $m = 0 \dots 2^{j-1}$ is needed; therefore, we get a (j, m) subband map of the processes under study.

We have developed an extension of the Abry-Veitch estimator for the WPT decomposition, capable of performing an estimation from partial (for fixed j and variable m), global (for all j and m) or arbitrary decompositions of the subband map. Section 3.3.1 presents the analytic elements of the estimator, together with its application to synthetic FGN traces and real traffic traces. The contributions of this chapter regarding the WPT estimator have been described in [RS02, RS04a, RS04b, Min05].

3.2 DWT-based estimation

3.2.1 Nice properties of the wavelet transform

Why is the wavelet transform so appealing for studying scaling processes? The following wavelet properties are the key [PW00]:

(1) The wavelet basis, being constructed through a dilation procedure, is intrinsically invariant with scale, just like scaling processes. This makes it well suited for analyzing such processes.

(2) The correlation of a scaling process in the time domain is *absorbed* by the wavelet transform, thus decorrelating the process in the wavelet domain.

(3) The polynomial trends of order up to $N-1$ present in the scaling process are eliminated by the wavelet transform, as long as the mother wavelet $\psi(t)$ has N vanishing moments:

$$\int t^k \psi(t) dt = 0, \quad k = 0 \dots N \tag{3.1}$$

3.2.2 DWT of self-similar processes

The wavelet coefficients $d_X(j,k)$ of an H-SS process $X(t)$ reproduce the scaling property:

$$(d_X(j,0), d_X(j,1) \dots d_X(j,N_j-1)) =^d 2^{j(H+\frac{1}{2})}(d_X(0,0), d_X(0,1) \dots d_X(0,N_j-1)) \tag{3.2}$$

where N_j is the number of wavelet coefficients at scale j. For second-order processes, a consequence of Equation 3.2 is:

$$E(d_X(j,k)^2) = 2^{j(H+\frac{1}{2})} E(d_X(0,k)^2) \tag{3.3}$$

In the case that $X(t)$ has stationary increments, H-SSSI, the following properties hold:

P1 H-SSSI The wavelet coefficients for a certain scale j are a stationary process. In this case, Equation 3.3 reduces to:

$$E(d_X(j,k)^2) = 2^{j(H+\frac{1}{2})} C(H,\psi t)\sigma^2, \quad \forall k \tag{3.4}$$

where $C(H,\psi t) = \int |t|^{2H} (\int \psi(u)\psi(u-t)du)dt$ and $\sigma^2 = E(X(1)^2$.

P2 H-SSSI Given the covariance of an H-SSSI process $X(t)$,

$$E(X(t)X(s)) = \frac{\sigma^2}{2}[|t|^{2H} + |s|^{2H} - |t-s|^{2H}] \tag{3.5}$$

and given that $N \geq H + \frac{1}{2}$, then the correlation between wavelet coefficients at j,k and j',k' is small and depends on N:

$$E(d_X(j,k)d_X(j',k')) \approx |2^j k - 2^{j'} k'|^{2H-2N} \quad \text{when} \quad |2^j k - 2^{j'} k'| \to +\infty \tag{3.6}$$

3.2.3 DWT of LRD processes

Recall the properties of the wavelet basis presented in Section 2.4.1. For a LRD process, these features engender the following key properties [VA99] of the wavelet coefficients $d_X(j,k)$ over a range of scales $j = j_1, \ldots, j_2$ where the power-law scaling holds.

P1 LRD Due to Proposition 2.1, the process $\{d_X(j,k), k \in \mathbb{Z}\}$ is stationary if $N \geq (\alpha - 1)/2$ and the variance of the wavelet coefficients accurately reproduces the underlying scaling behaviour of the data:

$$\mu_j = E\{d_X(j,k)^2\} \sim 2^{j\alpha}\ c_f C(\alpha, \psi), \quad j \to +\infty \tag{3.7}$$

where α and c_f are the LRD parameters, and C is given by the integral

$$C(\alpha, \psi) = \int_{-1/2}^{1/2} |f|^{-\alpha} |\Psi(f)|^2\ df \tag{3.8}$$

P2 LRD Due to Proposition 2.1 and property P1, the $d_X(j,k)$ are a collection of quasidecorrelated random variables. This makes the time correlation caused by LRD disappear in the wavelet domain(j,k). Moreover, the higher N is, the shorter is the correlation:

$$E(d_X(j,k)\ d_X(j,k')) \approx |k - k'|^{\alpha - 1 - 2N}, \quad |k - k'| \to +\infty \tag{3.9}$$

3.2.4 Summary for scaling processes

Let $X(t)$ be either an H-SSSI process, LRD process, a generalized second-order stationary $\frac{1}{f}$-spectrum process, or a fractal process. The wavelet coefficients of such a process, $\{d_X(j,k), k \in \mathbb{Z}\}$ are stationary if $N \geq (\alpha - 1)/2$, and its variance follows Equation 3.7 within a range of scales $j_1 \ldots j_2$. Then,

- If the original process was a H-SSSI process, $\alpha = 2H - 1$, $C(\alpha, \psi)$ is to be identified from Equation 3.4, and $j_1 = -\infty, j_2 = +\infty$. Therefore the alignment is *strict* across all the scales; ideally over an infinite range.

- In the case of an LRD process, α comes from the definition of the spectral density of such a process: $\Gamma_X(f) \sim c_f |f|^{-\alpha}$. $C(\alpha, \psi)$ is to be identified from Equation 3.4, j_1 is to be identified from the LD, and $j_2 = +\infty$. In this case the alignment holds only from a certain lower cut-off scale j_1 but extends asymptotically towards infinity.

- For a generalized second-order stationary $\frac{1}{f}$ process, α is defined also from $\Gamma_X(f) \sim c_f |f|^{-\alpha}$, for $f_1 \leq f \leq f_2$, $C(\alpha, \psi) = \int |f|^{-\alpha} |\Psi(f)|^2 df$, and j_1 and j_2 are derived from f_1 and f_2.

- Finally, for a general fractal process $\alpha = 2h + 1$, expressions for $C(\alpha, \psi)$ can be found in the literature (see [AFTV00] for the references), $j_1 = 1$, and j_2 can be identified from the LD.

3.2.5 Extension to multifractal processes

The analysis of the scaling behavior in the wavelet domain, shown in Equation 3.7, can be extended to other scaling processes that cannot be expressed with a single scaling parameter. For multifractional Brownian Motion processes such as those presented in Section 1.2.7, the scaling exponent $h(t)$ is no longer constant in time and depends on the local properties of the process. In this case, $E[(X(t+\tau)-X(t))^2] \approx |\tau|^{2h(t)}$ for $\tau \to 0$, and the wavelet analysis gives a time-dependent LD in which the slope $\alpha(t) = 2h(t)+1$.

For general multifractal processes things are not so nice. To begin with, the small scales of these processes show a behavior that is extremely variable over time. The local regularity of almost every sample path $|X(\omega,t+\tau)-X(\omega,t)| \sim |\tau|^{h(\omega,t)}$ for $\tau \to 0$ (where ω is an element of the probability space underlying the process) is actually a fractal. It is then impossible to follow the time variations of the scaling exponent; instead, it is studied statistically. An example of such a study is the *Hausdorff multifractal spectrum D(h)* which consists of accounting the set of points where $h(\omega,t) = h$. A way to obtain the multifractal spectrum are the *partition functions* $S_q(t) = \int |X(\omega,t+\tau)-X(\omega,t)|^q dt$, which exhibit power-law behavior $S_q(t) \sim |\tau|^{\zeta(q)}$. The connection with wavelets comes from considering the wavelet coefficients as the increments of the sample path. Therefore, the wavelet coefficients are studied not only in their second-order moment, but on the q^{th} moment, $E\{d_X(j,k)^q\}$ and $\alpha = \zeta(\omega,q) + q/2$.

Throughout this thesis we will focus our attention on monofractal (self-similar, LRD) processes.

3.2.6 The LogScale diagram

Property **P2 LRD** is the key to understanding the advantages of performing LRD and self-similar analysis in the wavelet domain. Contrary to the problematic situation in the time domain due to the long-range dependence and possible non-stationarities of the original process $X(t)$, in the wavelet domain we deal with the stationary, SRD processes $d_X(j,k)$ for each j. This results in a very favorable statistical property: the stationarity in the wavelet domain results in a low variability of averages across the time index k, contrary to what happens in the time domain.

The following expression has an important role in our analysis:

$$\widehat{\mu_j} = \frac{1}{n_j} \sum_{k=1}^{n_j} d_X(j,k)^2 \tag{3.10}$$

where, as in Equation (2.37), n_j is the number of available wavelet coefficients at scale j.

The properties of the random variable $\widehat{\mu_j}$ are [VA99]:

- it is a nonparametric, unbiased and efficient estimator of the variance μ_j of the process $d_X(j,k)$ for a certain j.
- is a way of concentrating the second-order behaviour of $X(t)$ at scale j.
- $\widehat{\mu_j}$ are weakly dependent, making the analysis at each scale almost decoupled from those at other scales.

Therefore, a study of $\widehat{\mu_j}$ as a function of j will give us an analysis of the second-order dependence of $X(t)$ on scale j. The underlying power-law dependence on j of the variance μ_j of the processes $d_X(j,k)$ at each scale can be easily analyzed by taking the logarithms of both sides of Equation (3.7).

$$\log_2(\mu_j) = j\alpha + \log_2(c_f C) \tag{3.11}$$

This relation strongly suggests a *linear regression* approach for estimating (α, c_f), where the slope of the regression would estimate α and the intercept would be related to c_f. In other words, the scaling parameters could be extracted by considering a plot of

$$y_j = \log_2(\widehat{\mu_j}) = j\widehat{\alpha} + \log_2(\widehat{c_f C}) \tag{3.12}$$

against j. In [VA99], Abry and Veitch call this plot (together with confidence intervals about the y_j) the *LogScale Diagram* (LD). The complication here is the nonlinearity introduced by the $\log_2$, that biases the estimation. The problem comes from the fact that the expectation of the logarithm is not equal to the logarithm of the expectation; $E[\log()] \neq \log(E[])$. We will see in the next paragraph how this problem can be circumvented under reasonable hypotheses. Simplifying things slightly, we confirm that the fundamental approach is indeed a linear regression of y_j. In particular, a *weighted* linear regression will be used as the variances of the y_j vary with j due to the different quantity of available samples n_j at each scale; recall that the DWT approximately halves the samples at each iteration of the pyramidal algorithm.

Examples of LogScale Diagrams with their correspondent estimated regression lines are given in Figure 3.1 and Figure 3.2. The first plot is for a 131072-samples FGN synthetic trace with Hurst parameter $H = 0.8$ (`fgn08.mat`), while the second is for the `BC-pAug89` Bellcore trace aggregated at 10 msec (`pAug8910ms.mat`). Further information about the two traces can be found in Sections 3.2.9 and 3.2.10, respectively. In both cases, the DWT decomposition was performed using the `db1` wavelet. Confidence intervals about the y_j, shown as vertical lines at each scale j in the LDs of Figure 3.1 and Figure 3.2, are seen to increase monotonically with j as one moves to larger and larger scales, due to the decrease in n_j as j increases. The intervals are derived from the variance $\sigma_j^2 = Var\{y_j\}$ under Gaussian assumptions of the detail coefficients distributions [VA99]. The diagrams were obtained by the Matlab code developed by Patrice Abry and Darryl Veitch, available at [VA].

3.2.7 Using the LogScale Diagram

The LD allows us to visualize, detect and estimate the scaling properties present in the signals under study. The *detection* of scaling with the LogScale Diagram consists of the identification of region(s) of alignment and the determination of their lower and upper cutoff octaves, j_1 and j_2, respectively. It must be emphasized that the concept of alignment defined in terms of the confidence intervals for the y_j, and not in terms of the alignment of the y_j themselves. As mentioned earlier, the $\widehat{\mu_j}$, and hence the y_j, are weakly dependent, resulting in a natural and desirable variation around the calculated regression line.

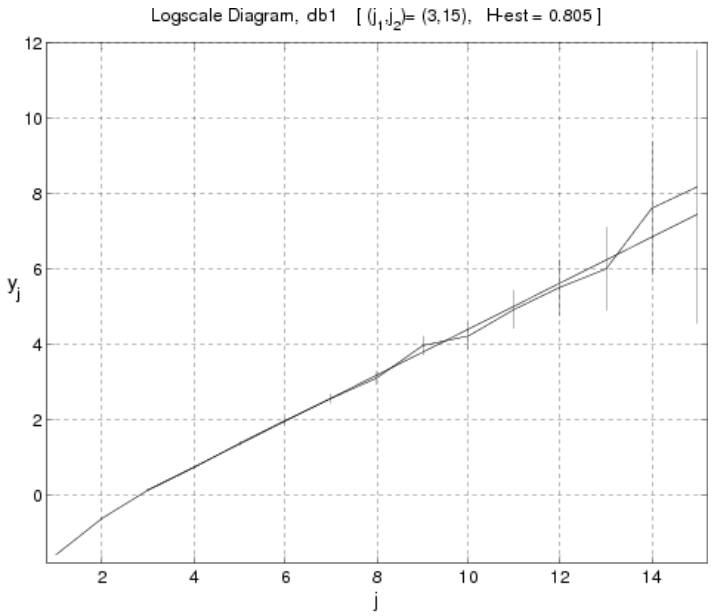

Figure 3.1: LogScale Diagram and estimated regression line for a 131072-samples FGN synthetic trace with $H = 0.8$, using `db1` wavelet.

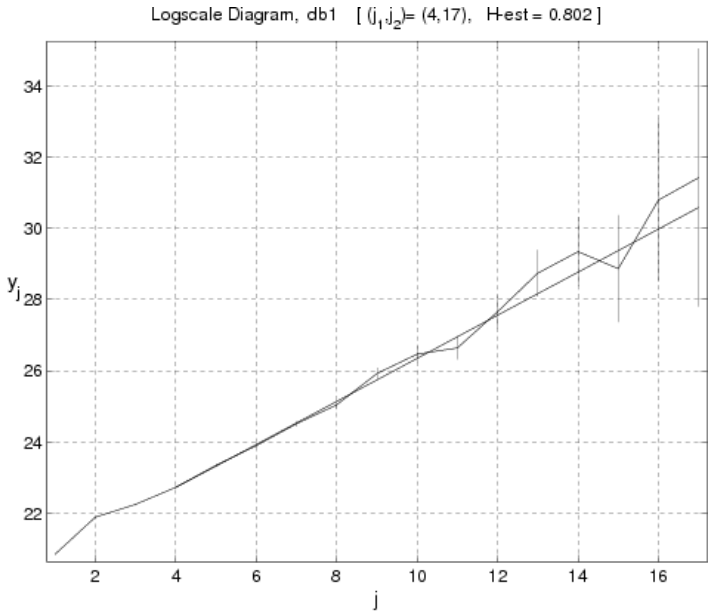

Figure 3.2: LogScale Diagram and estimated regression line for the `BC-pAug89` Bellcore trace aggregated at $\delta = 10$ msec, using `db1` wavelet.

As far as the practice selection of the cutoff scales j_1 and j_2 is concerned, a useful heuristic is that the regression line should cut each of the confidence intervals within it. This can be formalized by a *Chi-squared goodness of fit test* [VAT03], where the critical level of the goodness of fit statistic is monitored as a function of the endpoints of the alignment range. At least in the case of the lower scale, this can make a very clear and relatively objective choice of cutoff possible. Applying this test to the LDs of Figure 3.1 and Figure 3.2, we observed alignment over scales $(j_1, j_2) = (3, 15)$ and $(j_1, j_2) = (4, 17)$, respectively.

3.2.8 Analytic elements of the Abry-Veitch estimator

Linear regressions

First, some standard results on one-dimensional weighted linear regressions of the random variables y_j on $j \in [j_1, j_2]$ will be recalled. The hypothesis of linear regression is

$$E\{y_j\} = bj + a \tag{3.13}$$

where b and a are real constants. Now we define

$$S = \sum 1/\sigma_j^2 \tag{3.14}$$

$$S_j = \sum j/\sigma_j^2 \tag{3.15}$$

$$S_{jj} = \sum j^2/\sigma_j^2 \tag{3.16}$$

where σ_j^2 is a weight associated with y_j. The usual unbiased estimator $(\widehat{b}, \widehat{a})$ of (b, a) is therefore

$$\widehat{b} = \frac{\sum y_j(Sj - S_j)/\sigma_j^2}{SS_{jj} - S_j^2} = \sum w_j \; y_j \tag{3.17}$$

$$\widehat{a} = \frac{\sum y_j(S_{jj} - S_j j)/\sigma_j^2}{SS_{jj} - S_j^2} = \sum v_j \; y_j \tag{3.18}$$

where the weights w_j and v_j satisfy

$$\sum w_j = \sum j \; v_j = 0$$
$$\sum j \; w_j = \sum v_j = 1$$

Furthermore, setting

$$\sigma_j^2 = Var\{y_j\} \tag{3.19}$$

(i.e., relating the weights to the variances of the y_j) then it can be shown that $(\widehat{b}, \widehat{a})$ is the *Minimum Variance Unbiased Estimator* (MVUE) [VA99].

The y_j statistic

In our model, $\log_2(\widehat{\mu_j})$ is the variable y_j of the desired linear regression satisfying $E\{y_j\} = bj + a$. Since

$$E\{\log_2(\widehat{\mu_j})\} \neq \log_2(E\{\widehat{\mu_j}\}) = j\alpha + \log_2(c_f C) \tag{3.20}$$

this condition is not exactly satisfied, however. Therefore, small corrective deterministic factors g_j, discussed below, are introduced and redefine the y_j as

$$y_j = \log_2(\widehat{\mu_j}) - g_j \tag{3.21}$$

so that by the new definition it holds

$$E\{y_j\} = j\alpha + \log_2(c_f C) \tag{3.22}$$

Now we assume that the following idealizations hold true [AFTV00].

Idealization 1. *The processes $d_X(j,k)$ are Gaussian for each fixed j.*

Idealization 2. *For each fixed j, the processes $d_X(j,k)$ are stationary sequences of i.i.d. random variables.*

Idealization 3. *The processes $d_X(j,k)$ and $d_X(j',k)$, $j \neq j'$ are independent.*

Idealizations 2 and 3 are both justified by property **P2 LRD**, while 1 is justified by numerical evidence. The idealizations may appear very restrictive at first. However, the method is based on properties **P1 LRD** and **P2 LRD**, and the idealizations only make easier the quantitative the analysis. Under the three idealizations, it can be shown [VA99] that g_j is a negative, increasing function of n_j only, given by

$$g_j = \psi(n_j/2)/\ln 2 - \log_2(n_j/2) \tag{3.23}$$

where $\psi(x) = \Gamma'(x)/\Gamma(x)$ is the *Psi function* and $\Gamma(x)$ is the *Gamma function*.

The idealizations also cause the y_j to be scaled and shifted logarithms of Chi-squared variables with variance

$$\sigma_j^2 = \zeta(2, n_j/2)/\ln^2 2 \tag{3.24}$$

where $\zeta(z,v)$ is the *generalized Riemann Zeta function*.

Finally, under the same idealizations and also assuming n_j is "large enough", it can be established that

$$g_j \;\rightarrow\; \frac{\log_2 e}{n_j} \tag{3.25}$$

$$\sigma_j^2 \;\rightarrow\; \frac{2\log_2^2 e}{n_j} \tag{3.26}$$

$$y_j =_d N\left(j\alpha + \log_2(c_f C), \frac{2\log_2^2 e}{n_j}\right) \tag{3.27}$$

where $N(\mu, \sigma^2)$ is a Gaussian random variable with mean μ and variance σ^2. Therefore, $y_j = \log_2 \mu_j$ is asymptotically normally distributed and $\widehat{\alpha}$, which consists of a sum of approximately Gaussian variables, can be considered as approximately Gaussian distributed. Confidence intervals are computed for y_j and $\widehat{\alpha}$ assuming these normal distributions.

Definition of the joint estimator

The LRD estimator $(\widehat{\alpha}, \widehat{c_f})$ and another, related estimator $(\widehat{\alpha}, \widehat{c_f C})$ are defined as follows. The quantity $\widehat{c_f C}$ is of interest because, like $\widehat{\alpha}$, its statistical properties are entirely independent of the specific form of the mother wavelet. It is nonetheless very closely related to $\widehat{c_f}$, yet has the advantage of being amenable to a detailed analysis. The study of $\widehat{c_f}$ is rendered far more complex because of the wavelet dependence, which enters explicitly via the integral (3.8).

Skipping the mathematical details discussed in [VA99], the joint estimator $(\widehat{\alpha}, \widehat{c_f C})$ is given by

$$\widehat{\alpha} = \widehat{b} \tag{3.28}$$

$$\widehat{c_f C} = p\, 2^{\widehat{a}} \tag{3.29}$$

where

$$p = \prod \frac{\Gamma(n_j/2) \exp[\psi(n_j/2) v_j]}{\Gamma(v_j + n_j/2)}$$

is a bias-correcting factor. Finally, defining the estimator $\widehat{C}$ of the integral (3.8) as

$$\widehat{C} = C(\widehat{\alpha}, \psi_0) \tag{3.30}$$

then we can obtain $(\widehat{\alpha}, \widehat{c_f})$ as

$$\widehat{\alpha} = \widehat{b} \tag{3.31}$$

$$\widehat{c_f} = \widehat{c_f C} / \widehat{C} \tag{3.32}$$

Here are summarized the main properties of the estimators defined above [VA99]:

- $(\widehat{\alpha}, \widehat{c_f C})$ is *unbiased*, even for data of finite size;
- $\widehat{C}$ is *asymptotically unbiased* and *efficient*;
- $\widehat{c_f}$ is *asymptotically unbiased* and *efficient*;
- $\widehat{\alpha}$ is approximately *Gaussian distributed*;
- $\widehat{c_f C}$ and $\widehat{c_f}$ are approximately *log-normally distributed*.

Abry and Veitch also developed the MRA initialization procedure described in Section 2.4.7 for the LogScale analysis of intrinsically discrete data, such as that provided by network traffic traces. This procedure enhances the accuracy of the scaling estimator.

In our experiments, we will only deal with the estimation of α, but future investigations will be dedicated to $\widehat{c_f}$.

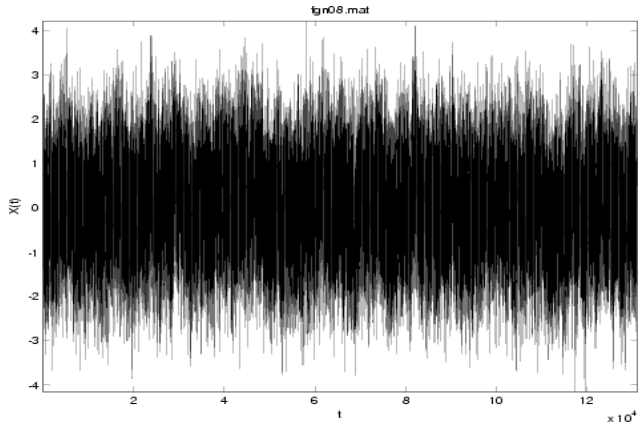

Figure 3.3: FGN synthetic trace with $H = 0.8$ (`fgn08.mat`).

3.2.9 Application to FGN synthetic traces

We first apply the estimator of Equation (3.31) to four FGN synthetic traces with Hurst parameter $H = 0.5$ (`fgn05.mat`), $H = 0.7$ (`fgn07.mat`), $H = 0.8$ (`fgn08.mat`) and $H = 0.9$ (`fgn09.mat`), and with unit power and zero mean. As an example, in Figure 3.3 we have plotted `fgn08.mat`.

We use H rather than α as it is more customary in teletraffic studies; recall from Equation (1.9) that

$$H = \frac{1+\alpha}{2} \tag{3.33}$$

All traces are composed of 131072 samples and were generated using the Matlab® 7.0.1 Wavelet Toolbox command `wfbm.m`. The estimations, whose results using the `db1` wavelet family are summarized in Table 3.1, were performed by the code developed by Patrice Abry and Darryl Veitch, available at [VA]. Since the number of vanishing moments of chosen wavelet family influences the analysis, some experiments were performed with higher order wavelets. The results were almost without changes, as shown in Table 3.1 for `db3`.

Trace	`fgn05.mat`	`fgn07.mat`	`fgn08.mat`	`fgn09.mat`
(j_1, j_2)	(3,15)	(3,15)	(3,15)	(3,15)
$\widehat{H}$ `db1`	0.510	0.706	0.805	0.904
$[95\%CI]$ `db1`	[0.502,0.518]	[0.698,0.714]	[0.797,0.813]	[0.896,0.912]
$\widehat{H}$ `db3`	0.505	0.707	0.808	0.908
$[95\%CI]$ `db3`	[0.497,0.513]	[0.699,0.716]	[0.800,0.816]	[0.899,0.916]

Table 3.1: DWT-based H estimations for four FGN synthetic traces, with `db1` and `db3`.

The tables are organized as follows: in the first row, there are the names of the synthetic traces, while the second row contains the cutoff scales for the alignment range of the LogScale Diagram, selected with the goodness of fit test described in Section 3.2.6. The third row shows the estimated H values, obtained from Equations (3.31) and (3.33). Finally, in the fourth row there are the 95%

confidence intervals for the estimations, derived under Gaussian assumptions on $y_j = \log_2(\widehat{\mu_j})$. The conclusion is that the estimated values are very close to the theoretical ones; this confirms the goodness of the estimator. Furthermore, the synthetic FGN traces are free of polynomial trends, as shown by the almost identical results obtained with `db1` and `db3` wavelets.

The previous results have been obtained with single traces. A more complete study has been performed by running the FGN generator 1000 times and computing the histogram of the analysis of the 1000 independent generations. In this case, `db2` wavelet and $j_1 = 3, j_2 = 14$ have been used. Figure 3.4 and Table 3.2 show the results. Note the approximately normal distribution of the results. A slight bias toward a higher value of H is detected, especially in the case of high values of H; in any case, the bias is very little and probably is caused by the fact that the scales for the linear regression have been chosen constant for each run, contrary to the advices of Abry and Veitch, who stress the importance of performing a detailed study of each trace. The automatic procedure has kept us from providing such an analysis.

Traces	$H = 0.5$	$H = 0.7$	$H = 0.8$	$H = 0.9$
Mean $\widehat{H}$	0.5001	0.7042	0.8055	0.9066
Std dev $\widehat{\sigma}$	0.0039	0.0041	0.0042	0.0043

Table 3.2: DWT-based H estimations for 1000 independent runs of 131072 samples FGN synthetic traces, with `db2`, $j_1 = 3$ and $j_2 = 14$.

3.2.10 Application to real traffic traces

As discussed in Chapter 1, the presence of LRD in traffic data of diverse types is now well accepted. Some of the most detailed evidence comes from the Bellcore Ethernet traces described in [LTWW94]. The traces in question have become in some sense *de facto* standards in the field of traffic analysis. There exists more than one Bellcore trace, but through the present work we will only consider the `BC-pAug89` and `BC-OctExt` traces.

The file `pAug89.dat` consists of one million rows in two columns, where each row corresponds to one Ethernet frame. The first column provides the timestamp in seconds (measured from the beginning of the trace) for the *end* of the frame. The second column contains the size of the frame (in bytes). The actual traffic consists, therefore, of an alternating sequence of disjoint frames and silent periods. Frames have a maximum and minimum size, whereas silences have a minimum but not maximum duration. Since silences are not restricted to multiples of bytes (that is, time is not slotted), we could define the Ethernet arrival process in continuous time. We do not do this, but consider discretized versions where the process is aggregated within a window of size δ seconds. This simplifying procedure is common practice and was justified in [AV98]. Therefore, each sample of our aggregated trace will correspond to the amount of bytes transported by the Ethernet network during each segment of size δ. This produces a time series that is actually the instantaneous bitrate carried by the network, measured at the 10 msec scale. In order to get meaningful results, δ must

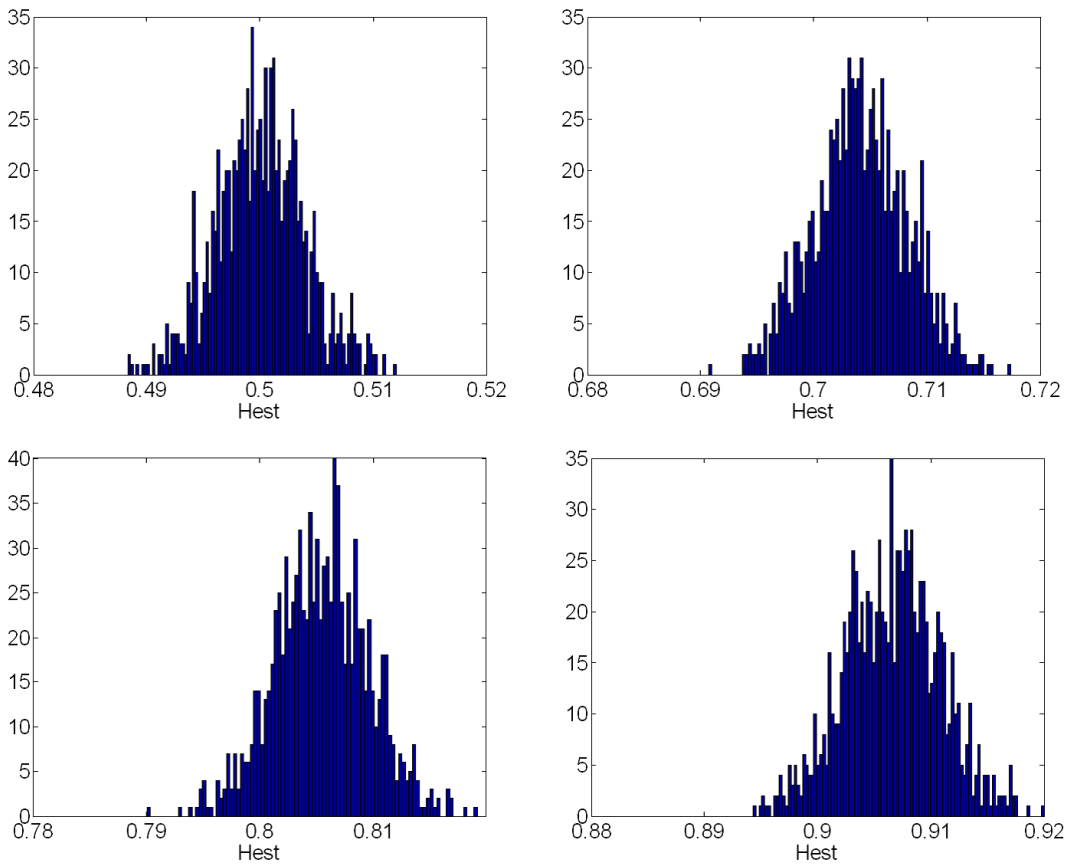

Figure 3.4: Histogram of the analysis of 1000 FGN synthetic traces of 131072 samples with $H = 0.5$ (top left), $H = 0.7$ (top right) , $H = 0.8$ (top left), and $H = 0.9$ (bottom right). The analysis was performed with `db2` wavelet and $j_1 = 3, j_2 = 14$.

be chosen carefully. It cannot be less than the (real) accuracy of the capturing device; a good rule-of-thumb is to choose δ at least an order of magnitude higher than the accuracy of the capture card. On the other hand, if the resulting trace has a lot of zero samples (as is the case when δ is very little) the capture presents non-stationarities (bursts of zeros followed by non-zero samples) that bias the estimation. For our analysis, we chose $\delta = 10$ msec and called the corresponding 314282-samples trace `pAug8910ms.mat`.

In order to avoid the bias caused by possible polynomial trends, Abry and Veitch recommend performing several analyses with increasing numbers of vanishing moments until the results stabilize. Applying the H estimator of Equation (3.31) to this trace and using the `db1`, `db2` and `db3` wavelet families, one gets the results shown in Table 3.3. This confirms what was stated in [LTWW94] and [VA99], where the pAug89 trace aggregated at 10 msec is shown to exhibit long-range dependence with Hurst parameter close to 0.8. The results are almost identical for the three analyses, thus confirming the inexistence of polynomial trends. Figure 3.2 shows the LogScale Diagram for the `db1` case, which is very similar to the diagrams for the other two wavelets.

Wavelet	Scales (j_1, j_2)	$\widehat{H}$	95% conf. int.
db1	(4,17)	0.802	(0.795, 0.810)
db2	(4,12)	0.803	(0.795, 0.810)
db3	(4,11)	0.805	(0.797, 0.813)

Table 3.3: DWT-based H estimations for the BC-pAug89 trace aggregated at 10 ms.

We perform a similar analysis of another of the Bellcore set traces, the one known as BC-OctExt. This trace is known to be highly variable. It corresponds to 35 hours of data coming from the outbound link of the Bellcore network, and includes a complete daily cycle. In this case the aggregation is performed at the $\delta = 1$ second scale. The long-memory and LRD properties of the trace are confirmed by the Hurst parameter estimation shown in Table 3.4. The results are also coherent with those described in [VA99].

Wavelet	Scales (j_1, j_2)	$\widehat{H}$	95% conf. int.
db1	(4,17)	0.934	(0.928, 0.939)
db2	(2,11)	0.936	(0.930, 0.942)
db3	(4,8)	0.925	(0.909, 0.940)

Table 3.4: DWT-based H estimations for the OctExt1s.mat trace.

3.3 WPT-based estimation

3.3.1 WPT of LRD processes

In Equation (3.7), we expressed the variance of the DWT coefficients at scale j as a function of the scaling parameters. Since WPT is a generalization of DWT, it makes sense now to generalize that expression by calculating the variance of the WPT decomposition of a LRD process. The result is exposed in the following proposition and in the subsequent proof.

Proposition 3.1. *Given an* ideal *LRD process in which the spectrum follows exactly (not only asymptotically) the following spectrum*[1]*,*

$$S(f) = c_f \ |f|^{-\alpha}, \forall f$$

the variance of the WPT coefficients of such a process at scale j and subband m satisfies

$$\mu_{j,m} = E\{d_X(j,k,m)^2\} = 2^{j\alpha} \ c_f C \ \frac{(m+1)^{1-\alpha} - m^{1-\alpha}}{2^{1-\alpha} - 1} \tag{3.34}$$

[1]The long-range dependence property in the frequency domain is given by Equation (1.12):

$$S(f) \sim c_f \ |f|^{-\alpha}, \quad f \to 0$$

where c_f, C and α are the same quantities defined for DWT; from Equation (3.7), this can be equivalently expressed in terms of the variance μ_j of the DWT coefficients as

$$\mu_{j,m} = \mu_j \ \frac{(m+1)^{1-\alpha} - m^{1-\alpha}}{2^{1-\alpha} - 1} \tag{3.35}$$

Proof. The power of subband m at scale j can then be calculated as

$$P_{j,m} = \int_{m\ 2^{-j}}^{(m+1)\ 2^{-j}} S(f)\ df = \frac{c_f}{1-\alpha}\ 2^{-j(1-\alpha)}\ [(m+1)^{1-\alpha} - m^{1-\alpha}]$$

The variance of the WPT coefficients can now be written as

$$\mu_{j,m} = P_{j,m}\ 2^j = 2^{j\alpha}\ \frac{c_f}{1-\alpha}\ [(m+1)^{1-\alpha} - m^{1-\alpha}]$$

Setting $m = 1$ in the last equation, we obtain the DWT case:

$$\mu_{j,1} = \mu_j = 2^{j\alpha}\ \frac{c_f}{1-\alpha}(2^{1-\alpha} - 1)$$

Moreover, from Equation (3.7) we know that in the DWT decomposition it holds

$$\mu_j = 2^{j\alpha}\ c_f C$$

Putting the above equations together, we finally obtain

$$\begin{aligned}
\mu_{j,m} &= 2^{j\alpha}\ \frac{c_f}{1-\alpha}\ [(m+1)^{1-\alpha} - m^{1-\alpha}] \\
&= 2^{j\alpha}\ \frac{c_f}{1-\alpha}\ (2^{1-\alpha} - 1)\ \frac{(m+1)^{1-\alpha} - m^{1-\alpha}}{2^{1-\alpha} - 1} \\
&= 2^{j\alpha}\ c_f C\ \frac{(m+1)^{1-\alpha} - m^{1-\alpha}}{2^{1-\alpha} - 1} \\
&= \mu_j\ \frac{(m+1)^{1-\alpha} - m^{1-\alpha}}{2^{1-\alpha} - 1}
\end{aligned}$$

which is the desired result. ■

If we wanted to extend this proposition to a general LRD process whose spectrum follows Equation (1.12), an extra additive term $A(f)$ should be included in expression (3.35) in order to include the bias caused by the asymptotical behavior.

3.3.2 The Wavelet Packet variance map

In Section 2.5, we have seen how the WPT can be obtained by a generalization of the DWT. In the same way, we can now extend the DWT-related analysis tools discussed in Section 3.2 to the WPT environment. In order to achieve this purpose, we will assume that the same idealizations about the wavelet coefficients (*i.e.*, their Gaussianity, stationarity and independence) hold true.

By analogy with Equation (3.10) and Equation (3.21), we firstly define the following quantities:

$$\widehat{\mu_{j,m}} = \frac{1}{n_j} \sum_{k=1}^{n_j} d_X(j,k,m)^2 \tag{3.36}$$

$$y_{j,m} = \log_2(\widehat{\mu_{j,m}}) - g_{j,m} \tag{3.37}$$

where $g_{j,m}$ is the the generalization of the corrective term g_j defined in Equation 3.23. Since g_j depends only on n_j, the number of wavelet coefficients at scale j, we assume $g_{j,m}$ to be dependent on $n_{j,m}$, the number of wavelet coefficients at scale j and subband m.

It is evident that the three-dimensional plot of $y_{j,m}$ against j and m is a generalization of the LogScale Diagram y_j defined in Section 3.2.6, in the sense that the correspondent LD can be obtained by "slicing" it by the $m = 1$ plane:

$$y_j = y_{j,(m=1)} \tag{3.38}$$

We call this plot the *Wavelet Packet variance map* [RS04b]. Examples of WP variance maps are given in Figure 3.5 and Figure 3.6. The first plot is for the 131072-samples FGN synthetic trace with Hurst parameter $H = 0.8$ discussed in Section 3.2.9 (`fgn08.mat`), while the second is for the `BC-pAug89` Bellcore trace aggregated at $\delta = 10$ msec (`pAug8910ms.mat`), described in Section 3.2.10. In both cases, the WPT decomposition was performed using the `db1` wavelet.

3.3.3 Analytic elements: the WPT-based estimator

We now discuss how the WP variance map can be used for the Hurst parameter estimation. Putting together Equation (3.37) and the logarithms of both members of Equation (3.35), we obtain

$$\begin{aligned} E\{y_{j,m}\} &= \log_2(\mu_j) + \log_2\left[\frac{(m+1)^{1-\alpha} - m^{1-\alpha}]}{2^{1-\alpha} - 1}\right] \\ &= j\alpha + \log_2(c_f C) + \log_2\left[\frac{(m+1)^{1-\alpha} - m^{1-\alpha}]}{2^{1-\alpha} - 1}\right] \end{aligned}$$

This relation suggests that it is possible to estimate α (or, equivalently, H) in different ways:

- via a *global* estimation, by performing a weighted curve fitting (the generalization of the weighted linear regression) across j and m.
- via a *partial* estimation with a *fixed m*, by setting m and performing a weighted linear regression across j, being $m = 1$ the DWT case;
- via a *partial* estimation with a *fixed j*, by setting j and performing a weighted linear regression across m;
- via a *partial* estimation with a *general choice* of subbands and decomposition levels.

Regarding the weights, we apply the same reasoning presented in [AV98] and make them equal to the inverse of the variance of the y_j. For our experiments (reported in the subsequent two paragraphs) we will perform:

- a *weighted linear regression* for the partial estimation, via the *polyfit3* Matlab function;
- a *least square curve fitting* for the global and general estimations. Unfortunately, we found no weighted curve fitting for Matlab, and we could only implement a *weightless* version of our algorithm. As we will see, this affects our estimator with some bias for the higher values of H.

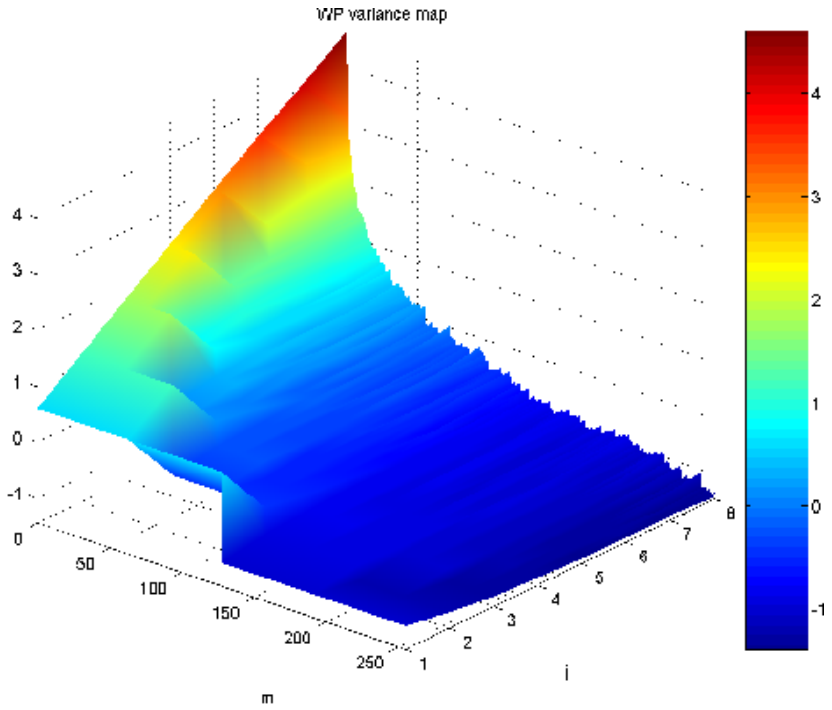

Figure 3.5: Wavelet Packet variance map for the `fgn08.mat` synthetic trace, using `db1` wavelet, for $J = 6$.

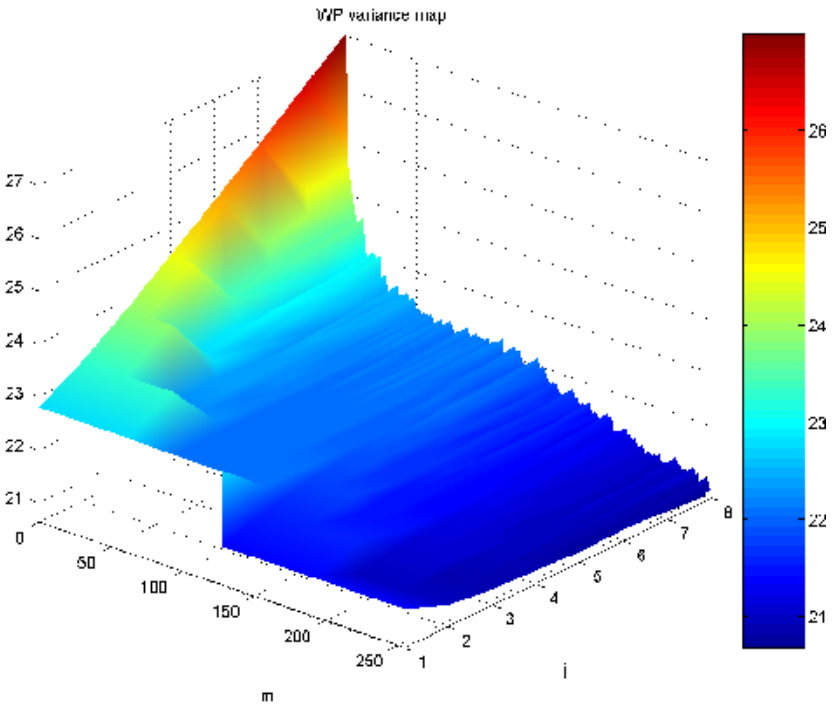

Figure 3.6: Wavelet Packet variance map for the `BC-pAug89` Bellcore trace aggregated at $\delta = 10$ msec, using `db1` wavelet, for $J = 6$.

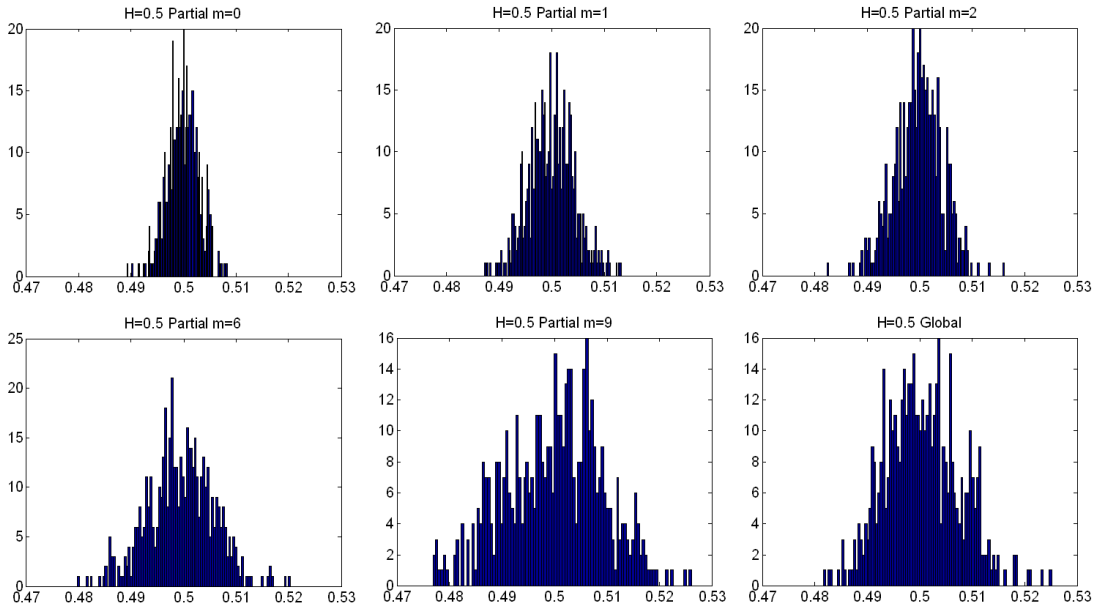

Figure 3.7: Histogram of the H estimation of 1000 independent realizations of FGN ($H = 0.5$), with `db1`, for $J = 10$.

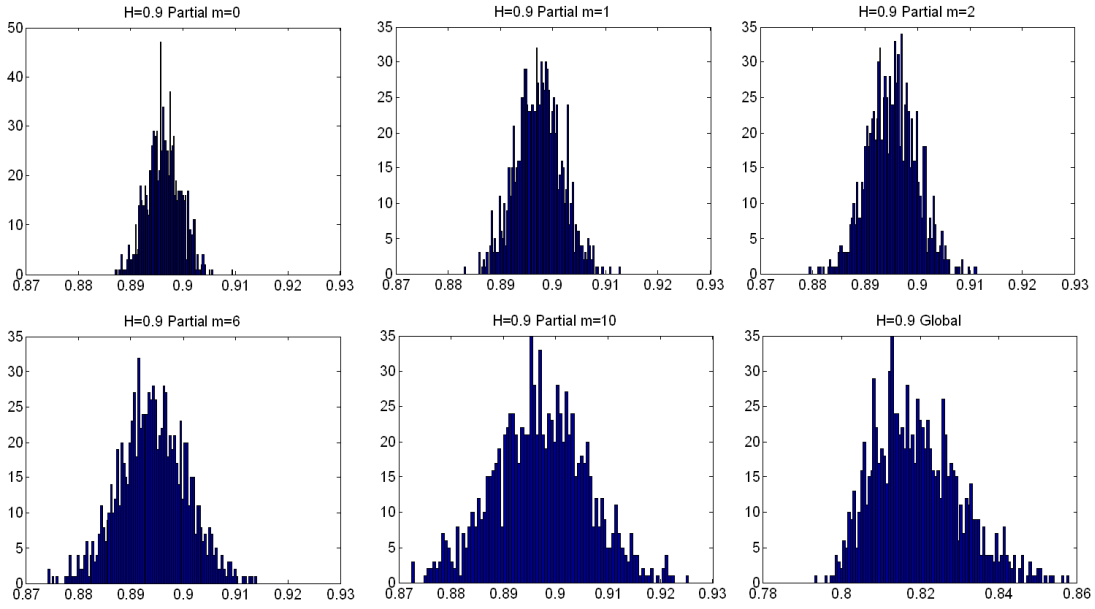

Figure 3.8: Histogram of the H estimation of 1000 independent realizations of FGN ($H = 0.9$), with `db1`, for $J = 10$.

3.3.4 Application to synthetic FGN traces

We now apply the two WPT-based estimators (partial and global) to the FGN synthetic traces `fgn05.mat`, `fgn07.mat`, `fgn08.mat` and `fgn09.mat` discussed in Section 3.2.9. The corresponding results are summarized in Table 3.5. The total number of available scales was $J = 10$ (due to the computational complexity involved), while the chosen wavelet family was the `db1`. Since the dependence on the wavelet was already studied for the DWT-based LogScale Diagram, finding almost the same results, we will only present here the Haar wavelet.

Trace	`fgn05.mat`	`fgn07.mat`	`fgn08.mat`	`fgn09.mat`
Partial ($m = 0$)	$\widehat{H} = 0.4991$	$\widehat{H} = 0.6993$	$\widehat{H} = 0.7995$	$\widehat{H} = 0.8997$
Partial ($m = 1$)	$\widehat{H} = 0.5029$	$\widehat{H} = 0.7034$	$\widehat{H} = 0.8036$	$\widehat{H} = 0.9039$
Partial ($m = 2$)	$\widehat{H} = 0.4954$	$\widehat{H} = 0.6948$	$\widehat{H} = 0.7946$	$\widehat{H} = 0.8944$
Partial ($m = 3$)	$\widehat{H} = 0.5035$	$\widehat{H} = 0.7037$	$\widehat{H} = 0.8038$	$\widehat{H} = 0.9040$
Partial ($m = 4$)	$\widehat{H} = 0.5022$	$\widehat{H} = 0.7028$	$\widehat{H} = 0.8031$	$\widehat{H} = 0.9034$
Partial ($m = 5$)	$\widehat{H} = 0.4918$	$\widehat{H} = 0.6910$	$\widehat{H} = 0.7906$	$\widehat{H} = 0.8902$
Partial ($m = 6$)	$\widehat{H} = 0.4989$	$\widehat{H} = 0.6988$	$\widehat{H} = 0.7990$	$\widehat{H} = 0.8993$
Partial ($m = 7$)	$\widehat{H} = 0.5032$	$\widehat{H} = 0.7026$	$\widehat{H} = 0.8024$	$\widehat{H} = 0.9021$
Partial ($m = 8$)	$\widehat{H} = 0.5034$	$\widehat{H} = 0.7045$	$\widehat{H} = 0.8050$	$\widehat{H} = 0.9057$
Partial ($m = 9$)	$\widehat{H} = 0.5063$	$\widehat{H} = 0.7086$	$\widehat{H} = 0.8097$	$\widehat{H} = 0.9108$
Global (j, m)	$\widehat{H} = 0.4991$	$\widehat{H} = 0.6968$	$\widehat{H} = 0.7826$	$\widehat{H} = 0.8434$

Table 3.5: WPT-based H estimations for four FGN synthetic traces.

Looking at the table, we can see that the partial estimations give very good results for all traces, though accuracy decreases when m grows. As far as the global estimations are concerned, their precision seems to decrease as H takes values close to one, due to the bias introduced by the weightless curve fitting. The higher scales (whose sample variance estimation is very unreliable) bias the estimation.

The data presented in the previous paragraph, though valid, is not necessarily statistically meaningful. That is why we performed the same analysis with 1000 independent FGN realizations of 131072 samples for each value of H, using `db1` wavelet at $J = 10$. We will present only the results for $H = 0.5$ and $H = 0.9$, since those obtained for $H = 0.7$ and $H = 0.8$ fall between them. Figures 3.7 and 3.8 show the histograms of the obtained estimations of H for some of the partial decompositions and the global estimation, while Table 3.6 provides the mean and the variance of the estimations.

	H=0.5		H=0.7		H=0.8		H=0.9	
	Mean $\widehat{H}$	Std dev $\widehat{\sigma}$	Mean $\widehat{H}$	Std dev $\widehat{\sigma}$	Mean $\widehat{H}$	Std dev $\widehat{\sigma}$	Mean $\widehat{H}$	Std dev $\widehat{\sigma}$
Partial ($m = 0$)	0.4998	0.0030	0.6964	0.0031	0.7962	0.0031	0.8963	0.0031
Partial ($m = 1$)	0.4999	0.0043	0.6971	0.0045	0.7970	0.0044	0.8973	0.0043
Partial ($m = 2$)	0.4998	0.0045	0.6955	0.0044	0.7951	0.0045	0.8951	0.0046
Partial ($m = 6$)	0.4995	0.0063	0.6948	0.0062	0.7942	0.0063	0.8942	0.0064
Partial ($m = 9$)	0.4999	0.0093	0.6967	0.0089	0.7966	0.0088	0.8963	0.0090
Global (j, m)	0.5005	0.0071	0.6991	0.0079	0.7796	0.0068	0.8192	0.0110

Table 3.6: WPT-based estimations for 1000 independent realizations of FGN.

The conclusions about the mean values of H are the same as those already explained for the individual traces: the partial estimations are quite accurate, even for $m = 9$. The histograms and the standard deviation show that the higher the m, the more dispersed the results, as is expected due to the lower number of subbands used in the estimation. For the global estimator, the mean H suffers from the bias phenomenon already mentioned in the analysis of isolated traces. Regarding its variance, it is comparable to that of the partial estimations with higher m. Comparing the results with those obtained with the DWT and shown in Table 3.2, and regarding the dispersion of the partial estimates, WPT performs better for $m = 0$, equal for $m = 1$ (what is not surprising, since the $m = 1$ case is exactly the DWT, and only slightly worse for the higher values of m. This validates our algorithm and opens the way for the development of a fast, non-orthogonal, incomplete subband, but still accurate estimators of the scaling parameter, for example by using the $m = 6$ decomposition.

3.3.5 Application to the Bellcore traces

Performing the WPT-based estimation on the `BC-pAug89` trace described in Section 3.2.10, we obtained the results summarized in Table 3.7. As in the previous paragraph, we used the `db1` wavelet family and set $J = 10$.

Partial											Global
m	0	1	2	3	4	5	6	7	8	9	j, m
$\widehat{H}$	0.7735	0.7871	0.8458	0.6650	0.7748	0.8407	0.8513	0.6202	0.7112	0.7549	0.8154

Table 3.7: WPT-based H estimations for the `BC-pAug89` trace aggregated at $\delta = 10$ ms.

Since we expected an estimated H value close to 0.8, results are not very good for the partial estimators ($0.6650 < H < 0.8513$), but the problem does not come from the analysis method, that has been validated previously. Somehow, the different choice of subbands for each estimation is telling us that this trace is not as easy to characterize as the FGNs studied previously. As we will see in Chapter 5, the problem comes from the nonstationarity of the scaling parameters, which makes the variances at each scale to change continuously. The global estimator, on the other hand, still returns a Hurst parameter which is close to the *mean* value of 0.8.

Regarding the `BC-OctExt` trace, the estimations of H are shown in Table 3.8. The conclusions are similar to the ones already found for the other trace: the variation range found in the partial estimations ($0.9008 < H < 0.9468$) is an evidence of the non-stationarity of H. The global estimator, on the other hand, suffers from the same phenomenon already detected in Section 3.3.4 for the $H = 0.9$ FGN traces, in which the estimation was biased to a lower value.

Partial											Global
m	0	1	2	3	4	5	6	7	8	9	j, m
$\widehat{H}$	0.9432	0.9289	0.9008	0.9142	0.9468	0.9167	0.9232	0.9017	0.9239	0.9338	0.8593

Table 3.8: WPT-based H estimations for the `BC-OctExt` trace aggregated at $\delta = 1$ s.

3.4 Summary of the chapter

This chapter has reviewed the properties of the DWT-based estimator of the fractal parameters of network traffic, developed by Abry and Veitch and based on the LogScale Diagram. The key is the analysis of variance (ANOVA) performed by the wavelet transform, which turns the power-law spectral density of self-similar and LRD processes into a straight line when represented in a log-log diagram, making it much easier to estimate.

Being our aim the development of an algorithm to track the evolution of the scaling parameters of traffic, we turned our attention towards alternative wavelet transforms capable of adapting themselves to changing conditions. The wavelet packet transform (WPT) is the first such transform.

We have developed a WPT-based estimator, built on a graphic tool called the *Wavelet Packet variance map* [RS04b, RS04a]. Our intention was to use jointly the real-time splitting algorithm mentioned in 2.5.3, capable of performing a progressive orthogonal tiling, which would update the best basis for the WPT decomposition as the traffic process evolves, and then perform the WPT estimation at the basis change points [VA99]. We found some problems and later abandoned the idea and turned our attention towards the variance change detection techniques described in Chapter 4.

However, the WPT can stand by itself as a contribution, and provides several modes of estimating the scaling parameter of processes. Our WPT-based algorithm provides both partial (in the sense of limited to DWT-style decompositions with fixed m) and global (for all values of j and m) estimations. We have shown that the estimator can actually work with any combination of subbands, though its accuracy is limited by the lack of implementation of a weighted curve-fitting algorithm. Finally, our WPT-based analysis applied to the Bellcore traces has shown clear evidences of the non-stationarity of their scaling parameters.

The WPT partial estimator paves the way for fast, though not so accurate estimators. In the uncomplete frequency decompositions the number of available samples for the regression is lower, and this speeds the process. Despite their lack of accuracy, these estimations could be useful in certain time-critical situations (where the speed of the estimation is more important than its precision), due to their less intensive computational load. This is a topic for future research.

Chapter 4

Variance change detection methods

4.1 Introduction

Our aim is to develop wavelet-based methods capable of tracking the variance structure of network traffic processes and detecting their changes, thus segmenting traffic into regions with homogeneous behavior. We approach the problem by decomposing the original process in a multiresolution analysis via the wavelet transforms described in Chapter 2, and then applying a *variance change points algorithm* to each one of the scales in which the signal is decomposed. From all the algorithms that can be found in the statistical literature, we have selected the following two:

- The *Iterated Cumulative Sum of Squares* (ICSS), described in Section 4.3.
- The *Schwarz Information Criterion* (SIC), presented in Section 4.4.

The ICSS consists of computing the cumulative sum of squares of the samples and comparing it with the result provided by an ideal variance-homogeneous sequence. It is a well-known technique, but lacks flexibility in the selection of the significance level α of the estimation (critical values c_α are computed using Monte Carlo simulation), and it is difficult to build a real-time implementation. The Schwarz Information Criterion (SIC), based on information theory concepts, is a more powerful approach. The intuition behind SIC is that a sequence with a variance change point has higher entropy (disorder) than a sequence with constant variance. The SIC formalism includes an analytic expression for the critical value c_α.

This chapter begins with the statement of the variance change detection problem, together with an auxiliary technique known as the Binary Segmentation procedure. The chapter follows with the description of the ICSS and SIC methods, including examples of how they work and an empirical assessment of their statistical power.

4.2 Statement of the variance change detection problem

Let $x_1, x_2, \ldots, x_N$ be a sequence of independent normal random variables with parameters (η_1, σ_1^2), (η_2, σ_2^2),..., (η_N, σ_N^2). Assume that $\eta_1 = \eta_2 = \ldots = \eta_N = \eta$ and that η is *known*. The general formulation of the problem consists of testing the null hypothesis H_0

$$\sigma_1^2 = \sigma_2^2 = \ldots = \sigma_N^2 = \sigma^2 \tag{4.1}$$

versus the alternative H_1

$$\sigma_1^2 = \ldots = \sigma_{k_1}^2 \neq \sigma_{k_1+1}^2 = \ldots = \sigma_{k_2}^2 \neq \ldots \neq \sigma_{k_q+1}^2 = \ldots = \sigma_N^2$$

where q is the *unknown* number of change points and $1 < k_1 < k_2 < \ldots < k_q < N$, $k_i \in \mathbb{N}$ are the *unknown* positions of the change points.

4.2.1 Binary segmentation procedure

Finding simultaneously the quantity and the position of multiple change points is a challenging problem. There is a method, known as *binary segmentation*, which simplifies the problem. Based on the same principle as the iterative procedure proposed for the ICSS (see Section 4.3), the binary segmentation can be described as follows:

1) look for a single change point in the whole sequence.

2) if there is no change, stop the algorithm.

3) if there is a change, then this change point divides the original sequence into two subsequences.

4) for each subsequence, detect a change point, as in the first step, and continue the process until no more changes are found in any of the subsequences.

From the above discussion, it should be clear that in order to use the binary segmentation, we only have to decide between two hypotheses:

- no change point is detected.

- a single change point is detected.

Thanks to the binary segmentation procedure, we can redefine the problem stated in Section 4.2 as testing H_0 against the following alternative H_1:

$$\sigma_1^2 = \ldots = \sigma_{k_0}^2 \neq \sigma_{k_0+1}^2 = \ldots = \sigma_N^2 \tag{4.2}$$

where $1 < k_0 < N$ is the (unknown) position of the change point.

The single change point detection problem is far easier to solve, as is shown in the following sections for the ICSS and SIC cases.

4.3 The ICSS algorithm

We now discuss a procedure for the variance changes detection called Iterated Cumulative Sum of Squares (ICSS) [IT94]. Introduced in the context of financial series, it studies the detection of multiple changes of variance in a sequence of independent observations. The algorithm uses centered and normalized cumulative sums of squares to search for change points systematically at different pieces of the signal to be analyzed (in our case, the series of the wavelet detail coefficients at a given scale j).

4.3.1 The cumulative sum of squares (CSS) statistic

Expressing the above in a formal manner, let x_k be a series of $k = 1 \dots N$ independent random variables with zero mean and variances σ_k^2, $k = 1, \dots, N$. The cumulative sum of squares (CSS) statistic D_k is defined as

$$D_k = \left| \frac{\sum_{i=1}^{k} x_i^2}{\sum_{i=1}^{N} x_i^2} - \frac{k}{N} \right|, \quad k = 1, \dots, N \tag{4.3}$$

so that, by definition, it holds $D_N = 0$. Furthermore, we assume $D_0 = 0$.

The plot of D_k against k will oscillate around zero for series with homogeneous variance. When there is a sudden change in variance, the slope of D_k suffers a drastic change, creating either a peak or a valley, according to whether the variance changes to a smaller or greater value. Some boundaries for D_k have to be defined for the statistic in order to detect the real change points; i.e., we have to define some critical level in order to distinguish between random oscillations (for peaks under the critical level) and real variance changes (peaks greater than the critical value).

Under constant variance, the normalized statistic $\sqrt{N/2}D_k$ behaves asymptotically like a Brownian bridge (see Definition 4.1), for which a table with the associated critical levels D^* can be found in [IT94]. The table is expressed in terms of the probability p of misclassifying a real change point as a negligible variation. For example, the critical value of $\sqrt{N/2}|D_k|$ for a maximum error of $1 - p = 0.95$ is $D^* = 1.358$.

Definition 4.1. *Given a Brownian motion process $W(t)$, with $E[W(t)] = 0$ and $E[W(t)W(s)] = s, \quad 0 \leq s < t$, a Brownian bridge $W^0(t)$ is defined as follows:*

$$W^0(t) = W(t) - tW(1) \quad 0 \leq t \leq 1$$

for which $E[W^0(t)] = 0$, $E[W^0(t)W^0(s)] = s(1-t)$ for $0 \leq s < t$, and $W^0(0) = W^0(1) = 0$.

4.3.2 Multiple change point detection with the ICSS

With the aforementioned statistic and its associated critical values we can meaningfully detect a single variance change in a time series. If one is looking for multiple variance changes, as is our case, one can perform a binary segmentation procedure. But the authors of [IT94] warned about the

masking effect of the CSS statistic, that may cause some true change points to be missed, and also detect some untrue changes. That is why Inclán and Tiao propose a heuristic, systematic procedure derived from the binary segmentation that iteratively re-evaluates each change point candidate. This is called the Iterated CSS (ICSS).

Let $x[k_1 : k_2]$ denote the subseries $x_{k_1}, x_{k_1+1} \dots x_{k_2}$, and let $D_k(x[k_1 : k_2])$ denote the CSS statistic computed on the $x[k_1 : k_2]$ subseries, for $k_i \in \mathbb{N}$. Then, the iterated version of the algorithm is defined as follows:

Step 0: Let $k_1 = 1$.

Step 1: Compute $D_k(x[k_1 : N])$, and take k^* as the point where $\max_k |D_k(x[k_1 : N])|$ is attained.

Step 2: Compute the statistic $M(k_1 : N) = \max_{k_1 \le k \le N}(\sqrt{(N - k_1 + 1)/2}|D_k(x[k_1 : N])|)$. If $M(k_1 : N) > D^*$, consider that k^* is a change point and continue with Step 3a; if not, there is no evidence of a variance change in the series and the algorithm stops.

Step 3a: Let $k_2 = k^*$. Compute $D_k(x[k_1 : k_2])$, the CSS applied to the subseries $x[k_1 : k_2]$. If $M(k_1 : k_2) > D^*$, a new change point is found. Repeat Step 3a until $M(k_1 : k_2) < D^*$, and therefore no evidence of change in $k_1 \dots k_2$ is found, and the first candidate change point is $k_{first} = k_2$.

Step 3b: Perform a similar procedure starting from the first changepoint found in Step 1, and advance towards the end of the series. Redefine $k_1 = k^* + 1$. Compute $D_k(x[k_1 : N])$ and repeat Step 3b until $M(k_1 : N) < D^*$. Then, define the last candidate change point as $k_{last} = k_1 - 1$.

Step 3c: If the first and last candidates coincide, $k_{first} = k_{last}$, there is just one change point in the series, and the algorithm stops. If $k_{first} < k_{last}$, keep both values as change point candidates, and repeat Steps 1, 2 and 3 in the subseries defined by k_{first} and k_{last}; that is, $k_1 = k_{first} + 1$ and $N = k_{last}$. Each time Steps 3a and 3b are performed, the result can be one or two more candidate change points.

Step 4: Let J_N be the number of change point candidates found, and reorder them in a vector, cp in increasing order. Add the two extreme values $cp_0 = 0$ and $cp_{J_{N+1}} = N$. Re-evaluate each one of the J_N change candidates by computing $D_k(x[cp_{j-1} + 1 : cp_{j+1}]), \quad j = 1 \dots J_N$. If $M(cp_{j-1} + 1 : cp_{j+1}) < D^*$, discard the change point candidate; otherwise, keep it.

Step 5: Repeat Step 4 until the number of change points does not change, and the points found in each iteration are *close* to those on the last pass. Then, the algorithm has converged.

Figure 4.1 illustrates the complete ICSS procedure for a signal composed of 3 segments of 1024 samples of Gaussian white noise (actually, FGN with $H = 0.5$) with variances 1, 2 and 1, and a total length of 3072 samples. The analysis is performed at the 95% significance level, with a shift tolerance (for Step 5) of 1 sample. The signal is shown in a). The first step of the procedure is the computation of the CSS statistic for the whole sequence $(1 : 3072)$, shown (in absolute value) in b)

together with the normalized critical value $\frac{1.358}{\sqrt{N/2}}$. The CSS attains its maximum at $k = 1053$ and exceeds the critical value, thus signalling the presence of a change candidate. Then, the CSS for subseries $(1 : 1053)$ is analyzed (shown in c), finding that the maximum does not exceed the critical value, and therefore determining $k_{first} = 1053$. d) shows the CSS statistic for the $(1054 : 3072)$ subseries, with a maximum at $k = 2048$ that exceeds the critical value. The CSS for the $(2049 : 3072)$ subseries is analyzed in e), with no change points. Therefore, $k_{last} = 2048$. Once both extremes have been found, we study the center of the sequence. f) shows the CSS statistic for the $(1054 : 2048)$ subseries, with no change candidate. At this point we have 2 candidate change points at positions $k_1 = 1053, k_2 = 2048$. We re-evaluate them (Step 4) by computing the CSS statistic for subseries $(1 - 2048)$ for k_1 and $(1054 - 3072)$ for k_2, as shown in g) and h), finding that k_2 is confirmed with no change in its positions, and that k_1 changes to 1081. Since the absolute maximum deviation of k_1 exceeds the shift tolerance, the algorithm performs another iteration where the segments $1 - 2048$ and $1082 - 3072$ are re-evaluated, returning change candidates at positions $k_1 = 1081$ and $k_2 = 2048$. The algorithm has converged and stops.

4.3.3 Complexity

Due to its dependence on the actual number of change points, which in turn forces the algorithm to perform more or fewer iterations, it is not easy to analyze the computational complexity of the ICSS statistic. We have developed a *fast* implementation for Matlab that computes the cumulative sum of squares just once, at the beginning of the procedure. The CSS of the subseries is derived from this first computation just by a substraction and a normalization, making the whole procedure very light in computational terms. From our empirical experience, we hypothesize that the complexity of the algorithm is on the order of the quantity of samples; i.e, $O(N)$.

4.3.4 Empirical assessment of the power of the ICSS algorithm

We now present a simple, empirical approach for assessing the power of the ICSS procedure. It consists of generating 1000 independent realizations of uncorrelated white gaussian noise (WGN) with a variance change point in the middle of the series, for different ratios of variance: $1.2, 1.3, 1.4, 1.5, 1.75, 2, 3$, and 4. These values are of the order of the changes we will see in traffic traces. Each WGN trace has a length of 2000 samples, with the true change point at position 1001. The analyses have been performed at the 95% and 99% significance levels.

The results are shown in Figures 4.2 and 4.3, with the histograms of the detected change points, and Tables 4.1 and 4.2, with the mean and the standard deviation of the change points. The tables also show the amount of "good" change points found in the range of ± 10 positions of the true change (positions $990 - 1010$, corresponding to the variation of the true location of $\pm 0.5\%$ of the series length), and the total quantity of detected change points.

The results highlight the influence of the variance ratio in the change detection. For variance ratios under 1.5, less than half of the changes are detected. Another interesting result is the positive

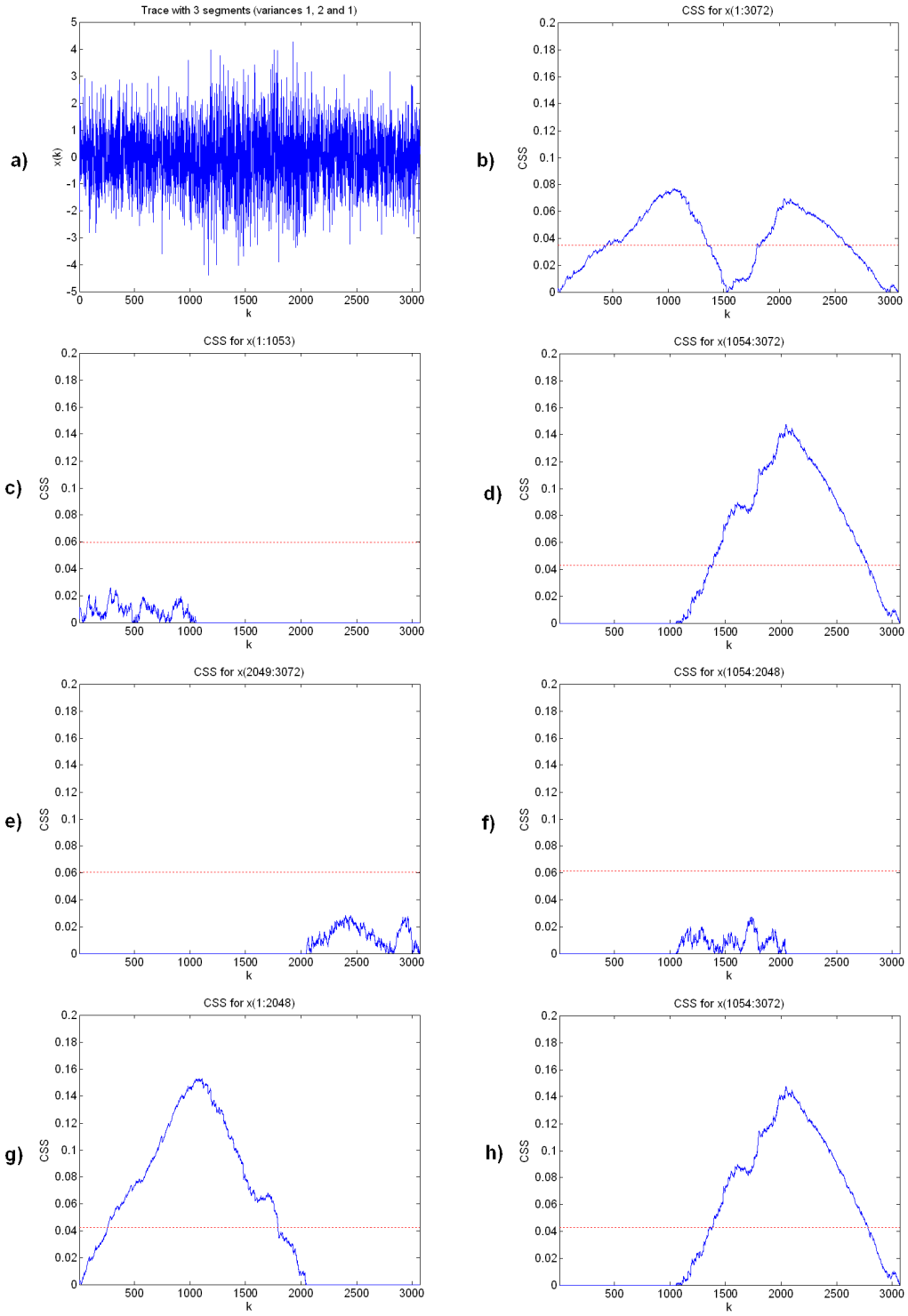

Figure 4.1: The ICSS algorithm applied to 3 segments of 1024 samples of uncorrelated Gaussian white noise (FGN with $H = 0.5$) with variances 1, 2 and 1. The normalized critical value $\frac{1.358}{\sqrt{N/2}}$ is shown as a red line.

bias introduced by the lower variance ratios in the change location. This makes the change to be detected *a posteriori*; i.e, some positions after the true change point, as if the change needed some more samples to become evident, slightly delaying the detection of the change points. The dispersion due to the false change points is also very dependent on the variance ratio; the higher the change, the lower the variance and the better localized the change is. Regarding the significance level, the results obtained with 95% tend to detect more false changes (around 100), while with 99% a lower amount is found (around 20).

Variance ratio	Change points	±10 positions	Mean	Std. deviation
1.2	1072	**206** (19.3%)	1023.0	213.2
1.3	1077	**349** (32.4%)	1021.8	170.2
1.4	1106	**438** (39.6%)	1023.9	177.1
1.5	1094	**521** (47.6%)	1014.1	157.3
1.75	1100	**682** (62.0%)	1016.0	158.6
2	1092	**754** (69.0%)	1001.5	157.2
3	1095	**872** (79.6%)	1002.6	157.2
4	1114	**929** (83.4%)	996.4	168.2

Table 4.1: Results for the detected change points with the ICSS at the 95% significance level.

Variance ratio	Change points	±10 positions	Mean	Std. deviation
1.2	454	**12** (2.6%)	1034.8	202.2
1.3	1016	**330** (32.5%)	1025.8	117.3
1.4	1011	**424** (41.2%)	1022.9	92.4
1.5	1016	**508** (19.3%)	1019.3	89.0
1.75	1021	**681** (66.7%)	1017.0	76.2
2	1021	**794** (77.8%)	1011.1	76.1
3	1020	**888** (87.1%)	1004.0	68.2
4	1025	**935** (91.2%)	1004.5	76.9

Table 4.2: Results for the detected change points with the ICSS at the 99% significance level.

As an illustration of the effect of a correlated signal in the performance of the ICSS statistic, we present in Figure 4.4 a similar analysis, with an FGN signal whose Hurst parameter is $H = 0.9$, thus indicating strong long-range dependence and correlation. The analysis has been performed only for the extreme values of the variance ratio (1.2 and 4), for the two significance levels (95% and 99%). From the histograms it is obvious that a huge quantity of false change points are detected, though it does not depend on the variance ratio. Besides, the false points appear more or less uniformly along the whole trace. For the true points, the detection algorithm performs worse than the uncorrelated input case; i.e., less true changes are detected. What is interesting is the role of the significance level in the quantity of false changes: the higher the significance level, the fewer the false changes.

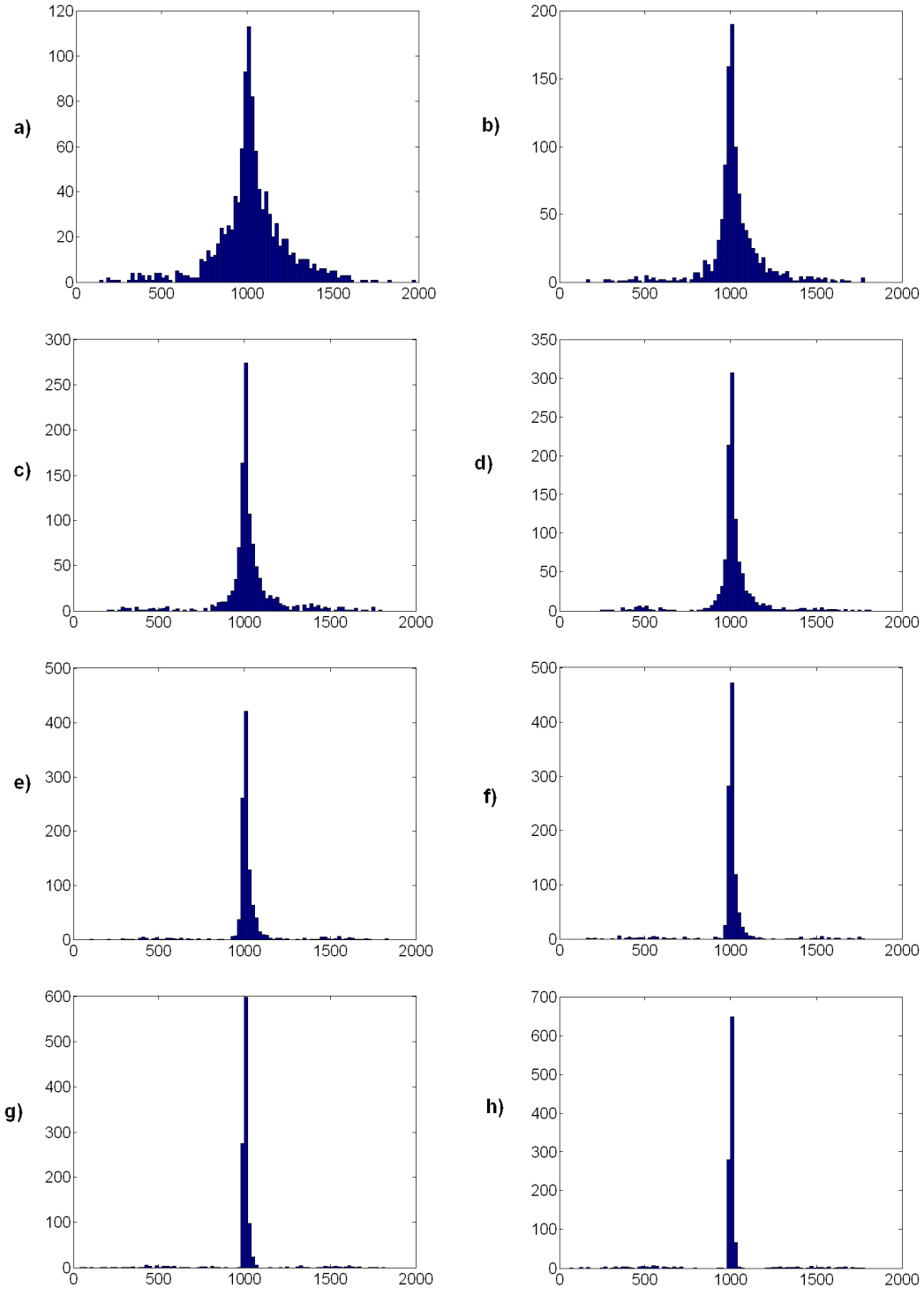

Figure 4.2: Histograms of detected change points with the ICSS at the 95% significance level, for the variance ratios: a 1.2, b 1.3, c 1.4, d 1.5, e 1.75, f 2, g 3, and h 4.

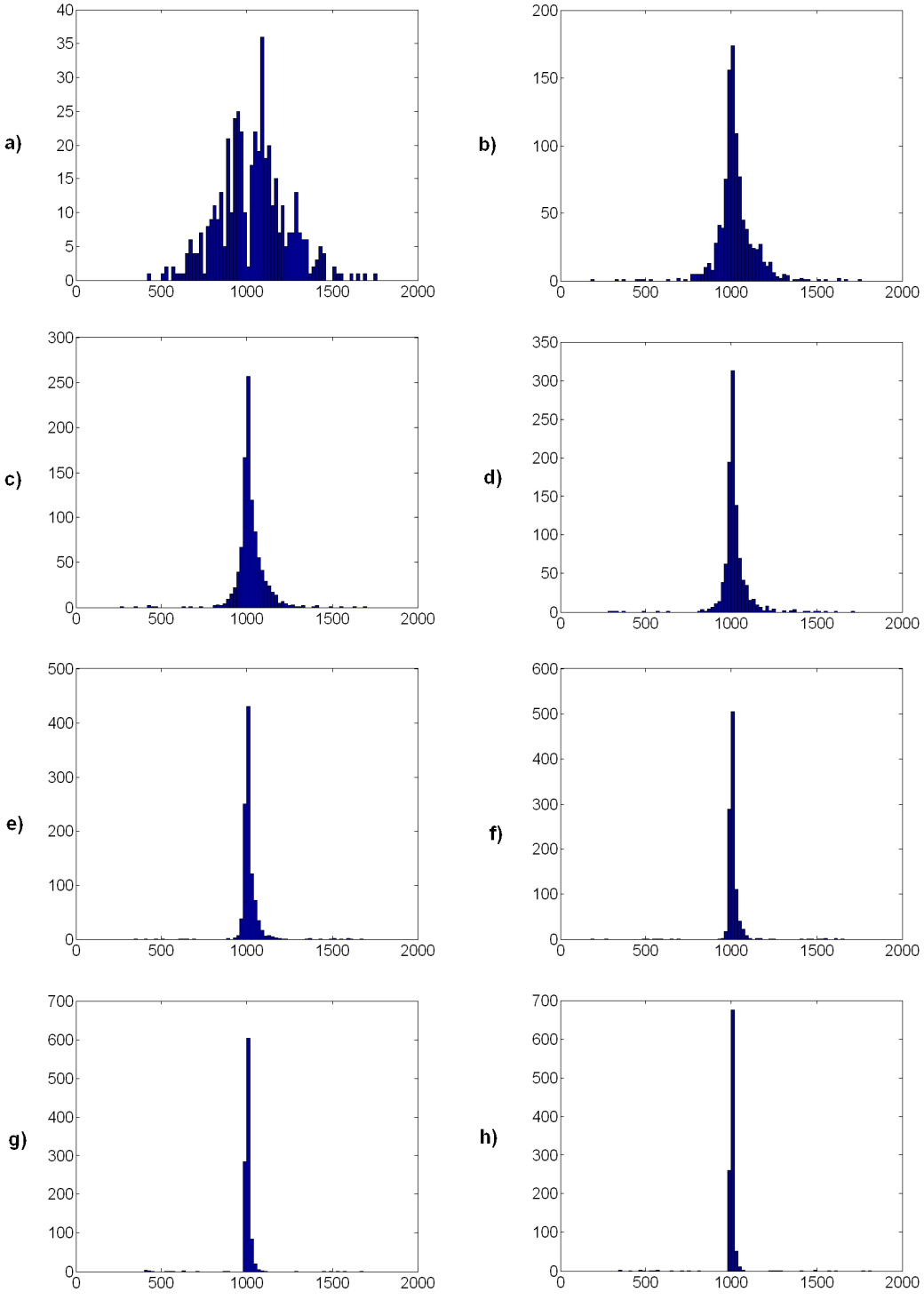

Figure 4.3: Histograms of detected change points with the ICSS at the 99% significance level, for the variance ratios: a 1.2, b 1.3, c 1.4, d 1.5, e 1.75, f 2, g 3, and h 4.

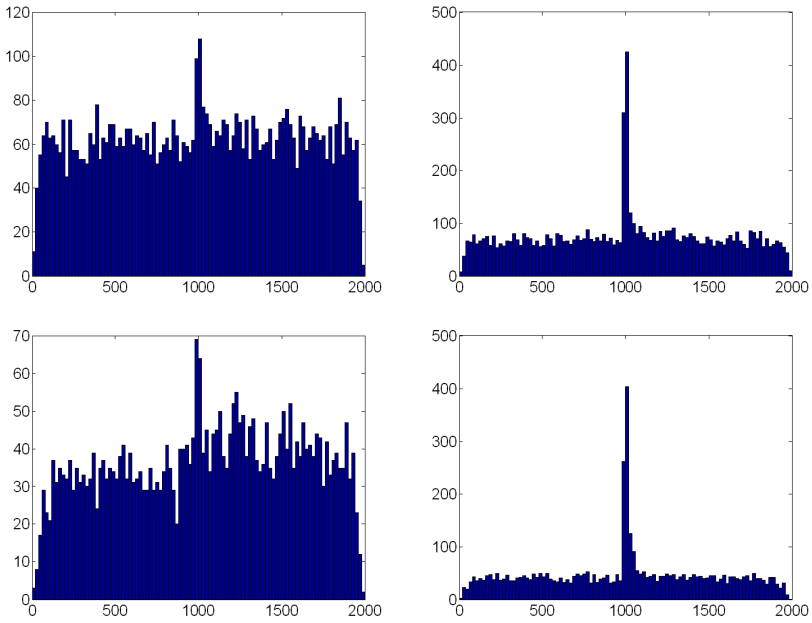

Figure 4.4: Histograms of detected change points with the ICSS on a FGN signal with $H = 0.9$. From top to bottom, left to right: variance ratio 1.2 at the 95% significance level, variance ratio 4 at the 95% significance level, variance ratio 1.2 at the 99% significance level, and variance ratio 4 at the 95% significance level.

4.4 The Schwarz Information Criterion (SIC)

After having analyzed in Section 4.3 the ICSS algorithm, we now discuss another procedure for the variance changes detection, based on the Schwarz Information Criterion (SIC) [CG97]. Introduced in the context of stock price analysis, it studies the problem of multiple variance change points by applying an information theoretic approach.

4.4.1 Information criterion under the Gaussian model

The use of information theory criteria for selecting a model that fits the analyzed data was introduced by Akaike [Aka73], who developed the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC), both widely used in statistical analysis. The statistic that will allow us to decide whether a change point exists or not is the Schwarz Information Criterion (SIC) [CG97], whose general definition is

$$SIC(k) = -2 \ \log L(\hat{\theta}) + p \ \log N, \quad 1 < k < N \tag{4.4}$$

where $L(\hat{\theta})$ is the maximum of the likelihood function for the model (containing the dependence from k), p is the number of free parameters in the model and N is the sample size.

In the following proposition, we derive the expressions for the SIC statistic in the Gaussian case, under the two hypotheses H_0 (no change is present) and H_1 (a single change is present).

Proposition 4.1. *Assuming a Gaussian distribution for the data, the expression of the SIC statistic under the null hypothesis H_0 is given by*

$$SIC(N) = N \ \log 2\pi + N \ \log \hat{\sigma}^2 + N + \log N \tag{4.5}$$

where

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \eta)^2$$

while under the alternative hypothesis H_1 it holds

$$SIC(k) = N \ \log 2\pi + k \ \log \hat{\sigma_1}^2 + (N-k) \ \log \hat{\sigma_2}^2 + N + 2 \ \log N, \ 1 < k < N \tag{4.6}$$

where

$$\hat{\sigma_1}^2 = \frac{1}{k}\sum_{i=1}^{k}(x_i - \eta)^2$$
$$\hat{\sigma_2}^2 = \frac{1}{N-k}\sum_{i=k+1}^{N}(x_i - \eta)^2$$

Proof. Starting from the H_0 case (which corresponds to $p = 1$), the expression for the SIC statistic is given by

$$SIC(N) = -2 \ \log L(\hat{\sigma}^2) + \log N$$

Since in this case the sequence x_i, $i = 1, \ldots, N$ does not possess any change point, the SIC statistic only depends on N. For this reason, we have written $SIC(N)$ instead of $SIC(k)$. As the data follow a Gaussian distribution, that is

$$x_i =_d N(\eta, \sigma^2), \quad i = 1, \ldots, N$$

the probability density function (pdf) for the i^{th} sample is given by

$$f(x_i; \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \ \exp\left[-\frac{(x_i - \eta)^2}{2\sigma^2}\right]$$

The random variables x_i, $i = 1, \ldots, N$ are independent by hypothesis, therefore the joint pdf for the whole data set $\underline{x} = [x_1, x_2, \ldots, x_N]$ (that is, the likelihood function for the model) can be written as

$$f(\underline{x}\,; \sigma^2) = L(\sigma^2) = \prod_{i=1}^{N} f(x_i; \sigma^2) = (2\pi\sigma^2)^{-N/2} \ \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \eta)^2\right]$$

The Maximum Likelihood Estimator (MLE) [SW86] of the quantity σ^2 is then given by

$$\hat{\sigma}^2 = \arg\max_{\sigma^2} \ L(\sigma^2) = \arg\max_{\sigma^2} \ \log L(\sigma^2)$$

Since it holds

$$\log L(\sigma^2) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \eta)^2$$

in order to find $\hat{\sigma}^2$, we have to solve the equation

$$\frac{\partial \log L(\sigma^2)}{\partial \sigma^2} = \frac{N}{2\sigma^4}\left[\frac{1}{N}\sum_{i=1}^{N}(x_i-\eta)^2 - \sigma^2\right] = 0$$

The result is trivially

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\eta)^2$$

Substituting it into the $SIC(N)$ expression finally yields

$$\begin{aligned} SIC(N) &= -2\ \log L(\hat{\sigma}^2) + \log N = -2\left[-\frac{N}{2}\ \log(2\pi\hat{\sigma}^2) - \frac{N}{2}\right] + \log N \\ &= N\ \log(2\pi) + N\ \log\hat{\sigma}^2 + N + \log N \end{aligned}$$

As far as the H_1 case (corresponding to $p = 2$) is concerned, the expression for the SIC statistic is given by

$$SIC(k) = -2\ \log L(\hat{\sigma_1}^2, \hat{\sigma_2}^2) + 2\ \log N, \quad 1 < k < N$$

The data distribution now is

$$x_i =_d \begin{cases} N(\eta, \sigma_1^2) & i = 1, \ldots, k \\ N(\eta, \sigma_2^2) & i = k+1, \ldots, N \end{cases}$$

so that the pdf for the i^{th} sample is

$$f(x_i; \sigma_1^2, \sigma_2^2) = \begin{cases} (2\pi\sigma_1^2)^{-1/2}\ \exp\left[-(x_i-\eta)^2/(2\sigma_1^2)\right] & i = 1, \ldots, k \\ (2\pi\sigma_2^2)^{-1/2}\ \exp\left[-(x_i-\eta)^2/(2\sigma_2^2)\right] & i = k+1, \ldots, N \end{cases}$$

Given the independence of the random variables x_i, $i = 1, \ldots, N$ we can write the likelihood function for the model as

$$\begin{aligned} f(\underline{x}\ ; \sigma_1^2, \sigma_2^2) = L(\sigma_1^2, \sigma_2^2) &= \prod_{i=1}^{N} f(x_i; \sigma_1^2, \sigma_2^2) \\ &= (2\pi\sigma_1^2)^{-k/2}(2\pi\sigma_2^2)^{-(N-k)/2}\ \cdot \\ &\quad \cdot \exp\left[-\frac{1}{2\sigma_1^2}\sum_{i=1}^{k}(x_i-\eta)^2 - \frac{1}{2\sigma_2^2}\sum_{i=k+1}^{N}(x_i-\eta)^2\right] \end{aligned}$$

Performing calculations similar to those done in the H_0 case, for the ML estimators of the quantities σ_1^2 and σ_2^2 it holds

$$\hat{\sigma_1}^2 = \arg\max_{\sigma_1^2}\ L(\sigma_1^2, \sigma_2^2) = \frac{1}{k}\sum_{i=1}^{k}(x_i-\eta)^2$$

$$\hat{\sigma_2}^2 = \arg\max_{\sigma_2^2}\ L(\sigma_1^2, \sigma_2^2) = \frac{1}{N-k}\sum_{i=k+1}^{N}(x_i-\eta)^2$$

Substituting the above expressions into the $SIC(k)$ formula, we finally obtain

$$\begin{aligned} SIC(k) &= -2\ \log L(\hat{\sigma_1}^2, \hat{\sigma_2}^2) + 2\ \log N \\ &= -2\ \left[-\frac{k}{2}\ \log(2\pi\hat{\sigma_1}^2) - \frac{N-k}{2}\ \log(2\pi\hat{\sigma_2}^2) - \frac{k}{2} - \frac{N-k}{2}\right] + 2\ \log N \\ &= N\ \log(2\pi) + k\ \log\hat{\sigma_1}^2 + (N-k)\ \log\hat{\sigma_2}^2 + N + 2\ \log N,\ 1 < k < N \end{aligned}$$

which is the desired result. ■

4.4.2 The minimum information criterion

The decision to accept H_0 or H_1, based on the principle of minimum information criterion, is taken as follows [CG97]:

- if $SIC(N) \leq \min_{1<k<N} SIC(k)$, then accept H_0;
- if $SIC(N) > SIC(k)$ for some k, then estimate the position k_0 of the change point as

$$\widehat{k_0} = \arg \min_{1<k<N} SIC(k) \tag{4.7}$$

The intuition behind the minimum information criterion is the identification of a change in the time series with the presence of *disorder*, which in information theory terms is entropy. What we are doing is computing the entropy of the whole sequence $(1 : N)$, and the entropy of all the possible subsequences $(1 : k)$ and $(k + 1 : N)$. If no change point is present, the entropy of the whole trace is similar to the entropy of the subsequences; but if a variance change point is present at position k, then the entropy of the whole trace is higher than the entropy of both subsequences $(1 : k)$ and $(k + 1 : N)$, and the k that maximizes the difference in entropies is the change position candidate.

4.4.3 Significance level under the Gaussian model

When the SIC values are very close, one may question whether the small differences among them are caused by the fluctuation of the data or by a real change. In order to make the results about the change points statistically convincing, a *significance level* $\alpha > 0$ and its associated *critical value* $c_\alpha \geq 0$ are introduced. Instead of accepting H_0 when $SIC(N) \leq \min_{1<k<N} SIC(k)$, we do it if $SIC(N) \leq \min_{1<k<N} SIC(k) + c_\alpha$. If, on the contrary, H_0 is rejected, then Equation (4.7) is still valid.

Skipping the mathematical details reported in [CG97], the relation between c_α and α is given by

$$1 - \alpha = Pr\{SIC(N) < \min_{1<k<N} SIC(k) + c_\alpha | H_0\} \tag{4.8}$$

while the critical value c_α, under Gaussian assumptions, can be computed (by setting the significance level α) as follows:

$$c_\alpha = \left\{ -\frac{1}{a(N)} \log\log \left[1 - \alpha + \exp\left(-2\, e^{\,b(N)} \right) \right]^{-1/2} + \frac{b(N)}{a(N)} \right\}^2 - \log N \tag{4.9}$$

where

$$\begin{aligned} a(N) &= (2 \log\log N)^{1/2} \\ b(N) &= 2 \log\log N + \frac{1}{2} \log\log\log N - \log \Gamma\left(\frac{1}{2}\right) \end{aligned}$$

The chosen value of α influences the "significance" of the detected variance change points, in the sense that their number increases as α increases.

Our Matlab implementation of the SIC algorithm [Min05] allows for tuning the *resolution* of the SIC algorithm at a given scale by two free parameters:

- the *significance level* α;
- the *minimum segment size* n_{min}, that is the minimum length of every subsequence in which the binary segmentation procedure divides the original trace (this size, in general, can be greater than the minimum distance between two detected change points).

As for the first variable, the value of the second influences the number of estimated change points. In particular, their quantity increases as n_{min} decreases. The conclusion is that the exhaustiveness and the readability of the results founded by the SIC algorithm derive from a *trade-off* between an "enough low" α value and an "enough high" n_{min} value. This will be more clear in Chapters 6 and 7.

4.4.4 Empirical assessment of the power of the SIC algorithm

Following the approach presented in Section 4.3.4, we provide some empirical results on the power of the SIC procedure from the analysis of 1000 independent samples of 2000-samples-long uncorrelated gaussian white noise with a variance change point in the middle.

The results are shown in Figures 4.5 and 4.6 (histograms of the detected change points) and Tables 4.1 and 4.2. Again, the results show the important role of the variance ratio, whose influence on the quantity and quality (in the sense of proximity to the true change) of the detected change points is very high. The 1.5 variance ratio is again the border above which more than 50% of "good" change points are detected. For the 99% significance case the SIC performs better than ICSS at the medium and higher variance ratios, and practically equal for the lower variance ratios, in terms of quantity of "good" change points, mean and specially in the dispersion of points around the mean. For the 95% case the results are not as good, and the ICSS seems to perform better, although at the higher variance ratios the SIC still outperforms ICSS, and even at the lower variance ratios the number of false points of SIC (30-40) is lower than those provided by the ICSS (around 100). In both cases, the bias introduced by the SIC seems to be lower than that of the ICSS, since the mean values obtained by SIC are closer to the true value of 1001.

When a correlated input is given to the SIC, we get similar results to those found with the ICSS. Figure 4.7 shows the results of the analysis of an FGN signal with $H = 0.9$. The main difference with the ICSS case is the decrease in the quantity of detected false changes, for all the situations. Therefore, SIC performs better than ICSS even in the case of correlated input, and what is more interesting, increasing the SIC significance level can somehow compensate the correlation. We will use this fact in Section 7.1.

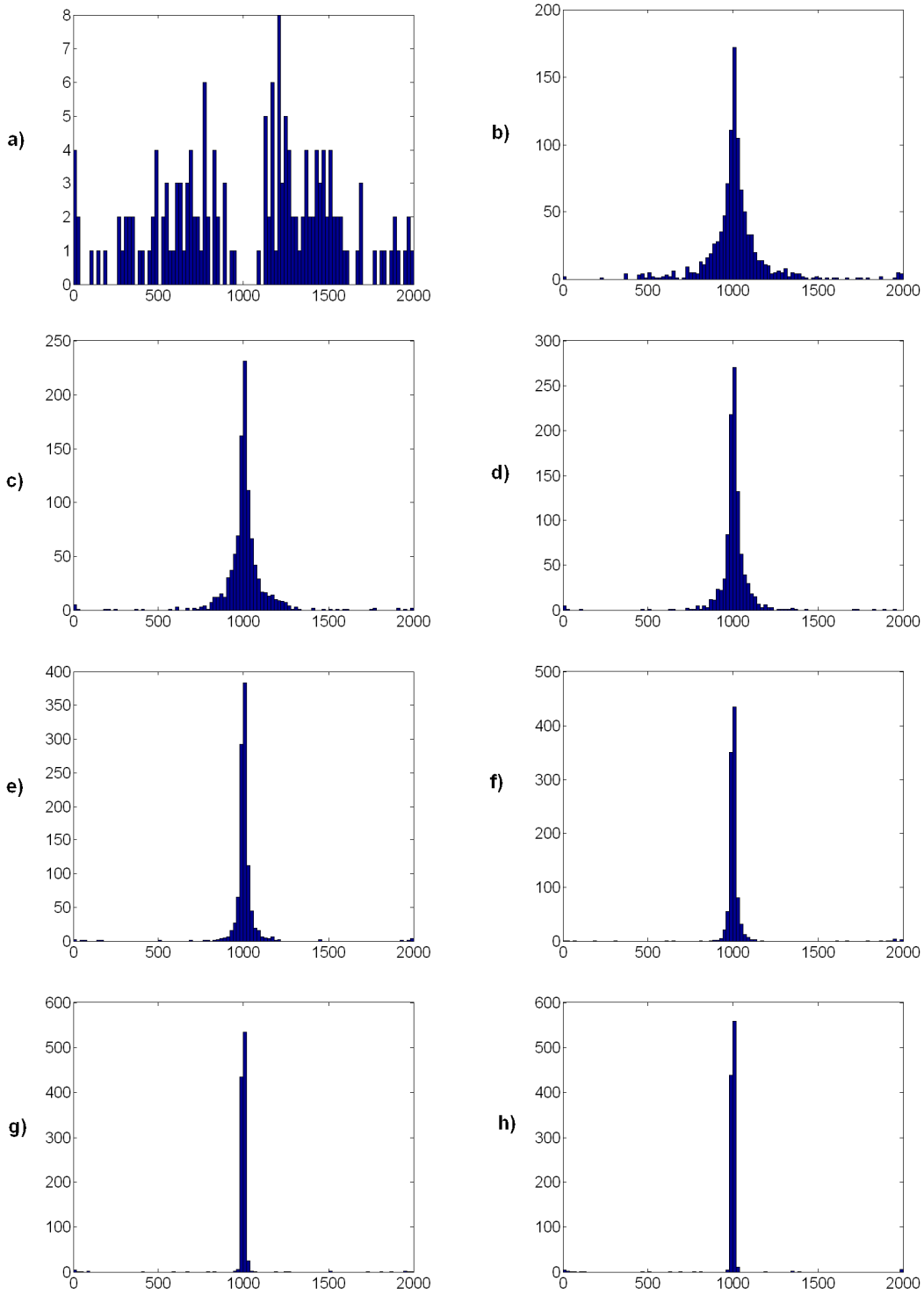

Figure 4.5: Histograms of detected change points with the SIC at the 95% significance level, for the variance ratios: a 1.2, b 1.3, c 1.4, d 1.5, e 1.75, f 2, g 3, and h 4.

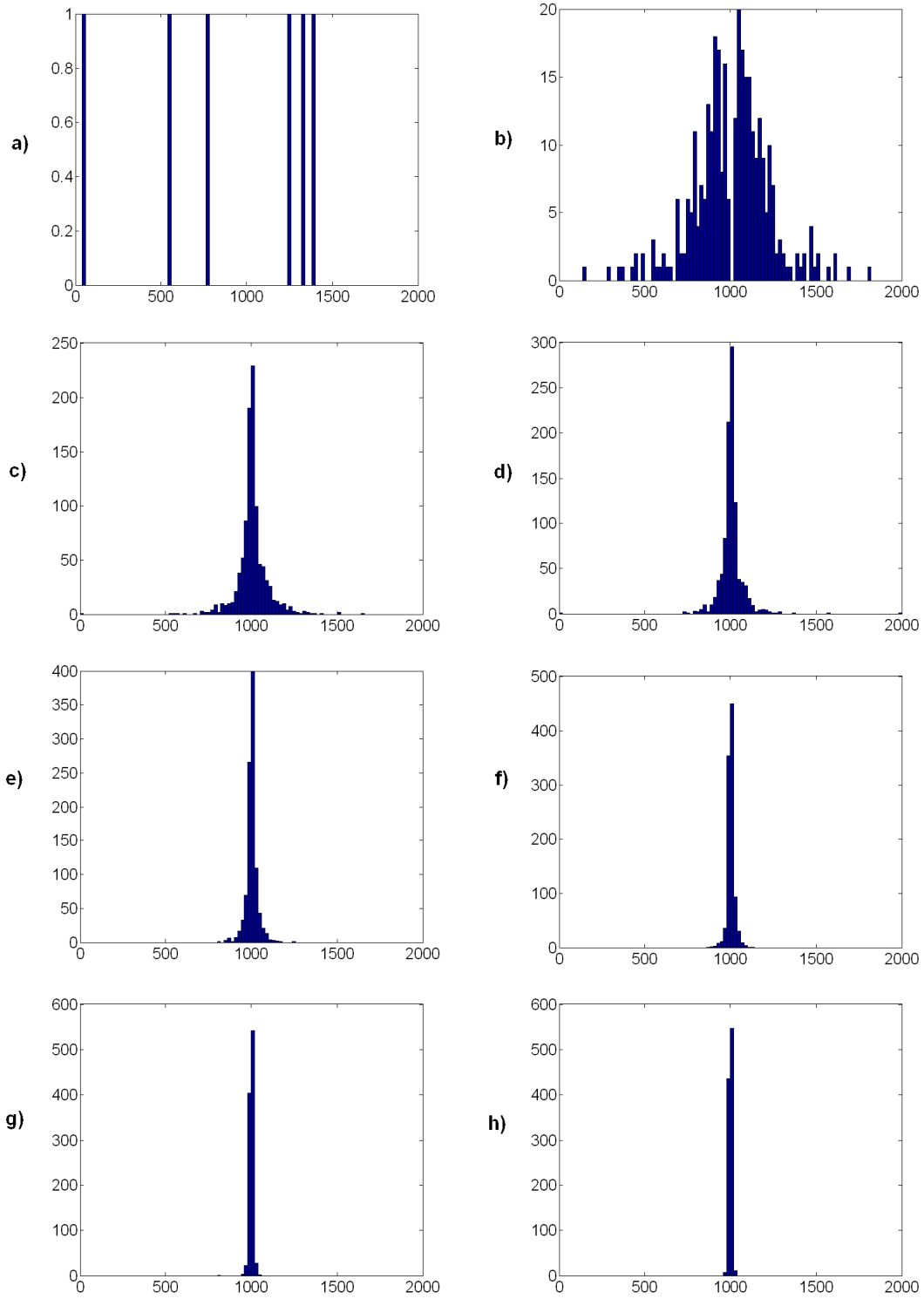

Figure 4.6: Histograms of detected change points with the SIC at the 99% significance level, for the variance ratios: a 1.2, b 1.3, c 1.4, d 1.5, e 1.75, f 2, g 3, and h 4.

Variance ratio	Change points	±10 positions	Mean	Std. deviation
1.2	162	**0**	1032.1	493.7
1.3	1036	**283**	1014.6	188.9
1.4	1033	**393**	1008.9	153.2
1.5	1035	**488**	1005.2	121.6
1.75	1035	**675**	1006.6	112.9
2	1030	**786**	1010.2	121.4
3	1030	**969**	1000.5	119.0
4	1039	**997**	996.5	129.4

Table 4.3: Results for the detected change points with the SIC at the 95% significance level

Variance ratio	Change points	±10 positions	Mean	Std. deviation
1.2	6	**0**	890.5	534.3
1.3	324	**324**	1006.7	224.5
1.4	1001	**419**	1006.8	96.1
1.5	1002	**507**	1006.2	75.9
1.75	1000	**666**	1004.0	34.2
2	1002	**803**	1004.4	21.0
3	1001	**946**	1001.6	10.5
4	1001	**983**	1001.5	5.7

Table 4.4: Results for the detected change points with the SIC at the 99% significance level

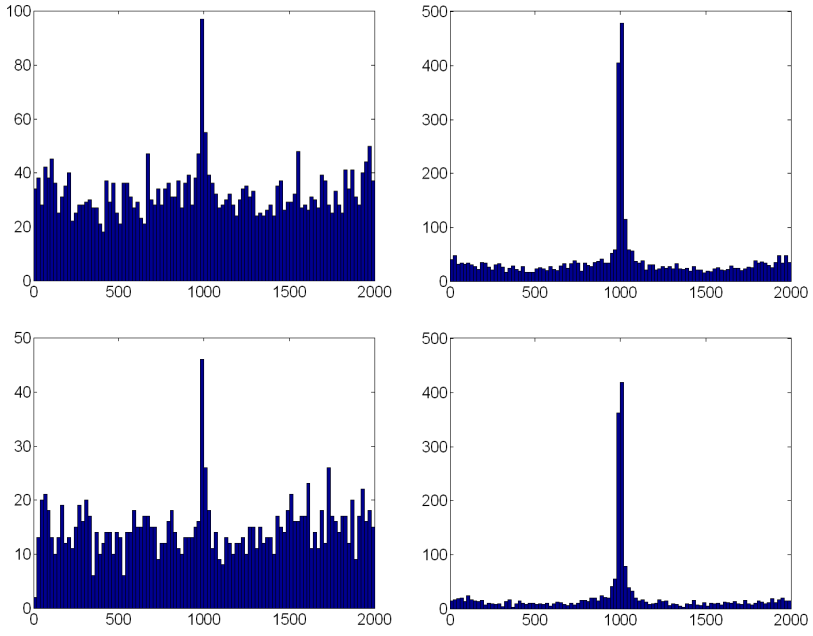

Figure 4.7: Histograms of detected change points with the SIC on a FGN signal with $H = 0.9$. From top to bottom, left to right: variance ratio 1.2 at the 95% significance level, variance ratio 4 at the 95% significance level, variance ratio 1.2 at the 99% significance level, and variance ratio 4 at the 95% significance level.

4.5 Summary of the chapter

We have presented the ICSS and SIC variance change detection algorithms, and have provided some results from an empirical assessment of their power. We have seen that the significance level at which the variance change test is performed is the most influential parameter of the algorithms. The results are also highly dependent on the variance change ratio: the higher variance difference, the better detection ratios we get.

Are any of them better than the other? To begin with, the authors of SIC provided an analytical expression for the significance level, while this parameter can only be obtained by the MonteCarlo method for the ICSS. Regarding the empirical test of the power of the methods, two different situations have been tested: uncorrelated and correlated processes.

- For the uncorrelated case, and at the 95% significance level, ICSS seems to perform slightly better (in terms of "good" change points, mean and dispersion) than SIC at the lower variance ratios, while SIC outperforms ICSS at the medium and higher variance ratios. For the 99% significance level SIC performs almost equal to or better than ICSS. Besides, the (almost always positive) bias introduced by the SIC seems to be lower than that introduced by the ICSS; i.e, the means obtained by SIC are closer to the true value.
- When the input signal is correlated, SIC finds more true changes than ICSS for the 99% significance level, and performs only slightly worse for the 95% case. However, SIC is much better than ICSS in terms of the amount of false changes, at any significance level. This is important, since although the orthogonal Discrete Wavelet Transform theoretically decorrelates completely the wavelet coefficients, in practical situations we have detected residual correlations. Besides, the use of redundant wavelet transforms such as MODWT, DTWT and DDDWT will introduce different levels of correlation in the time series. Therefore, SIC-based methods will be more robust against residual correlation than those that use the ICSS.

Though SIC seems to be better suited for our purposes than ICSS, their performance is not so different so as to abandon the latter. In the next chapter we will study the performance of both variance change methods when applied to the output of the wavelet transforms.

Chapter 5

Segmentation of scaling processes

5.1 Introduction

This chapter argues the need for an algorithm capable of segmenting a scaling process in regions with a homogenous variance structure. Such an algorithm would open the way for detecting transitions between regions with different scaling parameters. We first give a comprehensive overview of the previous works done about this topic by other authors. Then we propose a method based on the multiresolution detection of variance changes and the automatic clustering and alignment of such changes across scales. While the wavelet transform and the variance change detectors have already been described in Chapters 2 and 4, the last piece of our method, an alignment procedure based on the Hough transform, is described in detail in this chapter. We also present some of the implementation issues to be solved regarding each of the steps of our method. The chapter ends with the description of the analysis methodology and the set of traces used in the evaluation of the algorithms.

The main contributions of this chapter are the description of the previous works related to the study of the non-stationarity of the scaling parameters of traffic, and the development of an automatic clustering and alignment detection procedure based on the Hough transform [Min05, RMSP05b, RMSP05a, RMS06, RZM06].

5.2 Nonstationarity of the scaling parameters of traffic

Chapter 3 has addressed the estimation of the Hurst parameter under the hypothesis of *monofractality*, that is, assuming the constancy of the scaling parameters across the entire trace. This is not a good model in real applications since it is difficult to find *stationary* empirical traces with a large number of observations. In other words, long-term measurements of real traffic tend to be *nonstationary*. The causes for this phenomenon can be, for example, the daily, weekly, monthly and even annually patterns of use, as was shown in [PTZD05] for DSL access links and in [RGK+03]

for SNMP traffic, while other studies have found that the superposition of TCP sessions on an uncongested network exhibit intrinsical nonstationarity, as explained in [CCLS01]. In such a complex system as the Internet, with its protocols dynamics and changing behavior of users, it is difficult to identify one single source of nonstationarity.

The changing behavior of the network affects not only the traffic volume, but also other, more subtle characteristics of traffic, such as its scaling properties. Some studies have found that H is correlated with the network load: Ethernet traffic was found to have a higher H during busy periods [LTWW94], while VBR video has shown to exhibit higher H during high-activity periods [BSTW95].

5.3 Previous works

We face a general situation in which network traffic is highly variable both in its volume and its scaling properties, which in turn makes difficult the prediction of network performance (recall the strong influence of scaling on the behavior of queues and buffers, as shown in Section 1.3.2). This fact opens the way to an almost unexplored field of research. The vast existing literature on traffic modeling, and indeed on teletraffic performance analysis in general, is dominated by *scaling-stationary models*. To our knowledge, few works have discussed the nonstationarity of scaling parameters. We can group them as follows: studies that address the influence of non-stationarity on the estimators, on-line estimation methods, and characterizations of traffic whose scaling parameters are not stationary.

5.3.1 Influence of non-stationarity in the estimation of traffic parameters

One of the first studies to tackle the issue of non-stationarity and self-similar traffic was [DLO+94], where the authors warned about the possibility that traffic volume variations could result in misleading estimations of the scaling parameters.

Molnár and his colleagues [MD00, DM99] have been especially active in this field, studying the bias introduced by level shifts, linear and polynomial trends in the performance of the estimators of long-range dependence and self-similarity, such as the R/S statistic and the wavelet-based methods.

Abry, Roughan and Veitch studied the robustness of the LogScale Diagram when mean and variance shifts are present [RV99a], finding that the LD is quite robust, though they assume a constant scaling parameter across the traces. The wavelet transform is capable of filtering polynomial trends, given a mother wavelet with the appropriate number of vanishing moments.

5.3.2 On-line estimation methods

There is an on-line version of the Abry-Veitch estimator that performs a progressive (in the sense of cumulative) computation of subband variance and returns updated estimates using all available

samples [RVA98]. It can work either in a cumulative mode, performing an estimation from the initial time t_0 up to the current time t, or in a constant-length time window mode. The cumulative method actually returns a *mean* value of H with increasing accuracy as time advances, due to the availability of more data for the variance sample estimators, that are updated each time new time samples are captured. The second mode resets the variance counters each time the time window starts. Figure 5.1 shows an example of the output of the progressive LogScale Diagram when applied to a FGN trace with $H = 0.7$. The method does not provide a reliable estimation after a certain number of samples are acquired, and needs some warm-up time until it converges to the true value of the Hurst parameter of the source. The authors of the method estimate the transient period in 4096 samples. In our example each sample corresponds to a bin of $\delta = 10$ ms, and 5000 samples seem to be necessary in order to ensure the convergence of the estimator for this particular trace, thus needing almost a minute in order to get a good estimation.

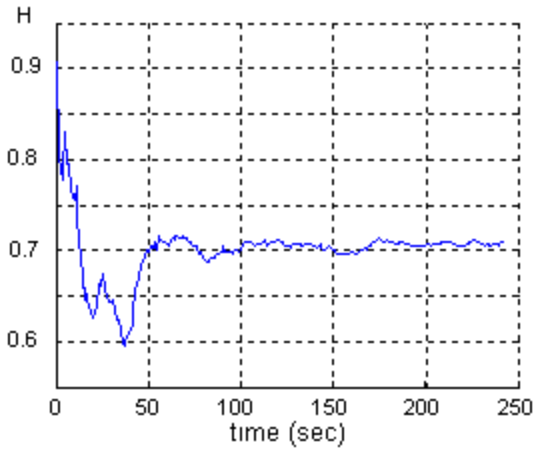

Figure 5.1: Example of the progressive version of the Abry-Veitch estimator, for a FGN source with $H = 0.7$.

Following the aforementioned approach, Uhlig [Uhl02] developed the 3D-LD, an extension of the LogScale Diagram that is able to show graphically the evolution of the scaling properties of traffic. This tool computes the LD over constant-length windows and represents the variance of each scale as a function of time.

Abry and Veitch [VA01] developed a statistical test of the constancy of the scaling parameter over constant-length windows that is able to detect changes in H but is insensitive to changes in the variance structure; i.e., it can detect variations in the slope of the LogScale Diagram but is not able to detect changes in the second order parameter. The algorithm consists of splitting the trace into m segments (windows) with the same length, performing the estimation of the scaling parameters separately ($\alpha_1 \dots \alpha_m$), and testing the constancy of the α_i with a Uniformly Most Powerful Invariant (UMPI) test. The authors study the trade-off between time resolution and the power of the statistic: if m is large the resolution is high, but on the other hand there will be few scales for the LogScale Diagram-based estimation, and therefore the power of the constancy test

will decrease. The test is applied to the Bellcore set of traces (`BC-pAug89`, `BC-pOct89`, `BC-OctExt`, and `BC-OctExt4`), with $m = 12$, showing a very high variability of α along the traces, even for the cases where the statistical test does not reject the null hypothesis H_0 (no change is present). For the `BC-pAug89` trace, in particular, the value of α goes from 0.3 to 0.8, corresponding to Hurst parameters in the range $0.65 - 0.9$. These values are coherent with the measures performed by the same authors in [AV98] (see Section 5.3.3). The main drawback of the statistical test is its limitation to long constant-length windows.

Sanihoglu and Tekinay [ST02] studied the real-time applicability of the wavelet and R/S methods for the estimation of the scaling parameters of traffic. Their work was focused on the constraints to be found in the real implementations of the estimators, such as the wavelet filter length, number of scales, computational complexity and length of buffered traffic. Theirs is a theoretical work based on partially-overlapped windows.

Finally, an accuracy-based approach has been reported for the real-time estimation of multifractality, where the window length is variable and related to the confidence interval of the estimation [AAI05, AAI06]. It is an interesting idea: the method accumulates samples until the confidence interval of the multifractal estimator is below a certain value determined by the user, and then the estimation is performed. The window length automatically adapts to the statistical properties of data (which is also one of the properties of our method, though our criterium for the segmentation is different).

5.3.3 Characterization of scaling-nonstationary traffic

Abry, Roughan and Veitch, in their first paper about the LogScale Diagram [AV98], studied the changes in the Hurst parameter for the Bellcore set by dividing the traces in segments of similar length and applying the estimator to each one of the pieces. They found fluctuations in the value of the Hurst parameter. For example, the values of H for the `BC-pAug89` trace, split in 32 segments with the same length, are in the range $0.7 - 0.9$.

In a more detailed study, Abry, Veitch and Roughan analyzed 6 months of Ethernet traces from a research center with the on-line version of the LogScale Diagram [RV99b]. They found some correlation between the diurnal load cycle and the scaling parameter, though it was dependent on the nature of the traffic (human or machine generated).

Furthermore, a wavelet-based study of TCP flow arrivals revealed nonstationarities, found no strict scaling, and generalized the study to high-order scaling [Uhl04]. It has been reported that some TCP mechanisms, such as Slow Start, Fast Retransmit, Fast Recovery, cumulative ACKs, and ACK compression are the cause of short-time bursts (a group of TCP segments sent back-to-back at line rate) which can impact queuing performance. Jiang and Dovrolis [JD03, JD05] have recently shown that those short-time bursts can be the reason for scaling phenomena at short-time scales, which appears in the LD as a characteristic *elbow* of the slope of the variance alignment and can give birth to two or more differentiated scaling regimes.

5.3.4 Mathematical methods

The idea of applying a variance change detection method to the output of the wavelet transform is not new, though to our knowledge nobody has applied it to the study of network traffic. Whitcher [Whi98] studied the use of the DWT and MODWT transforms together with the ICSS, for the characterization of the nonstationary series of Nile River minimum water levels, vertical ocean shear measurements, and atmospheric phenomena such as the Madden-Julian oscillation.

5.4 Scope of our work

We begin by repeating the experiments performed by Abry and Veitch in [AV98] with the Bellcore `BC-pAug89` trace, split in 32 segments. The authors gave little data about the aggregation level, the wavelet family and the scales used for the regressions. Assuming an aggregation level of $\delta = 10$ msec, $j_1 = 3$, $j_2 = 12$, and using the `db3` wavelet, we get Figure 5.2 (top), where H ranges from 0.67 to 0.86 and each segment corresponds to 9822 samples (98.2 seconds of real time). The analyses with other wavelets (`db1-db4`) return similar results. The data obtained is coherent with the *mean* value of 0.8 for H, though they clearly state that the scaling behavior of the trace is not stationary. The same figure also depicts the mean traffic volume of the Bellcore trace, showing that though there is some positive correlation between volume and H, specifically at the beginning of the trace, there are also counterexamples of such a direct relation; see for example the volume peak around segment 27, which coincides with a stable zone for H.

But why the choice of 32 segments? The same figure shows the results for the 64 segments case, for which the Hurst parameter ranges from 0.63 to 0.92 and each segment corresponds to 4911 samples (49.1 seconds of real time). Though the general shape of the figure is similar to the 32 segments case, there are some values that differ considerably. See for example segments 10, 21, 41 or 58. In the 32 segments analysis the extreme values of H of those segments are masked by the surrounding samples, making H take not so extreme values. Of course we have to consider the confidence intervals (which are inversely proportional to the segment length) when comparing both diagrams; one could think that the values found in the aforementioned segments can be explained by the uncertainty of the estimation. But there are regions that do not change appreciably their behavior, such as the "stable" zone defined by segments 24-34 (64 segments) or by 12-17 (32 segments), with a Hurst parameter stabilized slightly above 0.8.

The other real trace considered in our experiments also belongs to the Bellcore data set: `BC-OctExt`. As described in Section 3.2.10, this trace is known to be much more variable than `BC-pAug89`. When aggregated at $\delta = 1$ second and analyzed with the `db3` wavelet between $j_1 = 4$ and $j_2 = 14$, the LogScale Diagram returns a Hurst parameter of 0.944 (0.932, 0.956). Figure 5.3 shows the 32- and 64-segment analysis [1], which confirms the not-so-clear correlation between traffic

[1] where each segment corresponds to 3838 samples -1 hour and 4 minutes- and 1919 samples -32 minutes- respectively.

volume and Hurst parameter, and also the extreme variability of the scaling parameters of the trace, which includes values far over 1 and under 0.5, confirming its nonstationarity [AV98].

In order to localize better the H-constant regions, we could perform a deeper analysis (128 or more segments) and then join the segments with a similar behavior, but we would find a problem of accuracy caused by the confidence intervals; furthermore, why should the boundaries coincide with multiples of the segment size? Our aim is to develop a method capable of finding the boundaries of the regions with homogenous scaling behavior, without having to perform the constant-length window procedure.

5.5 Description of our method

The main matter of our work is the localization of the instants that bound the segments with homogeneous fractal behavior. Since our estimation methods rely on the computation of the sample variances at each scale, our aim is naturally related to two other problems: the detection of variance change points, and of their alignment in time. After this *change points detection* phase, the estimation of the scaling parameters for each segment can be performed by the usual static estimators described in Chapter 3. Figure 5.4 illustrates the elements that compose our algorithm. Let us explain the different steps of our method with an example.

5.5.1 An example

Let $x(n)$ be a time series composed of two different segments of FGN with the same mean and variance, but with different Hurst parameters, say 0.5 for the first segment and 0.9 for the second one. Without loss of generality, we assume that both segments are of the same length and therefore the change point is in the middle of the series. Figure 5.5 shows such a trace, where the first segment resembles white Gaussian noise, while the second segment exhibits strong low-frequency components.

When the trace is decomposed by the multiresolution analysis (MRA) provided by a wavelet transform at the J level, we get a set of J detail time series and one approximation. Each one of these time series represents the time evolution of its associated subband. In particular, the variance of each subband can be tracked. Figure 5.6 shows the detail coefficients of the FGN 05-09 trace obtained with the Discrete Wavelet Transform. Note the different length of each scale due to the downsampling procedure included in the DWT algorithm.

Figure 5.7 shows the superposition of the LogScale Diagrams for both segments, being the flat LD the one of the $H = 0.5$ segment. This representation does not have any connection to the LD of the *whole* $x(n)$ trace, and is presented here just to make easier for the reader the comparison between the variance distribution across scales. In such a situation, the variance change point algorithm would detect a decrease in the variances of scales $j = 1$ and 2, no change or a slight change in scale $j = 3$, and an increase in the variances at scales $j = 4 \ldots 15$. Regarding the practical inexistence

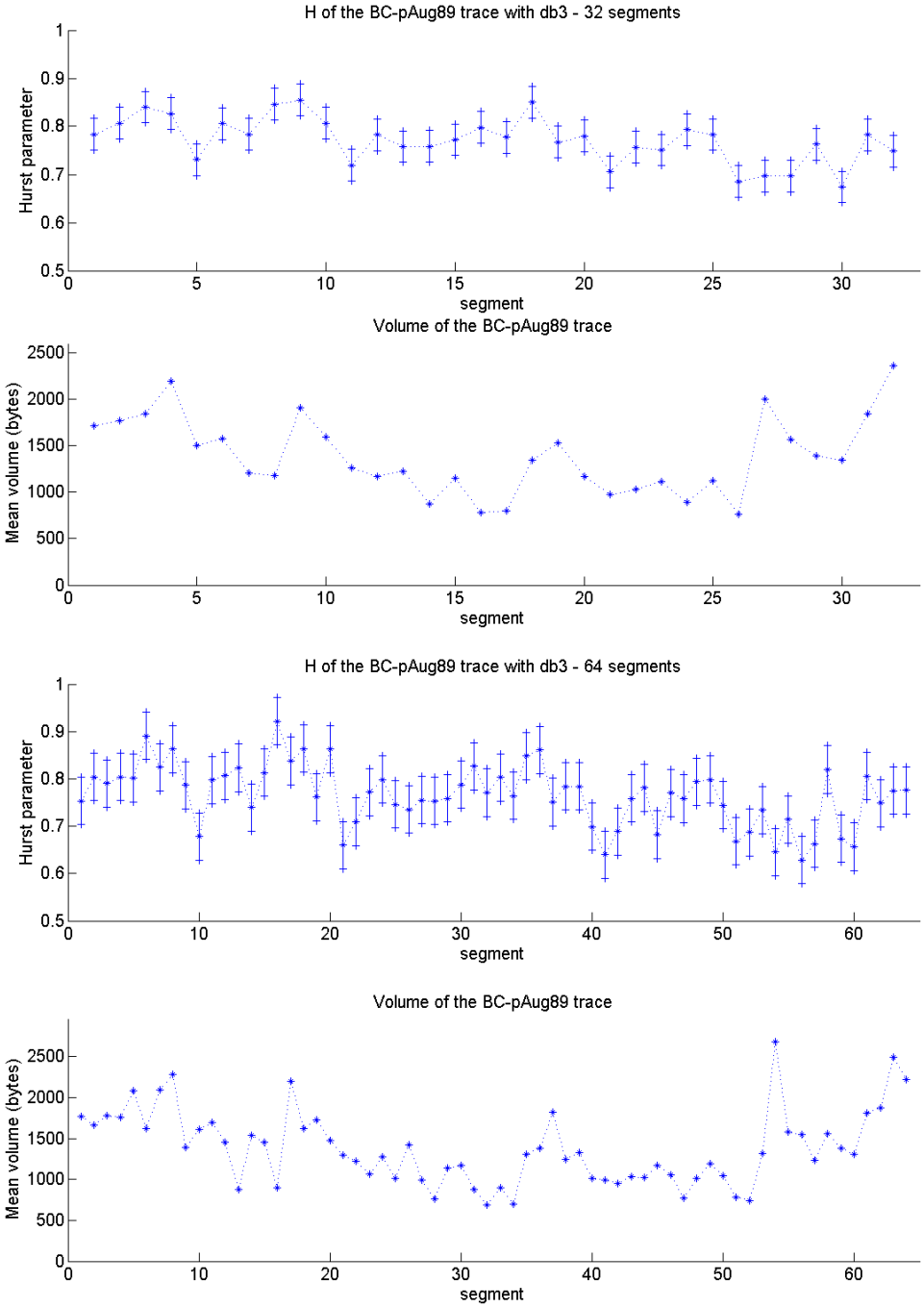

Figure 5.2: From top to bottom: 1) Evolution of the Hurst parameter of the `BC-pAug89` trace, split in 32 segments, with $\delta = 10msec$, $j_1 = 3$, $j_2 = 12$, and `db3`. The vertical lines are the 95% confidence intervals of each estimation. 2) The mean volume of the `BC-pAug89` trace in each of the 32 segments. 3) The Hurst parameter analysis, for 64 segments. 4) The mean volume analysis, for 64 segments.

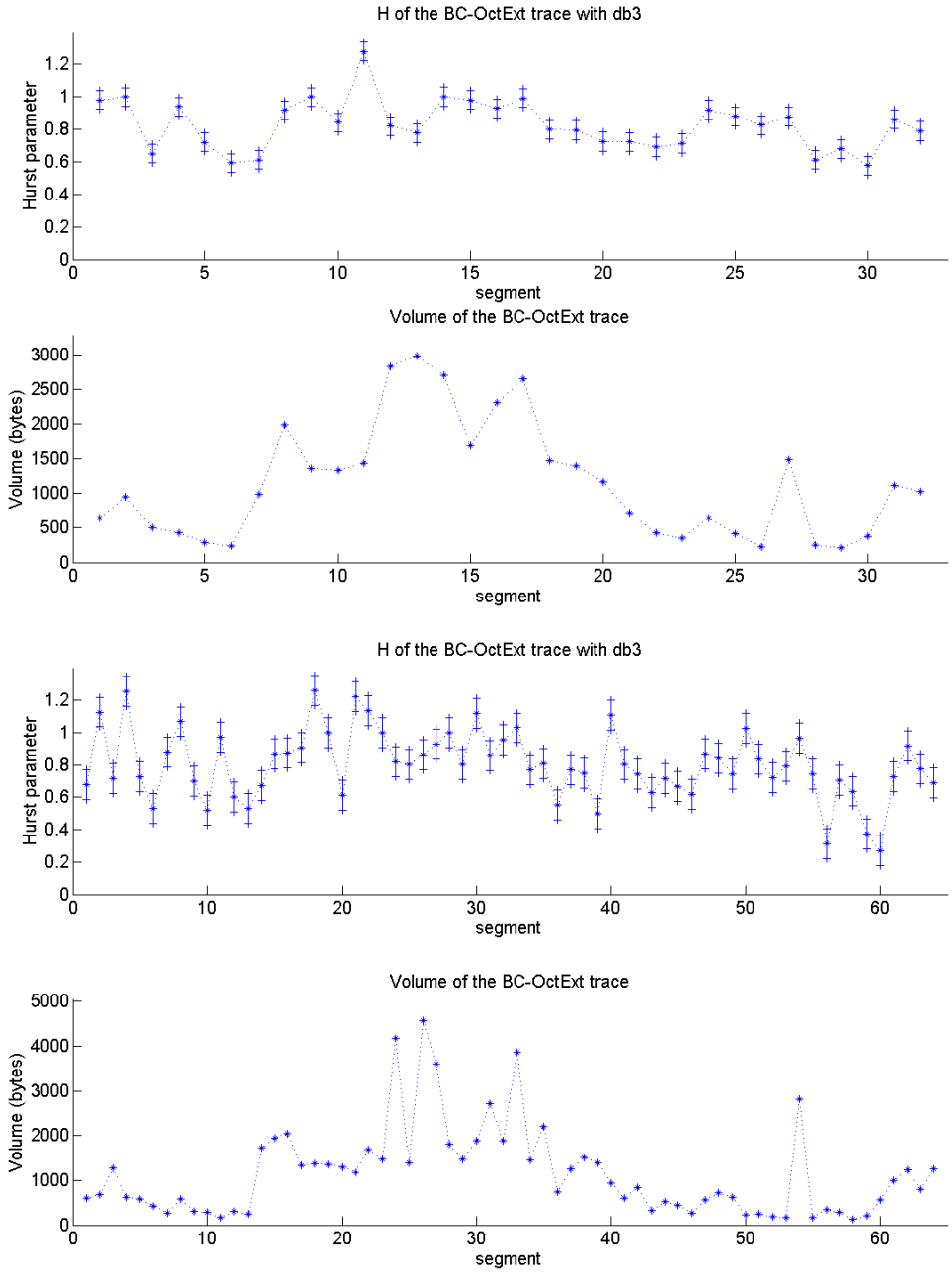

Figure 5.3: From top to bottom: 1) Evolution of the Hurst parameter of the `BC-OctExt` trace, split in 32 segments, with $\delta = 10msec$, $j_1 = 3$, $j_2 = 12$, and `db3`. The vertical lines are the 95% confidence intervals of each estimation. 2) The mean volume of the `BC-OctExt` trace in each of the 32 segments. 3) The Hurst parameter analysis, for 64 segments. 4) The mean volume analysis, for 64 segments.

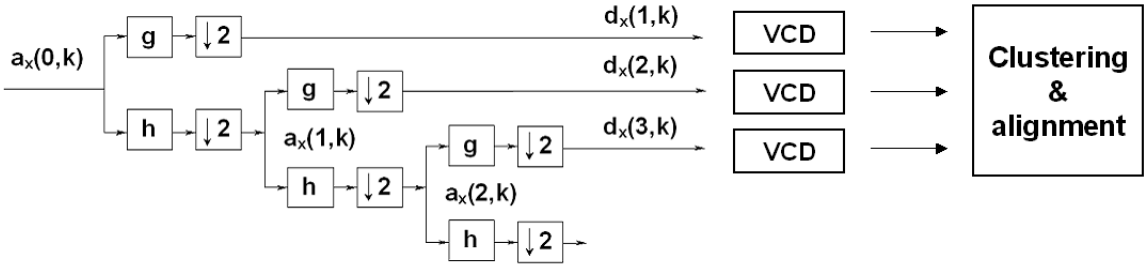

Figure 5.4: Scheme of the joint DWT-variance change detection method. VCD stands for Variance Change Detection algorithm.

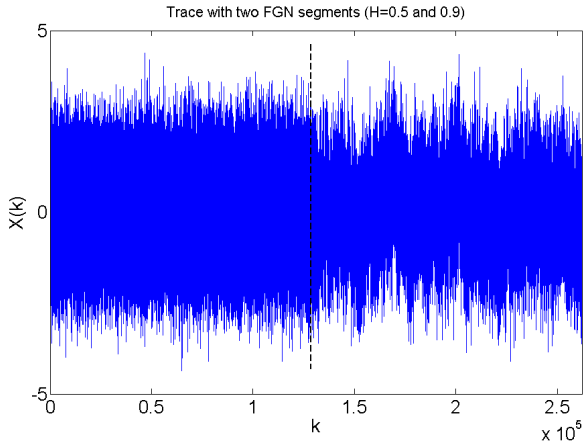

Figure 5.5: Trace composed of two FGN segments with Hurst parameters $H = 0.5$ (left) and $H = 0.9$ (right).

of a variance change at $j = 3$, we call this phenomenon a *blind point*, in the sense that no variance change takes place at that scale. If we represent the location of the change point for each scale, we get a diagram such as the one shown in the left of Figure 5.8.

If we plot not only the position, but also the *type* of variance change (increase or decrease) and its size (the variance ratio), we get the diagram shown in the right of Figure 5.8, where the *orientation* of the triangles denotes the type of change (upwards means increase, downwards means decrease) and the size of the triangle is related to the logarithm of the variance ratio. This representation gives a clear idea of the magnitude of the changes at each scale, but it is not as convenient for visualizing the position of the changes. As expected, the lower scales show a variance decrease, while the higher scales increase their variance. For the third scale, a change is still detected, though of a very low magnitude (almost invisible in the right side of the figure), and its location seems to have been changed to the right of its expected position, probably because of the difficulty experienced by the variance change detection method when working with such a low variance ratio.

Apart from the aforementioned bias in the blind zone, there is also some variation in the localization of the changes due to issues related with the wavelet; for example, the DWT performs

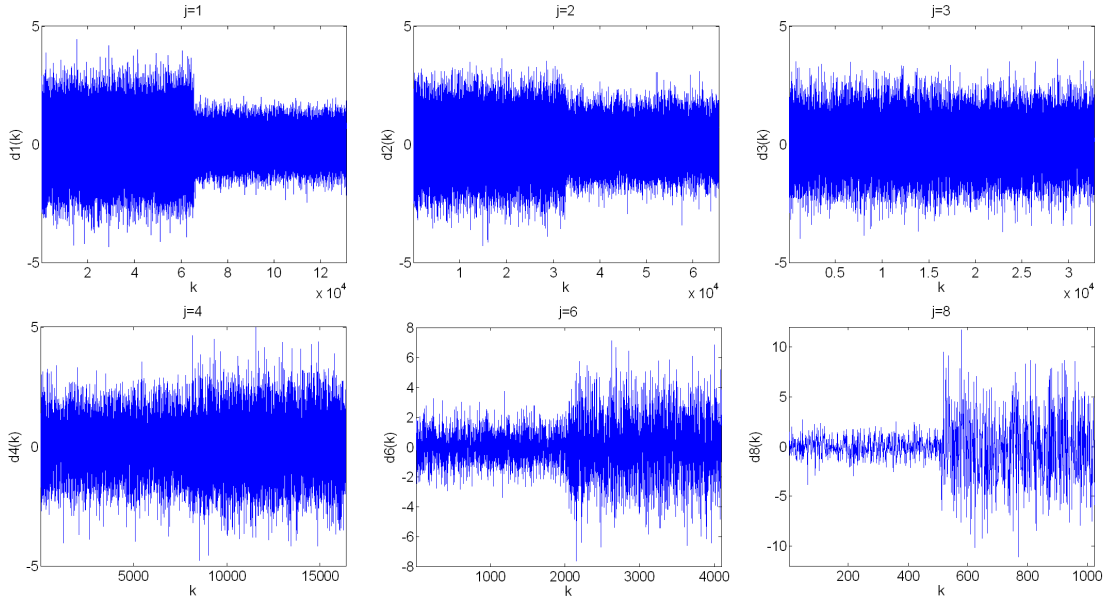

Figure 5.6: DWT detail coefficients versus time for the trace composed of two FGN segments with Hurst parameters $H = 0.5$ and $H = 0.9$.

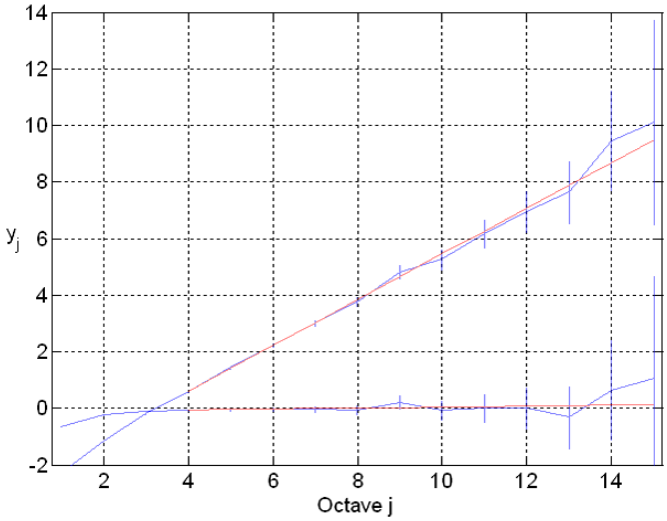

Figure 5.7: Variance distribution at each scale for two FGN traces with scaling parameters $H = 0.5$ (flat spectrum) and $H = 0.9$.

especially bad at the higher scales (see the bias at scale $j = 9, 11$ and 13). Therefore, we need a final step capable of automatically detecting the alignment of the change points and to somehow *correct* the undesired variations by averaging the positions of teh changes detected across scales. The method should therefore be capable of *clustering* the points in groups with an approximately similar x-axis location, and *deciding* which of the change points correspond to a true variance structural change, and then perform the average.

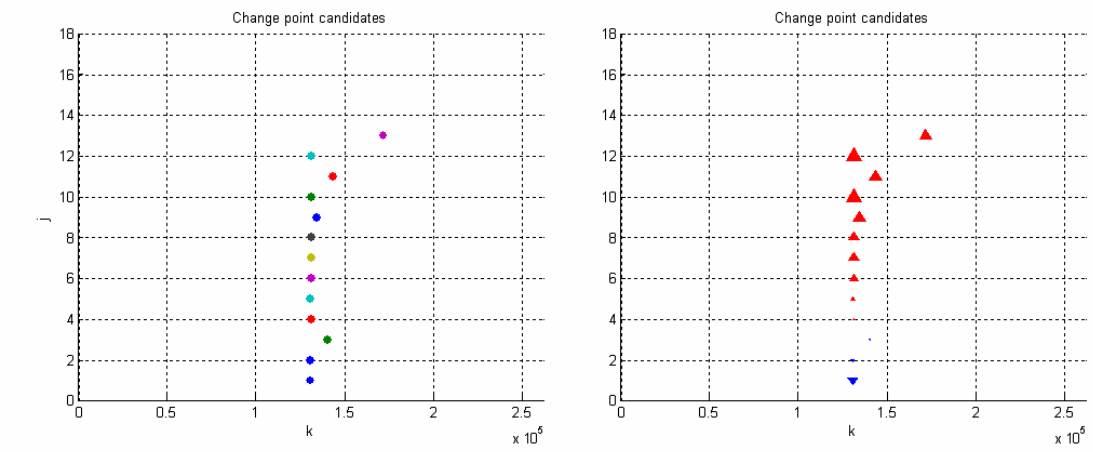

Figure 5.8: Detected variance change points for the trace with a Hurst parameter change ($0.5 \rightarrow 0.9$). *Left*: the position of the change points. *Right*: size and sign of the detected variance change.

5.5.2 Issues related to our method

Summarizing the previous section, our traffic segmenting algorithm is composed of the following steps:

1) Perform a wavelet-based decomposition of the original trace.
2) For every scale, apply an algorithm for *variance change detection* to the wavelet coefficient series.
3) Perform a clustering process in order to identify the change points aligned in time and pass them through a quorum vote. The *simultaneous* detection of changes across an *adequate* number of scales will indicate a change point in the Hurst parameter.
4) Average the positions and provide the initial and final points of the segment.
5) Finally, if required, perform the estimation of the scaling parameters in the detected segments.

Let us now discuss some issues regarding the practical implementation of each of the steps.

Wavelet transform

As far as the first step is concerned, the main issue is the choice of the appropriate wavelet transform. In particular, we have experimented with the wavelet transforms that were presented in Chapter 2. We started with the DWT, but found that its lack of resolution at the higher scales is a serious drawback for locating the lower frequency change points. Furthermore, the lack of time shift invariance made the results dependent on the *phase* of the input trace.

The downsampling causes the number of DWT samples available at each scale to decrease with increasing scale, due to the downsampling discussed in Section 2.4.6. The problem is illustrated in Figure 5.9, where each detail coefficient at the 3^{rd} scale covers the same time segment as $2^3 = 8$ original samples. This prevents us from having a reliable detection of change points at the higher

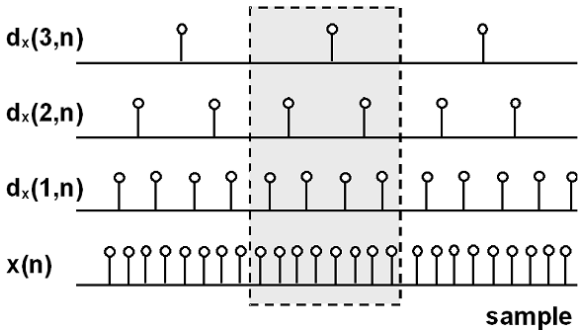

Figure 5.9: Illustration of the loss of time resolution at the higher scales of the DWT, due to the down-sampling process.

scales, since the low-pass filtering smoothes the variance changes and makes the change point be undetected. With regard to that, it is noteworthy to observe that a candidate change point must be detected at an *adequate* number of scales (say, greater than half the number of the available scales) in order to be *elected*.

That is why we tried the MODWT and its associated maximally oversampled analysis. An MODWT-based algorithm provides higher resolution for the detection of change points. At the higher scales, where LRD is actually to be detected, DWT-based algorithms perform badly because of the limited number of samples available. The MODWT corrects this problem at the cost of a larger number of computations and greater memory requirements. Unfortunately, it also introduces correlation at higher scales, contradicting the assumption of independence in ICSS and SIC. These methods assume uncorrelated and Gaussian inputs, and the presence of correlation at the higher scales of the MODWT makes the methods detect false, spurious change points. Since the correlation effect increases with the length of the mother wavelet, the shortest possible wavelet should be used. In our experiments, we have used the Daubechies family, starting with `db1` and increasing the number of vanishing moments until the analysis is stabilized. In most cases, `db2` or `db3` are sufficient.

We tried a couple of approaches in order to overcome the correlation problem. First, we saw that forcing some parameters of the change point detection algorithms (such as its statistical significance and the associated critical value of the statistic involved in the change point detection) we could minimize the number of spurious change points. This approach can be understood as raising the *exigence* for deciding if a change point candidate is to be accepted as a true change or not. This empirical approach raised some questions about the statistical meaningfulness of the whole procedure, though the results obtained were not bad. We later applied a technique called *Equivalent Degrees of Freedom (EDOF)* [PW02, Whi98] that is used with the MODWT in order to correct the correlation phenomenon when one is interested in computing the sample variance of the wavelet coefficients, as is our case. It consists of taking just some of the coefficients, instead of computing the sample variance over all of them. Actually, this is somehow equivalent to our first method; instead of raising the critical level of the statistic that helps us decide if a change point

is valid, we reduce the value of the statistic with the EDOF approach and keep the critical value associated for an uncorrelated signal. However, we found that the EDOF approach did not return good results [ZR06] and abandoned it.

We then shifted our attention to the DTWT, hoping we could find a trade-off between time resolution and correlation, keeping as much as possible the time-shift invariance property.

Apart from deciding on the wavelet transform to be used, other wavelet-related issues must be taken into account. The choice of the mother wavelet is important, since it affects the estimation in several ways. As with any filtering process, there are boundary effects due to the convolution of finite-length sequences that influence the accuracy with which the wavelet variance is estimated; boundary coefficients should not be used. Another effect is the phase shift introduced by the mother wavelet. This can be corrected or minimized by the correct choice of the wavelet family (symlets, for example). There is also a trade-off between wavelet filter length L, computational complexity, and accuracy. Increasing L allows a better estimation of wavelet variance but requires a greater number of computations and involves fewer non-boundary coefficients.

Variance change detection method

The second step of our method is strongly dependent on the choice of the *variance change points algorithm*. Chapter 4 presented the *Iterated Cumulative Sum of Squares* (ICSS), described in Section 4.3, and the *Schwarz Information Criterion* (SIC), described in Section 4.4.

Recall that the main conclusions of Chapter 4 were that ICSS performs slightly better for uncorrelated sources, while SIC is better with correlated signals. Since we expect a residual correlation from the DWT, and different degrees of correlation with the redundant wavelet transforms (MODWT and DTWT), SIC seems to be better suited to our purposes, though we will analyze and compare the performance of both methods.

Alignment detection

Finally, in the third step we have to deal with the selection and *clustering* of the *candidate* change points, in order to detect if a change is present in enough scales and close enough in time such as to consider it as a variance structure change. An image processing technique, built on the Hough transform, can help us. The Hough Transform is used in image processing for detecting features of a particular shape through a polar representation of lines [Rus99].

Given the dispersion around the true change location, we will have to define a *tolerance* around the (unknown) change position, and a *quorum* for a voting process that will decide if enough clustered points are aligned in order to consider it as a Hurst parameter change location.

The following section describes the Hough-based procedure in detail.

5.6 Hough transform-based clustering and alignment

5.6.1 The linear Hough transform

The *Hough transform* [Rus99] can be used to isolate features of a particular shape within a *binary image*. Since it requires the desired features to be specified in some parametric form, the classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. A *generalized Hough transform* can be employed in applications where a simple analytic description of features is not possible. The main advantage of the Hough transform is that it is tolerant of gaps in feature descriptions and is relatively unaffected by image noise.

Our interest will be concentrated on the *linear Hough transform* $H(\rho, \theta)$ of a binary image $I(x, y)$:

$$H(\rho, \theta) = \int_D I(x, y)\ \delta(\rho - x\cos\theta - y\sin\theta)\ dxdy \tag{5.1}$$

where $I(x, y)$ can take the value 0 or 1 (depending if the pixel is white or black), and D is the region (x, y) of the space domain where the binary image is defined.

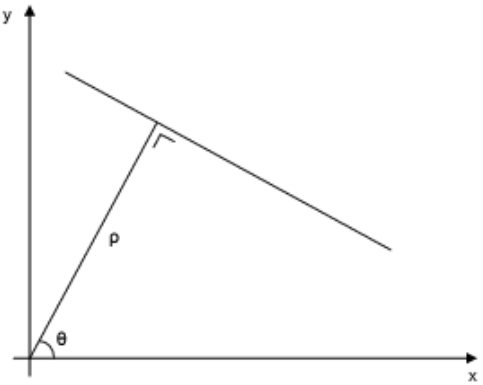

Figure 5.10: Parametric description of a straight line. Reproduced from [Min05].

The main idea underlying the linear Hough transform is the *polar representation* of a line in the (x, y) plane (see Figure 5.10):

$$x\cos\theta + y\sin\theta = \rho \tag{5.2}$$

This equation indicates that the Hough transform maps a *line* in the space domain (x, y) to a *point* in the Hough domain (θ, ρ), where $\rho > 0$ is the length of the normal segment from the origin to this line and $\theta \in [0, \pi)$ is the orientation of ρ with respect to the x axis. In other words, for any point (x, y) on this line, θ and ρ are constant.

The dual interpretation of the above is that a *point* (x_0, y_0) in the space domain is mapped in the Hough domain to a *cosinusoid* given by the equation:

$$\rho(\theta) = \sqrt{x_0^2 + y_0^2}\ \cos\left(\theta - \arctan\frac{y_0}{x_0}\right) \tag{5.3}$$

The conclusion is that points which are collinear in the cartesian image space are easily detected in the Hough space, as they yield curves which intersect at a common (ρ, θ) point.

In order to illustrate the linear Hough transform, Figure 5.11 shows a binary image and its Hough transform. The 250x250 pixels image contains a square whose dimensions are 150x150 pixels; therefore, there are alignments at positions 50 and 200. The images have been inverted for printing purposes; i.e, in the original the background is black (corresponding to value 0) and the rectangle is white (corresponding to 1). The resolution of the Hough transform in both axis ρ and θ is 1 unit, corresponding to 1 pixel and 1 degree, respectively.

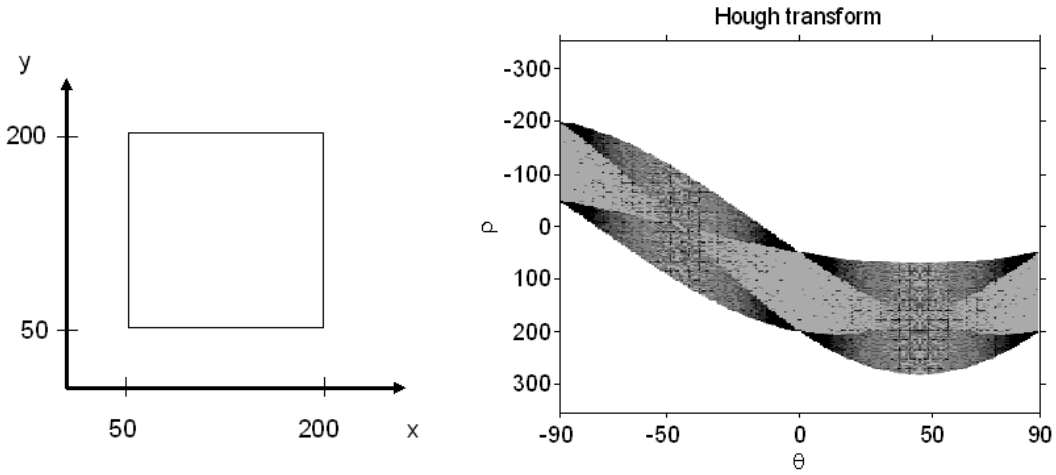

Figure 5.11: Image containing a square and its corresponding Hough transform (*right*). Both images are inverted for printing purposes.

The two bright spots in the $\theta = 0°$ and $\pm 90°$ directions of the Hough domain (which is symmetric by definition) at $\rho = \pm 50$ and $\rho = \pm 200$ provide the parameterization for the lines that compose the square.

As far as the implementation is concerned, the Hough space is quantized into finite intervals or *accumulator cells* (*i.e.*, a multidimensional array). Both θ and ρ resolutions can be tuned (see Section 5.6.3 for an example). As the algorithm runs, each point in the space domain is transformed into a discretized cosinusoid in the Hough domain and the accumulator cells which lie along this curve are incremented. Peaks in the accumulator matrix represent strong evidence that a corresponding straight line exists in the image.

5.6.2 Implementation details

The first issue to be discussed is our Matlab implementation of the clustering procedure, whose main steps are:

1) Generate a digital image of JxN pixels set to 0 (black), where N is the data size and J the number of available scales.

2) If the variance change detection method at scale j ($j = 1 \dots J$) detects a change point positioned at time k ($k = 1 \dots N$), set the correspondent pixel to 1 (white).

3) Perform the Hough transform of the so-obtained image, changing from the (j, k) domain to the (ρ, θ) domain.

4) Select the portion of the accumulator matrix referring to the $\theta = 0$ (vertical) direction, since we are looking for vertically aligned points.

5) Find the ρ_{MAX} coordinates corresponding to the maxima of such array (*i.e.*, to the "better aligned" segments of candidates as change points).

6) Convert every polar coordinate ρ_{MAX} to the corresponding k_{MAX} position.

7) Take the set of k_{MAX} values as the estimated global change points vector $\underline{\widehat{k}}^{\,*}$.

The above exposed clustering procedure may be tuned by three parameters:

- The **time resolution** Δk of the method. The k axis is divided in intervals of length Δk and consider two votes at different scales to be assigned to the same candidate if they fall within the same interval.
- The ***quorum*** Q, which is the minimum number of votes needed by a candidate in order to be elected. It is beyond doubt that the critical choice resides here, since we have to decide how many scales are *enough* to indicate a change in the Hurst parameter.
- The **maximum number** of global change points n_{MAX}. This indicates the maximum number of potential candidates, so that choosing a conservative value (for example $n_{MAX} = 100$) does not affect results at all.

We finally describe the output of the clustering algorithm, which is given by:

- the two-columns matrix H_{votes}, containing the centers of the voted intervals (of size Δk) in the first column and the corresponding number of votes in the second column;
- the vector of the global estimated change points $\underline{\widehat{k}}^{\,*}$.

The computation of the $\underline{\widehat{k}}^{\,*}$ elements is performed by the following steps:

1) Take memory of the detected change points at all available scales by the auxiliary vector $\underline{k_0}$.

2) Select the $n < n_{MAX}$ rows of H_{votes} corresponding to the candidates that reach the *quorum*.

3) Calculate every $\underline{\widehat{k}}^{\,*}$ element as the *corrected mean* of the correspondent detected change points contained in the $\underline{k_0}$ vector. In this way we minimize the dispersion of the change location estimation.

5.6.3 Application to a synthetic trace with changes in the Hurst parameter

We now apply the automatic clustering procedure based on the Hough transform to the trace of Figure 5.12 (left), composed of 3 segments with different Hurst parameters ($H = 0.8$ for $1 \leq k \leq 65536$, $H = 0.9$ for $65537 \leq k \leq 98304$, and $H = 0.7$ for $98305 \leq k \leq 131072$). More precisely, the method is applied to the change points diagram shown in the right side of Figure 5.12, whose Hough transform of the diagram is shown in Figure 5.13. On the left side the resolution for θ is $1°$, while the right side has been computed with a resolution of $60°$ and one of the bins centered at $0°$, which corresponds to the X axis. The second representation makes it easier to identify the ρ of the change point alignments by inspecting the bin centered at $\theta = 0$. Two clear alignments are detected around $\rho = 65000$ and $\rho = 100000$, corresponding to the change alignments at $k = 65537$ and $k = 98305$. A third, weaker line is seen close to $\rho = 58000$, corresponding to the biased change point detected at scale $j = 6$ for the first transition.

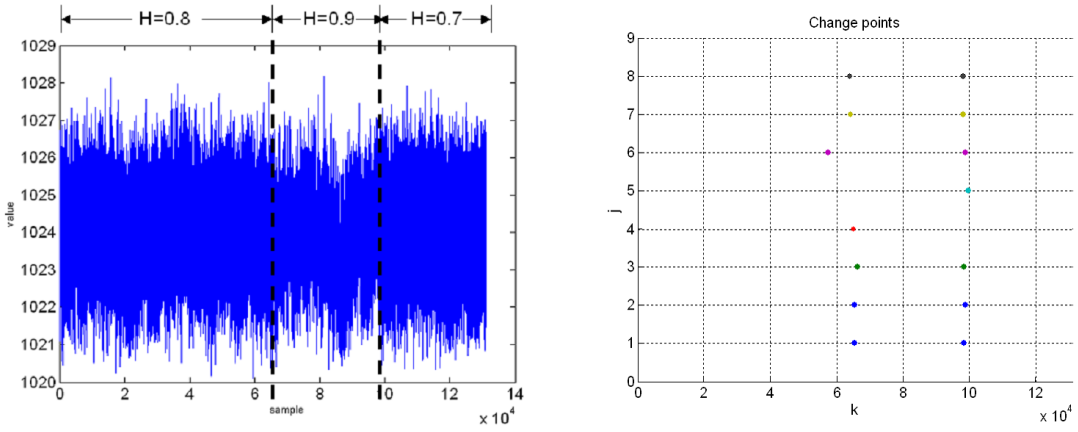

Figure 5.12: *Left:* FGN trace with 3 segments with $H = 0.8$, $H = 0.9$ and $H = 0.7$. The change points are at $k_1 = 65537$ and $k_2 = 98305$. *Right:* Detected variance change points at each scale.

Choosing $\Delta k = 1500$, $Q = 3$, $n_{MAX} = 100$, we obtained the following results:

$$\underline{\widehat{k_0}}^{\,*} = [57326, 64036, 64138, 65016, 65472, 65502, 66368, 98145, 98172, 98304, 98390, 98772, 98856, 99744]$$

$$H_{votes} = \begin{pmatrix} 57655 & 63724 & 65241 & 66759 & 98621 & 100138 \\ 1 & 2 & 3 & 1 & 6 & 1 \end{pmatrix}^{\mathsf{T}}$$

$$\underline{\widehat{k}}^{\,*} = [65330, 98440]$$

As we can see, the estimated global change points vector $\underline{\widehat{k}}^{\,*}$ is close to the theoretical one $\underline{k}^{\,*} = [65537, 98305]$.

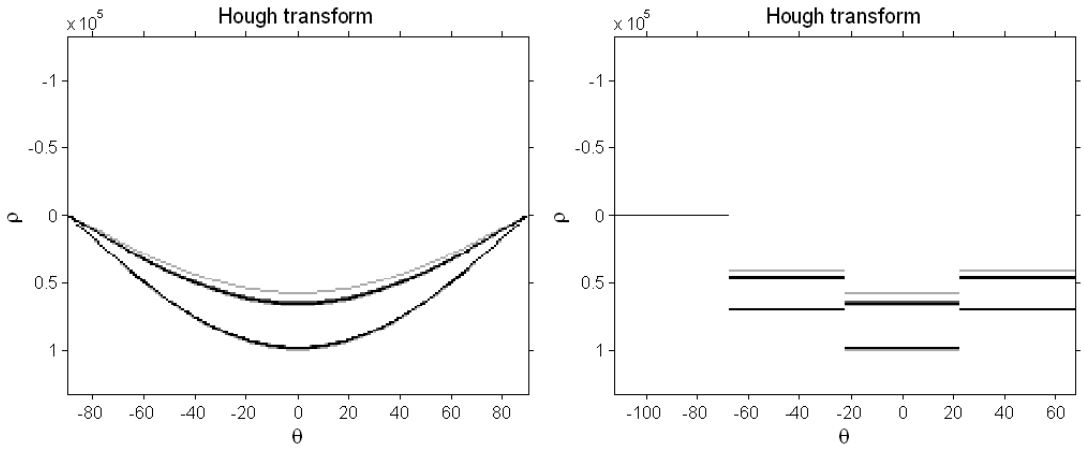

Figure 5.13: The Hough transform of the change point diagram of Figure 5.12. *Left*: θ resolution 1°. *Right*: θ resolution 60°.

5.7 Interpreting the results of the segmentation method

As has been described in the previous paragraphs, our method is able to locate the simultaneous change of variance across scales, which can correspond to either a variance change of the total process (i.e., in all scales) or to a Hurst parameter change. The output of the algorithm is the set of segments in which the original trace is divided. Inside each of the segments a variance and scaling parameter estimation can be done. Therefore, we can provide a piece-wise segmentation of $var(t)$ and $H(t)$.

Interpreting the results of our method is easy when one is working with synthetic traces. A shift in the traffic volume leads to a variance change at every scale and all changes are simultaneously positive (increase) or negative (decrease), while a change in the scaling parameter produces different signs in the variance change (the scales above the blind point will increase while the others will decrease, or vice versa) and possibly a blind point in the middle scales. If we assume a self-similar model, we will look for strict variance alignments; in contrast, an LRD model only needs alignments at higher scales. Real traffic traces are usually more complex to interpret as we can find simultaneous changes of variance and scaling parameter, or alignments in limited ranges of scales. Therefore, what we find are the boundaries of the segments with homogenous variance structure, which in turn coincide with the regions where the scaling parameters of traffic remain constant.

One could argue that since we assume an LRD model in our analysis, we lose generality due to our focus on monofractality rather than multiscaling or multifractality. However, recent studies have shown that a Poisson model for sub-second scales can be employed with core network traffic, while at higher scales piecewise-linear nonstationarity and long-range dependence seem to fit well with measurements of real networks [KMFB04]. Uhlig [Uhl04] has recently identified two different scaling regimes in network traffic: multifractality at small timescales, and LRD at large timescales. These findings, especially the former, reinforce the need for a method like ours, capable of detecting the changes in the low-frequency LRD components; besides, even if the trace under study is not

LRD, we can still look for less-rigorous alignments, like the typical effect produced by congestion or rate limitation of TCP traffic on WANs that increases the value of the variances over selected scales (usually at scales below round-trip time, where TCP rate adaption occurs). In this sense, our work can be useful when used at the higher timescales, meaning that the granularity of our method will be of seconds or above. This timescale of work is appropriate, since the computations involved in our method would not be possible to perform at a micro- or millisecond scale. Therefore, our methods will not be useful for adapting the behavior of TCP (such as the fractal-aware congestion control described in [HGHP02]), but it would be appropriate for resource control described in [EC05], or the effective bandwidth estimator described in [YTJ01]. Our method would trigger the update of these algorithms.

Finally, we want to emphasize that our approach is more general than the constant window-based methods, since we can localize the variance-transition points to any position and scale. In addition, we can monitor the second order nonstationarities of the time series (both for the whole process and scale-by-scale).

5.8 Analysis methodology

For each of the wavelet transforms considered, and for both variance change point detection methods (ICSS and SIC), we will analyze the following situations:

- A pre-computed FGN trace with Hurst parameter $H = 0.8$ (`fgn08.mat`) whose length is 131072 samples, with zero mean and unit variance. The trace has been generated with the Matlab FBM generator. No change point should be detected. The trace is shown in Figure 5.14 (left).
- The same FGN trace with an abrupt increase of the mean (from $\mu_1 = 0$ to $\mu_2 = 5$) for its second half part. No change point should be detected. The trace is shown in Figure 5.14 (right).
- A FGN trace of 262144 samples with $H = 0.8$ and a variance increase (from $\sigma_1^2 = 1$ to $\sigma_2^2 = 4$) in its second half part. The change point should be detected at all scales and be aligned at position $k = 65537$. The trace is shown in Figure 5.15 (left).
- A FGN trace of 262144 samples trace with a Hurst parameter change, the first and the second half of which are given by the `fgn05.mat` and the `fgn09.mat` pre-computed traces, respectively. A Hurst change point should be detected at position $k = 131073$. The trace is shown in Figure 5.15 (right).
- A similar trace, with a not so abrupt change in the Hurst parameter (from $H = 0.7$ to $H = 0.8$ is also analyzed.
- A trace with 3 segments of FGN ($H = 0.8, 0.9$ and 0.7) with changes at positions $t_1 = 65537$ and $t_2 = 98305$ and a total length of 131072 samples. The trace is shown in Figure 5.16 (left).

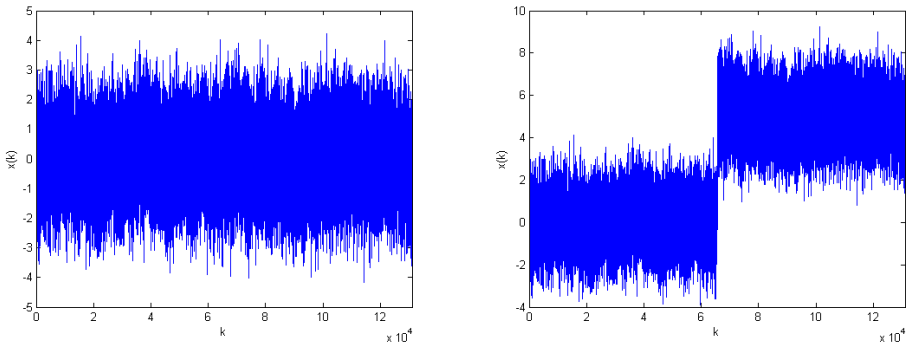

Figure 5.14: *Left*: FGN trace with $H = 0.8$. *Right*: the same trace with a mean increase of value 5 in its second half part.

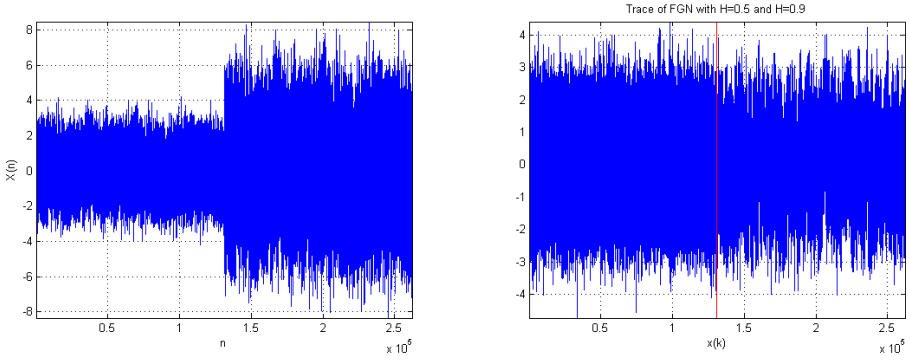

Figure 5.15: *Left*: FGN trace with $H = 0.8$ with $\sigma_1^2 = 1$ and $\sigma_2^2 = 4$. *Right*: FGN trace with two segments with $H = 0.5$ and $H = 0.9$ ($\sigma^2 = 1$).

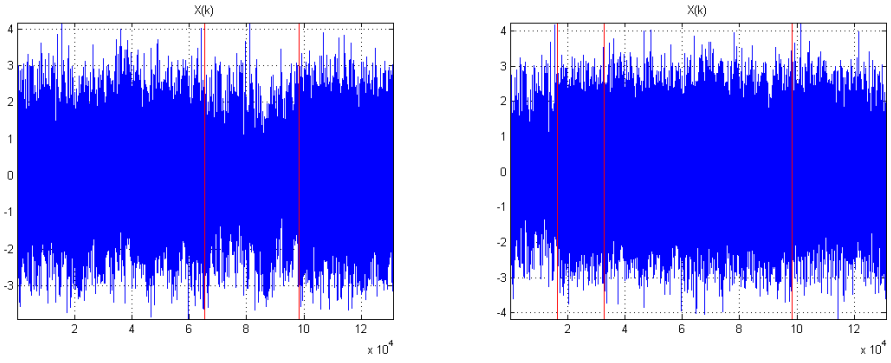

Figure 5.16: *Left*: FGN trace with three segments with different Hurst parameter ($H = 0.8, 0.9$ and 0.7). The change points are located at $t_1 = 65537$ and $t_2 = 98305$. *Right*: FGN trace with four segments with different Hurst parameter ($H = 0.9, 0.5, 0.7$ and 0.8). The change points are located at $t_1 = 16385$, $t_2 = 32769$, $t_3 = 98305$. The variance is constant and unitary ($\sigma^2 = 1$).

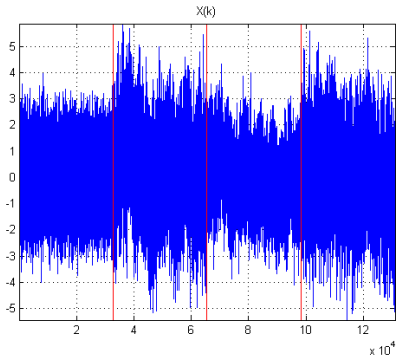

Figure 5.17: FGN trace with four segments with different variance and Hurst parameter: $H = 0.5, \sigma^2 = 1$, $H = 0.9, \sigma^2 = 2$, $H = 0.9, \sigma^2 = 1$, $H = 0.5, \sigma^2 = 2$. The change points are located at $t_1 = 32769$, $t_2 = 65537$ and $t_3 = 98305$.

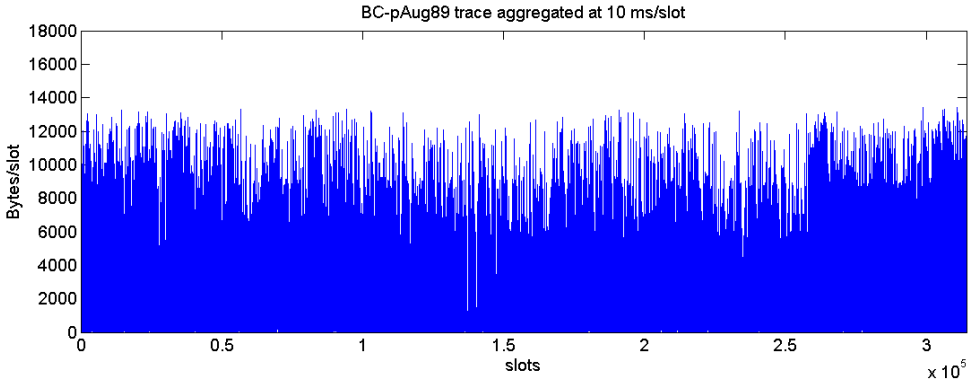

Figure 5.18: The Bellcore `BC-pAug89` trace aggregated at the 10 ms scale.

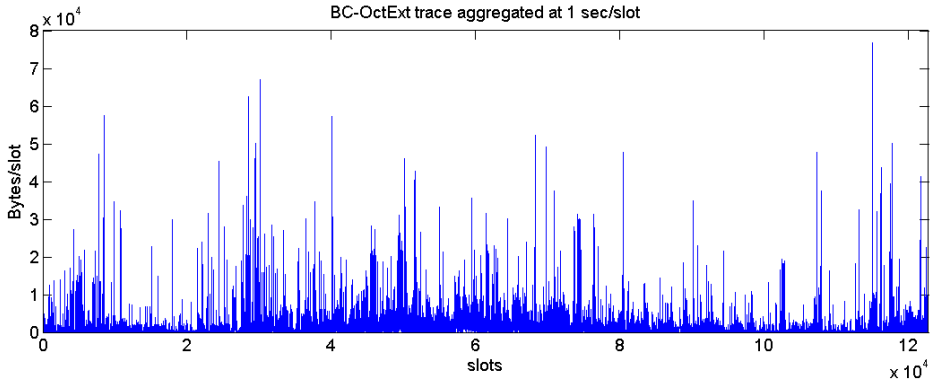

Figure 5.19: The Bellcore `BC-OctExt` trace aggregated at the 1 s scale.

- A more elaborate example considers a 131072 samples trace composed of four segments with $H = 0.9$ ($N_1 = 16384$ samples of the `fgn09.mat` trace), $H = 0.5$ ($N_2 = 16384$ samples of the `fgn05.mat` trace), $H = 0.7$ ($N_3 = 65536$ samples of the `fgn07.mat` trace) and $H = 0.8$ ($N_4 = 32768$ samples of the `fgn08.mat` trace). The mean and variance stay constant (0 and 1, respectively). The trace is shown in Figure 5.16 (right).

- A 131072 samples trace composed of four segments of 32768 samples each, with $H = 0.5, \sigma^2 = 1$, $H = 0.9, \sigma^2 = 2$, $H = 0.9, \sigma^2 = 1$, $H = 0.5, \sigma^2 = 2$. The trace, shown in Figure 5.17, includes two simultaneous changes of variance and Hurst parameter at $k = 32769$ (H increase, variance increase) and $k = 98305$ (H decrease, variance increase).

- In order to empirically assess the power of the procedure, we generated 1000 independent realizations of 262144 samples-long traces with $H = 0.5$ and $H = 0.9$ in the first and second half of the trace, respectively. The quantity and quality (in terms of its closeness to the true change point, histograms and dispersion of the detected changes) are analyzed.

- Finally, the Bellcore `BC-pAug89` and `BC-OctExt` trace aggregated at the 10 ms a 1 s scales, respectively, are analyzed as examples of real traffic. More details of these traces can be found in Section 3.2.10 and 5.4. The traces are shown in Figures 5.18 and 5.19.

Not all the situations will be analyzed for each algorithm; for the cases where no new results are obtained, its description will be skipped.

5.9 Summary of the chapter

This chapter has addressed the need for algorithms that can track the scaling parameters of traffic, and has presented a comprehensive list of works done by other authors regarding the topic. We propose an algorithm built from three pieces: a wavelet transform, a variance change detection method applied to each of the wavelet subbands, and a clustering and alignment detection method. We have discussed several issues regarding the practical implementation of the method, and have proposed the Hough transform as the clustering procedure. Finally, the chapter has described the analysis methodology that will be used in the following chapters. Chapters 6, 7 and 8 will deal with the DWT, MODWT and DTWT-based methods, respectively.

Chapter 6

DWT-based segmentation methods

6.1 Introduction

This chapter presents the issues related to the joint wavelet-variance change detection methods based on the Discrete Wavelet Transform, namely DWT-ICSS and DWT-SIC. The performance of the algorithms will be studied for the cases of synthetic LRD and real traffic traces. We start by presenting some issues related to the DWT-based method such as the scale alignment and the lack of resolution at the higher scales. A characterization of the statistical distribution of the DWT coefficients is provided, due to its influence in the performance of the segmentation methods. A pure FGN trace is analyzed, in order to evaluate the insensitivity of our algorithms to deviations from the ideal case. We later test the (desired) *blindness* of our methods to changes in the series' mean. Then an analysis of the detection capabilities of variance and Hurst parameter changes (either single or multiple) is performed, together with an empirical assessment of the test power. The case of a simultaneous variance and Hurst parameter change is also studied. The methods are then applied to real traffic traces from the Bellcore data set. Finally, a progressive, near-real-time version of the algorithms is described.

Some of the results of the DWT-based methods have been published in the following references: [RS04a, RS04b, RS05b, RSdA05, RS05c], while the progressive version of the algorithm was described in [RMCS04, RS05a, Min05].

6.2 DWT-based algorithms

The main challenge of the DWT-based algorithms comes from the different quantity of coefficients available at each scale, since each branch of the DWT filter bank suffers a different number of decimations, as discussed in Section 2.4.6. For example, if the original trace $X(t)$ has 1024 samples and we perform the decomposition with the `db1` wavelet family, the sequences $d_X(j,k)$ will have

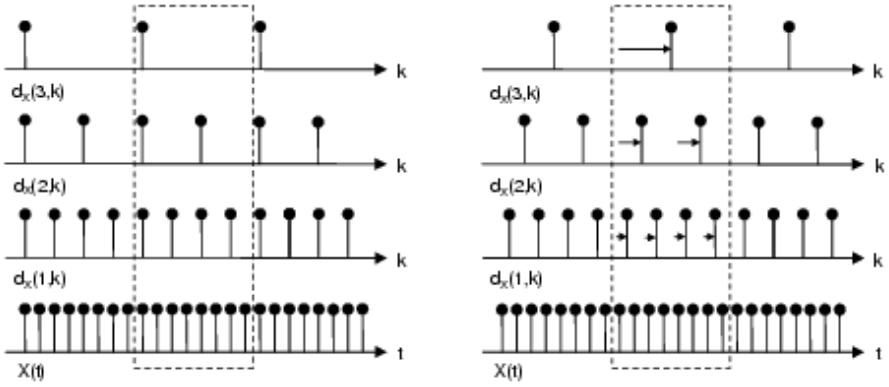

Figure 6.1: *Left*: time alignment between the DWT coefficients at scales $j = 1, 2, 3$, using *db1* wavelet. *Right*: the same DWT coefficients, after the phase correction.

$n_1 = 512$ samples for $j = 1$, $n_2 = 256$ samples for $j = 2$, $n_3 = 128$ samples for $j = 3$ and so on. This causes the sequences $d_X(j,k),\ j = 1, \ldots, J$ to be *misaligned*, as shown in Figure 6.1 (left).

Therefore, the question is: how can we align the position of the wavelet coefficients with their *zone of influence*? We have to find the time alignment between the original trace and the DWT coefficients at different scales. For example, again from Figure 6.1 (left), we see that the segment $X(9), \ldots, X(16)$ of the original trace is related to:

- the coefficients $d_X(1,5), \ldots, d_X(1,8)$ at scale $j = 1$;
- the coefficients $d_X(2,3), d_X(2,4)$ at scale $j = 2$;
- the coefficient $d_X(3,2)$ at scale $j = 3$.

In order to provide a good estimation of the change points vector $\underline{k}$ at scale j, a scale-dependent *alignment* is needed, as shown in Figure 6.1 (right). If we use the `db1` wavelet family, this correction will be

$$\widehat{\underline{k}}\,'(j) = \widehat{\underline{k}}(j) \cdot 2^j \tag{6.1}$$

while for the other wavelet mothers the expression depends on the scale and on the delay introduced by each filter bank (see Section 2.4.5).

The other challenge of the DWT-based methods, as we will see in the following paragraphs, is the smoothing effect suffered by the variance changes at the higher scales. Recall that the aforementioned scales correspond to the lower frequencies of the signal, which makes the changes appear less abrupt than at the lower scales. This will redound in a decrease in the quantity of detected change points at those scales.

In the following sections we present some examples that highlight the properties and performance of the DWT-ICSS and DWT-SIC algorithms, following the methodology described in Section 5.8.

6.3 Statistical distribution of the DWT coefficients

Before applying the variance change detection methods, one should check the distribution of the wavelet coefficients. The DWT transforms are, basically, the output of a difference operator. As was described in Chapter 2, the mother wavelet is an oscillating function with positive and negative regions, and its associated filters reproduce this behavior. The best example of this property is the Haar (`db1`) wavelet, which is exactly a one-tap difference operator. The higher order Daubechies wavelets and the other wavelet families exhibit a *smoother* differentiating behavior, but essentially the effects are the same: the wavelet coefficients tend to follow a Laplacian distribution centered at zero, as is expected from a differentiating operator.

The Laplacian-shaped distribution at the output of the DWT is inadequate if we want to apply the variance change detection methods to the DWT coefficients; recall that the ICSS and SIC statistics expect a Gaussian input. Fortunately, the Laplacian distribution, though clearly differentiated from the Gaussian, still keeps some properties in common with the latter, such as symmetry around the mean and its light-tailed nature. Actually, both the Gaussian and Laplacian as well as the Dirac delta and uniform distributions are particular cases of the so-called *Generalized Gaussian* distribution [DMGFRD01]. Such a distribution has been found to appropriately describe the distribution of the wavelet coefficients of images [MV02] and, more important to our interests, the wavelet coefficients of real network traffic traces [Glo05]. The probability density function of the GGD is given by

$$f(x) = \frac{\lambda p}{2\Gamma(\frac{1}{p})} e^{-\lambda |x-m|^p} \tag{6.2}$$

where $0 \leq p < +\infty$ is the shape parameter, λ relates to the variance of the distribution, m is the mean, and

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt, \quad z > 0 \tag{6.3}$$

The shape parameter p plays an important role, since it modulates the appearance of the distribution. Figure 6.2 shows the probability density function[1] for some values of p, with the following particular cases:

- $p = 0$: Dirac's delta function.
- $p = 1$: Laplacian distribution.
- $p = 2$: Gaussian function.
- $p = +\infty$: Uniform distribution.

We are aware of and assume the limitation implied by using ICSS and SIC for non-gaussian input, but to our knowledge few works have considered the case of variance change detection in non-gaussian time series. Among them, the most important for our interests was carried out by Chen and Gupta [CG00], who studied the asymptotic distribution of the SIC statistic under restricted[2]

[1] The Matlab code for the generation and estimation of the Generalized Gaussian Distribution was developed by Minh N. Do [Do].

[2] They assume that the first four moments of the input random variable behave as those of the Gaussian distribution. These conditions seem to be rather *technical* and not have an important influence in the final result.

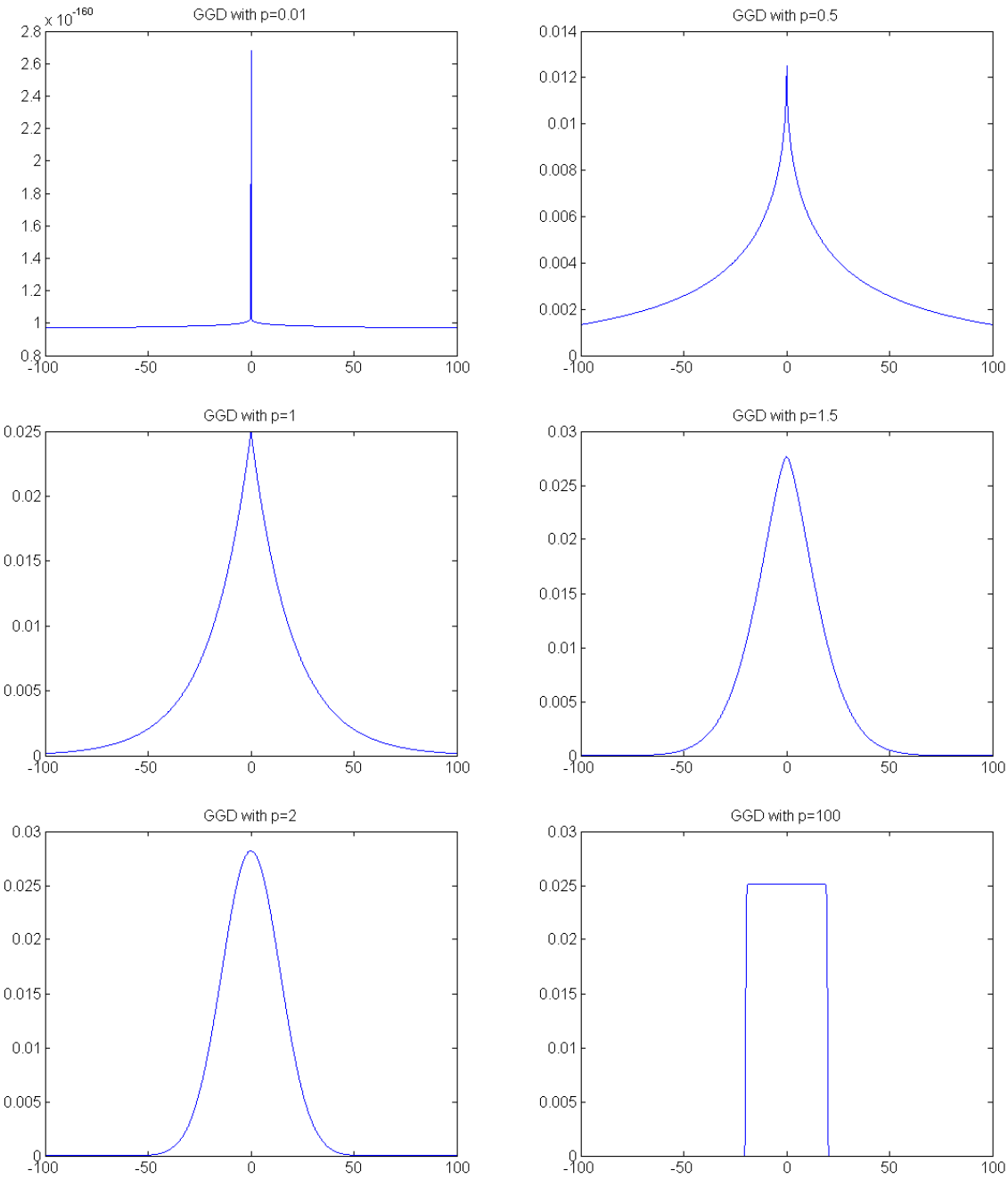

Figure 6.2: Probability density function of the Generalized Gaussian distribution with $m = 0$, $\lambda = \frac{1}{20}$ and different values of the shape parameter p. From top to bottom, left to right, $p = 0.01$ (almost Dirac's delta), $p = 0.5$, $p = 1$ (Laplacian), $p = 1.5$, $p = 2$ (Gaussian), $p = 100$ (almost uniform).

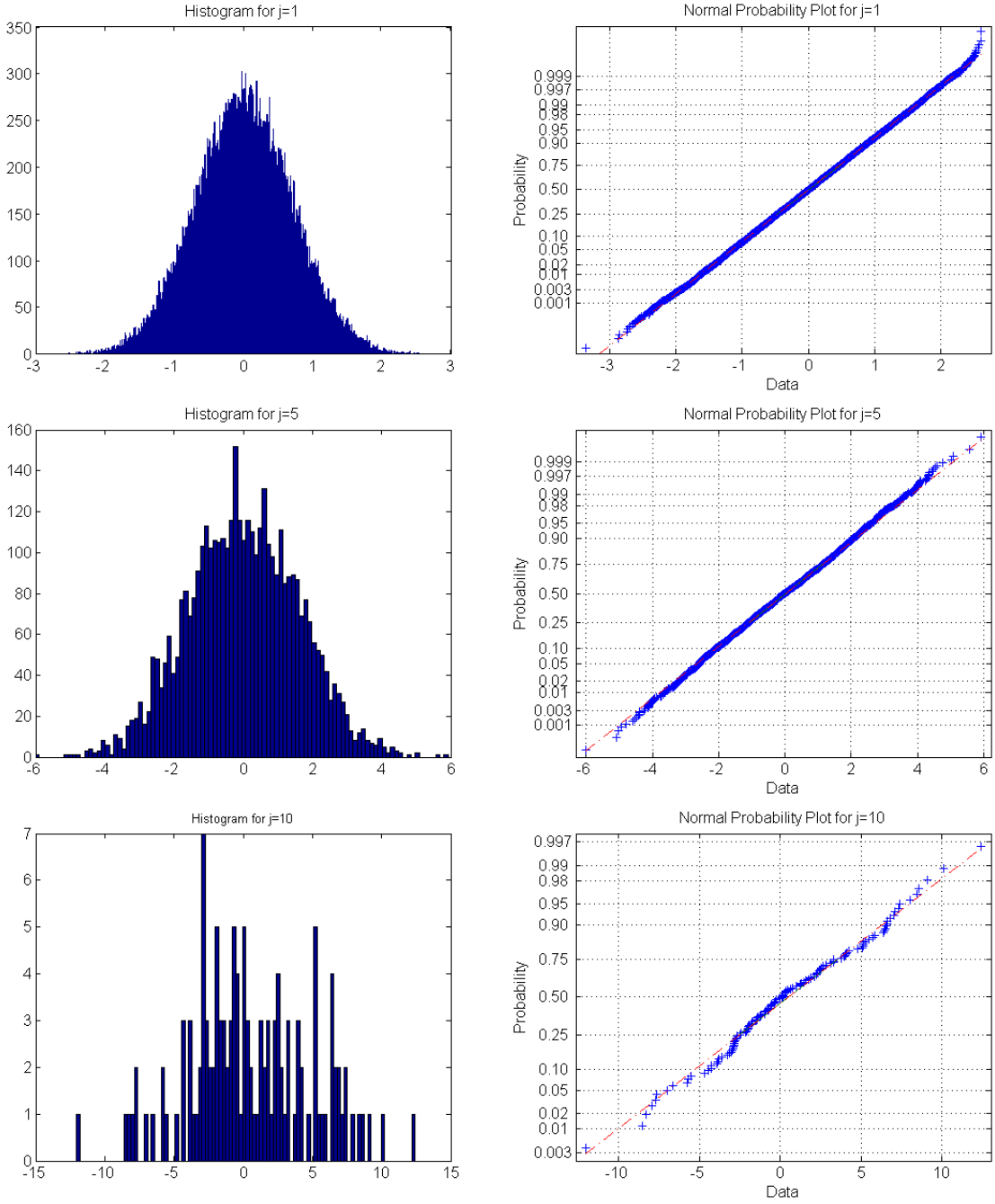

Figure 6.3: Histograms (*left*) and probability plots for graphical normality testing (*right*) of the DWT coefficients of the FGN trace with $H = 0.8$ when analyzed with `db1`. From top to bottom, the analysis for the coefficients at scales $j = 1$, $j = 5$ and $j = 10$.

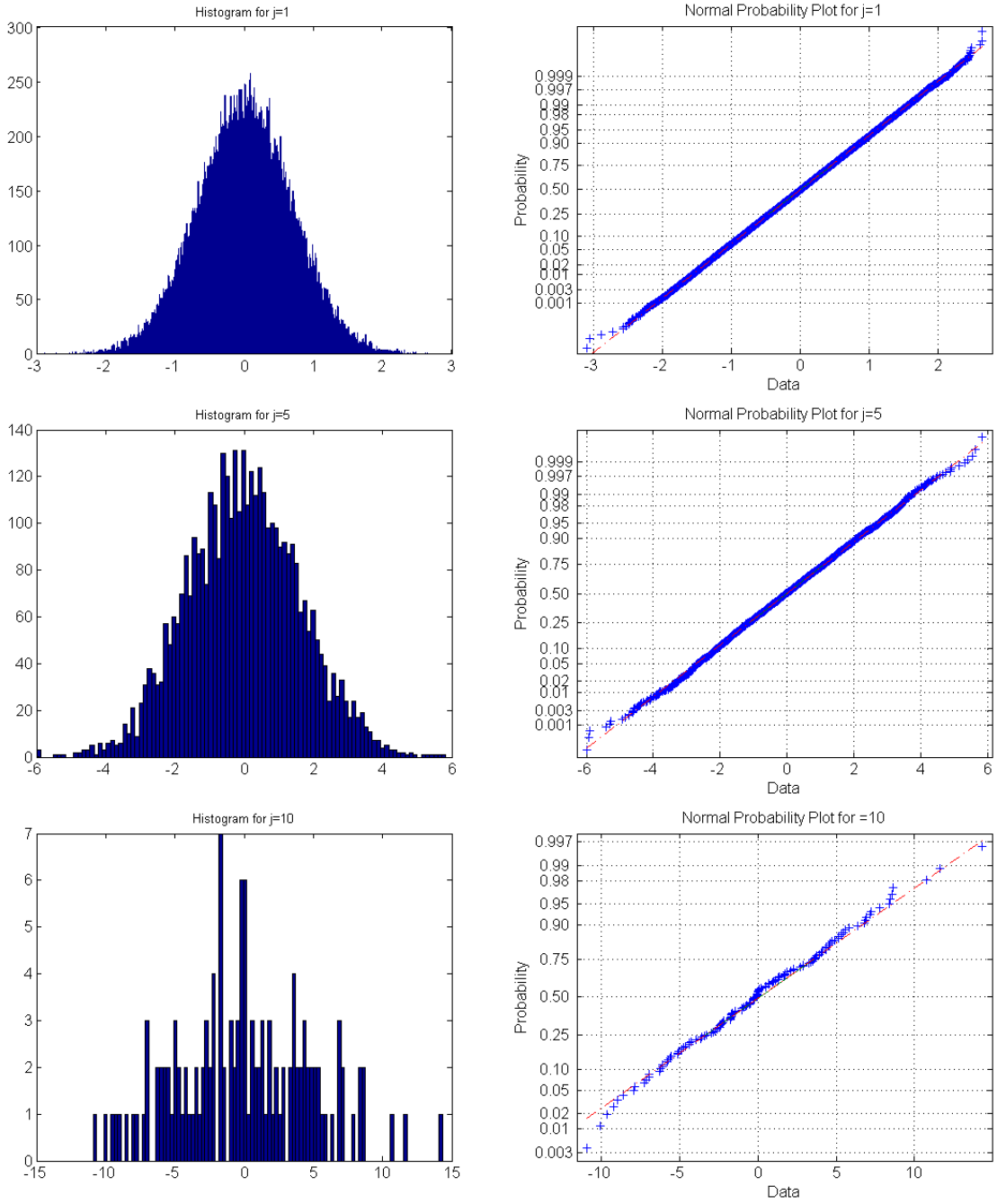

Figure 6.4: Histograms (*left*) and probability plots for graphical normality testing (*right*) of the DWT coefficients of the FGN trace with $H = 0.8$ when analyzed with `db3`. From top to bottom, the analysis for the coefficients at scales $j = 1$, $j = 5$ and $j = 10$.

non-normality and found it to follow the same distribution as in the Gaussian input case. No equivalent study has been found for the ICSS, but since both methods rely on the Cumulative Sum of Squares (CSS) statistic, we hypothesize that the behavior of the ICSS under non-normality will not deviate too much from the results obtained in the Gaussian case.

We comment now on the distribution of the wavelet coefficients of the synthetic traces that we have used in out experiments. Figures 6.3 and 6.4 show the histograms and the probability plots for graphical normality testing of the FGN trace with $H = 0.8$ when analyzed with `db1` and `db3`, respectively. In both cases, the Gaussian nature of the original trace keeps the distribution even at the higher scales, for which the probability plots is a more precise instrument than the histogram. The estimated shape parameter of the GGD is shown in Table 6.1. The first scale seems to be almost perfectly Gaussian, while the higher scales present a shape parameter slightly higher than 2. No important differences have been found regarding the wavelet order; the use of `db1` or `db3` does not influence the distribution of the wavelet coefficients. Similar results were found for other values of H in the range $0.5 \leq H < 1$.

Scales	$j = 1$	$j = 5$	$j = 10$
`db1`	1.97	2.16	2.12
`db3`	1.99	1.99	2.16

Table 6.1: GGD shape parameter of the DWT decomposition of the FGN trace with $H = 0.8$.

6.4 Analysis of a pure FGN trace

We begin with the analysis of FGN traces of length 131072 samples, whose Hurst parameter ($H = 0.8$), variance (unit) and mean (zero) are constant. We only describe the results obtained with the $H = 0.8$ case, which we consider representative of the typical H values found in network traffic. The results are similar for other values of H in the range $0.5 - 0.9$. We are aware of the statistical unreliability of the approach and do not pretend the results to be generalizable, since we rely on just a realization; we rather provide them as an example of how the algorithms work at each of the steps. Later we will provide more statistically-rooted results by performing the same analysis over 1000 realizations of a similar trace.

The analysis is performed at the $J = 15$ level[3]. The method should not return any change point. However, the residual correlation introduced by the wavelet transform when long wavelet filters are used may fool the method. The DWT detail coefficients (using the `db1` wavelet family) are plotted in Figure 6.5.

[3]Scales 16 and 17, though available ($2^{17} = 131072$), have so few coefficients and are so unreliable that, following the advice from Abry and Veitch [VA], we skip them.

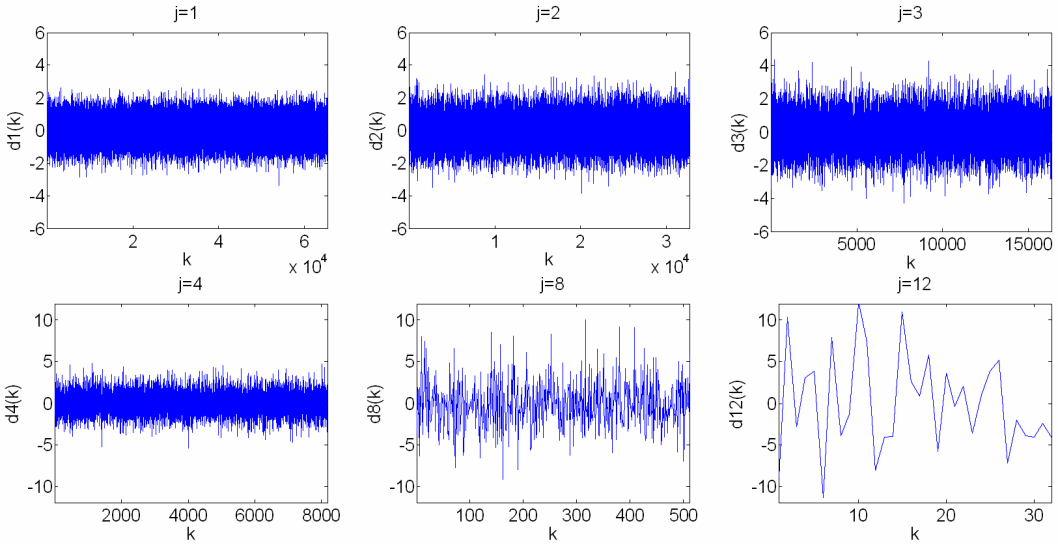

Figure 6.5: DWT decomposition of the FGN trace, for $j = 1, 2, 3, 4, 8$ and 12, using `db1` wavelet. Notice the difference in the units of the y axis.

6.4.1 DWT-ICSS

Figure 6.6 shows the ICSS statistic computed at the 95% significance level at scales $j = 1, \ldots 4, 8$ and 12, together with their associated normalized critical values. Notice the difference in the lengths of the coefficient series, motivated by the downsampling performed by the DWT. The figures depict the different series aligned in time; this means that the shorter series (the higher scales) have been *expanded* in their x axis in order to coincide with the longer ones. Since the trace possesses homogeneous variance, we observe that:

- the plots of the details $d_X(j,k) = d_j(k)$ do not exhibit a change in variance for any j;
- the DWT-ICSS statistic $D(j,k) = D_j(k)$ oscillates around zero for all j.
- no value of the DWT-ICSS statistic exceeds the critical value D^*.

We performed the DWT-ICSS analysis at the 95% and 99% significance level, for a set of Daubechies mother wavelets with different number of vanishing moments. The results obtained with the DWT-ICSS at the 95% significance level are:

- `db1`: no change points.
- `db2`: no change points
- `db3`: three change points at $(j = 2, k = 87276, k = 99006)$, and $(j = 8, k = 62995)$.
- `db4`: four change points at $(j = 2, k = 87442, k = 99772)$, $(j = 7, k = 38685)$, and $(j = 12, k = 117274)$.
- `db5`: five change points at $(j = 9, k = 91849)$, $(j = 11, k = 118328)$, $(j = 12, k = 114688)$, $(j = 13, k = 98304)$, and $(j = 14, k = 90112)$.

while DWT-ICSS at the 99% significance level returned:

- `db1`: no change points.

- `db2`: no change points
- `db3`: one change point at $(j = 2, k = 87276)$.
- `db4`: two change points at $(j = 2, k = 87442)$ and $(j = 12, k = 117274)$.
- `db5`: three change points at $(j = 9, k = 91849)$, $(j = 12, k = 114688)$, and $(j = 13, k = 98304)$.

As we expected, for the lower order wavelets no changes are detected, while the higher order analysis return isolated, spurious change points that are not significant (no relevant alignment across scales is detected in any case). Performing the analysis above the `db5` wavelet made no sense, due to to the high amount of spurious points. Besides, the higher the significance level, the less spurious points survive.

6.4.2 DWT-SIC

For the Schwarz Information Criterion case, Figure 6.7 shows the $SIC(k)$ statistic, together with a horizontal line at $SIC(n) - c_\alpha$, where c_α is the critical level computed at the 95% significance level. Recall that the minimum information criterion states that the null hypothesis H_0 (no change is present) is accepted when $SIC(n) \leq \min_{1<k<n} SIC(k) + c_\alpha$. Therefore, when the horizontal line is under the minimum value of the $SIC(k)$, the algorithm decides that no change is present, as is the case of all the scales plotted in Figure 6.7.

For the DWT-SIC case at the 95% significance level, we get the following results:

- `db1`: one change point at scale $(j = 15, k = 65536)$.
- `db2`: one change point at $(j = 6, k = 130880)$
- `db3`: one change points at $(j = 2, k = 87376)$
- `db4`: four change points at $(j = 7, k = 254)$ and $(j = 12, k = 117274, k = 120724, k = 127622)$.
- `db5`: 16 change points, all at scales over $j = 9$, concentrated at the beginning and the end of the sequence.

while the 99% significance level analysis returns:

- `db1`: one change point at $(j = 15, k = 65536)$.
- `db2`: no change points.
- `db3`: no change points.
- `db4`: three change points at $(j = 12, k = 117274, k = 120724, k = 127622)$
- `db5`: three change points at $(j = 12, k = 114688, k = 121241, k = 124518)$

As expected, the higher the significance level, the less spurious points are detected. Compared to the results obtained for the ICSS, the higher performance of SIC working at 99% in front of the residual correlation is evident, as was expected from the tests performed in Chapter 4. However, the 95% significance level gives very bad results for the higher order wavelets.

Summarizing, we can conclude that apart from some, non-aligned spurious change points, the analysis of homogenous FGN traces with the DWT-based methods give good results, as long as the wavelet order is on the order of 1-4. No alignment of two points at different scales was found; therefore, a quorum of 3 aligned change points seems enough for our clustering procedure.

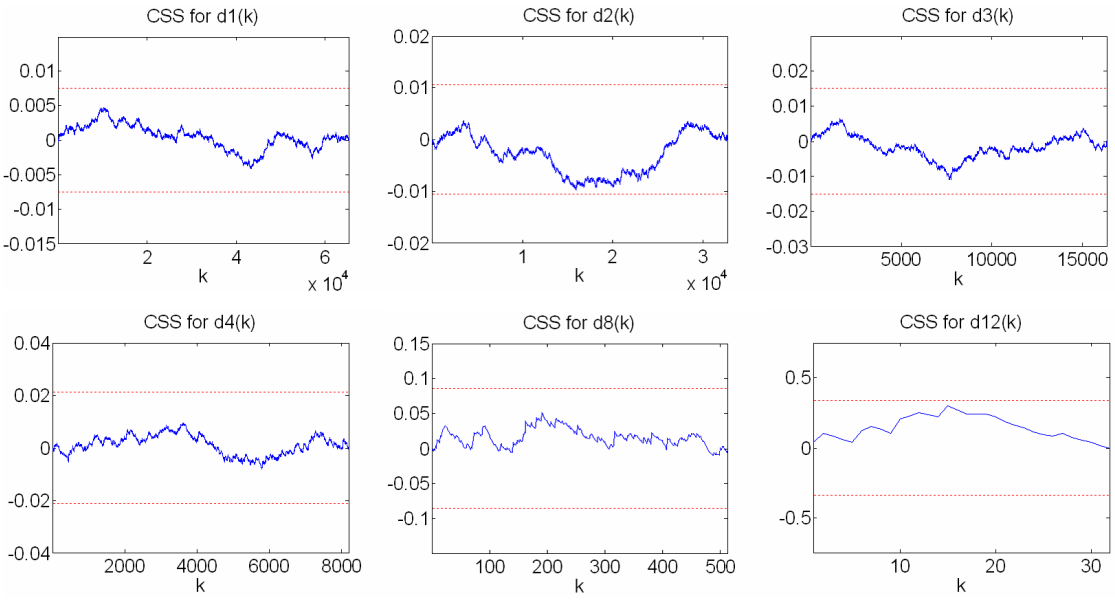

Figure 6.6: CSS statistic applied to the wavelet details of the FGN trace, for $j = 1, 2, 3, 4, 8$ and 12, using `db1` wavelet at the 95% significance level. The associated critical levels $\pm \frac{1.358}{\sqrt{T/2}}$ are depicted as red horizontal lines. Notice the difference in the units of the y axis.

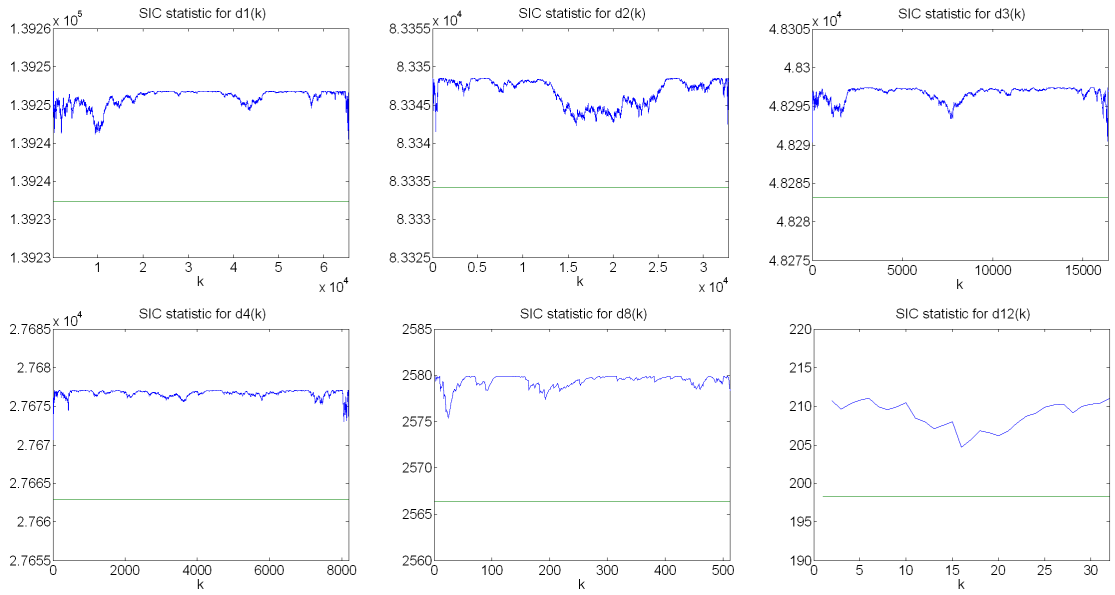

Figure 6.7: SIC(k) statistic applied to the wavelet details of the FGN trace, for $j = 1, 2, 3, 4, 8$ and 12, using `db1` wavelet at the 95% significance level. The associated critical level $SIC(n) - c_\alpha$ is plotted as a horizontal line. Notice the difference in the units of the y axis.

6.5 Analysis of the effects of a mean change

We now consider a trace of 262144 samples (plotted in Figure 5.14, right) composed as follows: the first half includes the `fgn08.mat` synthetic trace discussed in Section 3.2.9 (with mean $\mu_1 = 0$), whereas the second half was obtained by adding to every sample of the same trace a constant value $\mu_2 = 5$. Our method should ideally be blind to changes in the series' mean, but the impurities of the actual trace (which is an approximation of an ideal FGN) and the non-ideality if the ICSS and SIC algorithms may make the methods detect false changes. The DWT analysis is performed at $J = 15$ scales, for the 99% significance level, using `db1-db5`.

6.5.1 DWT-ICSS

For the DWT-ICSS, the results are:

- `db1`: no change points.
- `db2`: four change points at $(j = 9, k = 65027, 66044)$ and $(j = 11, k = 63550, 73479)$.
- `db3`: five change points at $(j = 2, k = 87276)$, $(j = 8, k = 63757, 68330)$ and $(j = 11, k = 63608, 69391)$.
- `db4`: five change points at $(j = 2, k = 87442)$, $(j = 11, k = 63663, 69280)$ and $(j = 12, k = 62086, 68985)$.
- `db5`: five change points at $(j = 10, k = 64572, 68427)$, $(j = 11, k = 63715, 67356)$ and $(j = 12, k = 58982)$.

It seems that the higher scales of the wavelets above `db1` are sensitive to the mean change, though the alignment, if any, is limited to a couple of scales. When the clustering and alignment method with a quorum of 3 or more aligned points is applied to the trace, no alignment is detected. Therefore the quorum used in our following experiments should be above 3.

6.5.2 DWT-SIC

For the DWT-SIC, the results are:

- `db1`: one change point at $(j = 9, k = 65536)$.
- `db2`: no change points.
- `db3`: three change points at $(j = 11, k = 63608, 67463, 69391)$.
- `db4`: no change points.
- `db5`: no change points.

Though some spurious points appear near the mean change, the SIC-based method is far less sensitive to level shifts. The results are similar to or better than those obtained with the original, mean-constant FGN trace. As was the case with DWT-ICSS, no change alignments are detected, though the SIC method comes closer to the ideal result (no changes detected).

6.6 Analysis of the effects of a variance change

Contrary to the previous sections, we introduce here variance changes that our method *must* detect. We begin studying a trace with a single variance change in the middle. The first half of our zero mean trace (plotted in Figure 5.15, left) has a Hurst parameter $H = 0.8$ and variance $\sigma_1^2 = 1$. The second half has the same H, but different variance $\sigma_2^2 = 4$. The DWT has been performed at the deepest level ($J = 17$), using the Haar (`db1`) wavelet, though we have used only the $j = 1 \dots 15$ scales. The detail coefficients for some of the scales can be seen in Figure 6.8, together with their autocorrelation functions in Figure 6.9, where it can be seen that the series are almost uncorrelated, thanks to the decorrelating properties of the DWT (see Section 3.2), but still there is some residual correlation.

6.6.1 DWT-ICSS

As can be seen in Figure 6.8, the DWT detail coefficients exhibit an evident change of variance around the middle of the series (corresponding to point $t_1 = 131072$ in the original trace), though as the scale increases, the change point is not so clear. This in turn explains why this change point is accurately located as the absolute maximum of the ICSS statistic for the lower scales, but not as clearly for the higher scales, as shown in Figure 6.10. Notice the bias phenomenon in the higher scales ($j = 11$ and $j = 13$) that makes the minimum of the CSS be deviated towards the right side of the true position of the changepoint. For $j = 11$ there are actually two local minima, one around position $k = 64$ (the true change point) and the other, which happens to be the global minimum, around position $k = 80$. For the $j = 13$ case, the true change (now to be found at $k = 16$) has disappeared, while the global minima is clearly located at $k = 21$.

Figure 6.11 shows the position of the detected change points by the ICSS at each scale, for the two significance levels (95% and 99%) using `db1` wavelet. The 95% case returns some false points, such as the two extra changes at the 7^{th} scale and the displaced points at the 11^{th} and 13^{th} scale. This bias comes from the behavior of the CSS statistic commented on the previous paragraph. These kinds of situations will be handled via the automatic alignment-detection-and clustering procedure implemented with the Hough Transform described in Section 5.6.

When the analysis is performed at the 99% significance level (Figure 6.11, right), the results improve slightly. Ideally, the change should be located at the middle of the series at all scales. The change point is clearly located (with some fluctuations in the position) at the lower scales (up to $j = 12$), with the exception of $j = 11$, where the point is located at $k = 163840$. The other case of bias of the CSS that we found at scale $j = 13$ does not appear in Figure 6.11 because for scales above $j = 12$ the CSS statistic is under the critical level D^*, and therefore no change point is considered valid. This is the consequence of the *smoothing* effect of the higher scales of the DWT (a signal filtered by a cascade of low-pass filters, followed by a high-pass filter). In order to correct this, we would need a wavelet transform with higher time resolution at the higher scales (lower frequencies); this approach will be explored in Chapter 7.

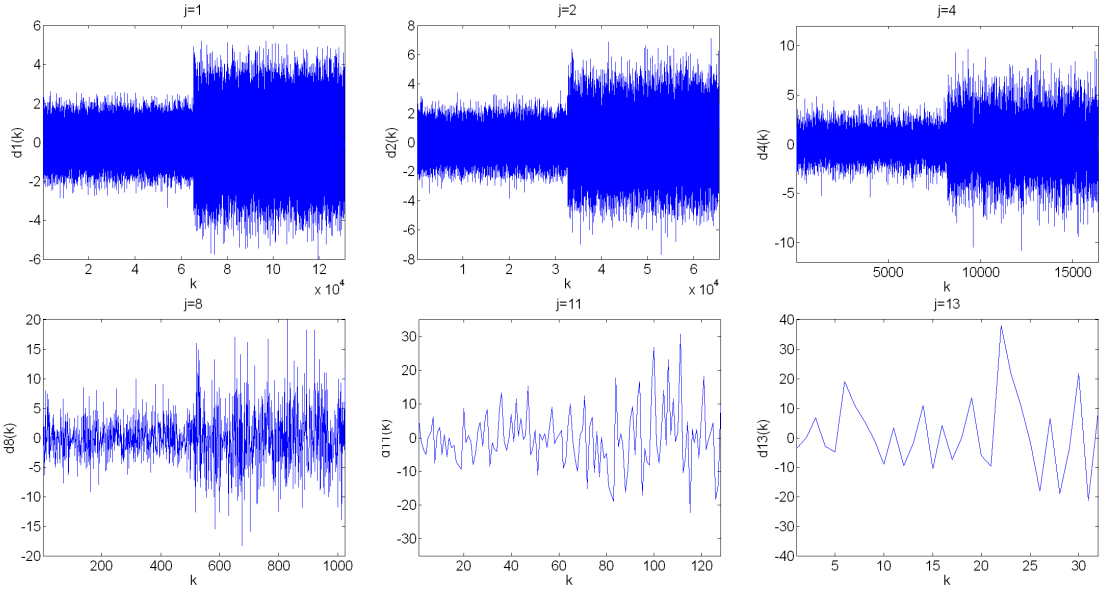

Figure 6.8: DWT detail coefficients of the trace with a variance change, for $j = 1, 2, 4, 8, 11, 13$, using `db1` wavelet.

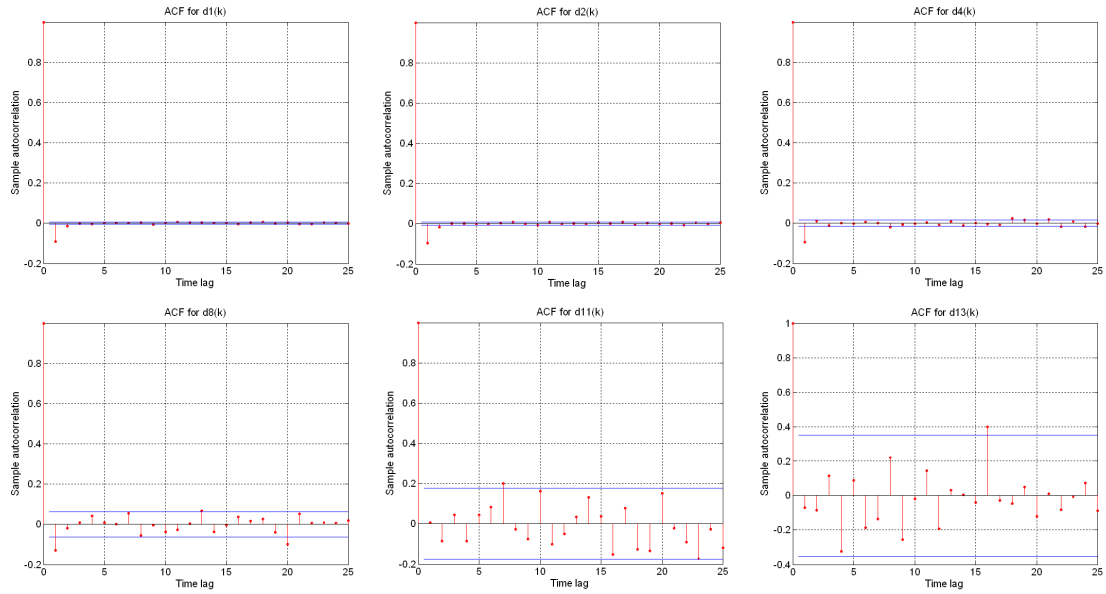

Figure 6.9: Sample autocorrelation of the DWT detail coefficients, using `db1` wavelet, of a FGN trace with a variance change, for $j = 1, 2, 4, 8, 11, 13$, with the 95% confidence intervals (horizontal lines).

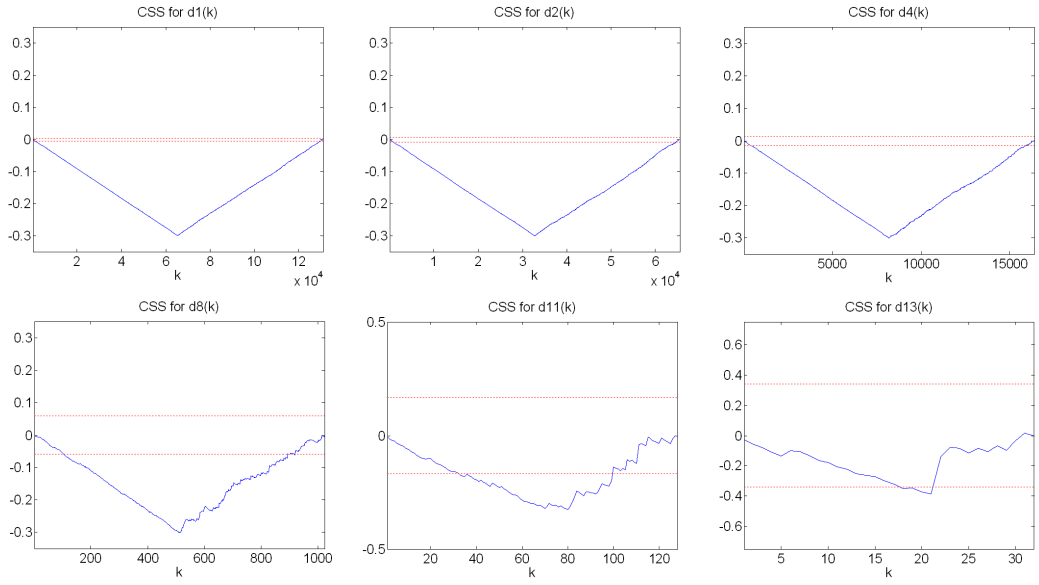

Figure 6.10: CSS statistic applied to the wavelet detail coefficients of a FGN trace with a variance change, for $j = 1, 2, 4, 8, 11, 13$, using `db1` wavelet. The 95% critical values $\pm\frac{1.358}{\sqrt{T/2}}$ are the horizontal red lines.

Regarding the influence of the wavelet order, Figure 6.12 shows the detected change points at the 99% significance level, when wavelets with a higher number of vanishing moments are used (`db2` and `db3`). No relevant differences are detected when compared with the `db1` case, apart from a certain increase in the bias of the higher scales, emphasized by the greater smoothing effect introduced by longer wavelet filters. The nonexistence of polynomial trends in the synthetically generated FGN traces explains the (almost) invariance of the results.

Finally, the clustering/alignment procedure is applied to the change point diagrams aforementioned. Testing the algorithm with different values for resolution and quorum we get the results shown in Table 6.2. The results show the (relative) insensitivity of the method with the wavelet order and the significance level, though the change points diagram vary when these parameters change (as shown in Figures 6.11 and 6.12). This insensitivity is the result of the non-linear thresholding effect introduced by the alignment detection method through the voting procedure, since it eliminates almost all the spurious points, while keeping the true changes. The resolution parameter has a bigger influence; the higher the resolution, the higher the variance of the alignment (see the errors of 24 samples obtained with the resolution of 200 samples). However, if we plot the relative error (in percentage, relative to the trace length) of the alignment localization, as shown in Figure 6.13 for values between 5 and 3000, the conclusion is that the method is quite insensible to resolution changes, at least for the variance change case; this effect can be explained by the special nature of the trace, where few points can be found in distant positions from the true change position. Finally, the quorum parameter may limit the detection capabilities of the method when a high resolution is used; see the cases when no alignment is detected because the dispersion of the five closest points are bigger than the required resolution. From the data we conclude that quorums of 3-4 points, and

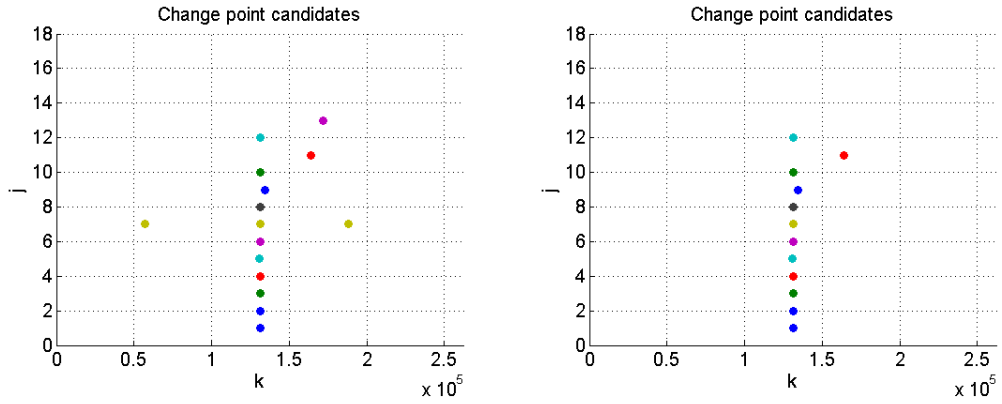

Figure 6.11: detected change points when the DWT-ICSS algorithm is applied to the trace with a variance change, using `db1` wavelet, for the 95% (left) and 99% (right) significance level.

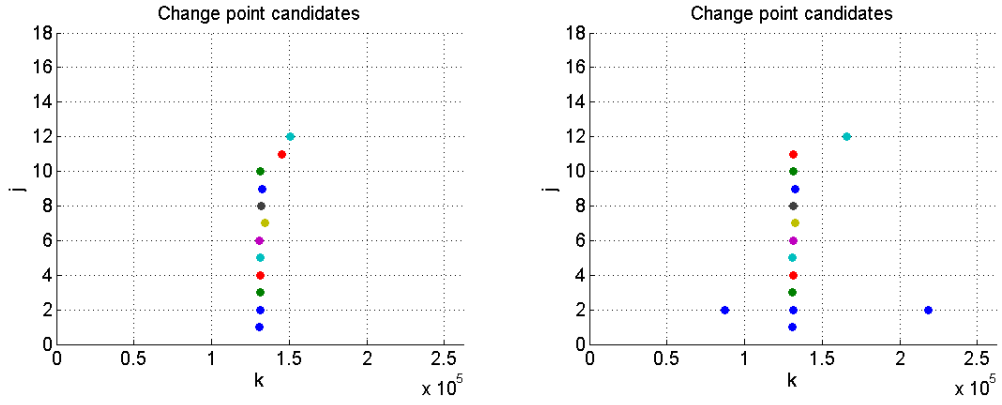

Figure 6.12: The detected change points when the DWT-ICSS algorithm is applied to the trace with a variance change, at the 99% significance level, using `db2` (left) and `db3` (right).

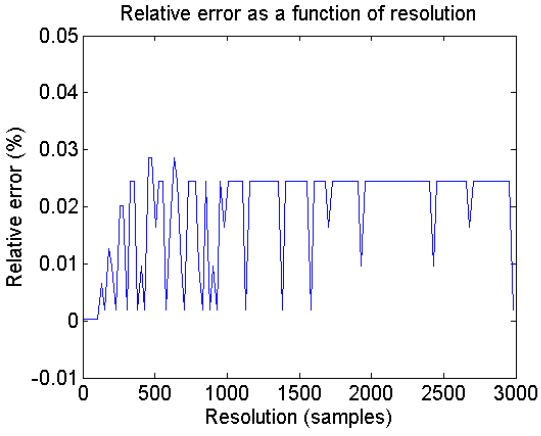

Figure 6.13: Relative error percentage as a function of the resolution of the Hough transform, for the DWT-ICSS analysis (`db1`, 99% significance level) of a trace with a variance change.

	quorum (votes)	q=3			q=5		
	resolution (samples)	$r = 5$	$r = 20$	$r = 200$	$r = 5$	$r = 20$	$r = 200$
95%, `db1`	alignments	131073	131073	131097	131073	131073	131097
	absolute error	0 samples	0 samples	24 samples	0 samples	0 samples	24 samples
99%, `db1`	alignments	131073	131073	131097	131073	131073	131097
	absolute error	0 samples	0 samples	24 samples	0 samples	0 samples	24 samples
99%, `db2`	alignments	131072	131072	131084	none	none	131084
	absolute error	1 sample	1 sample	11 samples			11 samples
99%, `db3`	alignments	131071	131070	131078	none	131070	131078
	absolute error	2 samples	3 samples	5 samples		3 samples	5 samples
99%, `db4`	alignments	none	131069	131074	none	none	131074
	absolute error		4 samples	1 sample			1 sample

Table 6.2: Results of the alignment procedure applied to the DWT-ICSS analysis of an FGN trace with a variance change. The alignments positions and its absolute error are shown.

high resolution (around 5 samples) are appropriate for the variance change case.

6.6.2 DWT-SIC

Figure 6.14 shows the $SIC(k)$ statistic and its associated "corrected" minimum critical value, applied to the `db1` wavelet details. The minimum of the $SIC(k)$ statistic, which is in the middle of the trace in almost all scales, signals the change position. The results are similar to those found with the ICSS; even the bias phenomenon at the higher scales $j = 11, j = 13$ appears again. We could conclude that the problem is in the wavelet transform, and not in the change detection methods, but the authors of the change detection methods warn about the bias introduced by the cumulative sum of squares (not only used by the ICSS, but also by the variance estimation that is part of SIC).

The position of the detected change points by the SIC is plotted in Figure 6.15. As was found in the ICSS case, the higher the significance level, the less the detected points, though in this case the two false points at scale 16^{th} remain as valid (though the nonexistence of aligned changes at the lower scales can give us a hint about its untruth), while a (presumably true) change detected at the 12^{th} scale is lost when the analysis is performed at 99% significance level. The bias phenomenon at the 11^{th} scale is strong enough to survive the critical level in any situation. Apart from this, the performance of SIC and ICSS for this trace is similar. We will not extract general conclusions about the goodness of the algorithms until the statistical test described in the following section.

Regarding the influence of the wavelet order, Figure 6.16 shows the (relative) invariance of the results with the number of vanishing moments of the chosen wavelet, with a slight improvement in the alignment of the true change points.

The results from the alignment detection algorithm are shown in Table 6.3. They are similar to those found in the ICSS case, with a slightly worse performance of the SIC method when the required quorum is high and the resolution is low.

The results are good, even better than those of the DWT-ICSS method. Regarding the influence of the resolution, we get similar results to those shown in Figure 6.13, with a maximum error of 0.027%. Again, the single-change nature of the trace explains the results.

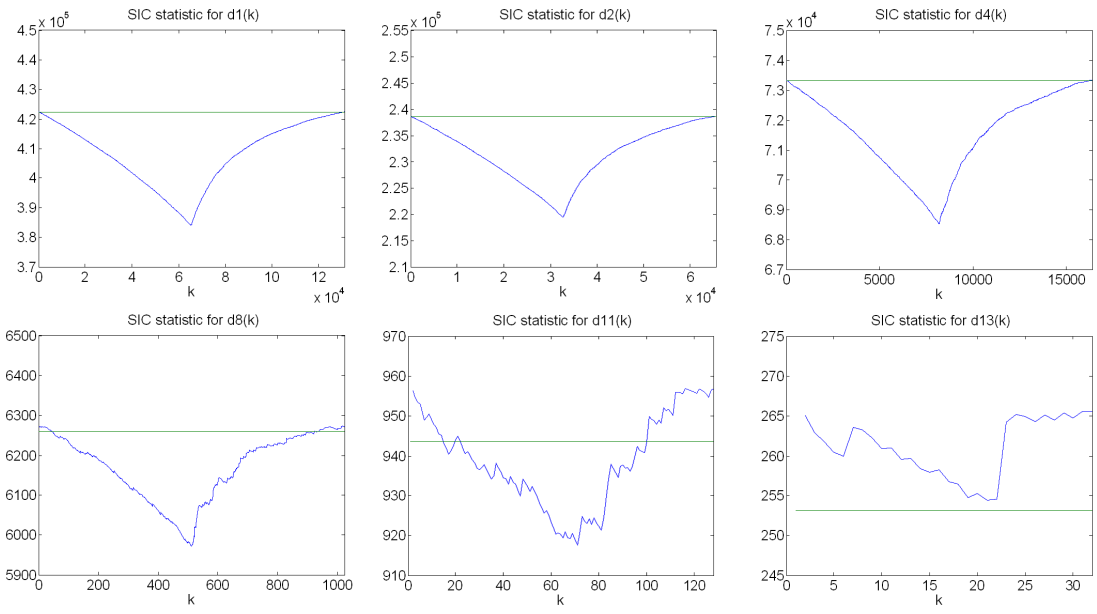

Figure 6.14: SIC(k) statistic applied to the wavelet details of the FGN trace with a variance change, for $j = 1, 2, 3, 4, 8$ and 12, using `db1` wavelet at the 95% significance level. The associated critical level $SIC(n) - c_\alpha$ is plotted on the horizontal line.

6.7 Application to synthetic traces with changes in the Hurst parameter

Let us now discuss what happens when the Hurst parameter changes, whereas the mean and the variance stay constant. For this purpose, we firstly consider a trace of 262144 samples, the first and the second half of which are given by the `fgn05.mat` and the `fgn09.mat` trace, respectively.

6.7.1 DWT-ICSS

As can be seen in Figure 6.17, the change point location is evident at almost every scale, with the exception of $j = 3$, due to the blind point phenomenon described in Section 5.5. As outlined in Section 4.3, a more drastic change in variance results in a higher slope (or, equivalently, in a higher peak) for $D_j(k)$. This can be easily checked by looking at Figure 6.18: as in the variance change case, the change point is located at the peaks (absolute maxima) of the ICSS statistic.

Figures 6.19, 6.20 and 6.21 show the detected changes for a set of different configurations of the significance level and the mother wavelet. What we can see from these diagrams is that the changes at scales $j = 1 \ldots 3$ are negative; i.e., the left side of the trace has a bigger variance than the right side. The situation is the opposite for the higher scales ($j > 4$). This is the situation shown in Figure 5.7. Above scale 13 the smoothing effect of the DWT makes the ICSS unable to find the changes. All the detected changes for $j \leq 10$ are quite well located, with the exception of scale $j = 3$ (which corresponds to the "blind point" phenomenon illustrated in Section 5.5 and Figure

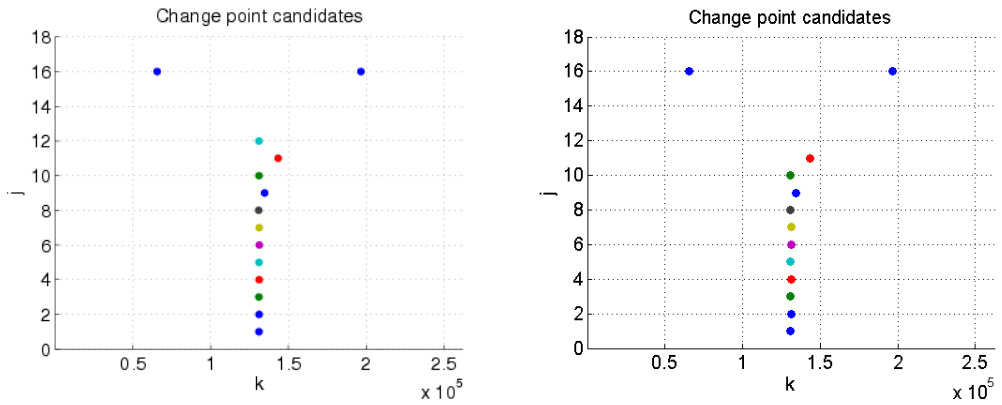

Figure 6.15: Detected change points when the DWT-SIC algorithm is applied to the trace with a variance change, using `db1` wavelet, for the 95% and 99% significance level (left and right, respectively).

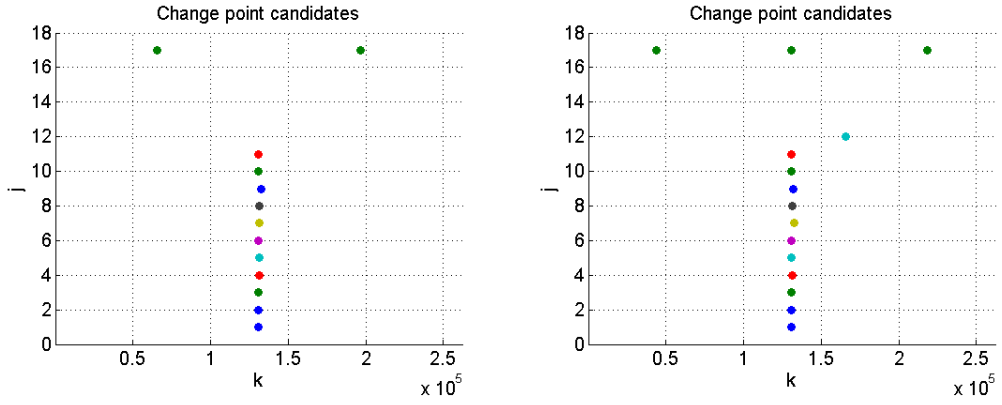

Figure 6.16: Detected change points when the DWT-SIC algorithm is applied to the trace with a variance change, at the 99% significance level, using `db2` and `db3` wavelets (left and right, respectively).

	quorum (votes)	q=3			q=5		
	resolution (samples)	$r = 5$	$r = 20$	$r = 200$	$r = 5$	$r = 20$	$r = 200$
95%, `db1`	alignments	131073	131073	131089	none	none	131089
	absolute error	0 samples	0 samples	16 samples			16 samples
99%, `db1`	alignments	131073	131073	131091	none	none	131091
	absolute error	0 samples	0 samples	18 samples	none		18 samples
99%, `db2`	alignments	131072	131069	131088	none	131069	131088
	absolute error	1 sample	4 samples	15 samples		4 samples	15 samples
99%, `db3`	alignments	131072	131070	131084	none	131070	131084
	absolute error	1 sample	3 samples	11 samples		3 samples	11 samples
99%, `db4`	alignments	none	131068	131070	none	none	131070
	absolute error		5 samples	3 samples			3 samples

Table 6.3: Results of the alignment procedure applied to the DWT-SIC analysis of an FGN trace with a variance change.

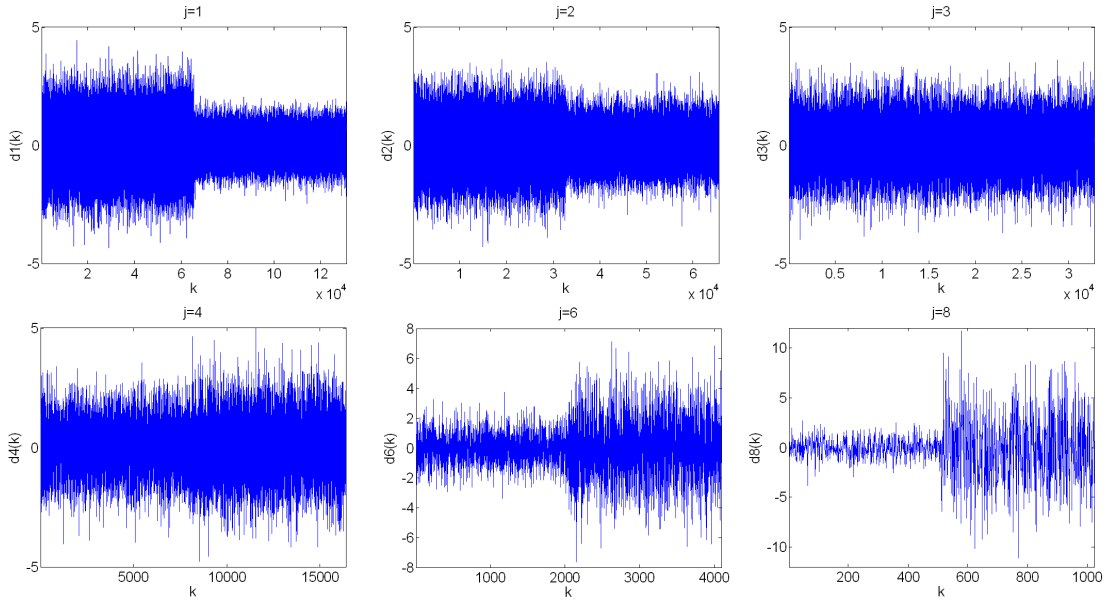

Figure 6.17: DWT detail coefficients of the trace with a Hurst parameter change ($0.5 \rightarrow 0.9$), for $j = 1, 2, 3, 4, 6$ and 8, using `db1` wavelet. The change point is located in the middle of the series, at $t_1 = 131072$.

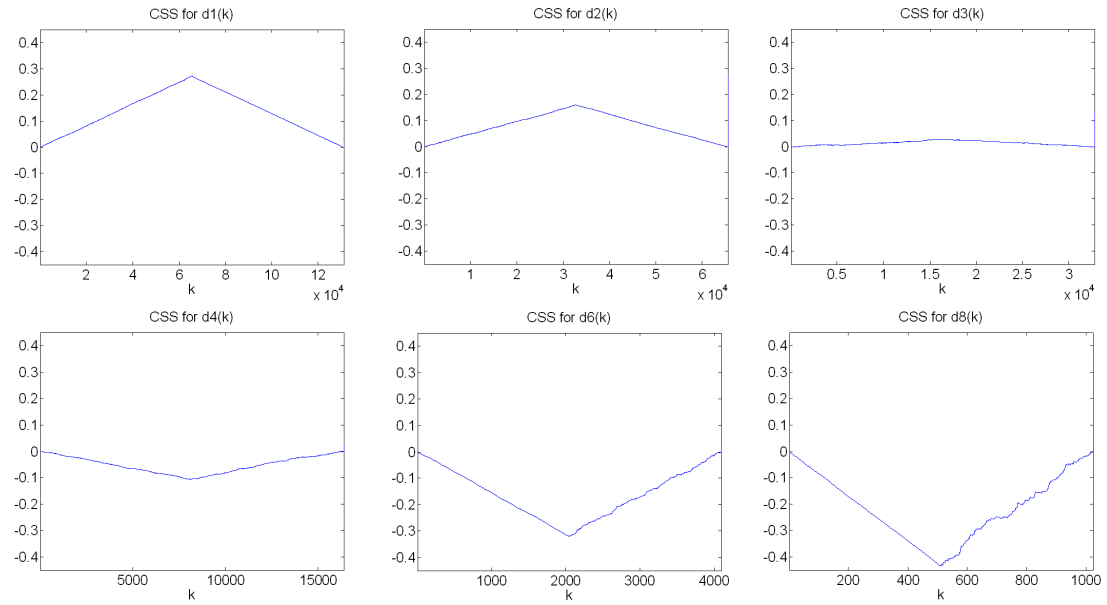

Figure 6.18: CSS statistic applied to the DWT detail coefficients of the trace with a Hurst parameter change ($0.5 \rightarrow 0.9$).

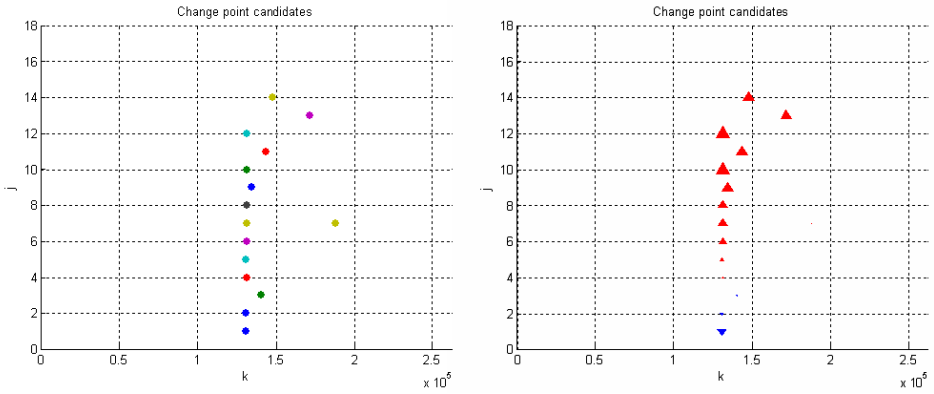

Figure 6.19: Detected change points when the DWT-ICSS is applied to the trace with a Hurst parameter change ($0.5 \rightarrow 0.9$), using `db1` wavelet, at the 95% significance level. *Left*: the position of the change points. *Right*: size and sign of the detected variance change.

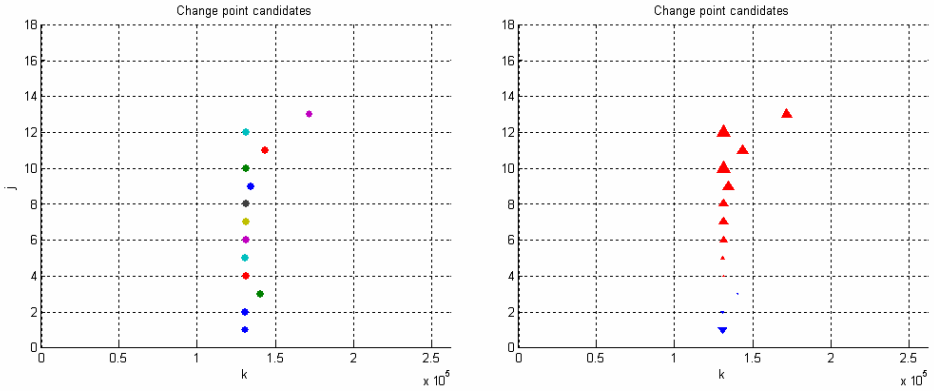

Figure 6.20: Detected change points when the DWT-ICSS is applied to the trace with a Hurst parameter change ($0.5 \rightarrow 0.9$), using `db1` wavelet, at the 99% significance level. *Left*: the position of the change points. *Right*: size and sign of the detected variance change.

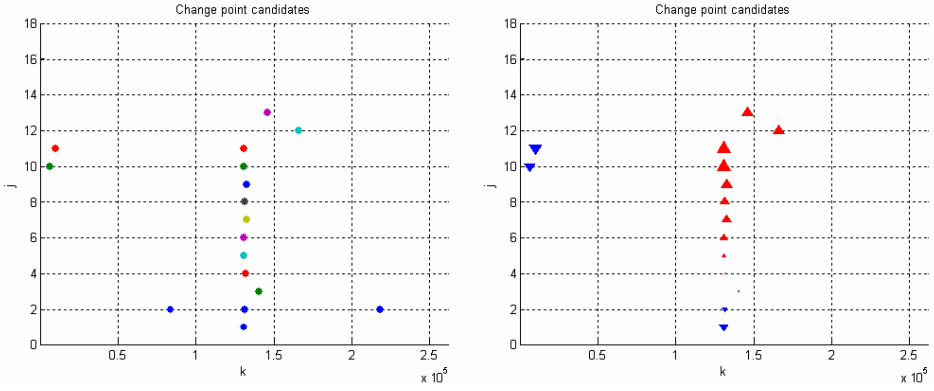

Figure 6.21: Detected change points when the DWT-ICSS is applied to the trace with a Hurst parameter change ($0.5 \rightarrow 0.9$), using `db3` wavelet, at the 99% significance level. *Left*: the position of the change points. *Right*: size and sign of the detected variance change.

5.7) and the dispersion found at the higher scales (specifically at scales 9, 11 and 13), which can be explained as a collateral effect of the smoothing property of the DWT at the lower frequencies.

When the analysis is performed at the 95% significance level (Figure 6.19), the results do not change too much with respect to the 99% case, with the exception of an almost imperceptible false change point at scale $j = 7$ located around position $k = 190000$, possibly caused by a local fluctuation of variance that is big enough to exceed the critical value, and the detection of a change point at scale 14.

Finally, performing the analysis with a different wavelet (in this case, `db3`) at the 99% significance level, we get the results shown in Figure 6.21. The main differences are the false change points detected at the beginning of the trace for the higher scales, which are a product of the boundary effects of the transform; some spurious changes at the second scale; and finally the bias introduced by the ICSS in the "true" change point at the higher scales. The longer filter length of the higher order wavelet mother introduces a slightly higher correlation; this, combined with the boundary effects, fools the ICSS. But it must be emphasized that the true changes are still well located, at least at the lower scales.

Regarding the alignment detection algorithm, we get the results shown in Table 6.4. The results are slightly worse than those found for the variance change case, due to the higher dispersion of change points around the blind zone (caused by the low variance ratios) and on the higher scales (where the smoothing effect of the transform is stronger). However, with a 200-sample resolution we catch the change point in almost every situation. We conclude that a Hurst parameter change needs a more favorable combination of quorum and resolution; $q = 3 - 5$ changes and $r \geq 200$ samples seem a good choice.

	quorum (votes)	q=3			q=5		
	resolution (samples)	$r = 5$	$r = 20$	$r = 200$	$r = 5$	$r = 20$	$r = 200$
95%, `db1`	alignments	none	none	131087	none	none	131087
	absolute error			13 samples			13 samples
99%, `db1`	alignments	none	none	131087	none	none	131087
	absolute error			13 samples			13 samples
99%, `db2`	alignments	none	none	131050	none	none	131050
	absolute error		none	23 samples			23 samples
99%, `db3`	alignments	131072	131073	131073	none	none	none
	absolute error	1 sample	0 samples	0 samples			
99%, `db4`	alignments	none	none	131049	none	none	131049
	absolute error		none	24 samples			24 samples

Table 6.4: Results of the alignment procedure applied to the DWT-ICSS analysis of the trace with a Hurst parameter change ($0.5 \rightarrow 0.9$).

6.7.2 DWT-SIC

Figure 6.22 shows the detected change points when the SIC statistic is used as a change point detector, for the $H = 0.5 \rightarrow H = 0.9$ transition case using `db1` wavelet at the 99% significance level, while Figure 6.23 repeats the analysis using `db3` wavelet. In both cases the results are much

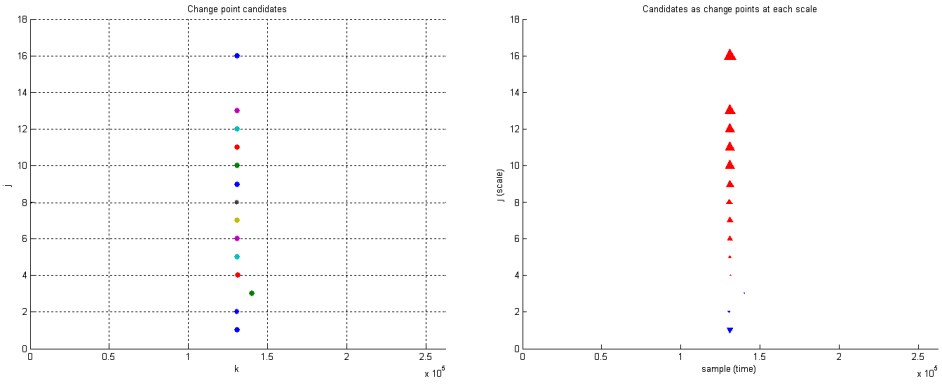

Figure 6.22: Detected change points when the DWT-SIC is applied to the trace with a Hurst parameter change ($0.5 \rightarrow 0.9$), using `db1` wavelet, at the 99% significance level. *Left*: the position of the change points. *Right*: size and sign of the detected variance change.

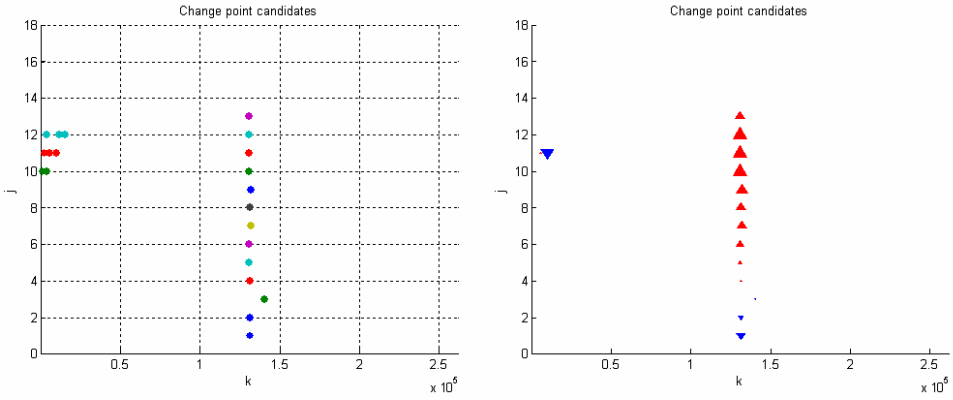

Figure 6.23: The detected change points when the DWT-SIC is applied to the trace with a Hurst parameter change ($0.5 \rightarrow 0.9$), using `db3` wavelet, at the 99% significance level. *Left*: the position of the change points. *Right*: size and sign of the detected variance change.

	quorum (votes)	q=3			q=5		
	resolution (samples)	$r = 5$	$r = 20$	$r = 200$	$r = 5$	$r = 20$	$r = 200$
99%, `db1`	alignments	131072	131072	131073	131072	131072	131073
	absolute error	1 sample	1 sample	0 samples	1 sample	1 sample	0 samples
99%, `db2`	alignments	131073	131073	131078	131073	131073	131078
	absolute error	0 samples	0 samples	5 samples	0 samples	0 samples	5 samples
99%, `db3`	alignments	131072	131073	131074	131072	131073	131074
	absolute error	1 sample	0 samples	1 sample	1 sample	0 samples	1 sample
99%, `db4`	alignments	131073	131073	131081	131073	131073	131081
	absolute error	0 samples	0 samples	8 samples	0 samples	0 samples	8 samples

Table 6.5: Results of the alignment procedure applied to the DWT-SIC analysis of the trace with a Hurst parameter change ($0.5 \rightarrow 0.9$).

better than those given by the ICSS, with a very good alignment for the `db1` case, due to a very low dispersion around the true change. However, the SIC shows the sensitivity to the "blind zone" at scale $j = 3$, which is not caused by the variance change detection method but by the input data. For the `db3`-based analysis the SIC produce the false points detected at the beginning of the trace at the higher scales, while at the same scales the bias phenomenon does not appear. We conclude that `db2` and `db3` offer a good trade-off between accuracy of the detected changes and amount of spurious changes, and that the use of higher order wavelets should be avoided unless necessary for wiping out polynomial trends.

The alignment detection algorithm (Table 6.5) returns better results than those obtained with DWT-ICSS, as was expected from the good change alignment. SIC seems to be more accurate in the localization of each change point, and its alignment is therefore easier to find, even at very fine resolutions (higher scales). Even at very high resolutions (5 samples) and high quorum (5 samples), the method performs very well.

6.7.3 Empirical assessment of the power of DWT-ICSS when the Hurst parameter changes

The previous examples were useful for illustrating how the algorithms work and extracting some conclusions about the parameters used in the analysis, but we need a statistically meaningful analysis. In order to empirically assess the power of the procedure, we generated 1000 independent realizations of 262144 samples with $H = 0.5$ and $H = 0.9$ in the first and second half of the trace, respectively. The realizations are the same for all the results presented in this Section (for the DWT-ICSS method) and in Section 6.7.4 (for the DWT-SIC method), which allows us to perform a fair comparison.

The analysis was performed with DWT-ICSS using `db1` wavelet at the 95% and 99% significance level, for $j = 1 \dots 17$. Figures 6.24 and 6.25 show the histograms of some scales ($j = 1$ and $j = 2$ for being the most populated, $j = 3$ for being the blind point, $j = 4, 6$ and 8 as representatives of the higher scales) for the ICSS case. Tables 6.6 and 6.7 show the results obtained. For each scale, we include the true change position, the positions that are located in a (approximately) $\pm 0.5\%$ deviation from the true change point, the total number of detected change points, the "good" changes in the $\pm 0.5\%$ around the true change, its percentage (relative to the total number of detected changes), the mean of the detected change points, and the corrected mean of the "good" change points. Table 6.8 includes the results for a 99% significance level, done with `db3`.

The main conclusions extracted from the data are:

- The total amount of detected changes (per scale) seems to be quite stable for the scales up to $j = 13$ (around 1100 for the 95% significance level case and `db1`, around 1020 for the 99% `db1` case). For higher scales, it suddenly drops to very low values and goes to zero. For the 99% `db3` case, the total number is even higher for scales $9 - 13$, due to the spurious points introduced at those scales at the beginning of the trace (the same effect described in Sections

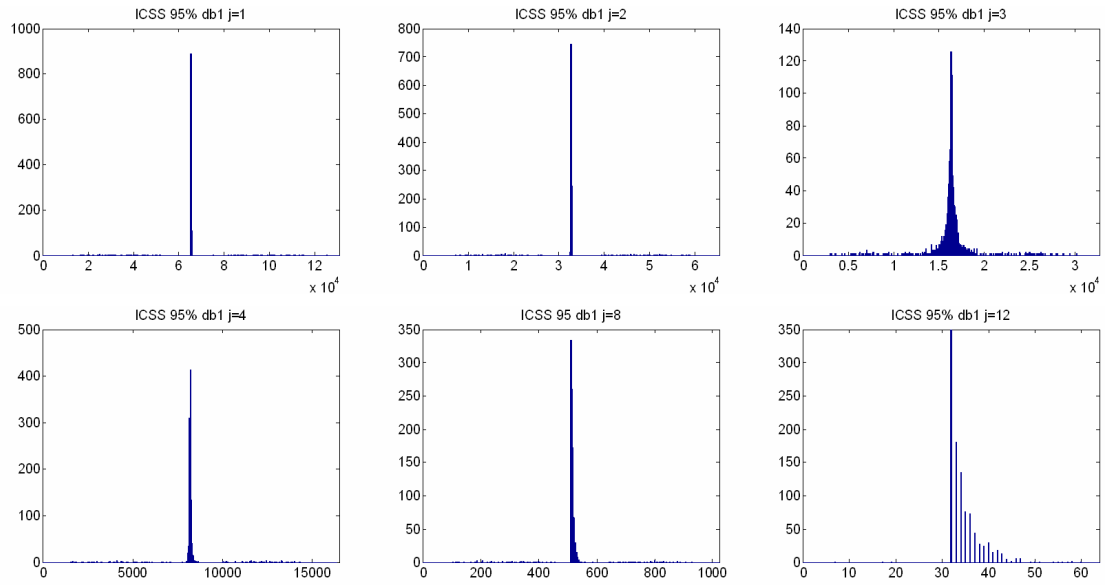

Figure 6.24: Histograms of the detected change points when the DWT-ICSS (with `db1`) is applied to 1000 independent realizations whose H parameter changes from 0.5 to 0.9 in the middle of the series, analyzed at the 95% significance level.

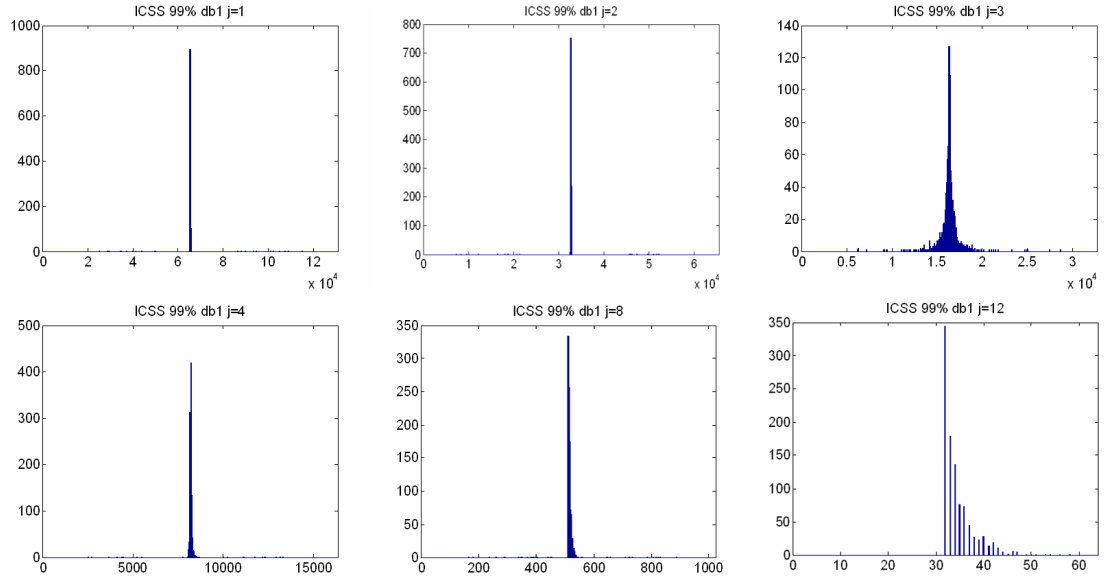

Figure 6.25: Histograms of the detected change points when the DWT-ICSS (with `db1`) is applied to 1000 independent realizations whose H parameter changes from 0.5 to 0.9 in the middle of the series, analyzed at the 99% significance level.

Scale	True change	Changes	"Good" changes	Raw mean	Corr. mean
1	65537 (65210,65864)	1106	**1000** (90.4%)	65423.2	65528.4
2	32769 (32606,32932)	1109	**989** (89.2%)	32817.1	32755.3
3	16385 (16304,16466)	1093	**239** (21.9%)	16336.3	16383.7
4	8193 (8153,8233)	1105	**726** (65.7%)	8238.2	8196.0
5	4097 (4077,4117)	1097	**860** (78.4%)	4132.6	4100.1
6	2049 (2039,2059)	1100	**827** (75.2%)	2049.7	2050.5
7	1025 (1020,1030)	1083	**809** (74.7%)	1034.7	1025.6
8	513 (511,515)	1096	**665** (60.7%)	515.2	512.9
9	257 (256,258)	1082	**629** (58.1%)	261.1	256.7
10	129 (129)	1059	**168** (15.9%)	132.2	129.0
11	65 (65)	1046	**159** (15.2%)	67.5	65.0
12	33 (33)	1020	**181** (17.7%)	34.6	33.0
13	17 (17)	1018	**207** (20.3%)	17.9	17.0
14	9 (9)	985	**242** (24.6%)	9.3	9.0
15	5 (5)	80	**0**	6.4	5.0
16	3 (3)	0	**0**		
17	2 (2)	0	**0**		

Table 6.6: Detected change points at each scale for a trace whose H parameter changes from 0.5 to 0.9 in the middle of the series, analyzed with the DWT-ICSS (`db1`) at the 95% significance level.

Scale	True change	Changes	"Good" changes	Raw mean	Corr. mean
1	65537 (65210,65864)	1022	**1000** (97.8%)	65501.0	65528.9
2	32769 (32606,32932)	1016	**998** (98.2%)	32798.9	32757.5
3	16385 (16304,16466)	1012	**233** (23.0%)	16292.1	16377.1
4	8193 (8153,8233)	1018	**747** (73.4%)	8222.9	8197.2
5	4097 (4077,4117)	1019	**871** (85.5%)	4105.7	4100.1
6	2049 (2039,2059)	1020	**837** (82.1%)	2049.1	2050.6
7	1025 (1020,1030)	1021	**808** (79.1%)	1026.0	1025.5
8	513 (511,515)	1025	**729** (71.1%)	513.0	512.9
9	257 (256,258)	1023	**700** (68.4%)	256.0	256.6
10	129 (129)	1023	**190** (18.6%)	128.0	129.0
11	65 (65)	1007	**182** (18.1%)	64.0	65.0
12	33 (33)	1007	**146** (14.5%)	32.1	33
13	17 (17)	1001	**169** (16.9%)	16.3	17
14	9 (9)	215	**21** (9.8%)	8.3	9
15	5 (5)	5	**0** (0.0%)	7.0	5
16	3 (3)	0	**0**		
17	2 (2)	0	**0**		

Table 6.7: Detected change points at each scale for a trace whose H parameter changes from 0.5 to 0.9 in the middle of the series, analyzed with the DWT-ICSS (`db1`) at the 99% significance level.

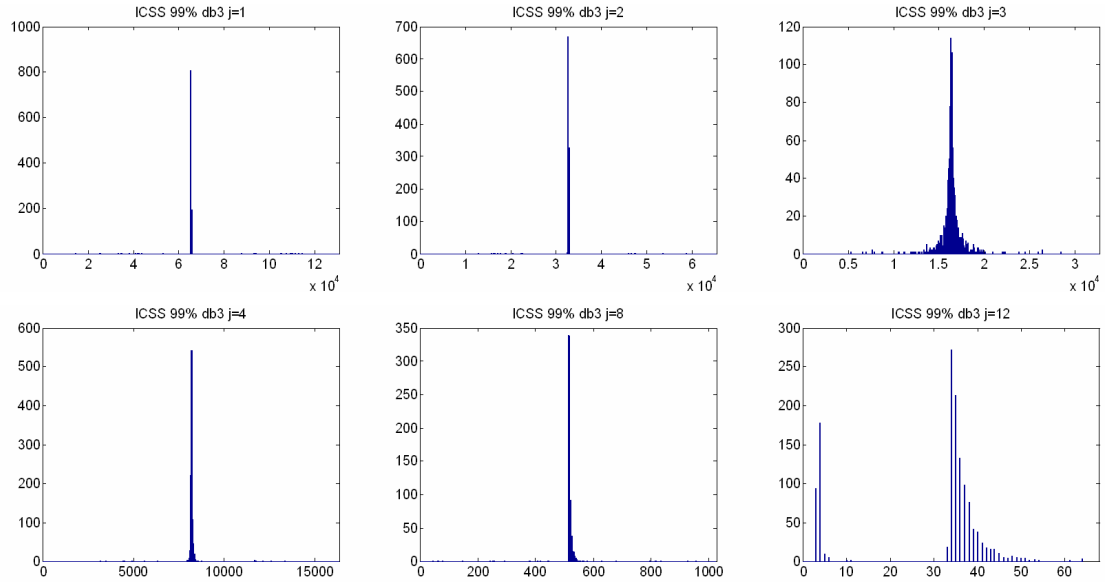

Figure 6.26: Histograms of the detected change points when the DWT-ICSS (with `db3`) is applied to 1000 independent realizations whose H parameter changes from 0.5 to 0.9 in the middle of the series, analyzed at the 99% significance level.

Scale	True change	Changes	"Good" changes	Raw mean	Corr. mean
1	65537 (65210,65864)	1025	**1000** (97.6%)	65742.3	65529.7
2	32769 (32606,32932)	1019	**996** (97.8%)	32667.7	32757.3
3	16385 (16304,16466)	999	**217** (21.7%)	16354.2	16386.5
4	8193 (8153,8233)	1014	**766** (75.5%)	8208.2	8198.4
5	4097 (4077,4117)	1019	**859** (84.3%)	4113.3	4101.2
6	2049 (2039,2059)	1026	**834** (81.3%)	2049.7	2052.0
7	1025 (1020,1030)	1025	**698** (68.1%)	1027.1	1027.1
8	513 (511,515)	1020	**493** (48.3%)	516.9	514.1
9	257 (256,258)	1047	**317** (30.3%)	254.0	257.8
10	129 (129)	1147	**46** (4.0%)	120.3	129.0
11	65 (65)	1211	**18** (1.5%)	58.5	65.0
12	33 (33)	1295	**18** (1.4%)	29.4	33.0
13	17 (17)	1139	**30** (2.6%)	17.8	17.0
14	9 (9)	279	**0** (0.0%)	1.4	
15	5 (5)	31	**0** (0.0%)	8.8	
16	3 (3)	0	**0**		
17	2 (2)	0	**0**		

Table 6.8: Detected change points at each scale for a trace whose H parameter changes from 0.5 to 0.9 in the middle of the series, analyzed with the DWT-ICSS (`db3`) at the 99% significance level.

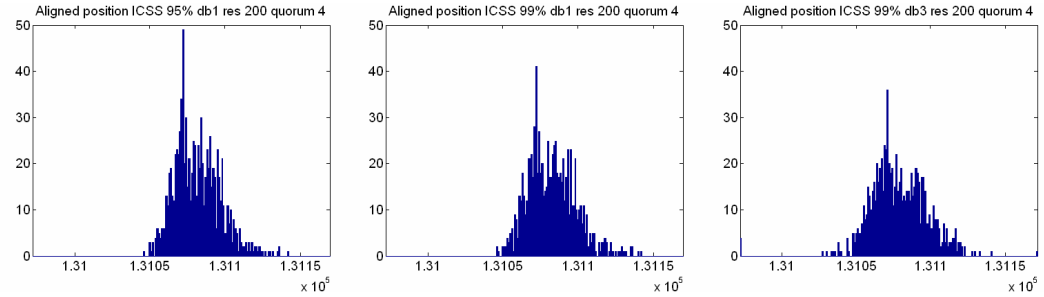

Figure 6.27: Histograms of the aligned change points when the DWT-ICSS is applied to 1000 independent realizations whose H parameter changes from 0.5 to 0.9 (15 scales, resolution=200 samples, quorum=4 changes). *Left*: `db1`, 95% significance level. *Middle*: `db1`, 99% significance level. *Right*: `db3`, 99% significance level. Note that the x axis has been zoomed on the region 131072 ± 100.

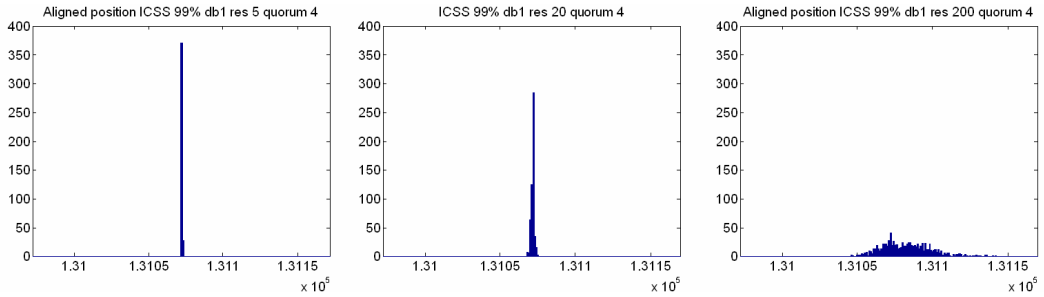

Figure 6.28: Histograms of the aligned change points when the DWT-ICSS is applied to 1000 independent realizations whose H parameter changes from 0.5 to 0.9, analyzed with the DWT-ICSS at the 99% significance level with `db1` at different values of resolution (15 scales, quorum=4 changes). *Left*: resolution=5 samples. *Middle*: resolution=20 samples. *Right*: resolution=200 samples.

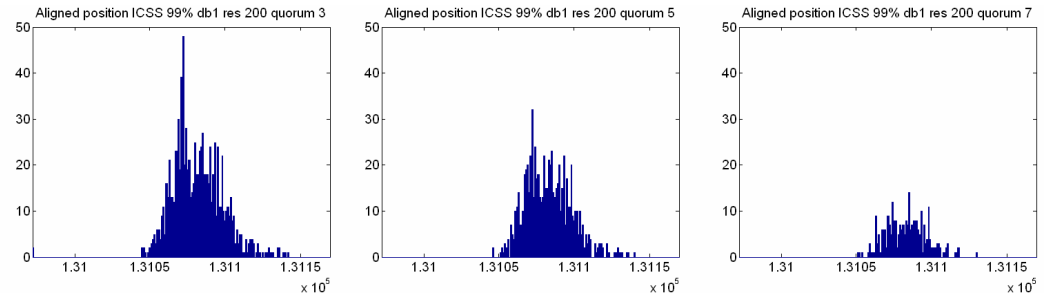

Figure 6.29: Histograms of the aligned change points when the DWT-ICSS is applied to 1000 independent realizations whose H parameter changes from 0.5 to 0.9, analyzed with the DWT-ICSS at the 99% significance level with `db1` at different values of quorum (15 scales, resolution=200 samples). *Left*: quorum=3 changes. *Middle*: quorum=5 changes. *Right*: quorum=7 changes.

		95% sign. level				99% sign. level			
		db1		db3		db1		db3	
resol.	quorum	Total	*True*	Total	*True*	Total	*True*	Total	*True*
5 smpls	3 chg	747	**726** (97.2%)	514	**506** (98.4%)	659	**642** (97.4%)	458	**451** (98.3%)
	4 chg	494	**494** (100%)	245	**245** (100%)	399	**399** (100%)	208	**208** (100%)
	5 chg	271	**271** (100%)	89	**89** (100%)	180	**180** (100%)	64	**64** (100%)
	6 chg	115	**115** (100%)	20	**20** (100%)	71	**71** (100%)	16	**16** (100%)
	7 chg	45	**45** (100%)	5	**5** (100%)	20	**20** (100%)	4	**4** (100%)
20 smpls	3 chg	859	**841** (97.9%)	697	**666** (98.4%)	787	**773** (98.2%)	662	**653** (98.6%)
	4 chg	615	**615** (100%)	418	**418** (100%)	536	**536** (100%)	369	**369** (100%)
	5 chg	395	**395** (100%)	210	**210** (100%)	293	**293** (100%)	166	**166** (100%)
	6 chg	197	**197** (100%)	47	**47** (100%)	125	**125** (100%)	35	**35** (100%)
	7 chg	70	**70** (100%)	12	**12** (100%)	41	**41** (100%)	11	**11** (100%)
200 smpls	3 chg	1001	**985** (98.4%)	978	**956** (97.8%)	985	**967** (98.2%)	972	**952** (98.0%)
	4 chg	922	**919** (99.7%)	864	**860** (99.6%)	880	**876** (99.5%)	856	**851** (99.4%)
	5 chg	793	**792** (99.9%)	671	**671** (100%)	744	**743** (99.9%)	632	**631** (99.8%)
	6 chg	613	**613** (100%)	414	**414** (100%)	524	**524** (100%)	370	**370** (100%)
	7 chg	381	**381** (100%)	191	**191** (100%)	274	**274** (100%)	168	**168** (100%)

Table 6.9: Detected aligned changes for a trace whose H parameter changes from 0.5 to 0.9, analyzed with the DWT-ICSS. The *true* changes are defined as those around ± resolution/2 positions around the true change position.

6.7.1 and 6.7.2), probably due to the residual correlation introduced by the longer wavelet at those scales.

- Regarding the *true* changes, for the `db1` cases its number decreases slowly from around 1000 for the first scale, and remains high until $j = 9$ (700 for 99%, 629 for 95%); for $j = 10$ and above, the number rapidly decreases and falls to zero in a few scales. For the `db3`, the drop starts at a lower scale ($j = 8$) and goes faster to zero. Therefore, only the scales up to $j \approx 9$ are useful for finding the alignments, and the higher the wavelet, the fewer the scales are useful.

- In general, the 99% significance level analysis returns fewer true and false points (per scale) than the 95% case, though its *goodness* (in terms of proximity to the true change point) is better. Therefore, the higher the significance level, the more precise the location of the changes. However, the significance cannot be increased arbitrarily, since valid changes can be taken as false ones. The 99% seems to offer a good performance trade-off.

- As the scale j increases, fewer true points per scale are detected, due to the bias phenomenon that disperses the points around the true position. This can be seen in the histogram of scale $j = 12$.

- Scale $j = 3$ is an exception in several ways: though the number of total detected changes is similar to that found in other scales, its "trueness" is far under the results found for its *neighbors* $j = 2$ and $j = 4$, and its histogram is clearly more dispersed. This is the result of the *blind zone* phenomenon[4].

[4]Recall that we call *blind zone* those scales where the variances are similar at both sides of the change point.

- The higher scales suffer from a positive bias that worsens when j increases (compare the histograms of $j = 8$ and $j = 12$). The (relatively) few points found at the higher scales and the alignment procedure will soften this effect.

- For both cases of significance, the raw mean obtained by averaging *all* the detected change points does a fair job and is quite approximate to the true change position. The corrected mean performs even better. The use of an algorithm capable of clustering the points in terms of their alignment would have a similar behavior, and therefore could improve the results. This algorithm will be presented in Section 5.6.

Finally, the results obtained with the clustering and alignment algorithm are shown in Figures 6.27, 6.28, and 6.29, and Table 6.9. In the latter the *true* changes are defined as those around ± resolution/2 positions around the true change position[5]. Three different analyses were performed by changing the wavelet (`db1`, `db3`), the significance level (95%, 99%), the resolution (5, 20, and 200 samples), and the quorum (3, 4 or 5 aligned changes).

- Regarding the significance level, the 95% case returns the best results, in terms of total changes and true changes, though some of them may be spurious or repeated changes. From the experience obtained with our previous experiments, we prefer choosing the 99% significance level.

- Regarding the wavelet, `db1` returns more total detected changes than `db3`, due to the smoothing effect of the latter. The shorter wavelet is also better at the number of true changes and its percentage, probably due to the higher number of spurious changes detected by `db3`.

- The higher the resolution, the more total and true changes are detected, as expected, while the opposite happens with the quorum, which can make the change detection drop to near zero.

- A good trade-off was found for the following combination of parameters: 99%, resolution=200 samples, quorum=4 changes. For the `db1` wavelet this combination detects 876 true changes out of 1000, more than 85% of the true change points, while keeping good selectivity against false changes (only four more points were found outside the ±50 samples region) and high exigency (alignment of 4 changes across scales[6]). For the `db3` case (useful in case there is some linear or parabolic trend superimposed in the data) we obtained slightly worse results, but still over 85% of the true change points.

We now repeat the experiment with a less abrupt H parameter change, where the first segment has $H = 0.7$, and the second one $H = 0.8$. The LogScale Diagrams for two such traces are shown in Figure 6.30, from which we can expect that scales $j = 3$ and $j = 4$ are the "blind zones",

[5] 131072 ± 2 positions for the case of resolution=5, ±10 positions for resolution=20, and ±100 positions for resolution=200.

[6] Recall that the tests performed with pure FGN traces never showed more than 2 aligned points across scales.

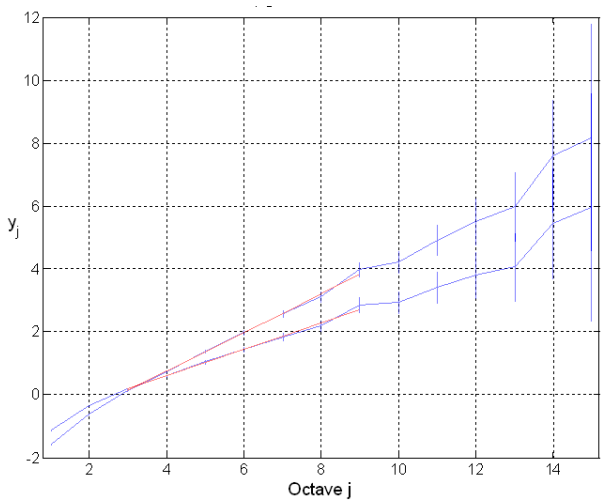

Figure 6.30: Variance distribution at each scale for two FGN traces with scaling parameters $H = 0.7$ and $H = 0.8$.

and that the variance changes are much less important than those found at each scale for the $H = 0.5 \rightarrow 0.9$ case (recall Figure 5.7). The proximity of the variances even at the higher scales (where the regression lines are more distant) will make the variance change detection algorithm return worse results. The results of the 1000 independent runs at 99% significance level are shown in Figure 6.31 and Table 6.10. As expected, scales 3 and 4 present very disperse histograms, and this phenomenon appears even at the higher scales. This redounds in a lower empirical power of the "good" change points, though the total number of detected changes is comparable to the $0.5 - 0.9$ case.

The results obtained with the clustering and alignment algorithm are listed in Table 6.11. Though compared with the $H = 0.5 \rightarrow 0.9$ the amount of detected changes was expected to drop, the fall is very important, even for the highest values of resolution. That is why we extended our analysis to resolutions of 1000, 2000 and 5000 samples. Applying the same reasonings we used for the $H = 0.5 \rightarrow 0.9$ case, we select the following combination of parameters: 99%, resolution=5000 samples, quorum=4 changes. With `db1` this combination detects 929 true changes out of 1000 (more than 90% of the true change points), maintaining the selectivity against false changes (only 16 more point were found), while the `db3` analysis performs even a bit better, with 960 true changes and only 17 false changes. We would probably lower the resolution to a value of 3500 or 4000 samples and still get more than 85% of the changes.

An important conclusion for our method is that we face a trade-off between the granularity of the changes of the Hurst parameter and the time resolution at which we can locate the change. An abrupt change of H ($0.5 \rightarrow 0.9$) may be located with an accuracy of 200 samples, while a smoother one ($0.7 \rightarrow 0.8$) needs at least 4000 samples to be detected. In the experiments with real traces, 5000 samples is the typical resolution we will use.

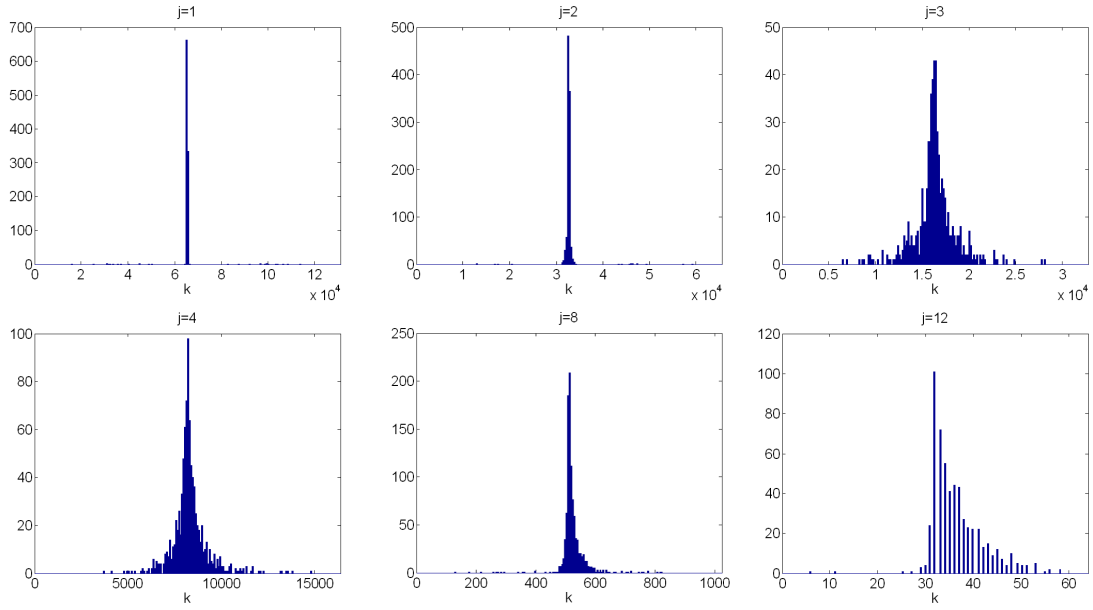

Figure 6.31: Histograms of the detected change points when the DWT-ICSS (with `db1`) is applied to 1000 independent realizations whose H parameter changes from 0.7 to 0.8 in the middle of the series, analyzed at the 95% significance level.

Scale	True change	Changes	"Good" changes	Raw mean	Corr. mean
1	65537 (65210,65864)	1028	**981** (95.4%)	65566.4	65516.6
2	32769 (32606,32932)	1017	**676** (66.5%)	32824.9	32761.4
3	16385 (16304,16466)	590	**48** (8.1%)	16198.0	16386.3
4	8193 (8153,8233)	909	**94** (10.3%)	8311.1	8197.2
5	4097 (4077,4117)	1021	**277** (27.1%)	4133.7	4097.6
6	2049 (2039,2059)	1014	**301** (29.7%)	2077.2	2049.6
7	1025 (1020,1030)	1011	**331** (32.7%)	1039.7	1025.3
8	513 (511,515)	1026	**246** (24.0%)	523.7	512.9
9	257 (256,258)	1008	**261** (25.9%)	264.3	256.8
10	129 (129)	1002	**93** (9.3%)	134.4	129.0
11	65 (65)	948	**111** (18.1%)	69.7	65.0
12	33 (33)	578	**72** (12.5%)	36.6	33
13	17 (17)	114	**7** (6.1%)	20.6	17
14	9 (9)	10	**0** (0.0%)	13.0	9
15	5 (5)	0	**0**		
16	3 (3)	0	**0**		
17	2 (2)	0	**0**		

Table 6.10: Detected change points at each scale for a trace whose H parameter changes from 0.7 to 0.8 in the middle of the series, analyzed with the DWT-ICSS (`db1`) at the 99% significance level.

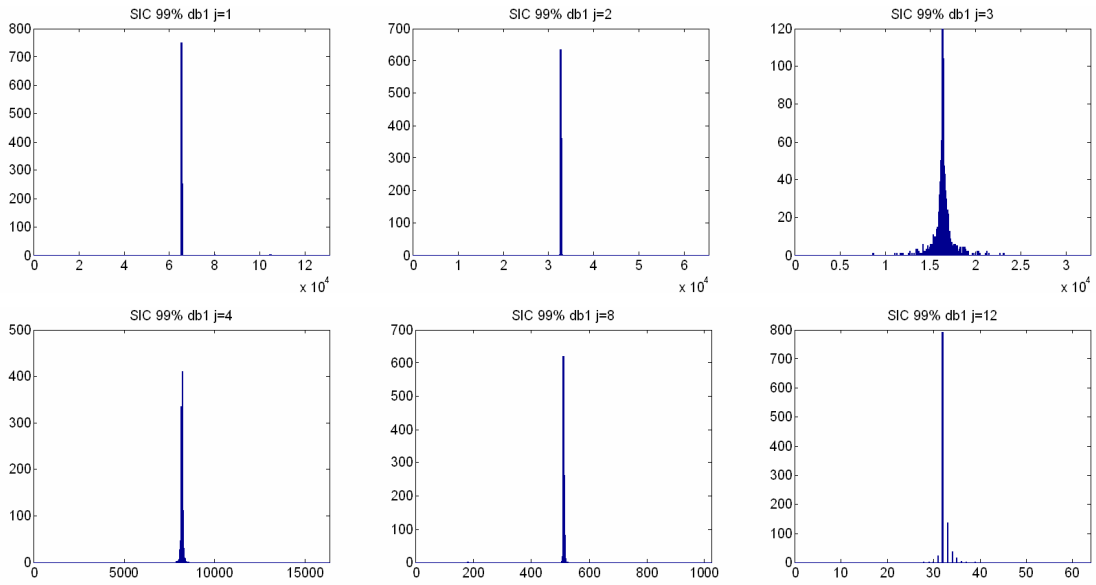

Figure 6.32: Histograms of the detected change points when the DWT-SIC (with `db1`) is applied to 1000 independent realizations whose H parameter changes from 0.5 to 0.9 in the middle of the series, analyzed at the 99% significance level.

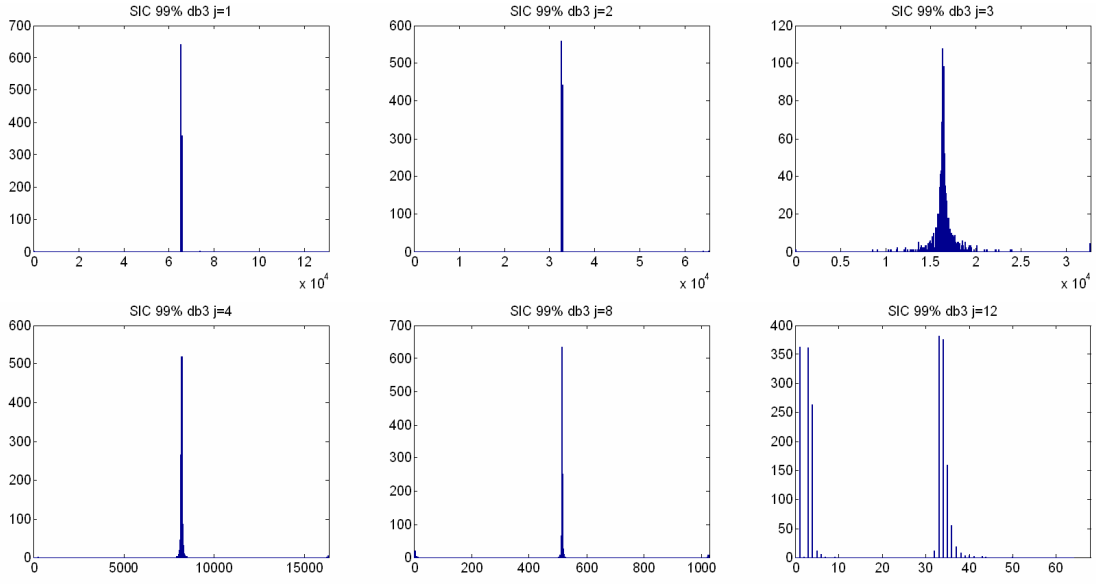

Figure 6.33: Histograms of the detected change points when the DWT-SIC (with `db3`) is applied to 1000 independent realizations whose H parameter changes from 0.5 to 0.9 in the middle of the series, analyzed at the 99% significance level.

		95% sign. level				99% sign. level			
		db1		db3		db1		db3	
resol.	quorum	Total	*True*	Total	*True*	Total	*True*	Total	*True*
5 smpls	3 chg	55	**47** (85.4%)	27	**27** (100%)	40	**32** (80.0%)	28	**27** (96.4%)
	4 chg	8	**8** (100%)	5	**5** (100%)	4	**4** (100%)	4	**4** (100%)
	5 chg	2	**2** (100%)	1	**1** (100%)	0	**0**	0	**0**
	6 chg	0	**0**	0	**0**	0	**0**	0	**0**
	7 chg	0	**0**	0	**0**	0	**0**	0	**0**
20 smpls	3 chg	83	**75** (90.3%)	47	**47** (100%)	52	**44** (84.6%)	42	**41** (97.6%)
	4 chg	11	**11** (100%)	6	**6** (100%)	6	**6** (100%)	5	**5** (100%)
	5 chg	3	**3** (100%)	1	**1** (100%)	0	**0**	0	**0**
	6 chg	0	**0**	1	**1** (100%)	0	**0**	0	**0**
	7 chg	0	**0**	0	**0**	0	**0**	0	**0**
200 smpls	3 chg	306	**276** (90.2%)	252	**210** (83.3%)	236	**208** (88.1%)	240	**212** (88.3%)
	4 chg	82	**82** (100%)	58	**55** (94.8%)	55	**53** (93.0%)	64	**64** (100%)
	5 chg	18	**18** (100%)	15	**15** (100%)	10	**10** (100%)	11	**11** (100%)
	6 chg	5	**5** (100%)	1	**1** (100%)	0	**0**	4	**4** (100%)
	7 chg	0	**0**	1	**1** (100%)	0	**0**	0	**0**
1000 smpls	3 chg	748	**611** (81.7%)	707	**573** (81.0%)	692	**564** (81.5%)	689	**554** (80.4%)
	4 chg	366	**335** (91.5%)	322	**293** (91.0%)	291	**260** (89.3%)	285	**262** (91.9%)
	5 chg	133	**128** (96.2%)	109	**103** (94.5%)	98	**90** (91.8%)	78	**76** (97.4%)
	6 chg	37	**36** (97.3%)	35	**34** (97.1%)	25	**24** (96.0%)	16	**16** (100%)
	7 chg	15	**15** (100%)	5	**5** (100%)	5	**4** (80.0%)	2	**2** (100%)
5000 smpls	3 chg	1012	**966** (95.5%)	1016	**970** (95.5%)	1012	**964** (95.3%)	1011	**979** (96.8%)
	4 chg	972	**945** (97.2%)	980	**958** (97.8%)	956	**929** (97.2%)	977	**960** (98.3%)
	5 chg	860	**850** (98.8%)	886	**878** (99.1%)	833	**823** (98.8%)	860	**857** (99.7%)
	6 chg	649	**645** (99.4%)	683	**681** (99.7%)	606	**606** (100%)	626	**625** (99.8%)
	7 chg	397	**397** (100%)	427	**426** (99.7%)	360	**360** (100%)	362	**362** (100%)
10000 smpls	3 chg	1025	**994** (97.0%)	1030	**999** (97.0%)	1011	**995** (98.4%)	1016	**994** (97.8%)
	4 chg	1000	**994** (99.4%)	1001	**998** (99.7%)	999	**993** (99.4%)	1000	**993** (99.3%)
	5 chg	987	**983** (99.6%)	983	**981** (99.8%)	977	**975** (99.8%)	974	**970** (99.6%)
	6 chg	930	**927** (99.7%)	918	**917** (99.9%)	898	**897** (99.9%)	884	**882** (99.8%)
	7 chg	783	**782** (99.9%)	766	**766** (100%)	716	**716** (100%)	723	**723** (100%)

Table 6.11: Detected aligned changes for a trace whose H parameter changes from 0.7 to 0.8, analyzed with the DWT-ICSS. The *true* changes are defined as those around ± resolution/2 positions around the true change position.

6.7.4 Empirical assessment of the power of DWT-SIC when the Hurst parameter changes

For the DWT-SIC case, we have repeated the experiment with 1000 realizations of the Hurst parameter change from $H = 0.5$ to $H = 0.9$. The results are shown in Figure 6.32 and Table 6.12 for `db1` at the 99% significance level, and in Figure 6.33 and Table 6.13 for `db3`, also at the 99% significance level. Finally, the results obtained with the clustering and alignment algorithm are shown in Table 6.14.

The main conclusions extracted from the data are:

- The total number of detected changes is, at the lower and middle scales (under $j = 12$), much closer to the ideal value (1000) than what was found for DWT-ICSS even in its best case (99% significance value). Therefore, it seems that DWT-SIC is less prone to false points, for both cases of the mother wavelet.

Scale	True change	Changes	"Good" changes	Raw mean	Corr. mean
1	65537 (65210,65864)	1005	**1002**(99.7%)	65626.7	65534.2
2	32769 (32606,32932)	1001	**998** (99.7%)	32764.0	32764.1
3	16385 (16304,16466)	938	**226** (24.1%)	16406.6	16382.8
4	8193 (8153,8233)	1001	**747** (74.6%)	8199.3	8194.9
5	4097 (4077,4117)	1001	**942** (94.1%)	4095.2	4098.1
6	2049 (2039,2059)	1001	**953** (95.2%)	2049.0	2049.0
7	1025 (1020,1030)	1001	**958** (95.7%)	1024.7	1024.6
8	513 (511,515)	1001	**894** (89.3%)	512.2	512.4
9	257 (256,258)	1002	**880** (87.8%)	255.9	256.4
10	129 (129)	1001	**161** (16.1%)	128.3	129.0
11	65 (65)	1000	**145** (14.5%)	64.2	65.0
12	33 (33)	1013	**138** (13.6%)	32.2	33.0
13	17 (17)	977	**152** (15.8%)	16.3	17.0
14	9 (9)	425	**59** (13.9%)	8.3	9.0
15	5 (5)	44	**4** (11.4%)	3.8	5.0
16	3 (3)	1577	**510** (32.3%)	2.2	3.0
17	2 (2)	0	**0**		

Table 6.12: Detected change points at each scale for a trace whose H parameter changes from 0.5 to 0.9 in the middle of the series, analyzed with the DWT-SIC (`db1`) at the 99% significance level.

Scale	True change	Changes	"Good" changes	Raw mean	Corr. mean
1	65537 (65210,65864)	1004	**1001**(99.7%)	65534.1	65535.1
2	32769 (32606,32932)	1006	**999** (99.3%)	32897.9	32767.4
3	16385 (16304,16466)	928	**214** (23.1%)	16379.0	16385.2
4	8193 (8153,8233)	1004	**789** (78.6%)	8200.1	8196.6
5	4097 (4077,4117)	1005	**938** (93.3%)	4086.7	4099.2
6	2049 (2039,2059)	1002	**951** (94.9%)	2049.8	2050.6
7	1025 (1020,1030)	1008	**943** (93.3%)	1019.4	1026.2
8	513 (511,515)	1043	**843** (80.8%)	494.9	513.6
9	257 (256,258)	1166	**753** (64.6%)	221.6	257.4
10	129 (129)	1502	**403** (26.8%)	88.6	129.0
11	65 (65)	1807	**433** (24.0%)	37.7	65.0
12	33 (33)	2020	**397** (19.7%)	18.7	33.0
13	17 (17)	1069	**268** (25.1%)	14.4	17.0
14	9 (9)	263	**66** (25.1%)	11.2	9.0
15	5 (5)	62	**5** (8.1%)	6.9	5.0
16	3 (3)	43	**6** (14.0%)	3.95	3.0
17	2 (2)	3159	**335** (14.0%)	2.9	2.0

Table 6.13: Detected change points at each scale for a trace whose H parameter changes from 0.5 to 0.9 in the middle of the series, analyzed with the DWT-SIC (`db3`) at the 99% significance level.

		95% sign. level				99% sign. level			
		db1		db3		db1		db3	
resol.	quorum	Total	*True*	Total	*True*	Total	*True*	Total	*True*
5 smpls	3 chg	994	**992** (99.8%)	714	**707** (99.0%)	982	**981** (99.9%)	666	**663** (99.5%)
	4 chg	955	**955** (100%)	509	**509** (100%)	936	**936** (100%)	428	**428** (100%)
	5 chg	848	**848** (100%)	282	**282** (100%)	795	**795** (100%)	199	**199** (100%)
	6 chg	638	**638** (100%)	129	**129** (100%)	544	**544** (100%)	91	**91** (100%)
	7 chg	415	**415** (100%)	37	**37** (100%)	303	**303** (100%)	28	**28** (100%)
20 smpls	3 chg	999	**997** (99.8%)	845	**835** (98.8%)	993	**993** (100%)	805	**800** (99.4%)
	4 chg	979	**979** (100%)	658	**657** (99.8%)	966	**966** (100%)	602	**602** (100%)
	5 chg	915	**915** (100%)	446	**446** (100%)	877	**877** (100%)	369	**369** (100%)
	6 chg	765	**765** (100%)	240	**240** (100%)	685	**685** (100%)	182	**182** (100%)
	7 chg	553	**553** (100%)	110	**110** (100%)	446	**446** (100%)	68	**68** (100%)
200 smpls	3 chg	1003	**1000** (99.7%)	1008	**972** (96.4%)	1000	**1000** (100%)	997	**959** (96.1%)
	4 chg	999	**998** (99.9%)	938	**928** (98.9%)	998	**998** (99.5%)	945	**928** (98.2%)
	5 chg	998	**998** (100%)	848	**845** (99.6%)	991	**991** (99.9%)	814	**812** (99.8%)
	6 chg	973	**973** (100%)	667	**666** (99.9%)	961	**961** (100%)	626	**626** (100%)
	7 chg	898	**898** (100%)	462	**461** (99.8%)	849	**849** (100%)	406	**406** (100%)

Table 6.14: Detected aligned changes for a trace whose H parameter changes from 0.5 to 0.9, analyzed with the DWT-SIC. The *true* changes are defined as those around ± resolution/2 positions around the true change position.

- The number of "good" change points is also slightly better than in the DWT-ICSS case. Furthermore, DWT-SIC keeps a very good quantity until a much more higher scale (886 for $j = 9$ with `db1`) than DWT-ICSS (780 for $j = 9$ with `db1`).
- For scales above $j = 9$, the results fall abruptly into a poor performance region, similar to what was found for the DWT-ICSS method.
- Regarding the wavelet order, `db3` exhibits a similar behavior to that seen in the DWT-ICSS case. We obtain similar results to what was found for DWT-SIC and `db1` at the lower scales, while for $10 \leq j \leq 13$, a lot of false points at the beginning of the trace makes the total number of detected changes increase spectacularly, though few of them are true changes.
- Regarding the alignment detection procedure, the results for SIC are better than those found with ICSS, for all the combinations of parameters.
  - 85% of valid changes at the 99% significance level, with `db1` and 4 aligned points, is attained even at the resolution of samples. It is, therefore, much more precise than the DWT-ICSS method. Even with `db3` one detects more changes for the same combination of parameters.
  - For the $H = 0.7 \rightarrow 0.8$ case the results shown in Table 6.15 are somewhat contradictory, showing a worse performance of DWT-SIC, though the 85% criterion is attained at more or less the same resolution as the DWT-ICSS case (around 3500-4000).

We conclude that the DWT-SIC performs better than the DWT-ICSS method for the more abrupt H changes, and vice versa. We hypothesize that the cause for this behavior is the implicit difference in the significance values of the SIC and ICSS methods. We saw in the previous examples

		95% sign. level				99% sign. level			
		db1		db3		db1		db3	
resol.	quorum	Total	*True*	Total	*True*	Total	*True*	Total	*True*
5 smpls	3 chg	29	**26** (89.7%)	30	**30** (100%)	14	**11** (78.6%)	12	**12** (100%)
	4 chg	5	**5** (100%)	3	**3** (100%)	0	**0**	1	**1** (100%)
	5 chg	1	**1** (100%)	0	**0**	0	**0**	0	**0**
	6 chg	0	**0**	0	**0**	0	**0**	0	**0**
	7 chg	0	**0**	0	**0**	0	**0**	0	**0**
20 smpls	3 chg	59	**53** (89.8%)	55	**47** (85.5%)	24	**20** (83.3%)	29	**27** (93.1%)
	4 chg	8	**7** (87.5%)	9	**8** (88.9%)	5	**5** (100%)	2	**2** (100%)
	5 chg	1	**1** (100%)	1	**1** (100%)	0	**0**	1	**1** (100%)
	6 chg	0	**0**	0	**0**	0	**0**	0	**0**
	7 chg	0	**0**	0	**0**	0	**0**	0	**0**
200 smpls	3 chg	275	**224** (81.5%)	264	**217** (82.2%)	187	**157** (84.0%)	195	**173** (88.7%)
	4 chg	73	**67** (91.8%)	73	**67** (91.8%)	34	**30** (88.2%)	41	**38** (92.7%)
	5 chg	16	**14** (87.5%)	11	**9** (81.8%)	8	**7** (87.5%)	5	**5** (100%)
	6 chg	4	**3** (75.0%)	2	**2** (100%)	0	**0**	1	**1** (100%)
	7 chg	0	**0**	1	**1** (100%)	0	**0**	1	**1** (100%)
1000 smpls	3 chg	741	**622** (83.9%)	751	**565** (75.2%)	616	**508** (82.5%)	633	**522** (82.5%)
	4 chg	336	**307** (91.4%)	322	**276** (85.7%)	243	**219** (90.1%)	253	**222** (87.8%)
	5 chg	116	**114** (98.3%)	101	**96** (89.7%)	62	**58** (93.5%)	72	**68** (94.4%)
	6 chg	26	**26** (100%)	34	**32** (94.1%)	11	**10** (90.9%)	15	**14** (93.3%)
	7 chg	5	**5** (100%)	4	**4** (100%)	2	**2** (100.0%)	1	**1** (100%)
5000 smpls	3 chg	1008	**947** (93.9%)	1094	**965** (88.2%)	999	**961** (96.2%)	1008	**971** (96.3%)
	4 chg	958	**926** (96.7%)	1005	**947** (94.2%)	936	**919** (98.2%)	952	**930** (97.7%)
	5 chg	826	**819** (99.2%)	853	**835** (97.9%)	767	**763** (99.5%)	780	**776** (99.5%)
	6 chg	603	**601** (99.7%)	621	**6815** (99.0%)	468	**467** (99.8%)	500	**499** (99.8%)
	7 chg	339	**339** (100%)	349	**348** (99.7%)	230	**230** (100%)	213	**213** (100%)
10000 smpls	3 chg	1017	**986** (97.0%)	1117	**992** (88.8%)	1000	**990** (99.0%)	1014	**992** (97.8%)
	4 chg	998	**983** (98.5%)	1038	**990** (95.4%)	997	**987** (99.0%)	998	**986** (98.8%)
	5 chg	962	**951** (99.6%)	980	**959** (97.9%)	927	**925** (99.8%)	925	**921** (99.6%)
	6 chg	873	**870** (99.7%)	846	**836** (98.8%)	748	**747** (99.9%)	729	**728** (99.9%)
	7 chg	679	**679** (100%)	628	**624** (99.4%)	469	**468** (99.8%)	461	**460** (99.8%)

Table 6.15: Detected aligned changes for a trace whose H parameter changes from 0.7 to 0.8, analyzed with the DWT-SIC. The *true* changes are defined as those around ± resolution/2 positions around the true change position.

that for the same significance level, the ICSS returns more points (either "good" or spurious), while the SIC is more strict. This makes the alignment procedure perform better with SIC when the Hurst parameter changes are more clear (abrupt), while in the case of the smoother H changes the ICSS, though it provides *not-so-valid* points, its number is higher and makes it easier for the alignment procedure to find positive results.

6.7.5 Traces with multiple Hurst parameter changes

A more elaborate example considers a 131072-samples trace with four segments of FGN, as plotted in the right side of Figure 5.16. The variance is constant for the entire trace ($\sigma^2 = 1$). The change points are located at $t_1 = 16385$, $t_2 = 32769$, and $t_3 = 98305$, dividing the segments with $H = 0.9$, $H = 0.5$, $H = 0.7$, and $H = 0.8$. We begin analyzing the trace with the DWT-ICSS algorithm. The correspondent DWT detail coefficients (using the `db1` wavelet family) and CSS statistics at scales $j = 1, \ldots, 6$ are plotted in Figure 6.34 and Figure 6.35, respectively. An inspection of Figure 6.35

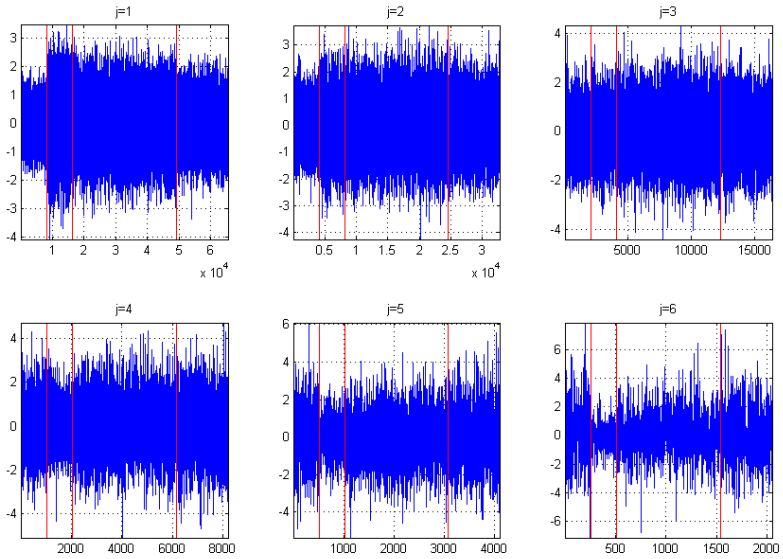

Figure 6.34: DWT details of the trace with four segments with different Hurst parameter, for $j = 1, \ldots, 6$, using `db1` wavelet. The change points are located at $t_1 = 16384$, $t_2 = 32768$, $t_3 = 98304$, and Hurst parameter values are $H = 0.9 \rightarrow 0.5 \rightarrow 0.7 \rightarrow 0.8$. The variance is constant.

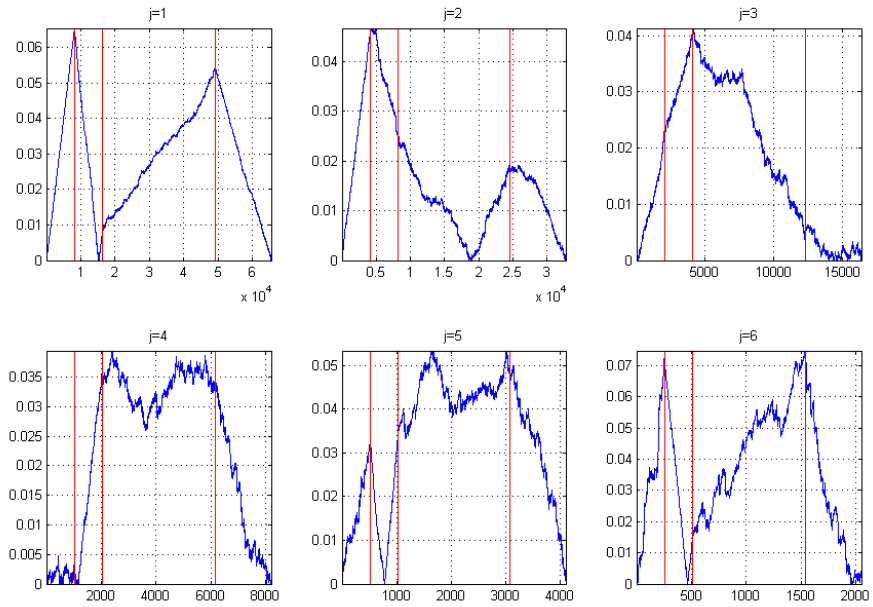

Figure 6.35: Absolute value of the CSS statistic, applied to the DWT details coefficients of the trace with four segments with different Hurst parameter, for $j = 1, \ldots, 6$, using `db1` wavelet. The change points are located at $t_1 = 16384$, $t_2 = 32768$, $t_3 = 98304$, and Hurst parameter values are $H = 0.9 \rightarrow 0.5 \rightarrow 0.7 \rightarrow 0.8$. The variance is constant.

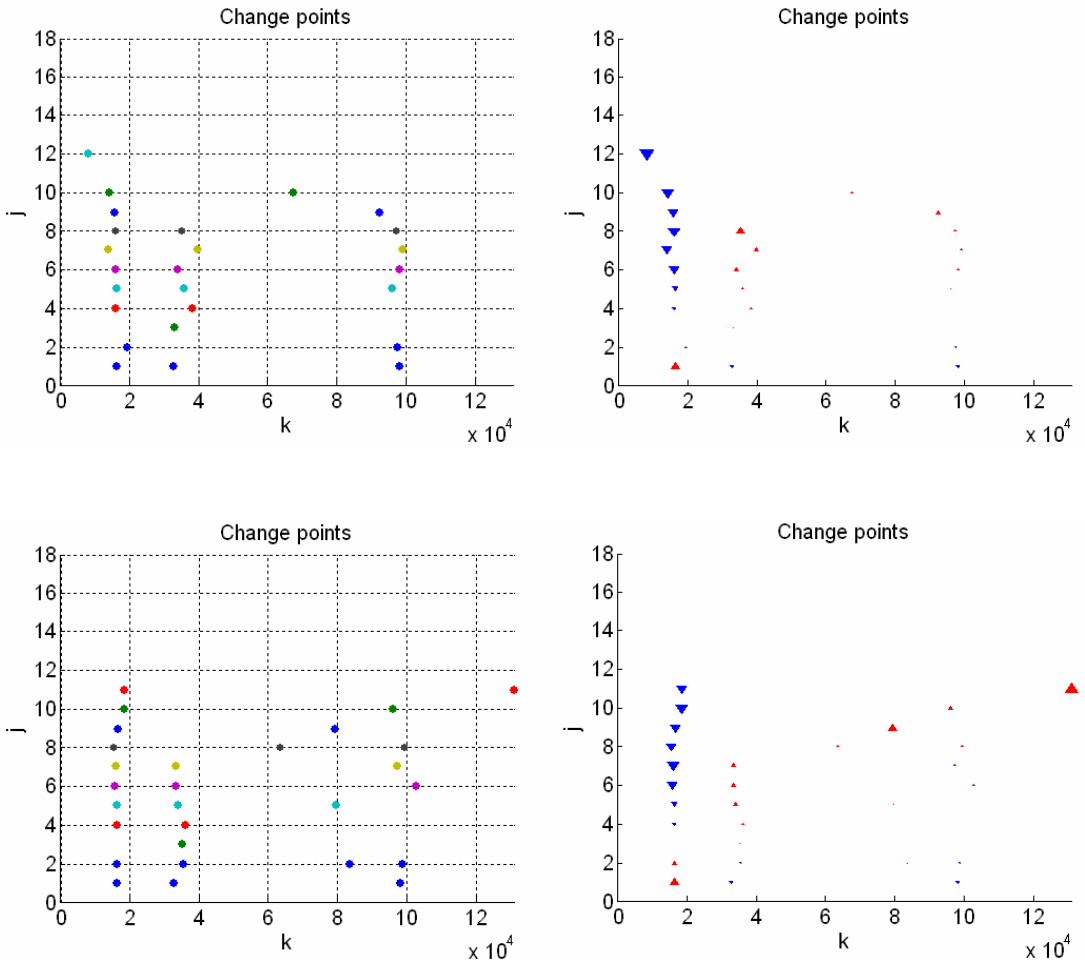

Figure 6.36: Detected change points when the DWT-ICSS is applied to the trace with four segments with different Hurst parameter, at the 99% significance level. *Top left*: the position of the change points (`db1`). *Top right*: size and sign of the detected variance change(`db1`). *Bottom left*: the position of the change points (`db3`). *Bottom right*: size and sign of the detected variance change(`db3`)

suggests the following conclusions:

- the first change point is detected as a clear peak at scales $j = 1$, $j = 2$, $j = 5$, $j = 6$ and as a local maximum at scale $j = 4$;
- the second change point is detected as a change of slope at scales $j = 1$, $j = 2$, $j = 5$, and $j = 6$, as a clear peak at scale $j = 3$ and as a local maximum at scales $j = 4$;
- the third change point is detected as a peak at scales $j = 1$, $j = 2$ and $j = 6$ and as a local maximum at scales $j = 4$, $j = 5$.

We can see the masking effect of the CSS statistic, when the time series have more than one change point. When the complete ICSS statistic is applied to the trace, we get the results shown

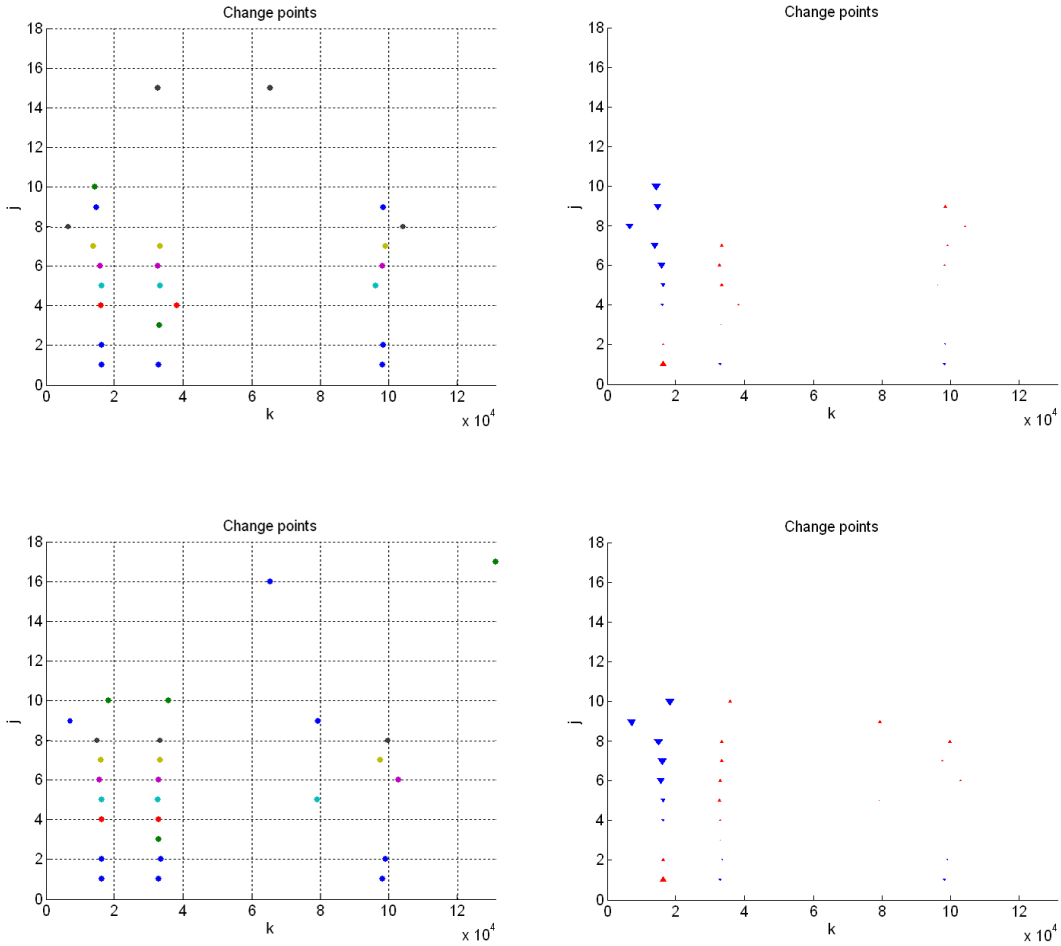

Figure 6.37: Detected change points when the DWT-SIC is applied to the trace with four segments with different Hurst parameter, at the 99% significance level. *Top left*: the position of the change points (`db1`). *Top right*: size and sign of the detected variance change(`db1`). *Bottom left*: the position of the change points (`db3`). *Bottom right*: size and sign of the detected variance change(`db3`)

in Figure 6.36. The first change point (the *abrupt* one, from $H = 0.5$ to $H = 0.9$) is correctly identified for scales up to $j = 10$, with the exception of the blind zone at $j = 3$. The second and third changes are not as clearly detected, due to their less abrupt nature. However, the mean of the detected change points is still quite close to the true change point. For the `db3`-based analysis, the results are similar, with a slight improvement in the detection of the two first change points (in terms of quantity of scales and dispersion), and more dispersion for the third (and more smooth) change.

For the DWT-SIC case, we will only show the diagram of the detected change points at the 99% significance level, for `db1` and `db3` (Figure 6.37). The results are similar, with the exception of the isolated false points at the highest scales (15, 16 and 17). For the `db3` case a couple of false points appear somewhat aligned by chance around $k = 79000$. The size analysis shows they are of little

importance, and anyway the alignment procedure will ignore them for quorums above 2.

Table 6.16 shows the results of the alignment procedure. Compared with those obtained in Sections 6.7.1 and 6.7.2 for single changes, the performance of the algorithms in front of multiple changes is worse. They need higher resolution or less strict alignment requirements (lower quorum) in order to detect the changes. The cause for this behavior is the masking effect of the variance change statistics and the proximity of the changes. That is why we studied the performance of the algorithms in terms of the separation between changes of H, as shown in the next Section.

	quorum (votes)	q=3			q=5		
	resolution (samples)	$r = 200$	$r = 1000$	$r = 5000$	$r = 200$	$r = 1000$	$r = 5000$
ICSS 99%, `db1`	alignments	16187	16201	15687 35748		16201	15687
			97840	97776			97776
ICSS 99%, `db3`	alignments	16397	16249 33126	16484 35157 98295		16249	16484
SIC 99%, `db1`	alignments	16422 32788	16288 33007 98543	15572 33007 98041		16288 33007	15572 33007 98041
SIC 99%, `db3`	alignments	16395 33031	16248 33008	16293 33008 98295		16248 33008	16293 33008

Table 6.16: Results of the alignment procedure applied to the DWT-ICSS and DWT-SIC analysis of an FGN trace with four segments with different values of H.

6.7.6 Empirical assessment of the power of the DWT methods in presence of multiple H changes

We studied the performance of the DWT-based methods for the case of two H changes ($H = 0.5 \rightarrow 0.9 \rightarrow 0.5$) in terms of the separation of the changes. Tables 6.17 and 6.18 show the results for for the DWT-ICSS and DWT-SIC with `db1`, while the DWT-SIC `db3` results can be found in Table 6.19. The values with the not apply (n/a) legend correspond to resolutions under or equal to the value of the separation parameter; in these cases, the same change point would appear as a true change point in the neighborhood of both change points, and therefore the results are incorrect.

The main conclusions extracted from the data are:

- The performance of the algorithms is very sensitive to the separation between the change points. For the shorter separations almost no change is detected by the methods. Separations under 5000 samples are difficult to detect. This can be explained by the fact that each change is detected across 8 to 10 scales, approximately, as was shown in the previous sections. The time dispersion created by the downsampling procedure of the DWT at the higher scales introduces an inaccuracy of 2^8 to 2^{10} samples in the position of the change alignments. This causes the minimum separation to be over 1000 samples, at least.

		separation (samples)									
		1000		5000		10000		50000		100000	
resolution	quorum	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch
20 samples	3 changes	0	0	0	0	11	19	8	16	10	7
	4 changes	0	0	0	0	3	0	0	1	0	0
	5 changes	0	0	0	0	0	0	0	0	0	0
200 samples	3 changes	0	0	163	0	84	694	870	748	790	567
	4 changes	0	0	25	0	23	441	705	648	505	287
	5 changes	0	0	1	0	0	186	440	422	196	77
1000 samples	3 changes	n/a	n/a	569	284	426	860	995	721	992	984
	4 changes	n/a	n/a	428	210	312	819	995	716	991	979
	5 changes	n/a	n/a	174	110	163	625	985	657	926	913
5000 samples	3 changes	n/a	n/a	n/a	n/a	1101	793	1000	997	1000	479
	4 changes	n/a	n/a	n/a	n/a	1092	793	1000	997	1000	479
	5 changes	n/a	n/a	n/a	n/a	1077	792	1000	997	1000	479
10000 samples	3 changes	n/a	n/a	n/a	n/a	n/a	n/a	1002	1000	1000	1000
	4 changes	n/a	n/a	n/a	n/a	n/a	n/a	1000	1000	1000	1000
	5 changes	n/a	n/a	n/a	n/a	n/a	n/a	1000	1000	1000	1000

Table 6.17: Results for the DWT-ICSS method (99% significance level, `db1`) for a trace with three regions with different H parameter ($H = 0.5 \rightarrow 0.9 \rightarrow 0.5$), as a function of the length of the central region (*separation*). The changes are defined as those around ± resolution/2 positions around the true change.

		separation (samples)									
		1000		5000		10000		50000		100000	
resolution	quorum	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch
20 samples	3 changes	0	0	0	0	70	73	3	21	8	5
	4 changes	0	0	0	0	9	6	0	2	0	0
	5 changes	0	0	0	0	1	0	0	0	0	0
200 samples	3 changes	0	0	114	2	279	825	896	819	679	630
	4 changes	0	0	15	0	77	596	778	744	584	445
	5 changes	0	0	3	0	7	282	511	508	297	142
1000 samples	3 changes	n/a	n/a	397	299	709	794	980	580	996	942
	4 changes	n/a	n/a	176	175	624	743	980	579	996	942
	5 changes	n/a	n/a	38	57	405	554	975	544	993	908
5000 samples	3 changes	n/a	n/a	n/a	n/a	1012	712	1000	1000	1000	941
	4 changes	n/a	n/a	n/a	n/a	910	709	1000	1000	1000	941
	5 changes	n/a	n/a	n/a	n/a	746	708	1000	1000	1000	999
10000 samples	3 changes	n/a	n/a	n/a	n/a	n/a	n/a	1000	1001	1000	999
	4 changes	n/a	n/a	n/a	n/a	n/a	n/a	1000	1000	1000	999
	5 changes	n/a	n/a	n/a	n/a	n/a	n/a	1000	1000	1000	999

Table 6.18: Results for the DWT-SIC method (99% significance level, `db1`) for a trace with three regions with different H parameter ($H = 0.5 \rightarrow 0.9 \rightarrow 0.5$), as a function of the length of the central region (*separation*). The changes are defined as those around ± resolution/2 positions around the true change.

		separation (samples)									
		1000		5000		10000		50000		100000	
resolution	quorum	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch	1^{st} ch	2^{nd} ch
20 samples	3 changes	0	0	1	0	0	0	82	43	46	57
	4 changes	0	0	0	0	0	0	5	2	0	1
	5 changes	0	0	0	0	0	0	0	0	0	0
200 samples	3 changes	0	0	109	6	489	747	908	703	807	615
	4 changes	0	0	17	0	197	457	783	472	558	457
	5 changes	0	0	2	0	36	197	511	187	267	200
1000 samples	3 changes	n/a	n/a	383	285	725	830	949	734	998	996
	4 changes	n/a	n/a	181	172	677	763	949	729	998	996
	5 changes	n/a	n/a	52	54	432	508	942	655	997	990
5000 samples	3 changes	n/a	n/a	n/a	n/a	1070	453	1000	999	1000	743
	4 changes	n/a	n/a	n/a	n/a	1010	453	1000	999	1000	743
	5 changes	n/a	n/a	n/a	n/a	896	453	1000	999	1000	743
10000 samples	3 changes	n/a	n/a	n/a	n/a	n/a	n/a	1000	1001	1000	1000
	4 changes	n/a	n/a	n/a	n/a	n/a	n/a	1000	1000	1000	1000
	5 changes	n/a	n/a	n/a	n/a	n/a	n/a	1000	1000	1000	1000

Table 6.19: Results for the DWT-SIC method (99% significance level, `db3`) for a trace with three regions with different H parameter ($H = 0.5 \rightarrow 0.9 \rightarrow 0.5$), as a function of the length of the central region (*separation*). The changes are defined as those around ± resolution/2 positions around the true change.

- As expected from previous experiments, the increases in quorum lower the change detection ratios, while the higher the resolution, the more changes are detected.

- For the lower detection ratios, the results for one of the changes seem to be better than or equal to those found for the other change. This seems to be some kind of *masking* effect which mitigates at the higher values of resolution and separation (i.e., the values for which the number of detected changes approaches the ideal value of 1000).

- The SIC-based method seems to perform slightly better than DWT-ICSS, if we consider the mean number of detected changes (including both changes) as the merit figure.

- Regarding the influence of the wavelet order in the performance of the SIC method, few differences are found in the tables.

- We conclude that our methods perform acceptably (in the typical case of 8-10 scales with aligned changes) for separations above 5000 samples, resolutions in the 1000-5000 samples range, and quorums of 3-4 changes. .

The previous conclusions apply to pure H changes. In real traces we expect to find simultaneous changes in the Hurst parameter and the variance is present. That is why we studied this situation in the following section.

6.8 Analysis of the effect of a simultaneous change of variance and Hurst parameter

The trace shown in Figure 5.17 (left), composed of four segments with $H = 0.5, \sigma^2 = 1$, $H = 0.9, \sigma^2 = 2$, $H = 0.9, \sigma^2 = 1$, $H = 0.5, \sigma^2 = 2$, will be analyzed with the DWT-ICSS and the DWT-SIC in this section. Note that the trace includes two simultaneous changes of variance and Hurst parameter at $k = 32769$ and $k = 98305$ and a variance change at $k = 65537$. The results are shown in Figure 6.38 for the DWT-ICSS and in Figure 6.39 for the DWT-SIC case. Table 6.20 shows the results after the alignment procedure.

Why do we try this experiment? The difference from the Hurst change cases already commented is the opposite or reinforcing effect of both kind of changes (global variance or H) in the relative changes of the variance at every scale. Comparing the *value and size* diagrams shown on the right of the figures, we arrive to the following conclusions:

- The variance changes at the highest scales of the first change ($k = 32769$) are reinforced due to the simultaneous increase of H and σ^2, while the decrease of variance at the lower scales due to the H change is compensated by the σ^2 increase, almost masking the variance changes at scales 1 and 2.

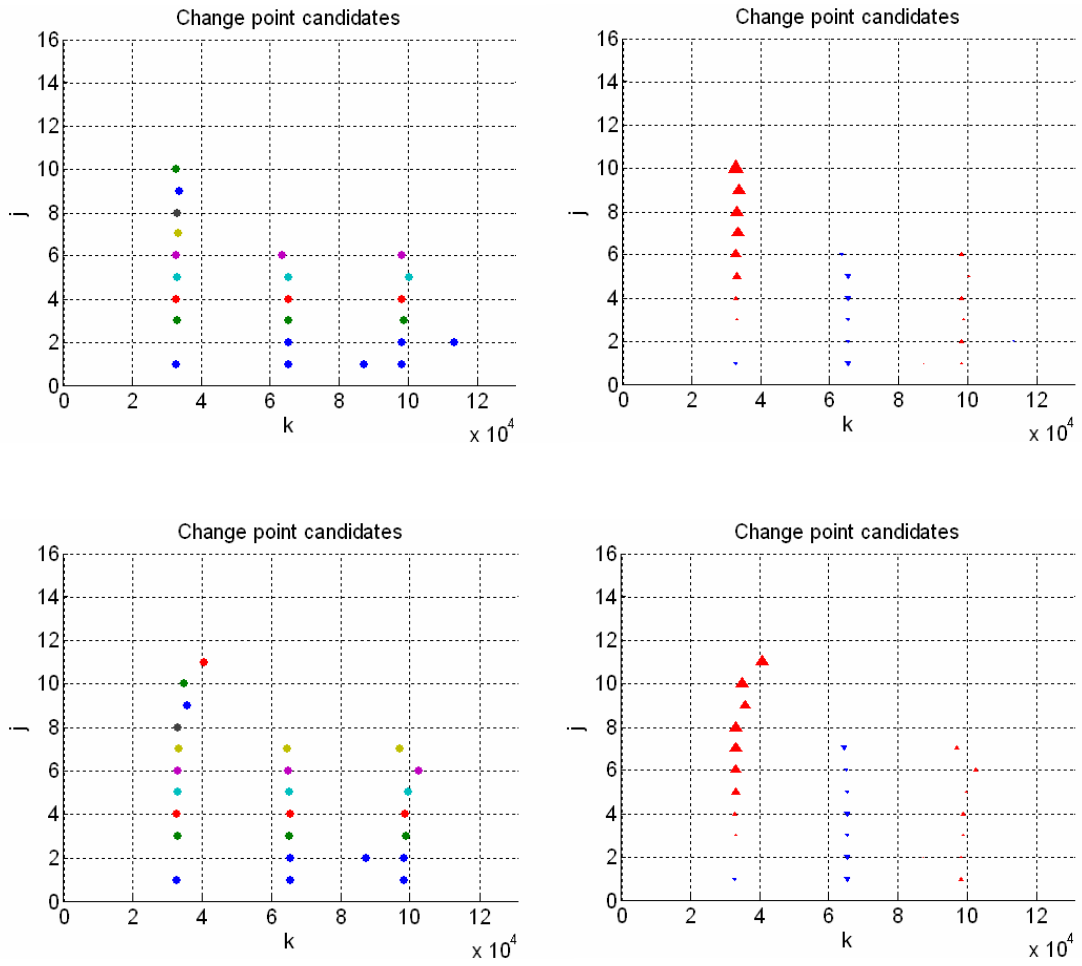

Figure 6.38: Detected change points when the DWT-ICSS is applied to the trace with four segments with different Hurst parameter, at the 99% significance level. *Top left*: the position of the change points (`db1`). *Top right*: size and sign of the detected variance change(`db1`). *Bottom left*: the position of the change points (`db3`). *Bottom right*: size and sign of the detected variance change(`db3`)

- The σ^2 change at $k = 65537$ is clearly detected as a decrease in the variance at 6-7 scales. The changes are limited to fewer scales and have a lower size than those found in Section 6.6 for a more abrupt variance with a ratio of 4, which were detected across 11-12 scales.

- For the second change, the situation is exactly the opposite of the first change. For the higher scales, the variance decrease due to the H change is compensated by the σ^2 increase, with an interesting effect: the changes of scales 7-11 (or 8-12) disappear due to the smoothing effect of the DWT, combined with the little size of the changes. Actually, no variance decrease is detected in the $k = 98305$ position, though the higher scales should display a decrease. Therefore the H change is masked and appears as a variance increase with a low variance change ratio.

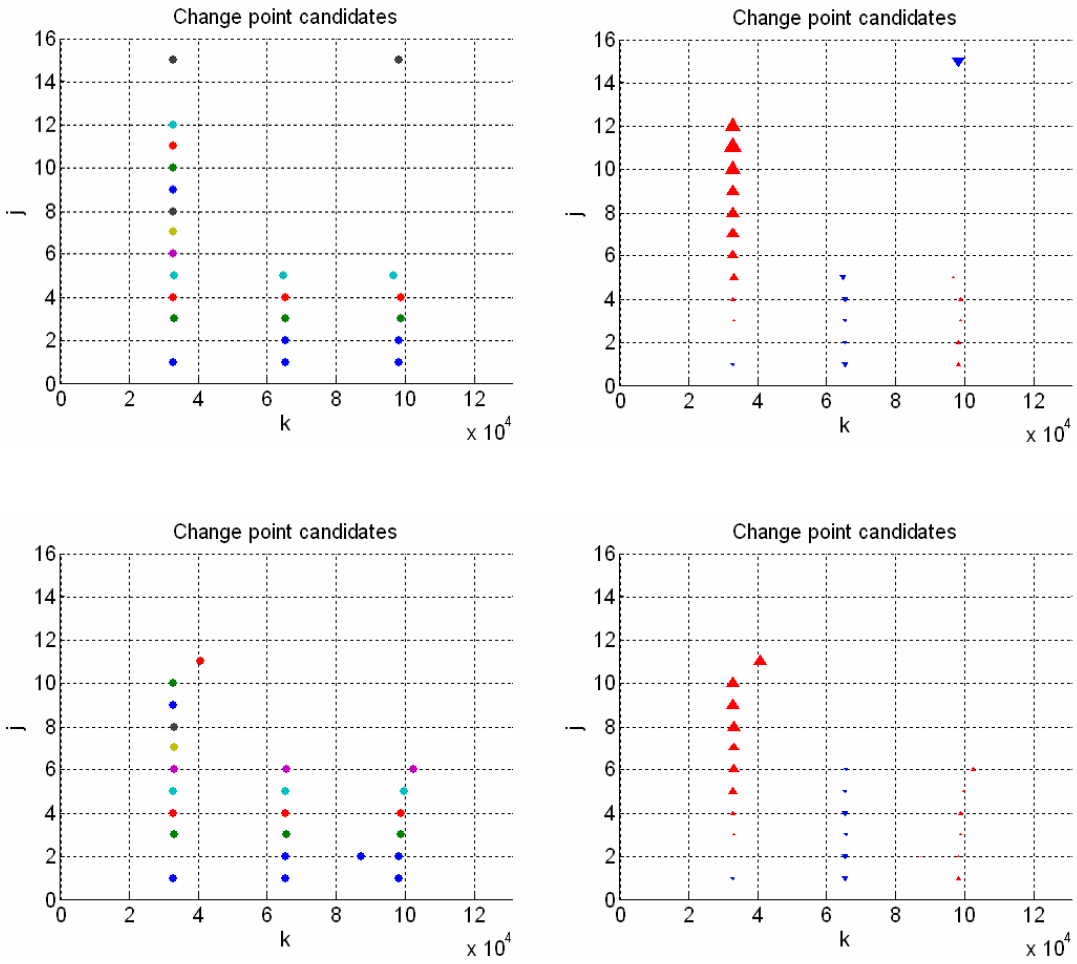

Figure 6.39: Detected change points when the DWT-SIC is applied to the trace with four segments with different Hurst parameter, at the 99% significance level. *Top left*: the position of the change points (`db1`). *Top right*: size and sign of the detected variance change(`db1`). *Bottom left*: the position of the change points (`db3`). *Bottom right*: size and sign of the detected variance change(`db3`)

- When the alignment detection method is applied to the change diagrams, the masking effect is confirmed: the third change is the most difficult to find, and needs a higher value of resolution in order to be detected. Regarding the comparison between methods, the results are similar, except for a slightly better behavior for DWT-SIC in the detection of the changes of H.

Though we have not tried the opposite cases (H increase + variance decrease and H decrease + variance decrease), we expect them to behave as the dual cases of those presented here. Therefore, we conclude that the simultaneous change of variance and H in a trace can suffer either a masking or reinforcement effect, depending on the increase/decrease combination, making it easier or more difficult its detection.

	quorum (votes)	q=3			q=5		
	resolution (samples)	$r = 200$	$r = 1000$	$r = 5000$	$r = 200$	$r = 1000$	$r = 5000$
ICSS 99%, `db1`	alignments	32784 65461	32968 65419 98507	32968 65419 98725		32968 65419	32968 65419 98725
ICSS 99%, `db3`	alignments	33031 65501	32949 65356 98615	32949 65356 98291		32949 65356	32949 65356 98554
SIC 99%, `db1`	alignments	32780 65472 98331	32826 65472 98576	32826 65472 98291	32780	32826 98576	32826 98291
SIC 99%, `db3`	alignments	32791 65515	32894 65476 98615	32894 65476 98834	32791 65515	32894 65476	32894 65476 98834

Table 6.20: Results of the alignment procedure applied to the DWT-ICSS and DWT-SIC analysis of an FGN trace with two simultaneous changes of variance and Hurst parameter at $k = 32769$ and $k = 98305$ and a variance change at $k = 65537$.

6.9 Application to real traffic traces

Let us now detect the change points of the `BC-pAug89` and `BC-OctExt` traces (described in Sections 3.2.10 and 5.4) with the DWT-based algorithms.

6.9.1 Statistical distribution of the DWT coefficients

We begin our analysis by studying the distribution of the coefficients, as we did with the synthetic FGN traces in Section 6.3. Figure 6.40 and 6.41 show the histograms and probability plots for the `BC-pAug89` and `BC-OctExt` traces (aggregated at 10 ms and 1 s, respectively) when analyzed with `db3`[7]. For the first trace the distribution is clearly Laplacian at the lowest scale (highest frequency), but it gets more and more Gaussian-like as the scales increases. This effect is caused by the iterative filtering of the DWT scheme. The highest scales have been convolved several times with the wavelet filter, and this seems to *soften* the distribution and make it converge to normal. For the `BC-OctExt` case the effect is not as efficient, since the original trace was much more variable and heavy-tailed than `BC-pAug89`, but the smoothing effect can also be seen in the figure.

	$j = 1$	$j = 5$	$j = 10$		$j = 1$	$j = 4$	$j = 8$
`BC-pAug89`	0.34	1.15	1.51	`BC-OctExt`	0.35	0.38	0.51

Table 6.21: GGD shape parameter of the DWT decomposition of the Bellcore traces.

Table 6.21 presents the estimation of the shape parameter of the distributions shown in the previous figures. The `BC-OctExt` trace at all scales and the first scale of `BC-pAug89` exhibit a *sharper-than-Laplacian* behavior, while the higher scales of the `BC-pAug89` trace fall in the zone between Laplacian and Gaussian. As we will see in the following sections, this will influence the amount of detected change points at each scale.

[7] The results obtained with `db1` are similar and have been omitted.

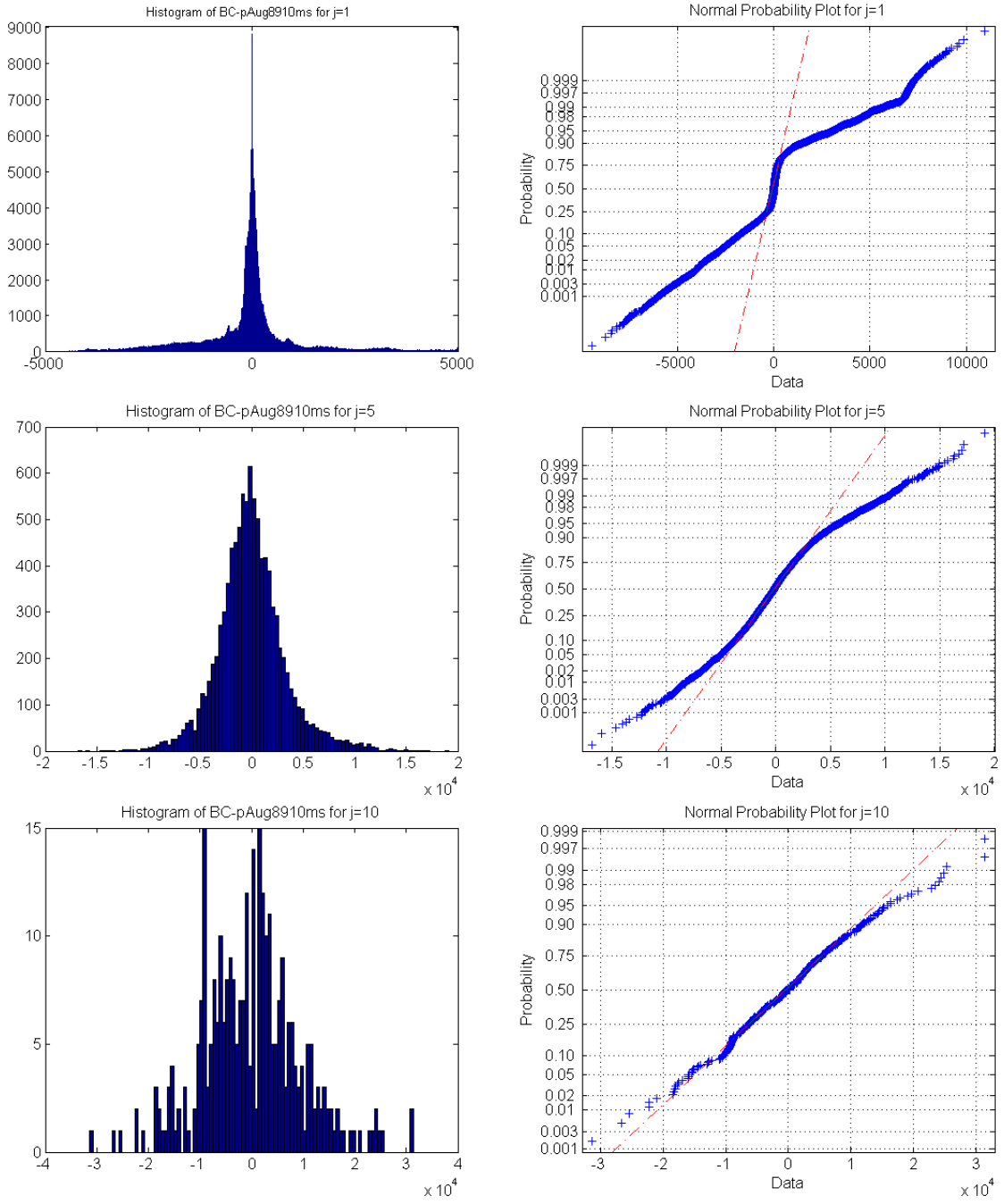

Figure 6.40: Histograms (*left*) and probability plots for graphical normality testing (*right*) of the DWT coefficients of the `BC-pAug89` (aggregated at 10 ms) trace when analyzed with `db3`. From top to bottom, the analysis for the coefficients at scales $j = 1$, $j = 5$ and $j = 10$.

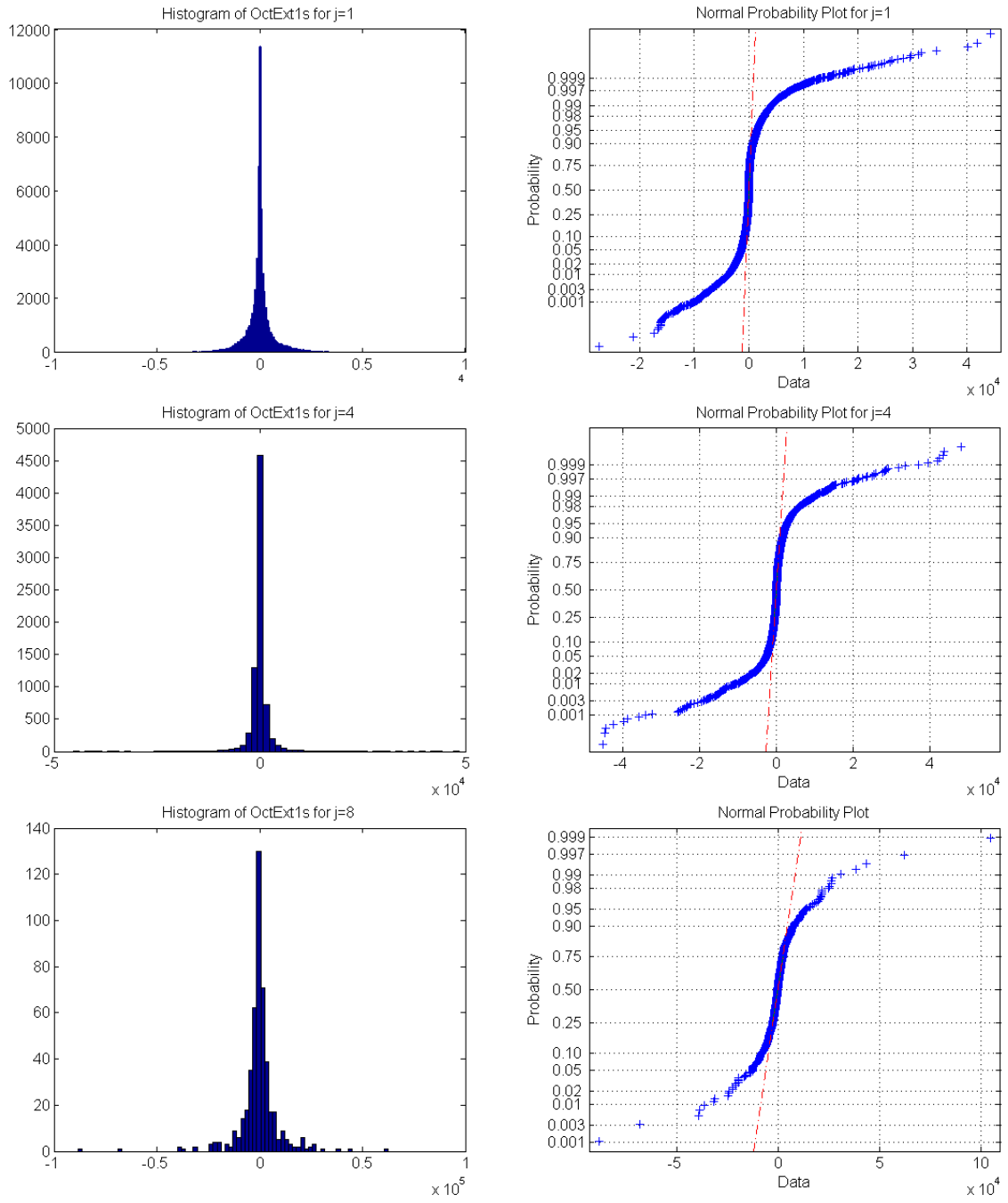

Figure 6.41: Histograms (*left*) and probability plots for graphical normality testing (*right*) of the DWT coefficients of the `BC-OctExt` trace (aggregated at 1 s) when analyzed with `db3`. From top to bottom, the analysis for the coefficients at scales $j = 1$, $j = 4$ and $j = 8$.

6.9.2 DWT-ICSS

`BC-pAug89` trace

The results of the DWT-ICSS algorithm are shown in Figure 6.42 for the case of the trace `BC-pAug89` (aggregated at 10 ms). The density of change points at the lower scales, caused by their *sharper-than-Laplacian* distribution, makes it difficult to obtain conclusions; that is why we will skip the first or second scales in some of the analyses. We will begin by focusing our attention on some segments. We selected two of these segments; the ones that comprise the positions $(75000 - 100000)$ and $(250000 - 275000)$. The segments of the trace and the results of the analysis are shown in Figure 6.43. The main conclusions are:

- Some abrupt changes in the traffic volume (approximate positions 80500, 85500, 259000 and 264500, grey lines) are detected as the alignment of change points across scales. These are examples of a variance change and not only of a mean change, that would pass unawarely.
- Very few of the detected changes belong to scales above 5 or 6, due to the lack of resolution of the DWT at the higher scales.
- No "pure" changes of the scaling parameter are detected (i.e., alignment of changes with the positive/negative pattern seen in the experiments with synthetic traces). This is not a surprise, since real traffic traces are much more complex than synthetic ones. In any case, changes in traffic volume are known to be related to changes in the scaling parameters of traffic; in this sense, our algorithm is still valid.
- There are few differences between the `db1` and `db3` analyses; the clearest alignments are detected in both cases, with maybe one less aligned change for the higher order wavelet. Real traffic traces are prone to trends; that is why we will perform the following study with the `db3` wavelet (as suggested by [VA99]) in order to filter linear and parabolic trends.

When the alignment detection method is applied, we get the results shown in Figure 6.44. We chose the following parameters for the analysis: 99% significance level, `db3` wavelet, resolution = 5000 samples and quorum = 4 changes (as suggested by the experiments described in Section 6.7). The figure shows:

1) The original trace, segmented at the aligned changes.
2) The change point diagram, segmented at the aligned changes.
3) The Hurst parameter estimation in the segments, with its associated confidence 95% intervals (horizontal lines). The lower scale j_1 used for regression has been chosen automatically with the *Chi-squared goodness of fit test* described in [VAT03], while the higher scale j_2 has been chosen as the highest among those considered as *safe* (i.e., with a sufficient number of wavelet coefficients) by the Matlab implementation of the LogScale Diagram [VA].
4) The Hurst parameter estimation in 32 constant length segments (the same analysis presented in Section 5.4). The scales for the linear regression have been chosen with the same methods described in the previous case.

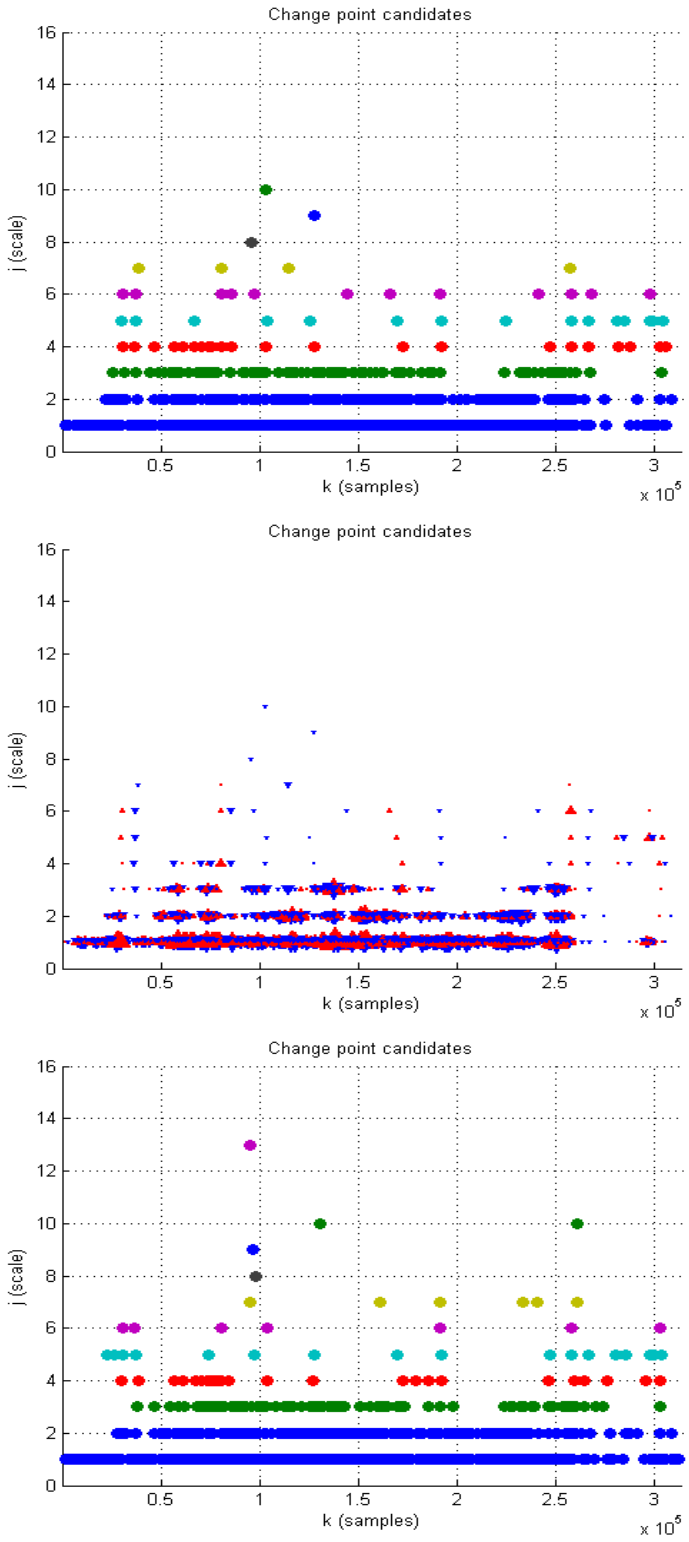

Figure 6.42: Change points at each scale for the DWT-ICSS analysis of the `BC-pAug89` trace (aggregated at 10 ms) at the 99% significance level, with `db1`. *Top*: position of each change point for `db1`. *Middle*: Sign and size of each change point for `db1`. *Bottom*: position of each change point for `db3`.

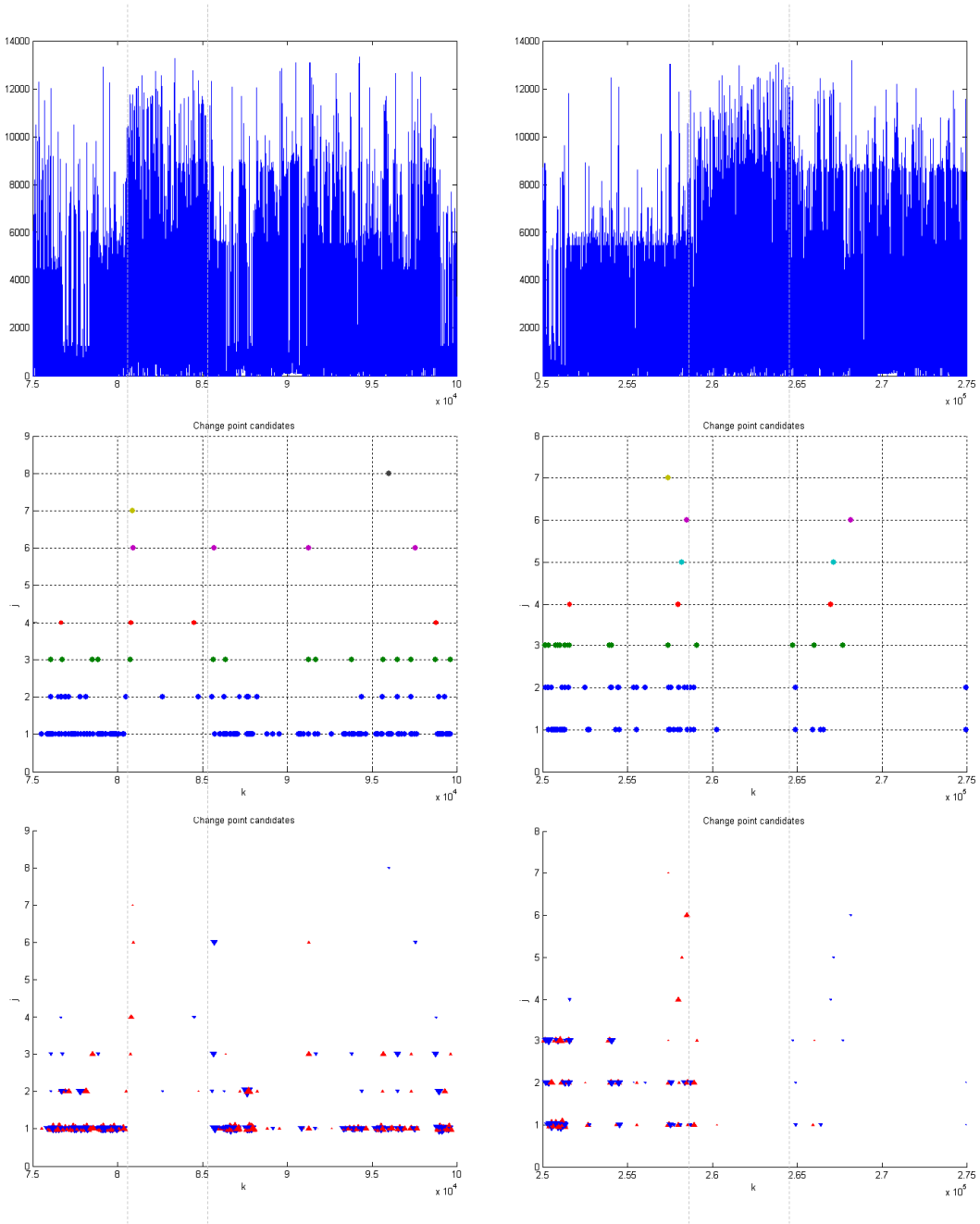

Figure 6.43: Change points at each scale for the DWT-ICSS analysis of the `BC-pAug89` trace, with `db1` wavelet, at the 99% significance level. *Left*: segment $75000 - 100000$. *Right*: segment $250000 - 275000$.

5) The variance of each of the segments, together with the associated 95% confidence intervals (assuming gaussiantity).
6) The mean volume of the original trace, segmented at the aligned changes.

The main conclusions are:

- The segmentation of the change point diagram detects without problems the alignments, though the dispersion at the lower scales causes some of the alignments to be biased from the *ideal* position. The (relatively high) value of the resolution parameter can also contribute to this effect.

- The time evolution of the Hurst parameter is tracked with very good results: the segments almost do not overlap their confidence intervals, implying that there is a real H change at those positions. There are some exceptions such as the segments around $k = 250000$; in this case we are detecting a traffic variance change instead of a Hurst parameter change (see later).

- The goodness of our segmenting algorithm is confirmed by the constant length H analysis, where most of the segments that overlap their confidence intervals correspond to regions where our segmenting algorithm estimates a constant Hurst parameter. Take the example of the first 3 constant length segments (with values around $H = 0.8$), that are considered as a single segment by our algorithm.

- The values of the Hurst parameter ranges from 0.54 to 0.88 for our segmenting method, while the constant length method returns values from 0.45 to 0.86. Although the range of H can seem very high, the trace is actually quite stable and oscillates around the *mean* value of 0.8, in agreement with the *static* analysis presented in Chapter 3 and reported by other authors.

- Though the confidence interval argument does not always hold when comparing the constant length analysis and our segmenting algorithm, recall that the validity of the constant length analysis is questionable and can be taken as a first approximation of the real evolution of the Hurst parameter, because of the arbitrariness of the constant length positions, while our method has strong physical roots (the multiresolution variance changes).

- The evolution of the variance explains some of the change points that do not exhibit an H change, such as the aforementioned change at $k = 250000$. Other examples of segments that show a variance change while keeping H constant (or almost constant) in 6.44 are the segments around $k = 75000$ and $k = 160000$.

- The opposite situation (changes of the Hurst parameter while the variance remains constant) is also detected in some segments, such as the change point located at $k = 100000$ and a couple of changes around $k = 175000$.

- The evolution of the traffic volume (or the variance) is not correlated with the behavior of the Hurst parameter, in contradiction with the results described in [LTWW94] and [BSTW95].

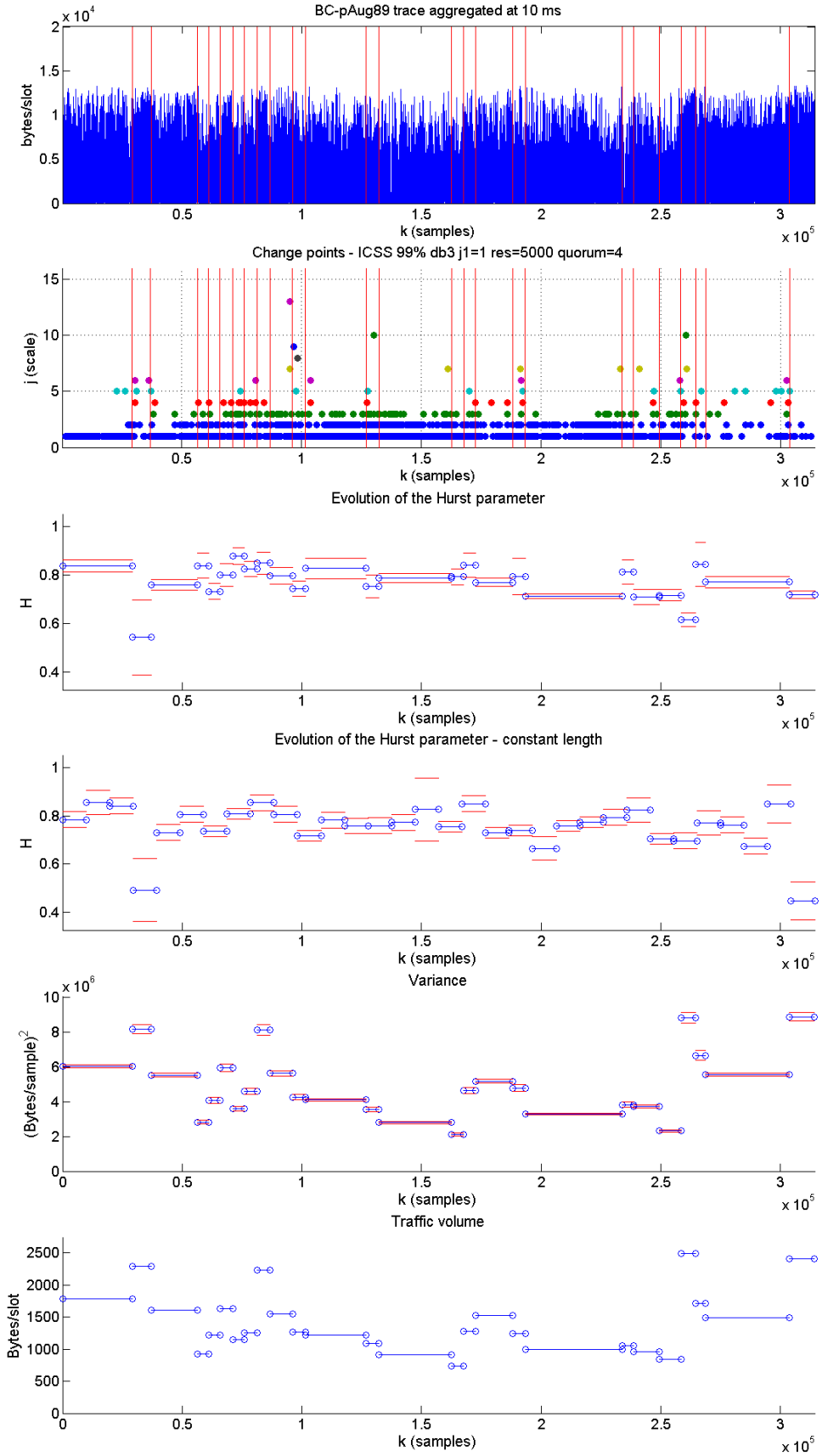

Figure 6.44: Segmentation of the `BC-pAug89` trace, with the ICSS method at 99% significance, with `db3`, resolution = 5000, quorum = 4.

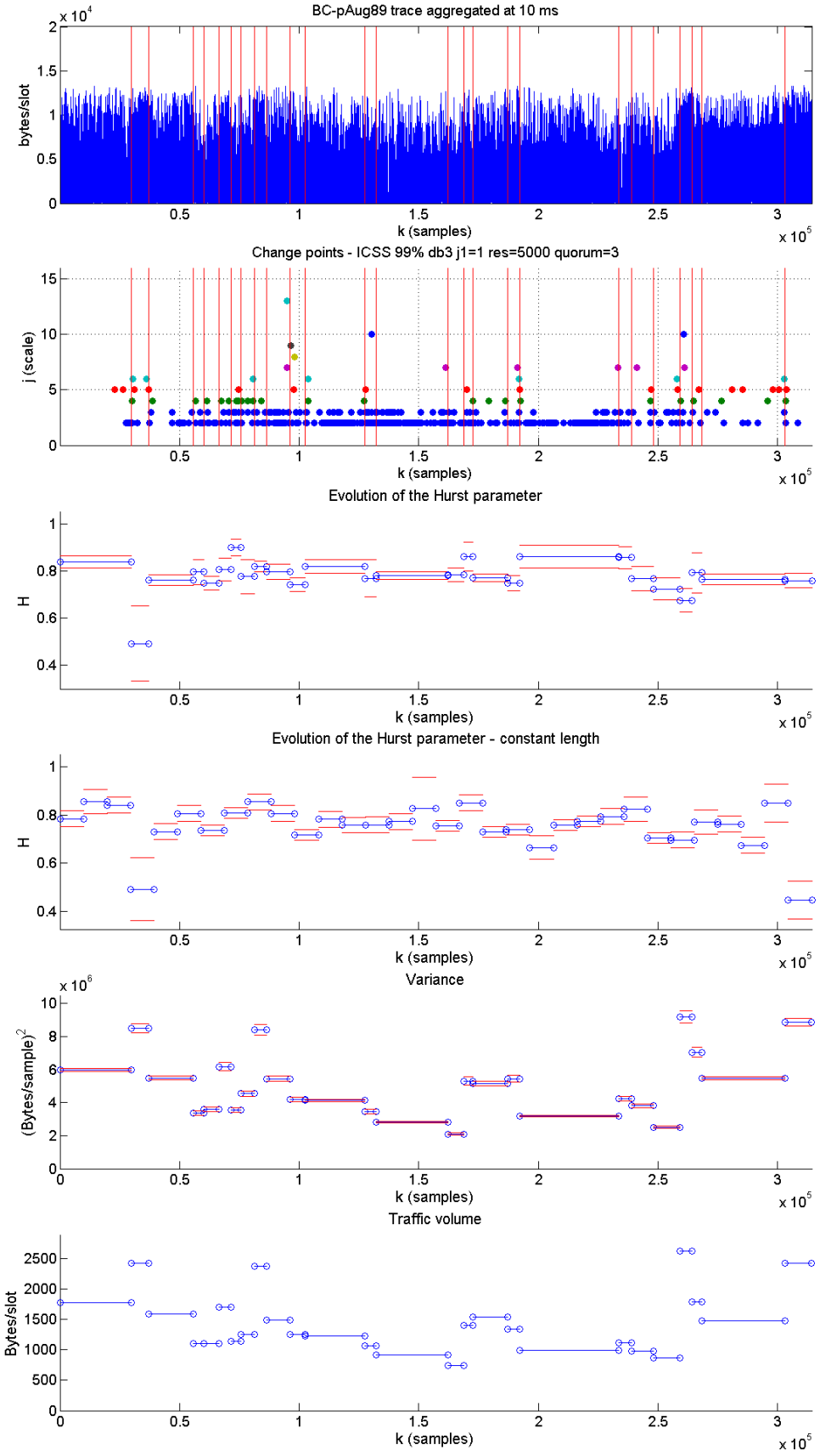

Figure 6.45: Segmentation of the `BC-pAug89` trace, with the ICSS method at 99% significance, with `db3`, resolution = 5000, quorum = 3. The first scale is ignored.

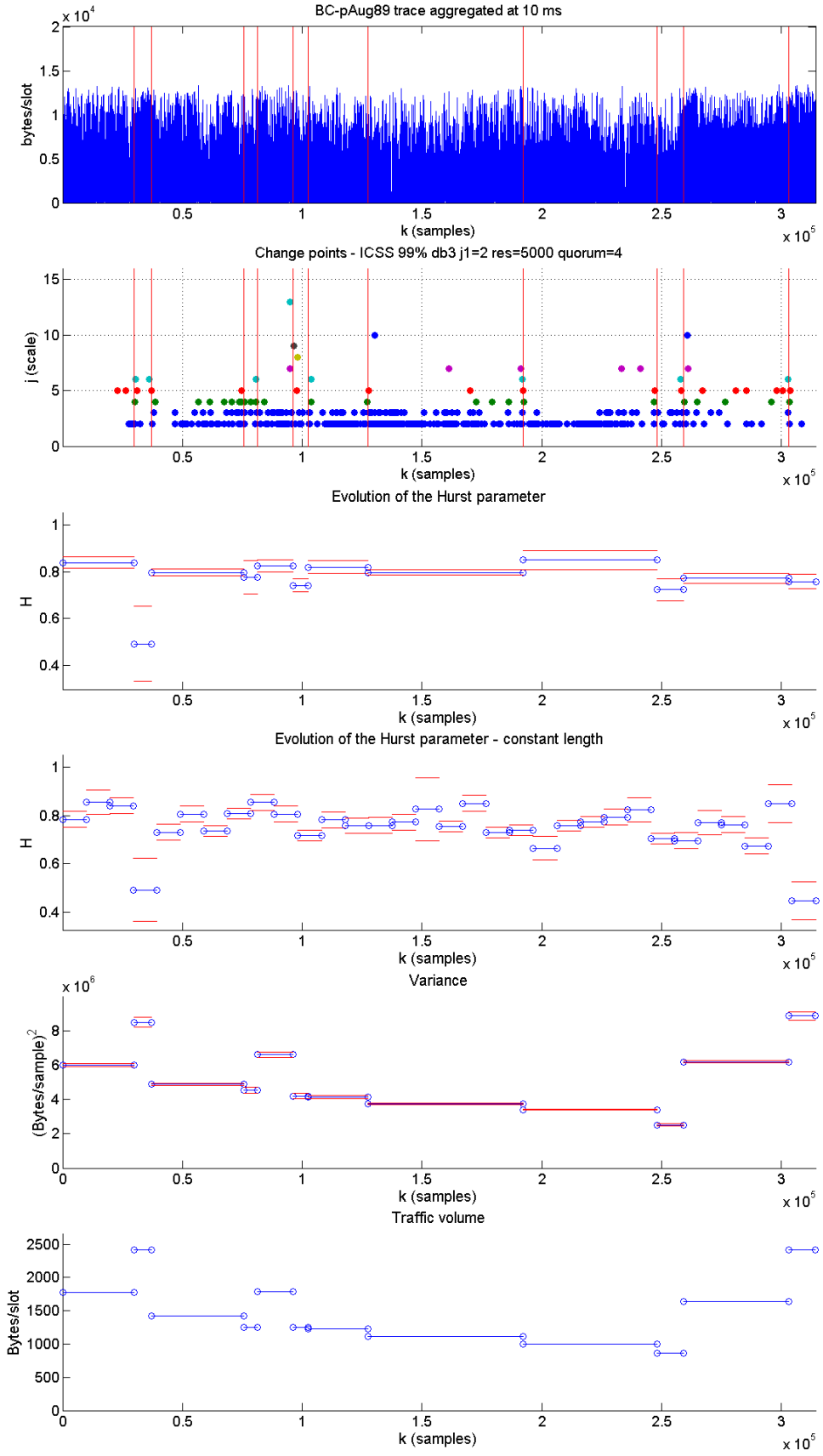

Figure 6.46: Segmentation of the `BC-pAug89` trace, with the ICSS method at 99% significance, with `db3`, resolution = 5000, quorum = 4. The first scale is ignored.

As has been already mentioned, the lower scales are very *noisy* in terms of the huge amount of changes. This increases both the computational load and the alignment bias. That is why we tried an analysis in which we ignore the first scale and decrease the quorum by one unit; these operations somewhat compensate their influence, since the *lost* quorum unit is actually the one given by the first scale for almost all positions. The results, shown in Figure 6.45, are almost the same as those obtained for the previous case. Therefore, we conclude that we can safely skip the first or second scales and still get the same results, with much fewer computations.

Figure 6.46 shows the case where the first scale is ignored and the quorum is maintained as 4. This would be equivalent to a quorum of 5 changes across all scales. Logically, we obtain a subset of the change alignments, corresponding to the regions where the variance changes are more abrupt (and therefore appear at more scales). The variation around the mean value of $H = 0.8$ is small, though the variations do exist, as shown by the almost non-overlapping confidence intervals. In some sense, increasing the quorum means losing details at the zones where the Hurst parameter is more or less constant, but we do not lose the abrupt H changes (such as the two regions around $k = 30000$ with $H \approx 0.5$ and $k = 250000$ with $H \approx 0.7$).

`BC-OctExt` trace

We now analyze the `OctExt1s.mat` trace aggregated at 1 s, which presents much higher variations than the `BC-pAug89` trace, as shown in Figure 6.47. The amount and size of the detected changes is higher than in the previous case, due to the higher sharpness of the probability density function. Regarding the influence of the wavelet, as in the study of the `BC-pAug89` trace, the `db3` analysis returns fewer change points than the `db1` case (typically one less scale for each aligned change point). We hypothesize that this effect is caused by the longer support of the `db3` wavelet, which makes the determination of the variance change point less accurate, due to its higher smoothing effect.

The following parameters were chosen for the analysis: 99% significance level, `db3` wavelet, resolution = 1000 samples and quorum = 5 changes. We tried again to get a variance and Hurst segmentation whose confidence intervals overlapping is minimized. The first scale is not considered for the analysis, due to its *noisy* nature, thus making the *real* quorum be 6 changes. The results are shown in Figure 6.48. The trace exhibits a much higher variance of the Hurst parameter, as was expected from the constant length analysis. The range of variation is between 0.1 and 1.55[8], though the sequence stays mainly in the typical $0.5 < H < 1$ range. Recall that $H > 1$ implies nonstationarity, as can be confirmed by looking at the behavior of the trace. The confidence interval argument we used in the analysis of the `BC-pAug89` trace is more difficult to apply here, due to the shorter length of the segments, which in turn limits the number of scales available for regression, which is one of the factors that influence the confidence intervals. In any case, most of the segments do not overlap their intervals; the cases where this happens belong to either a change of variance

[8] We considered the values under 0 and above 1.5 as outliers. This was confirmed by the extremely large associated confidence intervals.

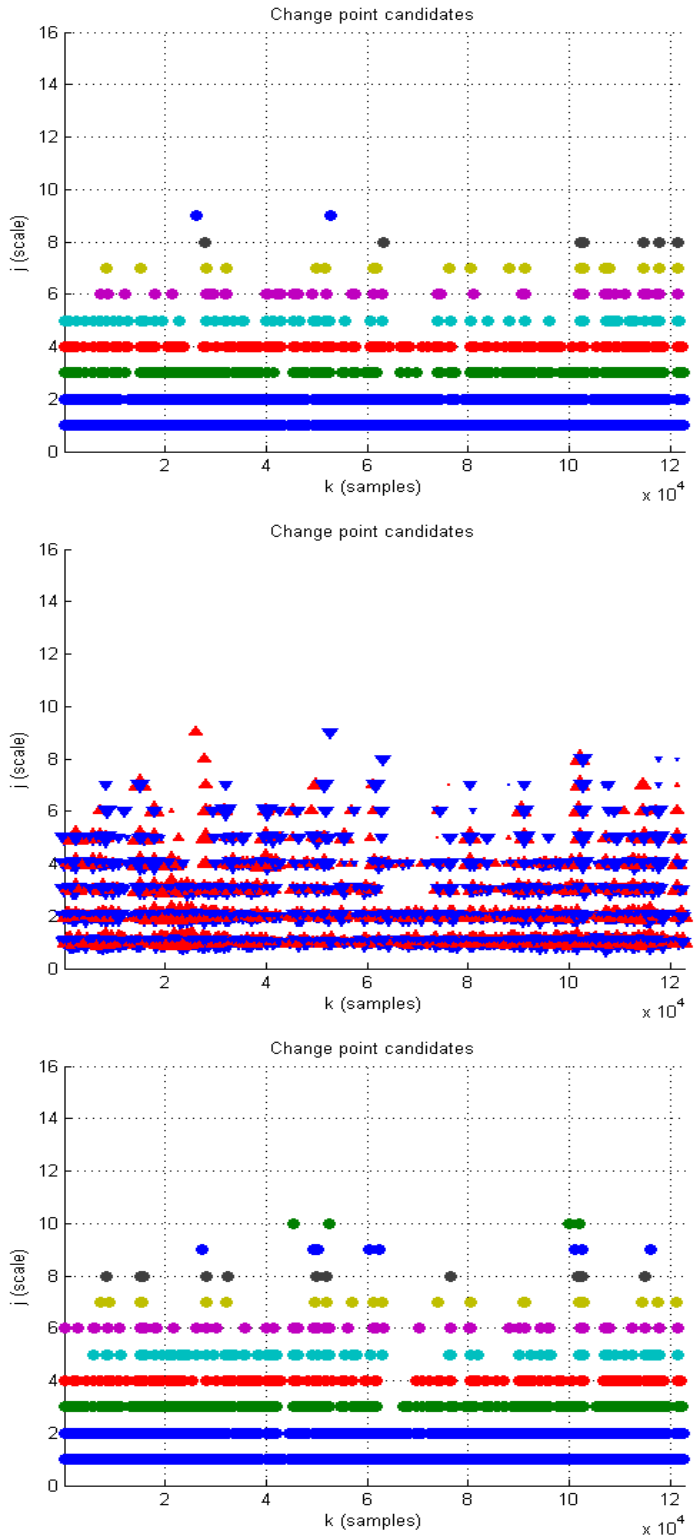

Figure 6.47: Change points at each scale for the DWT-ICSS analysis of the `BC-OctExt` trace, with `db1` wavelet, at the 99% significance level. *Top*: position of each change point, with `db1`. *Middle*: Sign and size of each change point. *Bottom*: position of each change point for `db3`.

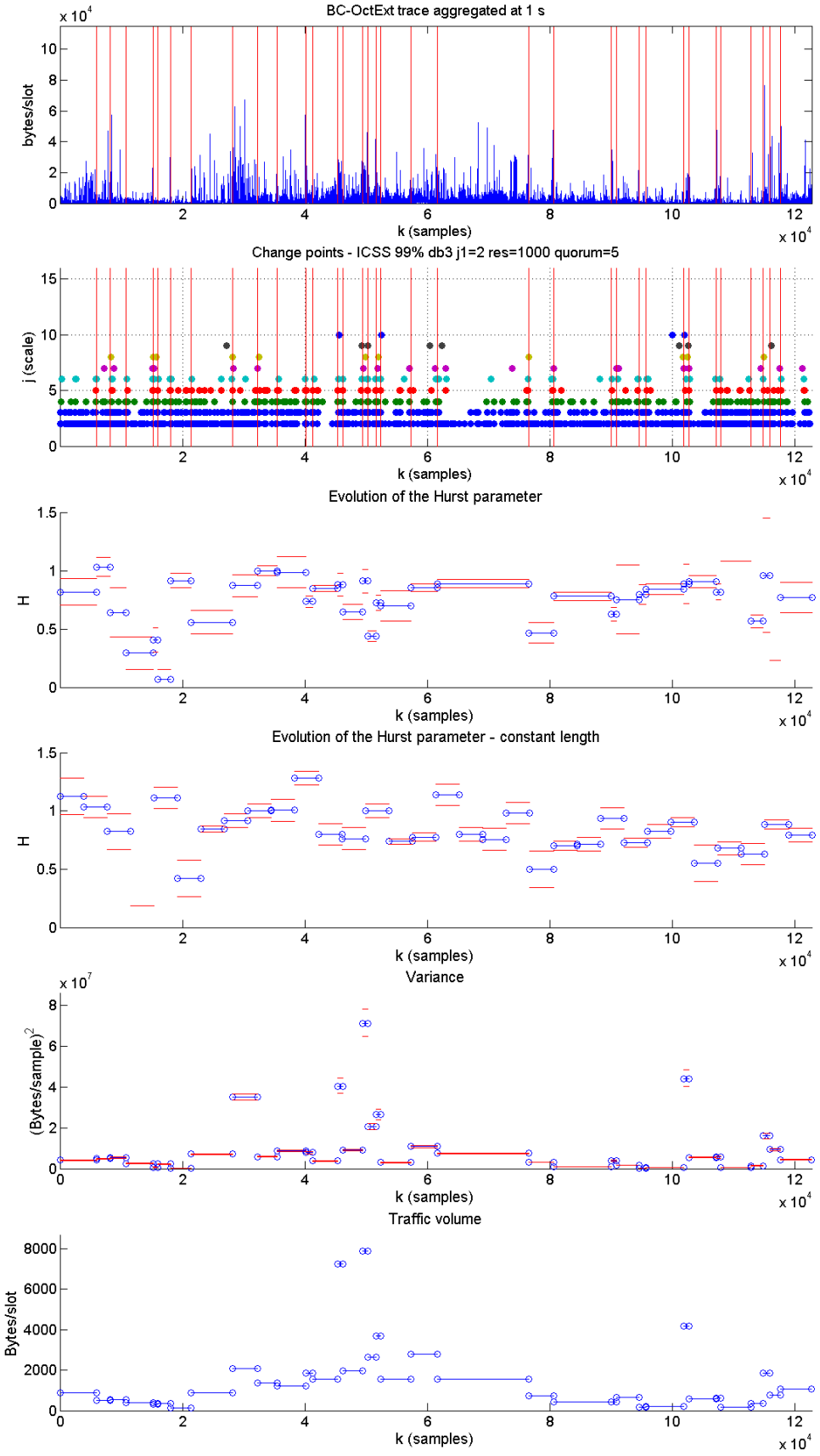

Figure 6.48: Segmentation of the `BC-OctExt` trace, with the ICSS method at 99% significance, with `db3`, resolution = 1000, quorum = 5. The first scale is ignored.

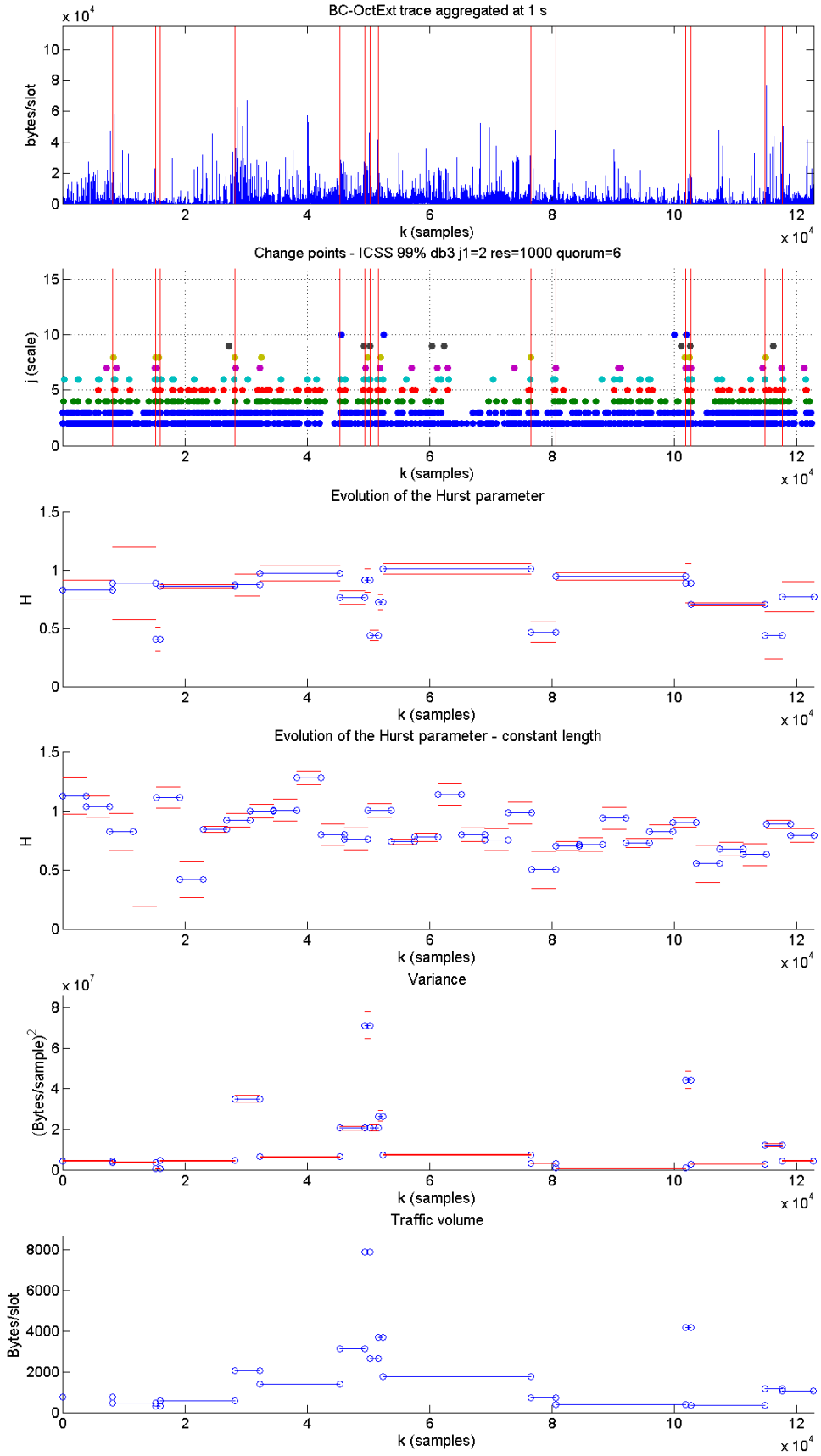

Figure 6.49: Segmentation of the `BC-OctExt` trace, with the ICSS method at 99% significance, with `db3`, resolution = 1000, quorum = 6. The first scale is ignored.

while H is constant (such as the change point located at $k \approx 63000$ or the opposite case (see for example the region from the origin up to around $k \approx 20000$, with almost no change of variance but with a highly variable Hurst parameter). Regarding the connection between traffic load and Hurst parameter, we again find no correlation between them.

We also tried with a quorum of 6 changes, attempting to simplify the diagram. The results of Figure 6.49 are not so appealing, since the change diagram is too simplified (for example, the long regions with $H \approx 1$ hide the rich behavior shown in Figure 6.48, with short regions alternating above and over $H = 1$). However, the results still include the abrupt falls to $H \approx 0.5$ that were already detected around $k = 50000$, $k = 78000$ and $k = 115000$.

6.9.3 DWT-SIC

`BC-pAug89` trace

Figure 6.50 shows the detected change points when using the SIC-based method with `db1` and `db3`. The results are similar to those obtained with DWT-ICSS, though for some alignments the number of detected change points is lower; the SIC seems to be more strict than the ICSS. Nevertheless, the SIC analysis is still able to identify the most important alignments and even enhance them, such as the clear alignment detected at $k = 300000$.

Figure 6.51 shows the results for the segmenting procedure with the same combination of parameters used in Figure 6.44 (99%, `db3`, resolution = 5000 samples and quorum = 4 changes). Given the decrease in the change detection, the segments found are fewer (and longer) than those found with the ICSS, thus redounding in a *simplification* of the segmentation diagrams. For example, we lose the rich behavior of H and the variance detected by the ICSS in the $50000 < k < 100000$ region with multiple changes in 4 aligned change points, that are missed by the SIC (only 3 changes are detected).

Given the stricter results returned by the SIC method, we tried two approaches: decreasing the quorum by one unit (Figure 6.52), and lowering the significance level to 95% (Figure 6.53). The decrease in the quorum returns almost the same results obtained with the ICSS, confirming that the reduction in the number of change points is almost linear. The other approach returns less segments (specially in the populated region $50000 < k < 100000$), but the clearest alignments are preserved.

Regarding the influence of the first scale, we repeated the experiment that consists of not considering the changes at $j = 1$ and decrease the quorum. The results shown in Figure 6.54 confirm that the detected alignments do not change from those shown in Figure 6.51.

`BC-OctExt` trace

The analysis of the `BC-OctExt` trace seems not to be very affected by the phenomenon mentioned in the previous paragraphs. When compared with the ICSS, the SIC method provides roughly the

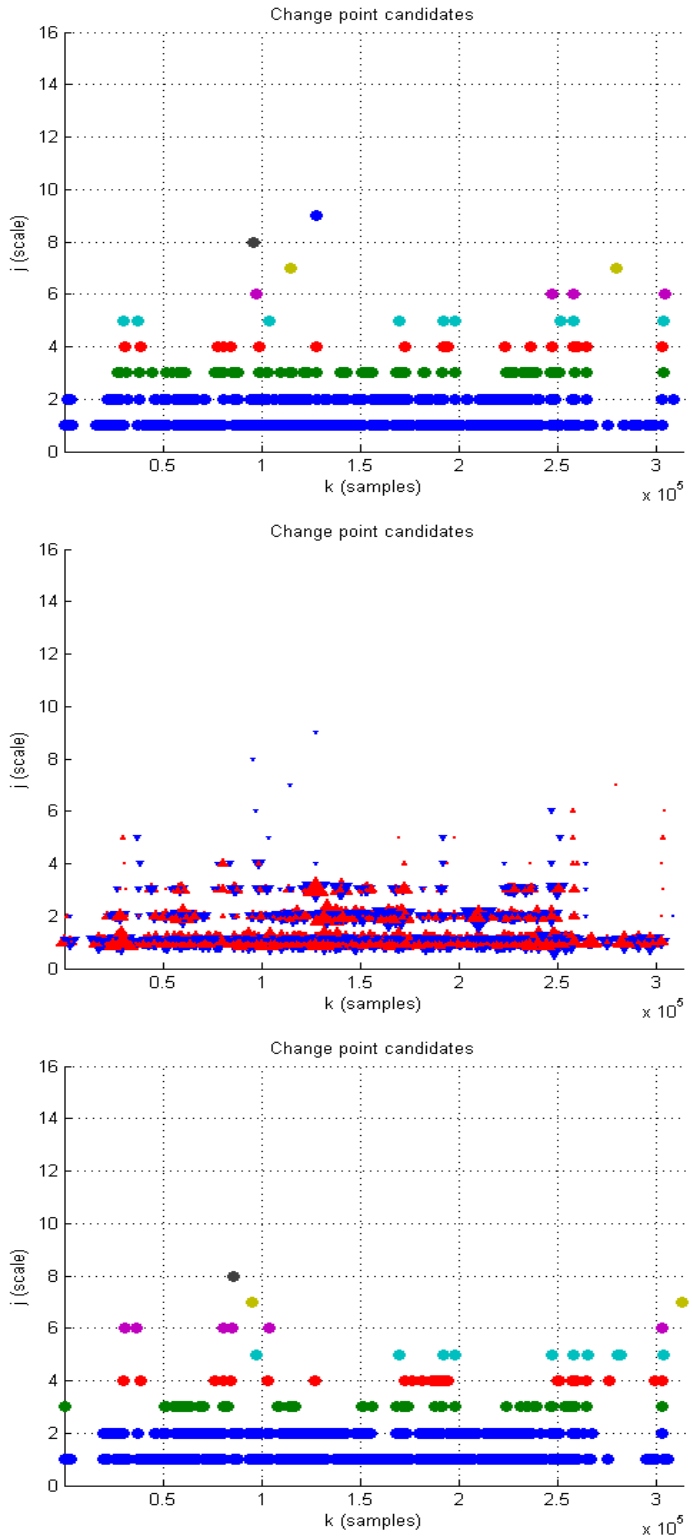

Figure 6.50: Change points at each scale for the DWT-SIC analysis of the `BC-pAug89` trace, with `db1` wavelet, at the 99% significance level, with `db1`. *Top*: position of each change point. *Middle*: Sign and size of each change point for `db1`. *Bottom*: position of each change point for `db3`.

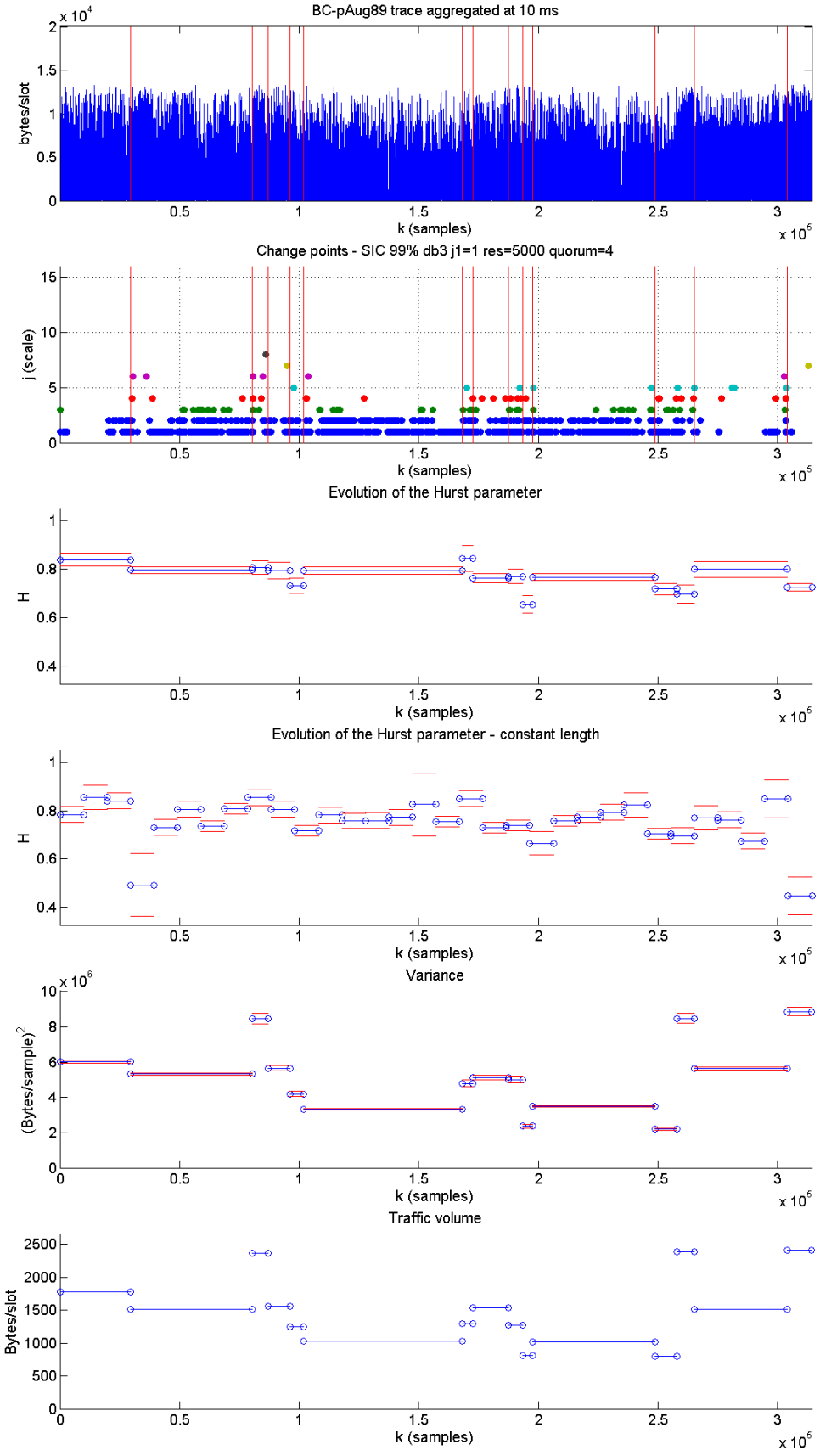

Figure 6.51: Segmentation of the `BC-pAug89` trace, with the SIC method at 99% significance, with `db3`, resolution = 5000, quorum = 4.

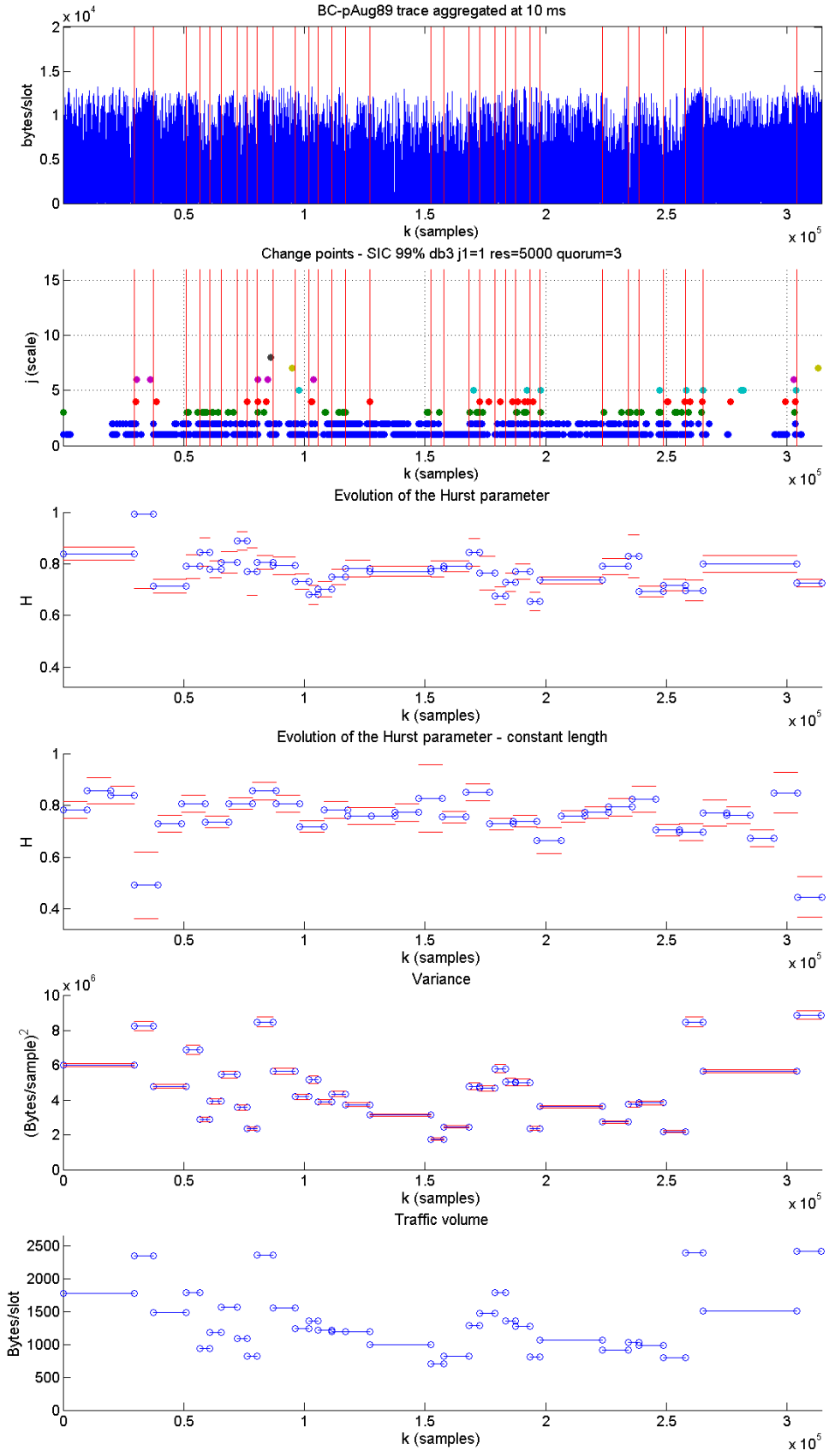

Figure 6.52: Segmentation of the `BC-pAug89` trace, with the SIC method at 99% significance, with `db3`, resolution = 5000, quorum = 3.

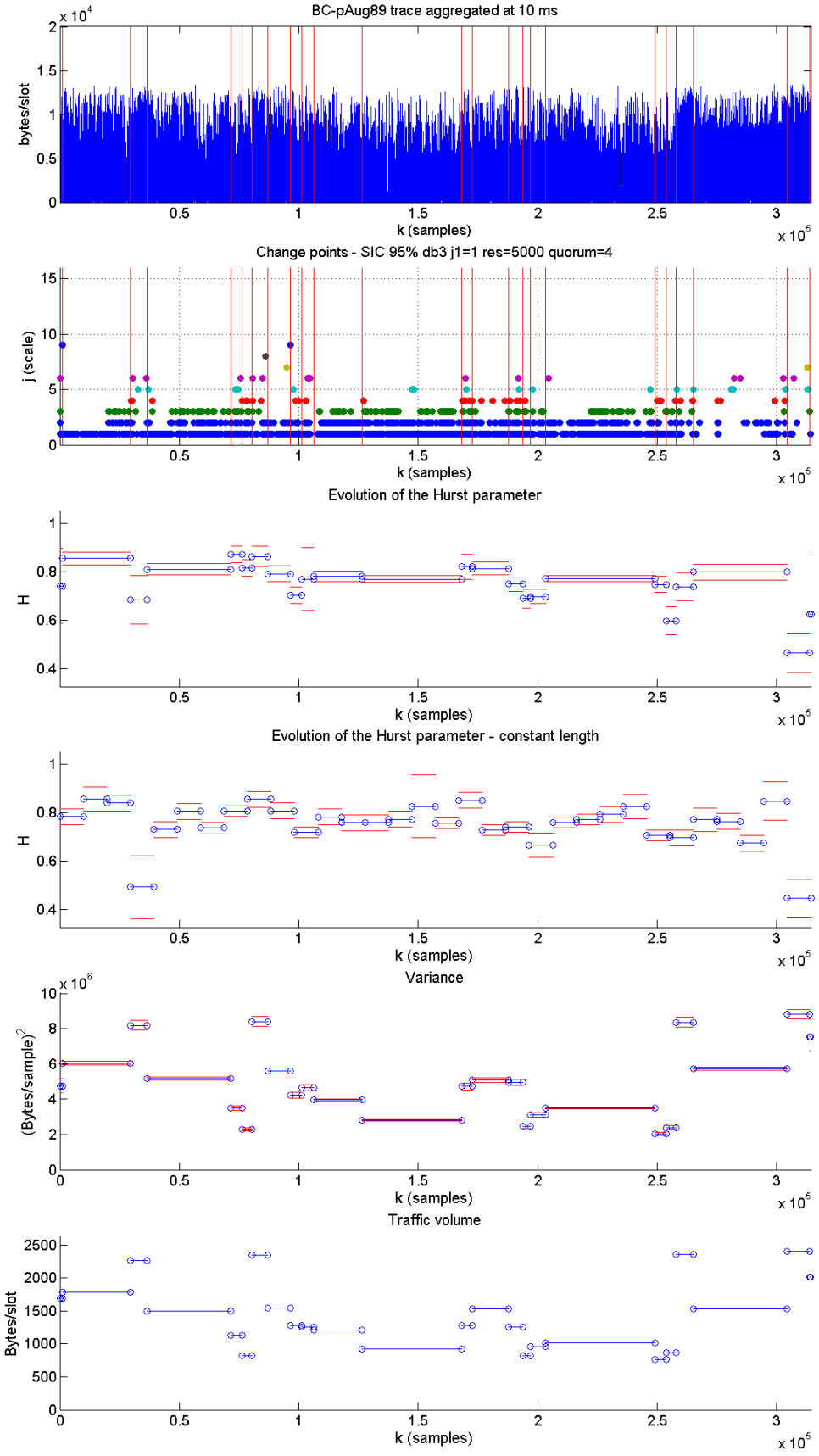

Figure 6.53: Segmentation of the `BC-pAug89` trace, with the SIC method at 95% significance, with `db3`, resolution = 5000, quorum = 4.

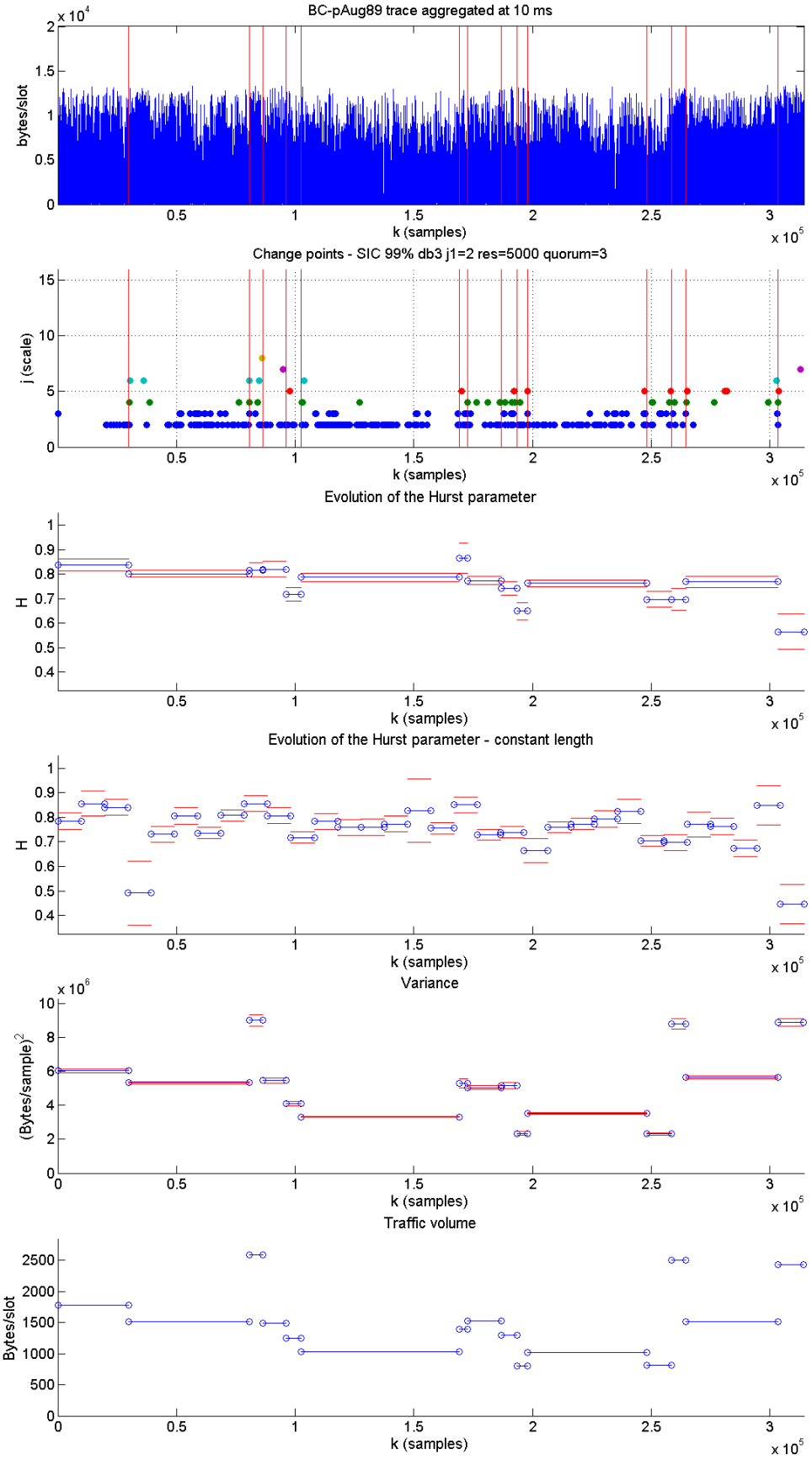

Figure 6.54: Segmentation of the `BC-pAug89` trace, with the SIC method at 99% significance, with `db3`, resolution = 5000, quorum = 3. The first scale is ignored.

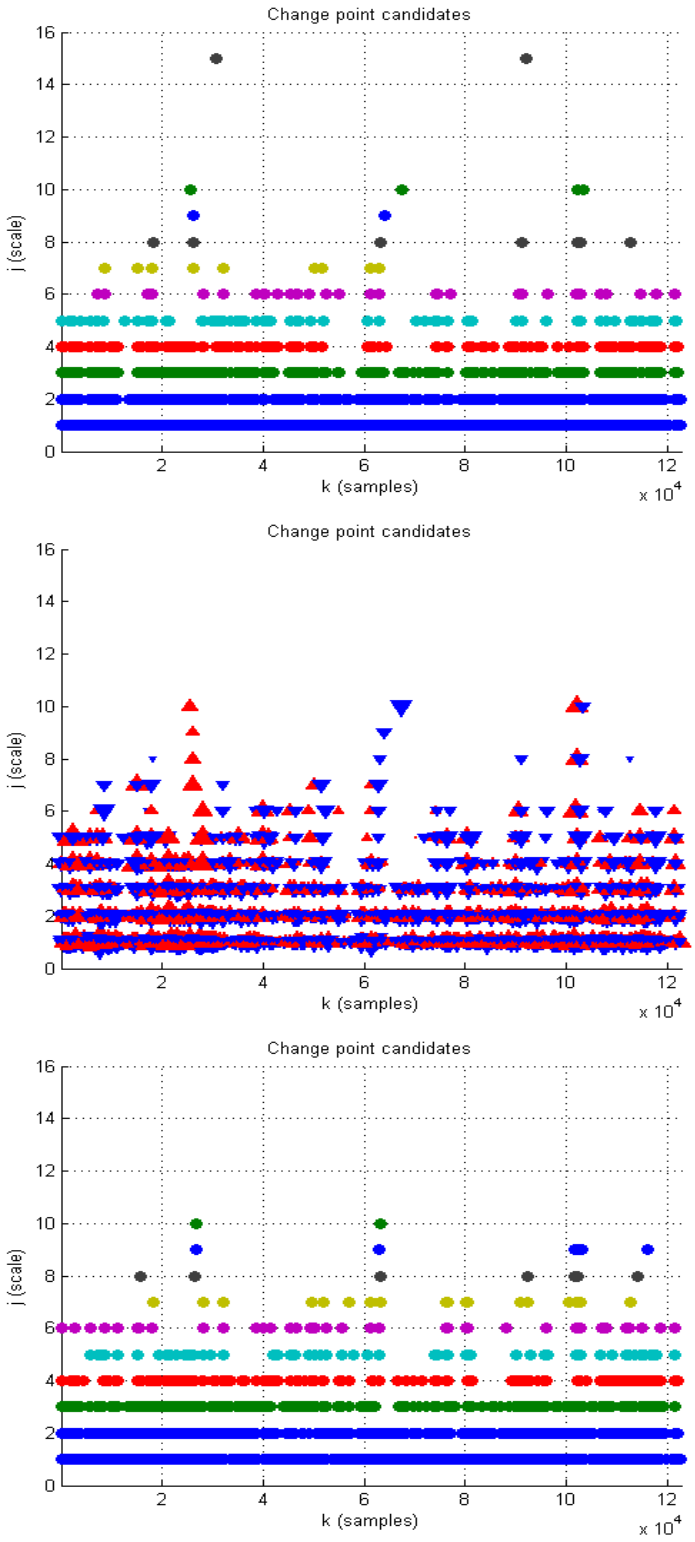

Figure 6.55: Change points at each scale for the DWT-SIC analysis of the `BC-OctExt` trace, with `db1` wavelet, at the 99% significance level. *Top*: position of each change point, with `db1`. *Middle*: Sign and size of each change point. *Bottom*: position of each change point for `db3`.

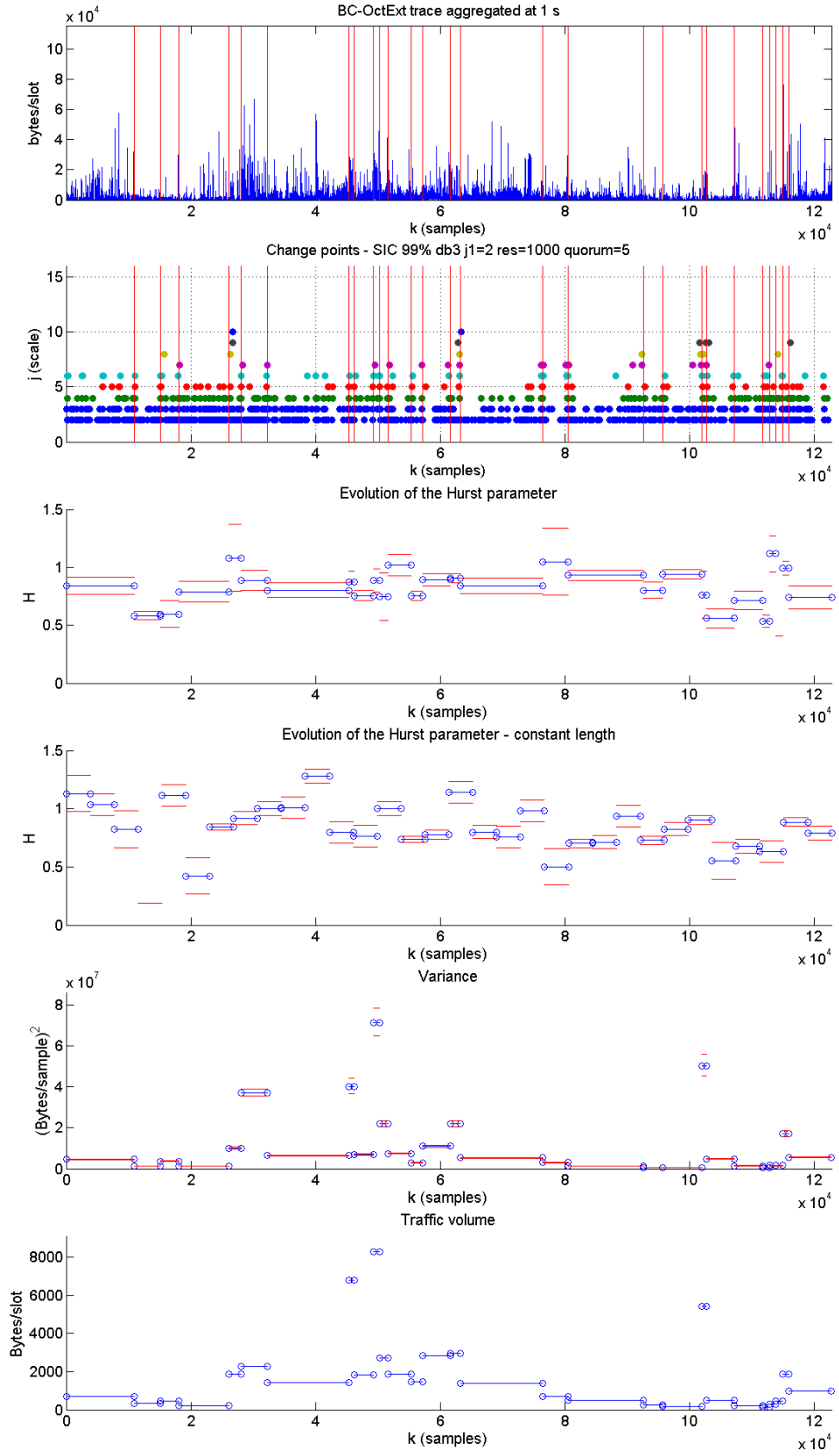

Figure 6.56: Segmentation of the `BC-OctExt` trace, with the SIC method at 99% significance, with `db3`, resolution = 1000, quorum = 5. The first scale is ignored.

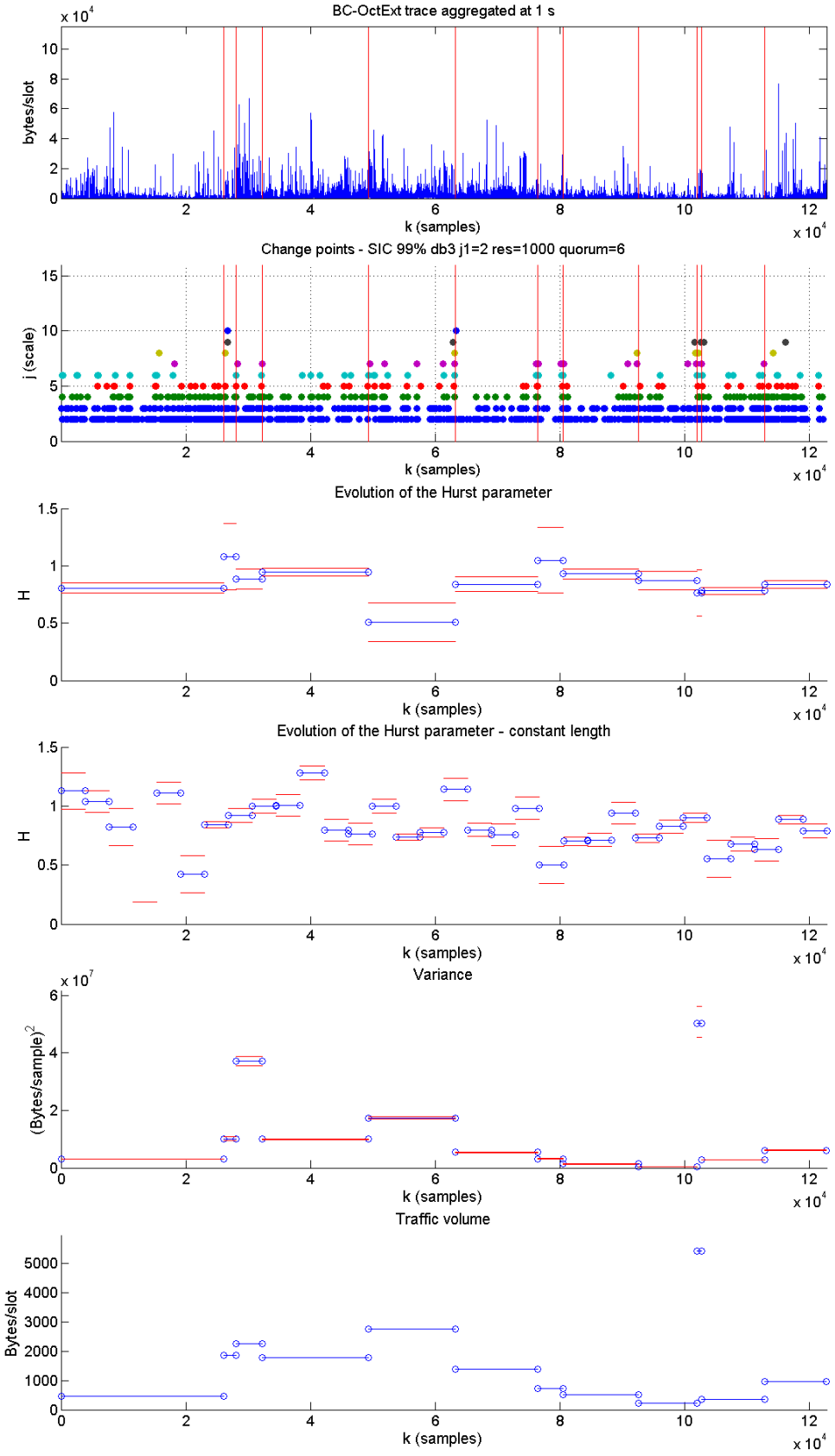

Figure 6.57: Segmentation of the `BC-OctExt` trace, with the SIC method at 99% significance, with `db3`, resolution = 1000, quorum = 6. The first scale is ignored.

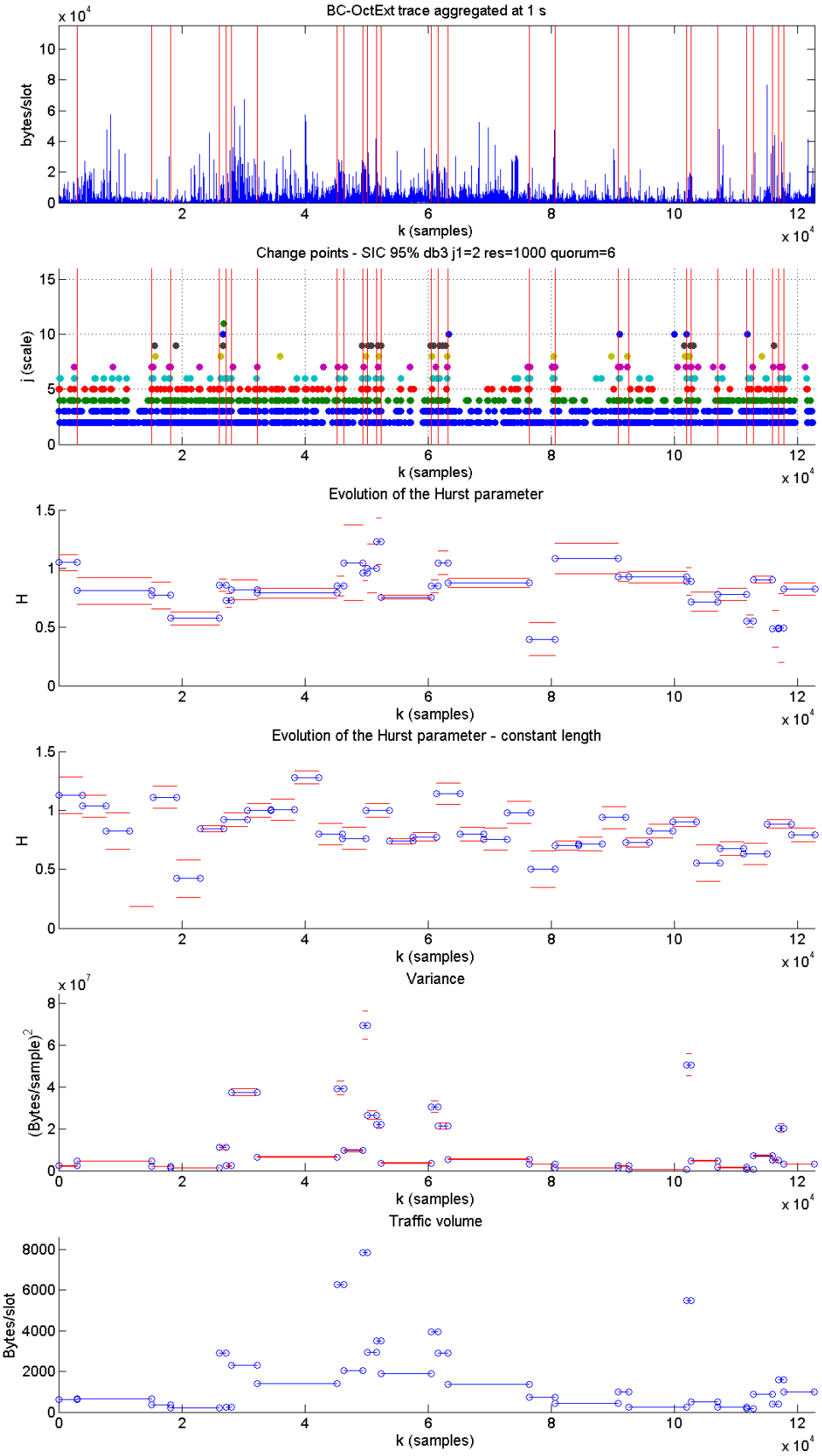

Figure 6.58: Segmentation of the `BC-OctExt` trace, with the SIC method at 95% significance, with `db3`, resolution = 1000, quorum = 5. The first scale is ignored.

same change points, as shown in Figure 6.55, though Figures 6.56 and 6.57 show fewer detected alignments than their ICSS counterparts, though as in the case of `BC-pAug89` trace, the more important alignments are still detected.

For the sake of completeness, we also include the results for the 95% significance level analysis with quorum = 5 changes, shown in Figure 6.56, from which we obtain similar results to those given by the ICSS method at the 99% significance and a quorum of 5 changes.

This section has presented the results of the segmentation algorithms when applied to a couple of real traffic traces which are representative of two different situations with moderate and extreme changes of variance and Hurst parameter. Regarding the behavior of the methods, our main conclusion is that in order to compare fairly the ICSS and the SIC methods for real traces, we have to either lower the quorum in one unit for the SIC study, or to lower the significance level from 99% to 95%.

6.10 Progressive version of the DWT-based methods

6.10.1 Sequential DWT-based algorithms

One of the main reasons behind our work on the wavelet-based segmentation and clustering algorithms is the possibility they provide for performing an *on-line* analysis of traffic. The main idea underlying this research area is that the change point detection method can be computed *sequentially*, due to the fact that the DWT can be performed progressively. That is, beginning at a certain origin $t = 0$, the SIC or ICSS statistics can be updated with new samples (or groups of samples) as time progresses and the change point identification can be consequently rerun. Another possibility is a sliding window approach, in which the ICSS or SIC methods would be applied only to a certain amount of the last samples captured by the analyzer. The open question here is the length of the window, i.e., the *memory* of the system.

The main problem related to the progressive algorithm is given by the binary segmentation procedure of the SIC and ICSS methods and the re-evaluation phase at the end of the ICSS, whose number of iterations cannot be predicted. This happens because the computations completely depend on the existence of a variance change point in the studied data. In other words, there is no way to predict the number of times that the rerun of the ICSS/SIC algorithm on the corresponding subsequences must be performed, since we do not previously know the number and location of the change points. This is why we cannot ensure that the complete process can be performed in real time, although our algorithms allow some truncation, such as setting the minimum segment size n_{min}.

This does not mean we have to abandon our goal of developing a real-time version of the algorithms. We have been able to adapt the methods and provide a *progressive* or *sequential* version, which is not 100% real-time, but approximates that. It consists of:

1) An efficient implementation of the most costly steps of the static ICSS and SIC methods, which was identified as the cumulated sum of squares of the samples of the time series. Our code for both ICSS and SIC methods computes the squares cumsum only once for the whole sequence and stores the result as a fast-access global variable that can be read by any of the subroutines that perform the iterative steps.

2) A simple method for updating the squares cumsum by adding the new samples and extracting the old ones (the latter applies only to the sliding window version of the algorithm, yet to be coded).

3) The introduction of a *granularity* parameter G, which is defined as the number of new samples added to the trace to be analyzed at every update. The G value clearly influences the computational load, since the number of reruns of the SIC algorithm at every scale is given by $\lfloor N/G \rfloor$, where N is the "off-line" data size.

In the following, we present two examples in which we evaluate the behavior of the DWT-ICSS algorithm when adding new samples to the original trace results in a new change point to be detected.

6.10.2 Application to a synthetic trace with changes in the Hurst parameter

We first apply the DWT-SIC sequential algorithm to a synthetic FGN trace composed of 3 segments with H=0.5, 0.9 and 0.5, with change points at $k = 32768$ and $k = 98304$. In Figure 6.59 we have plotted the three-dimensional change points diagram (top), obtained by using the `db1` wavelet family and setting $G = 8192$, and its projection on the (k, t) plane (bottom). The other free parameters were set to the same values used in the correspondent off-line analysis.

In the three-dimensional diagram, the k axis refers to the *change points position*, the t axis represents the *analyzed time* (as new samples are added), while the j axis is the *scale* at which the change is detected. It is readily apparent that this three-dimensional plot represents a generalization of the bidimensional change points diagrams previously analyzed in this section.

As we can see, the two change points are clearly detected at almost all scales and, what is more important, the detection of a change point is maintained as time progresses. In this sense, the algorithm is *coherent*. This is an important feature because false change candidates appear, but, thanks to the coherence of the algorithm, they can be easily detected, since they disappear as the algorithm progresses with new data. As an example, consider the spurious change point temporarily detected at $k \approx 85000, t \approx 105000, j = 6$, that appears in just one occasion and disappears as the next update of the algorithm. The ICSS method in some way corrects its previous incorrect estimation as it advances in the analysis.

Figure 6.60 shows the sliding window version of the algorithm, in which instead of accumulating samples from $k = 1$, the analysis is performed over a span of S samples. This is the progressive

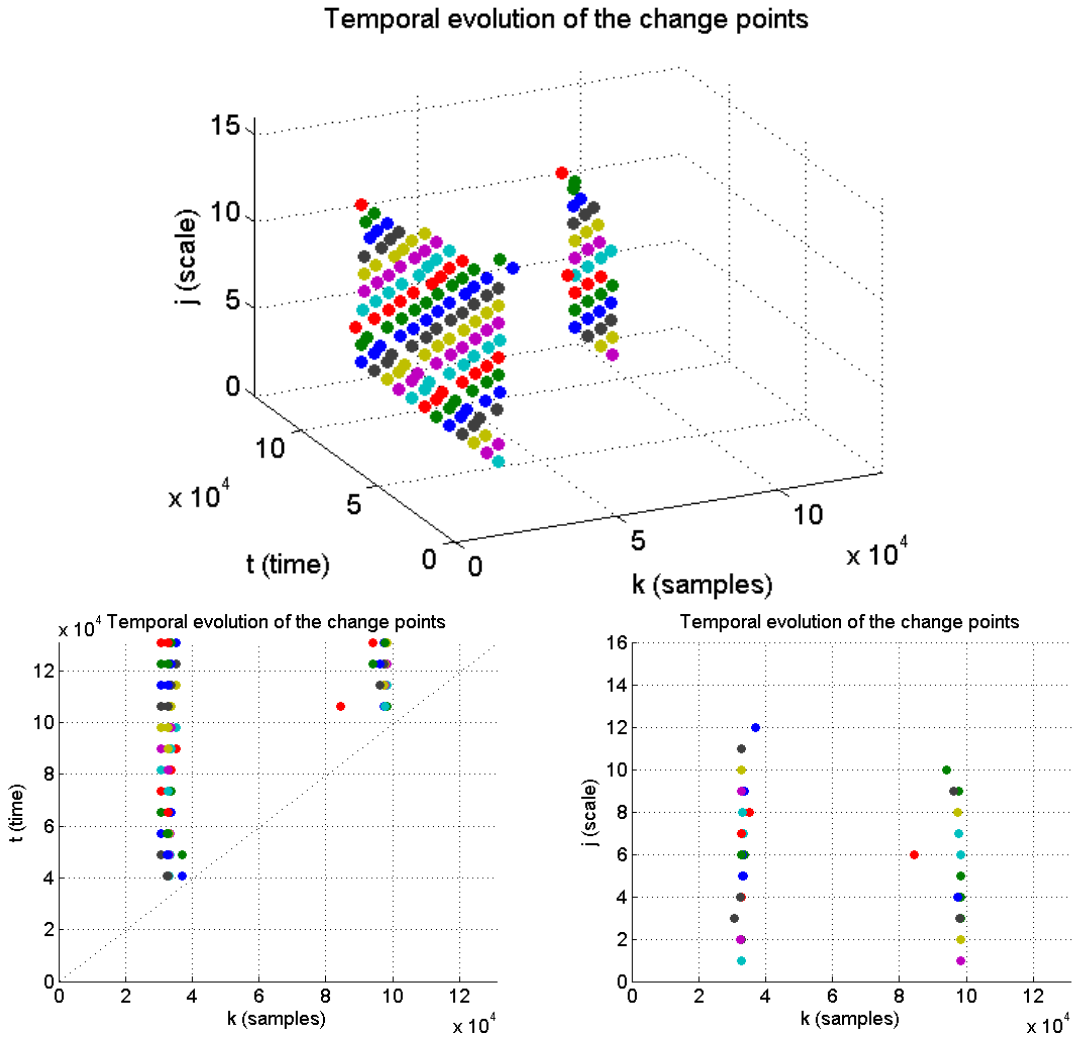

Figure 6.59: *Top*: Three-dimensional evolution of the DWT-ICSS algorithm when applied to a FGN trace composed of 3 segments with H=0.5, 0.9 and 0.5, with changepoints at $k = 32768$ and $k = 98304$. The analysis was performed at the 99% significance level with `db1` wavelet and granularity $G = 8192$ samples. *Bottom left*: The *time-versus-time* projection of the top side. The dotted line corresponds to the analysis progress. *Bottom right*: The *changes-versus-time* cumulated change diagram obtained by projecting the 3D diagram.

version of our algorithm.

6.10.3 Application to the `BC-pAug89` Bellcore trace

We now apply the DWT-ICSS progressive algorithm to the `BC-Aug89` trace described in Paragraph 3.2.10. The chosen granularity was $G = 32768$ and the span $S = 4 * G = 131072$ samples, while for the other parameters we chose the same values used in the previous example. Using the `db1`

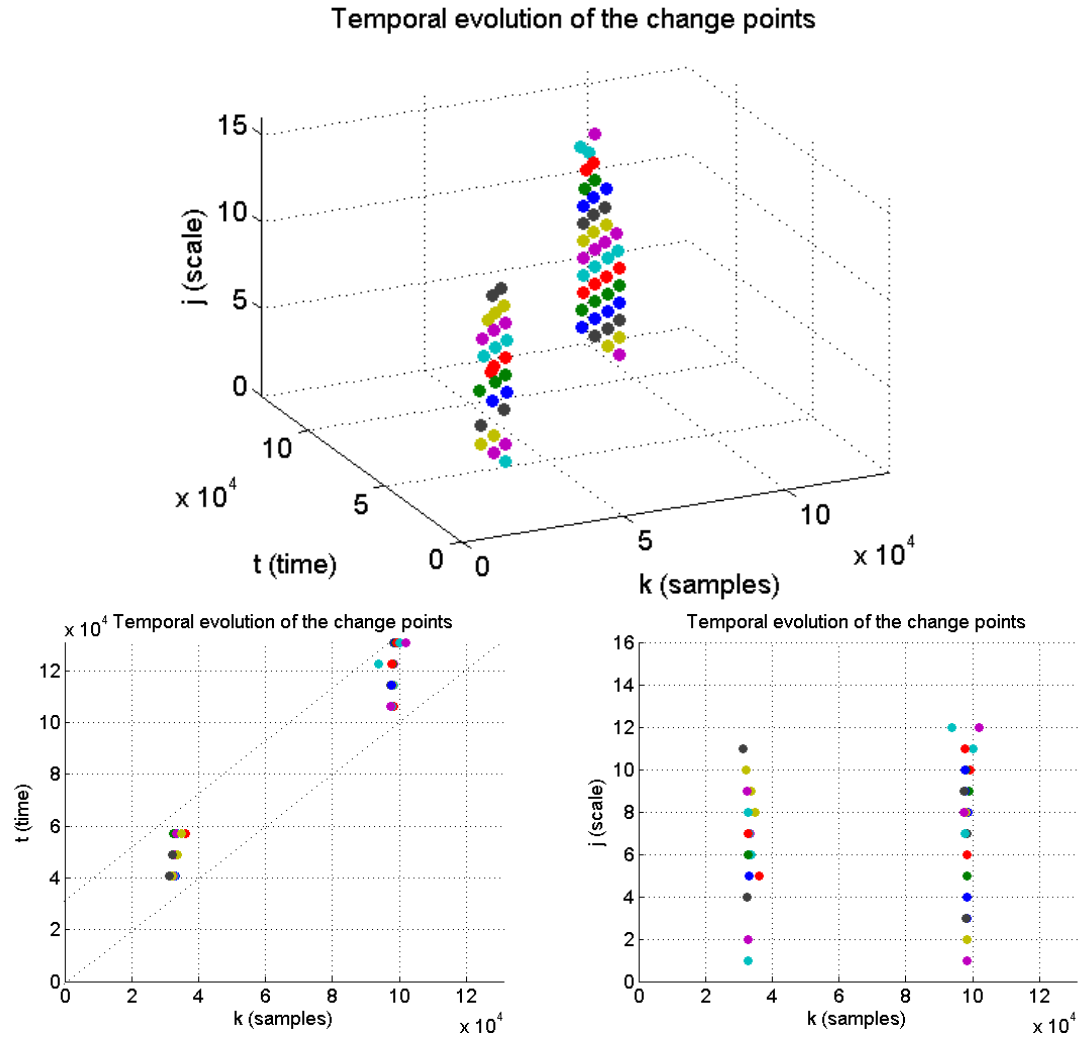

Figure 6.60: *Top*: Three-dimensional evolution of the DWT-ICSS algorithm when applied to a FGN composed of 3 FGN segments with H=0.5, 0.9 and 0.5, with changepoints at $k = 32768$ and $k = 98304$. The analysis was performed at the 99% significance level with `db1` wavelet, granularity $G = 8192$ samples and span of $S = 4 * G = 32768$ samples. *Bottom left*: The *time-versus-time* projection of the top side. The dotted lines correspond to the analyzed region. *Bottom right*: The *changes-versus-time* cumulated change diagram obtained by projecting the 3D diagram.

wavelet family, we obtained the three-dimensional change points diagram shown in Figure 6.61 (top). Looking at this plot, we can see again the coherence of the change candidates. Furthermore, its projection on the (k, j) plane (depicted in the bottom side of the figure) is similar (though not exactly equal) to Figure 6.42. The differences between the pictures come from the different samples set used for the analysis, which generates a different set of spurious changes. We conclude that the progressive version of our estimation algorithm seems to be as reliable as the off-line version.

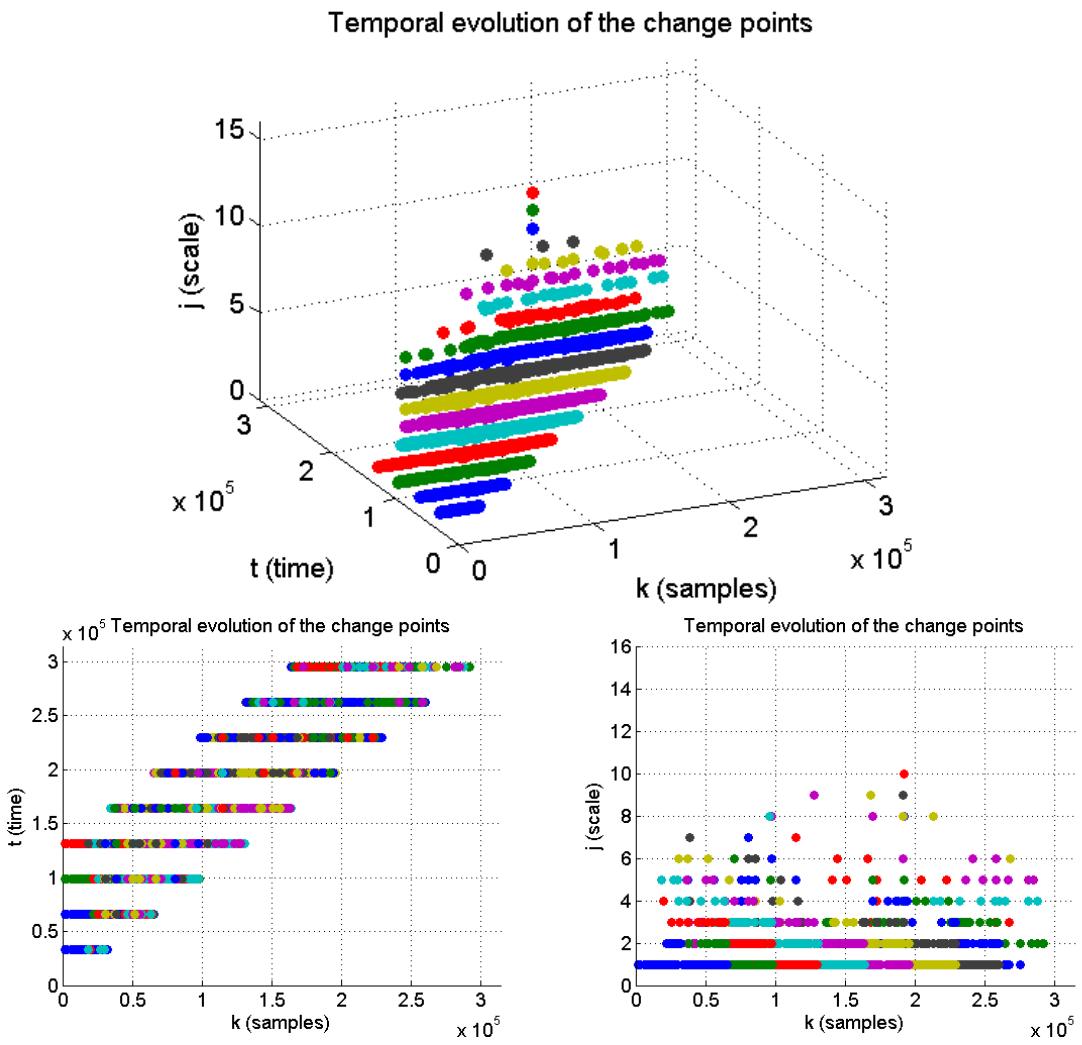

Figure 6.61: *Top*: Three-dimensional evolution of the DWT-ICSS algorithm when applied to the `BC-Aug89` trace aggregated at 10 ms. The analysis was performed at the 99% significance level with `db1` wavelet, granularity $G = 8192$ samples and span of $S = 4 * G = 32768$ samples. *Bottom left*: The *time-versus-time* projection of the top side. The dotted lines correspond to the analyzed region. *Bottom right*: The *changes-versus-time* cumulated change diagram obtained by projecting the 3D diagram.

6.11 Summary of the chapter

Throughout this chapter the DWT-based methods (DWT-ICSS, DWT-SIC) have been analyzed in several situations, including synthetic and real traffic traces. The main conclusions are:

- It has been checked that abrupt mean changes can create false changes, but we can easily discard them with the alignment detection procedure.
- Variance changes, on the other hand, are very clearly detected as the alignment of changes with

the same sign (increase or decrease) across scales, as long as the resolution of the alignment procedure is of the order of the variations of the positions of the changes and the quorum is not too high. Resolutions of 200 samples and quorums of 3 to 5 changes seem to be good choices for FGN synthetic traces.

- The changes in the Hurst parameter appear also as aligned variance changes, but the sign is different depending on the scale region, and can include a *blind* scale or scales in which no change or very little is detected. The higher dispersion of the points requires a higher resolution than that used for detecting pure variance changes, or a decrease in the quorum. A good parameter combination with which more than 85% of the true change points for a $H = 0.5 \rightarrow 0.9$ were found was: 99% significance level, `db1` wavelet, resolution=200 samples, quorum=4 changes. The use of a higher order wavelet such as `db3` provides slightly worse results, but still over 85% of the true change points. For smoother H changes the detection ratios decrease, and resolutions of 1000 to 5000 samples are required. DWT-SIC performs slightly better than DWT-ICSS for the more abrupt H changes, and vice versa, due to the fact that the ICSS returns more points (either "good" or spurious), while the SIC is stricter.

- Regarding the analysis of traces with multiple change points, we studied the performance of the algorithms as a function of the separation of the change points. As was expected, the smooth H changes require higher separations than the abrupt H changes. Separations under 5000 samples are very hard to detect, due to the smoothing effect of the DWT at the higher scales. Since change points are usually aligned across 8 to 10 scales, at least 1000 samples of separation are needed in order to detect two consecutive changes.

- The methods have been used for analyzing a couple of real traffic traces from the Bellcore dataset, segmenting them in regions with (almost) homogeneous variance structure and estimating the evolution of the variance and the Hurst parameter along time. We have identified multiple situations: pure H changes, pure variance changes, and simultaneous H and variance changes. In general terms, the SIC method seems to perform better than the ICSS (regarding the location of changes) when they are compared under the same conditions, but since SIC is slightly stricter than the ICSS, it can happen that the latter detects more changes. This can influence the performance of the whole procedure, since the quorum has to be decreased for the SIC.

- For the choice of parameters for the real traffic analysis, we recommend using a 99% significance level, resolutions between 1000 and 5000 samples, and quorums of 3 to 5 changes. Of course, the actual choice will depend on the application and the time granularity needs of the algorithm which will use the segmentation data.

- The results from the segmentation of real traces are *coherent* with those obtained with the constant-length window analysis, but our method gives more information about the evolution of the trace. The `BC-pAug89` trace has been found to have more variations than those usually reported in the literature, while `BC-OctExt` has been confirmed as a very variable trace. The

evolution of the traffic volume (or the variance) is not correlated with the behavior of the Hurst parameter, in contradiction with the results obtained by other authors.

- Finally, a progressive version of our algorithm has been described. This modification can work either in cumulative or sliding window modes, and is the first step towards a real-time version of our DWT-based algorithms.

The main problem of using the DWT with the variance change detection statistics is the increasing loss of time resolution with scale, which makes the indetermination of the true location of the change point more difficult. The other problem of the DWT-based schemes is the lack of time-shifting invariance. This can make the algorithm give different results for the same input series when the relative phase between the input and the filter bank system differ; i.e., $x(n)$ and $x(n-1)$ provide different results. This is, of course, undesired. These two problems caused us to try other, time-redundant wavelet transforms: the MODWT and the DTWT.

Chapter 7

Alternative segmentation methods (I): MODWT

7.1 Introduction

The previous chapter pointed out some of the drawbacks of the DWT, such as the smoothing effect and the lack of resolution, both phenomena happening at the higher scales. Though we are satisfied with the performance of the DWT-based methods, we have explored alternative wavelet transforms with better properties, such as the Maximal Overlap DWT and the Dual Tree WT. These new methods have not been explored as deeply as the DWT, since we only wanted to assess if they could be good alternatives. That is why this chapter and the next one provide only a subset of the tests performed with the DWT.

The MODWT is the other extreme of the DWT in terms of time redundancy, since it skips completely the downsampling step and produces a wavelet coefficient at each scale for each sample at the input. This means that for a J-level MODWT, $(J+1)N$ coefficients are obtained[1]. This solves the problem of the lack of coefficients at the higher scales. Besides, the coefficients have a very important property: they are perfectly aligned in time with the input sample (see Figure 7.1). On the other hand, as we saw in Section 2.6.4, the redundancy introduces a lot of correlation. Since the ICSS (and the SIC) expect an uncorrelated input this is a problem for our methods. In the first phase we tried to correct it by using the Equivalent Degrees of Freedom (EDOF) approach, as described in Section 7.3.1 with disappointing results. In the second phase we tried by increasing the values of the significance level and the minimum segment size n_{min}.

Following the same scheme presented in Chapter 6 for the DWT, this chapter presents the issues related to the MODWT-ICSS and MODWT-SIC methods, by studying its performance when applied to synthetic LRD and real traffic traces. All the results are compared with those

[1] N coefficients for each of the J details and the approximation.

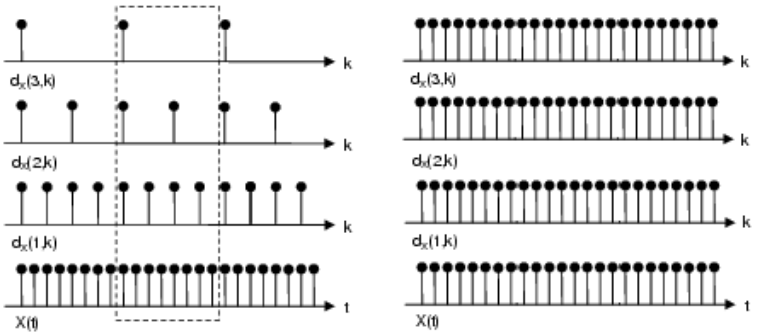

Figure 7.1: *Left*: time alignment between the DWT coefficients at scales $j = 1, 2, 3$, using *db1* wavelet. *Right*: the MODWT coefficients.

found with the DWT. The results presented in this chapter have been published mainly in the following references: [RS05c, RSdA05, Min05, RMSP05b, RMS06].

7.2 Statistical distribution of the MODWT coefficients

Figure 7.2 shows the distribution of the MODWT coefficients of an FGN trace at various scales. The coefficients tend to be normally distributed, though at the higher scales the distribution seems to deviate from Gaussian. However, the estimator of the shape parameter for the Generalized Gaussian distribution returns values of $1.98, 2.04$ and 1.99 for $j = 1, 5$ and 10, respectively, confirming the Gaussianity of the MODWT coefficients of FGN traces.

7.3 Analysis of the effects of correlation on a pure FGN trace

We start our analysis with an FGN trace of length 131072 samples, Hurst parameter $H = 0.8$, zero mean and unit variance, plot in Figure 5.14. The trace has been analyzed with the `db1` wavelet, at a depth $J = 12$. Figure 7.3 shows the MODWT detail coefficients at different scales, while Figure 7.4 plots the CSS statistic, with the 99% critical values of the statistic. Finally, Figure 7.5 plots the autocorrelation functions of the wavelet details at each scale.

The figures show clearly the presence of a high autocorrelation that increases with the scale. This effect induced by the MODWT was already described in Chapter 2, but here we see the consequences, when combined with the ICSS. The correlation makes the CSS statistic to take values far over the critical value, though there is no real variance change in the original series. Figure 7.6 shows the corresponding change point diagram, where a huge amount of (false) change points appear at the higher scales.

Does this invalidate the use of the MODWT? It would be a pity to miss out on the resolution properties of the transform. That is why we tried with two correlation-correcting approaches,

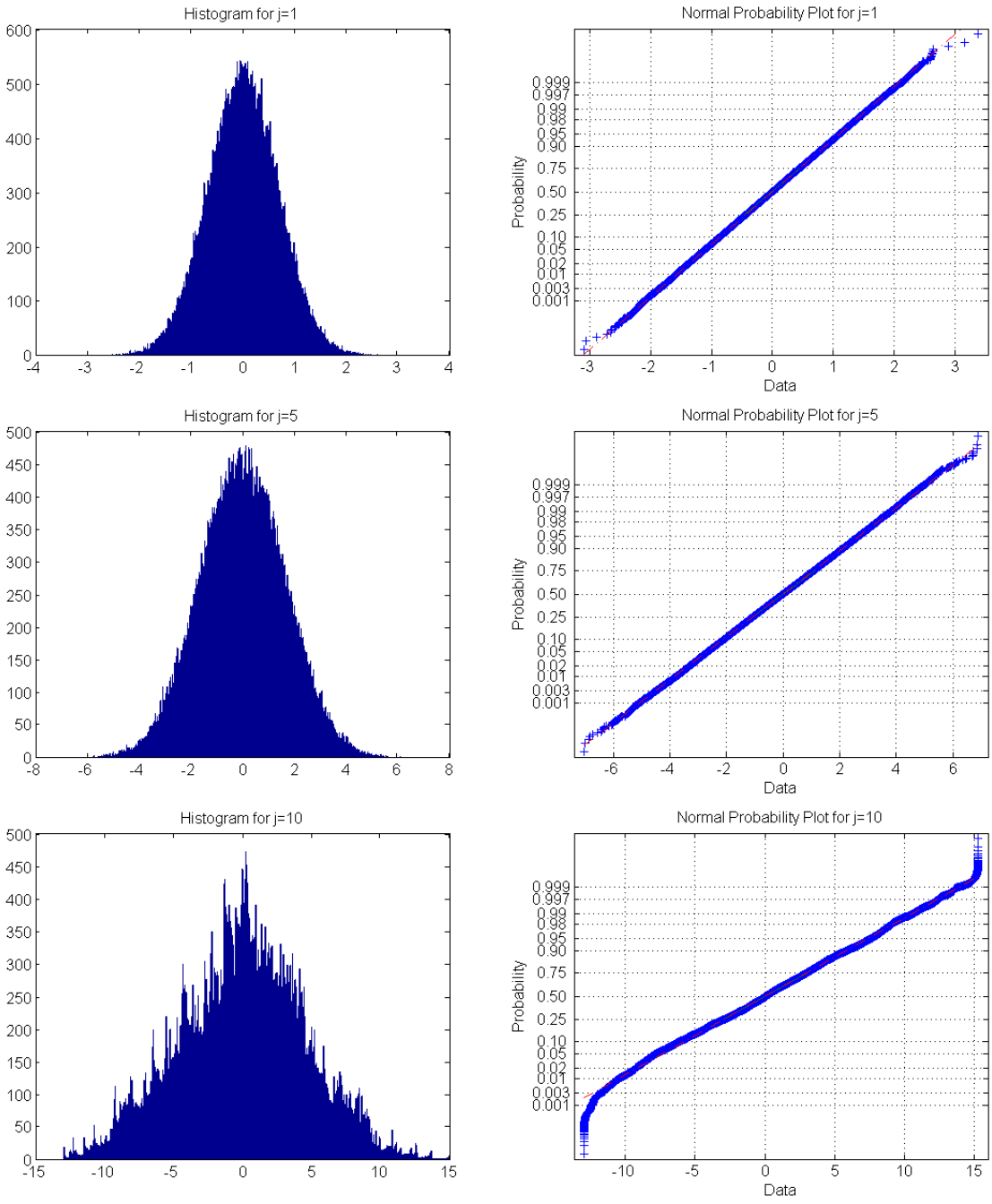

Figure 7.2: Histograms (*left*) and probability plots for graphical normality testing (*right*) of the MODWT coefficients of the FGN trace with $H = 0.8$ when analyzed with `db3`. From top to bottom, the analysis for the coefficients at scales $j = 1$, $j = 5$ and $j = 10$.

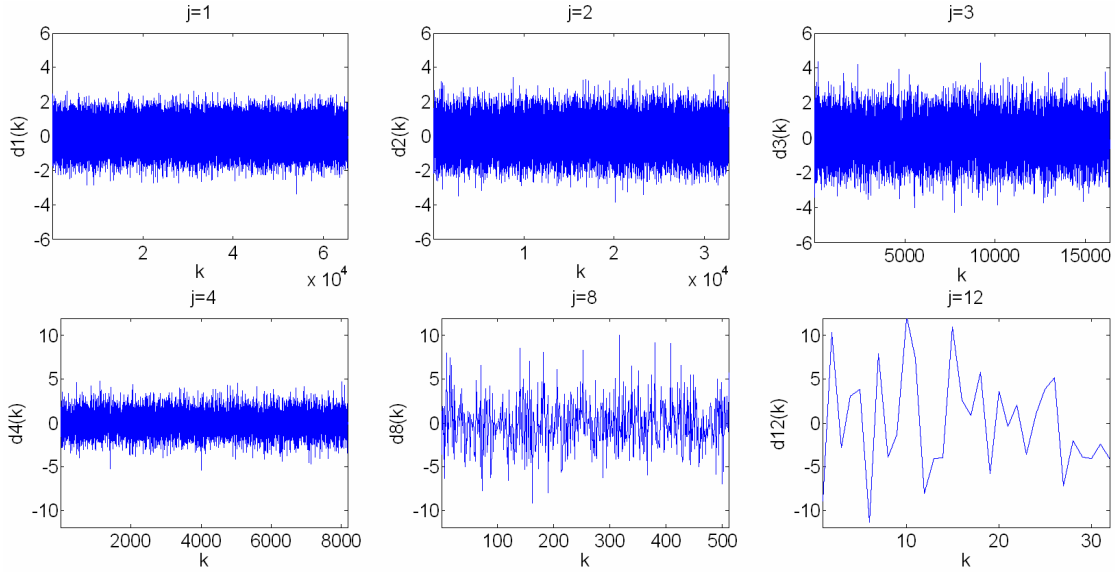

Figure 7.3: MODWT decomposition of the trace of Figure 5.14, for $j = 1, 2, 3, 4, 8$ and 12, using `db1` wavelet. Note the difference in the units of the y axis.

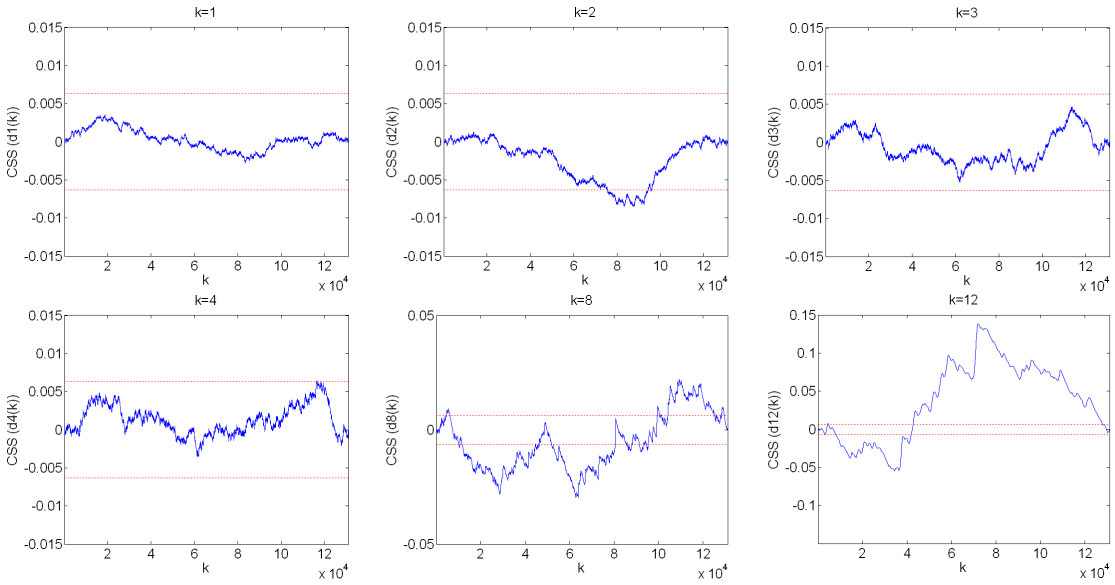

Figure 7.4: CSS statistic applied to the wavelet details of the trace shown in Figure 5.14, for $j = 1, 2, 3, 4, 8$ and 12, using `db1` wavelet at the 99% significance level. The associated critical levels $\pm\frac{1.628}{\sqrt{T/2}}$ are depicted as red horizontal lines. Notice the difference in the units of the y axis.

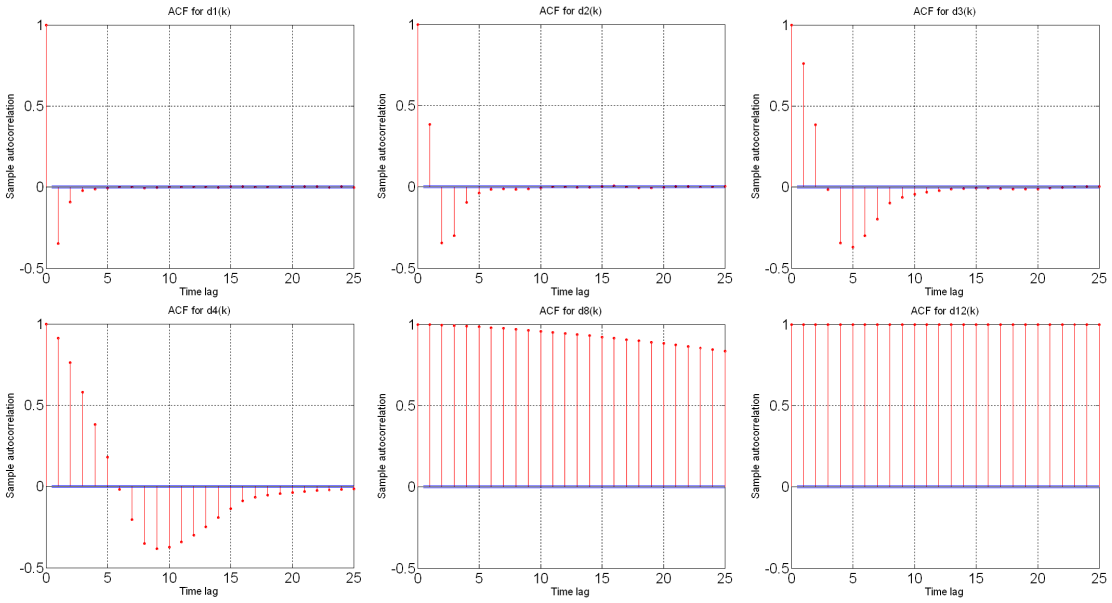

Figure 7.5: Sample autocorrelation of the MODWT detail coefficients of the trace of Figure 5.14 ($j = 1, 2, 3, 4, 8, 12$), with the 95% confidence intervals (horizontal lines).

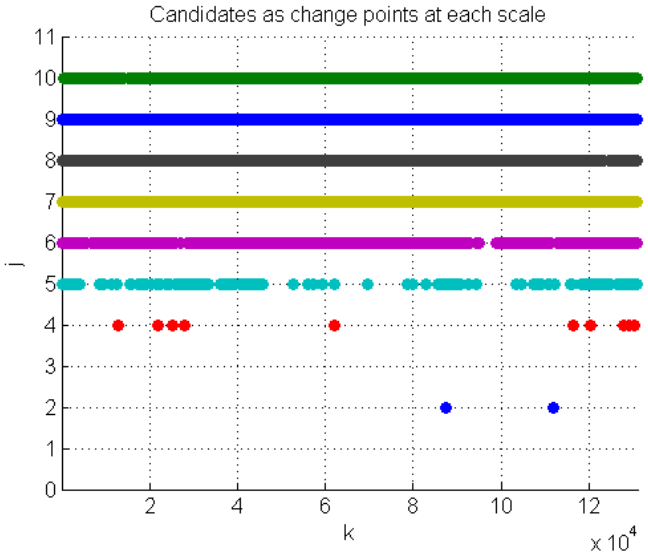

Figure 7.6: Change point diagram for the MODWT-ICSS method when applied to the FGN trace with H=0.8, at the 99% significance level, with `db1` wavelet.

one based in the Equivalent Degrees of Freedom (EDOF) concept, and another that consisted of modifying the values of α and n_{min}.

7.3.1 The failure of the EDOF approach

We tried to attack the problem by using a concept of equivalent degrees of freedom (EDOF) which may be helpful for wavelet-based variance estimation in case of correlation due to, for example, using MODWT. First, recall from Section 4.4 that the position of the change point of a series with length n is estimated by the SIC algorithm by $\hat{k}$ such that

$$SIC(\hat{k}) = min_{1<k<n-1} SIC(k) \tag{7.1}$$

where

$$SIC(n) = n\ \log 2\pi + n\ \log \hat{\sigma}^2 + n + \log n \tag{7.2}$$

$$SIC(k) = n\ \log 2\pi + k\ \log \hat{\sigma_1}^2 + (n-k)\ \log \hat{\sigma_2}^2 + n + 2\ \log n,\ 1 < k < n \tag{7.3}$$

are the expressions of the SIC statistic under the null hypothesis H_0 (no change is present) and the alternative hypothesis H_1 (a change is present), respectively.

Following [PW02], the wavelet variance for a stationary process $x(m),\ n = 1 \ldots n$ is defined for scale j as

$$\nu^2(j) = var(\tilde{d}(j,k)) \tag{7.4}$$

where $\tilde{d}(j,k))$ denotes the MODWT coefficients obtained by filtering $x(m)$. A fundamental property of wavelet variance is its ability to decompose original process variance across scales:

$$\sum_{j=1}^{J} \nu^2(j) = \sigma_x^2 \tag{7.5}$$

An unbiased estimator of $\nu^2(j)$ is given by

$$\hat{\nu}^2(j) = \frac{1}{M_j} \sum_{k=L_j}^{n} \tilde{d}^2(j,k) \tag{7.6}$$

where $M_j = n - L_j + 1$ and $L_j = (2^j - 1)(L-1) + 1$ for the wavelet filter of length L. If $\tilde{d}(j,k)$ had a Gaussian distribution and were uncorrelated, their sum of squares would follow (with a proper renormalization) a chi-square distribution with M_j degrees of freedom. Due to the correlation, however, some modification is required, for example by adjusting a degree of freedom and the approximation

$$\frac{\eta \hat{\nu}^2(j)}{\nu^2(j)} =^d \chi_\eta^2 \tag{7.7}$$

with η being a constant known as the *equivalent degrees of freedom* (EDOF) of the Chi-square distribution χ_η^2. One of the possible estimators of η is

$$\hat{\eta} = \frac{M_j \hat{\nu}^4(j)}{\hat{A}_j} \tag{7.8}$$

where

$$\hat{A}_j = \frac{\hat{\nu}^4(j)}{2} + \sum_{t=1}^{M_j-1} \hat{s}^2_{j,t} \tag{7.9}$$

and $\hat{s}^2_{j,t}$ is the usual biased estimator of autocovariance of $\tilde{d}(j,k)$.

In [Whi98] one can find a study where in the ICSS test statistics, used for the variance change point detection, sample size n was substituted for η in order to compensate the correlation resulting from using MODWT. We tried to employ a similar technique by rewriting Equation 7.1 to the form

$$S\hat{I}C(n) = \hat{\eta}\log 2\pi + \hat{\eta}\log\hat{\sigma}^2 + \hat{\eta} + \log\hat{\eta} \tag{7.10}$$

and

$$S\hat{I}C(k) = \hat{\eta}\log 2\pi + k\log\hat{\sigma_1}^2 + (\hat{\eta}-k)\log\hat{\sigma_2}^2 + \hat{\eta} + 2\log\hat{\eta} \tag{7.11}$$

Note that we kept n in estimating $\hat{\sigma}^2$, $\hat{\sigma_1}^2$, and $\hat{\sigma_2}^2$ when using their logarithms because change point k may appear at positions $2,\ldots,n-2$. We conducted a small numerical study (100 independent repetitions) using the Fractional Gaussian Noise generator described in [Sto], choosing a sample of size n=2048, Hurst parameter 0.6 and 0.9 in the segments of length 1000 and 1048 respectively. The variance of the whole process was kept constant and equal to 1, α was set to 0.1 (for the values of 0.05 and 0.01 the outcomes did not differ) and the Daubechies `db4` wavelet was used (no improvement observed with the `symlet4` filter). The results obtained were unappealing: on the scales $j = 3\ldots6$ we observed more or less uniformly distributed spurious change points instead of the desirable sharp peak around the true value, as can be shown in Figure 7.7.

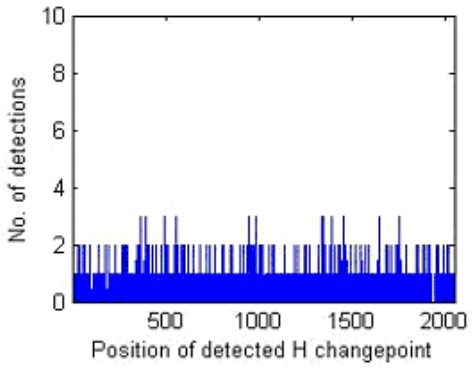

Figure 7.7: Failure of the SIC-EDOF-MODWT approach: real changepoint at 1000, maximum number of detections could be 100, scale $j = 6$.

A possible reason for the failure of this method could be using critical values of the distribution (derived under independence assumption in [CG97]) for the correlated data. Another explanation might be the bias (due to the correlation) of the estimators of variance in SIC calculations. Perhaps a more sophisticated modification of SIC method than ours would give better results.

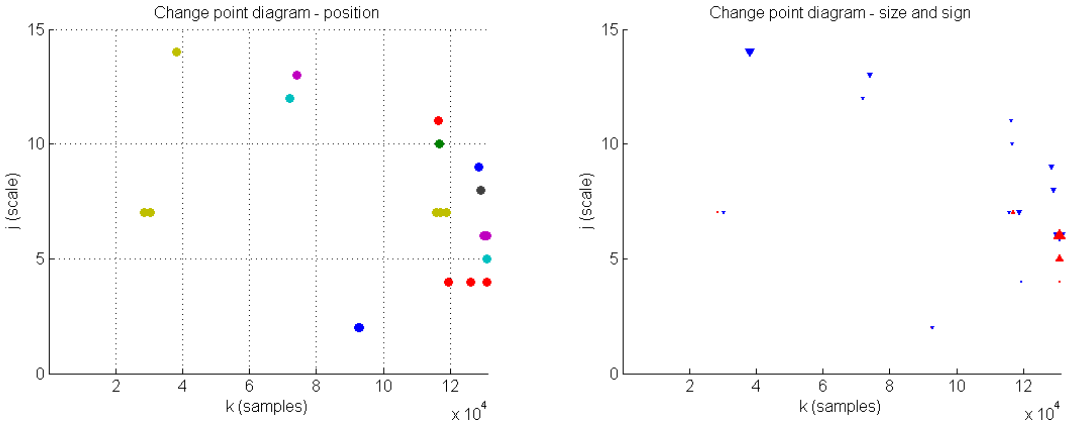

Figure 7.8: *Left*: Change point diagram obtained with the MODWT-SIC method using the parameters shown in Table 7.1. *Right*: Size and sign of the change points.

7.3.2 Modification of α and n_{min}

We found that we could still get some good results by tuning appropriately the two free parameters α (significance level) and n_{min} (the minimum segment size). The values of these parameters are modified with the scale index j; the significance level is decreased by 10, while n_{min} is increased by a factor of two. This has been done empirically, and although we know that the approach can be statistically meaningless, it gives acceptable results. However, we only applied this approach to the SIC statistic, due to the availability of an analytic expression for the critical level of the statistic. The ICSS critical level can only be obtained by MonteCarlo simulation.

Figure 7.8 shows the MODWT-SIC change point diagram for the first 14 scales of the FGN trace, with the `db1` wavelet and the parameters shown in Table 7.1. With this choice of parameters we *clean* the diagram, obtaining only some spurious points comparable to those found in Section 6.4 for the DWT-based analysis of the FGN trace, if we skip the points detected close to the end of the trace, which we consider border effects. The use of a higher order wavelet (such as `db3`) does not significatively affect either the quantity or the position of the spurious changes. Finally, the introduction of a mean change in the middle of the series (such as the case studied in Section 6.5 for the DWT) returns results very similar to those obtained with the constant, zero mean.

Scale j	1	2	3	4	5	6	7	8 . . . 14
α	1E-2	1E-2	1E-3	1E-4	1E-5	1E-6	1E-7	1E-7
n_{min}	1024	2048	4096	8192	16384	32768	65536	65536

Table 7.1: Parameters used in the MODWT-SIC analysis of the pure FGN trace with $H = 0.8$.

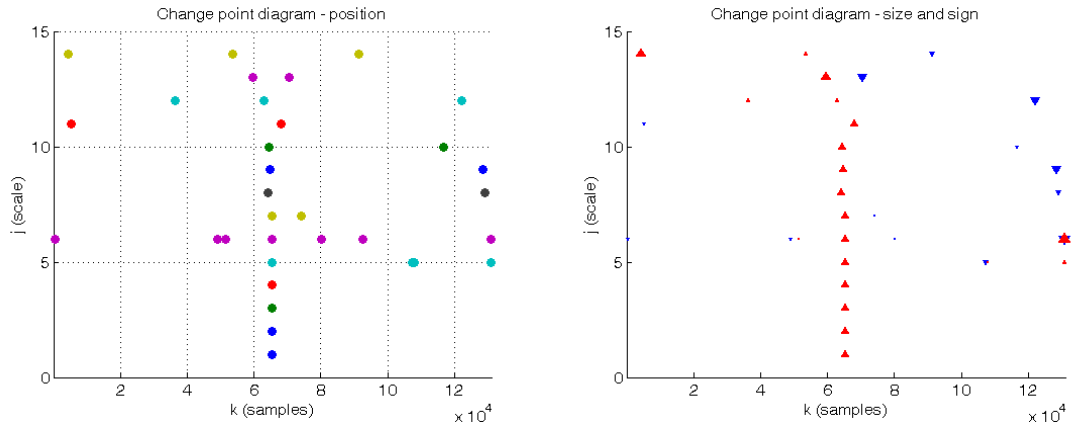

Figure 7.9: *Left*: Change point diagram obtained with the MODWT-SIC method for the variance change case using the parameters shown in Table 7.1 and `db1`. *Right*: Size and sign of the change points.

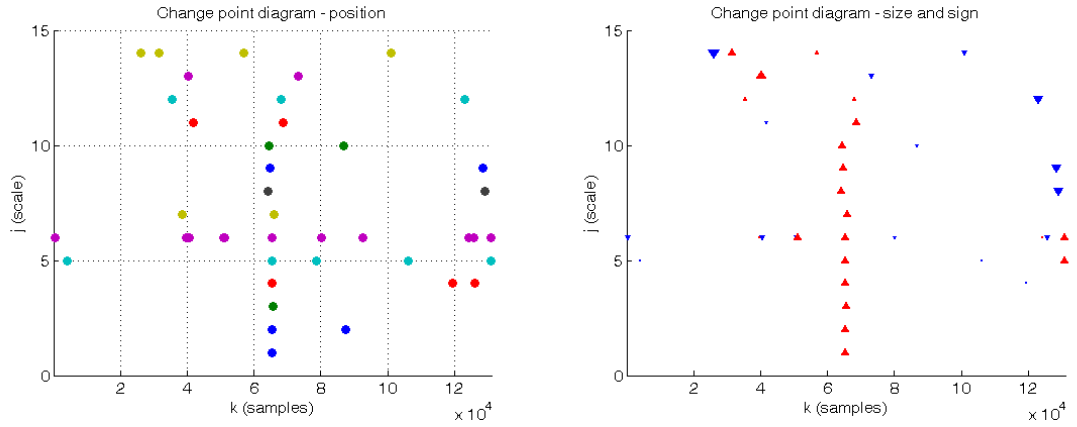

Figure 7.10: *Left*: Change point diagram obtained with the MODWT-SIC method for the variance change case using the parameters shown in Table 7.1 and `db3`. *Right*: Size and sign of the change points.

	quorum (votes)	q=3			q=5		
	resolution (samples)	$r = 5$	$r = 20$	$r = 200$	$r = 5$	$r = 20$	$r = 200$
`db1`	alignments	none	65533	65510	none	none	65510
	absolute error		4 samples	27 samples			27 samples
`db3`	alignments	65534	65534	65520	none	none	65520
	absolute error	3 samples	3 samples	17 samples			17 samples

Table 7.2: Results of the alignment procedure applied to the MODWT-SIC analysis of an FGN trace with a variance change. The alignments position and its absolute error are shown.

7.4 Analysis of the effects of a variance change

We will now discuss the effects of a variance change on the MODWT-SIC algorithm. We use the FGN trace with variances $\sigma_1^2 = 1$ for the first half and $\sigma_2^2 = 4$ for the second. The MODWT-SIC parameters were chosen as shown in Table 7.1. The change points diagram of Figure 7.9, obtained by using the `db1` wavelet, shows that the change point located at $k = 65536$ has been detected at almost all available scales, as expected, together with some spurious points that seem to concentrate at the end of the trace. When the analysis is performed with the `db3` wavelet, as shown in Figure 7.10, the results are similar.

If we compare the results with those found for the DWT-based method in Section 6.6, we find a slight improvement in the detection of the variance change at the higher scales, though more spurious points are also introduced. Nevertheless, once the alignment procedure is applied, the results are comparable with those provided by the DWT-based methods, as shown in Table 7.2.

7.5 Application to a synthetic trace with changes in the Hurst parameter

We face now the detection of changes in the scaling parameter of the series. We begin with the *ideal* case of a transition from $H = 0.5$ to $H = 0.9$. Figures 7.11 and 7.12 show the change point diagrams obtained with the parameters listed in Table 7.1, for the cases of `db1` and `db3`, respectively. Apart from the detection of spurious points (especially in the 6^{th} scale), we highlight the bias towards the lower positions detected at the higher scales, that seems to be more intense for the `db1` wavelet than for `db3`. We have not found a wavelet-based explanation for this behavior, since the MODWT is, by construction, time-invariant. Therefore, we think it has to be related to the residual correlation present at the higher scales. Regarding the influence of the wavelet, `db3` generates many more false points in the middle scales, probably due to their longer associated filters, which increase the effect of the correlation. Comparing the results with those obtained with the DWT in Section 6.7, the conclusion are similar to those found in the variance change case: the MODWT-based method extends the range of change point detection to the higher scales, at the cost of an increase of the false change points.

When the Hurst parameter goes through a much smoother change, such as $H = 0.7 \rightarrow 0.8$, we obtain the change points diagrams shown in Figures 7.13 and 7.14 for `db1` and `db3`, respectively. As expected, in these cases the H transition is harder to detect than the previous case. Comparing the results with those obtained with the DWT-SIC method (shown in Figure 7.15, the MODWT-SIC method provides some more change points at the higher scales, while the DWT-SIC misses them.

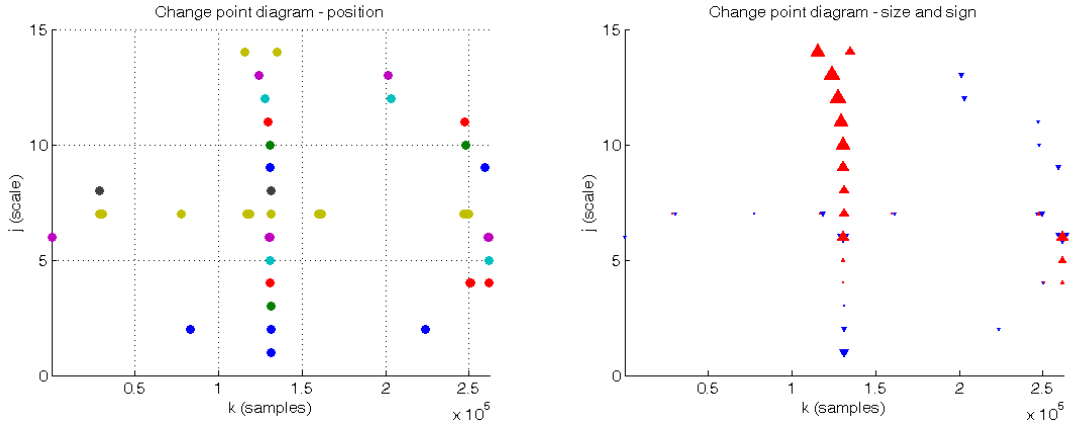

Figure 7.11: *Left*: Change point diagram obtained with the MODWT-SIC method for the $H = 0.5 \rightarrow 0.9$ Hurst parameter change case using the parameters shown in Table 7.1 and `db1`. *Right*: Size and sign.

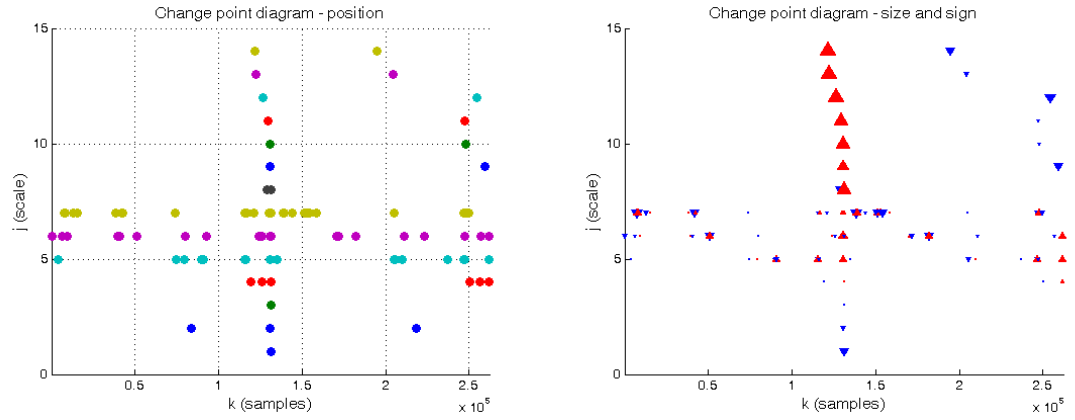

Figure 7.12: *Left*: Change point diagram obtained with the MODWT-SIC method for the $H = 0.5 \rightarrow 0.9$ Hurst parameter change case using the parameters shown in Table 7.1 and `db3`. *Right*: Size and sign.

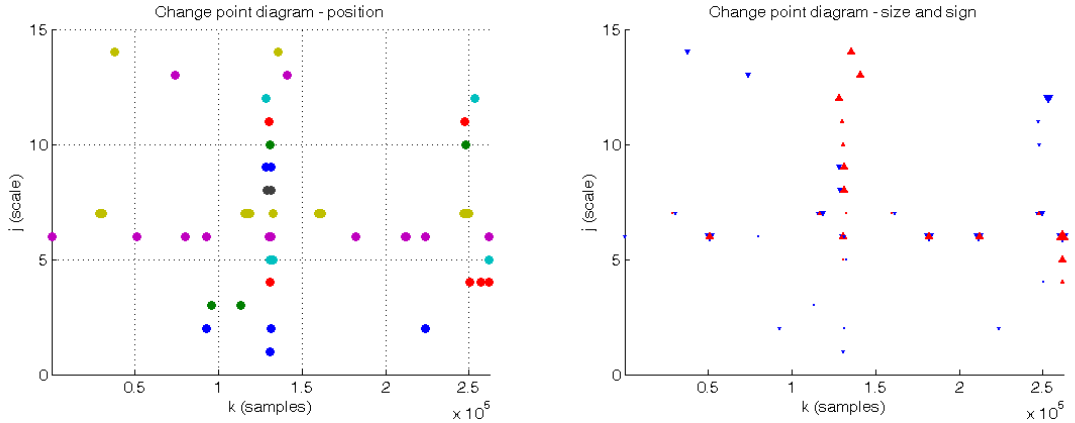

Figure 7.13: *Left*: Change point diagram obtained with the MODWT-SIC method for the $H = 0.7 \rightarrow 0.8$ Hurst parameter change case using the parameters shown in Table 7.1 and `db1`. *Right*: Size and sign.

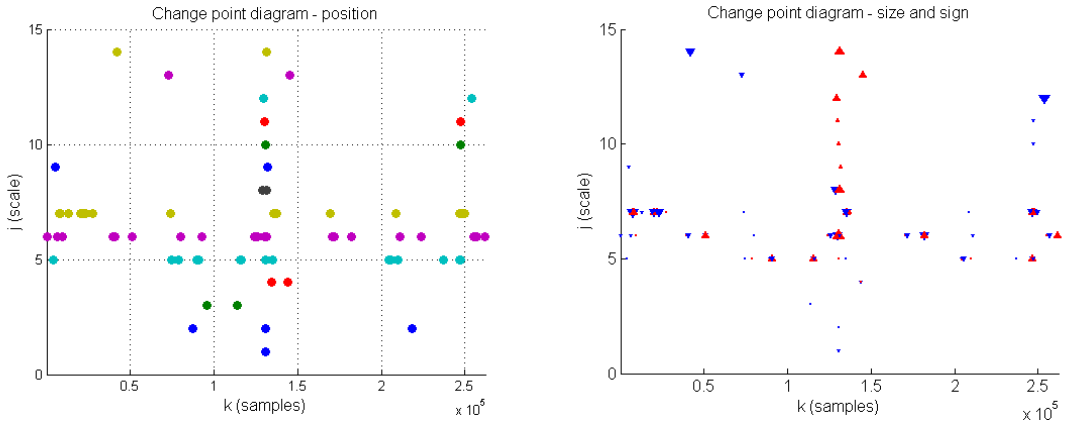

Figure 7.14: *Left*: Change point diagram obtained with the MODWT-SIC method for the $H = 0.7 \rightarrow 0.8$ Hurst parameter change case using the parameters shown in Table 7.1 and `db3`. *Right*: Size and sign of the change points.

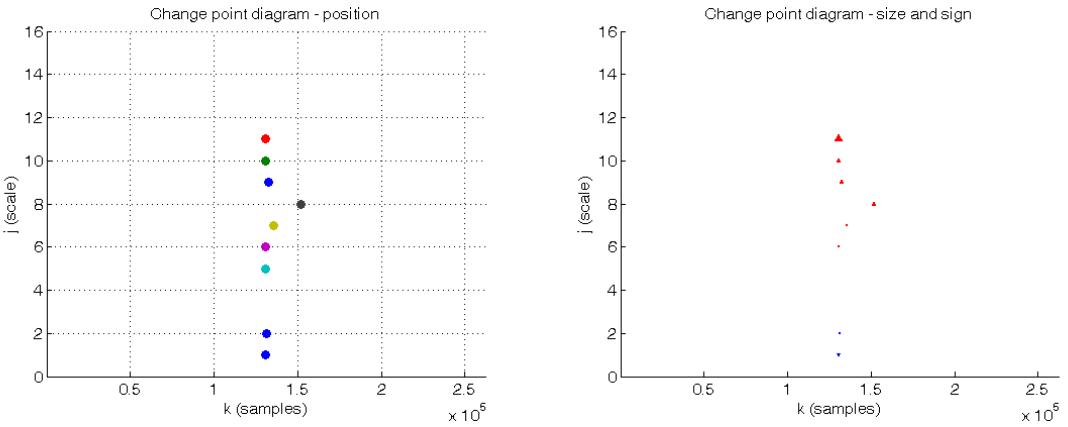

Figure 7.15: *Left*: Change point diagram obtained with the DWT-SIC method for the $H = 0.7 \rightarrow 0.8$ Hurst parameter change case at the 99% significance level, using `db1`. *Right*: Size and sign of the change points.

7.6 Empirical assessment of the power of MODWT-SIC when the Hurst parameter changes

As we did with the DWT-based algorithms, we performed an empirical test for assessing the power of the MODWT-SIC algorithm. The results are shown in Table 7.3 for the cases of a $H = 0.5 \rightarrow 0.9$ and a $H = 0.7 \rightarrow 0.8$ change.

The main conclusions extracted from the data are:

- In all the cases, a lot of spurious changes are detected, though more than 1000 points (the true changes) are detected around the $\pm$ resolution/2 positions margin.
- The higher the wavelet order, the more spurious points.

		$H = 0.5 \rightarrow 0.9$				$H = 0.7 \rightarrow 0.8$			
		db1		db3		db1		db3	
resolution	quorum	Total	*True*	Total	*True*	Total	*True*	Total	*True*
5 smpls	3 chg	35	**16** (44.4%)	17	**6** (35.3%)	3	**0** (0%)	10	(0%)
	4 chg	3	**2** (66.7%)	0	**0**	0	**0**	0	**0**
	5 chg	0	**0**	0	**0**	0	**0**	0	**0**
	6 chg	0	**0**	0	**0**	0	**0**	0	**0**
	7 chg	0	**0**	0	**0**	0	**0**	0	**0**
20 smpls	3 chg	256	**148** (57.8%)	188	**122** (64.9%)	35	**2** (5.7%)	14	**4** (28.6%)
	4 chg	39	**32** (82.1%)	23	**17** (73.9%)	2	**0** (0%)	1	**0** (0%)
	5 chg	3	**2** (66.7%)	1	**1** (100%)	0	**0**	0	**0**
	6 chg	0	**0**	0	**0**	0	**0**	0	**0**
	7 chg	0	**0**	0	**0**	0	**0**	0	**0**
200 smpls	3 chg	1145	**869** (75.9%)	1148	**829** (72.2%)	735	**255** (34.7%)	652	**217** (33.3%)
	4 chg	949	**845** (89.0%)	867	**767** (88.5%)	239	**123** (51.5%)	138	**69** (50.0%)
	5 chg	648	**631** (97.4%)	552	**524** (94.9%)	60	**34** (56.7%)	27	**17** (63.0%)
	6 chg	324	**321** (99.1%)	243	**238** (97.9%)	14	**6** (42.9%)	3	**3** (100%)
	7 chg	102	**102** (100%)	70	**70** (100%)	4	**1** (25%)	1	**1** (100%)
1000 smpls	3 chg	1988	**962** (48.4%)	2855	**958** (33.6%)	2008	**699** (34.8%)	2934	**606** (20.6%)
	4 chg	1242	**962** (77.5%)	1362	**958** (70.3%)	1125	**665** (59.1%)	1248	**557** (44.6%)
	5 chg	1062	**962** (90.6%)	1063	**945** (88.9%)	727	**523** (71.9%)	683	**427** (62.5%)
	6 chg	825	**780** (94.5%)	829	**789** (95.2%)	439	**350** (79.7%)	356	**250** (70.2%)
	7 chg	498	**474** (95.2%)	539	**523** (97.0%)	240	**202** (84.2%)	172	**130** (75.6%)
5000 smpls	3 chg	3608	**1000** (27.8%)	6047	**996** (26.6%)	3961	**930** (23.5%)	6437	**908** (14.1%)
	4 chg	2299	**1000** (43.5%)	3778	**996** (26.4%)	2384	**930** (39.0%)	4013	**908** (22.6%)
	5 chg	1644	**1000** (60.8%)	2368	**996** (42.1%)	1599	**930** (58.2%)	2513	**907** (36.1%)
	6 chg	1322	**1000** (75.6%)	1665	**996** (59.8%)	1244	**928** (74.6%)	1694	**906** (53.5%)
	7 chg	1169	**1000** (85.5%)	1296	**996** (76.6%)	1067	**913** (85.6%)	1274	**897** (70.4%)
10000 smpls	3 chg	4132	**997** (24.1%)	6859	**986** (14.4%)	4772	**974** (20.4%)	7149	**951** (13.3%)
	4 chg	2935	**997** (34.0%)	5071	**986** (19.4%)	3195	**974** (30.5%)	5453	**951** (17.4%)
	5 chg	2125	**997** (46.9%)	3635	**986** (27.1%)	2250	**974** (43.3%)	3967	**951** (24.0%)
	6 chg	1675	**996** (59.5%)	2631	**986** (37.5%)	1645	**974** (59.2%)	2807	**951** (33.9%)
	7 chg	1373	**996** (72.5%)	1948	**986** (50.6%)	1318	**972** (73.7%)	2029	**951** (46.9%)

Table 7.3: Results for the alignment detection algorithm for a trace whose H parameter changes from 0.5 to 0.9 and from 0.7 to 0.8, analyzed with the MODWT-SIC. The *true* changes are defined as those around ± resolution/2 positions around the true change position.

- The combination of a resolution of 1000 samples and a quorum of 5 alignments seems to be a good value for the tradeoff between position accuracy and number of changes detected (more than 90%), for the $H = 0.5 \rightarrow 0.9$ case when analyzed with `db1`. The `db3` analysis asks for a higher quorum.
- In the $H = 0.7 \rightarrow 0.8$ case the tradeoff is attained with a higher resolution value: 5000 samples, with a quorum of up to 7 aligned changes. The `db3` analysis, as expected, finds fewer true changes, but for all the values of quorum at the 5000 samples resolution, the number of detected true changes is over 90%.
- When compared with the DWT-SIC results shown in Section 6.7.4, the MODWT-SIC (with the particular parameter set chosen empirically in Section 7.3.2) performs slightly worse in terms of a higher resolution / lower quorum for attaining the same number of true change points. On the other hand, the MODWT-SIC is more consistent when it comes to comparing the results obtained with the higher values of the quorum; i.e., the changes detected with MODWT-SIC do not decay with the increase of the quorum as fast as those found with

DWT-SIC. This results from the higher amount of samples at each scale generated by the MODWT. In some sense, we could say that the changes detected with the MODWT are *safer* or more trustable than those found with the DWT, since the former approach performs well even at a high level of exigency (a high quorum).

7.7 Application to real traffic traces

Let us now detect the change points of the `BC-pAug89` and `BC-OctExt` traces (described in Sections 3.2.10 and 5.4) with the MODWT-based algorithms.

7.7.1 Statistical distribution of the MODWT coefficients

We check the normality assumption of the coefficients, as we previously did in the synthetic FGN traces case in Section 7.2. Figure 7.16 and 7.17 show the histograms and probability plots for the `BC-pAug89` and `BC-OctExt` traces (aggregated at 10 ms and 1 s, respectively) when analyzed with `db3`[2].

The results are similar to those obtained with the DWT (see Figures 6.40 and 6.41). The lower scales are clearly Laplacian for both traces, far from the Gaussian case. For the `BC-pAug89` trace the distribution comes closer and closer to Gaussian as the scale increases, while `BC-OctExt` keeps its Laplacian shape even at the higher scales.

	$j=1$	$j=5$	$j=10$		$j=1$	$j=4$	$j=8$
`BC-pAug89`	0.32	1.01	1.27	`BC-OctExt`	0.32	0.39	0.50

Table 7.4: GGD shape parameter of the MODWT decomposition of the Bellcore traces.

Table 7.4 presents the estimation of the shape parameter of the distributions shown in the previous figures. The results are very similar to those obtained with the DWT. `BC-OctExt` is *sharper-than-Laplacian*, while the higher scales of the `BC-pAug89` trace fall in the zone between Laplacian and Gaussian.

7.7.2 Segmentation of the Bellcore traces

`BC-pAug89` trace

We start our segmentation tests in real traffic traces by comparing the MODWT-SIC with the DWT-SIC using the most similar parameters we can. The values of n_{min} and the significance α are the same as those shown in Table 7.1.

Figure 7.18 shows the MODWT-SIC segmentation with `db3` wavelet, a resolution of 5000 samples and a 4 change quorum, comparable to the DWT-SIC segmentation shown in Figure 6.44.

[2]The results obtained with `db1` are similar and have been omitted.

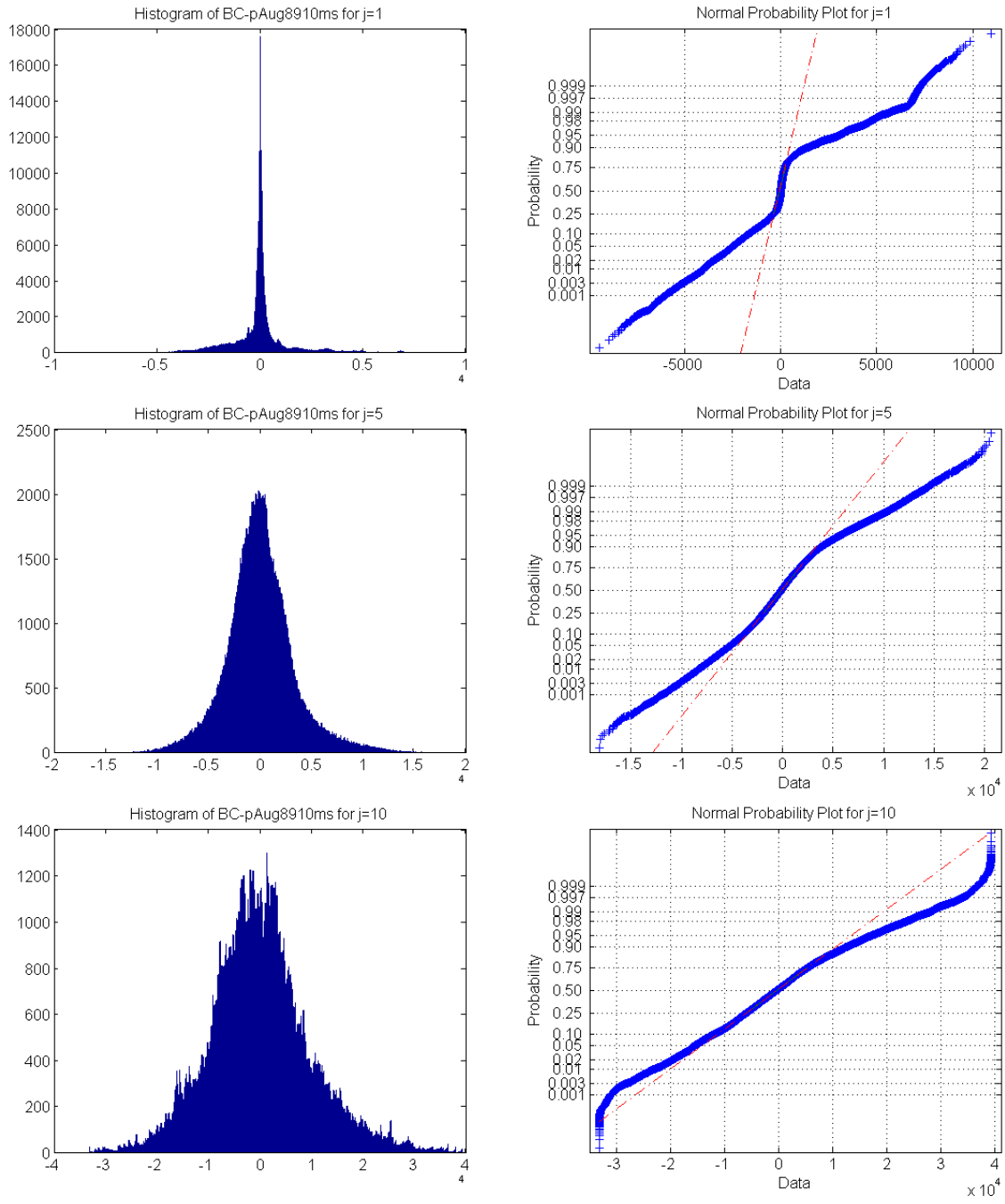

Figure 7.16: Histograms (*left*) and probability plots for graphical normality testing (*right*) of the MODWT coefficients of the `BC-pAug89` trace (aggregated at 10 ms) when analyzed with `db3`. From top to bottom, the analysis for the coefficients at scales $j = 1$, $j = 5$ and $j = 10$.

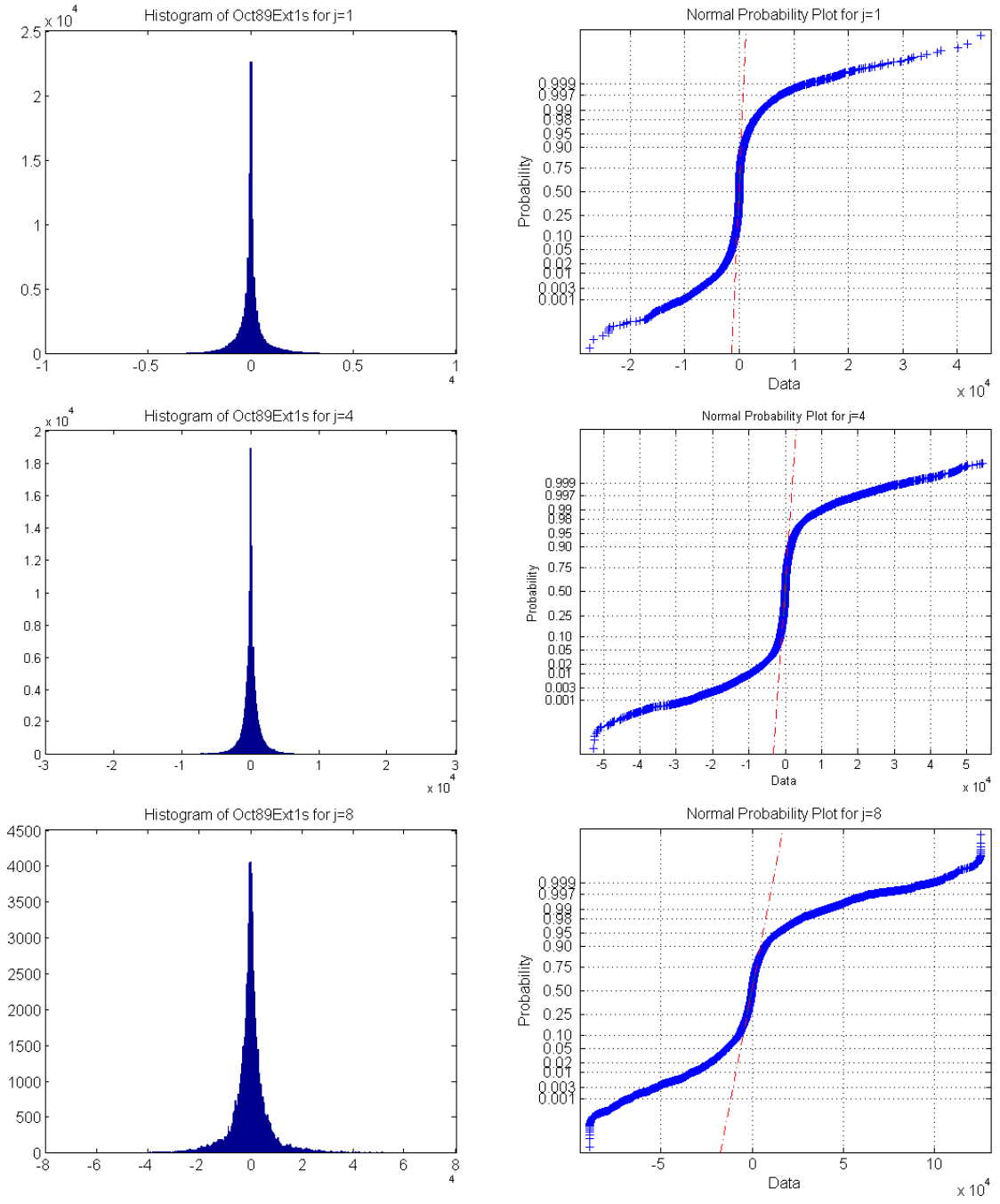

Figure 7.17: Histograms (*left*) and probability plots for graphical normality testing (*right*) of the MODWT coefficients of the `BC-OctExt` trace (aggregated at 1 s) when analyzed with `db3`. From top to bottom, the analysis for the coefficients at scales $j = 1$, $j = 4$ and $j = 8$.

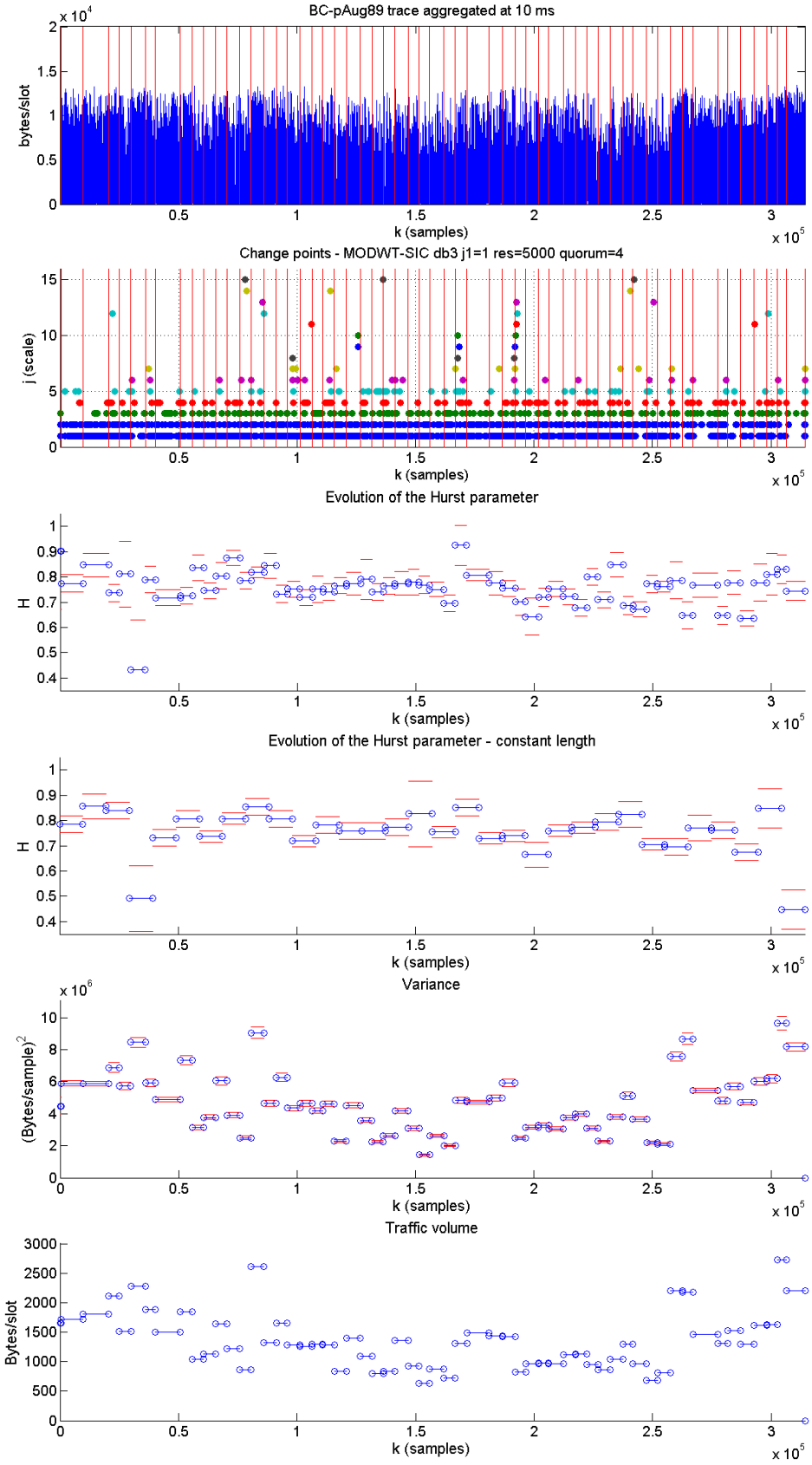

Figure 7.18: Segmentation of the `BC-pAug89` trace, with the MODWT-SIC method, `db3`, resolution = 5000, quorum = 4.

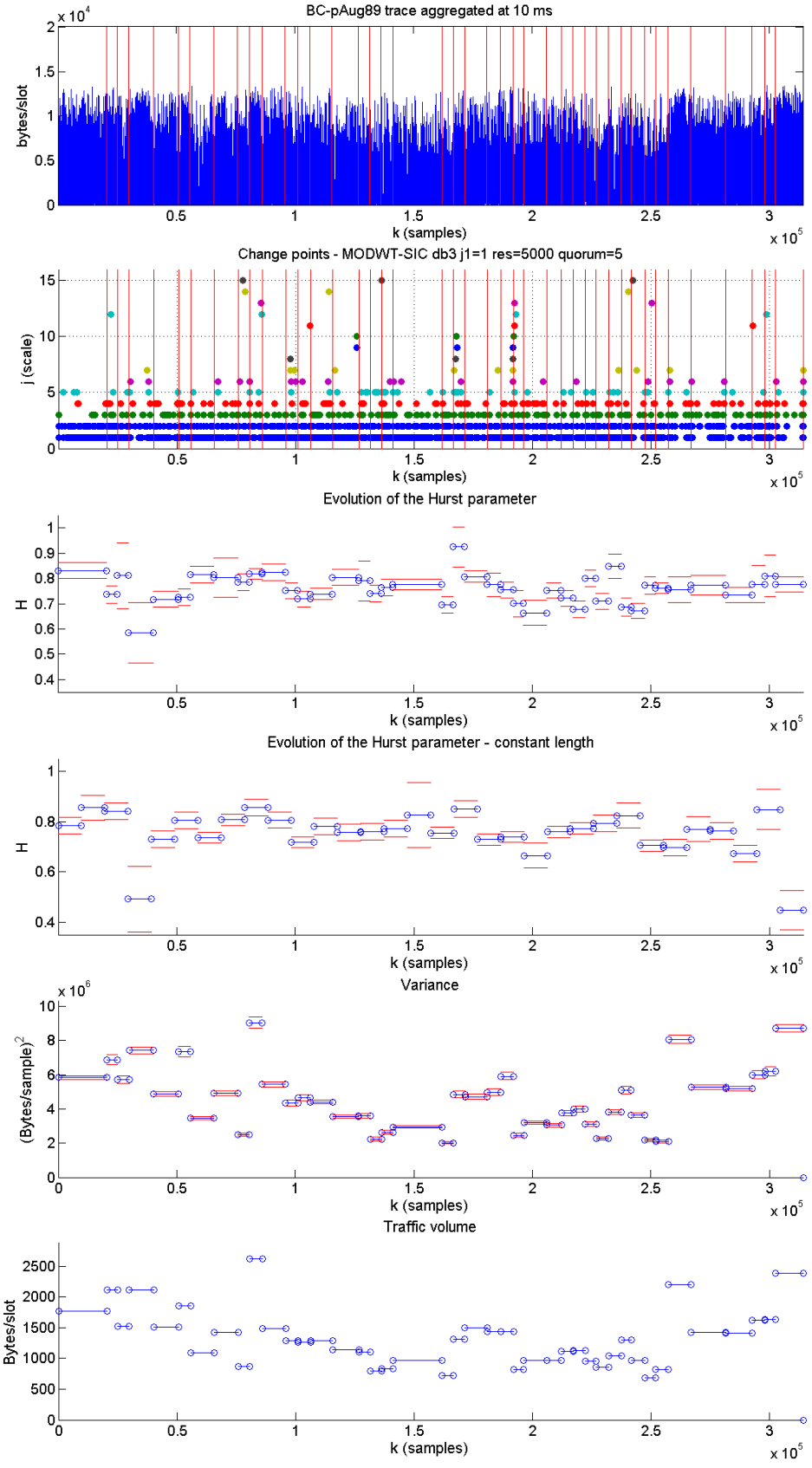

Figure 7.19: Segmentation of the `BC-pAug89` trace, with the MODWT-SIC method, `db3`, resolution = 5000, quorum = 5.

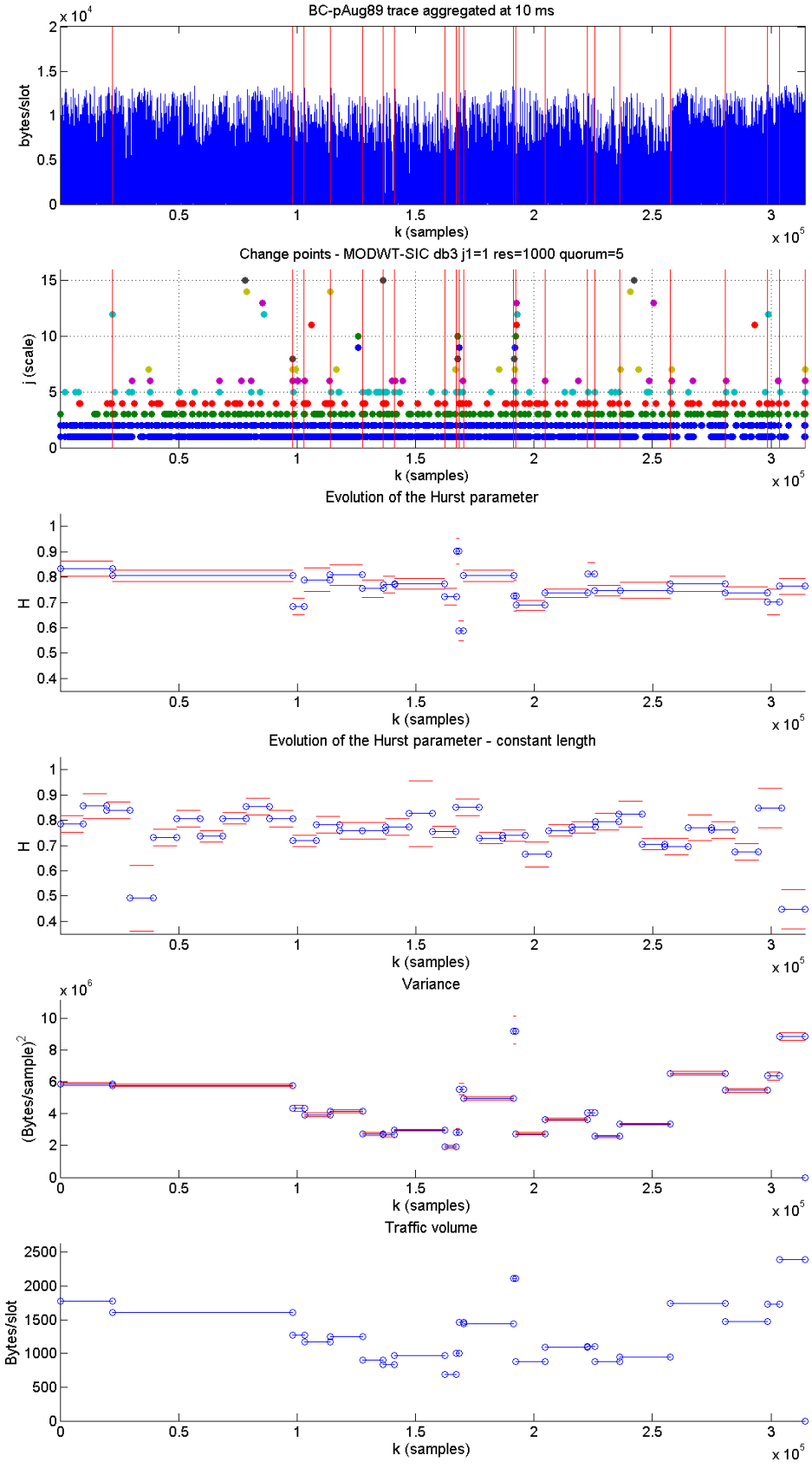

Figure 7.20: Segmentation of the `BC-pAug89` trace, with the MODWT-SIC method, `db3`, resolution = 1000, quorum = 5.

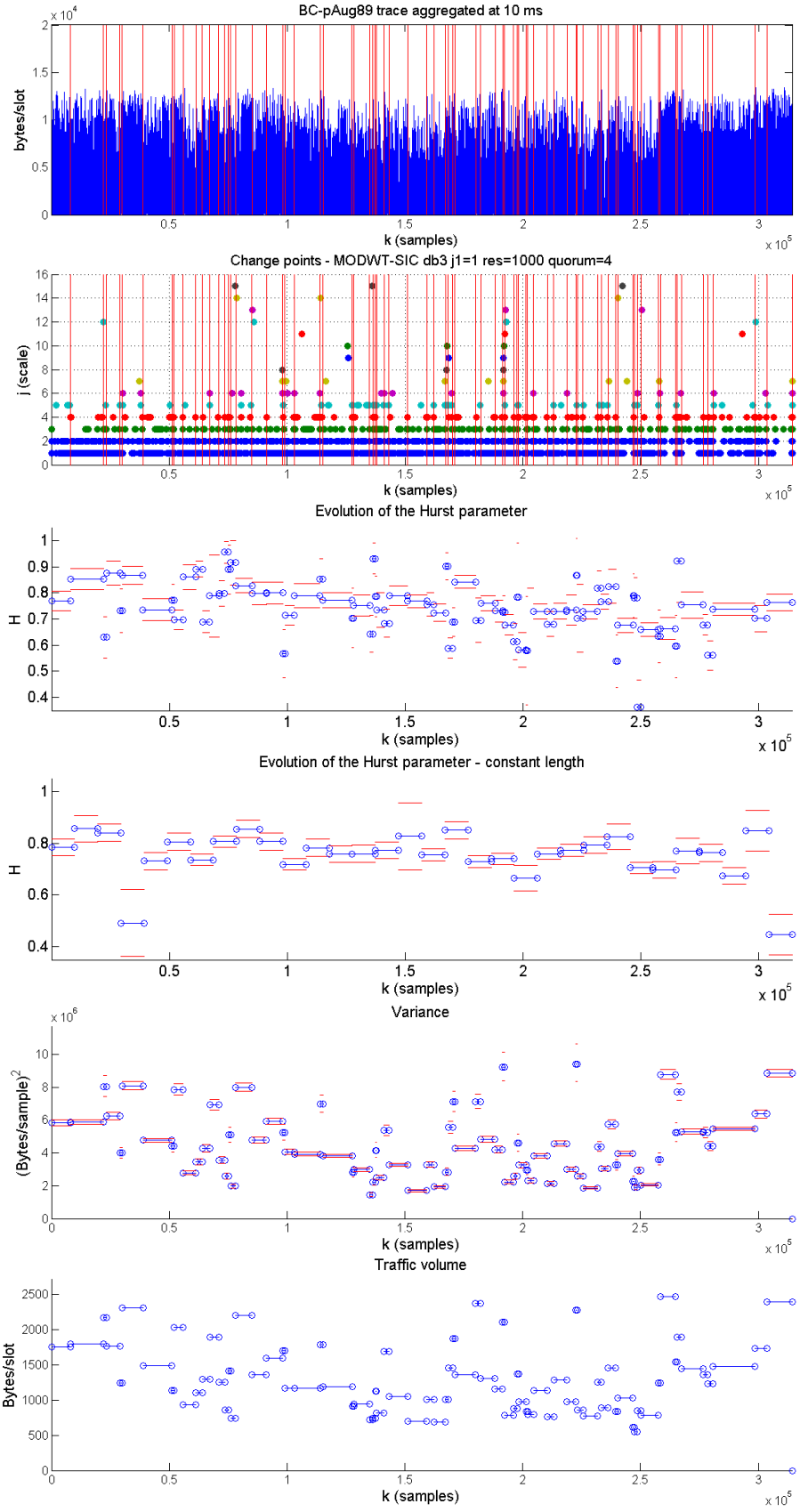

Figure 7.21: Segmentation of the `BC-pAug89` trace, with the MODWT-SIC method, `db3`, resolution = 1000, quorum = 4.

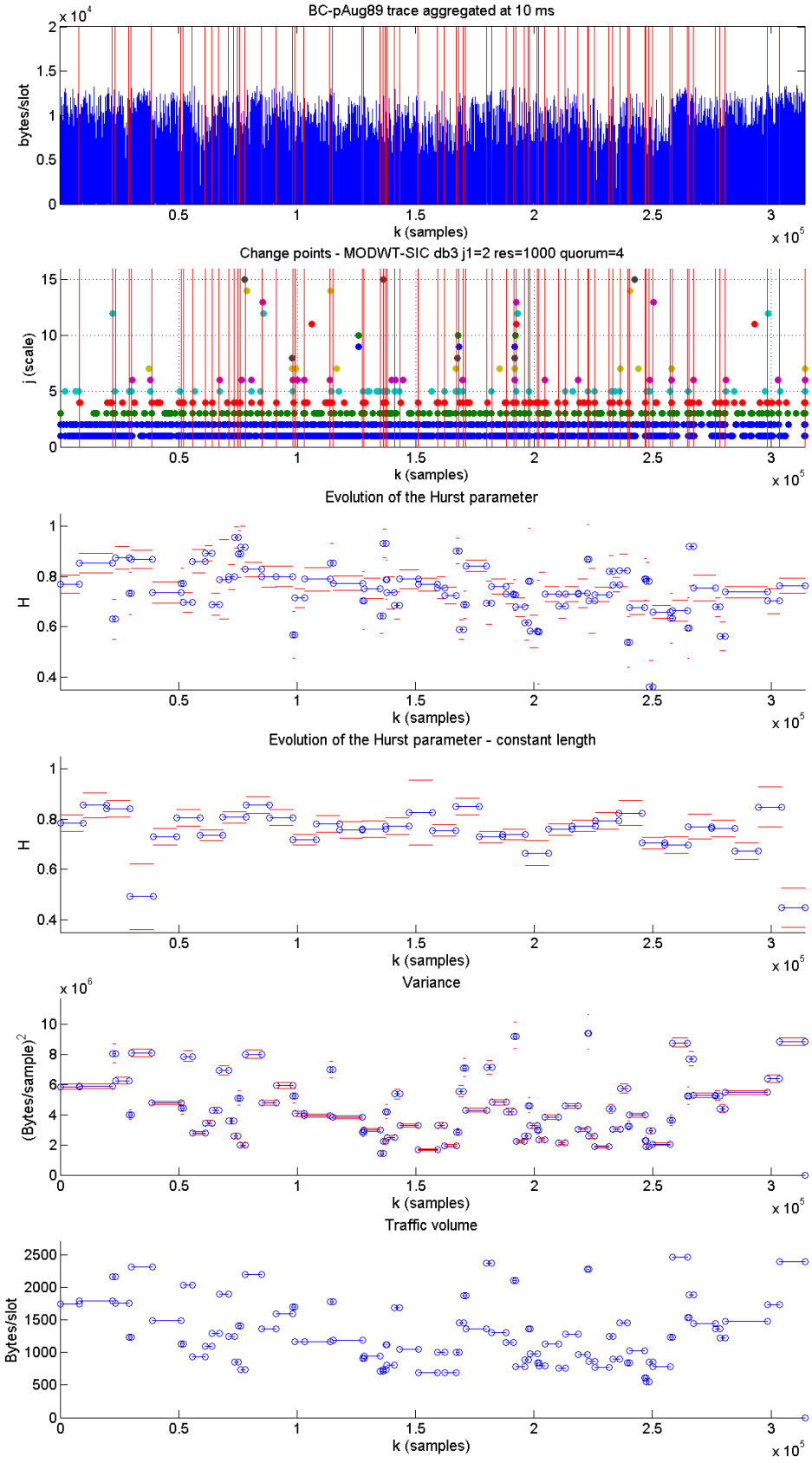

Figure 7.22: Segmentation of the `BC-pAug89` trace, with the MODWT-SIC method, `db3`, resolution = 1000, $j_1 = 2$, quorum = 4.

Surprisingly, and contrary to the results found for the synthetic FGN traces[3], MODWT-SIC returns many more change points than those found by DWT-SIC. Our first hypothesis for explaining this behavior was the coefficient distribution but, apart from the fact that at the higher scales DWT generates much fewer samples, the distributions are quite similar (compare Figures 6.40 and 7.16, and the GGD shape parameters in Tables 6.21 and 7.4) though the MODWT tends to be close to the Gaussian distribution. Furthermore, the increased resolution of the MODWT method should influence in the same way both the synthetic and real traces. Therefore, the only interpretation we find for this behavior is that the choice of the α and n_{min} parameters is specially favorable to the detection of change points at the higher scales.

Since the analysis returns a change practically every 5000 samples (coinciding with the resolution parameter), we tried with an increase in the quorum exigency to 5. Figure 7.19 shows the results: a much clearer picture of the segments, which is highly coherent with the constant-length H estimator (i.e., those constant-length segments whose confidence intervals overlap are detected as H-constant segments, such as the example of the region $0 < k < 20000$). Since the resolution still seems too high, we reduced it to 1000 samples with a quorum of 4 (Figure 7.21) and 5 changes (Figure 7.20). The $q = 5$ case does not track properly the evolution of H; for example, the $H \sim 0.5$ region around $k = 35000$ (actually, the $0.7 < H < 0.85$ regions at the beginning of the trace are detected as just two segments with $H \sim 0.8$. The analysis with a $q = 4$ quorum returns a very fragmented trace with a high number of segments whose confidence intervals are very large, thus diminishing the method's reliability. Finally, Figure 7.22 presents a $q = 4$ analysis where the first *noisy* scale has been omitted. The results are almost identical to those found when the scale was taken into account, and therefore the problem does not come from the highest frequency, short-time fluctuations.

To sum up, though the ranges of values of H and the general shape of the H segments are coherent with those found in Chapter 6 and with the mean value of $H \sim 0.8$, the correct parameter combination remains to be found. The lack of statistical reliability of the empirical choice of n_{min} and α discourages us from performing a systematic study of the best combination of the resolution and quorum values.

`BC-OctExt` trace

Figures 7.23 and 7.24 show the MODWT-SIC segmentation of the `BC-OctExt` trace with `db3`, a resolution of 1000 samples and a quorum of 4 and 3 scales respectively. In this case the results are quite similar to those found with the DWT-SIC with a higher quorum (see Figure 6.57 for $q = 6$ and Figure 6.56 for $q = 5$), thus confirming the tendency already detected in the analysis of FGN traces. Regarding the Hurst parameter values, we can confirm the high variability of H in the `BC-OctExt` trace, with values in the $0 < H < 1.3$ range.

[3]Recall that the analysis parameters were the same in both analysis.

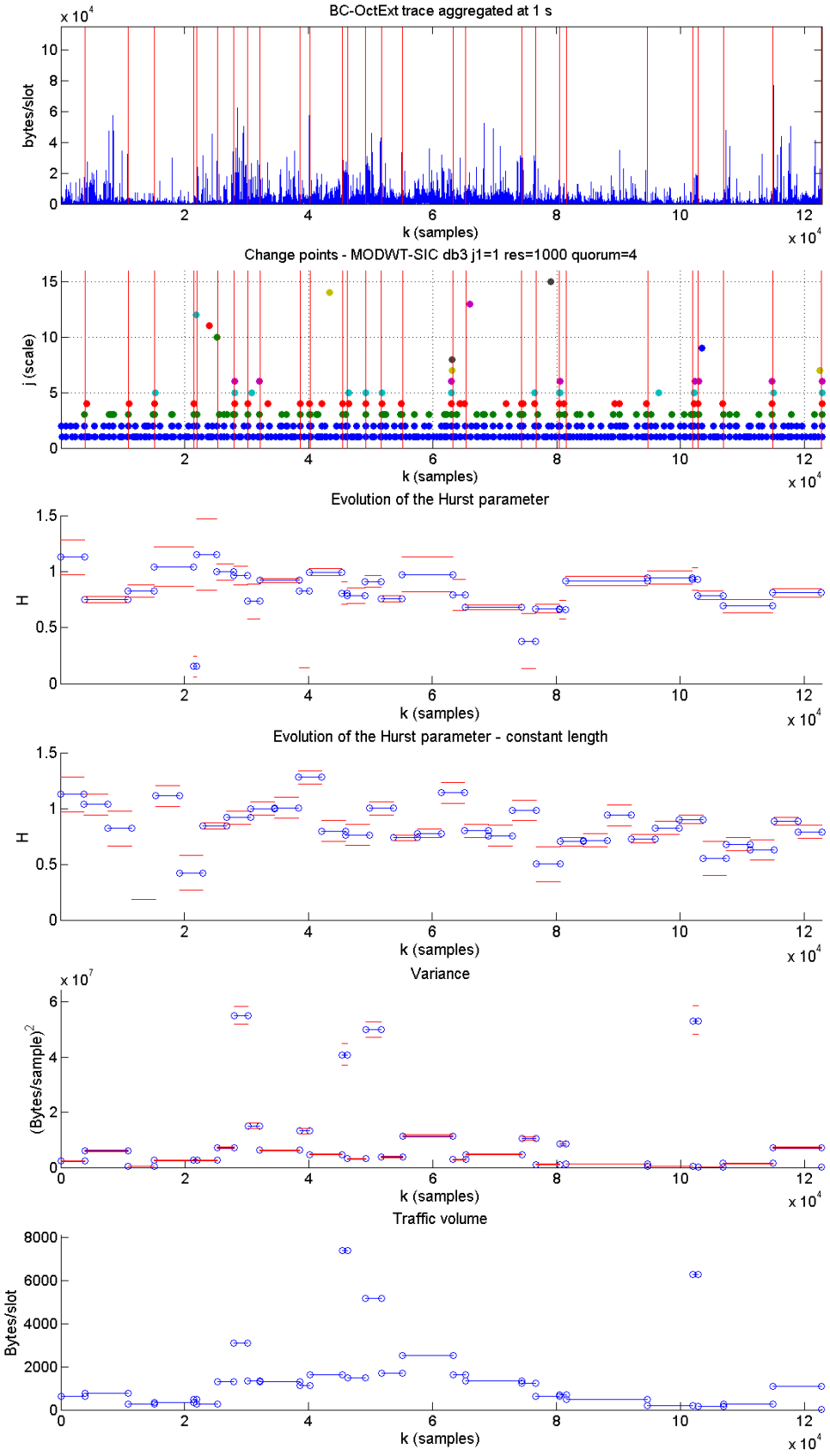

Figure 7.23: Segmentation of the `BC-OctExt` trace, with the MODWT-SIC method, `db3`, resolution = 1000, quorum = 4.

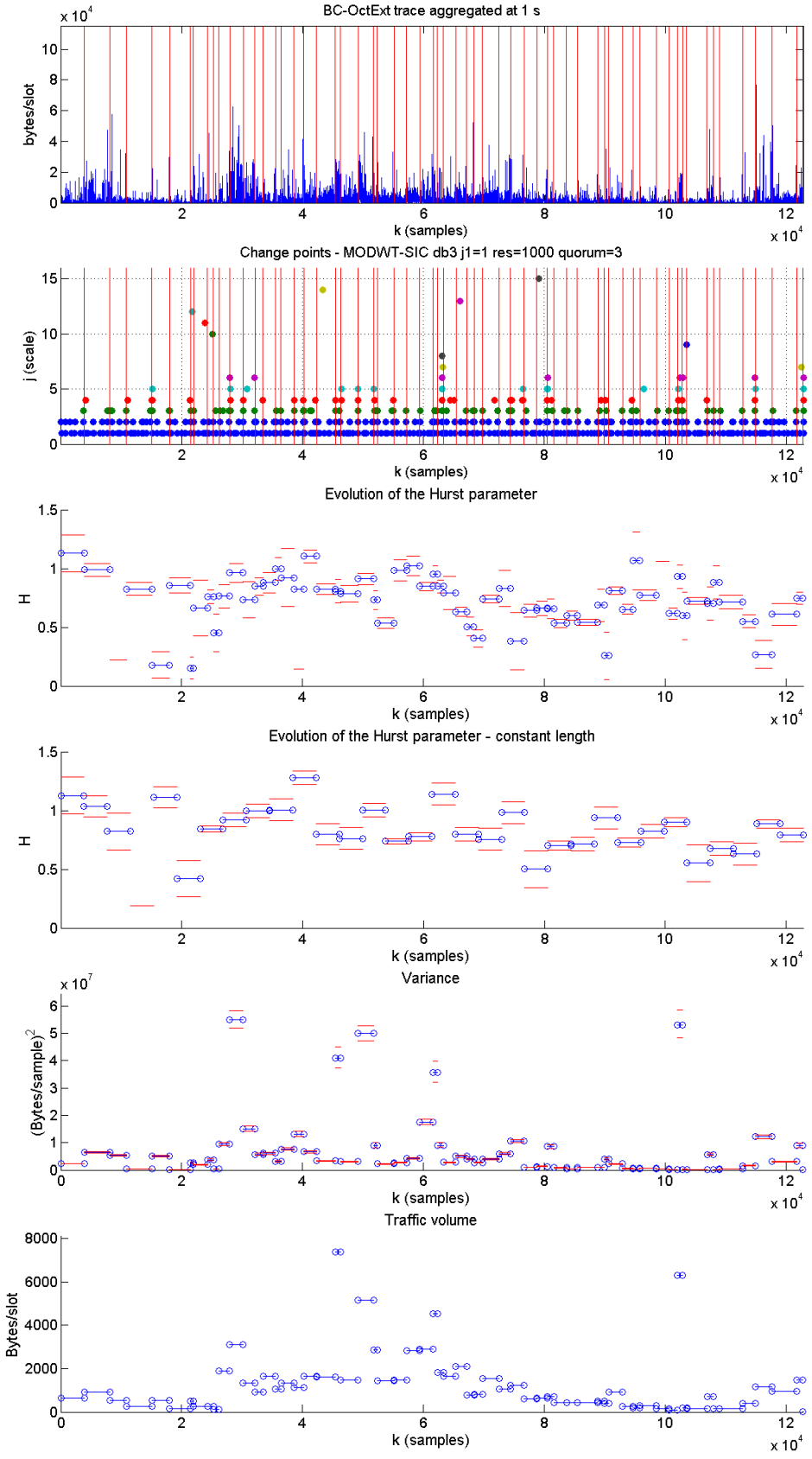

Figure 7.24: Segmentation of the `BC-OctExt` trace, with the MODWT-SIC method, `db3`, resolution = 1000, quorum = 3.

7.8 Summary of the chapter

As shown in the previous sections, the MODWT has the nice properties of time-invariance and is capable of producing the same amount of time samples at any time scale, thus allowing for a much more accurate location of the changes than the DWT. On the other hand there is a price to be paid: the decorrelation properties are lost due to the redundant, non-orthogonal nature of the transform.

Our first attempt to correct the correlation problem, based on the use of the Equivalent Degrees of Freedom (EDOF), clearly failed. But we have shown that good results can still be obtained by forcing some parameters (the significance level of the variance change detector, α, and the minimum length of the H-constant segment, n_{min}, to take unusual values. This of course may decrease the mathematical validity of the procedure and make it somewhat heuristic, but the results show both an agreement with those found with the DWT and in some cases an improvement (for example, when a specific change point is detected across more scales than when DWT was applied). The unusual significance values also prevents us from using the ICSS as a variance change detector, since we do not know the critical values for the statistic[4].

When tested against synthetic FGN traces, MODWT-SIC performs slightly worse than DWT-SIC: it needs a a higher resolution or a lower quorum in order to get similar change point detection capability. On the other hand, MODWT-SIC seems more robust than DWT-SIC in terms of its capability of change detection at the higher scales (and, consequently, performs better than DWT-SIC when the quorum is high), thanks to its higher time resolution at the lower frequencies.

Unfortunately, the lack of statistical reliability of the choice of n_{min} and α makes it difficult to use MODWT in the study of real traces. The empirically-derived values of the analysis parameters that we found to provide coherent segmentation results do not have any solid theoretical base. Therefore we do not recommend the use of MODWT as a replacement for DWT. However, we still look for a transform with a better resolution than DWT while keeping a low correlation. The Dual Tree Wavelet Transform has such properties, as we will see in the next chapter.

[4]The critical level values can be obtained via a MonteCarlo study, but we did not proceed in this direction because the MODWT-SIC results were not so good as to justify the effort

Chapter 8

Alternative segmentation methods (II): DTWT

8.1 Introduction

The main drawback of the MODWT is the high degree of correlation found at the higher time scales (i.e., the lowest frequencies). Though the fine tuning of some parameters produces good results with the MODWT-based methods, the use of other wavelet transforms may improve the performance of the algorithms. While the DWT (no correlation, lack of time resolution, time-variant) and the MODWT (high correlation, best time resolution, time-invariant) are two extremal cases, other transforms fall in the middle and may provide a better trade-off between correlation and time resolution.

The Dual Tree (Complex) Wavelet Transform (DTWT) in one such possibility. As shown in Section 2.8, the DTWT output is *almost* time-invariant, while producing only twice the amount of samples of those generated by a DWT decomposition and with only a slight increase in the correlation of the coefficients. The main challenge of the DTWT is the interpretation of its complex nature: two tree-shaped banks with different filters in both branches, as shown in Figure 2.24. We first treated each branch separately, and then found that the modulus of the complex output gives better results.

As in the previous chapters, this one describes the issues related to the DTWT-based methods and its performance when applied to synthetic LRD and real traffic traces. The DTWT methods are still in an early phase of development. That is why this chapter will provide only the first results of our experiments, but will not provide a complete, systematical study such as the one applied to DWT in Chapter 6 and will remain at an exploratory level, at least enough to compare the DWT and MODWT-based method with the new DTWT algorithms. The results presented in this chapter have been published in [ZR06], [RZ06] and [RZM06].

8.2 DTWT-based methods

8.2.1 Applying the DTWT to our problem

One question to be answered before using the DTWT for multiscale variance change detection is: which set of filters should be used with the DTWT? Recall from Section 2.8 that two different filter sets must be defined: the first-level biorthogonal filters, and the filters above the first level of decomposition. Kingsbury [Kin01] presents and evaluates different solutions for both choices, finding that longer filters improve shift invariance at the cost of a higher complexity. The trade-off calls for using middle-length filters. In our experiments we will use the following: the Near-symmetric $(13, 19)$ tap `near_sym_b` filters for the first level, and the Q-shift $(14, 14)$ tap `qshift_b` filters for the remaining levels. Being that the DTWT is just an alternative to the well founded DWT-based methods, we have not performed an extensive test campaign in order to determine if the other filters provide better results for our change detection algorithms. The aforementioned set of filters has been used in several papers (applied to image coding and also to one-dimensional signals) with good results.

The other important question to be answered is: how can we interpret its double-tree or complex output? We face two different possibilities:

- **Method 1**: Our first idea was to combine the output of both trees by applying the variance change detection points separately, adding all the detected changes, and then modifying (actually, doubling) the quorum to be reached in order to decide that a change is present. Since the intuition behind the DTWT is that each tree works with the even and odd samples respectively, we thought that working in the aforementioned way would allow us to detect the change points that were missed due to the time-variance of the DWT: if a tree missed one particular change point at a certain scale, the other tree would capture it. The results were unsatisfying, as is explained in Section 8.2.2.

- **Method 2**: Our second attempt was to compute the DTWT modulus as the random variable whose variance is studied, and then apply the usual quorum (as we did for the *single tree* transforms). We were reluctant to do so, since Section 2.8.6 showed the presence of a slight residual correlation (usually only at the first or second lag), and we thought we would get worse results. But it turned out to be a better approach than the previous one, as will be described in Section 8.2.3.

In order to provide a preliminary statistical analysis of the performance of the algorithms with both approaches, we designed two different scenarios and analyzed them with the four combinations: DWT-ICSS, DWT-SIC, DTWT-ICSS and DTWT-SIC. Because of its similar phase properties, `symlet4` filter was used for DWT and `near_sym_b` along with `qshift_b` were our choice for DTWT[1].

[1]We are aware of the fact that the use of `symlets` excludes the direct comparison of the results obtained in this test and those already described for the DWT- and MODWT-based methods using Daubechies' wavelets, but since

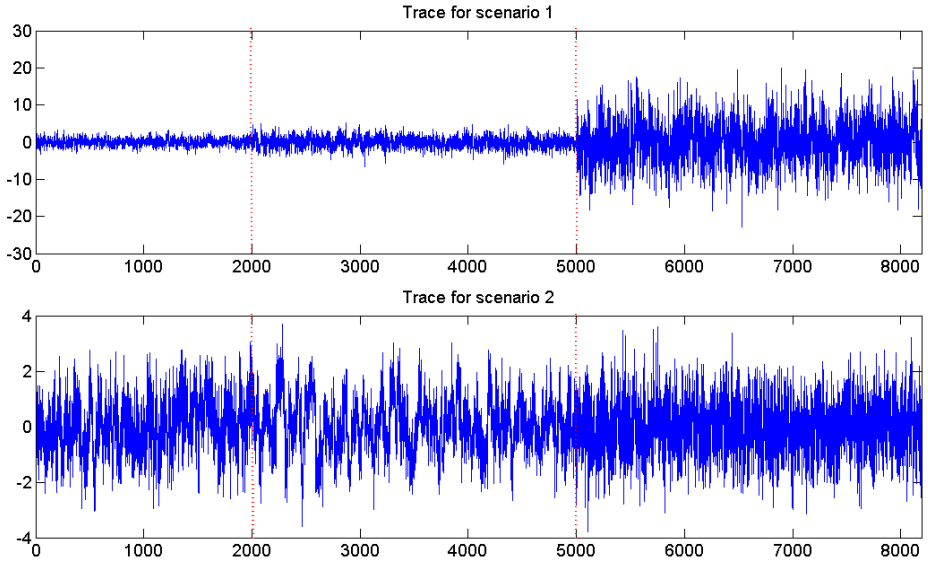

Figure 8.1: *Top*: Realization of a trace for Scenario 1. *Bottom*: Realization of a trace for Scenario 2.

All the tests were performed with fractional Gaussian noise (FGN) segments. We do not provide at this time any test over real traffic, since in this study we are rather more interested in comparing the algorithms from a statistical point of view, and this can only be achieved with synthetic traces where the true change points are known.

The first scenario includes a 8192-sample FGN trace with constant Hurst parameter ($H = 0.7$) and three segments with different variance ratios: 1, 1.5 and 6, providing an example of a smooth variance change at position 2001 and an abrupt one at position 5001 (Figure 8.1, top). The second scenario simulates a constant variance situation with Hurst parameter changes. The trace starts with $H = 0.8$, changes to $H = 0.9$ (smooth change) at 2001, and then changes again to $H = 0.6$ (abrupt change) at sample 5001 (Figure 8.1, bottom). The parameters used in the test were a 95% significance level, decomposition level $J = 6$ scales, $n_{min} = 1$, resolution=256 samples, quorum (1^{st} method) = 6 changes, and quorum (2^{nd} method) = 3 changes.

For each combination of transform and change point detection algorithm we performed a Monte Carlo test with 1000 runs. The Hough transform-based clustering procedure was included in the tests. We defined *successful detection* as the presence of the point in a neighborhood of length 129, centered in the true value of estimated parameters (± 64 points around the true change position). An interval of this size is about 1.5% of the length of the whole time series. The parameters to be studied are the ratio of successfully detected change points (a rough measure of the power of the tests), the mean change position found by the algorithms, its median, the variance around the true change point, and the skewness, also with respect to a true value.

this section is devoted to test the relative performance of the DTWT and DWT techniques, we wanted to be sure they were compared in a fair scenario. Of course, one could argue that even `symlets` are very different from the DTWT filters, and that the comparison is ill-poised, but to our knowledge this is the closest and fairest comparison we can attempt.

8.2.2 Method 1: Adding the changes of both trees

For the first point in the first scenario (Table 8.1, Figure 8.2), we can clearly see the higher power of the DWT-based methods (more points are detected by the DWT than by DTWT). On the other hand DTWT seems to localize the change point with a higher precision than DWT, a trend that was expected due to its time shift invariance. The changes detected with DWT show a higher variance. For the second point both DWT and DTWT methods perform quite well in terms of power, with almost 100% detection ratios, though DWT-ICSS suffers from misalignment (mean and median 24 points away from the true value), a fact that is reflected in a higher variance. All the methods have a comparable rate of asymmetry calculated by the skewness parameter.

For the second scenario (Table 8.2, Figure 8.6), we find a problem with the detection of the first point while still being able to locate the second one. Recall that the transition of the H parameter produces *blind zones*; i.e. after detecting a change at certain scales (in our case for the second point at $j = 1, 2$) no points are observed at higher scales ($j = 3, 4$) and reappear later ($j = 5, 6$). See Figure 8.6 for an illustration of this phenomenon. For the first point, however, a high detection rate was present only for $j = 1, 2$ and to a very small extent on the sixth scale. With a minimum number of votes equal to 3 it resulted in overall lack of power of the test. On a LogScale diagram, the *blind zone* would correspond to the situation when the linear regressions are close to each other. It may be likely that a larger sample size would help in this situation, but even if it were a case, this example shows the unwanted possibility of confusing process variance change with H transition. To distinguish between these situations one should perhaps keep some record of the scales at which points were detected and try to use the *blind zone* concept. Another conclusion we extract from this study is that for the second point DTWT methods suffer again for lack of power (although localization is better and not as dispersed as that of DWT-based methods, as expressed by the lower variance obtained with DTWT).

8.2.3 Method 2: Results with DTWT modulus

The problem we faced in the previous section was the relatively low power of the DTWT-based tests, that was reflected, for example, in the first scenario as change-point detection ratios at the level of 25% for the DTWT while being about 50% for DWT case. A reason for that was the way of treating information provided by both DTWT trees, i.e. combining the sets of detected candidate points and doubling the threshold for deciding which candidates become valid change-points. This was found to be very conservative and has been replaced by the following approach: instead of processing approximation $a_{xa}(j,k)$ and detail $d_{xb}(j,k)$ coefficients from both trees separately, the quantity $z_x(j,k) = \sqrt{a_{xa}^2(j,k) + d_{xb}^2(j,k)}$ is used as an input for ICSS algorithm. $z_x(j,k)$ may be regarded as a modulus of a complex number. Kingsbury [Kin01] provides a description of this approach along with a shift invariance analysis.

Scenarios 1 and 2 were repeated with the new modulus-based approach (Tables 8.1 and 8.2, Figures 8.3 and 8.5). In both cases a significant increase of detection ratio for the DTWT-modulus

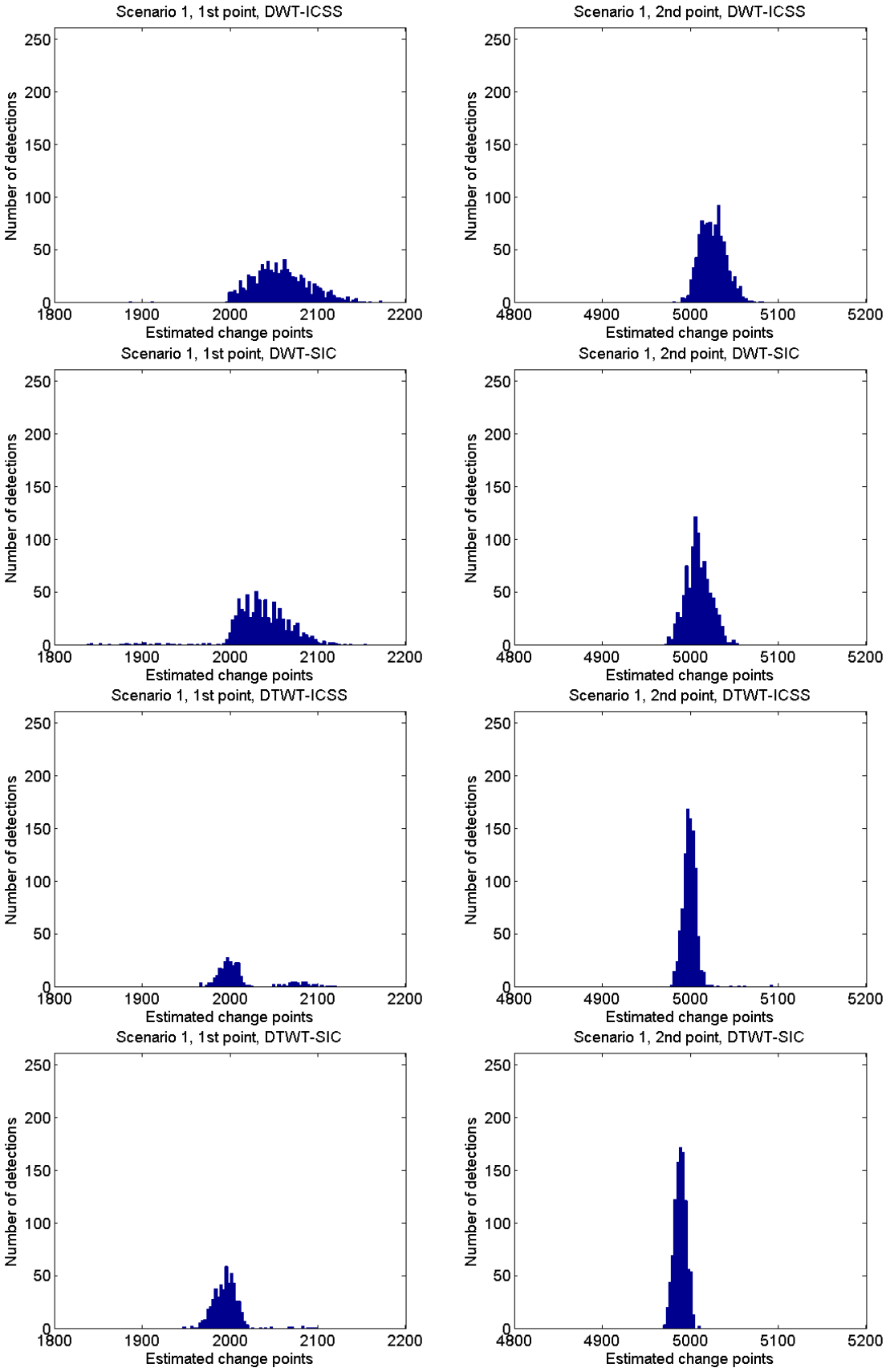

Figure 8.2: Detected changes for Scenario 1, for the first and second changes, for the combinations of DWT/DTWT and ICSS/SIC.

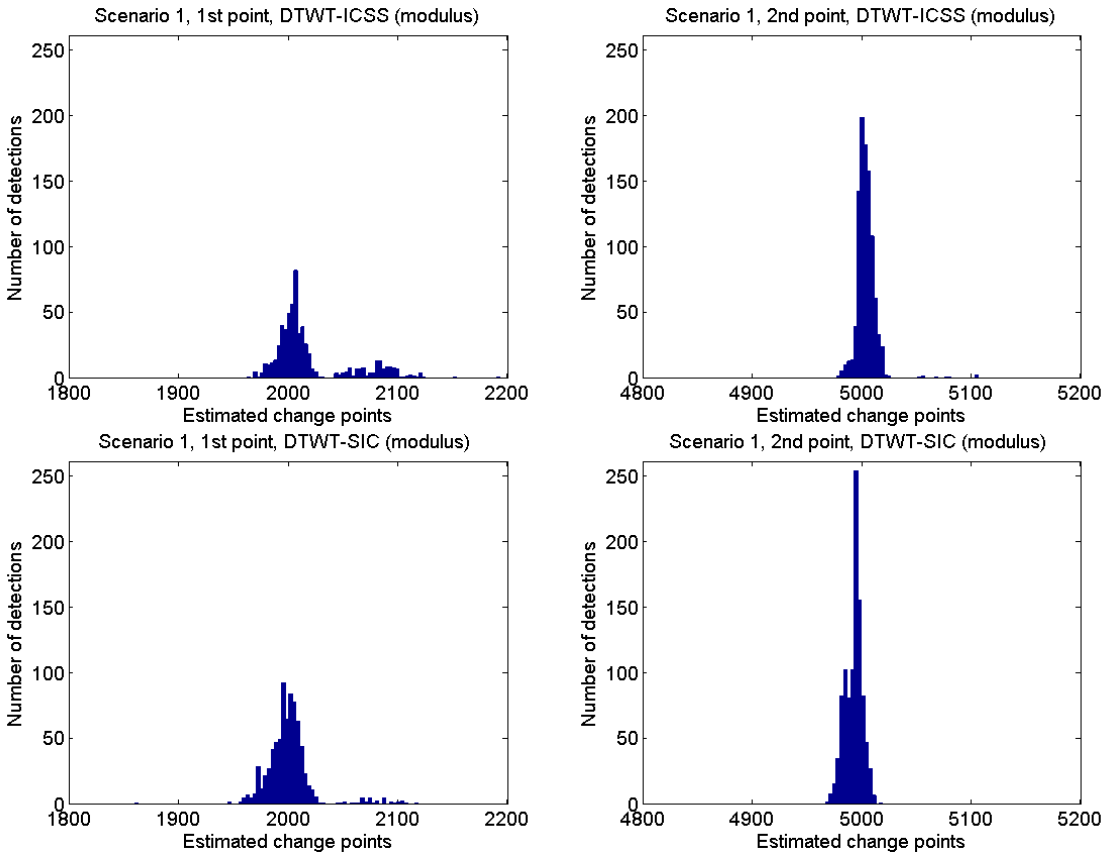

Figure 8.3: Detected changes for Scenario 1 and the DTWT-ICSS and DTWT-SIC with method 2 (modulus).

1^{st} **point**	**Det. ratio**	**Mean**	**Median**	**Variance**	**Skewness**
DWT-ICSS	54.0%	2038	2040	1678	1.20
DWT-SIC	64.9%	2029	2030	1191	1.28
DTWT-ICSS	24.1%	2000	1998	2604	1.48
DTWT-SIC	52.6%	1993	1994	236	-1.31
DTWT-ICSS mod	51.5%	2006	2004	322	1.94
DTWT-SIC mod	74.6%	1997	1999	222	-0.69

2^{nd} **point**	**Det. ratio**	**Mean**	**Median**	**Variance**	**Skewness**
DWT-ICSS	99.3%	5025	5024	778	1.34
DWT-SIC	100.0%	5009	5008	254	1.35
DTWT-ICSS	96.8%	4999	4999	36	0.36
DTWT-SIC	100.0%	4989	4989	192	-1.29
DTWT-ICSS mod	99.2%	5003	5003	57	1.70
DTWT-SIC mod	100.0%	4993	4994	131	-1.51

Table 8.1: Results for Scenario 1 (1000 independent runs, variance change).

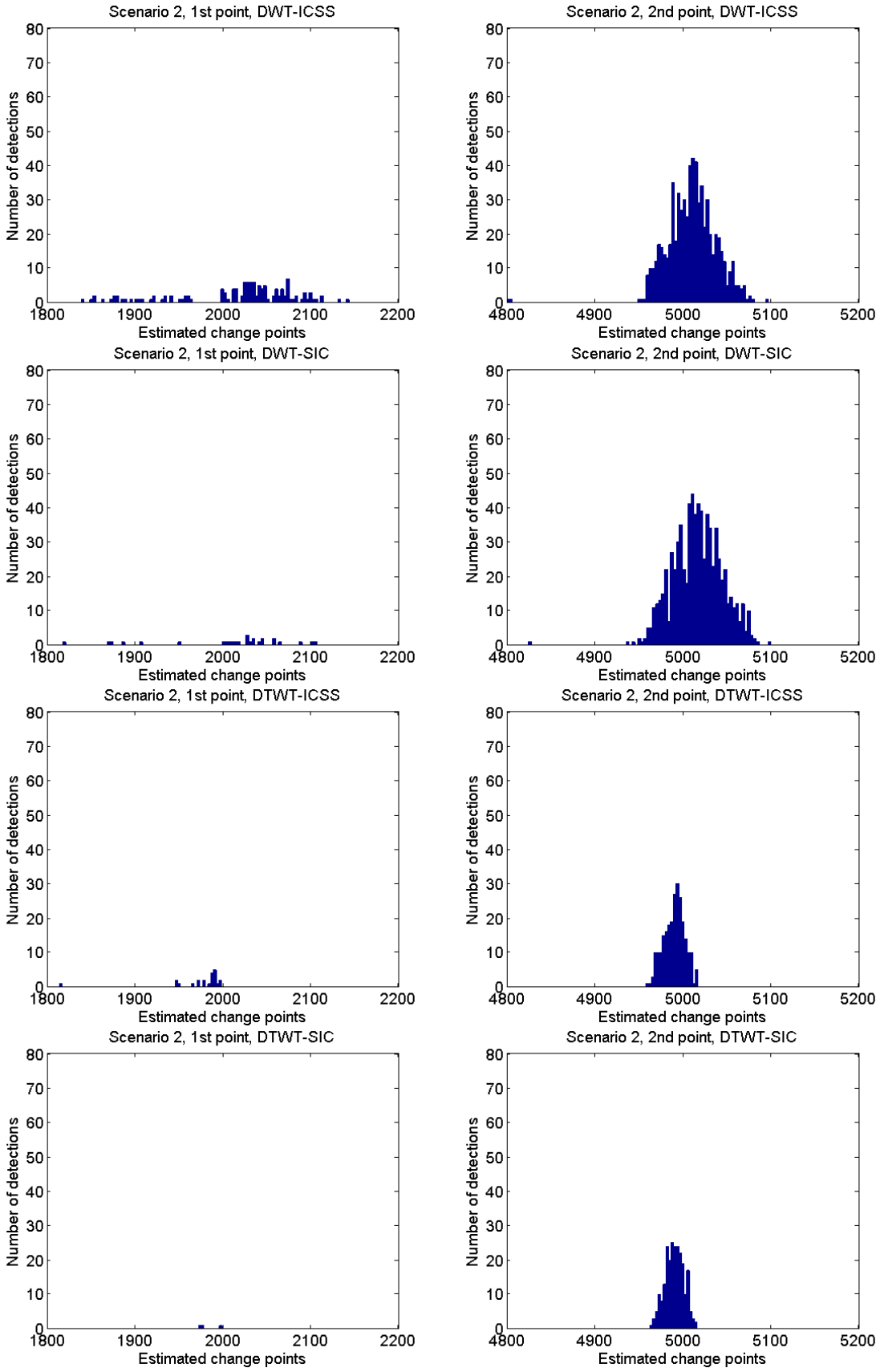

Figure 8.4: Detected changes for Scenario 2, for the first and second changes, for the combinations of DWT/DTWT and ICSS/SIC.

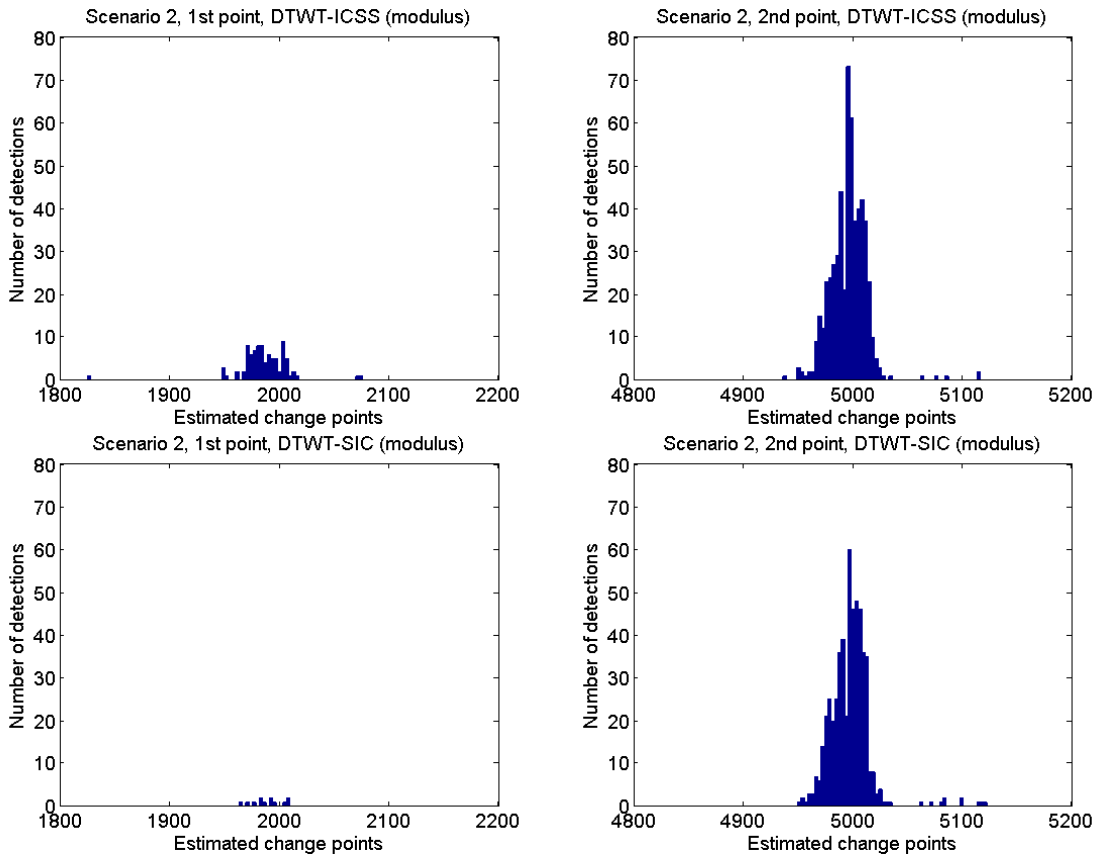

Figure 8.5: Detected changes for Scenario 2 and the DTWT-ICSS and DTWT-SIC with method 2 (modulus).

1^{st} **point**	**Det. ratio**	**Mean**	**Median**	**Variance**	**Skewness**
DWT-ICSS	7.6%	2022	2029	1362	0.71
DWT-SIC	1.8%	2024	2029	1148	0.95
DTWT-ICSS	2.1%	1981	1988	659	-1.63
DTWT-SIC	0.3%	n/a	n/a	n/a	n/a
DTWT-ICSS mod	8.5%	1986	1985	459	-1.51
DTWT-SIC mod	1.2%	1988	1988	347	-1.47

2^{nd} **point**	**Det. ratio**	**Mean**	**Median**	**Variance**	**Skewness**
DWT-ICSS	66.4%	5010	5010	661	0.92
DWT-SIC	72.9%	5014	5015	787	1.06
DTWT-ICSS	24.5%	4990	4990	261	-1.47
DTWT-SIC	23.5%	4991	4991	222	-1.45
DTWT-ICSS mod	54.9%	4995	4997	244	-1.28
DTWT-SIC mod	52.2%	4996	4997	226	-1.16

Table 8.2: Results for Scenario 2 (1000 independent runs, H change). The estimators based on sample size of 3 have been skipped as meaningless.

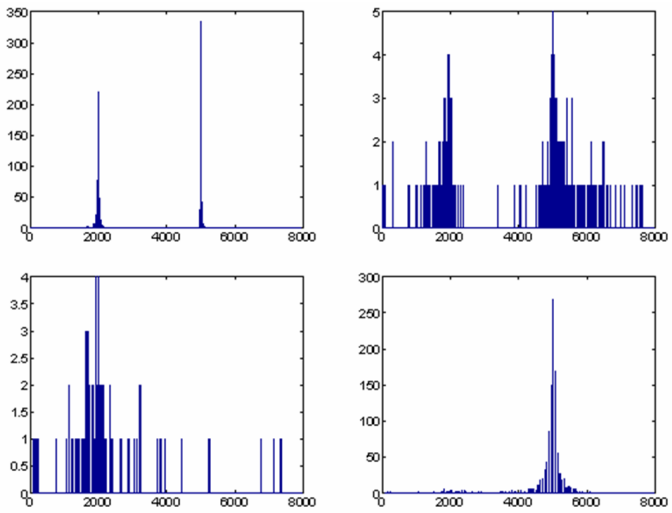

Figure 8.6: The *blind zone* phenomenon. From left to right and top to bottom, DWT-ICSS analysis of scenario 2 at scales 1, 3, 4 and 6. Scale 4 is the blind zone for the second H change. The first H change point disappears at scales higher than 4 due to the lack of resolution of DWT.

approach can be observed when compared to the old method (treating both trees separately). Results are now comparable to those provided by the DWT method, while keeping very good precision in true point localization. Note the still significant shift with respect to the true point for DWT despite using appropriate corrections for centering the transform's output. With these results we consider validated the DTWT-modulus method, which will be used in the rest of the chapter.

8.3 Statistical characterization of the DTWT coefficients of a pure FGN trace

This section studies the distribution and the correlation of the real and imaginary parts and the modulus of the DTWT coefficients of the usual synthetic FGN trace of length 131072 samples, Hurst parameter $H = 0.8$, zero mean and unit variance. The analysis was performed with the `near_sym_b` and `qshift_b` filters.

8.3.1 Distribution of the coefficients

The results are shown in Figure 8.7, and Table 8.3 includes the estimation of the Generalized Gaussian shape parameter for each data set. The first remarkable feature is the almost perfect normality of the real and imaginary parts of the coefficients for the lower scales. The higher scales deviate more from normality than the DWT and MODWT coefficients in the same situation (see Figures 6.4 and 7.2, respectively). Regarding the modulus, the distribution is obviously not Gaussian (though it does not deviate too much, as can be seen in the shape parameter) and resembles a Chi-

square χ^2_k with $k = 2$ or $k = 3$ degrees of freedom. A Chi-square with *exactly* two degrees of freedom is what is expected for the sum of the squares of two independent, identically distributed Gaussian distributions; the presence of the square root can explain the deviation towards $k = 3$. Though the distributions deviate from the normality expected by the variance change detectors, we expect this effect not to excessively influence our algorithm, as we will see later during the method's validation with synthetic traces.

Scales	$j = 1$	$j = 5$	$j = 10$
Real part	1.99	2.08	2.40
Imaginary part	1.97	2.06	2.31
Modulus	1.76	2.08	2.39

Table 8.3: GGD shape parameter of the DWT decomposition of the FGN trace with $H = 0.8$.

8.3.2 Correlation of the DTWT coefficients and its influence

Our aim now is to assess the degree of correlation and its influence in the results of our algorithms. As we did in the previous cases, the central role of the Cumulated Sum of Squares in our algorithms makes it the natural choice for calibrating the influence of the correlation. Figure 8.8 plots the autocorrelation functions of the DTWT modulus at each scale (with the 95% confidence intervals), while Figure 8.9 plots the CSS statistic (with the 99% critical values of the statistic). The figures show the residual autocorrelation already described in Section 2.8.6, where a Gaussian white noise was analyzed with the DTWT. The good news is that the autocorrelation is kept to an acceptable level. Figure 8.8 shows that no statistically significant correlation lasts more than two (in most cases, just one) lags, and that this correlation is further mitigated when the scale increases. Besides, Figure 8.9 proves the feasibility of using the DTWT together with the variance change detectors based on the CSS statistic: only one of the scales ($j = 2$) is close to producing a false change point, at the rather strict 99% significance level[2].

8.4 Analysis of the effects of a variance change

Let us now assess how variance changes are detected by DTWT, by analyzing the usual FGN trace ($H = 0.8$) with variances $\sigma_1^2 = 1$ for the first half and $\sigma_2^2 = 4$ for the second. The analyses have been performed with the DTWT-ICSS and DTWT-SIC methods, at both the 95% and 99% significance levels, using the $(13, 19)$ tap `near_sym_b` and the $(14, 14)$ tap `qshift_b` filters.

8.4.1 DTWT-ICSS

Figure 8.10 contains the multiscale change points diagrams for the DTWT-ICSS. For both significance levels the results are the same, that is, all the points that survived the 95% also passed the

[2]Such a false, isolated change point would be later filtered by the quorum mechanism incorporated in our method.

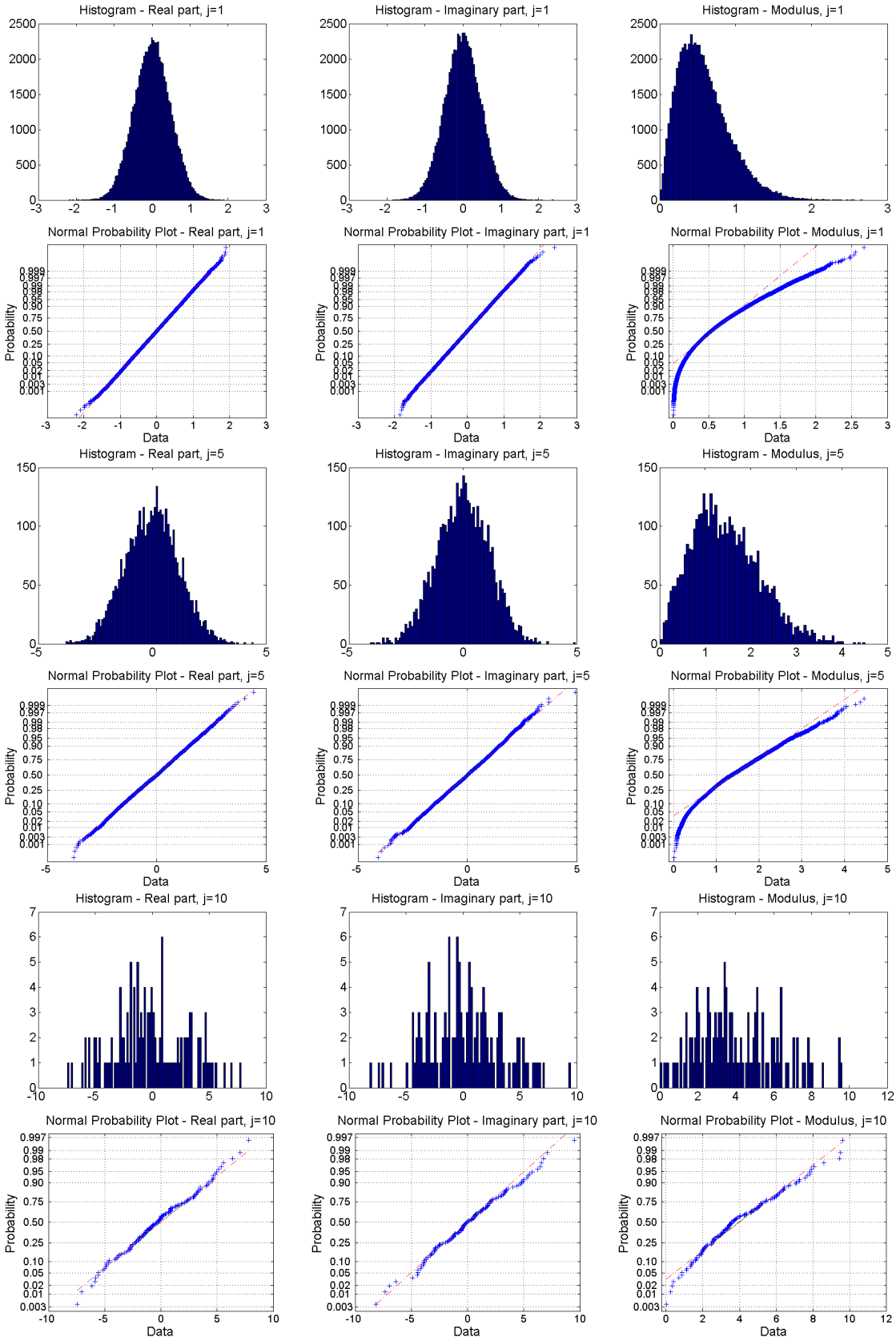

Figure 8.7: Histograms and normal probability plots of the real and imaginary parts and the modulus of the DTWT coefficients of the FGN trace with $H = 0.8$ when analyzed with the `near_sym_b` and `qshift_b` filters. From top to bottom, the analysis for the coefficients at scales $j = 1$, 5 and 10.

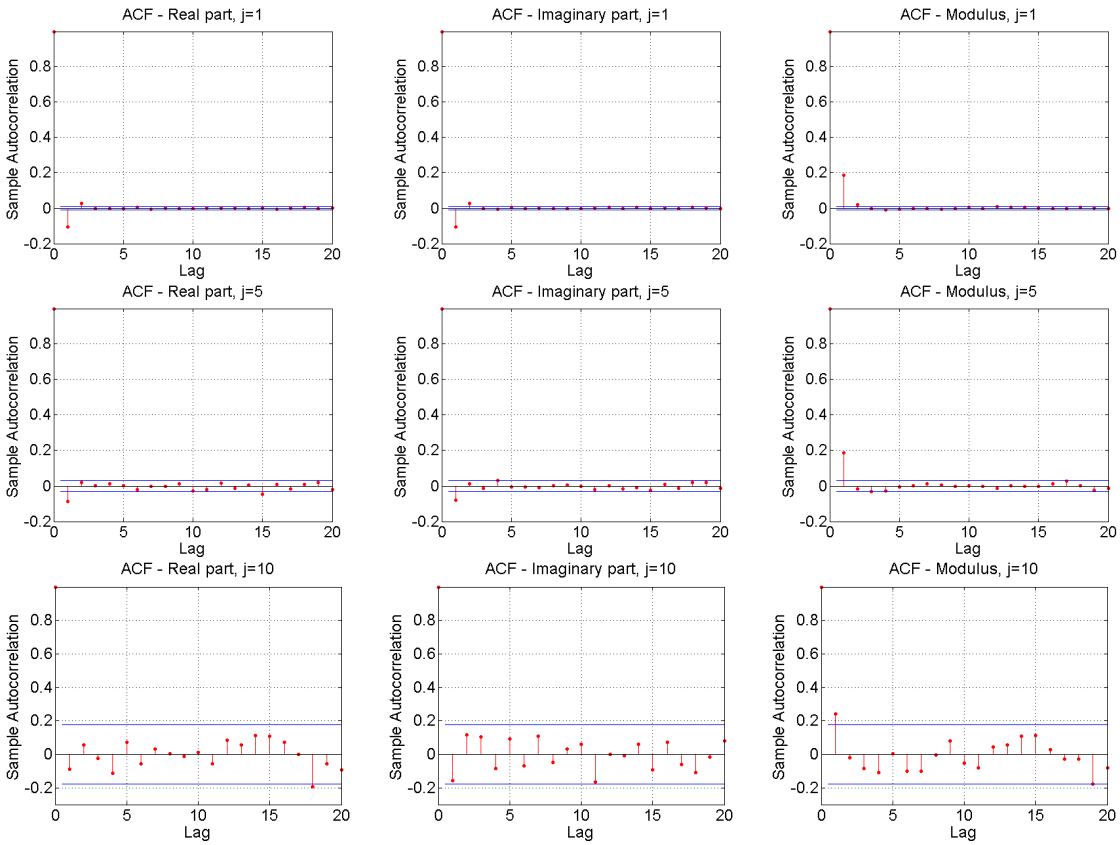

Figure 8.8: Sample autocorrelation of the real part, imaginary part and modulus of the DTWT coefficients of the FGN trace ($j = 1, 5, 10$), with the 95% confidence intervals (horizontal lines).

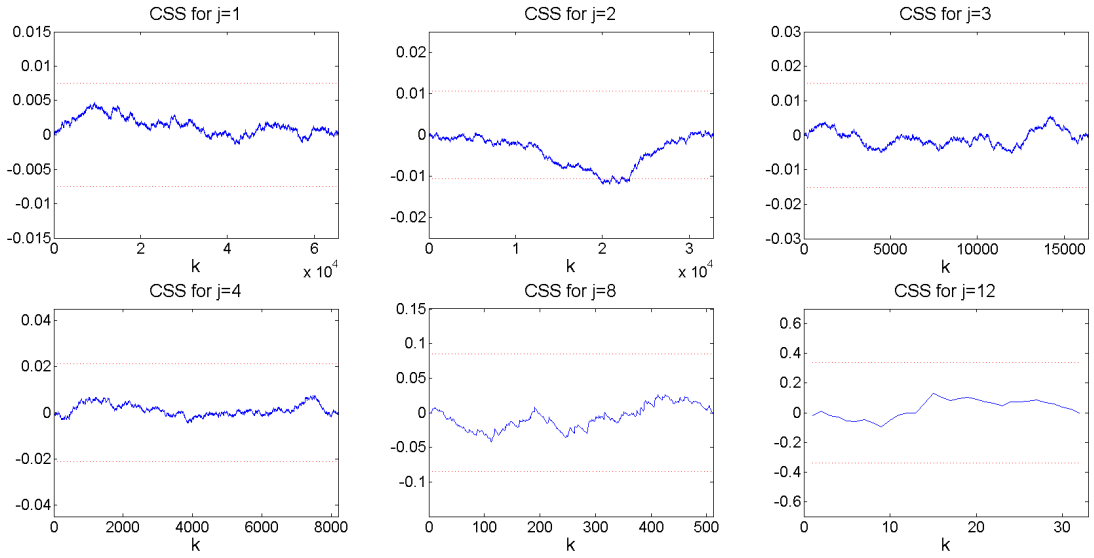

Figure 8.9: CSS statistic applied to the modulus of the DTWT coefficients of the FGN trace, for $j = 1, 2, 3, 4, 8$ and 12 at the 99% significance level. The associated critical levels $\pm\frac{1.628}{\sqrt{T/2}}$ are depicted as red horizontal lines. Notice the difference in the units of the y axis.

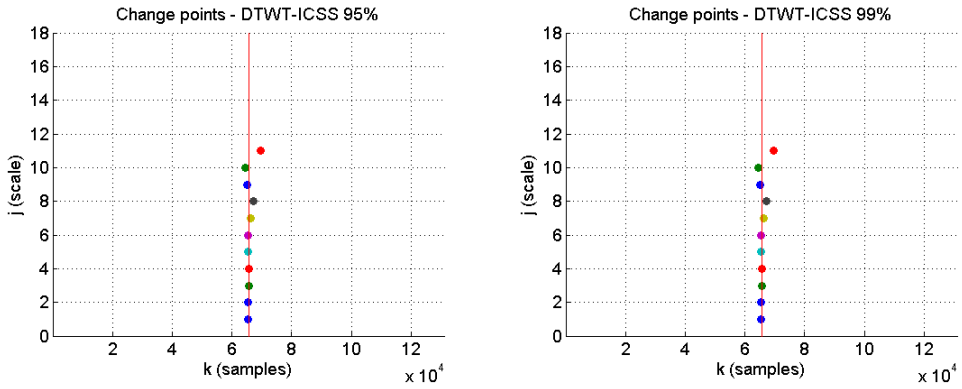

Figure 8.10: Change point diagram obtained with the DTWT-ICSS method for the variance change case. *Left:* 95% significance level. *Right*: 99% significance level.

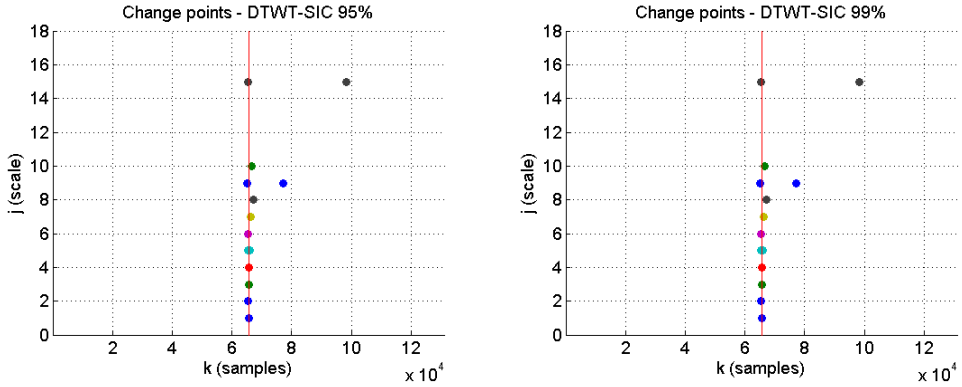

Figure 8.11: Change point diagram obtained with the DTWT-SIC method for the variance change case. *Left:* 95% significance level. *Right*: 99% significance level.

99% test. The change point located at $k = 65536$ is well localized at a good amount of scales (on the order of 10-12 scales), comparable to the results obtained with the DWT-ICSS method (see Figures 6.11 and 6.12). When the Hough transform-based alignment method is used, with a quorum of 4 changes and a resolution of 100 points, the mean change point is estimated at position $k = 65549$; lower resolutions (5 and 20) did not return any change point. In these terms the results are again similar to those obtained with the DWT-based methods.

8.4.2 DTWT-SIC

Figure 8.11 repeats the analysis of the previous section for the DTWT-SIC method. The results are, again, equal for the two significance levels. Some false change points (not present at the ICSS analysis) appear at the higher scales. After the alignment procedure, the change point is detected at $k = 65540$. In all senses, the results are comparable to the ones obtained with the ICSS and with the DWT-based methods. Since no significant differences were found in either significance levels, we will restrict the following analysis to the usual 99% significance level.

8.5 Application to a synthetic trace with changes in the Hurst parameter

The first trace to be analyzed contains a transition from $H = 0.5$ to $H = 0.9$ in the middle of a 262144 samples FGN trace. Figure 8.12 shows the distribution of the modulus of the DTWT for such a composed trace. Comparing it to Figure 8.7, we can see that in the H change case the distribution still resemble the expected χ^2_k Chi-square, though the higher scales tend to follow the $k = 1$ or $k = 2$. The results of our algorithm were not excessively affected.

Figures 8.13 and 8.14 show the change point diagrams obtained with the DTWT-ICSS and DTWT-SIC methods, respectively, at the 99% significance level. The alignment method (with resolution 200 and quorum 4) returned changes at $k = 131054$ for the ICSS and at $k = 131073$ for the SIC. The results are quite similar to those found for the DWT-based methods in Sections 6.7.1 and 6.7.2. In terms of amount of scales in which the change is detected, the DTWT gives in some cases one or two fewer scales than DWT, but on the other hand the DTWT are better localized than those returned by the DWT. This effect is a repetition of what we already saw in Section 8.2.1.

For the smoother change $H = 0.7 \rightarrow 0.8$, the distributions of Figure 8.15 and the change points diagrams in Figures 8.16 and 8.17 show not so good behavior. The histograms show the higher dispersion of the coefficients and its increase with the scale. On the other hand, normality seems to be maintained at a higher degree than in the $H = 0.5 \rightarrow 0.9$. DTWT-ICSS still resists the comparison with DWT-ICSS in terms of changes detected and dispersion around the true change point, but DTWT-SIC clearly shows very weak changes (note the *sizes*) and an increase in their dispersion. It seems that SIC is more sensitive to the non-Gaussianity of the coefficients. The consequences of this behavior will also appear in the following section, where we will empirically evaluate the power of the tests.

8.6 Empirical assessment of the power of the DTWT methods when the Hurst parameter changes

The results of the empirical test for assessing the power of the DTWT-based algorithms are shown in Tables 8.4 and 8.5 for the cases of $H = 0.5 \rightarrow 0.9$ and $H = 0.7 \rightarrow 0.8$, respectively. The main conclusions extracted from the data are:

- Contrary to what happened with the MODWT, few spurious or false points are detected (recall the theoretical amount of 1000 changes), thus confirming the influence of the better correlation properties. Besides, almost all the points belong to the *true* region, confirming the good dispersion properties of the DTWT-modulus approach.
- The comparison with DWT offers different results, depending on the variance change detector chosen. When comparing DWT-ICSS (Tables 6.9 and 6.11) versus DTWT-ICSS, for both cases $H = 0.5 \rightarrow 0.9$ and $H = 0.7 \rightarrow 0.8$, and for any combination of significance level, resolution

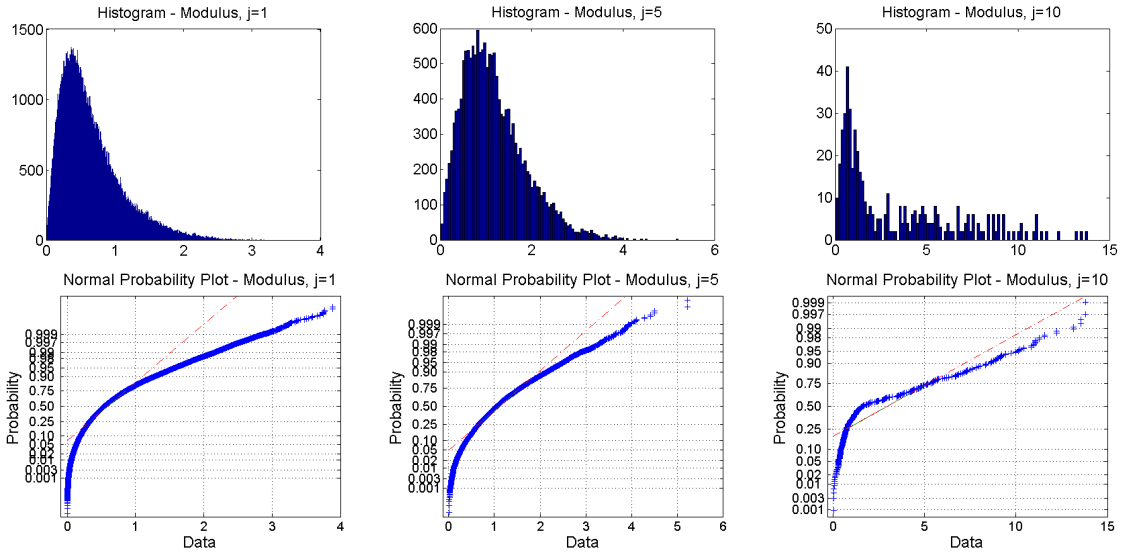

Figure 8.12: Histograms and normal probability plots of the modulus of the DTWT coefficients of the FGN trace with a change in its Hurst parameter from $H = 0.5$ to $H = 0.9$.

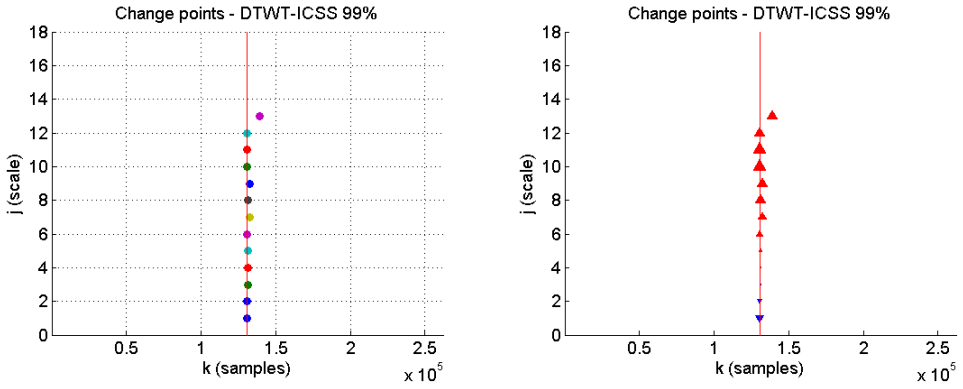

Figure 8.13: *Left*: Change point diagram obtained with the DTWT-ICSS method for the $H = 0.5 \rightarrow 0.9$ Hurst parameter change. *Right*: Size and sign of the change points.

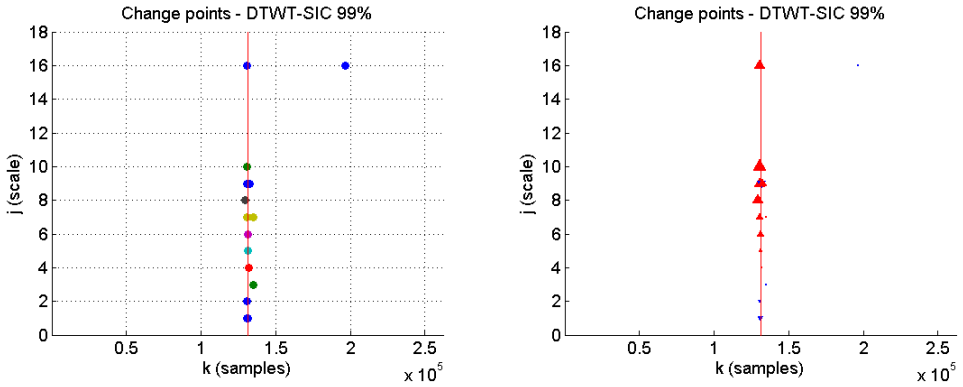

Figure 8.14: *Left*: Change point diagram obtained with the DTWT-SIC method for the $H = 0.5 \rightarrow 0.9$ Hurst parameter change. *Right*: Size and sign of the change points.

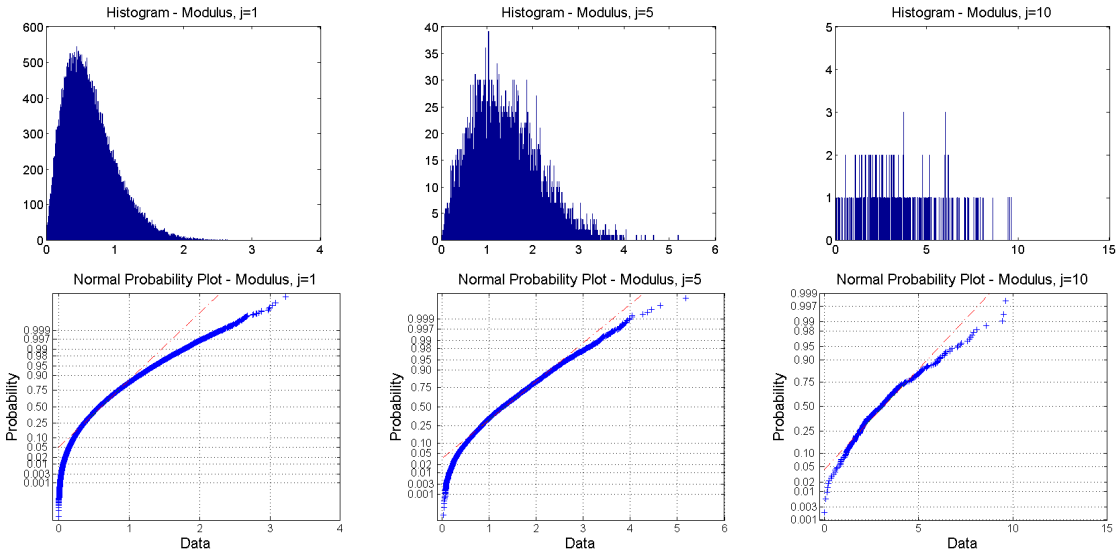

Figure 8.15: Histograms and normal probability plots of the modulus of the DTWT coefficients of the FGN trace with a change in its Hurst parameter from $H = 0.7$ to $H = 0.8$.

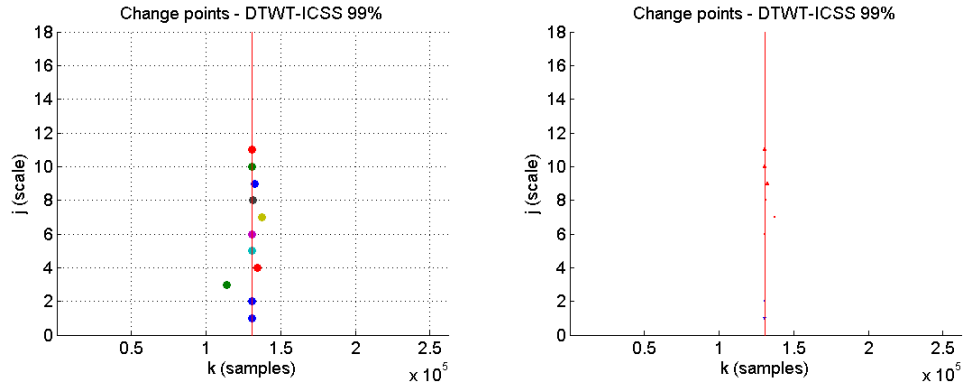

Figure 8.16: *Left*: Change point diagram obtained with the DTWT-ICSS method for the $H = 0.7 \rightarrow 0.8$ Hurst parameter change. *Right*: Size and sign of the change points.

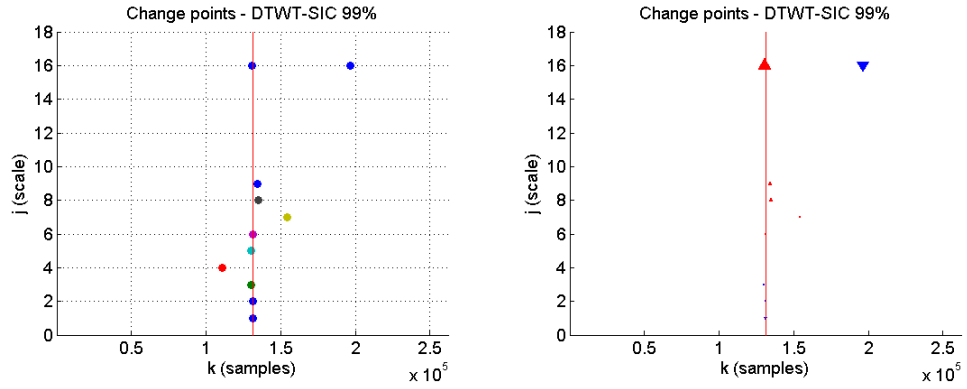

Figure 8.17: *Left*: Change point diagram obtained with the DTWT-SIC method for the $H = 0.7 \rightarrow 0.8$ Hurst parameter change. *Right*: Size and sign of the change points.

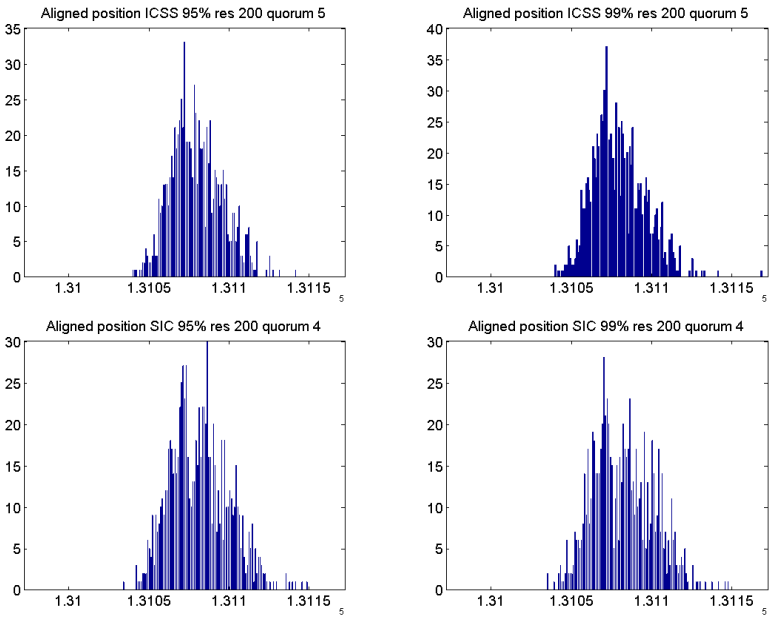

Figure 8.18: Histograms of the change points when DTWT is applied to 1000 traces whose H parameter changes from 0.5 to 0.9. *Top left*: DTWT-ICSS 95%, resolution=200 samples, quorum=5 changes. *Top right*: DTWT-ICSS 99%, r=200, q=5 . *Bottom left*: DTWT-SIC 95%, r=200, q=4. *Bottom right*: DTWT-SIC 99%, r=200, q=4. The x axis has been zoomed on the region 131072 ± 100.

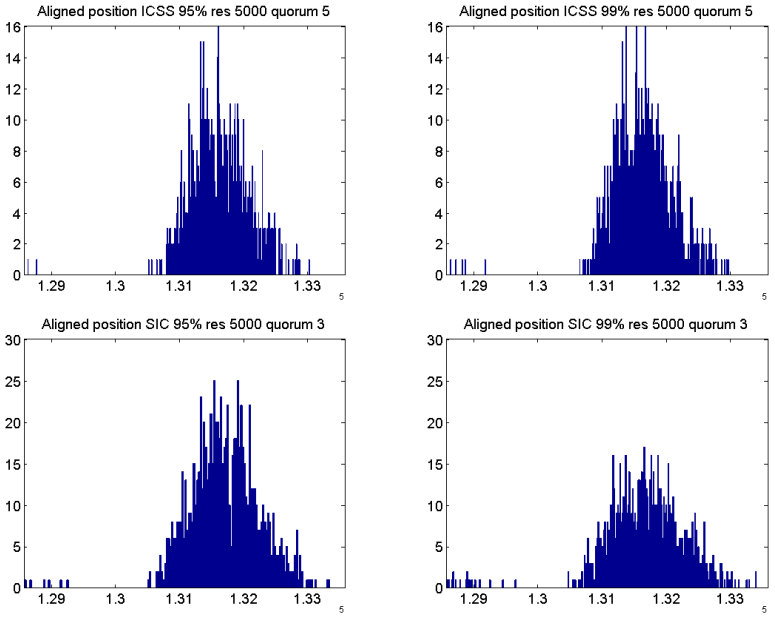

Figure 8.19: Histograms of the change points when DTWT is applied to 1000 traces whose H parameter changes from 0.7 to 0.8. *Top left*: DTWT-ICSS 95%, resolution=5000 samples, quorum=5 changes. *Top right*: DTWT-ICSS 99%, r=5000, q=5. *Bottom left*: DTWT-SIC 95%, r=5000, q=3. *Bottom right*: DTWT-SIC 99%, r=5000, q=3. The x axis is zoomed on the region 131072 ± 2500.

		ICSS 95%		ICSS 99%		SIC 95%		SIC 99%	
resol.	quorum	Total	*True*	Total	*True*	Total	*True*	Total	*True*
5 smpls	3 chg	806	**788** (97.8%)	753	**740** (98.3%)	631	**598** (94.8%)	500	**472** (94.4%)
	4 chg	562	**561** (99.8%)	509	**508** (99.8%)	335	**334** (99.7%)	212	**211** (99.5%)
	5 chg	346	**346** (100%)	293	**293** (100%)	126	**126** (100%)	464	**6** (100%)
	6 chg	182	**182** (100%)	134	**134** (100%)	42	**42** (100%)	12	**12** (100%)
	7 chg	63	**63** (100%)	40	**40** (100%)	9	**9** (100%)	4	**4** (100%)
20 smpls	3 chg	899	**877** (97.6%)	859	**841** (97.9%)	755	**715** (94.7%)	640	**611** (95.5%)
	4 chg	697	**696** (99.9%)	656	**655** (99.8%)	477	**472** (98.9%)	342	**340** (99.4%)
	5 chg	461	**461** (100%)	402	**402** (100%)	230	**230** (100%)	131	**131** (100%)
	6 chg	278	**461** (100%)	228	**228** (100%)	82	**82** (100%)	34	**34** (100%)
	7 chg	122	**461** (100%)	89	**89** (100%)	24	**24** (100%)	7	**7** (100%)
200 smpls	3 chg	1005	**985** (98.0%)	998	**981** (98.3%)	1018	**961** (94.4%)	978	**931** (95.2%)
	4 chg	965	**959** (99.4%)	957	**950** (99.3%)	923	**900** (97.5%)	848	**832** (98.1%)
	5 chg	870	**868** (99.8%)	842	**840** (99.8%)	763	**758** (99.3%)	641	**638** (99.5%)
	6 chg	719	**719** (100%)	673	**673** (100%)	518	**518** (100%)	370	**370** (100%)
	7 chg	509	**509** (100%)	460	**460** (100%)	300	**300** (100%)	184	**184** (100%)
1000 smpls	3 chg	1013	**1000** (98.7%)	1007	**1000** (99.3%)	1048	**985** (94.0%)	1005	**981** (97.6%)
	4 chg	1001	**1000** (99.9%)	1001	**1000** (99.9%)	1017	**985** (96.9%)	1002	**981** (97.9%)
	5 chg	992	**992** (100%)	991	**991** (100%)	1000	**983** (98.3%)	994	**978** (98.4%)
	6 chg	973	**973** (100%)	966	**966** (100%)	953	**945** (99.2%)	897	**889** (99.1%)
	7 chg	878	**878** (100%)	854	**854** (100%)	832	**830** (99.8%)	748	**747** (99.9%)
5000 smpls	3 chg	1009	**1000** (99.1%)	1003	**1000** (99.7%)	1071	**1000** (93.4%)	1005	**1000** (99.5%)
	4 chg	1001	**1000** (99.9%)	1000	**1000** (100%)	1018	**1000** (98.2%)	1002	**1000** (99.8%)
	5 chg	1000	**1000** (100%)	1000	**1000** (100%)	1004	**1000** (99.6%)	1000	**1000** (100%)
	6 chg	1000	**1000** (100%)	1000	**1000** (100%)	1001	**1000** (99.29%)	1000	**1000** (100%)
	7 chg	1000	**1000** (100%)	1000	**1000** (100%)	1000	**1000** (100%)	1000	**1000** (100%)
10000 smpls	3 chg	1004	**1000** (99.6%)	1001	**1000** (99.9%)	1076	**1000** (93.0%)	1004	**1000** (99.6%)
	4 chg	1000	**1000** (100%)	1000	**1000** (100%)	1020	**1000** (98.0%)	1002	**1000** (99.8%)
	5 chg	1000	**1000** (100%)	1000	**1000** (100%)	1005	**1000** (99.5%)	1000	**1000** (100%)
	6 chg	1000	**1000** (100%)	1000	**1000** (100%)	1001	**1000** (99.9%)	1000	**1000** (100%)
	7 chg	1000	**1000** (100%)	1000	**1000** (100%)	1000	**1000** (100%)	1000	**1000** (100%)

Table 8.4: Results for the alignment detection algorithm for a trace whose H parameter changes from 0.5 to 0.9, analyzed with the DTWT-ICSS and DTWT-SIC methods (`near_sym_b` and `qshift_b` filters). The *true* changes are defined as those around ± resolution/2 positions around the true change position.

and quorum, DTWT outperforms the DWT. And what is more important, we can be sure of the validity of the additional changes found by the DTWT, since practically no spurious changes are found (almost all detections are *true* changes). Regarding the comparison between DWT-SIC (Tables 6.14 and 6.15) and DTWT-SIC, the results are clearly against DTWT. For both H changes DWT provides more detections, especially at the lower resolutions and quorums. DTWT does not provide comparable results until the resolution rises to 200 samples for the $H = 0.5 \rightarrow 0.9$ case and to 1000 samples for the $H = 0.7 \rightarrow 0.8$ scenario.

- In general, the results obtained with the ICSS statistic are much better than those obtained with the SIC. It seems the ICSS adapts better to the non-Gaussian distribution of the DTWT modulus. In any case, the DTWT-SIC results are not so bad as to abandon the method.
- A good tradeoff for the $H = 0.5 \rightarrow 0.9$ case is obtained with a resolution of 200 samples and a quorum of 5 alignments for the DTWT-ICSS case, while 200 samples and 4 alignments seem to be the choice for DTWT-SIC. In these two situations we detect more than 83% of the true changes. Note the differential (in quorum) needed for the SIC method to provide results

		ICSS 95%		ICSS 99%		SIC 95%		SIC 99%	
resol.	quorum	Total	*True*	Total	*True*	Total	*True*	Total	*True*
5 smpls	3 chg	96	**85** (88.5%)	74	**67** (90.5%)	7	**2** (28.6%)	1	**1** (100%)
	4 chg	19	**18** (94.7%)	13	**13** (100%)	1	**1** (100%)	0	**0**
	5 chg	0	**0**	0	**0**	0	**0**	0	**0**
	6 chg	0	**0**	0	**0**	0	**0**	0	**0**
	7 chg	0	**0**	0	**0**	0	**0**	0	**0**
20 smpls	3 chg	130	**120** (92.3%)	100	**92** (92.0%)	41	**9** (22.0%)	9	**5** (55.5%)
	4 chg	29	**28** (96.6%)	20	**20** (100%)	9	**2** (22.2%)	3	**0** (0%)
	5 chg	2	**2** (100%)	2	**2** (100%)	1	**0** (0%)	0	**0**
	6 chg	0	**0**	0	**0**	0	**0**	0	**0**
	7 chg	0	**0**	0	**0**	0	**0**	0	**0**
200 smpls	3 chg	429	**370** (86.2%)	364	**330** (90.7%)	202	**88** (43.5%)	93	**55** (59.1%)
	4 chg	164	**158** (96.3%)	135	**128** (94.8%)	60	**34** (56.7%)	16	**9** (56.3%)
	5 chg	38	**37** (97.4%)	33	**33** (100%)	17	**8** (47.0%)	3	**3** (66.7%)
	6 chg	6	**6** (100%)	5	**5** (100%)	1	**0** (0%)	0	**0**
	7 chg	0	**0**	0	**0**	0	**0**	0	**0**
1000 smpls	3 chg	843	**710** (84.2%)	834	**685** (82.1%)	648	**409** (63.1%)	431	**321** (74.5%)
	4 chg	522	**479** (91.2%)	473	**426** (90.1%)	241	**173** (71.8%)	127	**105** (82.7%)
	5 chg	258	**253** (98.1%)	202	**195** (96.5%)	73	**59** (80.8%)	30	**28** (93.43%)
	6 chg	101	**100** (99.0%)	65	**65** (100%)	23	**20** (87.0%)	3	**3** (100%)
	7 chg	24	**24** (100%)	13	**13** (100%)	7	**6** (85.7%)	0	**0**
5000 smpls	3 chg	1011	**980** (97.0%)	1005	**979** (97.4%)	1090	**914** (83.8%)	981	**908** (92.6%)
	4 chg	993	**973** (98.0%)	995	**976** (98.1%)	913	**837** (91.7%)	767	**744** (97.0%)
	5 chg	950	**942** (99.2%)	994	**929** (98.4%)	680	**648** (95.3%)	478	**474** (99.2%)
	6 chg	816	**812** (99.5%)	800	**797** (99.6%)	409	**398** (97.3%)	228	**228** (100%)
	7 chg	625	**625** (100%)	564	**564** (100%)	177	**175** (98.9%)	75	**75** (100%)
10000 smpls	3 chg	1015	**998** (98.3%)	1003	**997** (99.4%)	1125	**976** (86.8%)	1016	**973** (95.8%)
	4 chg	1002	**998** (99.6%)	1000	**997** (99.7%)	1009	**958** (95.0%)	937	**915** (97.7%)
	5 chg	995	**993** (99.8%)	995	**993** (99.8%)	913	**888** (97.3%)	752	**746** (99.2%)
	6 chg	971	**970** (99.9%)	959	**958** (99.9%)	725	**715** (98.6%)	494	**494** (100%)
	7 chg	906	**906** (100%)	856	**856** (100%)	493	**487** (98.8%)	208	**208** (100%)

Table 8.5: Results for the alignment detection algorithm for a trace whose H parameter changes from 0.7 to 0.8, analyzed with the DTWT-ICSS and DTWT-SIC methods (`near_sym_b` and `qshift_b` filters). The *true* changes are defined as those around ± resolution/2 positions around the true change position.

comparable to those obtained by the ICSS method. Taking the aforementioned parameters, we get the histograms presented in Figure 8.18, where the change points appear well centered and slightly dispersed around the true change position $k = 131072$.

- In the $H = 0.7 \rightarrow 0.8$ case the tradeoff is attained with a higher resolution value of 5000 samples, and with a quorum of up to 5 aligned changes for the DTWT-ICSS, while the SIC method needs to lower the quorum to 3 alignments. With these parameters we get more than 90% of the true change points. This lack of performance (a higher resolution and a lower quorum) is obviously caused by the much smoother H change, which causes the variances at every scale to vary much less than in the $H = 0.5 \rightarrow 0.9$ and therefore more difficult to detect.
- The histograms shown in Figure 8.19 illustrate the higher dispersion (recall the different resolutions used in the X axis) and even some (slightly) positive bias (i.e., the detected changes tend to be to the right of the true change position $k = 131072$) for the $H = 0.7 \rightarrow 0.8$ case. Therefore, the detection of smooth H changes will always suffer a delay (in samples, or in time if a progressive algorithm was used) when compared to an abrupt H change.

8.7 Application to real traffic traces

We now apply our methods to the Bellcore dataset traces. As we did in the previous chapters, first we will analyze the distribution of the DTWT coefficients and then we will apply the segmentation method.

8.7.1 Statistical distribution of the DTWT coefficients

Figure 8.20 and 8.21 show the histograms and probability plots for the modulus of the DTWT when applied to `BC-pAug89` and `BC-OctExt` traces (aggregated at 10 ms and 1 s, respectively), while Table 8.6 presents the Generalized Gaussian shape parameters estimated for the distributions. The normality plots have been omitted, due to the clear non-Gaussianity that can be deduced from the histograms and the table (a Chi-square with 2 or 3 degrees of freedom could be, again, a better model for such distributions).

In terms of the GGD shape parameter, `BC-pAug89` can still be considered close to Gaussian at the higher scales. On the other hand, `BC-OctExt` is far from Gaussianity. This fact can strongly influence the segmentation algorithm.

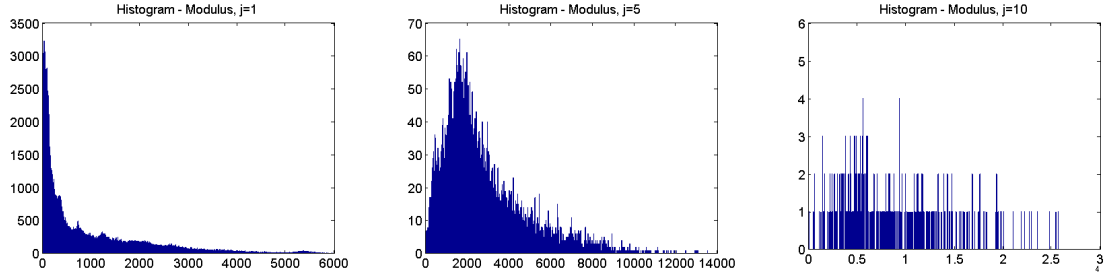

Figure 8.20: Histograms of the modulus of the DTWT coefficients of the `BC-pAug89` trace (aggregated at 10 ms) analyzed with `near_sym_b` and `qshift_b` filters. From left to right, the analysis for the coefficients at scales $j = 1$, $j = 5$ and $j = 10$.

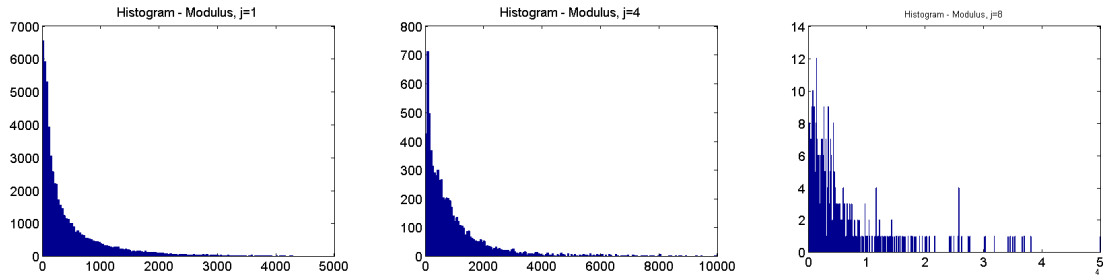

Figure 8.21: Histograms of the modulus of the DTWT coefficients of the `BC-OctExt` trace (aggregated at 1 s) analyzed with `near_sym_b` and `qshift_b` filters. From left to right, the analysis for the coefficients at scales $j = 1$, $j = 4$ and $j = 8$.

	$j=1$	$j=5$	$j=10$		$j=1$	$j=4$	$j=8$
BC-pAug89	1.46	1.44	1.90	BC-OctExt	0.89	0.78	0.96

Table 8.6: GGD shape parameter of the DTWT decomposition of the Bellcore traces.

8.7.2 Analysis of the Bellcore traces

Figures 8.22 and 8.23 show the DTWT-ICSS and DTWT-SIC segmentations of the BC-pAug89 trace, at the 99% significance level, while Figures 8.24 and 8.25 present the segmentation of the BC-OctExt trace. The analysis has been performed at different resolutions and quorums, in order to assess the influence of those parameters.

The main conclusions are:

- Compared with the DWT and MODWT-based methods, DTWT does not suffer from the *noise* at the lower scales that made us propose to ignore the first scale (the j_1 parameter). DTWT does not need such modification of the algorithm.
- Though the methods are not directly comparable due to the different filter sets and parameters, at the higher scales DTWT is less noisy than MODWT but still detects more changes than the DWT, thus extending its power to the lower frequencies.
- The time evolution of the Hurst parameter is tracked again with good results, in terms of the (almost non-existing) overlapping of the confidence intervals of the H segments. The only cases where this rule is broken are the beginning ($0 < k < 50000$) and the middle ($190000 < k < 260000$) of both the BC-pAug89 trace for both the DTWT-ICSS and the DTWT-SIC. In these cases we are detecting a traffic variance change instead of a Hurst parameter change, as can be confirmed from the variance plot. The BC-OctExt is, as was stated in previous chapters, much more variable and their H-constant segments do not overlap.
- Both the DTWT-ICSS and DTWT-SIC methods track quite well the changes of the Hurst parameter shown in the constant-length H analysis. The tracking is especially good for the case of the BC-OctExt trace and acceptable for the BC-pAug89 trace, though in the latter some H peaks are neglected, such as the $H \sim 0.5$ region around $30000 < k < 35000$, which was not always detected by the other methods. For example, DWT-ICSS was able to detect it in Figures 6.44, 6.45 and 6.46, but was missed by the DTWT-SIC at the 99% significance level in Figures 6.51, 6.52 and 6.54 and barely detected in Figure 6.53 with the 95% significance level. Here we have an example of the role of the significance level of the variance change detection method as the *sensitivity* of the overall method.
- Comparing the segmentation results to those offered by the other transforms, some specific H-constant regions (such as $190000 < k < 245000$ and $260000 < k < 305000$ for BC-pAug89, and $63000 < k < 80000$ for BC-OctExt) are well localized by the DTWT, as they were (at least for some combinations of the parameters) by the DWT and the MODWT.

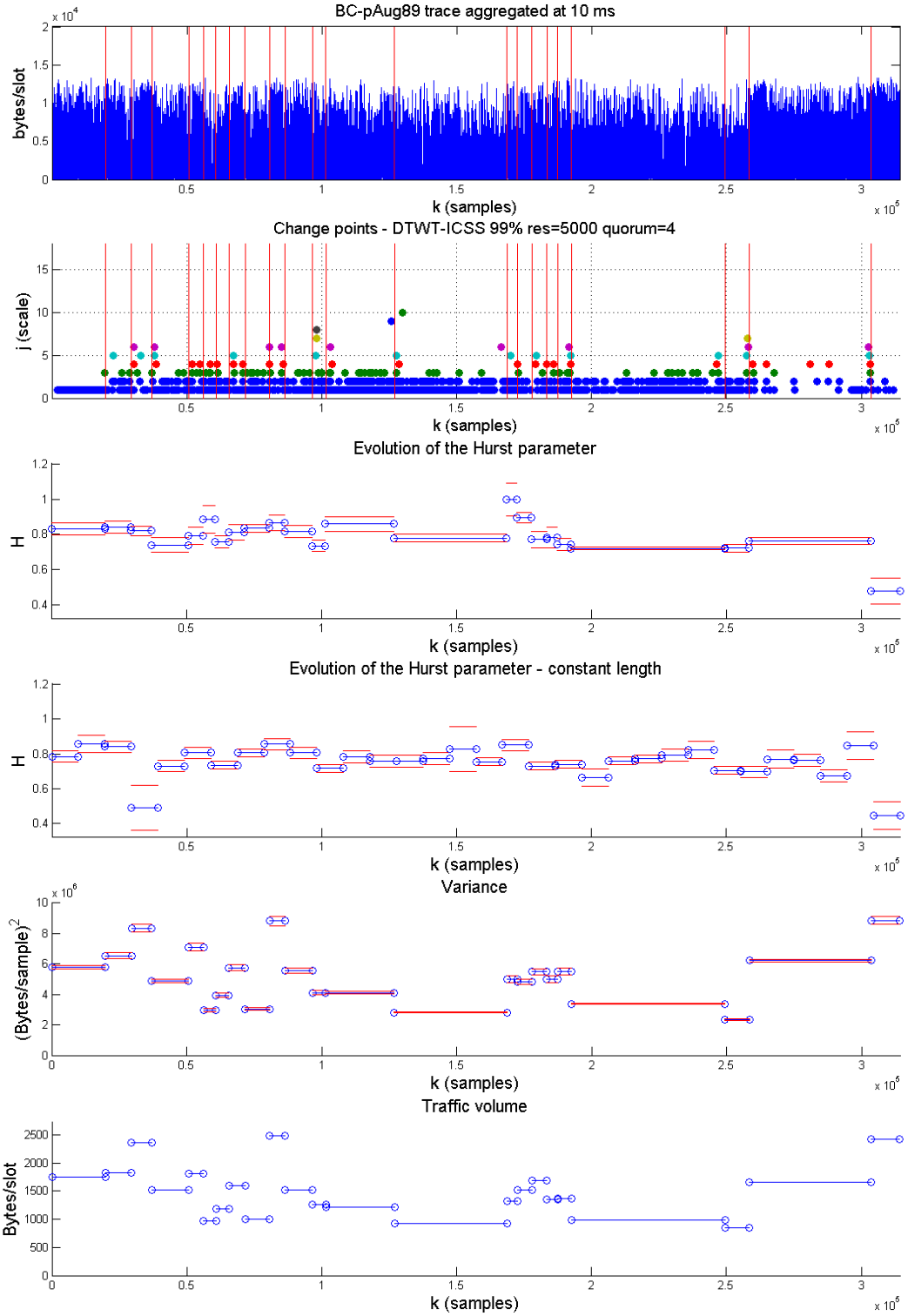

Figure 8.22: Segmentation of the `BC-pAug89` trace with the DTWT-ICSS method at the 99% significance level with resolution = 1000 samples and quorum = 5 aligned changes.

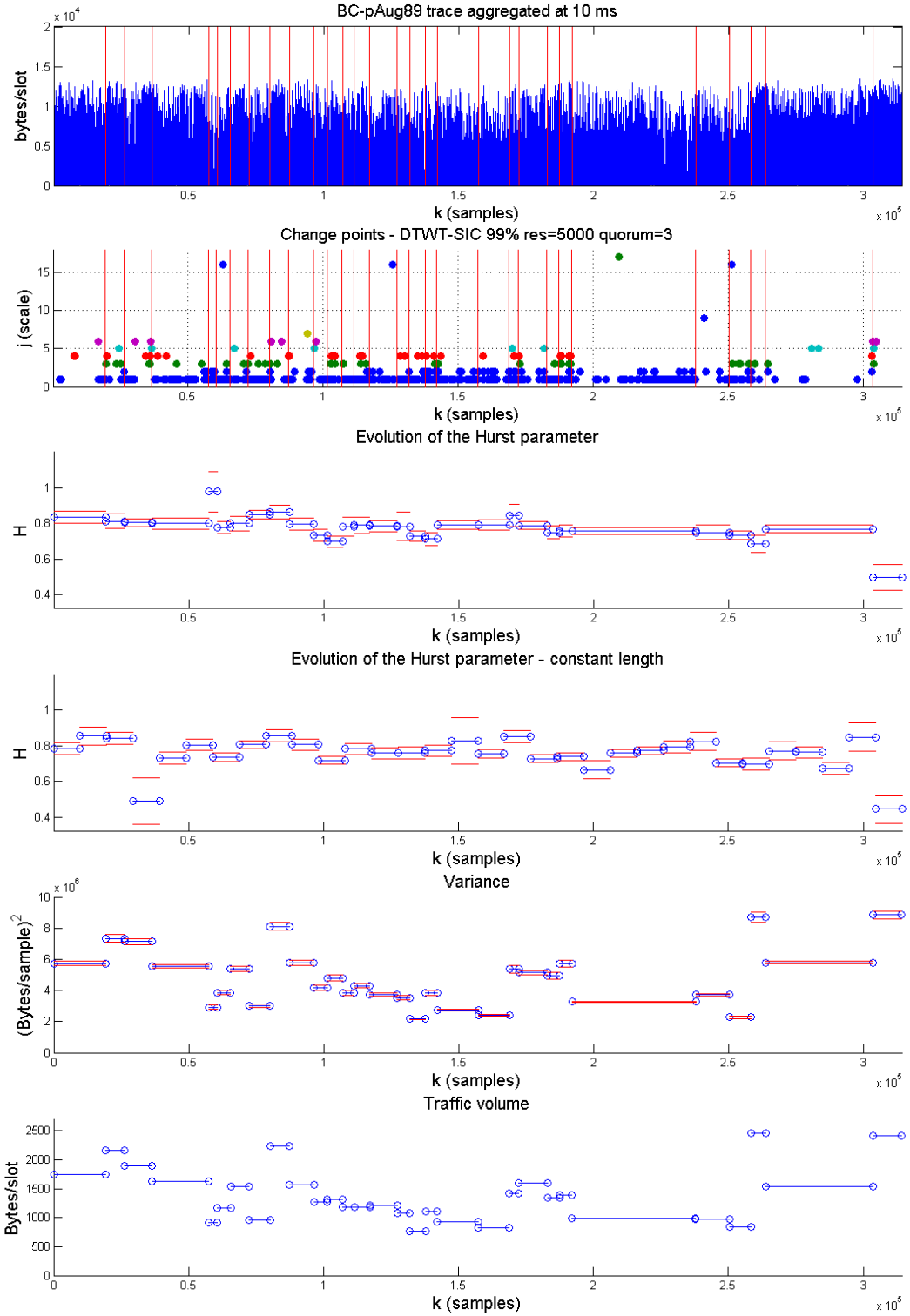

Figure 8.23: Segmentation of the `BC-pAug89` trace with the DTWT-SIC method at the 99% significance level with resolution = 5000 samples and quorum = 3 aligned changes.

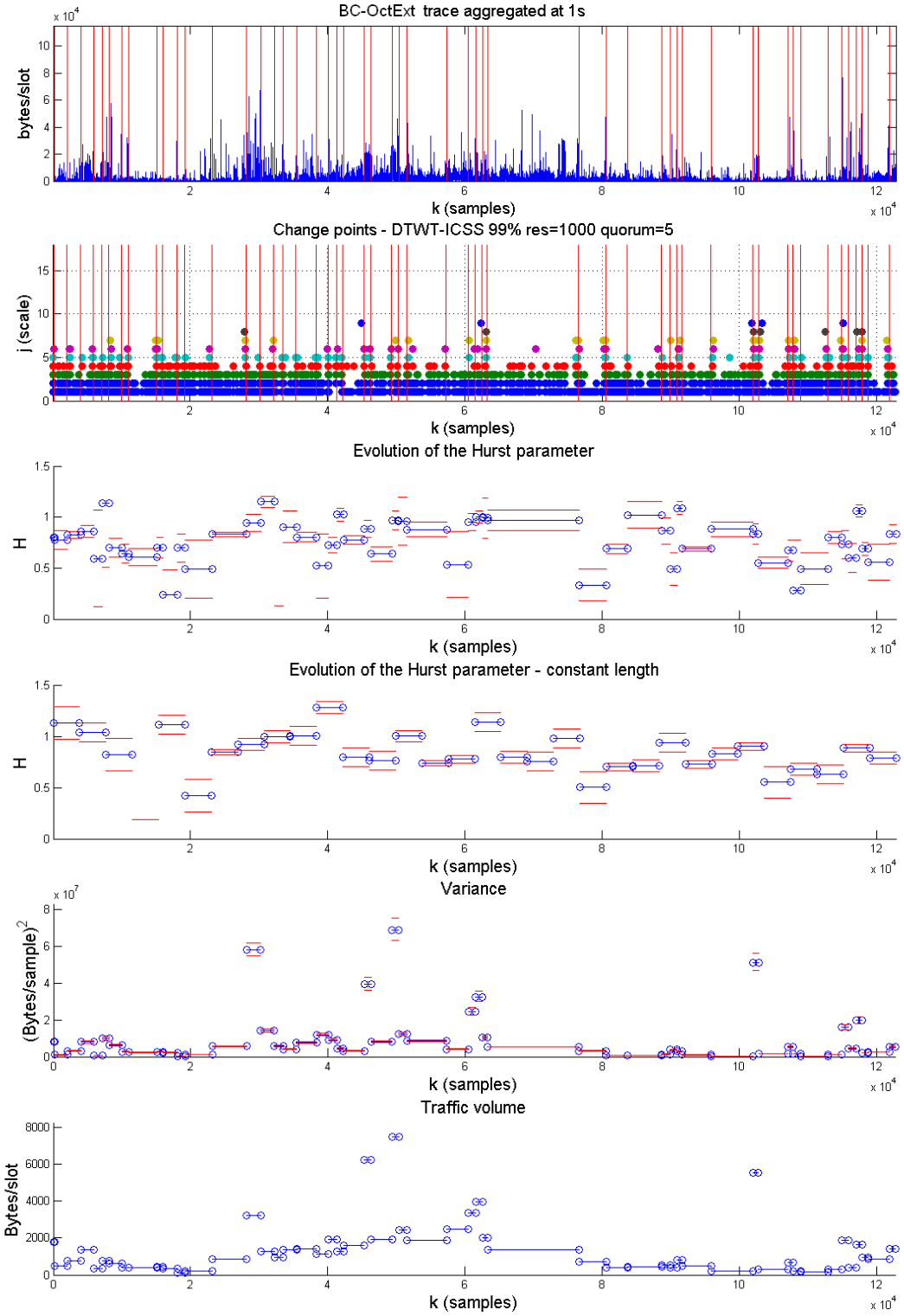

Figure 8.24: Segmentation of the `BC-OctExt` trace with the DTWT-ICSS method at the 99% significance level with resolution = 1000 samples and quorum = 5 aligned changes.

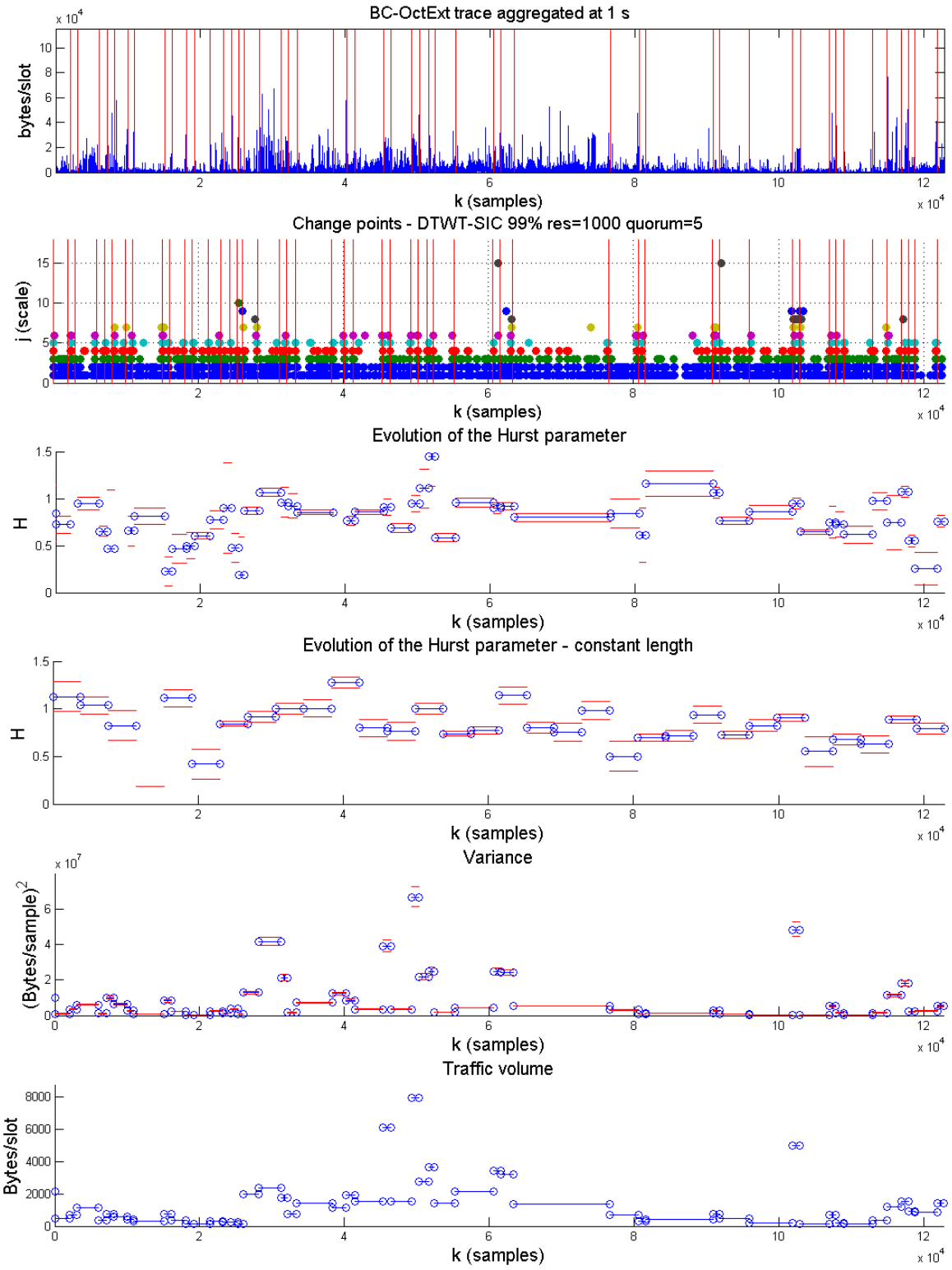

Figure 8.25: Segmentation of the `BC-OctExt` trace with the DTWT-SIC method at the 99% significance level with resolution = 1000 samples and quorum = 5 aligned changes.

8.8 Summary of the chapter

The topic of this chapter has been how to apply the Dual Tree (Complex) Wavelet Transform (DTWT) to our joint wavelet-VCD algorithm. The DTWT appeared as an interesting candidate since it falls somewhere in the middle of the correlation-vs-time-invariance trade-off of which the DWT and the MODWT represent the extremes.

We started by discussing how to interpret the complex nature of the transform, and through an empirical statistical test it was proven that computing the modulus of the transform gave better results than treating the two dual trees separately. The test also gave us the opportunity to compare the performance of the DTWT-modulus approach with the already studied DWT-based methods, finding that DTWT localization is better while maintaining approximately the same power as the DWT methods.

Having validated the DTWT-modulus approach, we started with the systematic characterization of its behavior, though not all the tests done in Chapter 6 to the DWT were repeated here. One of the first results was the finding of a Chi-square-type distribution in the modulus of the DTWT coefficients, as expected. Although this fact could influence the performance of the variance change detectors, the validation performed with synthetic traces has shown some differences in the performance of the algorithms. When it came to characterizing the DTWT methods when a variance or a Hurst parameter change is present, we found different performances for ICSS and SIC, behaving the former better than DWT-ICSS and the latter worse than DWT-SIC. Our conclusion was that SIC suffered more from the non-Gaussianity of the DTWT modulus than its counterpart. ICSS seems to be more distribution-independent than SIC, though the authors of both variance change detection methods ask for Gaussianity at their inputs.

Finally, the DTWT-mod-ICSS and DTWT-mod-SIC methods were applied to the usual Bellcore traffic traces, finding that the results found with the DTWT methods are comparable to the best ones obtained by the DWT and much better than those found with the MODWT.

To sum up, the DTWT-ICSS methods provide comparable or slightly better performance (in terms of amount of detected change points, and sensitivity to resolution and quorum changes) than DWT-ICSS, while DTWT-SIC results are a bit worse than those of the DWT-SIC.

Conclusions and topics for future research

Conclusions

The question answered by this thesis is: how to deal with the estimation of the fractal parameters of network traffic when such parameters are not stationary? Our answer has consisted of reviewing the work done by other authors, exploring different approaches, and developing algorithms capable of segmenting traffic in regions whose scaling parameters remain (approximately) constant.

We started with the state of the art review of the field of network traffic modeling [RSed], analyzing the evidences of the presence of scaling properties that cannot be captured by traffic models based on Poissonian or Markovian stochastic processes, a fact that has strong implications for network performance. On the other hand, the presence of scaling is encouraging for the development of new fractal-aware mechanisms such as traffic predictors, congestion control and resource-provisioning algorithms. Fractality has been found in several elements of IP networks (traffic sources, applications and services, protocols, file distributions, and users' behavior) and is here to stay. Moreover, the intrinsical non-stationarity of Internet traffic makes the fractal parameters change along time. This fact highlights the need for adaptive, scaling-change-aware network algorithms. Such algorithms will need a monitoring block at the input, capable of detecting the changes of the parameters. The review of the literature regarding the non-stationarity of the scaling parameters of traffic has revealed the existence of few algorithms that deal with this issue, and the fact that most of them work in off-line mode and apply a constant-length window approach. We argue that a more flexible, real-time-oriented estimator should be developed. That is why have tried to develop wavelet-based algorithms capable of segmenting the traffic into regions with a homogeneous variance structure (which in turn are scaling-constant).

Our segmenting algorithm is composed of three blocks: a wavelet transform, a variance change detection method applied to each of the wavelet subbands, and a clustering and alignment detection method based on the Hough Transform [Min05, RMSP05b, RMSP05a, RMS06]. The segmenter and is able to locate the simultaneous change of variance across scales, which can either correspond to a variance change or to a Hurst parameter change. The output of the method is a piecewise

segmentation of $var(t)$ and $H(t)$. Our approach is more general than the constant length window-based methods, since we can localize the variance-transition points to any position and scale. In addition, we can monitor the second order nonstationarities of the time series (both for the whole process or scale-by-scale).

The choice of the DWT as the tool for the study of scaling processes is justified due to its multiresolution analysis capabilities. Abry and Veitch exploited these features in their LogScale Diagram (LD), which is widely recognized as the best and most efficient estimator of the scaling parameters. On the other hand, DWT's resolution coarsens at the higher scales, and it lacks of shift invariance, thus making the MRA dependent on the time shifts of the original signal. The non-orthogonal, time-redundant MODWT is an alternative to the DWT due to its scale-constant resolution and its shift invariance, but at the price of a scale-dependent correlation. The WPT is another frequency-redundant transform which can be considered the counterpart of the MODWT in the frequency domain.

The Wavelet Packet Transform allows for total flexibility in choosing the transform basis, either orthogonal, incomplete or overcomplete. This transform opens the way to applying a time-varying basis analysis in which the signal is studied with a subband decomposition scheme that adapts to the spectral properties of the time series under study. We have developed a WPT-based estimator based on a graphic tool called the *Wavelet Packet variance map* [RS04a, RS04b]. Our intention was to use it jointly with the real-time splitting algorithm capable of performing a progressive orthogonal tiling, which would compute the best basis for the WPT decomposition as the traffic process evolves [RS02], but we stopped the development due to the problems found when trying to define the cost function for the basis choice, and turned our attention towards a time-dependent analysis of the variances at each scale. Nevertheless, the WPT-based estimator is a step ahead which provides several modes of estimation (with a limited or global subband set). The WPT-based analysis applied to the Bellcore traces has shown clear evidence of the non-stationarity of their scaling parameters [RS04a, RS04b, Min05].

Coming back to our time-segmenting approach, and since both the DWT and the MODWT represent extremes in the balance between orthogonality and redundancy, we explored the signal processing literature looking for other transforms that could provide a trade-off between correlation and time resolution, finding two candidates: the DTWT and the DDDWT. They both provide near-time-shift invariance and a moderated correlation (being the DTWT better than the DDDWT), while effectively doubling the sampling rate of the classical DWT.

Regarding the variance change detection algorithms, we identified the ICSS and SIC statistics as good candidates and provided an empirical assessment of their power in two different situations: uncorrelated and correlated processes. The performance comparison has been carried out in terms of the ratio of detected change points, its mean position and its dispersion. The most important conclusion is the important role of both the significance level at which the variance change test is performed and the variance change ratio. Regarding the goodness of each method we concluded that no technique is definitively better: for the uncorrelated case ICSS performs slightly better than

SIC at the lower variance ratios, while SIC outperforms ICSS at the medium and higher variance ratios. When the input signal is correlated SIC behaves better than ICSS, especially in terms of the amount of false changes. Since the alternative wavelet transforms we use in our methods (MODWT, DTWT) produce a correlated output, SIC seems to be better suited to our purposes than ICSS, but its performance is not as different as to abandon the latter.

The last three chapters of the thesis describe the performance of our method when the DWT, MODWT and DTWT are used with ICSS and SIC and the clustering and alignment method. Several tests have been performed on synthetic and real traffic traces in order to compare the results. Regarding the DWT case, the results for both ICSS and SIC are quite good [RS05a, RS05b, RMSP05b], though DWT-SIC performs slightly better (in terms of amount of detected changes and their localization) than DWT-ICSS for the more abrupt H changes, and vice versa. We have shown the blindness of the method to mean changes (which would correspond to traffic level shifts) and its sensitivity to variance and Hurst parameter changes. On the other hand, at the higher scales few changes were found due to the lack of resolution and the smoothing effect of the DWT variances. The results of the segmentation of the Bellcore dataset traces has found more variations than those usually reported in the literature, but are still coherent with those obtained with the constant-length window analysis. We have found no evidence of the correlation of traffic volume and H, in contradiction to the results obtained by other authors. Finally, a progressive version of our algorithm has been described. This modification can work either in cumulative or sliding window modes, and is the first step towards the implementation of a real-time version of the analyzer [RMCS04, RS05a, Min05, Per06].

Our first attempt to overcome the drawbacks of the DWT was to replace it with the MODWT, which has the nice property of being time-invariant and allows for an accurate location of the changes due to the high resolution at all scales. On the other hand, the decorrelation properties of the DWT are lost due to the redundant, non-orthogonal nature of the MODWT. The correlation increases the detection of false change points and must be corrected. We first tried with the Equivalent Degrees of Freedom (EDOF) approach, which failed [ZR06]. We then tried forcing the significance level and the minimum length to take unusual values. This approach can be mathematically questioned since the values have been derived heuristically, but the results are comparable to those found with the DWT [RS05c, RSdA05, RMS06]. For FGN traces MODWT-SIC performs slightly worse than DWT-SIC (in terms of a higher resolution or a lower quorum in order to get similar detection capability) but provides more valid changes at the higher scales.

Finally, we studied the Dual Tree Wavelet Transform (DTWT) due to its balance between a low (though not null) correlation and a better resolution than DWT [ZR06]. The complex nature of this transform presented an open question, since we could use the variance of both trees separately or computing the variance of the modulus of the complex coefficients [RZ06]. The second approach turned out to outperform DWT in terms of the localization of the changes, while its statistical power is still comparable to that of the DWT methods. In general, the DTWT-ICSS behaves better than DWT-ICSS and DTWT-SIC worse than DWT-SIC. We identified the non-Gaussianity of the DTWT modulus and the higher robustness of the ICSS to modifications of the input's distribution

as the reasons for this behavior. The results of the DTWT analysis of the Bellcore traces are comparable to the best ones obtained by the DWT and much better than those found with the MODWT.

One may argue about the validity of the values of the Hurst parameters found in the segments obtained by our algorithms, since sometimes we return what in principle can be considered as meaningless values ($H < 0$ or $H > 1$), but we have to consider that these strange values appear only when the segments are very short (thus reducing the statistical meaning of the estimation, which is highlighted in the large confidence intervals associated) or in the case of traffic volume non-stationarities. Nevertheless, we have proven that, in general, the results obtained by our segmentation methods are coherent with those found by the constant-length based methods.

To sum up, we have developed a family of wavelet-based segmentation algorithms capable of tracking the evolution of the scaling parameters of traffic across time that have been validated them with synthetic traces. The DWT-based segmenter returns good results with both ICSS and SIC, but is outperformed by the DTWT-ICSS. The methods have been applied to real traffic traces which were studied by other authors in a scaling-stationary fashion, finding a highly changing behavior and confirming the nonstationarity of their scaling parameters.

Topics for future research

What follows is a brief list of the most promising paths that can be followed from our previous research.

Non-Gaussian-aware algorithms

One of the problems we faced in our experiments was the non-normality of the wavelet coefficients. We identified the Generalized Gaussian Distribution as a better model for DWT and MODWT coefficients of real traces, and the Chi-square distribution for the modulus of the DTWT. Since the variance change detectors we use (ICSS and SIC) assume a Gaussian distribution at their input, we are aware of the fact that we walk a mathematically insecure path, though from the results obtained with synthetic tests traces we are reasonably sure of the correctness our methods.

To our knowledge, few works have considered the case of variance change detection in non-gaussian time series. Among them, the most important for our interests was carried out by Chen and Gupta [CG00], who studied the asymptotic distribution of the SIC statistic under *restricted non-normality* (it is assumed that the first four moments of the input random variable behave as those of the Gaussian distribution, conditions that seem to be rather technical and to not have an important influence in the final result) and found it to follow the same distribution as in the Gaussian input case. It would be interesting to try our methods with this generalized SIC statistic. Anyway, the expressions for the critical level of the generalized method are almost identical to those

developed for the Gaussian case. That is why we are reasonably sure of the results obtained with the Gaussianity assumption.

No equivalent study to the aforementioned has been found for the ICSS, but since both methods rely on the Cumulative Sum of Squares (CSS) statistic, we hypothesize that the behavior of the ICSS under non-normality will not deviate too much from the results obtained in the Gaussian case.

Extension of the LogScale Diagram to the alternative wavelet transforms

The Wavelet Packet Variance Map introduced in Chapter 3 was an extension of the LD to the Wavelet Packet transform. Though the expressions we provided for the WPT estimator are correct, the method remains to be tuned with the analytical derivation of the bias correction terms $g_{j,m}$ (where j is the scale index and m is the subband index) that were identified by Abry and Veitch as the solution for the inaccuracy introduced by the fact that the expectation of the logarithm is not equal to the logarithm of the expectation; $E[\log()] \neq \log(E[])$. However, the WPT results presented in Section 3.3.1 will only be slightly affected by this omission since, like g_j for the LD, the correction $g_{j,m}$ is small. Actually, in the first implementations of the LogScale Diagram the correction terms were neglected and the bias introduced was assumed [AV98]. Another open issue for the WPT estimator was the lack of implementation of a weighted curve-fitting algorithm[3] for the global WPT estimator (the one that uses all possible subbands for the estimation). The use of a weightless fitter redounded in some bias.

The development of an MODWT- or DTWT-based LD is also appealing (though mathematically complex), since it would allow us to estimate directly the scaling parameters of the detected segments, instead of first segmenting with the alternative wavelets and then performing the classical DWT-based LD estimation, as we do today. Of special interest would be the analytical expressions of the confidence intervals for the estimator.

Best-basis WPT-based segmenter

Our original idea regarding the use of the WPT-based estimator was to use it in order to compute in real time the optimal decomposition basis and therefore maximize the decorrelation properties of the wavelet analysis, and then perform the scaling parameters estimation [RS02]. Figure 8.26 illustrates the dynamical best basis selection. We already identified a fast splitting algorithm capable of performing a progressive orthogonal tiling developed by Sola and Sallent [SSR94] that could work jointly with the WPT-based estimator. Certain issues should be solved, being one of the most important the choice of the cost function of the best basis algorithm.

Regarding the the partial (in the sense of not using the full (m, j) dyadic subband set generated by the WPT) estimator, we already mentioned in Chapter 3 the possibility of using them as fast,

[3] actually, a 3-D surface-fitting algorithm.

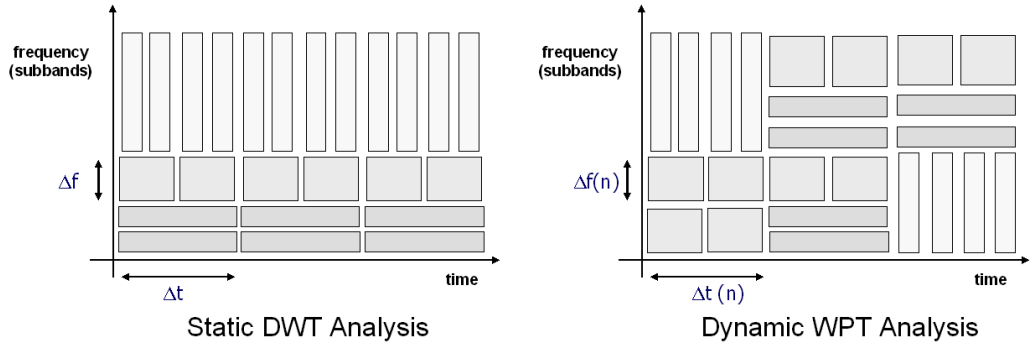

Figure 8.26: Comparison of the static DWT analysis (*left*) and the time-changing WPT base selection capability (*right*).

possibly not-so-accurate estimators. In the incomplete frequency decompositions the number of available samples for the regression is lower, a fact that speeds the process. Despite their lack of accuracy, these estimations could be useful in certain time-critical situations (where the speed of the estimation is more important than its precision), due to their less intensive computational load. This is another topic for future research.

Finally, we could consider the use of the MODWPT in order to develop a higher-time-resolution WPT variance map estimator, joining the time redundancy of the Maximal Overlap approach and the subband redundancy of the wavelet packets. We should, though, correct the time correlation that we already identified in the MODWT.

Real-time segmentation

Chapter 6 presented a first version of the progressive, near-real-time DWT-based estimator (see Section 6.10). The main problem we faced was the unpredictability of the amount of iterations of the variance change detection algorithms that were needed for a certain trace, and the fact that a criterium should be defined in order to stop the computation in case it was taking too much time to run in real time. Though our Matlab implementation of ICSS and SIC was highly optimized, a C/C++ version could be further improved by a skilled programmer, and the same goes for the Hough transform and its clustering and alignment algorithm (though this one is run just once per estimation). It would also be interesting to perform a detailed study of the *detection delay* in which our method incurs, i.e. the amount of *lookahead samples*[4] that the method needs in order to detect a change.

Another open question is the optimal length of the sliding window that goes over the trace, in order to minimize the risk of not detecting true change points. Finally, we should investigate the distribution of the lengths of the variance- and H-constant segments in real traffic traces, and

[4]We borrowed this term from the audio coding field, where it denotes the amount of *future* audio samples needed in order to efficiently compress the present audio frame in vocoders (such as G.723.1) and perceptual codecs (such as MP3).

characterize it. This would also give us some clues about how to customize or *tune* the algorithm's parameters.

All these methods should be implemented in a real system equipped with either a standard network card or a dedicated, high performance monitoring systems such as the Endace DAG® cards[5]. Some of the methods presented in this thesis have been (partially) implemented in standard Gigabit Ethernet cards [RMCS04] (with which we managed to analyze and generate fractal traffic at 500-600 Mbit/s) and a couple of DAG cards capable of working at full 1 Gbit/s speed [Per06].

Other wavelet transforms for the time-segmenting method

The wavelet-based signal processing field is an exciting and very active research area which still produces relevant contributions. Apart from the MODWT and the DTWT, other wavelet transforms appear as promising candidates to be used in our algorithm:

- **The Double Density Discrete Wavelet Transform (DDDWT)** [Sel01] is similar to the DTWT in time resolution (twice that of the DWT), near shift-invariance, and a reduced correlation [Sel01, SA04]. Appendix A provides a brief introduction to the DDDWT and its statistical properties. We prioritized the use of the DTWT due to its superiority regarding shift-invariance and correlation, but we think the DDDWT deserves some attention in order to assess its behavior with scaling processes.

- **The Double Density Dual Tree Discrete Wavelet Transform (DDDTDWT)** [Sel04], also developed by Ivan Selesnick, mixes the Dual Tree and the Double Density concepts. It is a dyadic wavelet tight frame based on two scaling functions and four wavelet filters. The output of the DDDTDWT is oversampled by a factor of 4 (2 due to the double density and 2 due to the complex dual tree), and is almost shift-invariant. We could compute the modulus of the complex output while still maintaining the double density resolution.

- **The Higher Density DWT or Expansive DWT** [Sel06]. Yet another of Selesnick's invention, the Expansive DWT is an almost shift-invariant, tight dyadic wavelet frame with two generators that doubles the sampling rate both in time and frequency, thus creating not only *intermediate time samples* (as the DDDWT does) but also *intermediate scales*, which in our opinion could be a very good way to improve the accuracy of the LogScale Diagram by having more resolution at both axes of the time-frequency plane. Figure 8.27 illustrates the tiling produced by the Expansive DWT.

For each of these carrying at least a preliminary study (including its sensitivity to mean, variance and Hurst parameter changes, together with an empirical assessment of the power of the test) would be worth trying.

[5] `http://www.endace.com`

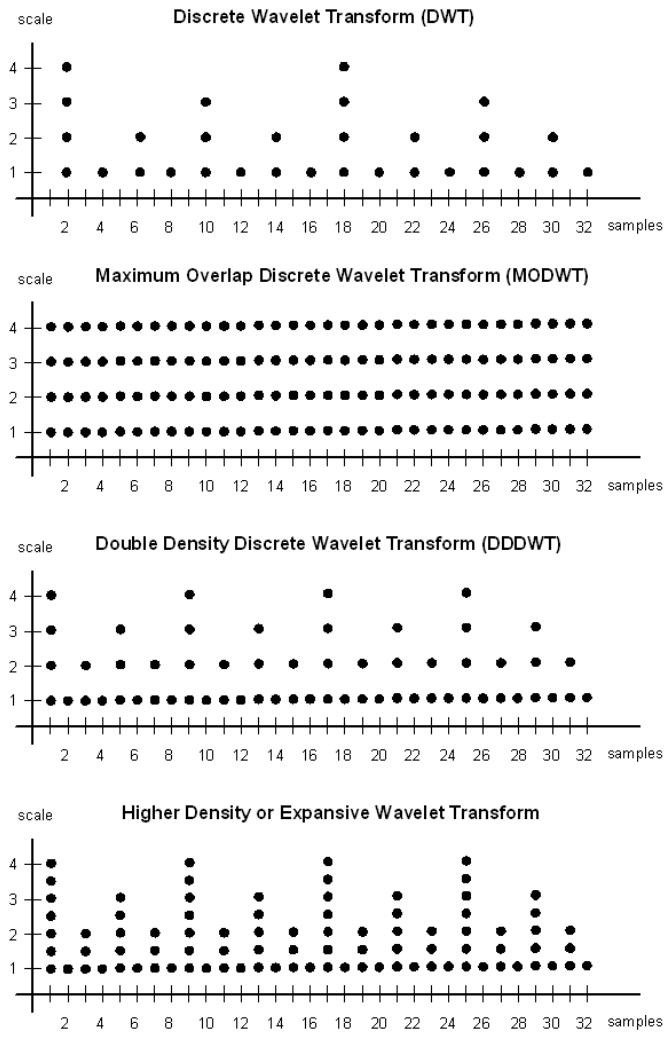

Figure 8.27: Comparison of the time-frequency plane sampling performed by the DWT, the MODWT, the DDDWT and the Higher density or Expanded DWT.

Extension to multifractal processes

Abry and Veitch extended their LogScale Diagram to multifractal processes [VA] by generalizing the variance computation to the q^{th} moment, and performing a linear regression in each q. In theory, we could easily extend our segmentation methods by performing q parallel q^{th} change detection multiscale procedures and aligning the results, but to our knowledge no one has yet tried developing higher moments' change detectors.

Since both ICSS and SIC rely on the cumulative sum of squares (CSS) statistic, it seems straightforward to generalize it to a cumulative (absolute) sum of the q^{th} moment. The critical level and significance values should be obtained by Montecarlo simulations, since no analytical work seems to have addressed this topic.

Atzori et al [AAI05, AAI06] recently developed an accuracy-based approach for the real-time estimation of multifractality, where the window length is variable and related to the confidence interval of the estimation. The method accumulates samples until the confidence interval of the multifractal estimator is below a certain value determined by the user, and then the estimation is performed. The window length automatically adapts to the statistical properties of data. Comparing the performance of this method and ours (in the case of monofractal traffic) would be interesting. Besides, the accuracy-based segmentation could be applied to our multiresolution analysis with the alternative wavelet transforms, and check the possible improvement of the results.

Appendix A

The Double Density Discrete Wavelet Transform (DDDWT)

A.1 Another *framelet* with redundancy 2

The Dual Density Wavelet Transform was recently developed by Ivan Selesnick [Sel01, SA04]. Technically, it is a *symmetric wavelet tight frame with two generators*. The main properties of this transform are:

- Its tight nature ensures that Parseval's Theorem holds and an ANOVA can be performed;
- It is almost shift-invariant, though in this aspect is not as good as the Dual Tree (Complex) Wavelet Transform (DTWT);
- It is overcomplete (redundant) by a factor of 2, independent of the number of scales over which the signal decomposition is performed;
- It employs one scaling function and two different wavelets $\phi_1(t)$ and $\phi_2(t)$, which are offset by half a sample.

In terms of redundancy, it is similar to the DTWT designed by Kingsbury (see Section 2.8), since it outputs twice the samples present at the input. However, its time-invariance is not as good as that of the DTWT.

A.2 Filter bank implementation

The basic block for the pyramidal implementation of the DDDWT is a three-channel filter bank with a low-pass output and two band-pass and high-pass subbands, as shown in Figure A.1. The

low-pass branch corresponds to a scaling function $\phi(t)$ (filter $h(k)$), while the band-pass and high-pass branches are related to two mother wavelets $\psi_1(t)$ and $\psi_2(t)$ (filters $g_1(k)$ and $g_2(k)$). This basic filter bank is iterated in the low-pass output, in a similar way as the DWT works. Figure A.2 shows the full filter bank implementation of the DDDWT.

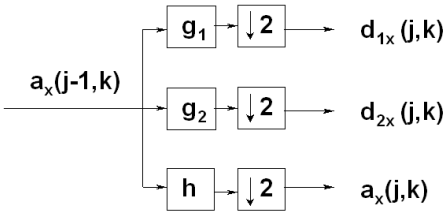

Figure A.1: The basic filter bank of the DDDWT.

Note that the basic filter bank is oversampled by a factor of 3/2, not by 2. Therefore, why should the transform be called *double density*? This is because the iteration makes the redundancy factor approach 2: a $J = 2$ filter bank is oversampled by 7/4, a $J = 3$ filter bank is oversampled by 15/8, and so on. A general decomposition at level J is oversampled by $\frac{2J-1}{J} = 2 - \frac{1}{J}$, and as $J \to \infty$, the factor tends to 2.

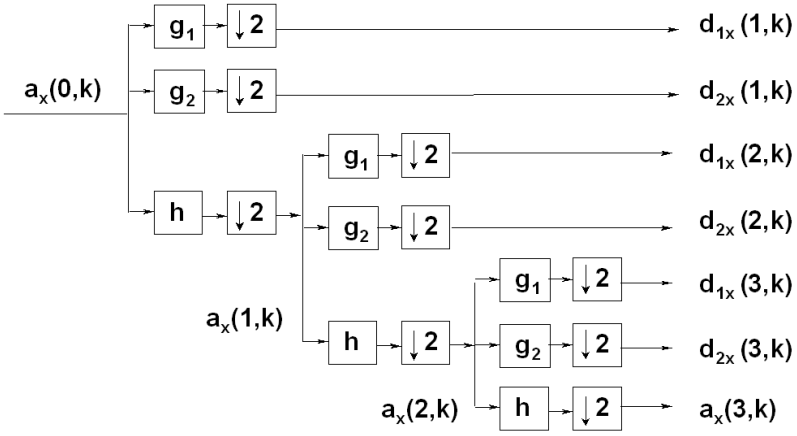

Figure A.2: The filter bank implementation of the DDDWT.

A.3 DDDWT filters

The description of the construction of the DDDWT filters and their associated wavelets and scaling function is out of the scope of this document; those interested can consult the details in [SA04]. The three-filter bank has more free parameters than the usual two-filter bank, thus expanding the possibilities for designing the wavelets. We will only comment that the options explored by Selesnick include some design constraints such as the perfect reconstruction property / energy preserving condition, and the presence of some symmetries (such as $\psi_1(t) = \psi_2(N - t)$, where

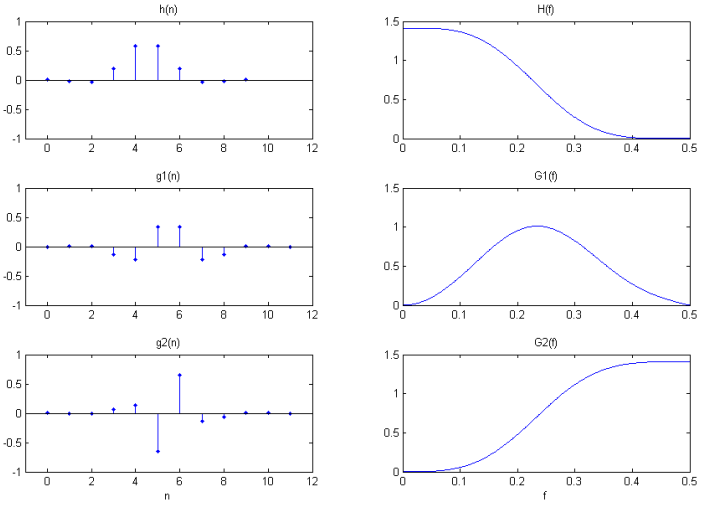

Figure A.3: An example of a DDDWT filter set. Impulse response (left) and frequency response (right) for the low-pass scaling filter $g(n)$, and the band-pass and high-pass wavelet filters $h_1(n)$ and $h_2(n)$.

N is the support of the wavelet) which make the design easier. Other choices include imposing the same constraints used for building the Daubechies DWT wavelets: minimal length under the constraint that up to K moments equal zero. Of course, now both wavelets have to verify the conditions, instead of having just one wavelet. This ensures the *blindness to polynomial trends* that we mentioned in Section 2.4.3. Figure A.3 illustrates the filter set corresponding to a DDDWT of the first type, with symmetrical wavelet mothers, as can be seen in Figure A.4.

Figure A.5 shows an example of the output of the first stage of the filter bank of the DDDWT, using the filters displayed in Figure A.3. The test signal is composed of two steps surrounded by zero level. It can be seen that intuitively the behavior is similar to other wavelet transforms: the

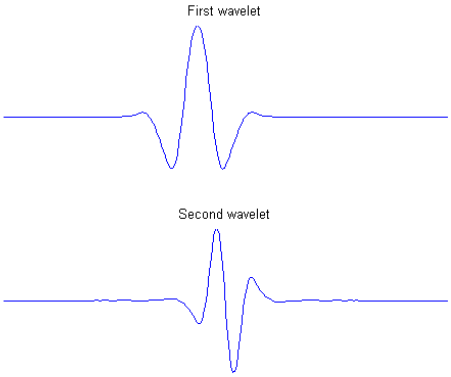

Figure A.4: The $\psi_1(t)$ and $\psi_2(t)$ wavelets corresponding to the DDDWT filter set shown in Figure A.3.

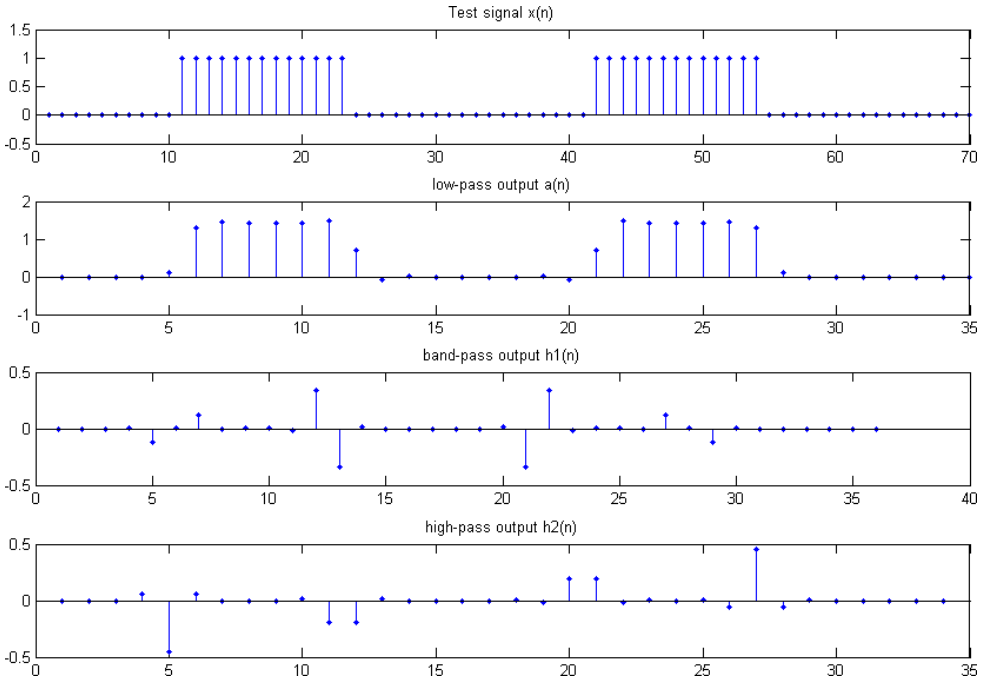

Figure A.5: A test signal $x(n)$ and its DDDWT analysis at the first stage of the filter bank.

approximation coefficients keep the lowest frequency components of the signal (the steps), while the wavelet coefficients match the higher frequency components (step boundaries). Figure A.6 shows another example, with a full $J = 3$ decomposition of a signal composed by some segments including zero level, a ramp (with a smooth, low-frequency component and an abrupt end) and a step. Note that the length of the original signal is $N = 128$, while the total number of wavelet coefficients is $N + \frac{N}{2} + \frac{N}{8} + \frac{N}{8} = \frac{15N}{8} = 240$.

A.4 MRA and ANOVA

As happens with the other wavelet transforms, the DDDWT is able to perform a multiresolution analysis, with the only difference being that two details are generated for each scale. Therefore, a signal $X(t)$ is decomposed in terms of an approximation at level J, $A_J(t)$ and a set of details $D_{j,1}(t)$ and $D_{j,2}(t)$. $D_{j,1}(t)$ is related to the band-pass wavelet $\psi_1(t)$, while $D_{j,2}(t)$ corresponds to the high-pass wavelet $\psi_2(t)$.

$$X(t) = A_J(t) + \sum_{j=1}^{J} (D_{j,1}(t) + D_{j,2}(t)) \tag{A.1}$$

Similarly to the DWT and MODWT, the j^{th} detail of $X(t)$ corresponds to the information that is removed when going from one approximation level to the next: $D_{j,1}(t) + D_{j,2}(t) = A_{j-1}(t) - A_j(t)$.

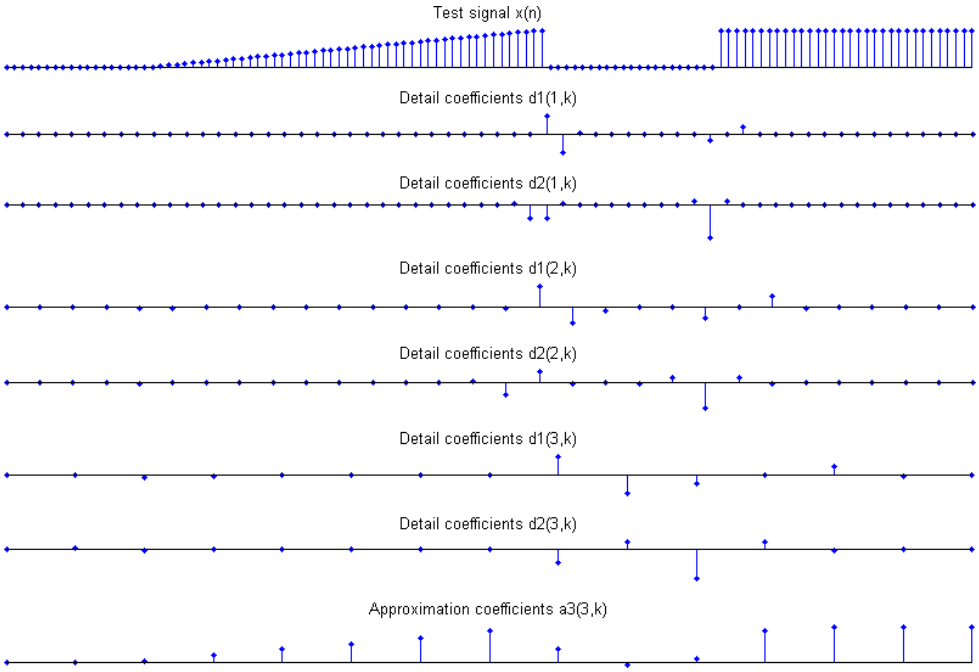

Figure A.6: A test signal $x(n)$ and its DDDWT analysis at level $J = 3$.

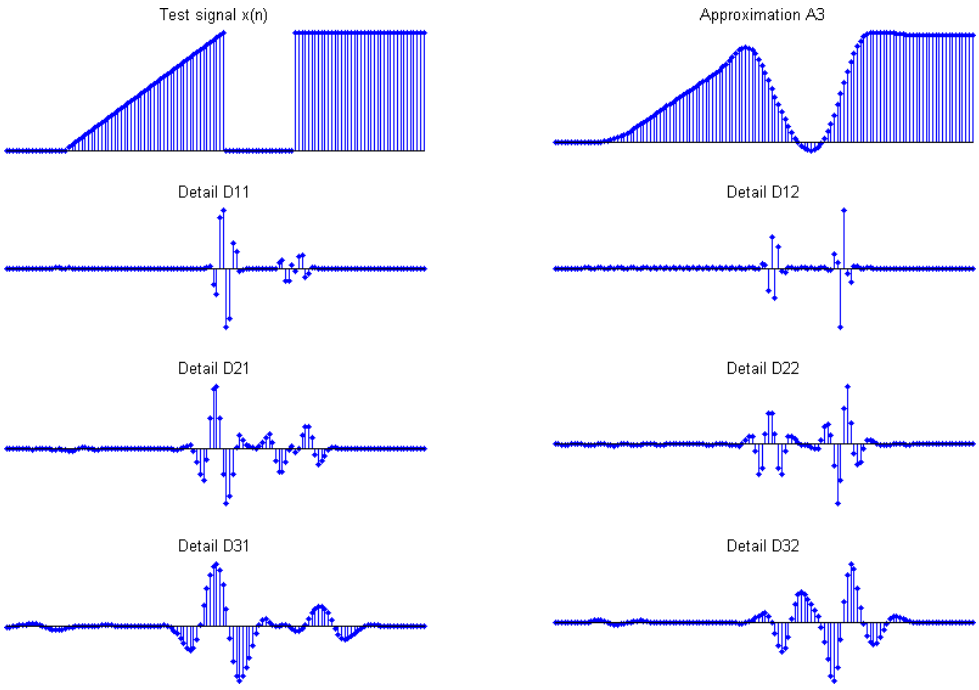

Figure A.7: Example of an MRA decomposition with the DDDWT.

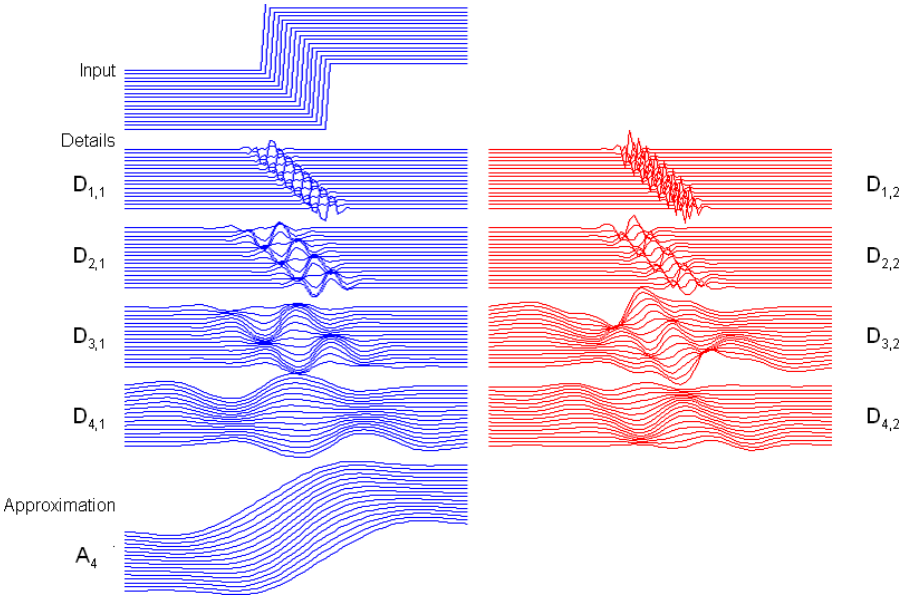

Figure A.8: Illustration of the near-shift-time invariance of the DDDWT. Sixteen time-shifted versions of a step function are analyzed and decomposed into a set of details at levels $j = 1 \ldots 4$ and an approximation at level $J = 4$.

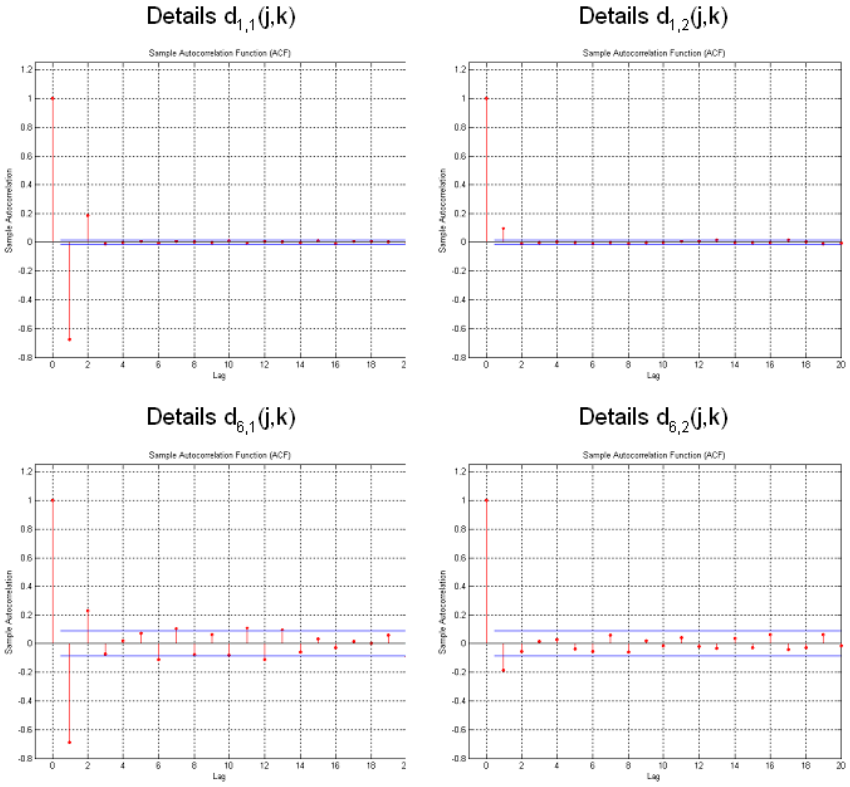

Figure A.9: Autocorrelation functions of DDDWT detail coefficients of a 32768-sample white noise signal.

Equation A.1 can then be rewritten as

$$\begin{aligned} X(t) &= D_{1,1}(t) + D_{1,2}(t) + A_1(t) \\ &= D_{1,1}(t) + D_{1,2}(t) + D_{2,1}(t) + D_{2,2}(t) + A_2(t) \\ &= D_{1,1}(t) + D_{1,2}(t) + D_{2,1}(t) + D_{2,2}(t) + D_{3,1}(t) + D_{3,2}(t) + A_3(t) \\ &... \\ &= \sum_{j=1}^{J} [D_{j,1}(t) + D_{j,2}(t)] + A_J(t) \end{aligned}$$

Regarding the analysis of variance, the situation is similar to that found with MODWT. An ANOVA can be performed on the transform coefficients $a_{j,k}$, $d_{j,1,k}$ and $d_{j,2,k}$, and not on the details and approximations.

$$\|X\|^2 = \sum_k a_{J,k}^2 + \sum_{j=1}^{J} \sum_k [d_{j,1,k}^2 + d_{j,2,k}^2] \neq \|A_J\|^2 + \sum_{j=1}^{J} [\|D_{j,1}\|^2 + \|D_{j,2}\|^2] \tag{A.2}$$

Figure A.7 shows an MRA decomposition of the same signal used in Figure A.6.

A.5 Shift invariance

As aforementioned, the shift invariance properties of the DDDWT are not as good as those of the MODWT (that is fully shift-invariant) and are slightly worse than those of the DTWT. Figure A.8 shows the same analysis performed for the DTWT in Section 2.8.4.

A.6 Correlation

Regarding the correlation issue, the results provided by the DDDWT are also a bit worse than those provided by the DTWT, but much better than those of the MODWT. Figure A.9 shows the autocorrelation function of the detail coefficients of the DTWT of a white noise signal. An interesting feature is that $d_{i,2}(j,k)$ seem to be less correlated than $d_{i,1}(j,k)$.

Acronyms

ADSL	Asymmetric Digital Subscriber Line
ANOVA	Analysis of Variance
ATM	Asynchronous Transfer Mode
CAC	Connection Admission Control
CIPP	Correlated Interarrival Poisson Process
CSS	Cumulative Sum of Squares
CWT	Continuous Wavelet Transform
DDDTDWT	Double Density Dual Tree (Complex) Discrete Wavelet Transform
DDDWT	Double Density Discrete Wavelet Transform
DFT	Discrete Fourier Transform
DTWT	Dual Tree (Complex) Discrete Wavelet Transform
DWT	Discrete Wavelet Transform
DWPT	Discrete Wavelet Packet Transform
EDOF	Equivalent Degrees of Freedom
FBM	Fractional Brownian Motion
FGN	Fractional Gaussian Noise
FIR	Finite Impulse Response (filter)
FTP	File Transfer Protocol
GGD	Generalized Gaussian Distribution
HP	High Pass
H-SS	Self-Similar process with Hurst parameter H
H-SSSI	Self-Similar process with Stationary Increments with Hurst parameter H
ICSS	Iterated Cumulative Sum of Squares
IP	Internet Protocol
LA	Least Asymmetrical (wavelets)
LAN	Local Area Network
LD	LogScale Diagram
LP	Low Pass
LRD	Long-Range Dependent (process)
mFBM	Multifractional Brownian Motion
mFGN	Multifractional Gaussian Noise

MLE	Maximum Likelihood Estimator
MODWPT	Maximal Overlap Discrete Wavelet Packet Transform
MODWT	Maximal Overlap Discrete Wavelet Transform
MMPP	Markov Modulated Poisson Processes
MPEG	Moving Picture Experts Group
MRA	Multiresolution Analysis
MVUE	Minimum Variance Unbiased Estimator
NIC	Network Interface Card
P2P	Peer to Peer
PDF	Probability Density Function
PDF	Power Spectral Density
QoS	Quality of Service
RTT	Round Trip Time
SIC	Schwarz Information Criterion
SRD	Short-Range Dependent (process)
SS	Self-Similar (process)
SSSI	Self-Similar (process) with Stationary Increments
STFT	Short Time Fourier Transform
SWT	Stationary Wavelet Transform
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VBR	Variable Bit Rate
VCD	Variance Change Detection
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WPT	Wavelet Packet Transform
WWW	World Wide Web

Bibliography

[AAI05] Luigi Atzori, Nicola Aste, and Mauro Isola. Estimation of Multifractal Parameters in Traffic Measurement: an Accuracy-based Real-time Approach. In *Proceedings of the IEEE International Conference on Communications, ICC 2005*, volume 1, pages 21–25, May 2005.

[AAI06] Luigi Atzori, Nicola Aste, and Mauro Isola. Estimation of Multifractal Parameters in Traffic Measurement: An Accuracy-based Real-time Approach. *Computer Communications*, 29(11):1879–1888, July 2006.

[AF94] P. Abry and P. Flandrin. On the Initialization of the Discrete Wavelet Transform. *IEEE Signal Processing Letters*, 1(2):32–34, 1994.

[AFTV00] P. Abry, P. Flandrin, M.S. Taqqu, and D. Veitch. *Self-similar Traffic and Network Performance*, Chapter Wavelets for the Analysis, Estimation and Synthesis of Scaling Data. John Wiley & Sons, 2000.

[Aka73] Hirotugu Akaike. Information Theory and an Extension of the Maximum Likelihood Principle. In *Proceedings of the* 2^{nd} *International Symposium on Information Theory*, pages 267–281, 1973.

[AV98] P. Abry and D. Veitch. Wavelet Analysis of Long-range Dependent Traffic. *IEEE Transactions on Information Theory*, 44:2–15, 1998.

[Ber94] J. Beran. *Statistics for Long-memory Processes*. Chapman and Hall, 1994.

[BSTW95] Jan Beran, Robert Sherman, Murad S. Taqqu, and Walter Willinger. Long-range Dependence in Variable-bit-rate Video Traffic. *IEEE Transactions on Communications*, 43(2/3/4):1566–1579, February/March/April 1995.

[CB97] M. Crovella and A. Bestavros. Self-similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, December 1997.

[CCLS01] Jin Cao, William S. Cleveland, Dong Lin, and Don X. Sun. On the Nonstationarity of Internet Traffic. In *ACM SIGMETRICS Performance Review*, volume 29, pages 102–112, 2001.

[CG97] Jie Chen and A.K. Gupta. Testing and Locating Variance Change Points with Application to Stock Prices. *Journal of the American Statistical Association*, 92(438):739–747, June 1997.

[CG00] Jie Chen and A. K. Gupta. Detecting Change Point for a Sequence of Random Vectors under Nonnormality. *Random Operators and Stochastic Equations*, 8(2):127–142, 2000.

[Coh92] A. Cohen. *Ondelettes, Analyses Multirésolution et Traitement Numérique du Signal*. PhD thesis, University of Paris IX, 1992.

[CW90] R. Coifman and W. Wickerhauser. Best-adapted Wave Packet Basis (technical report). Numerical Algorithms Research Group, Department of Mathematics, Yale University, 1990.

[CW92] R. Coifman and W. Wickerhauser. Entropy-based Algorithms for Best Basis Selection. *IEEE Transactions on Information Theory*, 38(2):713–718, 1992.

[Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

[DLO+94] N.G. Duffield, J.T. Lewis, N. O'Connell, R. Russell, and F. Toomey. Statistical Issues Raised by the Bellcore Data. In *Proceedings of the 11th IEE UK Teletraffic Symposium*, March 1994.

[DM99] Trang Dinh and Sándor Molnár. On the Effects of Non-stationarity in Long-range Dependence Tests. *Periodica Polytechnica*, 43(4):227–250, 1999.

[DMGFRD01] J. Armando Domínguez-Molina, Graciela González-Farías, and Ramón M. Rodríguez-Dagnino. A Practical Procedure to Estimate the Shape Parameter in the Generalized Gaussian Distribution. Technical report CIMAT I-01-18, Centro de Investigación en Matemáticas, Mexico (CIMAT), September 2001.

[DMM03] Trang Dinh Dang, Sándor Molnár, and István Maricza. Some Results on Multiscale Queueing Analysis. In *Proceedings of IEEE International Conference on Telecommunications (ICT)*, pages 1631–1638, March 2003.

[DMRW94] D. E. Duffy, A. A. McIntosh, M. Rosenstein, and W. Willinger. Statistical Analysis of CCSN/SS7 Traffic Data from Working CCS Subnetworks. *IEEE Journal of Selected Areas in Communications*, 12(3):544–551, April 1994.

[Do] Minh N. Do. Matlab Implementation of the Moment Matching and Maximum Likelihood Estimators for the Generalized Gaussian Density. University of Illinois at Urbana-Champaign (USA). Avaliable at http://www.ifp.uiuc.edu/∼minhdo/software/.

[EC05] H. Elbiaze and O. Cherkaoui. Exploiting Self-similar Traffic Analysis in Network Resource Control: the IP over WDM Networks Case. In *Proceedings of Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services, IEEE ICAS/ICNS*, page 65, October 2005.

[FG95] Y. Fan and N. Georganas. On Merging and Splitting of Self-similar Traffic in High-speed Networks. In *Proceedings of ICCC'95*, pages 8A.1.1–6, 1995.

[FGHW99] A. Feldmann, A. C. Gilbert, P. Huang, and W. Willinger. Dynamics of IP Traffic: a Study of the Role of Variability and the Impact of Control. In *Proceedings of ACM SIGCOMM 99*, pages 301–313, 1999.

[GB99] Matthias Grossglauser and Jean-Chrysostome Bolot. On the Relevance of Long-range Dependence in Network Traffic. *IEEE/ACM Transactions on Networking*, 7(5):629–640, 1999.

[Gib96] R. J. Gibbens. *Stochastic Networks: Theory and Applications*, chapter Traffic Characterization and Effective Bandwidths for Broadband Network Traces, pages 169–179. Oxford University Press, 1996.

[Glo05] Przemyslaw Glomb. Analysis of FGN and HTTP Requests Traces using Localized Multiscale H Parameter Estimation. In *Proceedings of the IEEE International Symposium on Applications and the Internet SAINT 2005*, pages 288–291, January 2005.

[GW94] Mark Garrett and Walter Willinger. Analysis, Modeling, and Generation of Self-similar VBR Video Traffic. In *Proceedings of ACM SIGCOMM 1994*, pages 269–280, August 1994.

[GWF99] A.C. Gilbert, Walter Willinger, and Anja Feldmann. Scaling Analysis of Conservative Cascades, with Application to Network Traffic. *IEEE Transactions on Information Theory*, 45(3):971–991, April 1999.

[HGHP02] G. He, Y. Gao, J.C. Hou, and K. Park. A Case for Exploiting Self-similarity of Network Traffic in TCP Congestion Control. In *Proceedings of IEEE ICNP 2002*, pages 34–43, November 2002.

[HKS99] Helmut Hlavacs, Gabriele Kotsis, and Christine Steinkellner. Traffic Source Modeling (technical report). Institute of Applied Computer Science and Information Systems, University of Vienna, 1999.

[HL86] H. Heffes and D. M. Lucantoni. A Markov-modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance. *IEEE Journal on Selected Areas in Communications*, 4(6):856–868, 1986.

[HL96] D. Heyman and T. Lakshman. Source Models for VBR Broadcast Video Traffic. *IEEE/ACM Transactions on Networking*, 4(1):40–48, February 1996.

[HMGBH06] Gerhard Hasslinger, Joachim Mende, Rudiger Geib, Thomas Beckhaus and Franz Hartleb. *U. Krieger, editor, Achievements on IP Traffic Characterization, Measurements and Statistical Methods. EuroNGI Deliverable D.WP.JRA.5.1.3*, chapter 6. Measurement and Characteristics of Aggregated Traffic in Broadband Access Networks, pages 68–78. EuroNGI Network of Excellence, November 2006.

[IT94] C. Inclán and G.C. Tiao. Use of Cumulative Sums of Squares for Retrospective Detection of Changes of Variance. *Journal of the American Statistical Association*, 89(427):913–923, September 1994.

[JD03] H. Jiang and C. Dovrolis. Source-level IP Packet Bursts: Causes and Effects. In *Proceedings of ACM SIGCOMM/Usenix Internet Measurement Conference*, October 2003.

[JD05] H. Jiang and C. Dovrolis. Why is the Internet Traffic Bursty in Short Time Scales? *ACM SIGMETRICS Performance Evaluation Review*, 33(1):241–252, February 2005.

[JLC04] S. Joo, C. Lee, and Y. Chung. Analysis and Modeling of Traffic from Residential High Speed Internet Subscribers. In *Proceedings of ICOIN 2004*, pages 410–419, February 2004.

[KGD03] P. Muraly Krishna, Vikram M. Gadre, and Uday B. Desai. *Multifractal Based Network Traffic Modeling*. Kluwer Academic Publishers, 2003.

[Kin00] Nick Kingsbury. Complex Wavelets and Shift Invariance. In *Proceedings of the IEE Colloquium on Time-Scale and Time-Frequency Analysis and Applications*. IEE, February 2000.

[Kin01] N.G. Kingsbury. Complex Wavelets for Shift Invariant Analysis and Filtering of Signals. *Journal of Applied and Computational Harmonic Analysis*, 10(3):234–253, May 2001.

[Kin04] Nick Kingsbury. Tutorial on Dual-tree Complex Wavelets. given to HASSIP Workshop, Cambridge, September 2004.

[Kin05] Nick Kingsbury. Matlab Code for the Dual Tree Wavelet Transform (version 4.3). Personal communication, December 2005.

[Kle76] Leonard Kleinrock. *Queueing Systems*. John Wiley & Sons, 1975-1976.

[KMFB04] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido. A Nonstationary Poisson View of Internet Traffic. In *Proceedings of IEEE INFOCOM 2004*, volume 3, pages 1558–1569, 2004.

[LTWW94] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the Self-similar Nature of Ethernet Traffic. *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.

[LW91] Will E. Leland and Daniel V. Wilson. High Time-resolution Measurement and Analysis of LAN Traffic: Implications for LAN Interconnection. In *Proceedings of IEEE INFOCOM 91*, pages 1360–1366, 1991.

[Mal89] S. Mallat. A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. *IEEE Pattern Analysis and Machine Intelligence Journal*, 11(7):674–693, July 1989.

[Mal98] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[Man83] Benoit Mandelbrot. *The fractal geometry of Nature*. W.H. Freeman and Company, 1983.

[Mar02] Evangelos P. Markatos. Tracing a Large-scale Peer to Peer System: an Hour in the Life of Gnutella. In *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2002.

[MD00] Sándor Molnár and Trang Dinh Dang. Pitfalls in Long Range Dependence Testing and Estimation. In *Proceedings of IEEE Globecom 2000*, volume 1, pages 662–666, November 2000.

[MDK99] R. Manivasakan, U. B. Desai, and A. Karandikar. CIPP Based Characterization of VBR Traffic and Related Statistical Multiplexer Performance. In *Proceedings of IEEE Globecom'99*, pages 5–9, December 1999.

[Min05] Flaminio Minerva. Wavelet Analysis of Long-range Dependent Traffic. Master's thesis supervised by David Rincón, Faculty of Telecommunications Engineering (Università di Pisa) and Technical School of Castelldefels (Universitat Politècnica de Catalunya), July 2005.

[MK98] Julian Magarey and Nick Kingsbury. Motion Estimation Using a Complex-valued Wavelet Transform. *IEEE Transactions on Signal Processing*, 46(4):1069–84, April 1998.

[MMOP02] Michel Misiti, Yves Misiti, Georges Oppenheim, and Jena-Michel Poggi. *Matlab Wavelet Toolbox*. The MathWorks, Inc., version 2.2, July 2002.

[MN68] B. Mandelbrot and J. Van Ness. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review*, (10):422–437, 1968.

[Mol04] Sándor Molnár. Tutorial on Traffic Analysis and Modeling. EuroNGI Summer School, University of Bamberg, Germany, October 2004.

[MV02] N. D. Minh and Martin Vetterli. Wavelet-based Texture Retrieval Using Generalized Gaussian Density and Kullback-Leibler Distance. *IEEE Transactions on Image Processing*, 11(2):146–158, 2002.

[Nor94] Ilkka Norros. A Storage Model with Self-similar Input. *Queueing systems*, 16:387–396, 1994.

[OY01] Yen-Chieh Ouyang and Li-Bin Yeh. Predictive Bandwidth Control for MPEG Video: a Wavelet Approach for Self-similar Parameters Estimation. In *Proceedings of the IEEE International Conference in Communications, ICC 2001*, volume 5, pages 1551–1555, June 2001.

[Peh97a] Jon Peha. Protocols Can Make Traffic Appear Self-similar. Technical report, Carnegie Mellon University, 1997.

[Peh97b] Jon Peha. Retransmission Mechanisms and Self-similar Traffic Models. In *Proceedings of IEEE/ACM/SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 47–52, January 1997.

[Per06] Iván Pérez. Analizador/Generador Gigabit de Altas Prestaciones (in Spanish). Master's thesis supervised by David Rincón, Technical School of Castelldefels (Universitat Politècnica de Catalunya), July 2006.

[PF95] Vern Paxson and Sally Floyd. Wide Area Traffic: the Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[PKC96a] Kihong Park, Gitae Kim, and M. Crovella. On the Relationship Between File Sizes, Transport Protocols and Self-similar Network Traffic. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP 96)*, volume 171-180, pages 171–180, October-November 1996.

[PKC96b] J-C. Pesquet, H. Krim, and H. Carfantan. Time Invariant Orthonormal Wavelet Representations. *IEEE Transactions on Signal Processing*, 44(8):1964–1970, August 1996.

[PM96] M. Parulekar and A. Makowski. Tail Probabilities for a Multiplexer with Self-similar Traffic. In *Proceedings of IEEE INFOCOM'96*, pages 1452–1459, 1996.

[PTZD05] D. Papagiannaki, N. Taft, Z. Zhang, and C. Diot. Long-term Forecasting of Internet Backbone Traffic. *IEEE Transactions on Neural Networks*, 16(5):1110–1124, September 2005.

[PV95] R.F. Peltier and J.L. Vehel. Multifractional Brownian Motion: Definition and Preliminary Results. INRIA Technical Report RR-2645, 1995.

[PW00] Kihong Park and Walter Willinger. *Self-similar Traffic and Network Performance*, chapter Self-similar Network Traffic: an Overview. John Wiley & Sons, 2000.

[PW02] Donald Percival and Andrew Walden. *Wavelet Methods fot Time Series Analysis*. Cambridge University Press, 2002.

[QF99] Pu Qian and Bruce A. Francis. *Topics in Control and its Applications*, chapter Solution of a Wavelet Crime, pages 143–156. Springer-Verlag, 1999.

[RCRB99] R. H. Riedi, M. S. Crouse, V. J. Ribeiro, and R. G. Baraniuk. A Multifractal Wavelet Model with Application to Network Traffic. *IEEE Transactions on Information Theory*, 45(4):992–1018, 1999.

[RGK+03] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang. Experience in Measuring Internet Backbone Traffic Variability: Models, Metrics, Measurements and Meaning. In *Proceedings of the International Teletraffic Congress (ITC-18)*, 2003.

[Rin05] David Rincón. *U. Krieger, editor, IP Traffic Characterization, Measurements and Statistical Methods - A Description of the Current State. EuroNGI Deliverable D.WP.JRA.5.1.1*, chapter 4.3 Reconstruction of LRD and self-similar traffic by wavelet techniques, pages 84–92. EuroNGI Network of Excellence, May 2005.

[RMCS04] David Rincón, Sergi Martínez, Cristina Cano, and Sebastià Sallent. Synthesis and analysis of fractal LAN traffic at high speeds In *Proceedings of LANMAN 2004, the 13th IEEE Workshop Local and Metropolitan Area Networks*, pages 259–264, April 2004.

[RMS06] David Rincón, Flaminio Minerva, and Sebastià Sallent. Segmentation of LRD Traffic with Wavelets and the Schwarz Information Criterion. In *Proceedings of the Passive and Active Measurement workshop (PAM 2006) (to be published)*, March 2006.

[RMSP05a] David Rincón, Flaminio Minerva, Sebastià Sallent, and Michele Pagano. Segmentación Temporal de Tráfico Fractal mediante Transformadas Wavelet Redundantes y Teoría de la Información (in Spanish). In *Proceedings of JITEL 2005 - V Jornadas de Ingeniería Telemática*, pages 621–628, September 2005.

[RMSP05b] David Rincón, Flaminio Minerva, Sebastiá Sallent, and Michele Pagano. Towards an Improved Characterization of Fractal Network Traffic. In *Proceedings of the* 3^{rd} *International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks - HetNets 2005*, pages WP041–WP046, July 2005.

[RS02] David Rincón and Sebastià Sallent. Progressive Real-time Estimation of Multifractal Network Traffic. In *Proceedings of the 12th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2002)*, pages 127–131, August 2002.

[RS04a] David Rincón and Sebastià Sallent. Caracterización de Tráfico Mediante Transformadas Wavelet Redundantes (in Spanish). In *Proceedings of URSI2004 - XIX Simposium Nacional de la Unin Cientfica Internacional de Radio*, pages 1–8, September 2004.

[RS04b] David Rincón and Sebastià Sallent. Characterizing Fractal Traffic with Redundant Wavelet-based Transforms. In *Proceedings of the First Workshop New Trends in Modelling, Quantitative Methods and Measurements - Design and Engineering of the Next Generation Internet*, pages 361–371, June 2004.

[RS05a] David Rincón and Sebastià Sallent. On-line Segmentation of Non-stationary Fractal Network Traffic with Wavelet Transforms and Log-likelihood-based Statistics. In *Proceedings of the* 3^{rd} *International Workshop on QoS in Multiservice IP Networks, QoSIP 2005. Also published in Lecture Notes in Computer Science 3375, Springer-Verlag*, pages 110–123, January 2005.

[RS05b] David Rincón and Sebastià Sallent. On the Use of the Schwarz Information Criterion for Analysis and Segmentation of Non-stationary LRD Traffic. In *Proceedings of the IEEE International Symposium on Applications and the Internet SAINT 2005*, January 2005.

[RS05c] David Rincón and Sebastià Sallent. Segmentation of Non-stationary Fractal Network Traffic with Wavelet Transforms and Maximum-likelihood Statistics. In *Proceedings of ICC 2005 - The IEEE International Conference on Communications*, volume 1, pages 11–15, May 2005.

[RSed] David Rincón and Sebastià Sallent. Scaling Properties of Network Traffic, to be included in *Encyclopedia of Internet Technologies and Applications*, Idea Group, 2007 (expected).

[RSdA05] David Rincón, Sebastiá Sallent, and Marilet de Andrade. Towards On-line Segmentation of Fractal Traffic. In *Proceedings of IEEE Infocom 2005 - student workshop*, pages 43–44, March 2005.

[Rus99] J.C. Russ. *The Image Processing Handbook (Third Edition)*. CRC Press, 1999.

[RV99a] Matthew Roughan and Darryl Veitch. Measuring Long-range Dependence under Changing Traffic Conditions. In *Proceedings of IEEE INFOCOM 99*, volume 3, pages 1513–1521, March 1999.

[RV99b] Matthew Roughan and Darryl Veitch. A Study of the Daily Variation in the Self-similarity of Real Data Traffic. In *Proceedings of the International Teletraffic Congress (ITC) 16*, pages 67–76, 1999.

[RVA98] Matthew Roughan, Darryl Veitch, and Patrice Abry. On-line Estimation of the Parameters of Long-range Dependence. In *Proceedings of IEEE Globecom 98*, volume 6, pages 3716–3721, November 1998.

[RW00] Rudolf Riedi and Walter Willinger. *Kihong Park and Walter Willinger, editors, Self-similar Traffic and Network Performance*, chapter Toward an Improved Understanding of Network Traffic Dynamics. John Wiley & Sons, 2000.

[RZ06] David Rincón and Piotr Zuraniewski. Wavelets-based Methods for Detection of Structural Changes in Highly Correlated Time Series. In *Proceedings of the EuroNGI Workshop on Traffic Engineering, Protection and Restoration for Next Generation Internet (to be published)*, May 2006.

[RZM06] David Rincón, Piotr Zuraniewski and Flamnio Minerva. *U. Krieger, editor, Achievements on IP Traffic Characterization, Measurements and Statistical Methods. EuroNGI Deliverable D.WP.JRA.5.1.3*, chapter 11. Algorithms for the Progressive Segmentation of LRD Traffic with Wavelet Transforms and Variance Change Detection, pages 115–149. EuroNGI Network of Excellence, November 2006.

[SA04] Ivan W. Selesnick and A. Farras Abdelnour. Symmetric Wavelet Tight Frames with Two Generators. *Journal of Applied and Computational Harmonic Analysis*, 17(2):211–225, April 2004.

[SBK05] I. W. Selesnick, R. G. Baraniuk, and N. Kingsbury. The Dual-tree Complex Wavelet Transform - a Coherent Framework for Multiscale Signal and Image Processing. *IEEE Signal Processing Magazine*, 22(6):123–151, November 2005.

[SCLA] Ivan Selesnick, Shihua Cai, Keyong Li, and A. Farras Abdelnour. Matlab Implementation of Wavelet Transforms. Polytechnic University & Metrotech Center, Brooklyn NY (USA). Avaliable at http://taco.poly.edu/WaveletSoftware/.

[Sel01] I. W. Selesnick. *A. Petrosian and F. G. Meyer, editors, Wavelets in Signal and Image Analysis: From Theory to Practice*, chapter The Double Density DWT. Kluwer, 2001.

[Sel04] Ivan Selesnick. The Double-density Dual-tree DWT. *IEEE Transactions on Signal Processing*, 52(5):1304–1314, May 2004.

[Sel06] Ivan Selesnick. A Higher-Density Discrete Wavelet Transform. *IEEE Transactions on Signal Processing*, 54(8):3039–3058, August 2006.

[SSR94] M. A. Sola and S. Sallent-Ribes. Progressive Orthogonal Tilings of the Time-frequency Plane. In A. K. Katsaggelos, editor, *Proceedings of SPIE, Vol. 2308, Visual Communications and Image Processing '94*, pages 1112–1123, September 1994.

[ST99] Z. Sahinoglu and S. Tekinay. On Multimedia Networks: Self-similar Traffic and Network Performance. *IEEE Communications Magazine*, 37(1):48–52, January 1999.

[ST01] Z. Sahinoglu and S. Tekinay. A Novel Adaptive Bandwidth Allocation: Wavelet-decomposed Signal Energy Approach. In *Proceedings of IEEE Global Telecommunications Conference - Globecom'01*, volume 4, pages 2253–2257, November 2001.

[ST02] Z. Sahinoglu and S. Tekinay. Efficient Parameter Selection for Use of Self-similarity in Real Time Resource Management. In *Proceedings of IEEE International Conference on Communications - ICC 2002*, volume 4, pages 2212–2216, 2002.

[Ste94] W.R. Stevens. *TCP/IP Illustrated (vol. 1).* Addison-Wesley, 1994.

[Sto] S. Stoev. Matlab Code for Generation of FGN. Boston University (USA). Avaliable at http://math.bu.edu/people/sstoev/.

[SW86] Henry Stark and John W. Woods. *Probability, Random Processes and Estimation Theory for Engineers.* Prentice-Hall International, 1986.

[Uhl02] Steve Uhlig. 3D-LD: a Graphical Wavelet-based Tool for Analyzing Non-stationary and Scaling Processes. Technical report, University of Namur, 2002.

[Uhl04] Steve Uhlig. Nonstationarity and High-order Scaling in TCP Flow Arrivals: a Methodological Analysis. *ACM SIGCOMM Computer Communications Review*, 34(2):9–24, April 2004.

[VA] Darryl Veitch and Patrice Abry. Darryl Veitch's Homepage - Matlab Code for the Logscale Diagram. University of Melbourne (Australia). Avaliable at http://www.cubinlab.ee.mu.oz.au/∼darryl/.

[VA99] Darryl Veitch and Patrice Abry. A Wavelet-based Joint Estimator of the Parameters of Long-range Dependence. *IEEE Transactions on Information Theory*, 45(3):878–897, April 1999.

[VA01] Darryl Veitch and Patrice Abry. A Statistical Test for the Time Constancy of Scaling Exponents. *IEEE Transactions on Signal Processing*, 49(10):2325–2334, October 2001.

[VAT03] D. Veitch, P. Abry, and M.S. Taqqu. On the Automatic Selection of the Onset of Scaling. *Fractals*, 11(4):377–390, 2003.

[VHA05] D. Veitch, N. Hohn, and P. Abry. Multifractality in TCP/IP Traces: the Case Against. *Computer Networks*, (48):293–313, June 2005.

[VTA00] D. Veitch, M.S. Taqqu, and P. Abry. Meaningful MRA Initialization for Discrete-time Series. *Signal Processing*, 80(9):1971–1983, September 2000.

[Whi98] Brandon Whitcher. *Assessing Nonstationary Time Series Using Wavelets.* PhD thesis, University of Washington, 1998.

[WTSW97] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. Self-similarity through High-variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, February 1997.

[YTJ01] Xiang Yu, Ian Li-Jin Thng, and Yuming Jiang. Measurement-based Effective Bandwidth Estimation for Long Range Dependent Traffic. In *Proceedings of the IEEE Region 10 International Conference on Electrical and Electronic Technology (TENCON)*, volume 1, pages 359–365, August 2001.

[ZR06] Piotr Zuraniewski and David Rincón. Wavelet Transforms and Change-point Detection Algorithms for Tracking Network Traffic Fractality. In *Proceedings of the* 2^{nd} *EuroNGI Conference on Next Generation Internet Design and Engineering (NGI 2006)*, pages 216–223, April 2006.