DATA COMPRESSION AND RECONSTRUCTION IN PROCESS ENGINEERING APPLICATIONS

by

Ceyda Önol

B.S., Chemical Engineering, Boğaziçi University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Chemical Engineering

Boğaziçi University

2012

ACKNOWLEDGEMENTS

First of all, I gratefully acknowledge my thesis supervisor Prof. Uğur Akman for all his kindness, patience and guidance throughout this study.

I would also like to express my gratitude to the members of the defense jury for their comments on my thesis.

I am also indebted to my father, Gökhan Önol and my mother, Saime Önol for their moral support and encouragement throughout my thesis work.

Finally, I would like to thank all my colleagues at Pfizer especially Güler Kızılcık, Belen Güngör and Bengü Gürler. Their friendship and understanding during this study is greatly acknowledged.

ABSTRACT

DATA COMPRESSION AND RECONSTRUCTION IN PROCESS ENGINEERING APPLICATIONS

Recent improvements in sensor technology have resulted in huge amount of measured process data along with the increasing need for compression prior to storage. Hence, efficient process data compression and reconstruction techniques gain importance in various tasks such as process monitoring, system identification, and fault detection to save storage space and facilitate data transmission between a data collecting node and a data processing node. Main purpose of this thesis work is to be able to achieve the highest degree of compression and de-noising while preserving the key features of the original data upon retrieval and decompression. With this aim, the employed are the most appropriate dimensionality reduction technique among Piecewise Aggregate Approximation (PAA), One Dimensional and Two Dimensional Discrete Cosine Transform (1D-DCT and 2D-DCT) and One Dimensional and Two Dimensional Discrete Wavelet Transform (1D-DWT and 2D-DWT) by adjusting the threshold parameter in filtering. The data sets used are PortSimHigh, PortSimLow, SELDI-TOF MS and TEP. These techniques are evaluated in terms of compression ratio, reconstruction error norm, % relative global error and % relative maximum error for different α-% thresholding levels. It is concluded that high compression levels cannot be generated with thresholding percentile values less than 90% in both DCT and DWT methods whereas the quality of reconstruction deteriorates at higher threshold levels in return for better compression. Furthermore, it is revealed that the efficacy of the compression methods strongly depends on the data characteristics. DCT is suitable for smooth data sets with random trends whereas DWT is preferred for the noisy data sets with high peak content. 2D-DCT and 2D-DWT are favored for the multivariable data sets with highly correlated columns.

ÖZET

PROSES MÜHENDİSLİĞİ UYGULAMALI VERİ SIKIŞTIRMA VE YENİDEN OLUŞTURMA

Sensor teknolojisindeki son gelişmeler sayesinde büyük miktarlarda proses verisi toplanabilmektedir. Fakat bu durum, veri arşivlemeyi kolaylaştırmak için yapılan veri sıkıştırma işlemine duyulan ihtiyacı arttırmıştır. Bunun sonucu olarak, verilerin daha az yer kaplaması ve veri toplayan ve işleyen düğümler arasındaki iletimi hızlandırmak için proses izleme, sistem tanımlama ve hata saptama gibi birçok alanda proses verisi sıkıştırma ve bu veriyi yeniden oluşturma teknikleri önem kazanmıştır. Bu tez çalışmasının ana amacı, orijinal veri setlerinin temel özelliklerini koruyarak yüksek derecelerde sıkıştırma oranları elde edebilmek ve bunun yanında gürültülü verilerden kurtulabilmektir. Bu amaçla, süzgeçleme işlemindeki eşik seviyesi ayarlanarak parçalı kümelemeyle yaklaşımlama, bir ve iki boyutlu ayrık kosinüs dönüşümü ve bir ve iki boyutlu ayrık dalgacık dönüşümü tekniklerinin verimlilikleri değerlendirilmiştir. Bu çalışmada, birbirinden farklı özellikleri olan PortSimHigh, PortSimLow, SELDI-TOF MS ve TEP veri setleri kullanılmıştır. Bahsi geçen sıkıştırma teknikleri, değişik eşik seviyeleri kullanılarak sıkıştırma oranı, yeniden oluşturma hata normu, % göreli global hata ve % göreli maksimum hata değerleri baz alınarak karşılaştırılmıştır. Ayrık kosinüs ve dalgacık dönüşümü metotları ile %90'dan küçük eşik seviyeleri kullanıldığında yüksek sıkıştırma oranlarının elde edilemediği fakat yüksek eşik seviyelerinde daha iyi sıkıştırma oranları karşılığında veriyi yeniden oluşturma kalitesinin kötüleştiği sonucuna varılmıştır. Ayrıca, sıkıştırma tekniklerinin verimliliğinin büyük oranla kullanılan veri setlerinin özelliklerine bağlı olduğu anlaşılmıştır. Ayrık kosinüs dönüşümü metodu rastgele eğilimleri olan düzgün veri setleri için tercih edilirken, ayrık dalgacık dönüşümü metodu çok fazla tepe noktası olan gürültülü veri setleri için daha uygundur. Üstelik, kolonları arasında ilişiği olan çok değişkenli veri setleri için iki boyutlu ayrık kosinüs ve dalgacık dönüşümü metotlarını kullanmak daha kazanımlıdır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a	Frequency scale (dilation)
A	Data matrix with dimensions M by N used in 2D-DCT
A_n	Approximations at level n
b	Position in time (translation)
B	2D-DCT transform coefficients matrix with dimensions M by N
c	Vector having n dimensions used in PAA
C	Coefs vector in which wavelet coefficients are assembled
cA_n	Approximation coefficients at level n
cD_n	Detail coefficients at level n in 1D-DWT
cD_n	Diagonal detail coefficients at level n in 2D-DWT
cH_n	Horizontal detail coefficients at level n in 2D-DWT
cV_n	Vertical detail coefficients at level n in 2D-DWT
d	A variable in the 2D-DCT equation
D_n	Details at level n in 1D-DWT
D_n	Diagonal details at level n in 2D-DWT
e	1D-DCT transform coefficients vector with length N
E	1D-DCT transform coefficients matrix
f_i	i^{th} element of the original signal
f'_i	i^{th} element of the reconstructed signal
f(t)	A time-varying signal used in 1D-DWT
f(l)	l^{th} element of the signal
f'(a,b)	Integral wavelet transform of f(t)
f(x,y)	Data set used in 2D-DWT
$g_{j,k}$	Wavelet coefficients in 1D-DWT
H_n	Horizontal details at level n in 2D-DWT
H(p)	Shannon entropy
I	Orthogonal DCT square matrix used in 2D-DCT
K	Observations
L	Window length
$\mathbf{m}_T$	Real-valued nonzero time series of length T

m/z	Mass to charge ratios
N	Hankel matrix
o	One dimensional data with length N used in 1D-DCT
O	Two dimensional data matrix used in 1D-DCT
P	Probability mass function
r	A variable in the 1D-DCT equation
S	Original signal used in 1D-DWT
U	Orthogonal DCT square matrix used in 1D-DCT
V_n	Vertical details at level n in 2D-DWT
w	Segment size used in PAA
$W_\phi(j_0,m,n)$	Approximation coefficients at level j_0 in 2D-DWT
$W_\psi(j,m,n)$	Detail coefficients at each level j in 2D-DWT
x	Composed vector in PAA having w dimensions
X	Original signal used in 2D-DWT
y	Composed vector in PAA having n dimensions

α-%	Thresholding levels used in filtering
$\phi(x,y)$	Scaling function used in 2D-DWT
$\psi(t)$	Wavelet function
$\psi_D(x,y)$	Diagonal wavelet function used in 2D-DWT
$\psi_H(x,y)$	Horizontal wavelet function used in 2D-DWT
$\psi_V(x,y)$	Vertical wavelet function used in 2D-DWT

LIST OF ACRONYMS/ABBREVIATIONS

1D-DCT	One Dimensional Discrete Cosine Transform
1D-DWT	One Dimensional Discrete Wavelet Transform
2D-DCT	Two Dimensional Discrete Cosine Transform
2D-DWT	Two Dimensional Discrete Wavelet Transform
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
HPF	High-pass filter
IDCT	Inverse Discrete Cosine Transform
IDWT	Inverse Discrete Wavelet Transform
KLT	Karhunen-Loeve Transform
LPF	Low-pass filter
MILP	Mixed-integer linear programming
NLP	Non-linear programming
NMPC	Nonlinear model predictive control
PAA	Piecewise Aggregate Approximation
PCA	Principal Components Analysis
RLE	Run-Length Encoding
SELDI-TOF MS	Surface-enhanced laser desorption/ionization time-of-flight mass spectrometry
SSA	Singular Spectrum Analysis
TEP	Tennessee-Eastman Plant
TM	Trajectory Matrix

1. INTRODUCTION

Due to advances in information systems technology, plant historians can archieve vast amount process data. In many applications, such as process monitoring, fault detection, fault identification, fault classification, image processing and signal processing, dealing with multi-dimensional data such as image data, stock market data and chemical process data, data compression is often required to save storage space and speed up retrieval time. The objective of this study is to be able to achieve the highest degree of compression while retaining the prominent features of the original data upon retrieval and decompression by selecting the most appropriate transform method and the optimum thresholding limit used in filtering.

There have been only a few studies related to process data storage and compression although there are many publications in the areas of image and signal compression. Hale and Sellars (1981) published one of the earliest papers on piecewise linear compression methods (boxcar and backward slope) that have been used on process monitoring and control systems in Du Pont factories. Afterwards, swinging door (Bristol, 1990) and piecewise linear online trending (PLOT) algorithms (Mah *et al.*, 1995) were developed by modifying the boxcar and backward slope algorithms. Piecewise linear compression techniques are the simplest dimensionality reduction methods and, thus, they are generally preferred in the chemical process industries.

Bakshi and Stephanopoulos (1996) and Misra *et al.* (2000) modified existing wavelet-packet decomposition algorithms for on-line data compression concluding that better compression can be achieved compared to batch methods due to efficient indexing and bookkeeping schemes in on-line compression methods. Furthermore, Vedam *et al.* (1998) developed B-Spline based data compression and de-noising algorithm, achieving high compression with small reconstruction error. On the other hand, Thornhill *et al.* (2004) showed that data compression can sometimes give misleading information about statistical properties of the data (e.g. mean and variance), concluding that compressed data can be useless for some tasks such as statistical monitoring.

Watson *et al.* (1998) compared the effectiveness of the piecewise linear compression techniques (boxcar and backward slope), transform compression methods (discrete Fourier, discrete cosine and discrete wavelet transform) and vector quantization by using plant data and concluded that the wavelet transform performs best in reconstruction since wavelets are well suited for describing localized changes. Only one dimensional transform techniques were studied in this paper, whereas two dimensional transform techniques could also be investigated for analyzing multiple time series with column-correlations in order to improve the compression/reconstruction performance. Hence, the central aim of this thesis is to extend the work of Watson *et al.* (1998) by investigating two dimensional discrete cosine and two dimensional discrete wavelet transform algorithms along with data sets with different column-correlation characteristics. Therefore this thesis work can be perceived as an overhauling of the work of Watson *et al.* (1998) using techniques that have become popular since then in mainly image processing.

The main purpose of this thesis is to compare different data compression and lossy/lossless reconstruction methods Piecewise Aggregate Approximation (PAA), One Dimensional and Two Dimensional Discrete Cosine Transform (1D-DCT and 2D-DCT) and One Dimensional and Two Dimensional Discrete Wavelet Transform (1D-DWT and 2D-DWT), including the thresholding method as a lossy compression step and ZIP as the lossless encoding algorithm by measuring compression ratio, reconstruction error norm, % relative global error and % relative maximum error for different α-% thresholding levels using the data sets PortSimHigh, PortSimLow, SELDI-TOF MS and TEP.

In Chapter 2, the fundamentals of data compression and lossy/lossless reconstruction including the compression methods (direct and transform methods), the algorithms used in lossless compression (Lempel-Ziv, Run-Length Encoding and Huffman Coding) and the filtering methods (thresholding and zero padding methods) are briefly reviewed. Definitions of compression ratio, % reduction and Shannon entropy which are used to measure the degree of compression are given. Illustrations of the filtering methods applied to transform coefficients are also presented in Section 2.4.

In Chapter 3, the characteristics of the data sets used in thesis (PortSimHigh, PortSimLow, SELDI-TOF MS and TEP) are given in detail to facilitate the interpretation of the compression/reconstruction results presented in later chapters.

Chapter 4 covers the two irreversible techniques; Piecewise Aggregate Approximation (PAA) which is the simplest data compression technique and data quantization. PAA technique is studied by using single representative columns of the PortSimHigh, SELDI-TOF MS and TEP data sets for different segment sizes. The effect of the frame size on compression is measured in terms of compression ratio, error norm and the Shannon entropy. Furthermore, quantization technique is studied to be able to improve the compression performance of PAA. Optimum frame size used in PAA and optimum number of digits kept after the decimal in quantization are also investigated.

In Chapter 5, Discrete Cosine Transform (DCT) is explained in detail. The formulas used for calculating 1D-DCT and 2D-DCT coefficients are given. Detailed 1D-DCT and 2D-DCT analyses are presented in Section 5.3 for the overall PortSimHigh, PortSimLow, SELDI-TOF MS and TEP data sets. Furthermore, 1D-DCT and 2D-DCT techniques are compared by using the overall data sets for different thresholding percentile values in the [15%-99.8%] range. The effect of the percentile values used in thresholding step on compression is measured in terms of compression ratio, mean error norm, % relative global error and % relative maximum error.

Chapter 6 is devoted to Discrete Wavelet Transform (DWT) which is favored in various applications such as on-line data compression and pattern-matching. The formulas used for calculating multi-level 1D and 2D wavelet transform coefficients (approximation and detail coefficients) are given. Detailed 1D-DWT and 2D-DWT analyses are presented in Section 6.3 using wavelet types `db1` for the PortSimHigh and PortSimLow data sets, `db4` for the SELDI-TOF MS data set, and `sym4` for the TEP data set at different wavelet-decomposition levels. In addition, the illustrations of the three-level wavelet decompositions with 1D-DWT and 2D-DWT using the wavelet type `db1` for the PortSimHigh and PortSimLow data sets are given in Sections 6.1.1 and 6.2.1, respectively. The decomposition level in DWT is selected so as to generate the same compression level produced by DCT at the percentile value of 99.5%, and thus, reconstruction error norms

produced in DCT and DWT can be compared at the same compression level. Furthermore, 1D-DWT and 2D-DWT techniques are compared by using the overall data sets for different thresholding percentile values in the [15%-99.8%] range. The effect of the percentile values used in thresholding step on compression is measured in terms of compression ratio, mean error norm, % relative global error and % relative maximum error.

Chapter 7 deals with the construction of a trajectory matrix for its use in two dimensional compression of one dimensional data. The transformation of one dimensional data into two dimensional data for compression and reconstruction is studied by composing the trajectory matrix and then applying the 2D-DCT method using single representative columns of the PortSimHigh and TEP data sets.

Chapter 8 is the overall summary of the previous chapters stating the results of this thesis work. A few comments are given related to future work in the light of the conclusions obtained in this study.

All of the computations are performed in MATLAB. The MATLAB codes used in this study are given in Appendix A.

2. FUNDAMENTALS OF DATA COMPRESSION AND RECONSTRUCTION

Data compression is required to save storage space and to speed up data transmission between a data collecting node and a data processing node. Data compression can be achieved by the elimination of redundant data. Compression algorithms transform a data set by removing repetitions and by removing or filtering noise in the data. Most algorithms consist of the combination of transformation, quantization and coding steps (Watson *et al.*, 1998) as illustrated in Figure 2.1.

2.1. Data Compression Methods

Compression methods can be classified into two groups:

(i) Direct Methods
- Piecewise Linear Compression
  - (i) Boxcar, Backward Slope and Swinging Door algorithms

(ii) Transform Methods
- Karhunen-Loeve Transform (KLT)
- Discrete Fourier Transform (DFT)
- Discrete Cosine Transform (DCT)
- Discrete Wavelet Transform (DWT)

The transform methods generally give better results than direct ones in which linear interpolation is used for reconstruction. Piecewise linear compression techniques perform well for steady-state operations and signals that have little noise (Bakshi and Stephanopoulos, 1996). However they do not take into account changes in other variables and this may affect correlation between signals leading to the loss of valuable correlation information (Imtiaz and Choudhury, 2007).

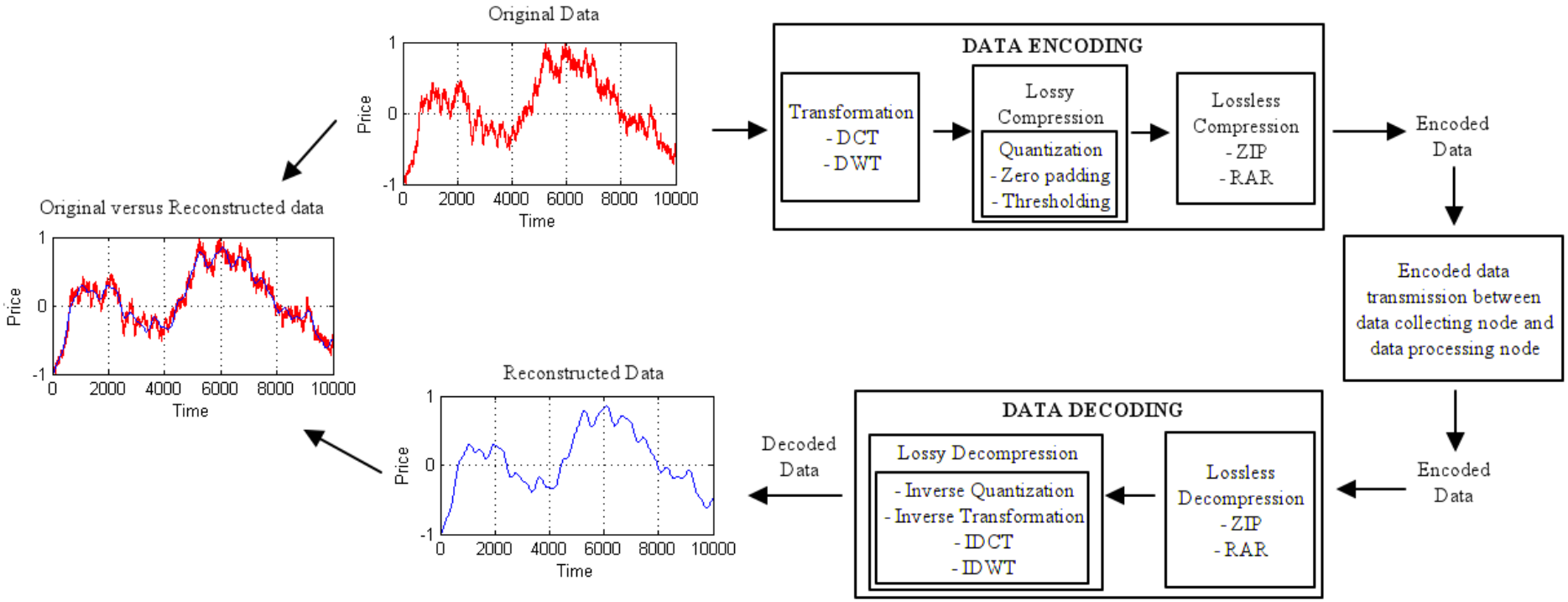

Figure 2.1. Components of a Data Encoding/Decoding Algorithms.

.

Karhunen-Loeve Transform (KLT), or Principal Components Analysis (PCA), mostly used in multivariate data analysis, gives the best possible compression ratio, however its application is difficult since transformation kernel is not separable and data dependent. On the other hand, Discrete Fourier Transform (DFT) is fast and easy method using both sine and cosine functions, its transformation kernel is linear, separable and symmetric, however compression ratio is not adequate. Discrete Cosine Transform (DCT) is asymptotically equivalent to KLT and it is the easiest method using only cosine waves (Khayam, 2003). It performs mapping from time to frequency domain. Discrete Wavelet Transform (DWT) is generally preferred when dealing with smaller details consisting of high frequencies as wavelets are suitable for sudden changes. Wavelets can also describe discontinuities in signal analysis as they deal with both time and frequency domains simultaneously whereas DCT operates only in frequency domain (Stark, 2005).

Piecewise Aggregate Approximation (PAA) is a fast and easy dimensionality reduction technique in which data are divided into equal sized frames within which the data are averaged to a constant value without transformation, however some important patterns can easily be discarded (Lkhagva *et al.*, 2006).

In this thesis work, the DCT and DWT methods will be used on the data sets that are described in Chapter 3. The PAA and quantization techniques will also be applied for further dimensionality reduction.

2.2. Data Compression Algorithms

There are two types of compression (encoding):

(i) Lossless (Reversible) Compression
- Restored data file is identical to the original data.
- It is used in ZIP, RAR, GIF and TIFF applications.

(ii) Lossy (Irreversible) Compression
- Restored data file is an approximation of the original data.

- It is used for visual and audio data such as JPEG and MPEG applications and also on Internet.

Modeling used in lossless data compression can be classified into two groups; statistical and dictionary-based. Statistical modeling encodes a symbol based on its occurrence probability, however in dictionary-based modeling, strings of symbols are encoded by a single code (Nelson and Gailly, 1996).

In lossless data compression, efficient redundancy removal is required. Following algorithms are generally used in reversible compression (Nelson and Gailly, 1996);

- Lempel-Ziv (LZ) Coding, in which a dictionary of previously seen strings of symbols is composed and these groups are replaced by a single code.
- Run-Length Encoding (RLE), in which repetitions are stored as a single data.
- Huffman Coding is the most commonly used algorithm, in which codes are assigned to symbols and a symbol with high occurrence probability generates a shorter code.

WinZIP is a file archiving application generally compressing files to roughly half the size of the original file (Sayood, 2003). RAR software is another application achieving somewhat better compression with high speed, for instance WinRAR can compress files up to 8589 billion GB (Salomon, 2007). Compression ratio is higher for data having repeated patterns.

For lossless compression algorithms, compression effect is measured by following definitions; compression ratio, % reduction and entropy;

$$\text{Compression ratio} = \frac{\text{Size before compression}}{\text{Size after compression}} \tag{2.1}$$

$$\%\ \text{reduction} = \left(1 - \frac{\text{Size after compression}}{\text{Size before compression}}\right) \times 100 \tag{2.2}$$

In information theory, Shannon entropy is generally used to measure encoded information in a message. Large entropy means that the message has high information content and data compression causes entropy reduction. Shannon entropy is defined in bits as following (Nelson and Gailly, 1996);

$$H(p) = -\sum_{j=1}^{n} p_j log_2 p_j \tag{2.3}$$

where, H denotes the Shannon entropy of a discrete random variable $X = (x_1, x_2, \dots, x_n)$ having the probability mass function $P = (p_1, p_2, \dots, p_n)$ satisfying the following constraints;

$$\sum_{i=1}^{n} p_i = 1\,,\ p_i \geq 0$$

In lossy compression, a slight distortion in images and sounds (almost unnoticeable to human eye and ear) can be accepted for better compression. In irreversible compression, data are firstly transformed into a transform domain by using one of the discrete transform techniques mentioned earlier. Then, the transform coefficients are rounded off according to a defined quality level in the quantization step, where the loss of signal occurs. Finally, quantized coefficients are compressed with lossless encodings such as ZIP and RAR (Nelson and Gailly, 1996).

2.3. Data Quantization and Transform Coefficients Filtering

Quantization step reduces number of bits (or, decreases number of digits) representing the data. It is expected that error in reconstruction will increase as the number of bits representing data decreases (Khayam, 2003). On the other hand, filtering of the transform coefficients discards most of the transform coefficients having relatively small amplitudes without causing significant reconstruction error (i.e., with acceptably insignificant loss of information content of the original data).

There are mainly two types of filtering applied to transform coefficients:

- Truncating/shrinking the number of transform coefficients by cutting from the end of the transform-coefficients vector

- Thresholding the insignificant transform coefficients throughout the transform-coefficients vector

In the first method, only the first few coefficients (the largest ones) are kept and rest of them are set to zero. In the reconstruction step, zero padding method inserts zeros to the end of the cut transform coefficients and then inverse transform (decoding) is applied. It is expected that there will be nearly no loss of information when zero padding is used for smooth signals with low-frequency content that do not contain significant noise since their transform coefficients die out exponentially towards almost zero. This method is not applicable for signals with high-frequency content since their transform coefficients do not die out, they persist to be significant throughout the transform-coefficients vector.

In thresholding, the transform coefficients the magnitudes of which are between user-defined threshold limits are set to zero. Information loss should be minimized by adjusting these upper and lower thresholds. Taking of the absolute values of transform coefficients, sorting them out, and setting the threshold limits by considering a percentile of the frequency distribution of the transform coefficients magnitudes may be necessary. For signals with high-frequency content, the thresholding method is the preferred one.

In the later chapters of this thesis work, the data compression and lossy/lossless reconstruction will be investigated by using compression techniques PAA, DCT and DWT, including the thresholding method as a lossy-compression step and ZIP as the encoding algorithm. These techniques will be compared by measuring compression ratio, % reduction, entropy and reconstruction error norm for different α-% thresholding levels.

2.4. Illustration of Transform Coefficients Filtering

PortSimHigh data set (stock market prices and their return values) described in Chapter 3 will be used here for the illustration of the zero padding and thresholding methods using the DCT technique. Return data series are generally used instead of price data series in stock market calculations and portfolio optimization.

Figure 2.2a and Figure 2.2b show the low frequency content (price) and high frequency content (return) data series (50th column of PortSimHigh data set) and their complete DCT coefficients are shown in Figure 2.2c and Figure 2.2d. Figure 2.2e and Figure 2.2f are given to illustrate the first 1000 DCT coefficients. Figure 2.2g and Figure 2.2h are given to show the zoomed first 100 DCT coefficients. As it is seen from these figures, the last 9000 transform coefficients of the price data are basically zero, thus they can easily be eliminated in order to achieve high compression levels without causing significant reconstruction error. However, unlike of the price transform coefficients, the transform coefficients of the return data persist to exist without any decay, therefore they cannot be eliminated as directly as those of the price data.

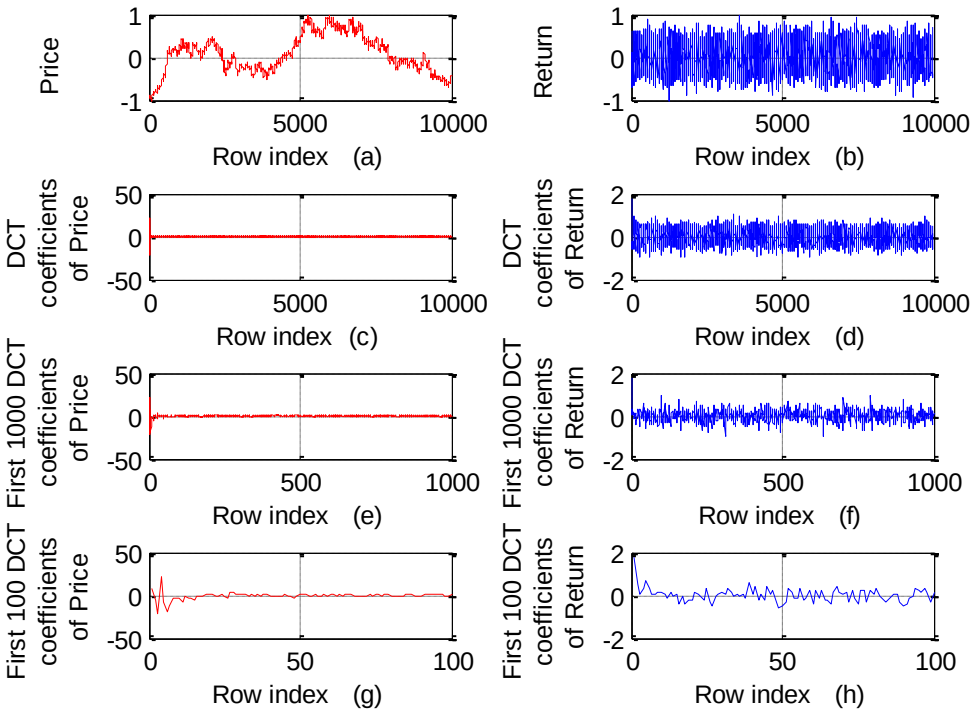

Figure 2.2. Stock Prices, Their Return Values, Full and Zoomed DCT Coefficients.

In the zero padding method, the first 100 transform coefficients are kept and the rest (9900 coefficients) are truncated (thus storing only the first 100 coefficients that are significant as shown in Figure 2.3c and Figure 2.3d) for both the price and return series. Actually, transform coefficients of the return data cannot be eliminated directly as they do not die out exponentially towards zero. However, for illustration purposes, transform coefficients of the return series are also truncated. In the reconstruction step, the stored

short DCT coefficient vector is padded with 9900 zeros to complete its length to 10000 (original data length) as illustrated in Figure 2.3a and Figure 2.3b.

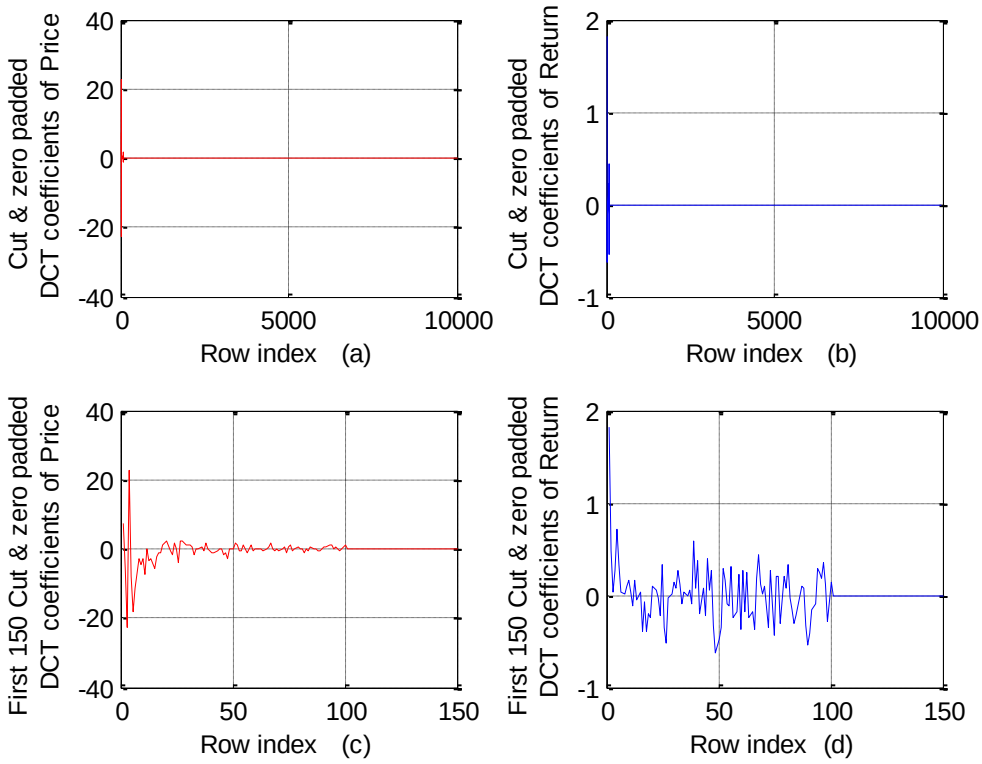

Figure 2.3. Cut and Zero Padded Full and Zoomed DCT Coefficients.

In thresholding method, upper and lower threshold limits are specified by calculating the percentile of the sorted transform coefficients after setting α-% as 99.8 and 70 for the price and return series as -1.9798 and 1.9798 for the price and -0.2794 and 0.2794 for the return, as illustrated in Figure 2.4a and Figure 2.4b. Transform coefficients between these limits are set to zero. Thresholded transform coefficients are shown in Figure 2.4c and Figure 2.4d. Figure 2.4e and Figure 2.4f illustrate the zoomed first 100 thresholded transform coefficients.

Plots of the original and reconstructed signals for price and return series after applying the zero padding method to the discarded 9900 transform coefficients are given in Figure 2.5a and Figure 2.5b. As it can be seen from Figure 2.5b, return data cannot be reconstructed thoroughly unlike of the price data as zero padding method is not suitable for data series having high-frequency content. Thus, reconstruction errors of the return data are much higher than those of the price data as illustrated in Figure 2.5c and Figure 2.5d which

show the reconstruction errors (= original - reconstructed). The zoomed reconstruction errors of the first 1000 prices and returns are also given in Figure 2.5e and Figure 2.5f. In addition, if fewer transform coefficients were truncated, there is no doubt that reconstruction would be better with less distortion.

Plots of the original and reconstructed signals for price and return data series after applying the thresholding method are given in Figure 2.6a and 2.6b. Return data set is reconstructed much better as compared to Figure 2.5b. The reconstruction errors of the return data set are slightly larger than those of the price data set as illustrated in Figure 2.6c and Figure 2.6d although smaller α-% is taken when thresholding the return data set. Fewer transform coefficients could be eliminated by using smaller α-% leading to a better reconstruction.

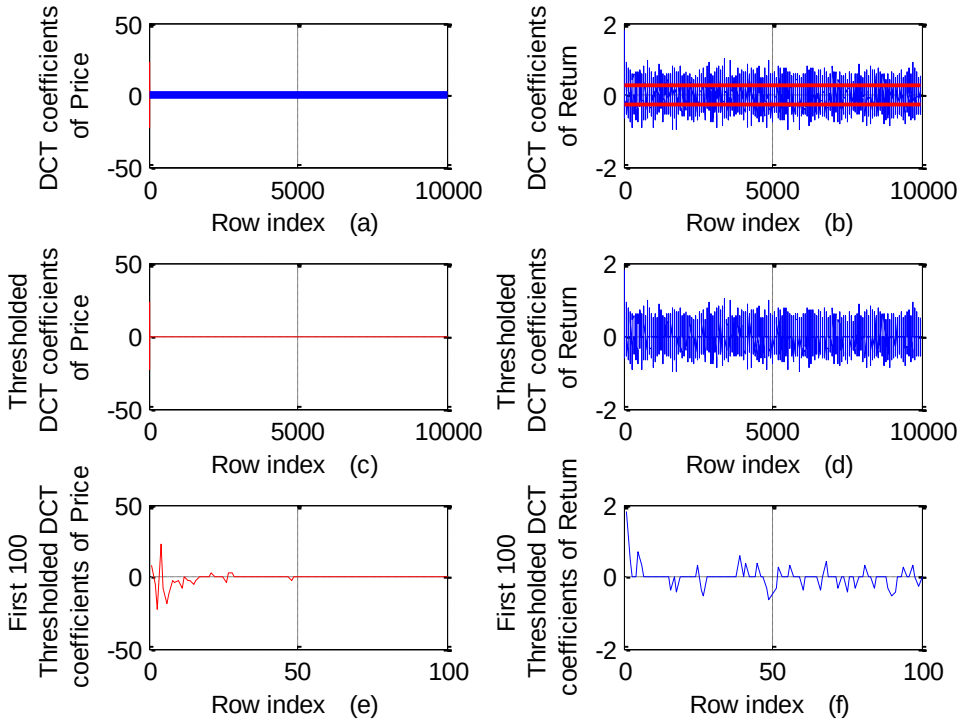

Figure 2.4. DCT Coefficients with Threshold Limits, Full and Zoomed Thresholded DCT Coefficients.

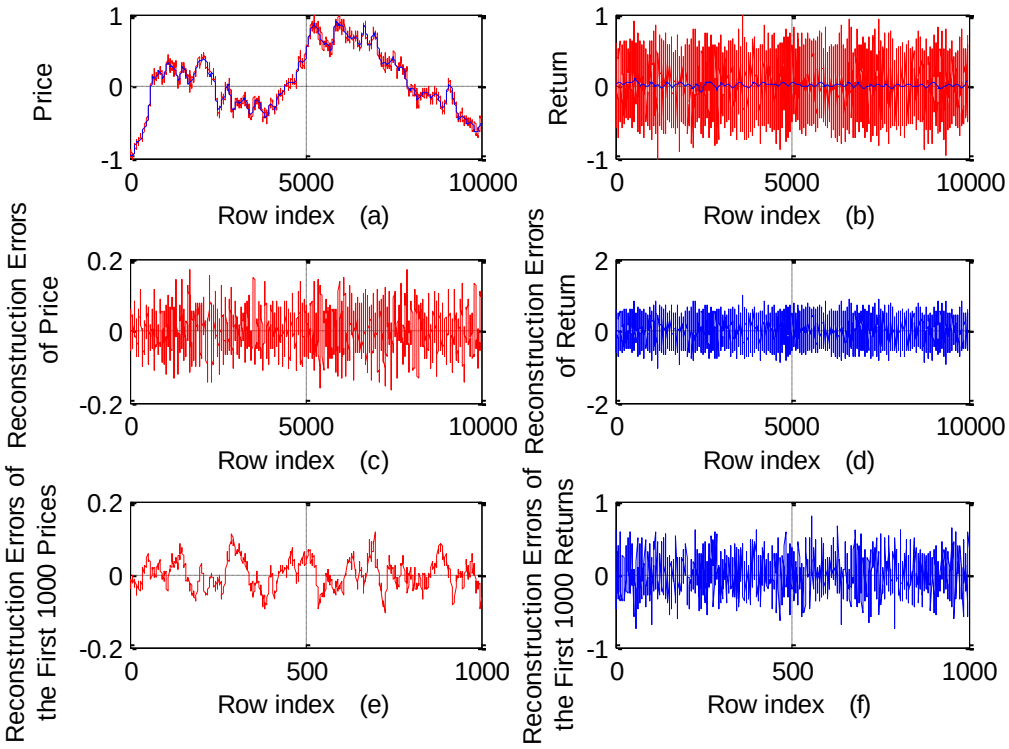

Figure 2.5. Original versus Reconstructed Signals and Reconstruction Errors after Applying the Zero Padding Method.

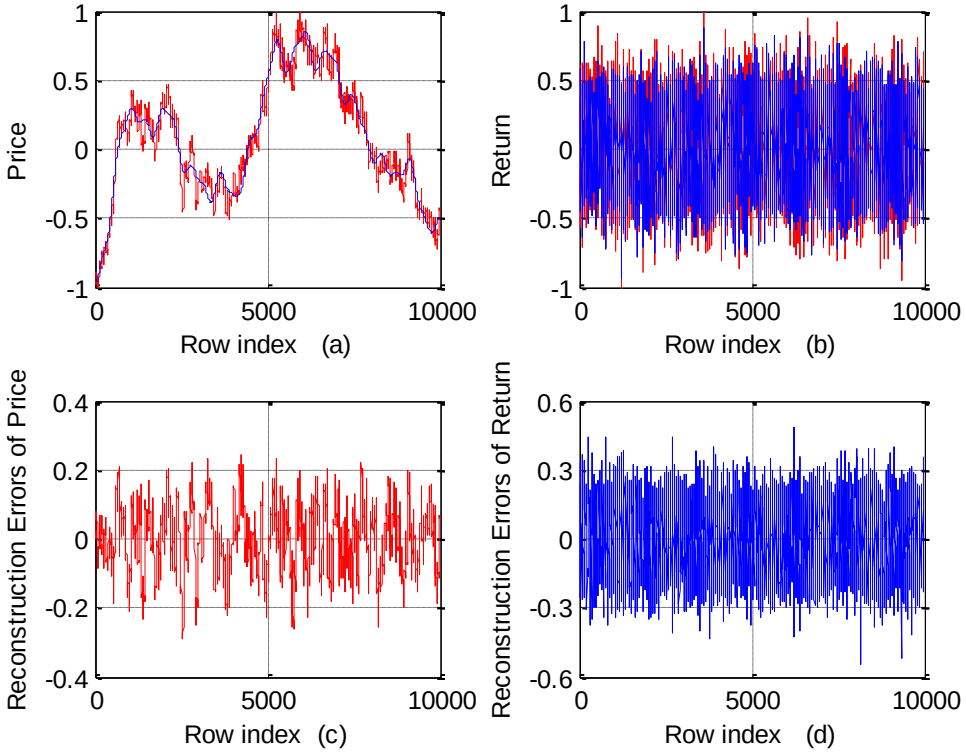

Figure 2.6. Original versus Reconstructed Signals and Reconstruction Errors after Applying the Thresholding Method.

Error in the reconstruction step increases as the number of discarded transform coefficients increases. It is important to adjust threshold limits and the number of cut transform coefficients in lossy quantization step as the difference between reconstructed and original signal should be minimized while maximizing compression.

3. DATA SETS USED AND THEIR CHARACTERISTICS

Vast variety of data sets are available for various applications. Each data set has specific features. It is usually said “Let the data speak for itself!” to emphasize the importance of the original untreated data itself. Statistical algorithms tailored to specific applications such as process monitoring, fault detection, image and signal processing dealing with multi-dimensional data such as image data, audio data, earthquake data, stock market data, bioinformatics data and chemical process data are gaining importance. In this chapter, the data sets used in compression studies are characterized in terms of smoothness, noise content and frequency distribution of their correlation coefficients. Thus, this chapter is expected to provide valuable insight on properties of the data sets used and facilitate the interpretation of the compression/reconstruction results presented in later chapters.

3.1. Synthetic Stock Market Data Sets

The synthetic stock market price data sets were generated using a MATLAB code that was developed based on hyperspherical decomposition algorithm. This algorithm and the code are parts of an unpublished work of the thesis advisor Prof. Uğur Akman enabling the generation of set of stock prices for arbitrary number of equities of arbitrary length with adjustable overall market correlation structure.

Each synthetic stock market data set is an ASCII text file of about 52.5 MB size consisting of 10000 rows and 500 columns. As far as Figure 3.1 and Figure 3.2 are concerned, it can easily be seen that data are highly correlated in case of PortSimHigh leading to higher correlations among the columns of PortSimHigh data set, whereas there is less correlation between those of PortSimLow data set.

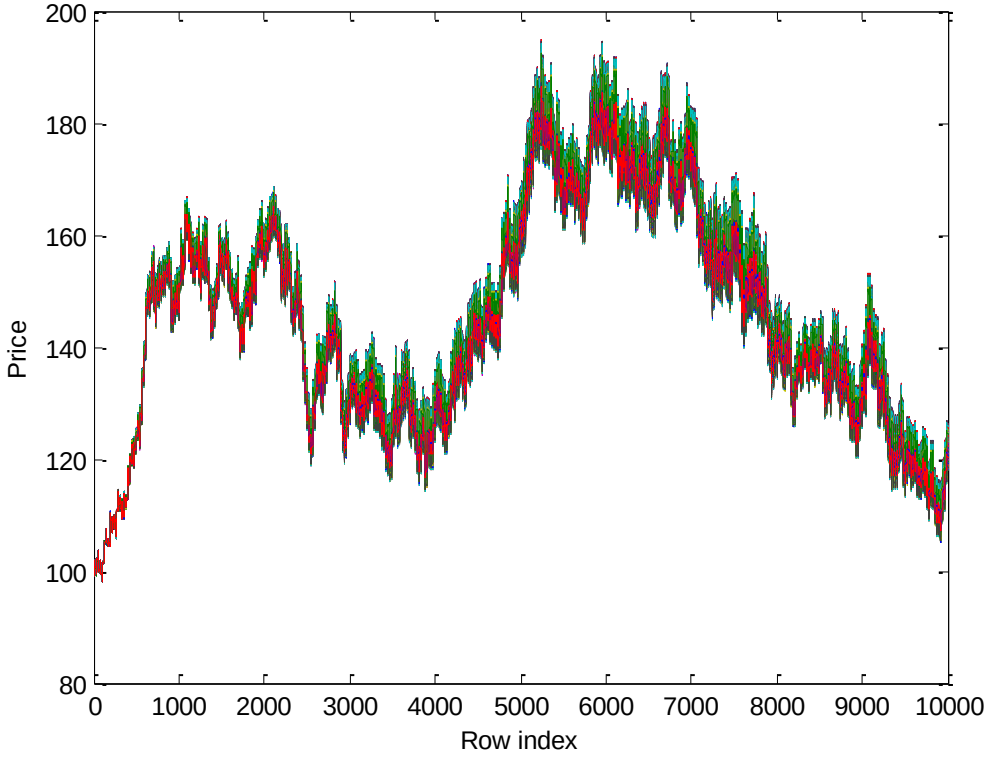

Figure 3.1. PortSimHigh Data Set Consisting of 10000 Rows and 500 Columns Representing Highly Correlated 500 Stock Prices.

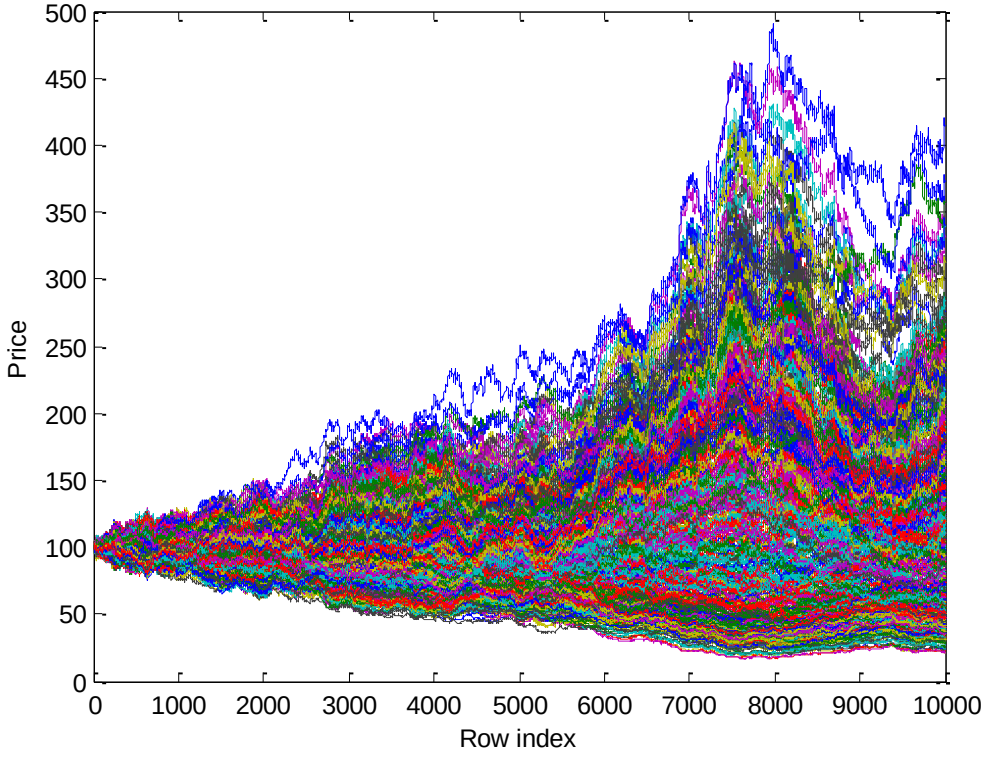

Figure 3.2. PortSimLow Data Set Consisting of 10000 Rows and 500 Columns Representing Less Correlated 500 Stock Prices.

3.2. Ovarian Cancer Mass Spectrometry (MS) Data Set

The surface-enhanced laser desorption/ionization time-of-flight mass spectrometry (SELDI-TOF MS) data are used to detect a disease from the circulating proteome such as plasma (Liu, 2012). MS produces high dimensional data consisting of molecule intensities for certain mass to charge (m/z) ratios. Each SELDI-TOF MS data set for a cancer sample provided by the home page of the National Cancer Institute[1] is a text file of about 5.5 MB size and six ovarian cancer samples (columns of data) with 337988 features (rows of data) are used in thesis.

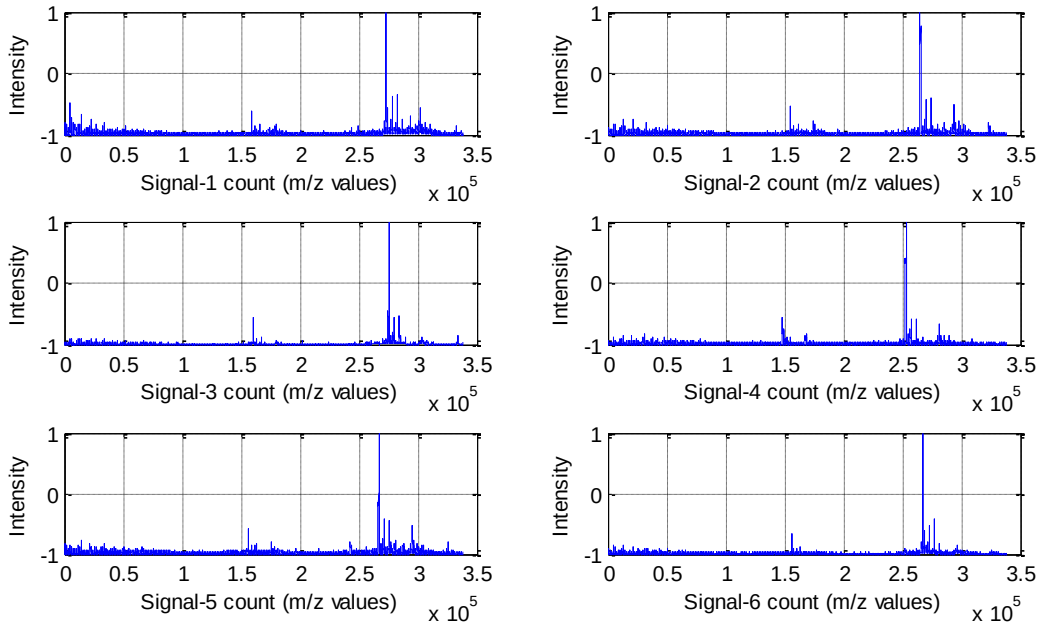

Figure 3.3. Scaled Intensities of the SELDI-TOF MS Data Set of Size 337988×6 for Ovarian Cancer Samples.

3.3. Tennessee-Eastman Plant (TEP) Data Set

The Tennessee-Eastman process (TEP) has been widely used as a benchmark simulation developed by Downs and Vogel (1993) for a real plant of the Eastman Chemical Company operating at Tennessee. The TEP deals with a multivariable, non-linear and unstable open-loop chemical plant in which product rate and composition should be maintained at desired levels while other variables are kept within shutdown limits

[1] http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp

(Zerkaoui *et al.*, 2010). TEP benchmark also includes various sensor measurement errors and possibilities for generating operational faults. McAvoy and Ye (1994) proposed a multi-loop (decentralized) PID control system for the TEP and then a nonlinear model predictive control (NMPC) strategy is followed by Ricker and Lee (1995). Besides the control issue, an optimization problem should be solved to minimize the operating cost of the plant by using techniques such as mixed-integer linear programming (MILP) or non-linear programming (NLP) (McAvoy, 1999).

The TEP, illustrated in Figure 3.4, produces two liquid products (G and H) and one by-product (F) from four gaseous reactants (A, C, D and E) and one inert component (B) by using the following five unit operations; a two-phase exothermic reactor, a condenser, a flash separator, a recycle compressor and a product stripper. The reactions are approximately first-order with respect to the reactant concentrations, irreversible and exothermic of the following form;

$$A(g) + C(g) + D(g) \rightarrow G(liq)$$
$$A(g) + C(g) + E(g) \rightarrow H(liq)$$
$$A(g) + E(g) \rightarrow F(liq)$$
$$3D(g) \rightarrow 2F(liq)$$

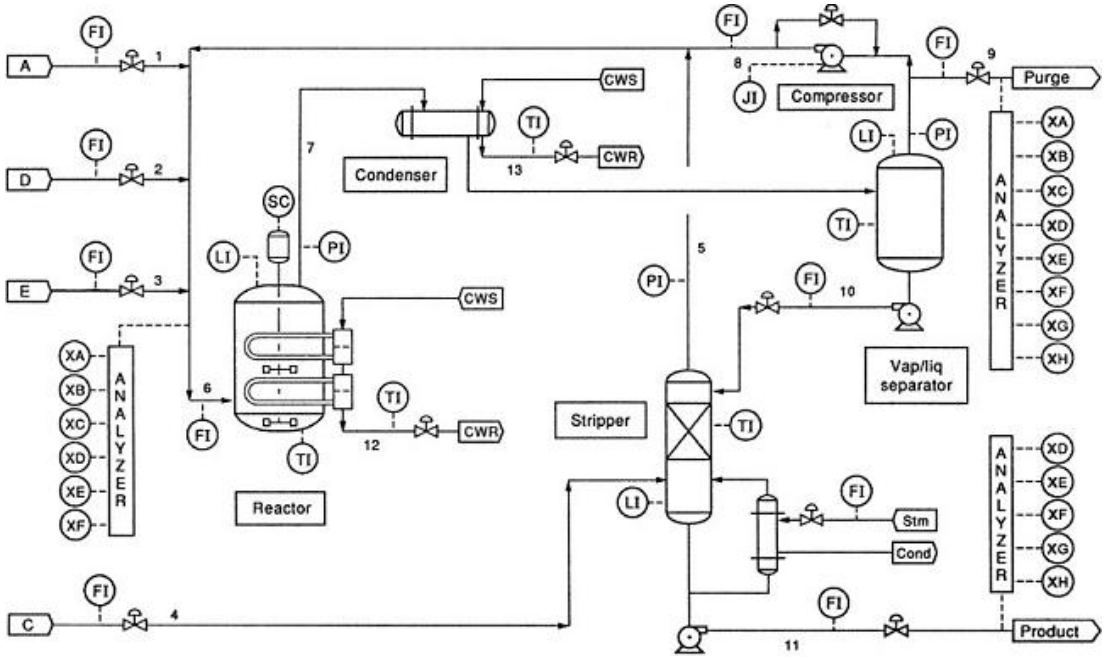

Figure 3.4. TEP Process (Downs and Vogel, 1993).

The gaseous reactants are fed to the reactor where the liquid products are formed in the presence of a non-volatile catalyst dissolved in the liquid phase. The product stream exiting the reactor passes through a condenser and a vapor-liquid (V/L) separator respectively. B and F are purged as vapor from the V/L separator. Non-condensed components recycle back to the reactor through a compressor, whereas condensed ones are pumped through a product stripper where G and H exit the base (Downs and Vogel, 1993).

There are six modes of process operation at three different G/H mass ratios where the ratio of 50/50 (G/H) with a production rate of 7038 kg/h for each product is the base mode (Ricardez-Sandoval *et al.*, 2009). The closed-loop control system requires the G/H ratio feedback to adjust the D/E ratio (Lu *et al.*, 2004).

Table 3.1. Operation Modes of the TEP (Zhao *et al.*, 2004).

mode	desired G/H mass ratio	desired production (kg/h)
1	50/50	14076
2	10/90	14077
3	90/10	11111
4	50/50	maximum
5	10/90	maximum
6	90/10	maximum

The process has 12 manipulated and 41 measured variables consisting of 22 continuous and 19 composition measurements (Ge and Song, 2007). It is assumed that all of the process measurements include Gaussian noise.

Table 3.2. Continuously Measured Process Variables (Ge and Song, 2007).

1	A feed	12	Product separator level
2	D feed	13	Product separator pressure
3	E feed	14	Product separator underflow
4	Total feed	15	Stripper level
5	Recycle flow	16	Stripper pressure
6	Reactor feed rate	17	Stripper underflow
7	Reactor pressure	18	Stripper temperature
8	Reactor level	19	Stripper steam flow
9	Reactor temperature	20	Compressor work
10	Purge rate	21	Reactor cooling water outlet temperature
11	Product separator temperature	22	Separator cooling water outlet temperature

Table 3.3. Manipulated Process Variables (Ge and Song, 2007).

1	D feed flow
2	E feed flow
3	A feed flow
4	Total feed flow
5	Compressor recycle valve
6	Purge valve
7	Separator pot liquid flow
8	Stripper liquid product flow
9	Stripper steam valve
10	Reactor cooling water flow
11	Condenser cooling water flow
12	Agitator speed

There are eight different disturbance scenarios given in Table 3.4 which can be simulated in order to test disturbance rejection and robustness of PI controllers. The process should recover quickly and smoothly from these disturbances (Downs and Vogel, 1993).

Table 3.4. Disturbance Scenarios for the TEP (Conradie and Aldrich, 2005).

Disturbance description	Type
A, B, C feed composition	Random variation
D feed temperature	Random variation
C feed temperature	Random variation
Reactor cooling water inlet temperature	Random variation
Condenser cooling water inlet temperature	Random variation
Reaction kinetics	Slow drift
Reactor cooling water valve	Sticking
Condenser cooling water valve	Sticking

Table 3.5. Process Operating Constraints (Jockenhövel *et al.*, 2003).

Process variable	Normal operating limits		Shut down limits	
	Low limit	High limit	Low limit	High limit
Reactor pressure (kPa)	None	2895	None	3000
Reactor level (m^3)	11.8	21.3	2.0	24.0
Reactor temperature (K)	None	423	None	448
Separator level (m^3)	3.3	9.0	1.0	12.0
Stripper base level (m^3)	3.5	6.6	1.0	8.0

Dynamic simulations of the TEP were performed in MATLAB by giving different fault disturbances to the process to obtain the data set to study its compression. The

Simulink code, "MultiLoop Skoge Model 1" available at Prof. Ricker's home page (http://depts.washington.edu/control/LARRY/TE/download.html) was executed in MATLAB to simulate the closed-loop dynamic behavior of the plant. In configuration parameters, solver type was set to "ODE 1" which is the Euler method by default. Simulation time and sample rate were taken as 500 hours and 100 samples per hour respectively. There are eight different disturbance options. The TEP simulation begins without any faults, however a disturbance was given after 100 hours and the simulation proceeds without any disturbances at the end of a time period of 100 hours. Thus, in this way, three different consecutive disturbances were given to the system till the end of the simulation and finally the output file having 50001 rows and 41 columns of about 23 MB size, containing sensor measurements from all available spots, was obtained.

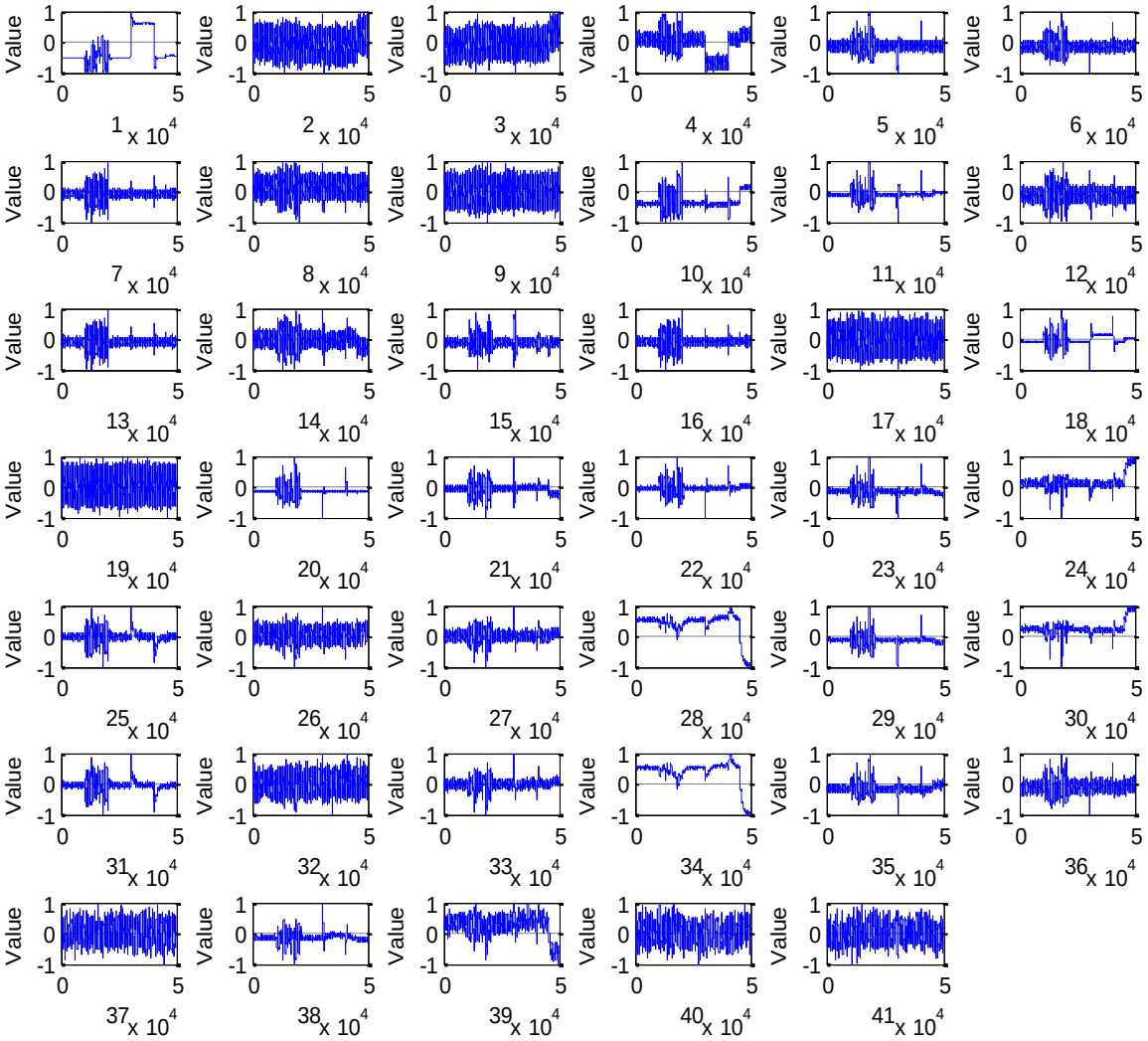

Figure 3.5. Complete Output Signals of the TEP as a result of Three Consecutive Fault Disturbances in Simulink in Scaled Format.

3.4. Correlation Properties of the Data Sets

Correlation matrix is generated by calculating the correlation coefficients between the columns of a data set. The diagonal elements of this symmetric matrix are one denoting the correlation of a column with itself. There is a high correlation between columns if most of the correlation coefficients are close to either one or minus one (minus values indicate inverse correlation), whereas coefficients approaching zero imply less correlation. In this section, correlation matrix of each data set mentioned above (PortSimHigh, PortSimLow, SELDI-TOF MS and TEP) is calculated and histograms of the matrix coefficients excluding the diagonal elements are plotted.

As far as Figure 3.6 is concerned, it is seen that almost all of the correlation coefficients of the PortSimHigh data set are around one, meaning that this data set has the highest correlation between its columns, whereas the correlation coefficients of the PortSimLow data set spread over the interval [-1,1] indicating less correlation. There is virtually no correlation between columns of both TEP and SELDI-TOF MS data sets as all of their correlation coefficients are close to zero. It can also be stated that there are equal number of column pairs with positive and negative valued coefficients of the TEP data set since the correlation coefficients are distributed symmetrically around zero. The effect of the correlation properties of these data sets will be investigated for compression/reconstruction studies in the following chapters.

The data sets to be used in compression studies can be characterized in terms of smoothness, noise content and frequency distribution of their correlation coefficients as shown in Table 3.6.

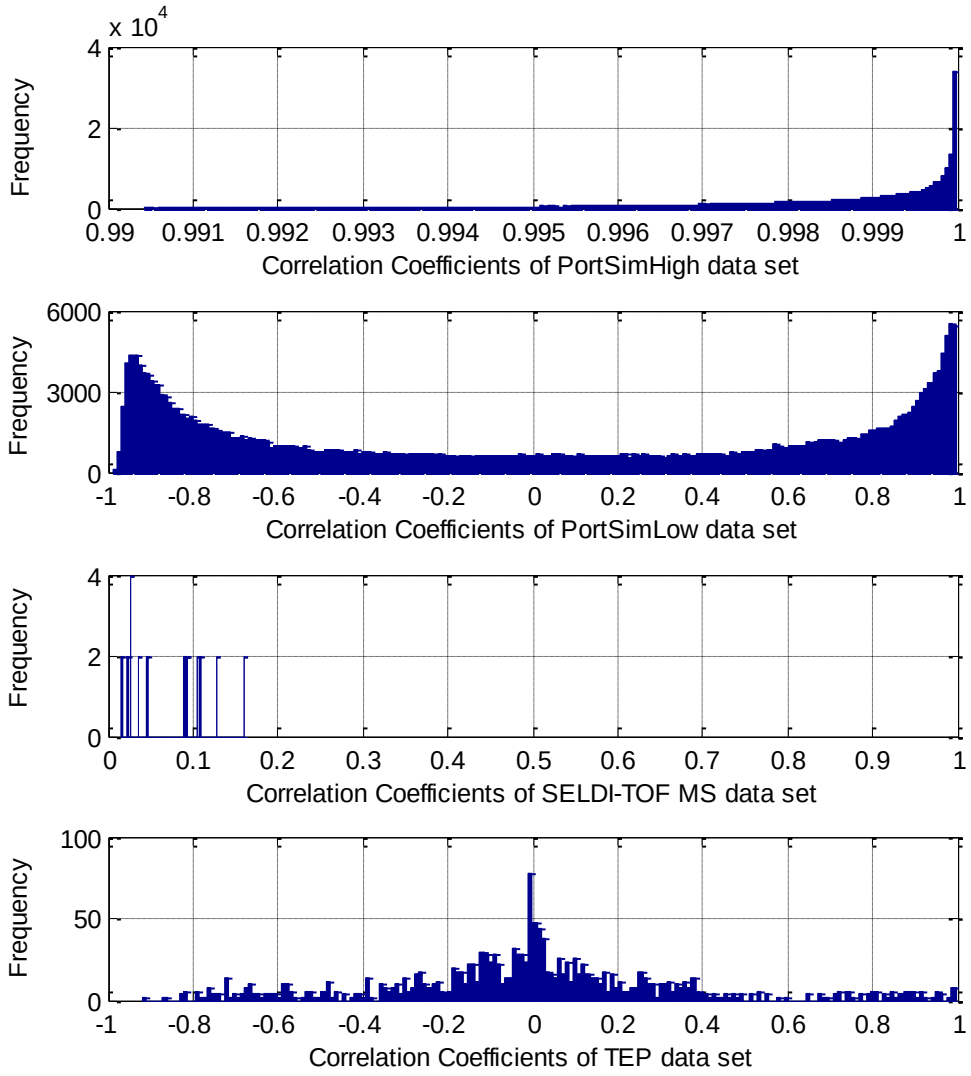

Figure 3.6. The Frequency Distributions of the Columnwise Correlation Coefficients of the Data Sets.

Table 3.6. Characteristics of the Data Sets used.

Data Set	Characteristics
PortSimHigh	Increasing and decreasing trend White noise Highly correlated
PortSimLow	Increasing and decreasing trend White noise Less correlated, including inverse correlation
SELDI-TOF MS	Sparse sharp peaks Significant baseline noise Highly uncorrelated
TEP	Upward and downward jumps at the location of fault disturbances Mixture of almost pure noise and level jumps in trendy signals Highly uncorrelated, including inverse correlation

4. DATA COMPRESSION VIA PIECEWISE AGGREGATE APPROXIMATION AND DATA QUANTIZATION

4.1. Piecewise Aggregate Approximation

Piecewise Aggregate Approximation (PAA) proposed by Keogh *et al.* (2000) is a dimensionality reduction technique in which data set in n dimensions is divided into w equal sized frames in each of which the data portions are represented with their arithmetic averages. As a result, n dimensional data set is reduced to w dimensions by composing another (approximating) data set consisting of the mean values as horizontal line segments.

A vector **c** of length n can be represented by another vector **x** having w dimensions and the i^{th} element of **x** is computed by the following equation (Keogh *et al.*, 2000);

$$x_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j \tag{4.1}$$

where $1 \leq w \leq n$

The segmented data set composed by the PAA cannot be reconstructed like other transformed data sets (such as those obtained via DCT or DWT), and for this reason, the PAA is a lossy compression method where the loss depends on the segment size. If only a few large segments are used, some important patterns in the original data can be lost. However, high compression ratios can be achieved by keeping only the segment coordinates instead of the point values on the horizontal segments. Maximum compression ratios are expected as the segmented data consist of repeating patterns which are favored in lossless compression algorithms, such as the ones used in ZIP or RAR archiving softwares.

In this chapter, the PAA technique is studied by using the 50^{th} column of the PortSimHigh data set, second column of the SELDI-TOF MS data set and the 30^{th} column of the TEP data set for 10 different segment sizes in the [15-150] range. The effect of the frame size on compression is measured in terms of compression ratio, error norm and the

Shannon entropy. Internal ZIP command in MATLAB version 7.7.0 (R2008b) is used as the lossless compression algorithm.

Two vectors are composed in PAA; **x** in w dimensions used in entropy calculation and **y** in n dimensions used in compression ratio and error norm calculations. Plots of these vectors composed for frame sizes 15 and 150 and the original data set are given in Figure 4.1 and Figure 4.2 for the PortSimHigh data set and Figure 4.5 and Figure 4.6 for the TEP data set. However, frame sizes are taken as 15 and 1000 for the SELDI-TOF MS data set as shown in Figure 4.3 and Figure 4.4. Plots of the complete output signals of the TEP data set and their segmented values generated for 150 and 1000 segments are also given in Figure 4.7, Figure 4.8 and Figure 4.9 respectively.

As far as Figure 4.1 and Figure 4.2 are concerned, it is seen that the segmented data set is not continuous, unlike the relatively smooth original data containing white noise. Besides, as the number of frames decreases, more data, especially the peak points, are missed. However, PAA is a useful method for visualization of the raw data. For instance, even as low as 15 segments as illustrated in Figure 4.1 may be considered as adequate to follow the major trends of the original data set of length 10000 visually. On the other hand, for 150 segments, the segmented values (bottom sub-window of Figure 4.2 that contains 150 points) and the original data (upper sub-window of Figure 4.2 that contains 10000 points) may even be undistinguishable to human eye.

As it can be seen from Figure 4.3 and Figure 4.4, in the SELDI-TOF MS data, there are sparse sharp peaks and significant baseline noise. For this reason, segmented values appear almost as a horizontal line for 15 segments since the sharp peak points are missed as shown in Figure 4.3. However, for 1000 segments, locations of the major peak points can be detected by segmented values (bottom sub-window of Figure 4.4) illustrating the process trends of the original data set (upper sub-window of Figure 4.4) of length 337988 visually.

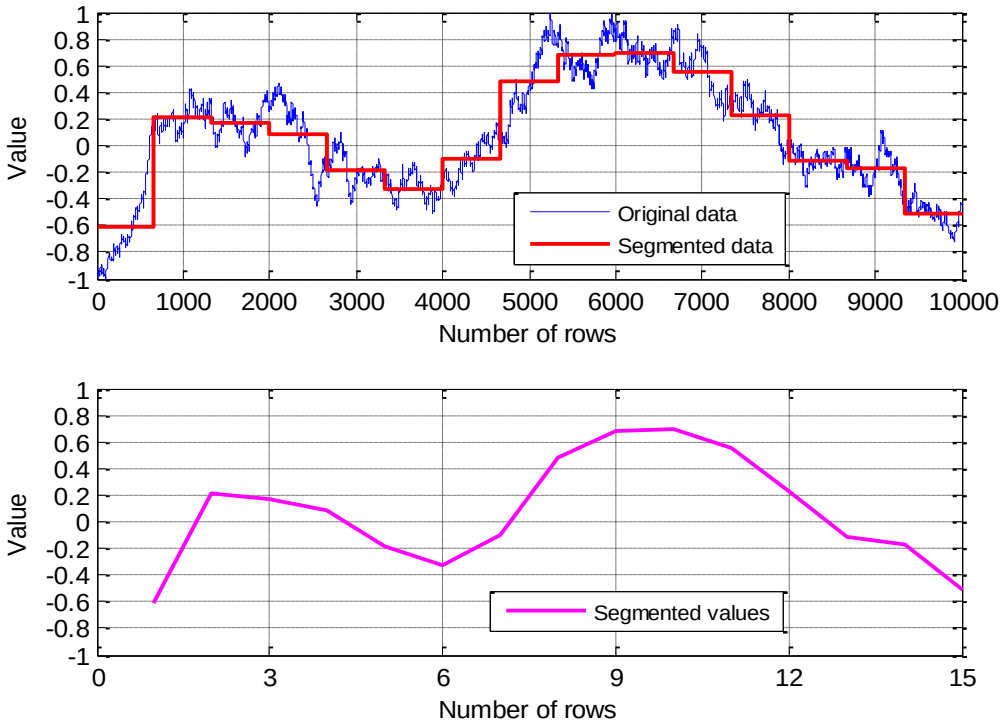

Figure 4.1. Original versus Segmented Data with 15 Segments using the 50th column of the PortSimHigh Data Set.

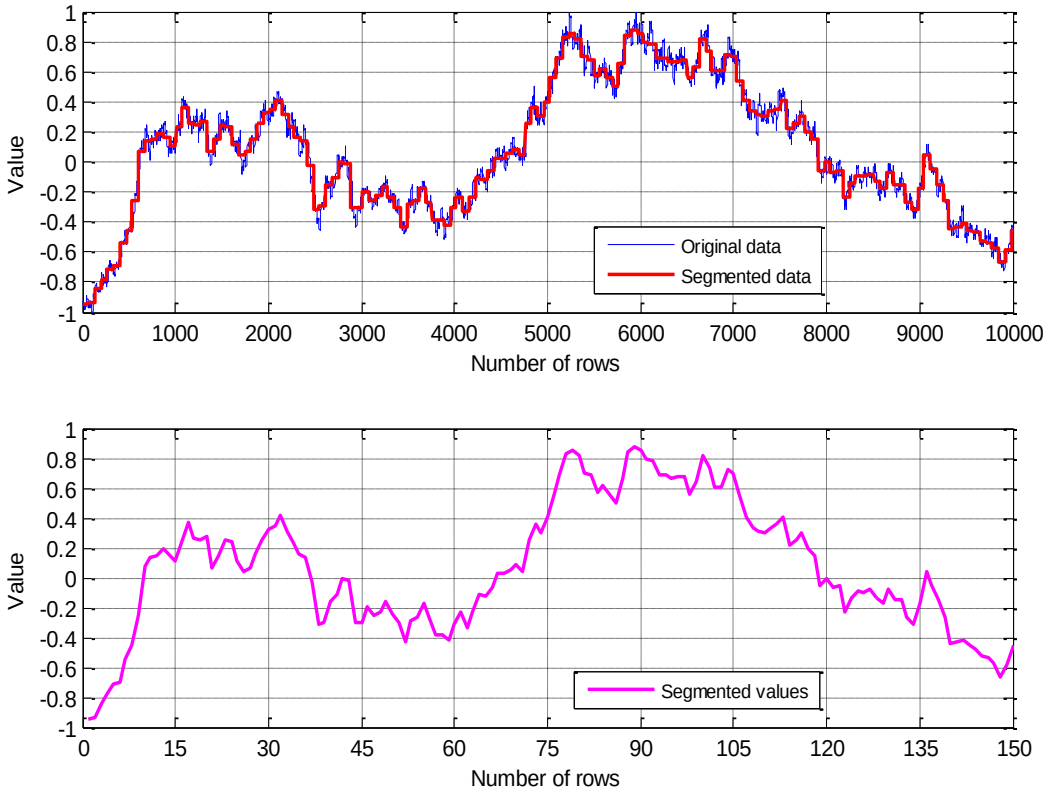

Figure 4.2. Original versus Segmented Data with 150 Segments using the 50th column of the PortSimHigh Data Set.

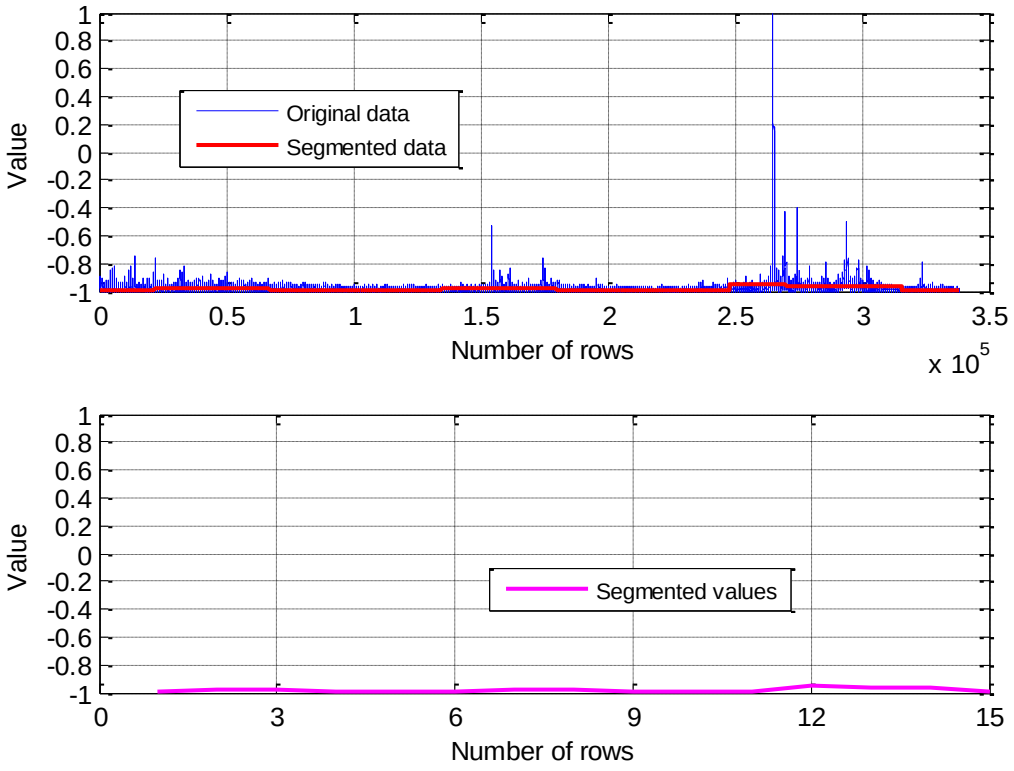

Figure 4.3. Original versus Segmented Data with 15 Segments using the Second column of the SELDI-TOF MS Data Set.

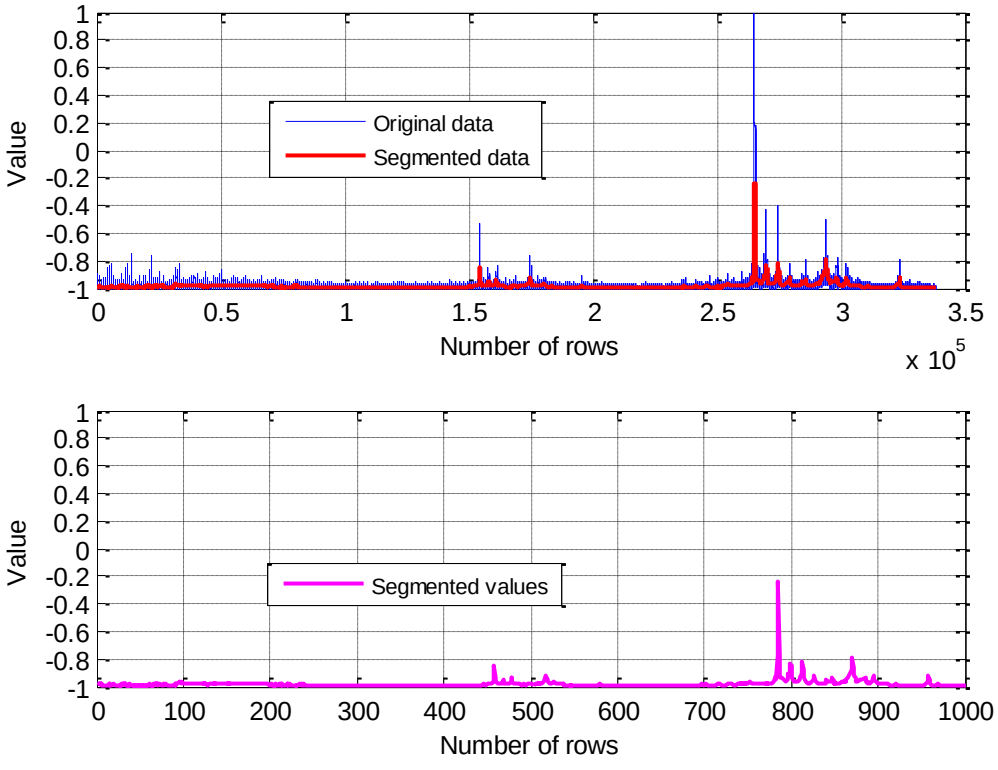

Figure 4.4. Original versus Segmented Data with 1000 Segments using the Second column of the SELDI-TOF MS Data Set.

As it can be seen from Figure 4.5 and Figure 4.6, the TEP data points are too close to each other and there are peaks and shifts in the data set. It is difficult to divide this noisy data set containing non-uniformly occurring peaks into equal sized frames, and thus, large segment sizes are required to visualize the original data thoroughly. For this reason, 15 segments are not adequate to illustrate the major trends of the original data set of length 10000 visually as shown in Figure 4.5. However, segmented values containing 150 points (bottom sub-window of Figure 4.6) do show peaks and shifts besides the process trends approximating the original data (upper sub-window of Figure 4.6).

Output signals of the TEP including all of the 41 measured variables and their segmented values generated for 150 and 1000 segments are shown in Figure 4.7, Figure 4.8 and Figure 4.9 respectively. 50001 dimensional data set is reduced to 150 and 1000 dimensions by composing the segmented data set consisting of the mean values of the horizontal line segments. The segmented data set composed by using 1000 segments approximates the output signals better than the data set consisting of 150 points as expected, since the loss in compression decreases as the segment size increases. However, segmented values consisting of only 150 points instead of the original data set of length 50001 are adequate for the visualization of the raw data which is favored in specific applications such as process monitoring and fault detection, as the major process events including important peak points, upward/downward shifts (observed in measurements one and four) and decreasing/increasing trends (observed in measurements 28 and 34) occurred due to the consecutive fault disturbances can easily be followed by plant operators as illustrated in Figure 4.8.

In addition, noise removal is one of the major advantages of the dimensionality reduction technique, PAA as it can be seen from Figure 4.8 that for 150 points, segmented values of the measurements consisting of almost pure noise (measurements nine and 19) are smoothened around a straight line. However, the amplitudes of the segmented data set generated by using 1000 segments are very close to those of the output signals of the TEP, thus noisy measurements cannot be removed thoroughly as it can be seen from Figure 4.9 although original data set is approximated perfectly.

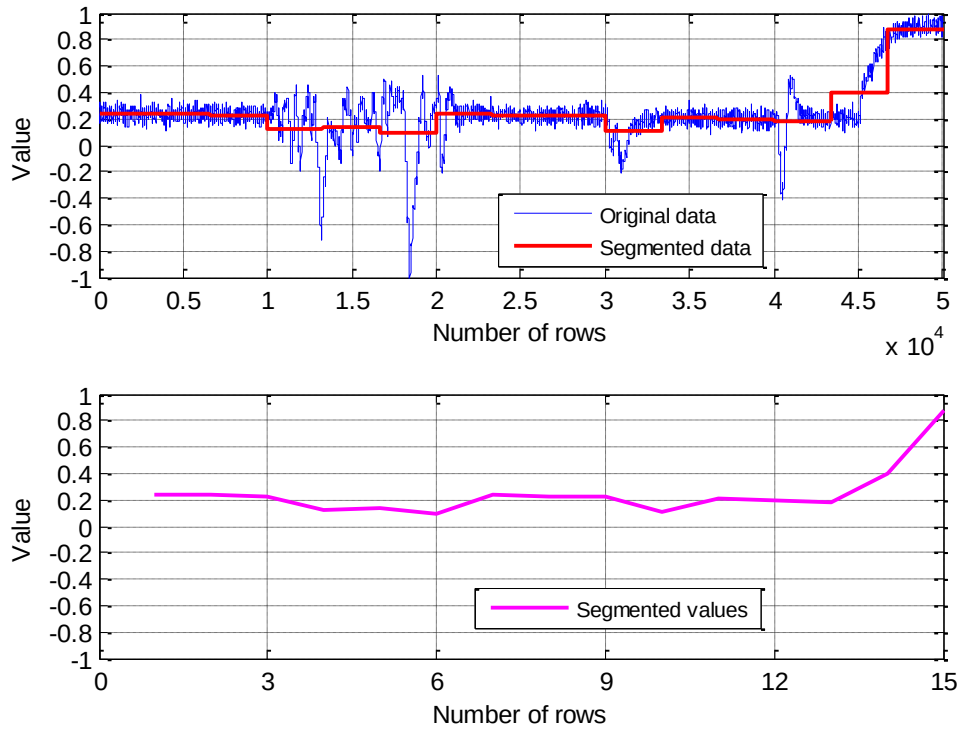

Figure 4.5. Original versus Segmented Data with 15 Segments using the 30th column of the TEP Data Set.

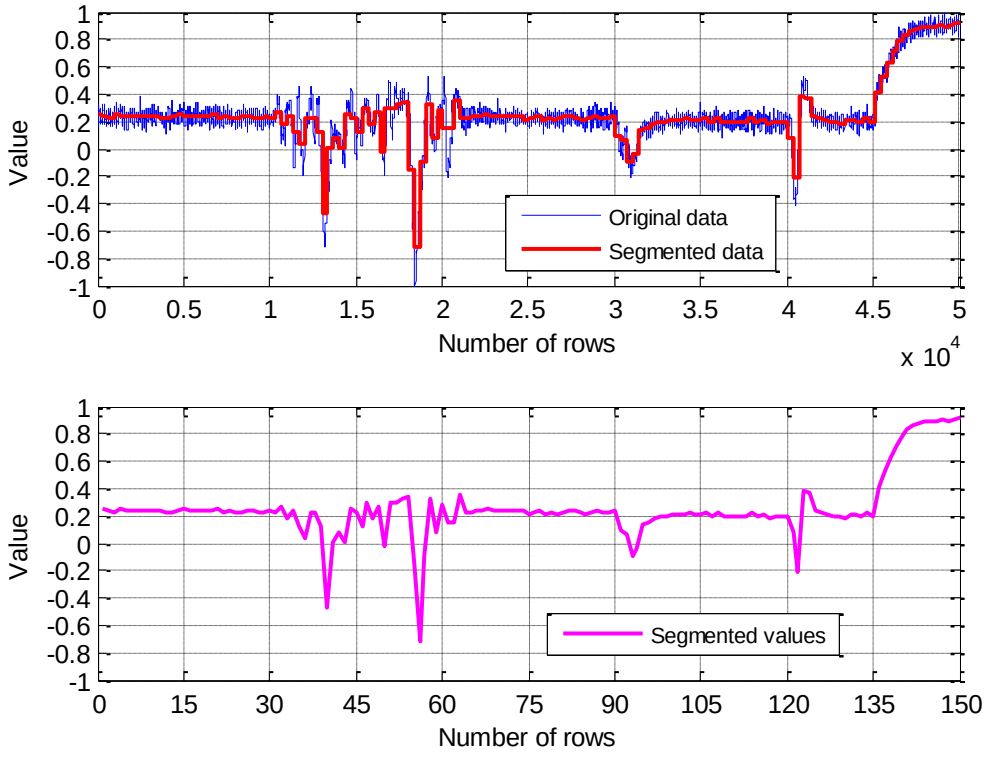

Figure 4.6. Original versus Segmented Data with 150 Segments using the 30th column of the TEP Data Set.

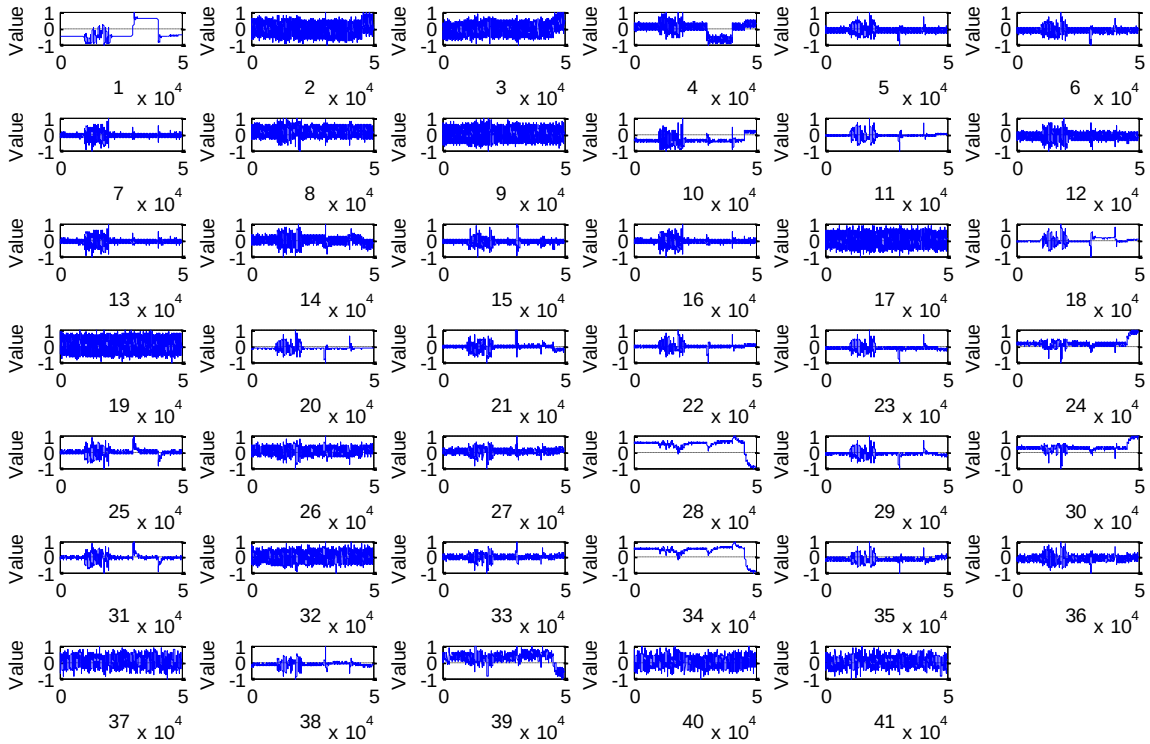

Figure 4.7. Complete Output Signals of the TEP as a result of Three Consecutive Fault Disturbances in Simulink in Scaled Format.

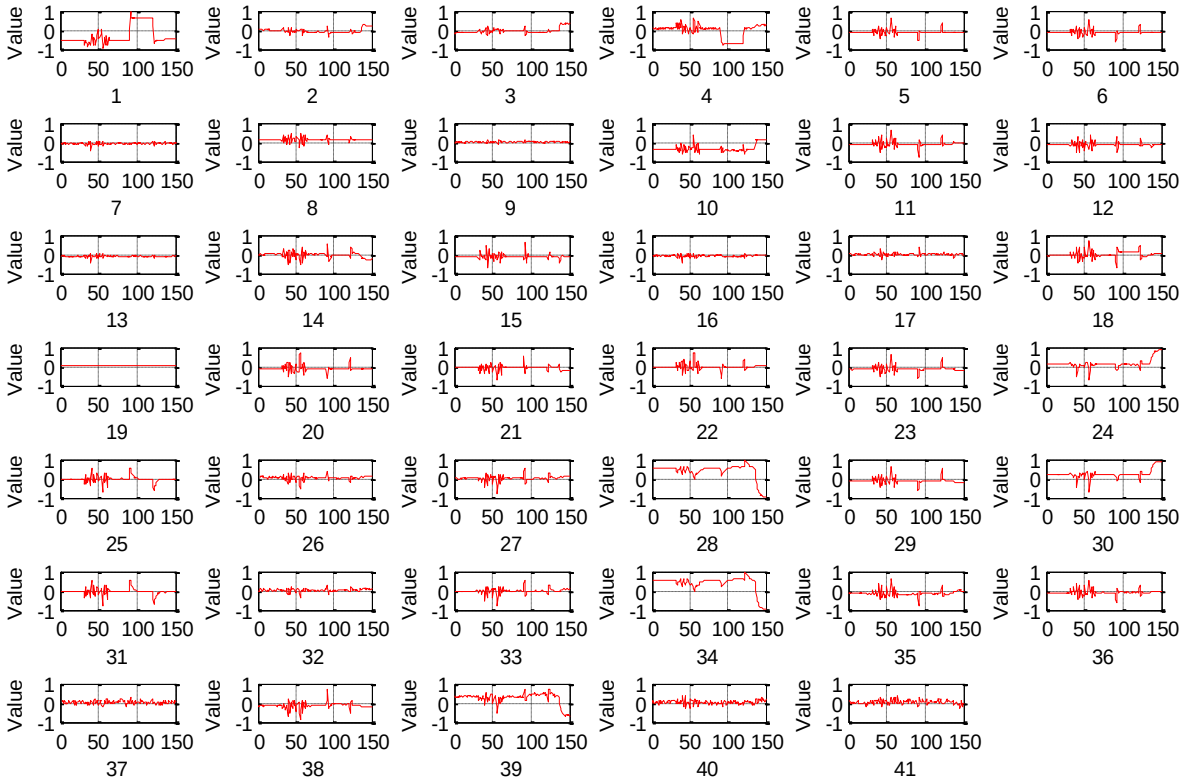

Figure 4.8. Segmented Values of the TEP Output Signals with 150 Segments as a result of Three Consecutive Fault Disturbances in Simulink in Scaled Format.

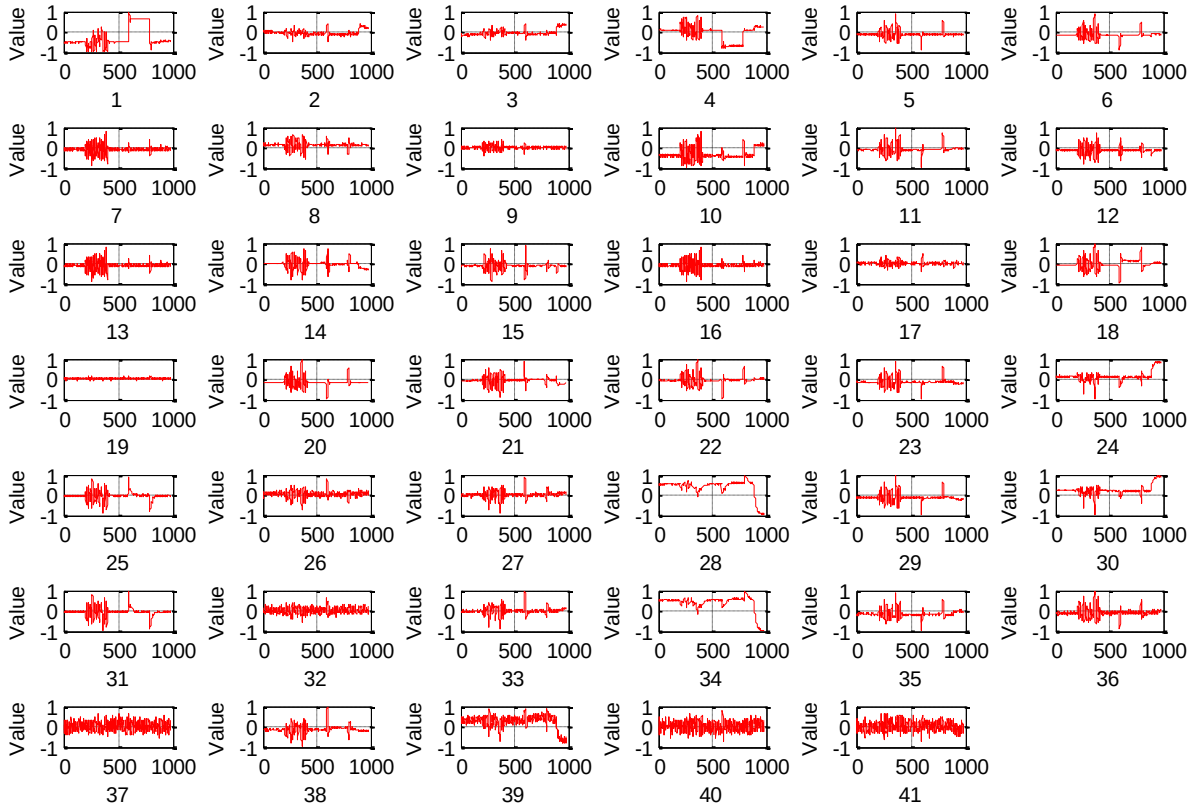

Figure 4.9. Segmented Values of the TEP Output Signals with 1000 Segments as a result of Three Consecutive Fault Disturbances in Simulink in Scaled Format.

Compression ratios, error norms and entropies of the segmented data (as computed using the equations given in Chapter 2) for different frame sizes are given in Figure 4.10, Figure 4.11 and Figure 4.12 for three data sets, respectively. Compression ratios and error norms increase steadily as the number of segments decreases. Reduction ratios of the segmented data set composed from both the TEP and SELDI-TOF MS data are much lower than those of the PortSimHigh data due to their high noise content with sudden changes. As a result, it is more difficult to visualize these two data sets by the PAA with higher error norms, compared to PortSimHigh data set.

Since the entropy of the original data is independent of the frame size, it is shown as constant via the thick horizontal line. As the number of segments decreases, entropies of the segmented data also decrease since original data set is represented by fewer bits. For instance, the Shannon entropy of the segmented data with 90 frames is almost half of that of the original data, in other words, the information content of the original data is reduced by half, as it can be seen from Figure 4.10, Figure 4.11 or Figure 4.12.

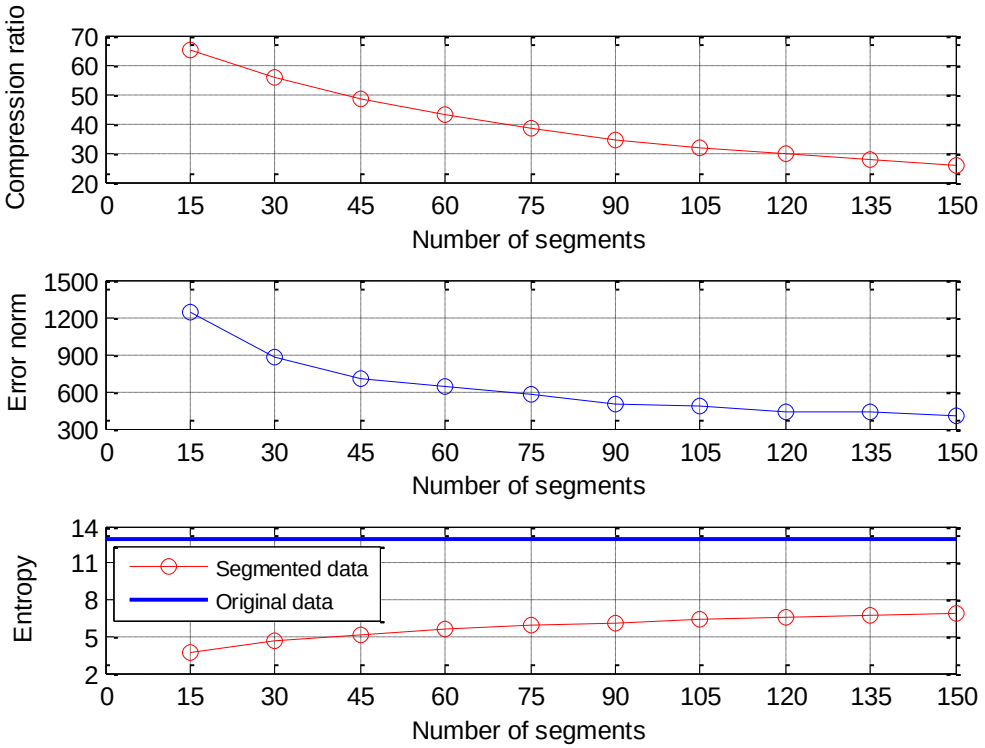

Figure 4.10. Compression Ratio, Error Norm and Entropy versus Number of Segments using the 50th column of the PortSimHigh Data Set.

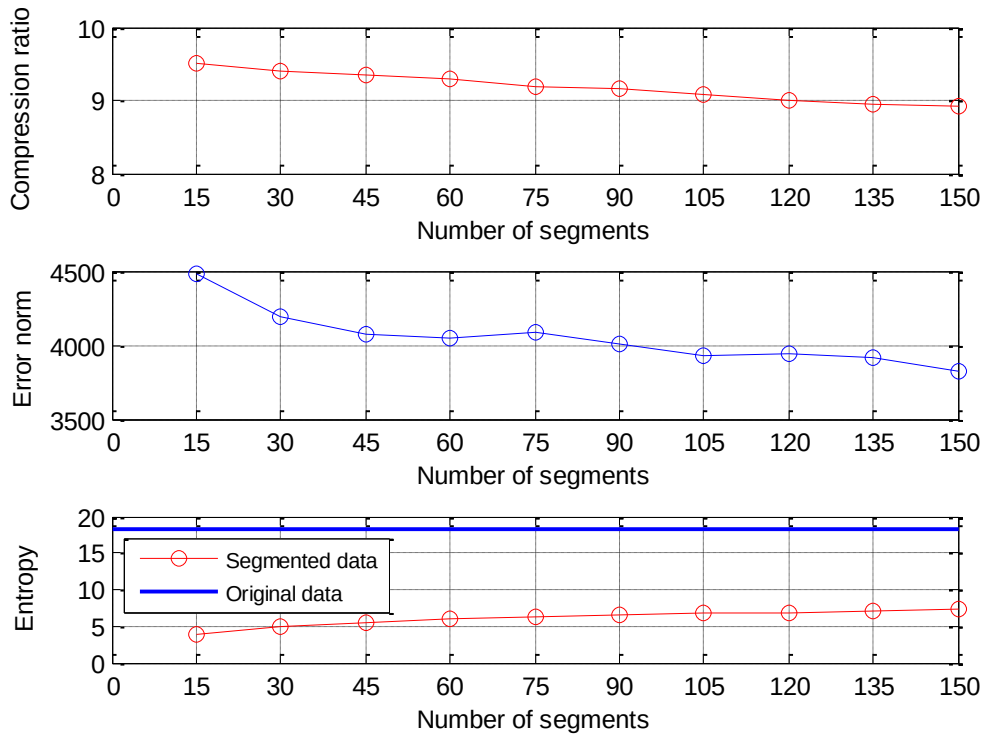

Figure 4.11. Compression Ratio, Error Norm and Entropy versus Number of Segments using the Second column of the SELDI-TOF MS Data Set.

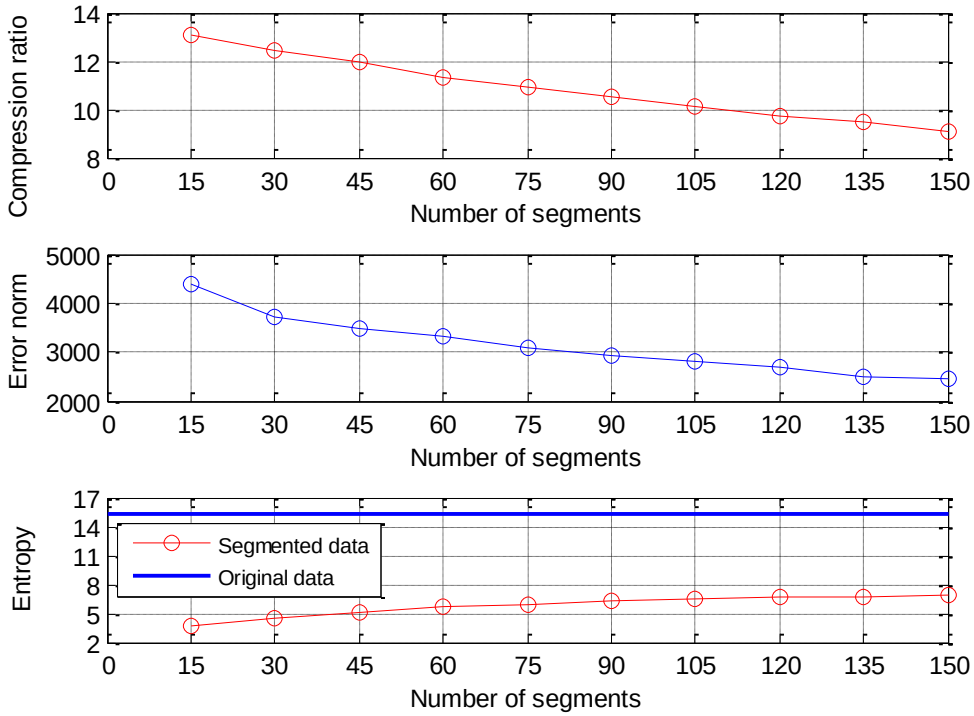

Figure 4.12. Compression Ratio, Error Norm and Entropy versus Number of Segments using the 30th column of the TEP Data Set.

4.2. Data Quantization

Quantization is generally applied to the transformed/dimensionality-reduced data sets for further compression by eliminating insignificant digits (Mitra and Acharya, 2003). Quantization is simply the rounding of numbers to the nearest integer or to a specified decimal place. The remaining digits consisting of repeating zeros can provide large compression. Amount of digit cutting depends on the magnitude of data value. For instance cutting all of the digits after decimal may be reasonable for large numbers such as large magnitude flow rate measurements in units cm^3/day. On the other hand, one or more digits after the decimal should be kept for small numbers such as mole fractions varying in the [0-1] range. Quantization is a lossy method depending on the amount of discarded digits and the quantized data become stepwise if quantization is significant.

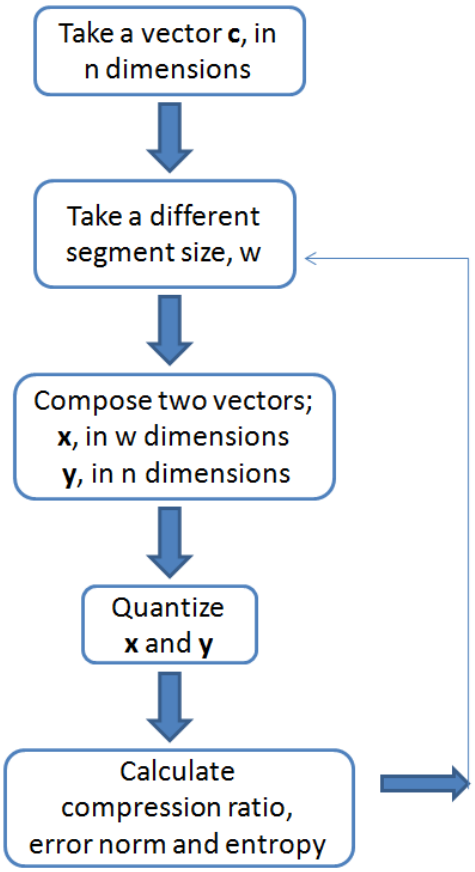

Figure 4.13. The Procedure used in Data Compression via PAA Technique Followed by Quantization.

In this section, quantization is applied to the segmented data set produced by the PAA in order to increase compression ratios a bit more without any significant increase in error norms. The segmented data set consists of numbers having 15 digits after decimal thus insignificant digits can easily be discarded for further compression. The vectors **x** and **y** composed in PAA are quantized by keeping only the first digit and the first three digits after the decimal respectively, then the effect of quantization on compression ratio, error norm and entropy are studied. Optimum frame size used in PAA and optimum number of digits kept after the decimal in quantization are also determined so as to maximize the ratio of compression ratio to error norm.

The segmented 50^{th} column of the PortSimHigh data set with 150 segments consisting of numbers having 15 digits after the decimal and its quantized form generated by keeping only the first three digits after the decimal are given in Figure 4.14. The effect of quantization step is not clear since these two data sets are overlapped.

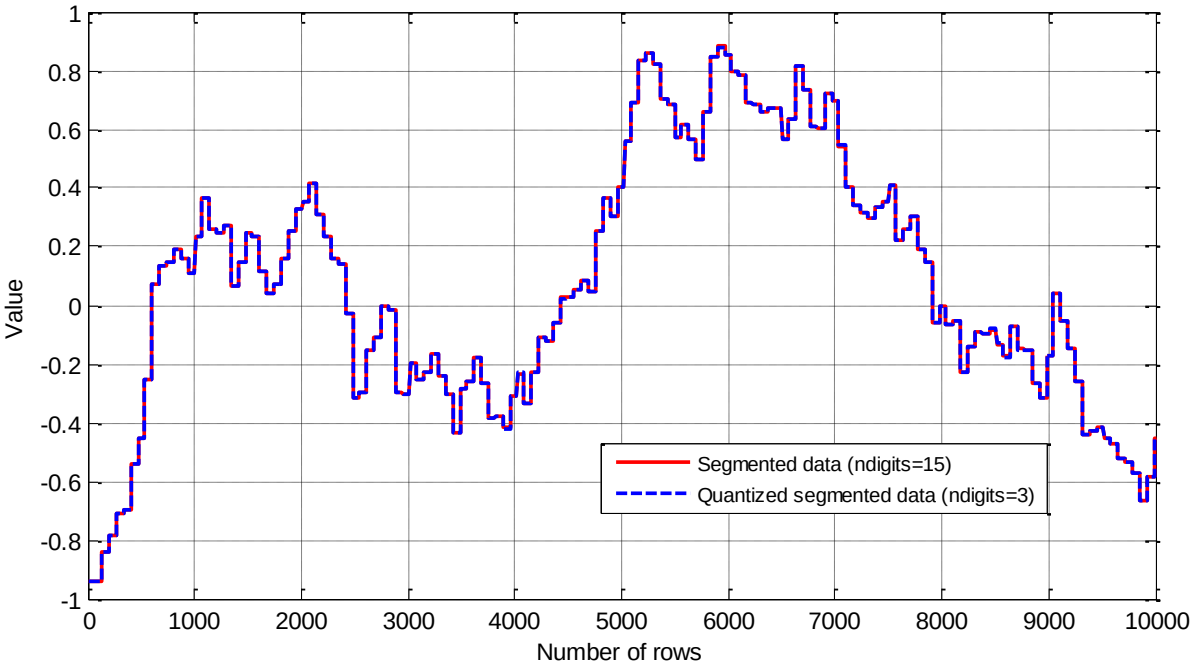

Figure 4.14. Segmented (ndigits=15) versus Quantized Segmented Data (ndigits=3) with 150 Segments using the 50th column of the PortSimHigh Data Set.

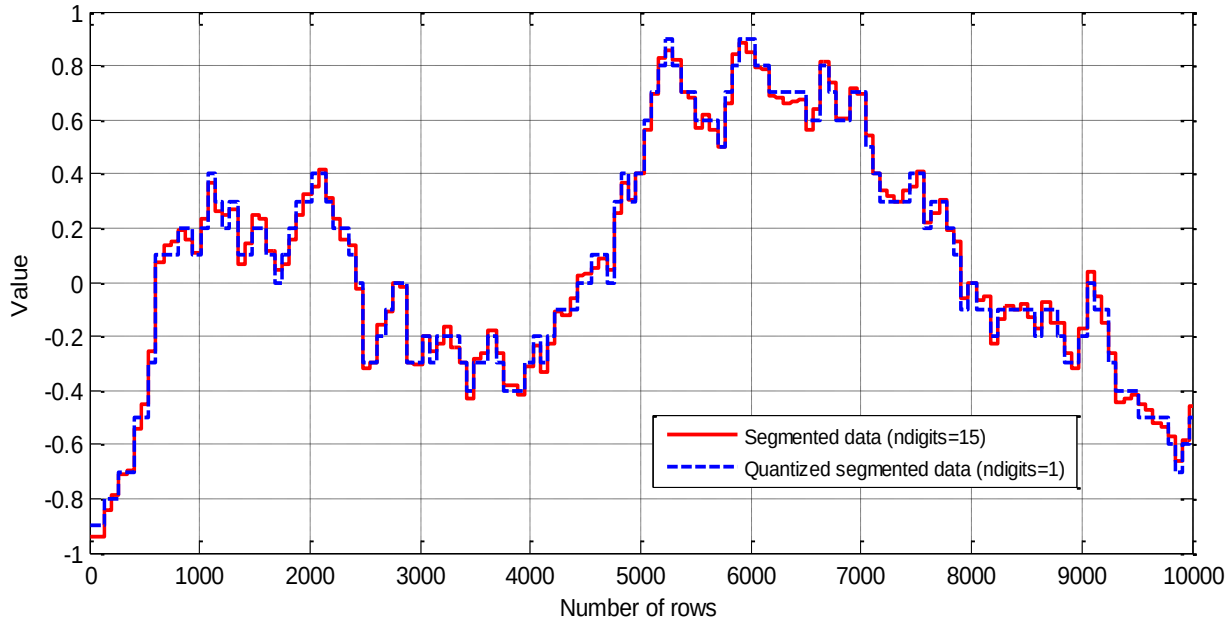

Figure 4.15. Segmented (ndigits=15) versus Quantized Segmented Data (ndigits=1) with 150 Segments using the 50th column of the PortSimHigh Data Set.

As it can be seen from Figure 4.14, there is not a clear distinction between segmented and quantized segmented data sets by taking the number of digits as low as three. For this reason, another plot is given in Figure 4.15 by taking the number of digits used in

quantization step as one. Thus, quantized data become stepwise as the quantization is more significant.

The segmented data set of the second column of the SELDI-TOF MS data composed for 1000 segments consisting of numbers having 15 digits after the decimal and its quantized form generated by cutting all of the digits after the third digit after the decimal are given in Figure 4.16. However, the quantized data set cannot be differentiated from the segmented data set by taking the number of digits as low as three. For this reason, another plot is given in Figure 4.17 by taking the number of digits used in quantization step as one obtaining stepwise quantized data.

The segmented 30th column of the TEP data set with 150 segments consisting of numbers having 15 digits after the decimal and its quantized form generated by keeping only the first three digits after the decimal are given in Figure 4.18. The effect of quantization step cannot be noticed apparently by taking the number of digits as three since these two data sets are overlapped. Thus, the number of digits used in quantization step is taken as one in order to observe the significance of the quantization step as illustrated in Figure 4.19.

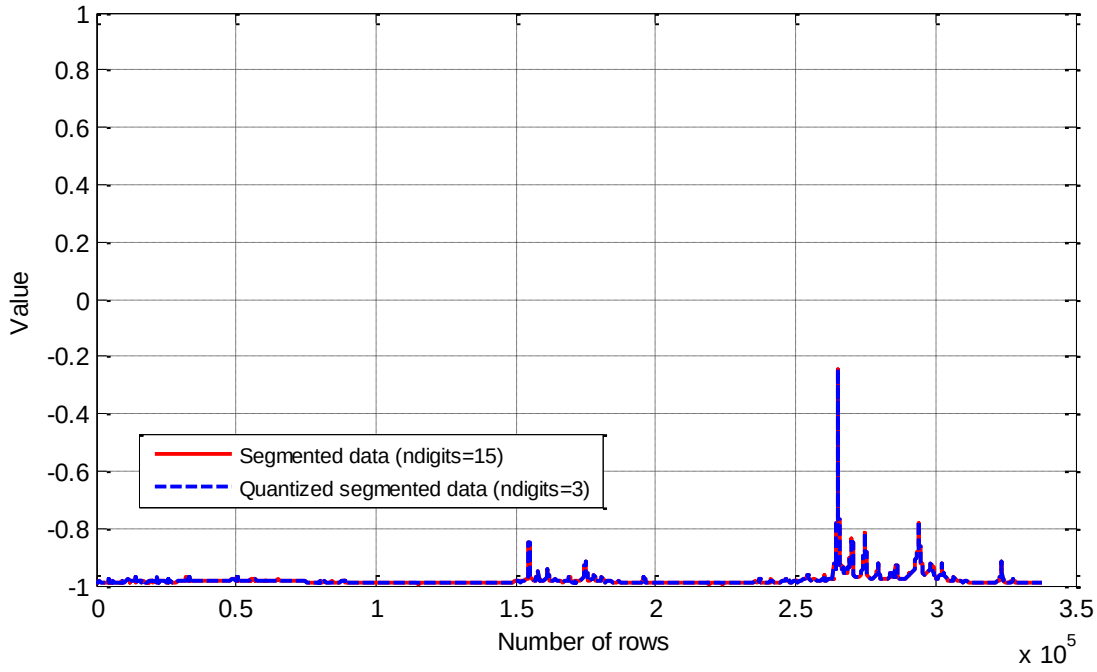

Figure 4.16. Segmented (ndigits=15) versus Quantized Segmented Data (ndigits=3) with 1000 Segments using the Second column of the SELDI-TOF MS Data Set.

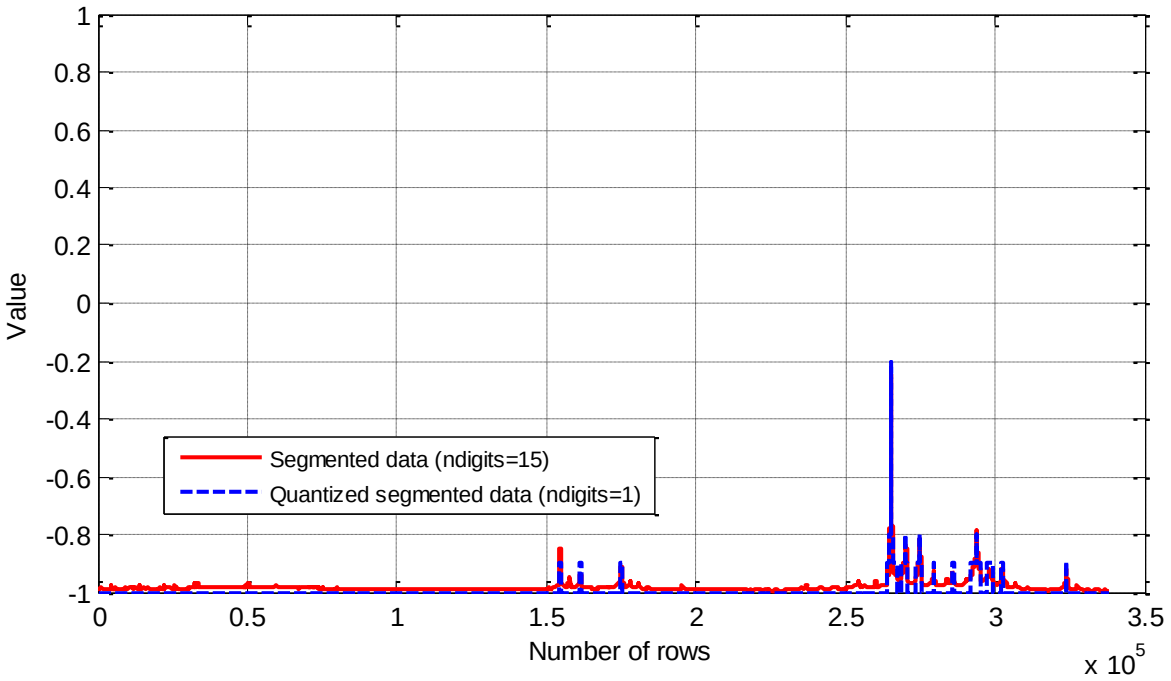

Figure 4.17. Segmented (ndigits=15) versus Quantized Segmented Data (ndigits=1) with 1000 Segments using the Second column of the SELDI-TOF MS Data Set.

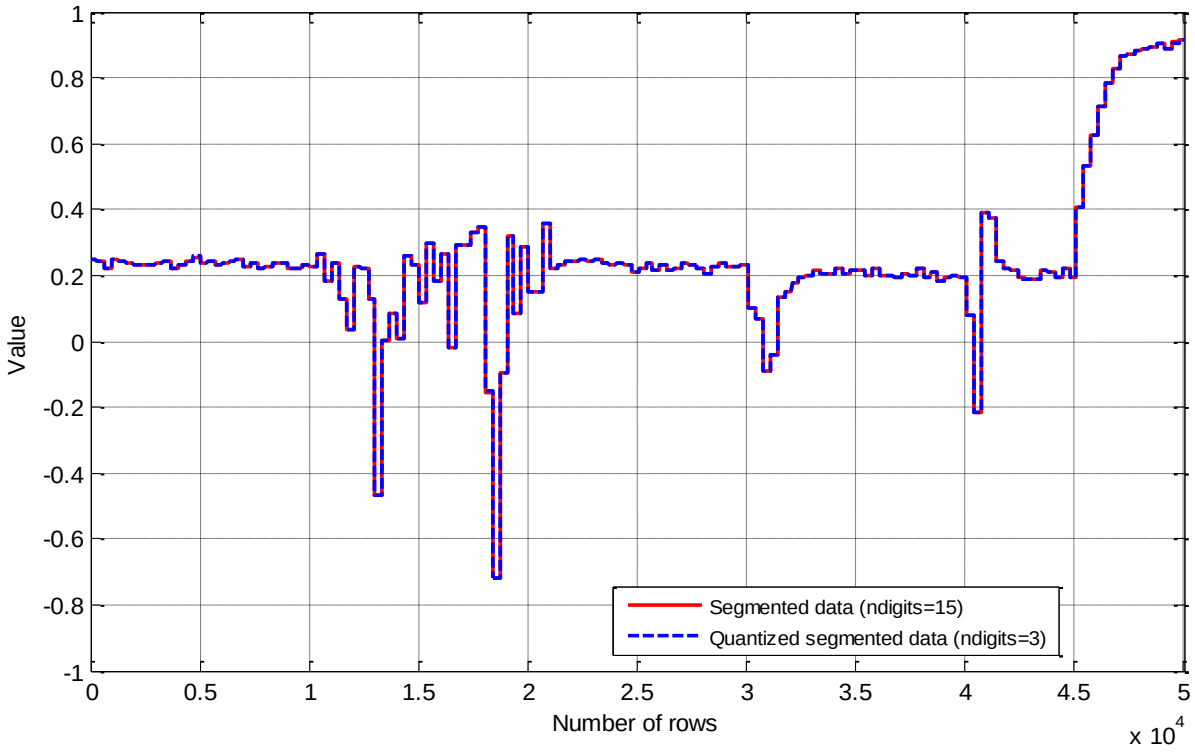

Figure 4.18. Segmented (ndigits=15) versus Quantized Segmented Data (ndigits=3) with 150 Segments using the 30th column of the TEP Data Set.

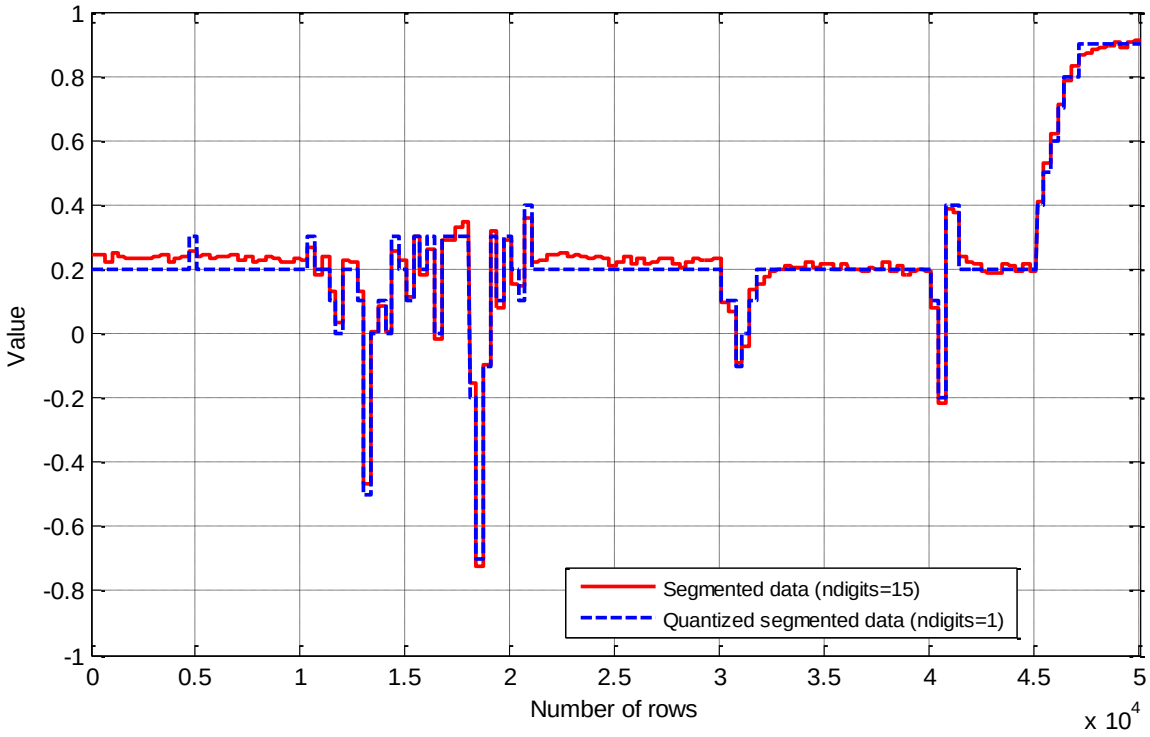

Figure 4.19. Segmented (ndigits=15) versus Quantized Segmented Data (ndigits=1) with 150 Segments using the 30th column of the TEP Data Set.

Figure 4.20 and Figure 4.22 show that higher compression ratios and error norms are obtained as the number of digits kept after decimal in quantization step decreases for the PortSimHigh and TEP data sets respectively. Inversely, smallest compression ratios are obtained by taking the number of digits as one for the SELDI-TOF MS data set as illustrated in Figure 4.21 as a horizontal line independent of the frame size. The reason is that the values of this segmented data are around -0.96 and by taking the number of digits as one, almost all of the segmented values become -1. Consequently, the desired compression level cannot be obtained as there are no repeating zeros replacing the discarded digits. The effect of quantization step becomes more clear as the number of segments used in the PAA increases. Error norms are not affected too much from quantization as numbers in segmented data are just rounded off, whereas compression ratio can almost be doubled with frames larger than 120 by using quantization step after applying the PAA as it is seen from Figure 4.20.

Entropies of the quantized data with one digit cannot be computed for the PortSimHigh and TEP data sets since small numbers become zero after quantization and the logarithm of zero is undefined. For that reason, entropies of the original data, segmented data with 15 digits and quantized segmented data with three digits are given for these two data sets. Entropy of the segmented data is not affected from quantization step as it is seen from Figure 4.20, Figure 21 and Figure 4.22, in other words information content of the segmented data remains the same after quantization.

Figure 4.23, Figure 4.24 and Figure 4.25 are given to determine the optimum frame size used in the PAA and the optimum number of digits kept after decimal in quantization so as to maximize the ratio of compression ratio to error norm. It is seen that optimum frame size is 150 with one digit for the TEP data, whereas the optimum point is obtained with 120 frames and one digit for the PortSimHigh data and 150 frames with three digits for the SELDI-TOF MS data.

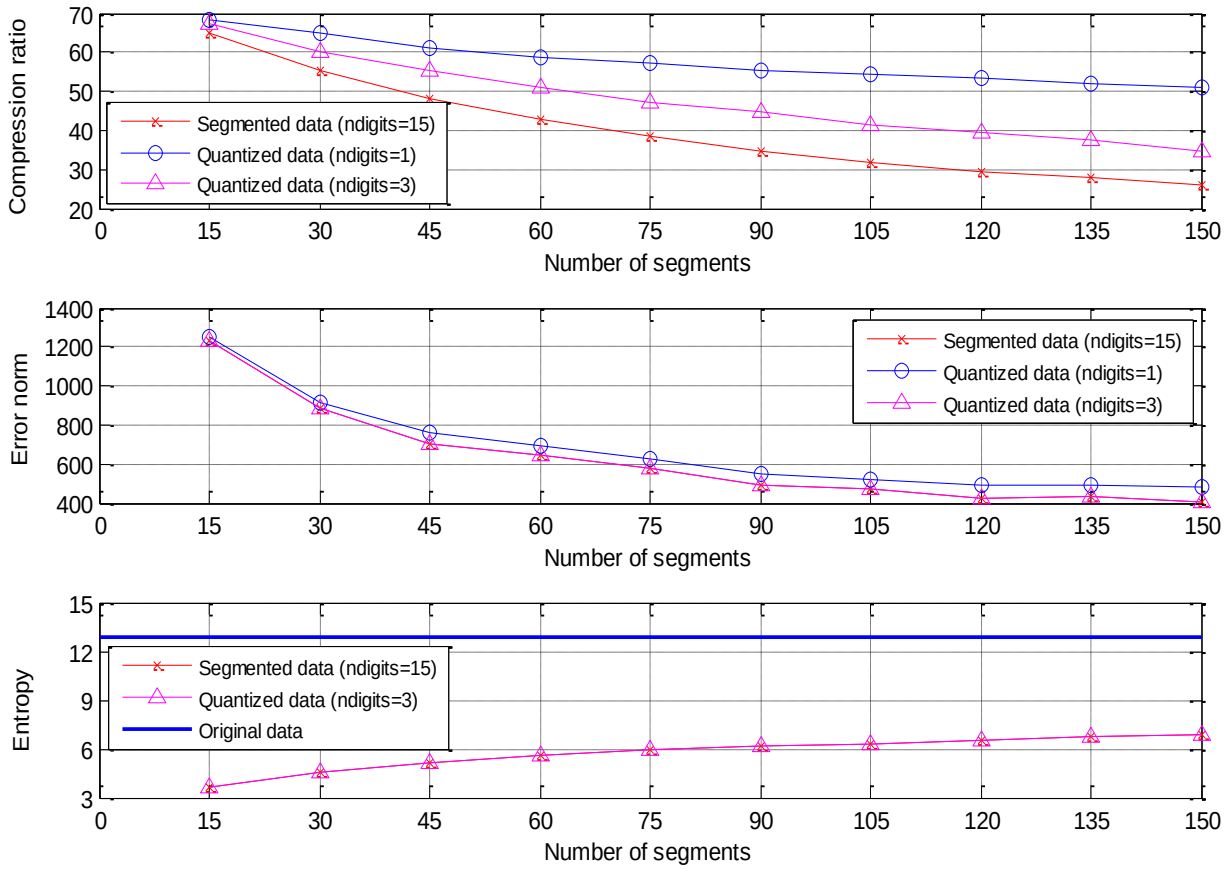

Figure 4.20. Segmented (ndigits=15) versus Quantized Segmented Data (ndigits=1 and ndigits=3) Results using the 50th column of the PortSimHigh Data Set.

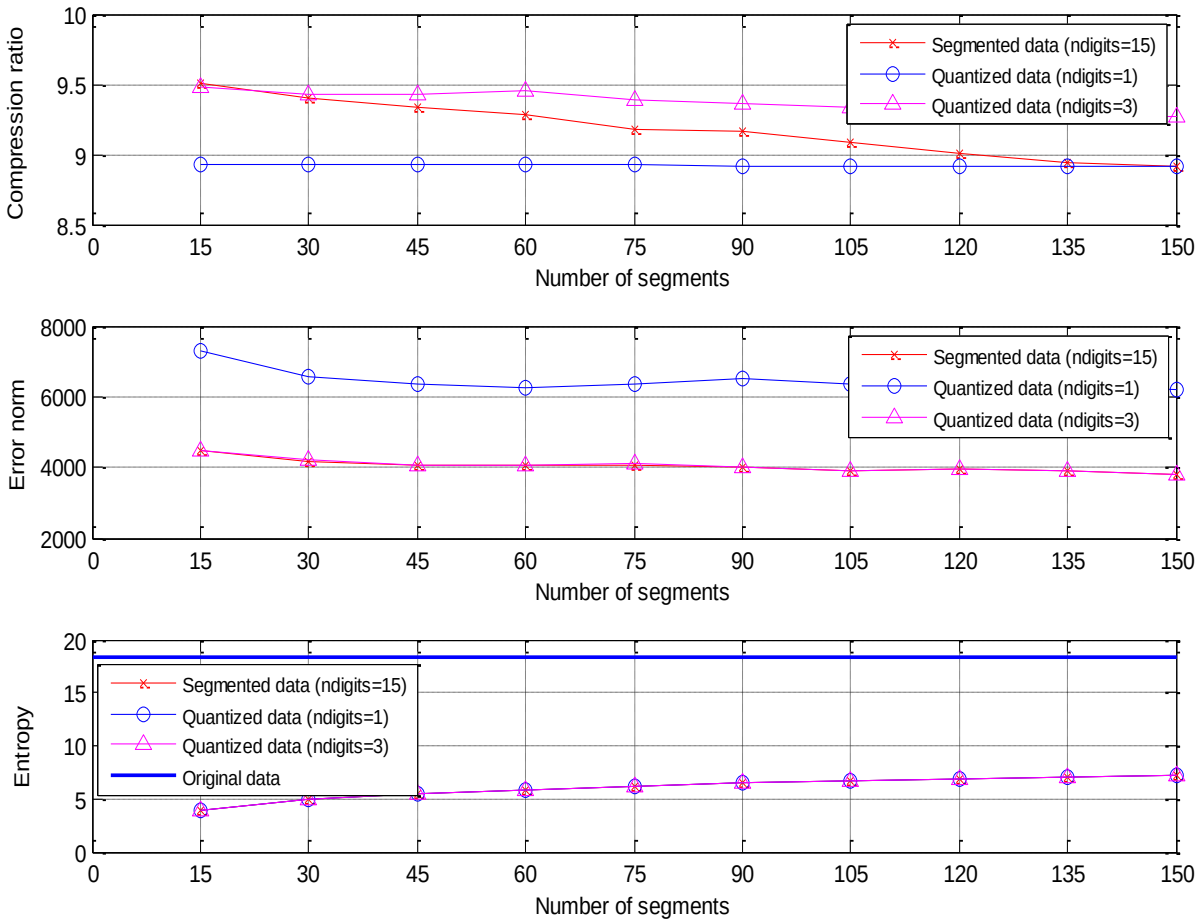

Figure 4.21. Segmented (ndigits=15) versus Quantized Segmented Data (ndigits=1 and ndigits=3) Results using the Second column of the SELDI-TOF MS Data Set.

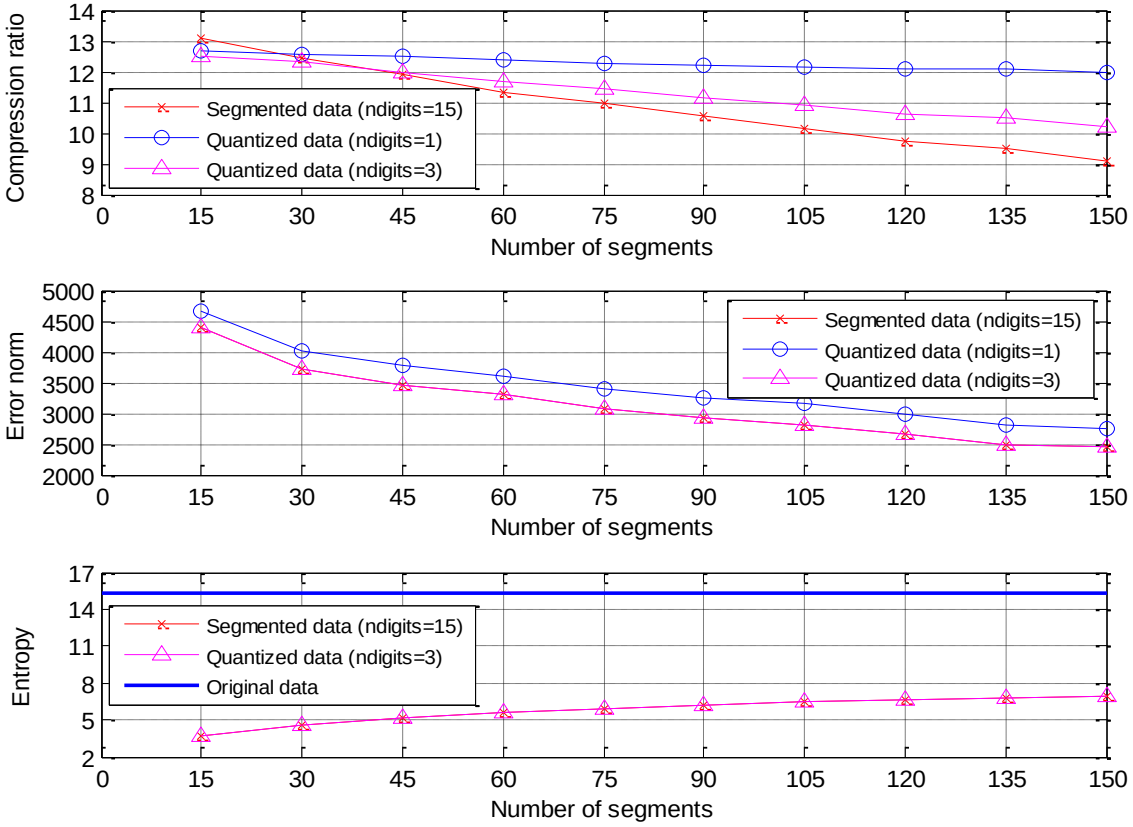

Figure 4.22. Segmented (ndigits=15) versus Quantized Segmented Data (ndigits=1 and ndigits=3) Results using the 30th column of the TEP Data Set.

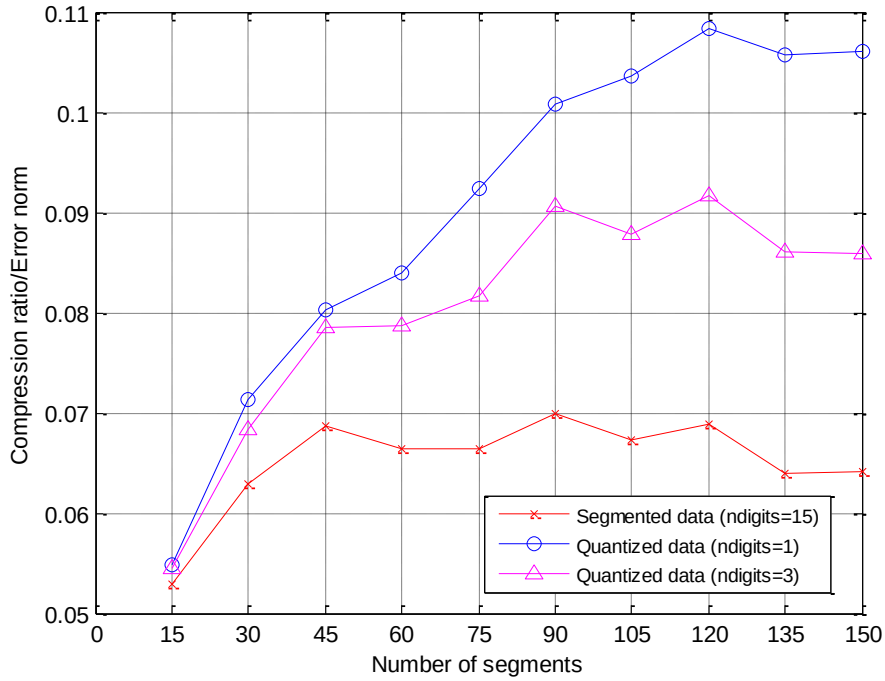

Figure 4.23. Compression Ratio/Error Norm versus Number of Segments using the 50th column of the PortSimHigh Data Set.

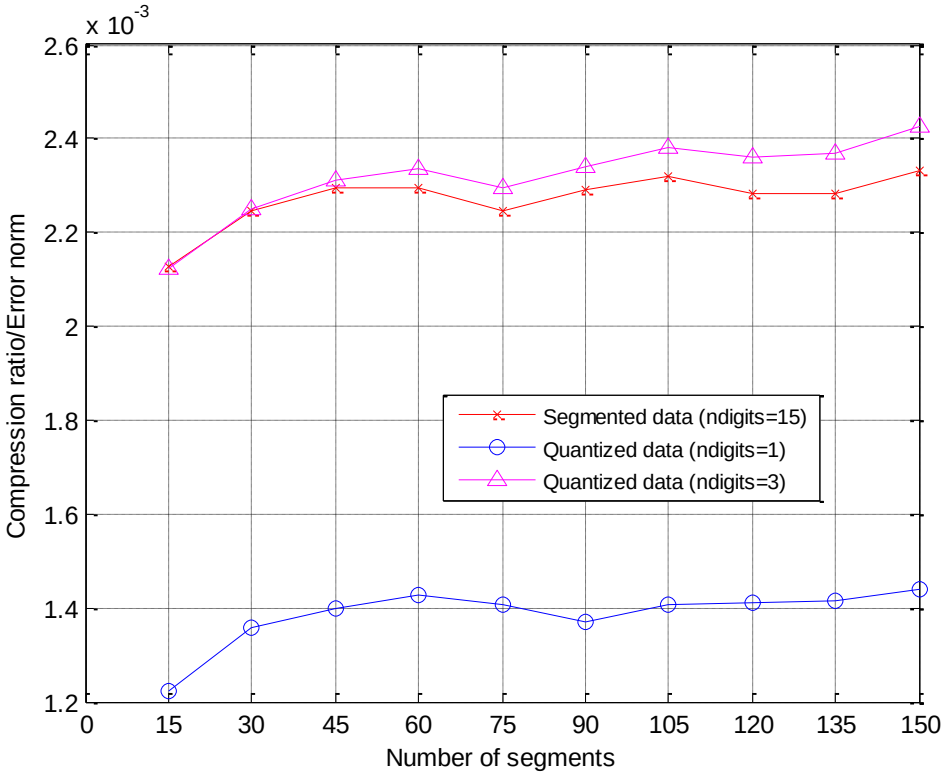

Figure 4.24. Compression Ratio/Error Norm versus Number of Segments using the Second column of the SELDI-TOF MS Data Set.

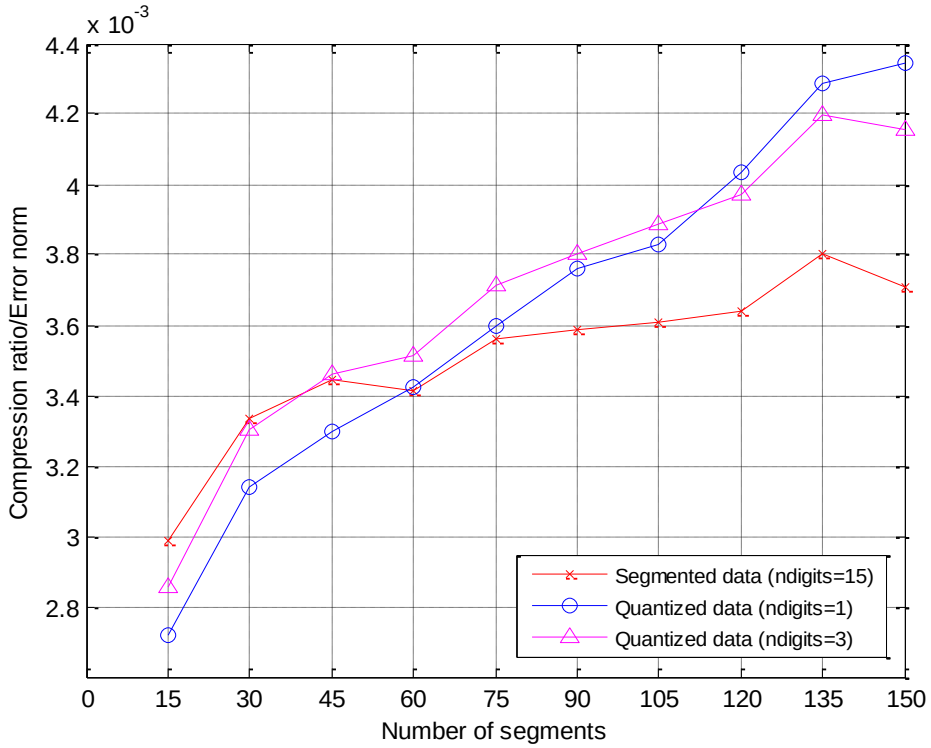

Figure 4.25. Compression Ratio/Error Norm versus Number of Segments using the 30th column of the TEP Data Set.

In this chapter, data compression is improved by using the hybrid method consisting of two irreversible techniques; the PAA and quantization. PAA is a fast and simple dimensionality reduction technique mostly used for the visualization of the original data which is favored for yielding high compression ratios due to the repeating patterns in segmented data. Furthermore, in process monitoring and fault detection/identification tasks, the major process events, upward/downward shifts and decreasing/increasing trends can easily be followed visually by the plant operators. However, high error norms are obtained in the PAA and there is also the possibility of discarding some important patterns in the original data set if large frame widths are used. In addition, it can be stated that PAA is not an appropriate method for noisy data sets especially the ones containing peak points. Compression ratios can further be increased by quantization due to the repeating zeros replaced with discarded digits without any significant increase in error norms. Quantization step becomes more effective as the number of frames used in PAA increases. The Discrete Cosine Transform (DCT) will be studied in the following chapter in order to reduce error norms considerably while maximizing reduction ratios.

5. DATA COMPRESSION VIA DISCRETE COSINE TRANSFORM

In transform methods, original data set is transformed into a different domain where it can be compressed better by using orthogonal basis functions such as sine, cosine and wavelets. Rao and Yip (2001) proposed that these basis functions should be independent to achieve high decorrelation, in other words, to reduce autocorrelation within a signal. The transform will be more efficient if most of the energy is packed in a few transform coefficients (Sayood, 2006).

Discrete Cosine Transform (DCT) shows strong energy compaction property meaning that the first few transform coefficients representing lower frequencies contain the most important information, whereas the rest of the coefficients representing higher frequencies contain the less important information which are not essential in reconstructing the original signal (Salomon, 2008). In other words, the amount of energy is mostly packed in the first few transform coefficients. Due to this property, high frequency coefficients which are close to zero can easily be discarded prior to encoding in order to achieve large compression without any significant loss of information.

DCT, similar to Discrete Fourier Transform (DFT), operates on a function at finite number of discrete data points using only cosine function as the basis function, whereas DFT uses both cosine and sine functions. DCT is used in many applications such as process monitoring, noise filtering, image processing, signal processing, etc. DCT is the easiest dimensionality reduction technique to achieve large compression. In DCT, each transform coefficient is encoded independently. DCT is a lossless transformation that does not actually perform compression, the amount of loss and compression is determined by the quantization step following transformation (Nelson and Gailly, 1996).

Large compression can be obtained by quantization due to the repeating zeros replaced with transform coefficients with small amplitudes without any significant distortion. Quantized transform coefficients are encoded and finally Inverse DCT is performed on the transform coefficients for decompression.

5.1. One Dimensional Discrete Cosine Transform

DCT performs reversible mapping from time to frequency domain. It is a fast, linear, invertible, separable and data independent transform utilizing Fast Fourier Transform conventions. One Dimensional Discrete Cosine Transform (1D-DCT) coefficients are calculated by using the following formula (MATLAB (version R2008b) Signal Processing Toolbox);

$$e(k) = r(k) \sum_{n=1}^{N} o(n) \cos \frac{\pi(2n-1)(k-1)}{2N} \quad \text{for } k = 1, \dots, N \tag{5.1}$$

where

$$r(k) = \begin{cases} \frac{1}{\sqrt{N}}, & k = 1 \\ \sqrt{\frac{2}{N}}, & 2 \leq k \leq N \end{cases} \tag{5.2}$$

where **o** denotes the set of data values with the length N and **e** is the set of N DCT transform coefficients.

Decompression is performed by the Inverse Discrete Cosine Transform (IDCT) on the transform coefficients. IDCT takes DCT coefficients and multiplies them with cosine functions and adds them to reconstruct the original data by using the following formula (MATLAB (version R2008b) Signal Processing Toolbox);

$$o(n) = \sum_{k=1}^{N} r(k) e(k) \cos \frac{\pi(2n-1)(k-1)}{2N} \quad \text{for } n = 1, \dots, N \tag{5.3}$$

where r(k) is defined in Equation 5.2.

DCT can operate on both one dimensional data $\mathbf{o}_{Nx1}$ and two dimensional data matrix $\mathbf{O}_{NXC}$ and can be represented in matrix form as;

$$\mathbf{e}_{\mathrm{Nx1}} = \mathbf{U}_{\mathrm{N\times N}} \times \mathbf{o}_{\mathrm{N\times 1}}$$

$$\text{where} \begin{cases} \mathbf{o}\text{: One dimensional data} \\ \mathbf{U}\text{: Orthogonal DCT square matrix} \\ \mathbf{e}\text{: DCT coefficient vector} \end{cases}$$

If DCT is applied to two dimensional data matrix, **O** then DCT transforms its columns.

$$\mathbf{E}_{\mathrm{NxC}} = \mathbf{U}_{\mathrm{N\times N}} \times \mathbf{O}_{\mathrm{N\times C}}$$

$$\text{where} \begin{cases} \mathbf{O}\text{: Two dimensional data matrix} \\ \mathbf{U}\text{: Orthogonal DCT square matrix} \\ \mathbf{E}\text{: DCT coefficient matrix} \end{cases}$$

5.2. Two Dimensional Discrete Cosine Transform

Two Dimensional Discrete Cosine Transform (2D-DCT) is computed by applying DCT in one dimension to each row of a data set, then to each column of the result, that is why it is also called "Blocked Transform" (Salomon, 2008). Following formula is used (MATLAB (version R2008b) Signal Processing Toolbox);

$$B_{pq} = d_p d_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N} \qquad (5.4)$$

$$\text{for} \begin{cases} 0 \le p \le M-1 \\ 0 \le q \le N-1 \end{cases}$$

where

$$d_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \le p \le M-1 \end{cases} \qquad (5.5)$$

$$d_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \le q \le N-1 \end{cases} \qquad (5.6)$$

where **A** denotes the data matrix with dimensions M by N and **B** is the 2D-DCT transform coefficients with dimensions M by N.

IDCT is similarly defined as (MATLAB (version R2008b) Signal Processing Toolbox);

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} d_p d_q B_{pq} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N} \quad (5.7)$$

$$\text{for } \begin{cases} 0 \le m \le M-1 \\ 0 \le n \le N-1 \end{cases}$$

where d_p and d_q are given in Equation 5.5 and Equation 5.6.

The topmost coefficient, called “DC Coefficient”, is the average value of the sample sequence having the most important information content than the high frequency components. All other transform coefficients are called as “AC Coefficients”. Magnitudes of these coefficients decrease as they move farther from the DC coefficient (Nelson and Gailly, 1996).

2D-DCT can be represented in matrix form, if **A** is a square matrix, as the following;

$$\mathbf{B}_{\mathrm{N\times N}} = \mathbf{I}^{\mathrm{T}}_{\mathrm{N\times N}} \mathbf{A}_{\mathrm{N\times N}} \mathbf{I}_{\mathrm{N\times N}}$$

$$\text{where } \begin{cases} \mathbf{A}\text{: Two dimensional square data matrix} \\ \mathbf{I}\text{: Orthogonal DCT square matrix} \\ \mathbf{B}\text{: DCT coefficient matrix} \end{cases}$$

5.3. Applications of One Dimensional and Two Dimensional Discrete Cosine Transforms

In this section, data compression and lossy reconstruction will be investigated by using the 1D-DCT and 2D-DCT compression techniques, the thresholding method as a lossy compression step and ZIP as the lossless encoding algorithm using the data sets PortSimHigh, PortSimLow, SELDI-TOF MS and TEP mentioned in Chapter 3 for 10 different percentile values of the frequency distribution of the transform coefficients in the

[15%-99.8%] range. Furthermore, for the percentile value 99.5%, detailed 1D-DCT and 2D-DCT analyses are given for each of the data sets.

The thresholding method is applied instead of zero padding in filtering since the data sets have high-frequency content. The effect of the percentile values used in thresholding step on compression is measured in terms of compression ratio, mean error norm, % relative global error and % relative maximum error. In addition, ratios of compression ratio to mean error norm are computed to determine the optimum percentile level. Shannon entropies of the thresholded transform coefficients cannot be computed since coefficients between threshold limits become zero after filtering and the logarithm of zero is undefined.

The computations were done in MATLAB (version R2008b) using MATLAB's `dct`, `dct2`, `idct` and `idct2` commands for calculating 1D-DCT and 2D-DCT coefficients and reconstructed signals, and MATLAB's internal `zip` command is used as the lossless compression algorithm.

Besides mean error norm, % relative global error and % relative maximum error are calculated to determine the effectiveness of 1D-DCT and 2D-DCT giving the overall and localized measure of error respectively (Watson *et al.*, 1998). They are defined as;

$$\%\ \text{Relative global error} = 100 \times \frac{\sum_i (f_i - f'_i)^2}{\sum_i (f_i)^2} \tag{5.8}$$

$$\%\ \text{Relative maximum error} = 100 \times \frac{max_i(|f_i - f'_i|)}{max_i(|f_i|)} \tag{5.9}$$

where f_i implies the i^{th} element of the original signal and f'_i is the i^{th} element of the reconstructed signal (Watson *et al.*, 1998).

5.3.1. One Dimensional Discrete Cosine Transform of the PortSimHigh Data Set

In this section, 1D-DCT is applied to the overall PortSimHigh data set (consisting of 500 stock prices) and for illustration purposes, only the first 16 stock prices of the

PortSimHigh data set are presented. All of the 16 columns of the data set show similar features as shown in Figure 5.1.

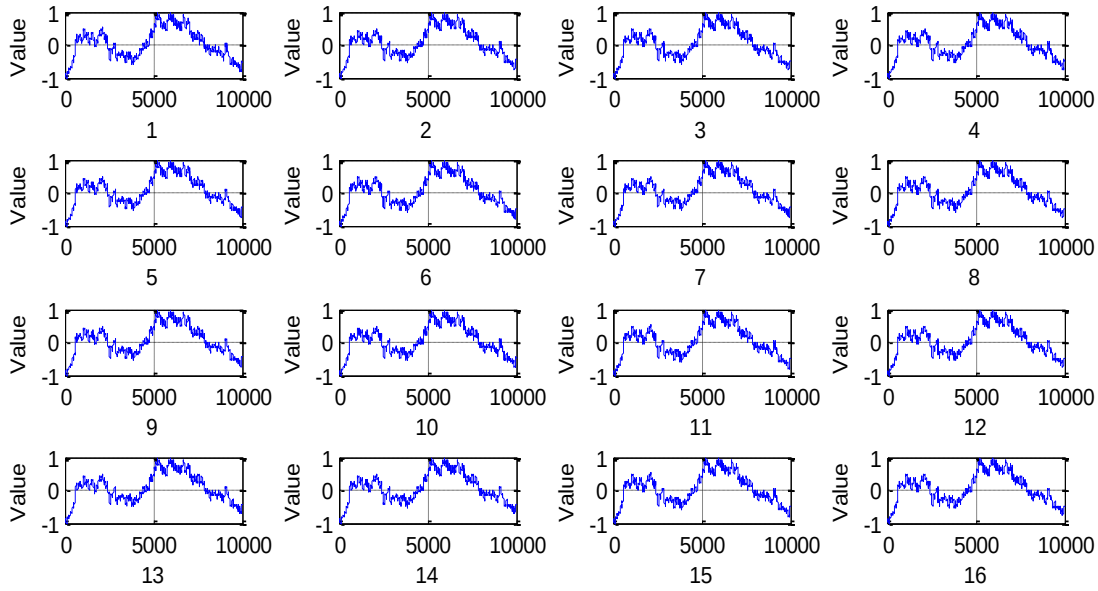

Figure 5.1. First 16 Stock Prices of the PortSimHigh Data Set in Scaled Format.

Transform coefficients of the first 16 stock prices of the PortSimHigh data set obtained with 1D-DCT before and after thresholding, in which the percentile value is taken as 99.5%, are shown in Figure 5.2 and Figure 5.3. Almost all of the coefficients after about first 100 die out exponentially towards zero. Thus, zero padding method could be applied for this case. Nevertheless, thresholding method is used to minimize distortion. After thresholding, most of the coefficients with small amplitudes become to zero. Actually, these two figures seem alike as most of the transform coefficients are very close to zero already before thresholding.

Semilog-log (upper sub-window of Figure 5.4) and log-log (bottom sub-window of Figure 5.4) plots of the sorted absolute values of the 1D-DCT coefficients of the overall PortSimHigh data set which are padded into a vector of size 5000000 and the threshold limit of 0.7969 denoted by the red horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 5.4. Transform coefficients above and below the threshold limit can be identified clearly in the log-log plot. The coefficients below the threshold limit (4975000 coefficients) are set to zero for compression, in other words 99.5% of the transform coefficients become zero. The

number of nonzero coefficients kept is 25000, which is only 0.5% of the number of data points.

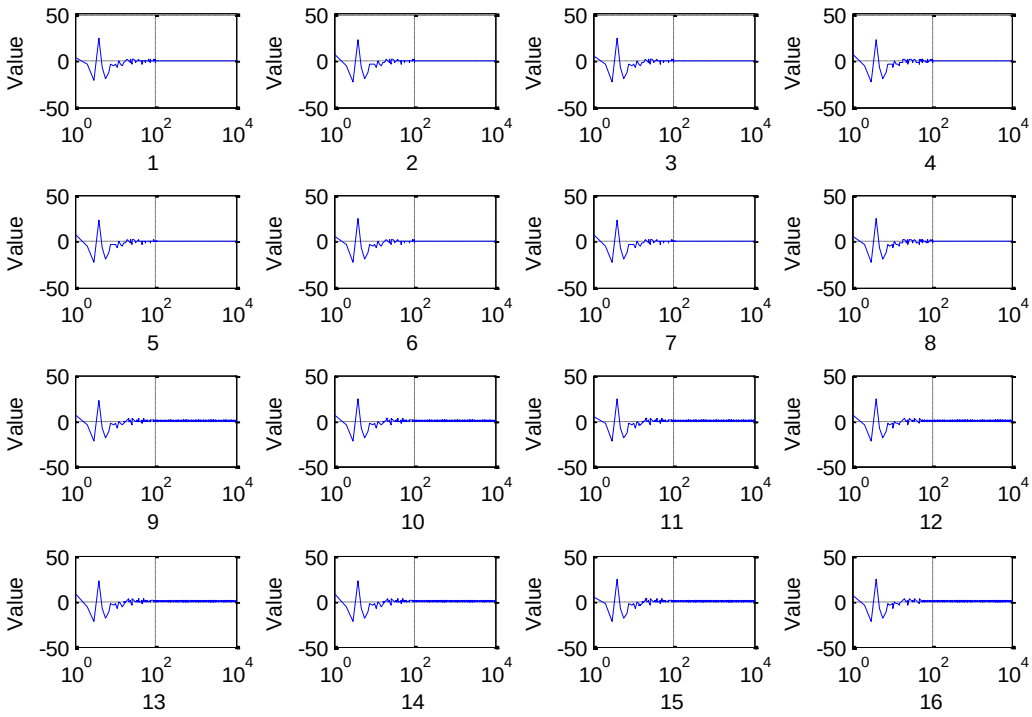

Figure 5.2. 1D-DCT coefficients of the PortSimHigh Data Set.

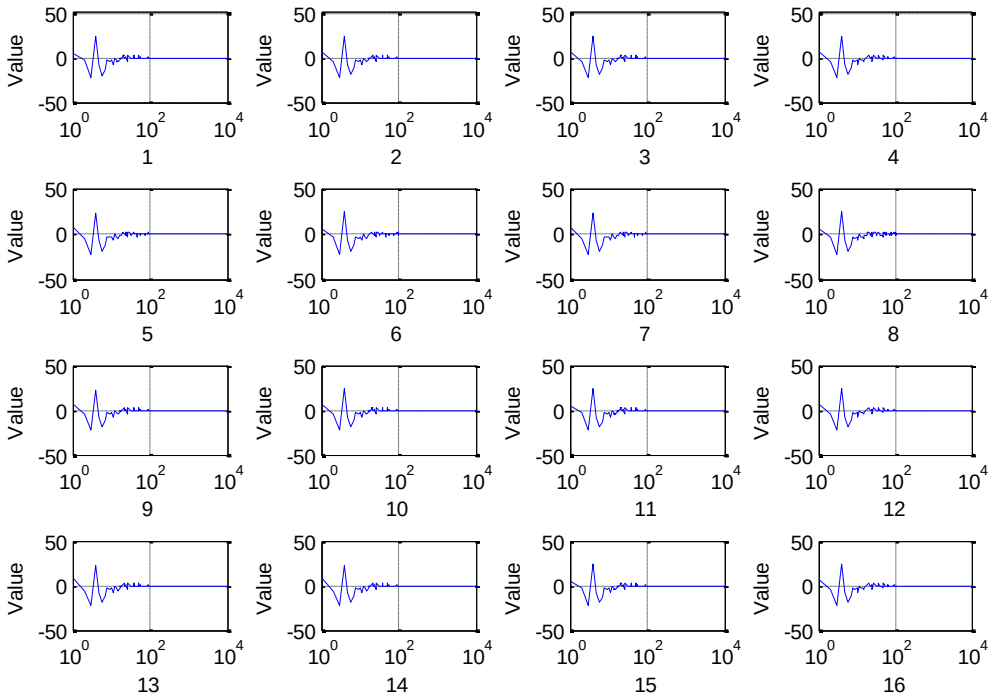

Figure 5.3. 1D-DCT coefficients of the PortSimHigh Data Set after Thresholding with α=99.5 %.

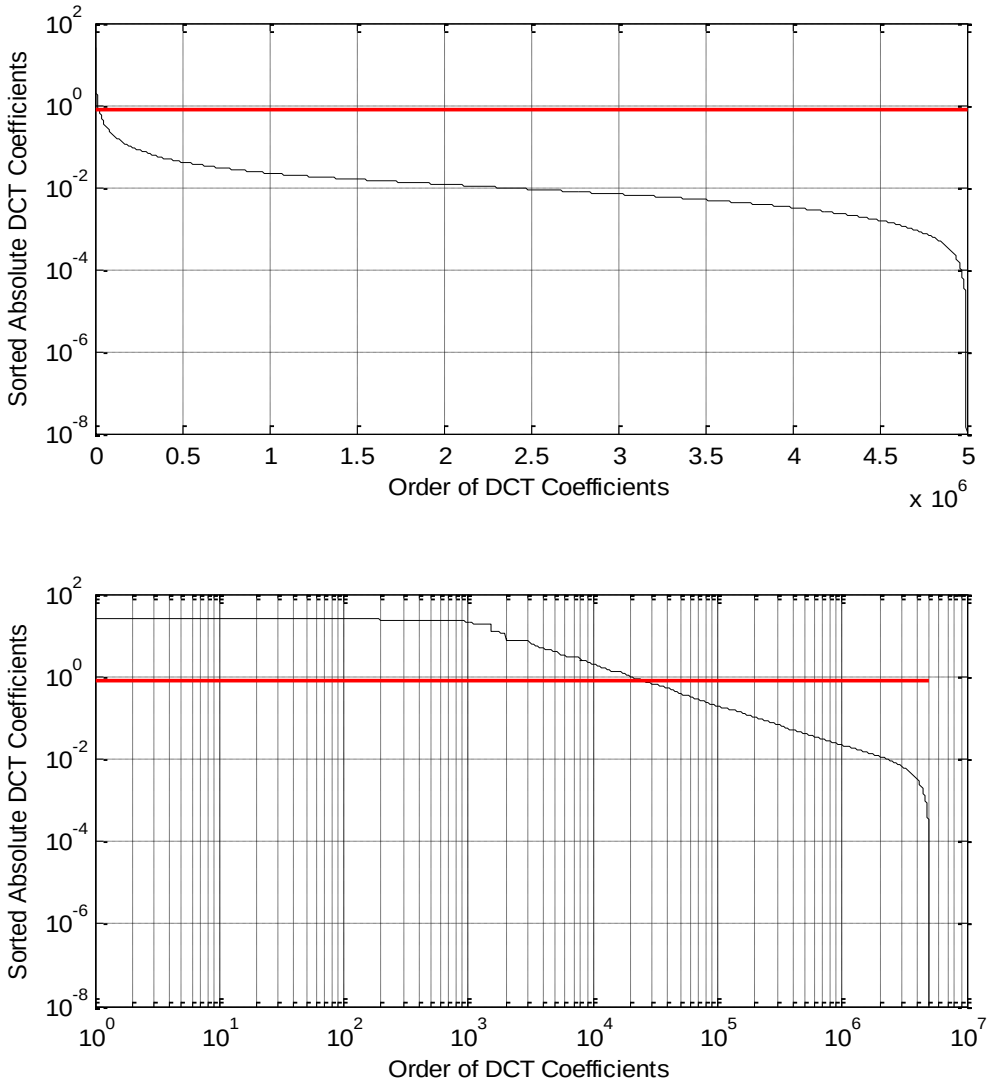

Figure 5.4. Semilog-log and Log-log Plots of Sorted Absolute 1D-DCT Coefficients of the Overall PortSimHigh Data Set for α=99.5 %.

Figure 5.5 is given to compare the sizes of the ZIP files of the original and encoded data sets. It is seen that the ZIP file of the overall PortSimHigh data set is nearly 51.1 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.93 MB. Thus, compression can be increased 55 times by applying 1D-DCT technique and taking the percentile value as 99.5% in thresholding step.

Reconstructed data are generated with 1D-iDCT (inverse DCT). The reconstructed data and their overlay with the originals are shown in Figure 5.6 and Figure 5.7 respectively. The major features of the stock prices including peaks and decreasing/increasing trends are reconstructed thoroughly as illustrated in Figure 5.6. In

addition, the amplitudes of the reconstructed data set (shown in red) are almost same as those of the original stock prices as shown in Figure 5.7 concluding that there is not any significant loss in information content.

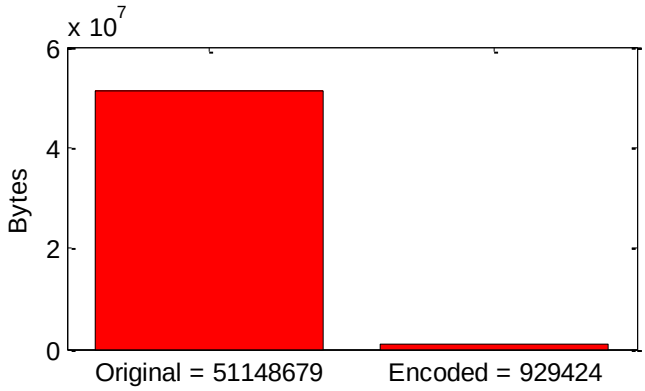

Figure 5.5. ZIP Compression Comparison of the Original and Encoded Overall PortSimHigh Data Set for α=99.5 % with 1D-DCT.

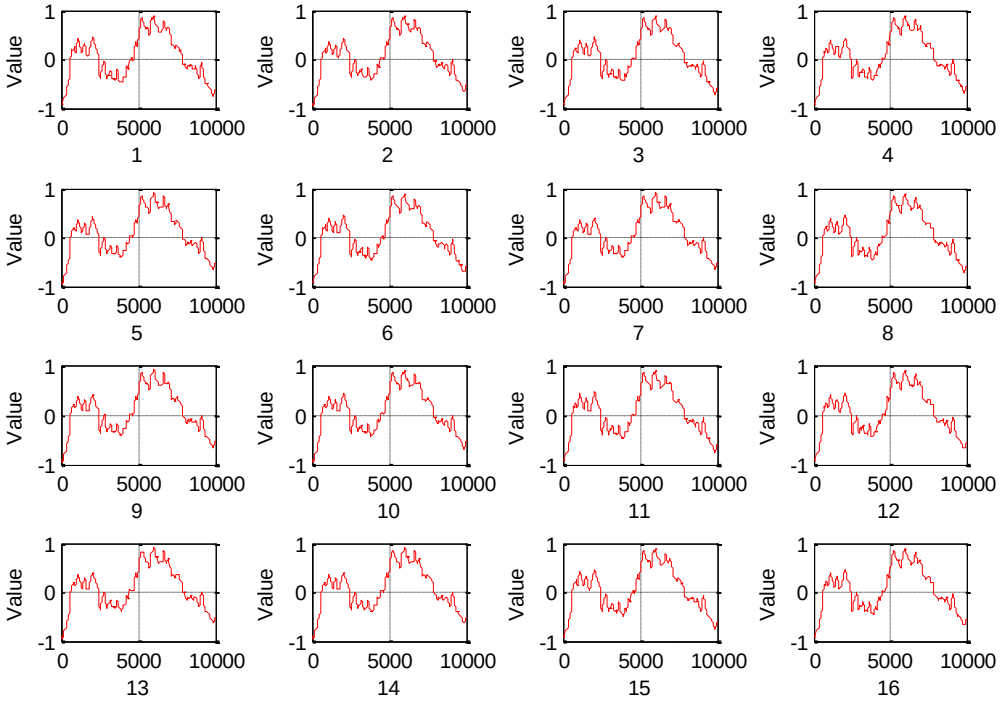

Figure 5.6. Reconstructed PortSimHigh Data Set with Inverse 1D-DCT.

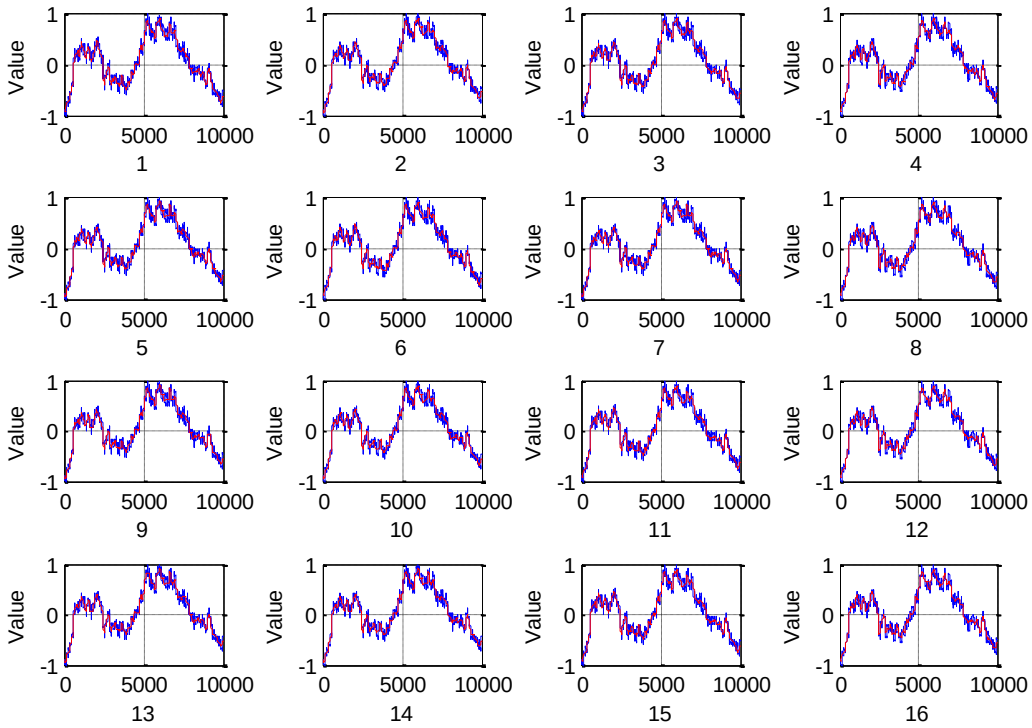

Figure 5.7. Original and Reconstructed Signals of the PortSimHigh Data Set with Inverse 1D-DCT.

In Figure 5.8, the original and reconstructed signals are plotted around the y=x line. Reconstructed data points around the y=x line indicates that reconstructed signals are very close to the original signals (all of the 16 measurements) whereas, reconstructed data points around the y=0 line shows that they are completely different from the original signals.

Error norms between original and reconstructed data sets are calculated per data column. Minimum error norm, that is 446.45, is obtained in the 267th column and the maximum error norm, that is 476.62, is obtained in the 62nd column of the PortSimHigh data set. As it can be seen from both the upper and the middle sub-windows of Figure 5.9, all of the reconstructed data points cluster around the y=x line meaning that reconstructed signals are very close to the original signals providing small error norms. Furthermore, reconstruction error norm values of each of the 500 columns of the PortSimHigh data set are given in the bottom sub-window of Figure 5.9.

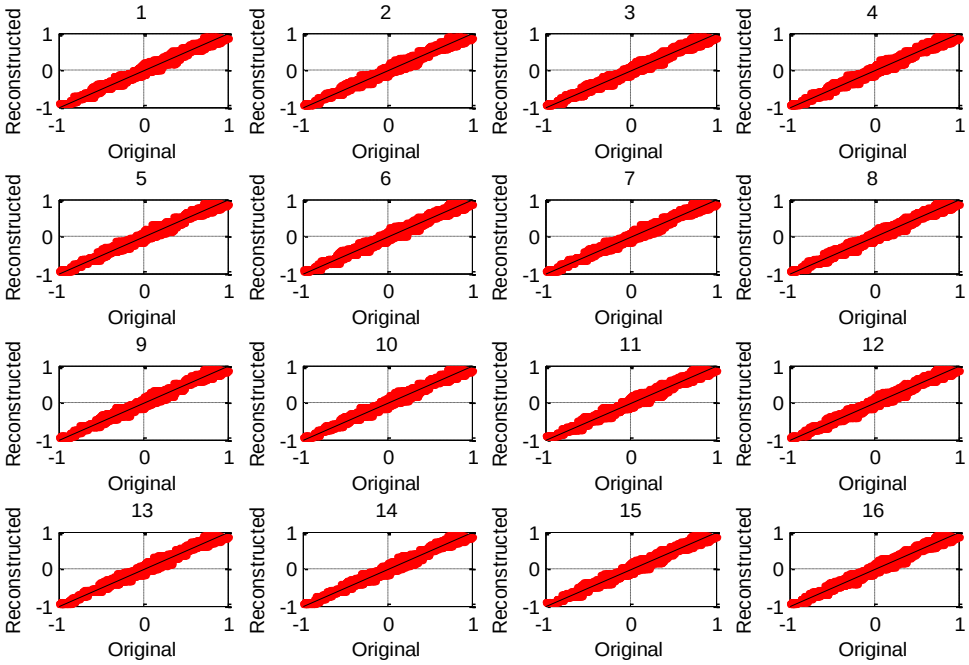

Figure 5.8. Reconstructed versus Original Signals of the PortSimHigh Data Set with Inverse 1D-DCT.

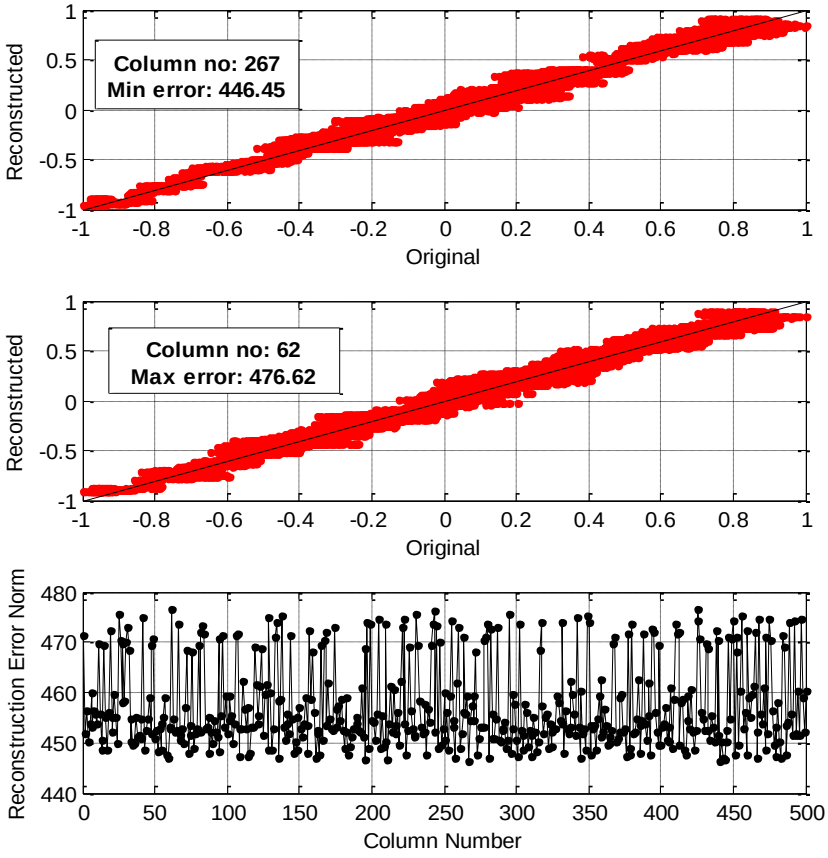

Figure 5.9. Reconstruction Error Norm Values of the PortSimHigh Data Set with Inverse 1D-DCT.

5.3.2. Two Dimensional Discrete Cosine Transform of the PortSimHigh Data Set

In this section, 2D-DCT is applied to the overall PortSimHigh data set (consisting of 500 stock prices), however, for illustration purposes, only the related figures of the first 16 stock prices of the PortSimHigh data set are presented. The first 16 stock prices were given before in the previous section in scaled format with Figure 5.1.

Transform coefficients of the stock prices of the PortSimHigh data set obtained with 2D-DCT before and after thresholding, in which the percentile value is taken as 99.5%, are shown in Figure 5.10 and Figure 5.11. It is seen that the first few transform coefficients (with large magnitudes) in the first column of the coefficients matrix store the most important information content than the high frequency components as mentioned in Section 5.2. Due to the energy compaction property, high-frequency coefficients which are close to zero (after about first 100 ones in all columns) can become zero (columns six, 12 and 15 in Figure 5.11) in the thresholding step. Such thresholding achieves high compression without any significant loss of information after reconstruction. It is also observed that most of the 2D-DCT coefficients are smaller than 1D-DCT coefficients indicating that 2D-DCT should be the preferred technique for the highly correlated PortSimHigh data set. Thus, larger compression ratios are expected with minimum amount of distortion due to these smaller coefficients as they can easily be discarded in the quantization step.

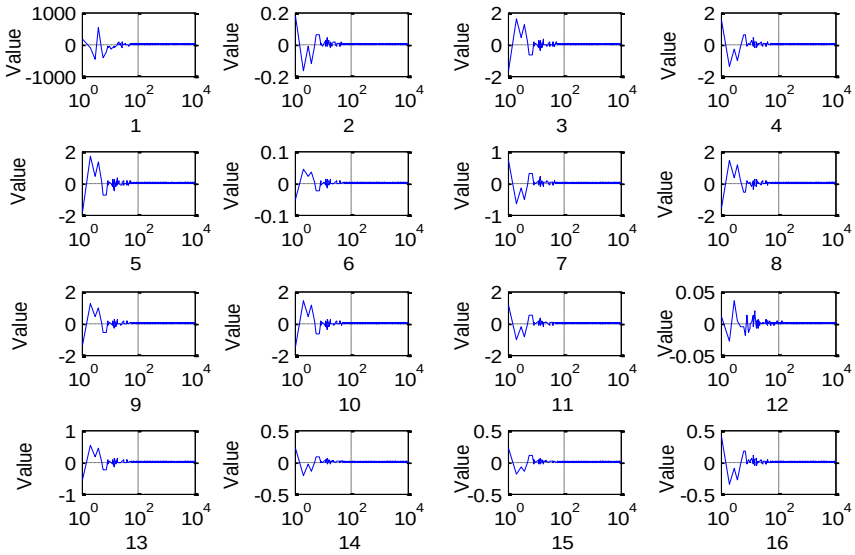

Figure 5.10. 2D-DCT Coefficients of the PortSimHigh Data Set.

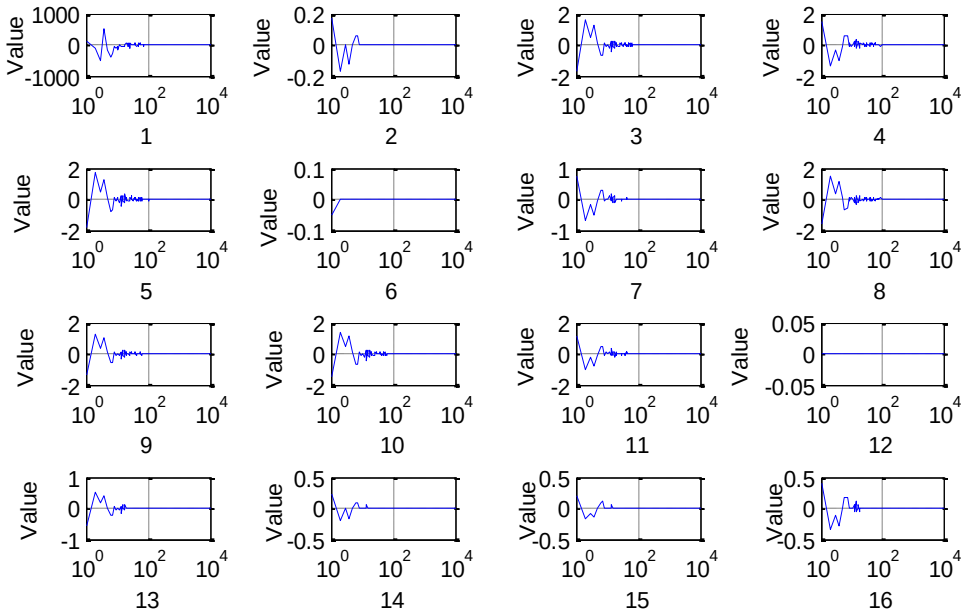

Figure 5.11. 2D-DCT Coefficients of the PortSimHigh Data Set after Thresholding with α=99.5 %.

Semilog-log (upper sub-window of Figure 5.12) and log-log (bottom sub-window of Figure 5.12) plots of the sorted absolute values of the 2D-DCT coefficients of the overall PortSimHigh data set which are padded into a vector of size 5000000, and the threshold limit of 0.042, denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 5.12. Transform coefficients above and below the threshold limit can be identified clearly in the log-log plot. The coefficients below the threshold limit (4975000 coefficients) are set to zero for compression, in other words 99.5% of the transform coefficients become zero. The number of nonzero coefficients kept is 25000, which is only 0.5% of the number of data points. In addition, the threshold limit is much smaller than that found with 1D-DCT, that is 0.7969, since smaller 2D-DCT coefficients are produced for the highly correlated data set.

Sizes of the ZIP files of the original and encoded data sets are compared in Figure 5.13. The ZIP file of the overall PortSimHigh data set is nearly 49.8 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.89 MB. Thus, compression can be increased 55.7 times by applying 2D-DCT technique and taking the percentile value as 99.5% in thresholding step. It can also be stated that the compression ratio obtained with

2D-DCT is slightly larger than that obtained with 1D-DCT, which was 55 as mentioned in Section 5.3.1.

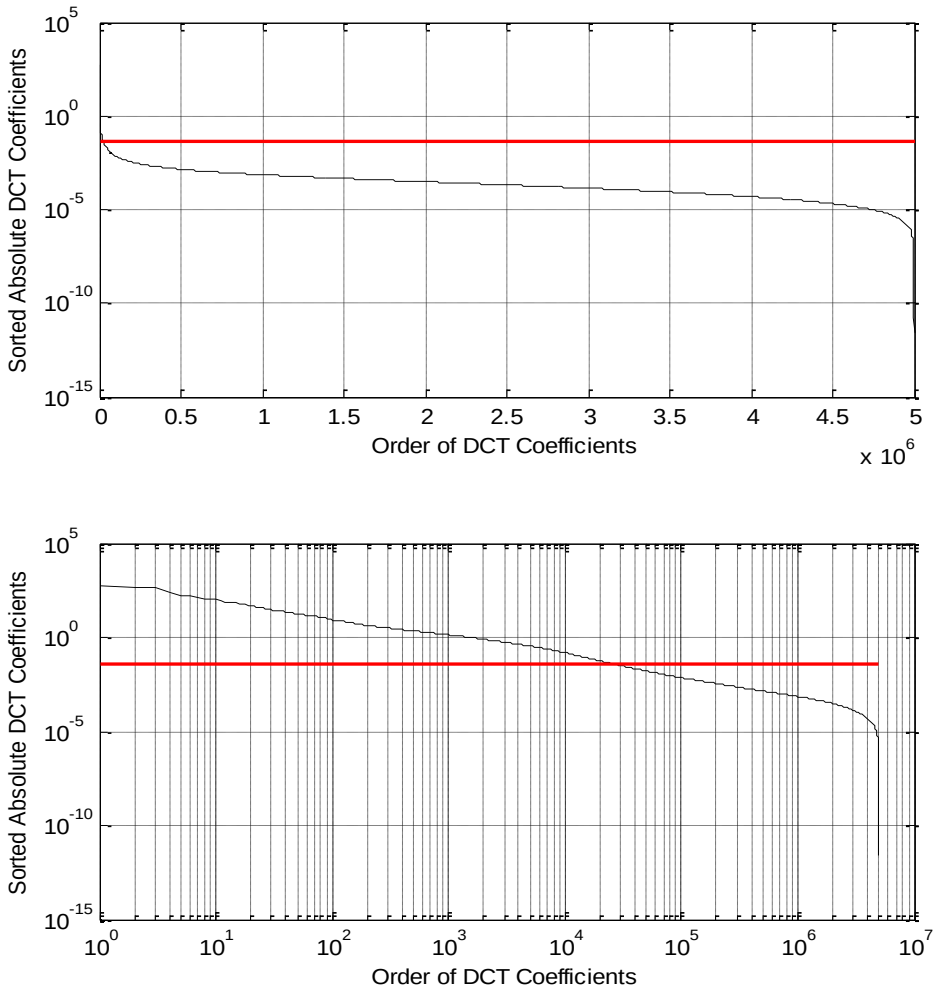

Figure 5.12. Semilog-log and Log-log Plots of Sorted Absolute 2D-DCT Coefficients of the Overall PortSimHigh Data Set for α=99.5 %.

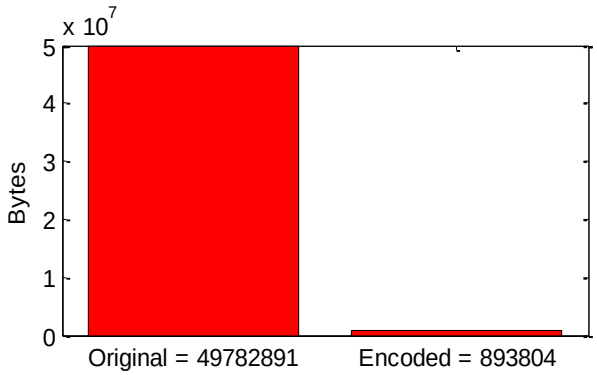

Figure 5.13. ZIP Compression Comparison of the Original and Encoded Overall PortSimHigh Data Set for α=99.5 % with 2D-DCT.

Reconstructed data are generated with 2D-iDCT (inverse DCT). The reconstructed data and their overlay with the originals are shown in Figure 5.14 and Figure 5.15 respectively. It is seen that original and reconstructed data are similar as illustrated in Figure 5.15 concluding that there is almost no distortion in the reconstructed data set (shown in red). It can also be stated that better reconstruction is obtained with 2D-DCT rather than 1D-DCT as the former technique is more appropriate for the highly correlated PortSimHigh data set.

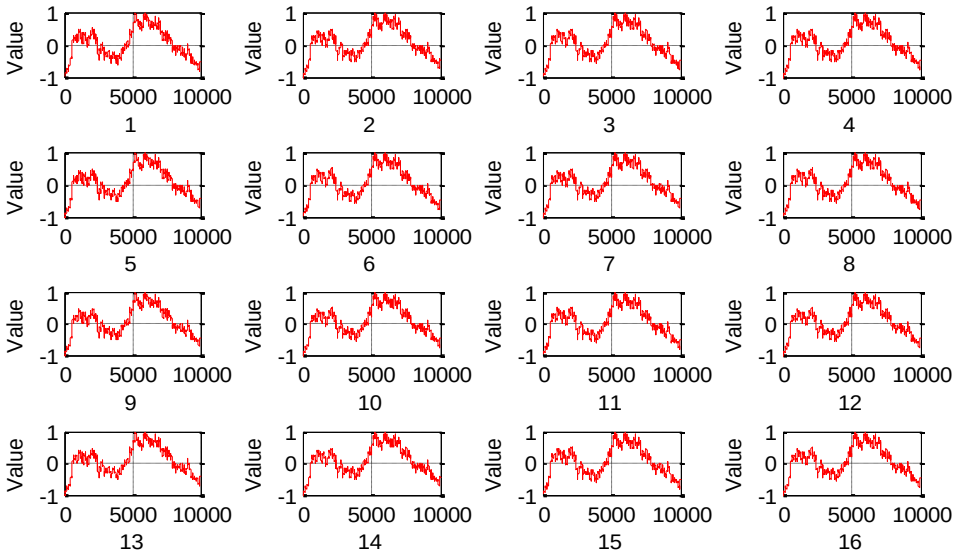

Figure 5.14. Reconstructed PortSimHigh Data Set with Inverse 2D-DCT.

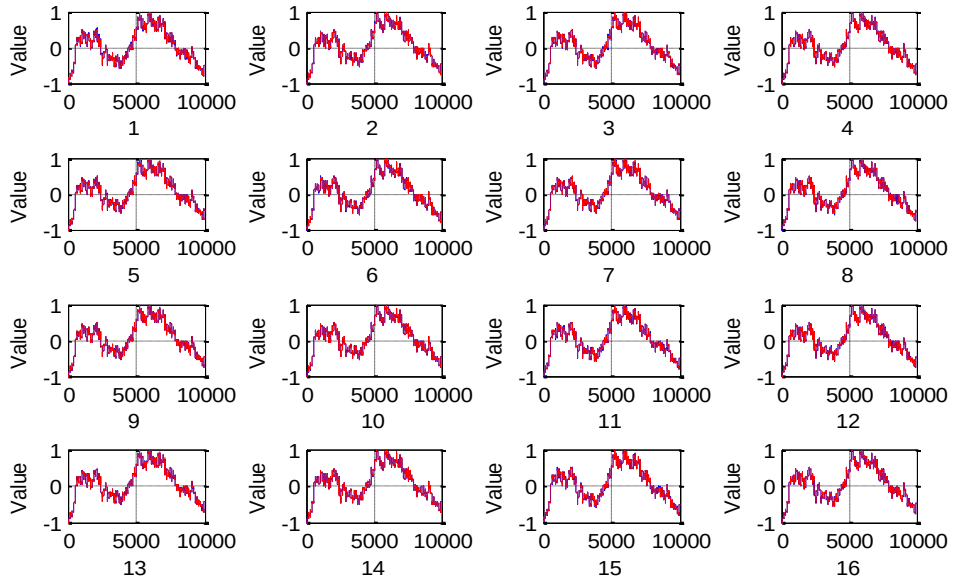

Figure 5.15. Original and Reconstructed Signals of the PortSimHigh Data Set with Inverse 2D-DCT.

The original and reconstructed signals generated with 2D-iDCT are plotted around the y=x line as shown in Figure 5.16. It is seen that all of the reconstructed data points are located around the y=x line showing that reconstructed signals are very close to the original signals (observed in all of the representative 16 stock prices).

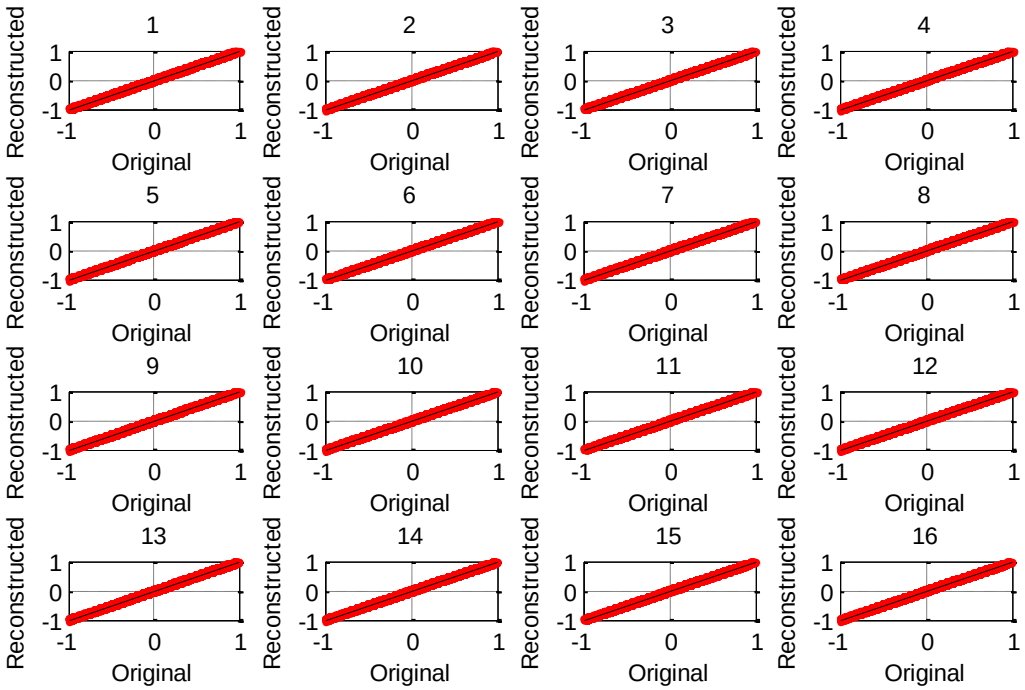

Figure 5.16. Reconstructed versus Original Signals of the PortSimHigh Data Set with Inverse 2D-DCT.

Reconstruction error norm values of the overall PortSimHigh data set calculated per data column are given in the bottom sub-window of Figure 5.17. Minimum error norm of 5.688, is obtained in the 474th column and the maximum error norm of 48.96, is obtained in the 426th column of the PortSimHigh data set after applying 2D-DCT. As it can be seen from both the upper and the middle sub-windows of Figure 5.17, original signals are decoded perfectly as reconstructed data points are located around the y=x line, representing small reconstruction error norms. It should also be mentioned that smaller reconstruction error norms (nearly 12-fold smaller than those produced with 1D-DCT) and slightly higher compression ratios are generated with 2D-DCT, concluding that this technique is clearly superior to 1D-DCT when they are applied to the highly correlated PortSimHigh data set.

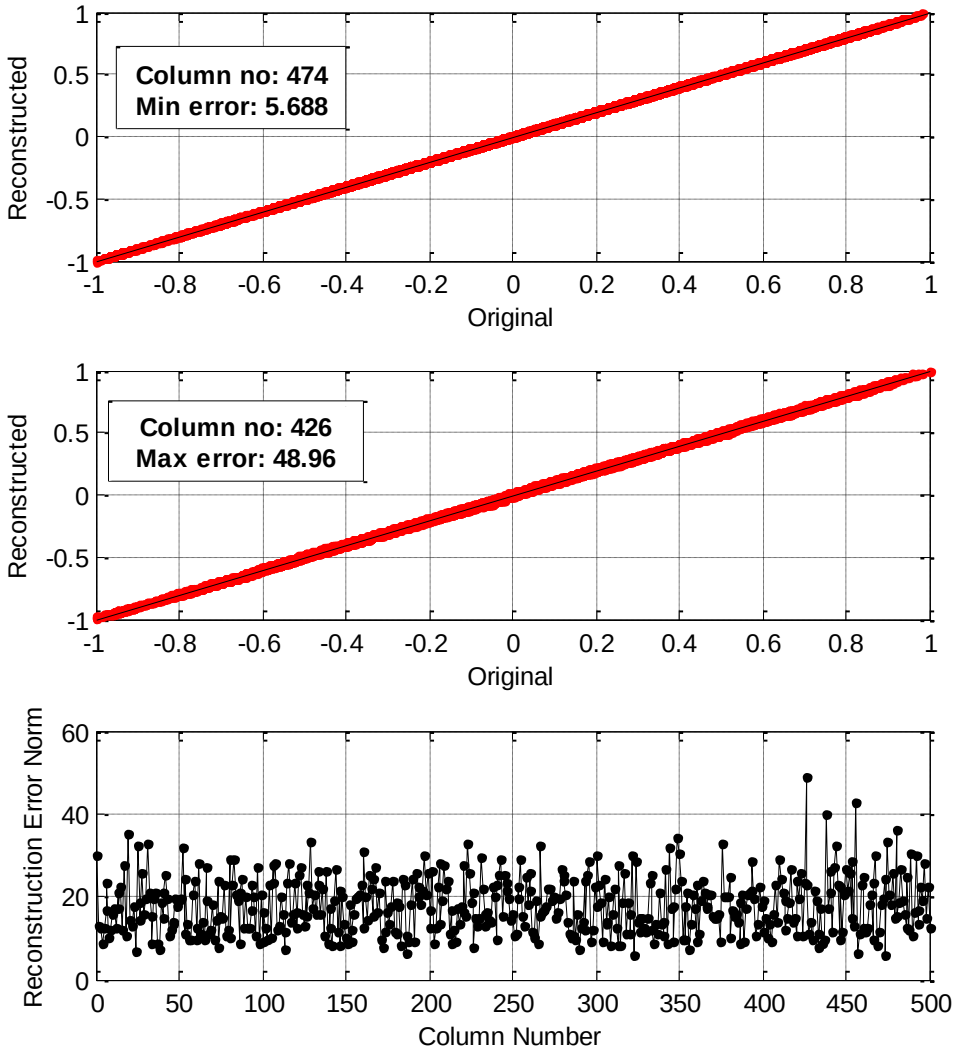

Figure 5.17. Reconstruction Error Norm Values of the PortSimHigh Data Set with Inverse 2D-DCT.

5.3.3. One Dimensional Discrete Cosine Transform of the PortSimLow Data Set

In this section, 1D-DCT is applied to the overall PortSimLow data set (consisting of 500 stock prices), however, for illustration purposes, only the first 16 stock prices of the PortSimLow data set are presented. Each of the 16 stock prices exhibits different features such as decreasing (prices of the first and 12th stocks) and increasing (prices of the fourth, sixth and 15th stocks) trends as shown in Figure 5.18.

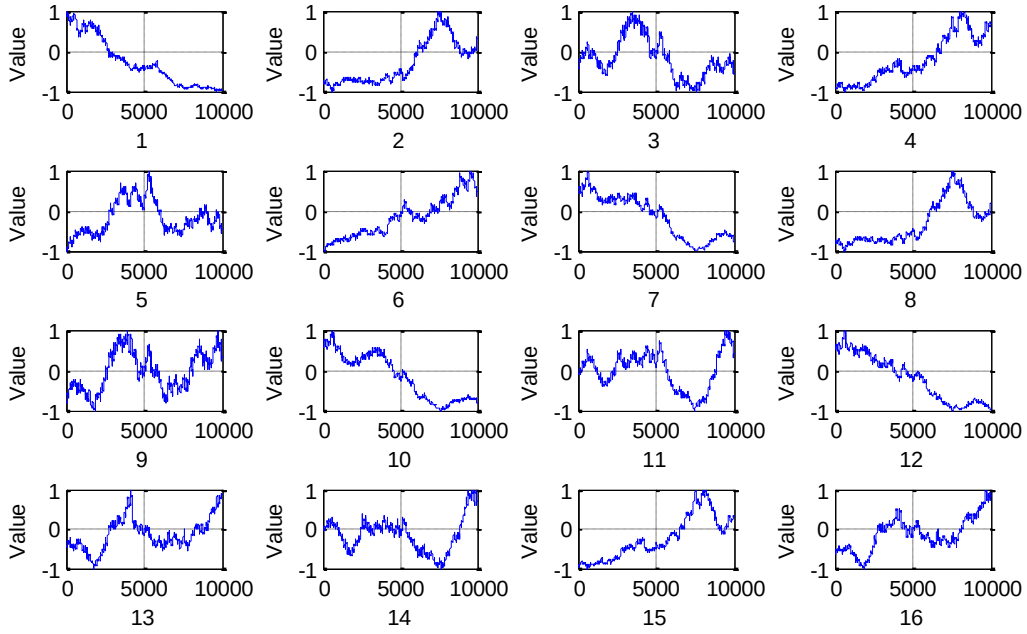

Figure 5.18. First 16 Stock Prices of the PortSimLow Data Set in Scaled Format.

Transform coefficients of the first representative 16 stock prices of the PortSimLow data set obtained with 1D-DCT before and after thresholding, in which the percentile value is taken as 99.5%, are shown in Figure 5.19 and Figure 5.20. Most of the transform coefficients after about first 100 are very small already and after thresholding, they become zero. Hence, there is not much difference between Figure 5.19 and Figure 5.20.

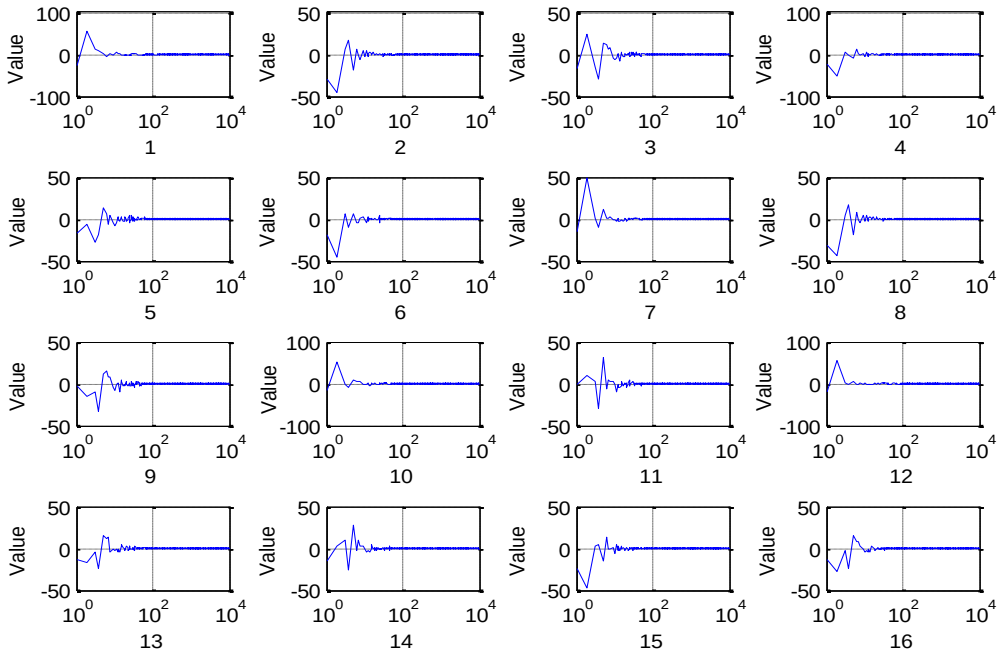

Figure 5.19. 1D-DCT Coefficients of the PortSimLow Data Set.

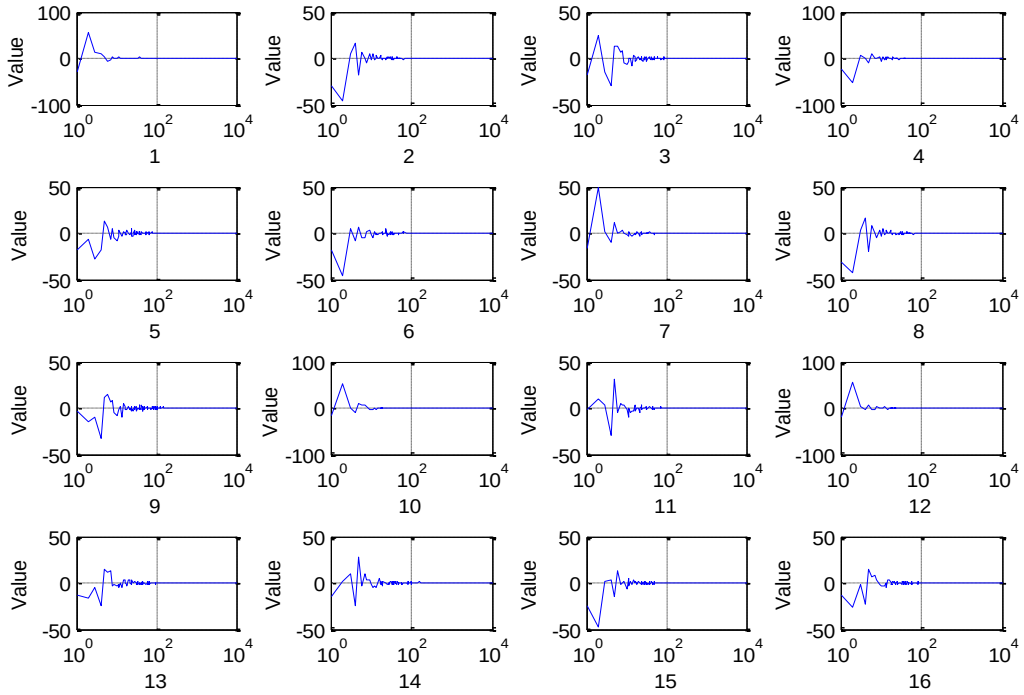

Figure 5.20. 1D-DCT Coefficients of the PortSimLow Data Set after Thresholding with α=99.5 %.

Semilog-log (upper sub-window of Figure 5.21) and log-log (bottom sub-window of Figure 5.21) plots of the sorted absolute values of the 1D-DCT coefficients of the overall PortSimLow data set, which are padded into a vector of size 5000000, and the threshold limit of 0.5722 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 5.21. Transform coefficients above and below the threshold limit can be identified clearly in the log-log plot. Transform coefficients below the threshold limit (99.5% of the transform coefficients) are set to zero and the number of nonzero coefficients kept is 25000, which is only 0.5% of the number of data points.

Sizes of the ZIP files of the original and encoded data sets are compared in Figure 5.22. It is seen that the ZIP file of the overall PortSimLow data set is nearly 51.1 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.93 MB improving compression 54.8 times with 1D-DCT technique and taking the percentile value as 99.5% in thresholding step.

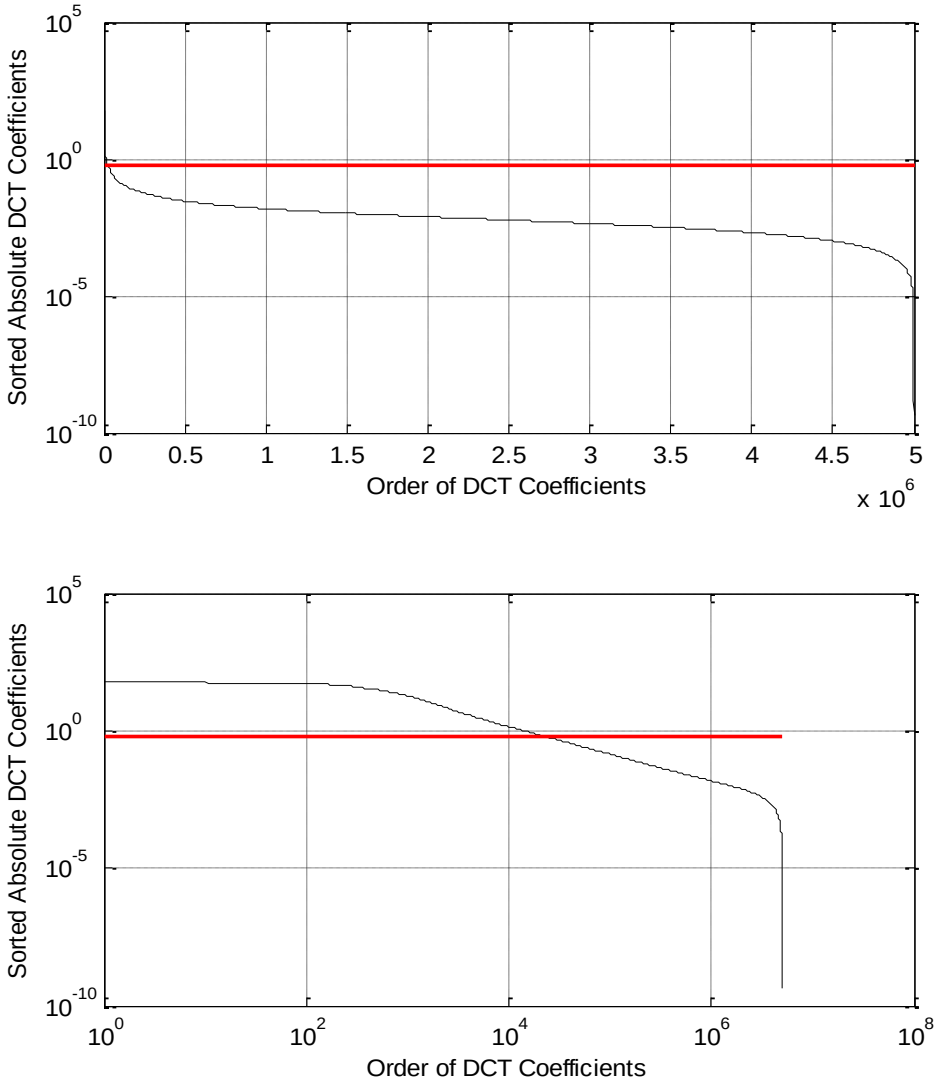

Figure 5.21. Semilog-log and Log-log Plot of Sorted Absolute 1D-DCT Coefficients of the Overall PortSimLow Data Set for α=99.5 %.

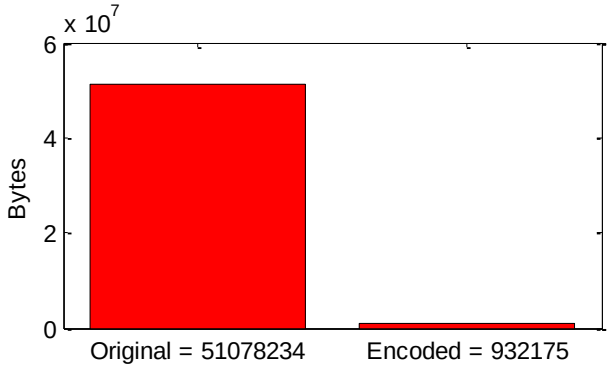

Figure 5.22. ZIP Compression Comparison of the Original and Encoded Overall PortSimLow Data Set for α=99.5 % with 1D-DCT.

Reconstructed data generated with 1D-iDCT (inverse DCT) and their overlay with the originals are shown in Figure 5.23 and Figure 5.24 respectively. It is seen that the major features of the stock prices including peaks and decreasing/increasing trends in the original data set are reconstructed thoroughly as shown in Figure 5.24 concluding that there is not any significant distortion in the decoded data set.

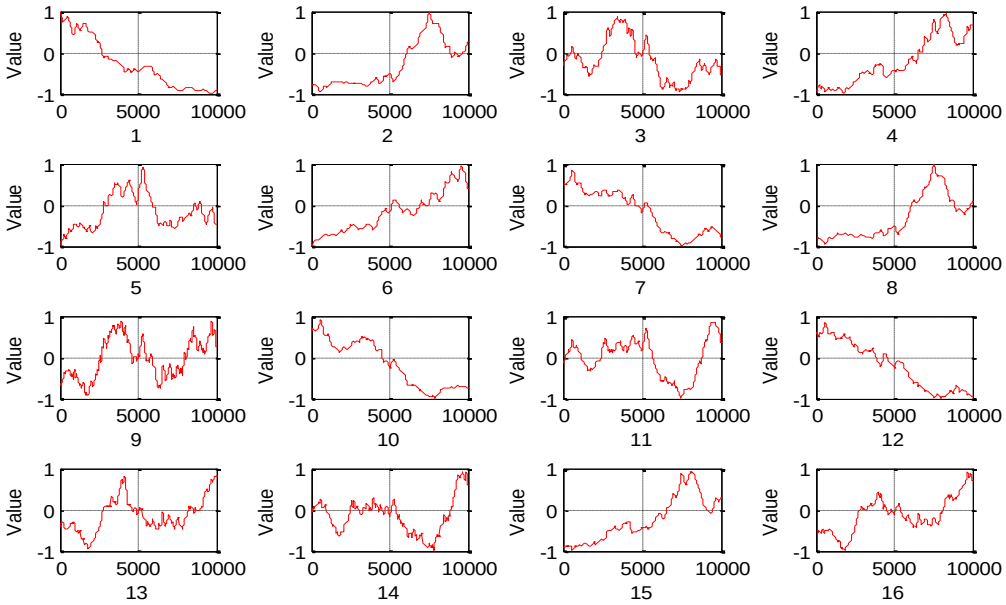

Figure 5.23. Reconstructed PortSimLow Data Set with Inverse 1D-DCT.

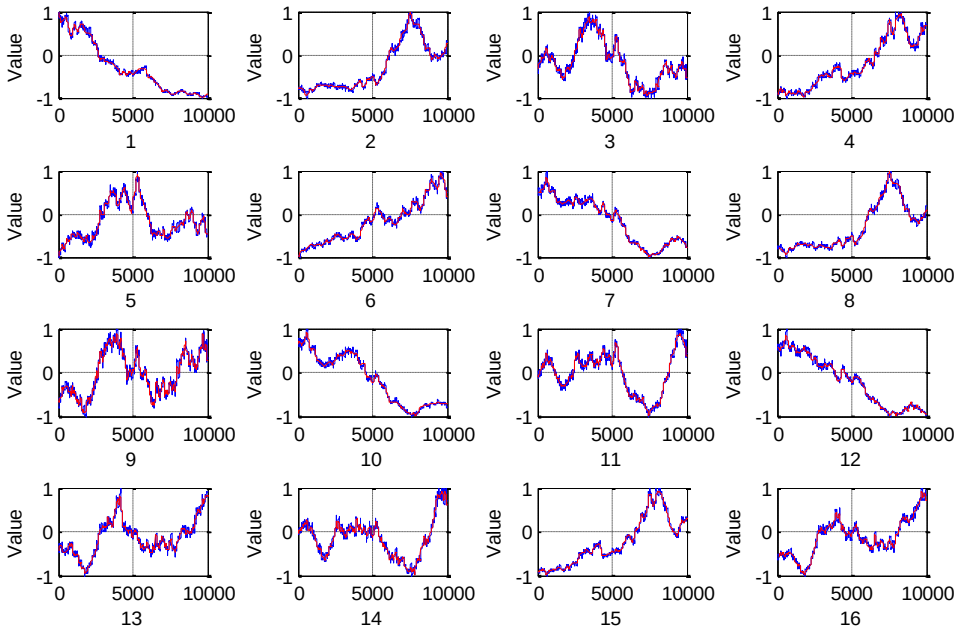

Figure 5.24. Original versus Reconstructed Signals of the PortSimLow Data Set with Inverse 1D-DCT.

The original and reconstructed signals generated with 1D-iDCT are plotted around the y=x line in Figure 5.25. Reconstructed data points around the y=x line indicate that original signals are decoded perfectly (prices of the first and 10th stocks), whereas there are some small distortions in reconstruction of the prices of the third and ninth stocks.

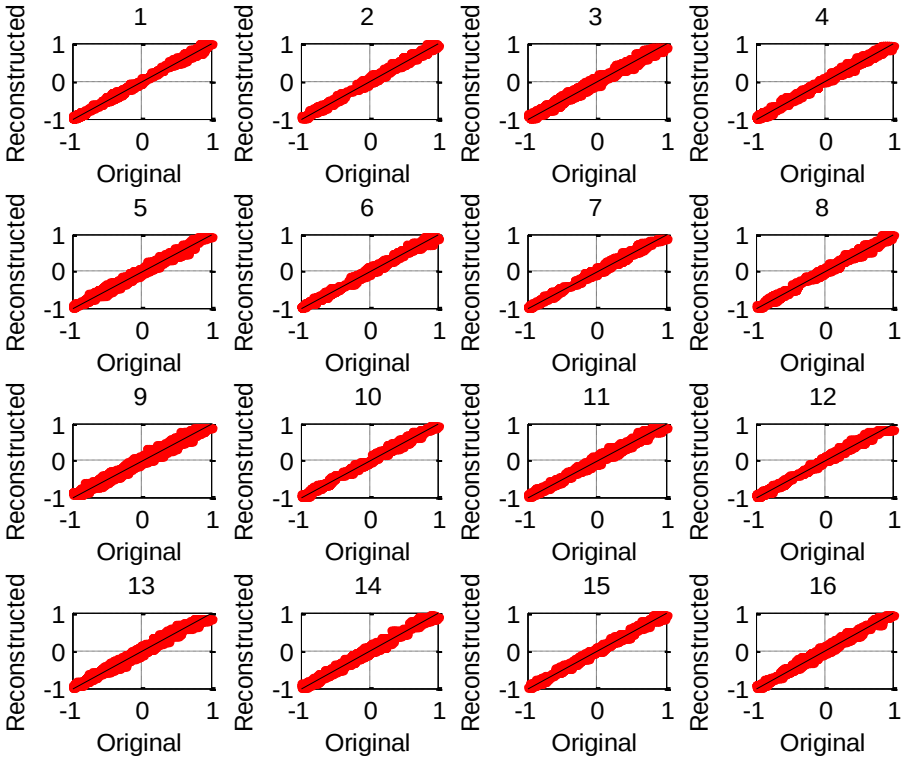

Figure 5.25. Reconstructed versus Original Signals of the PortSimLow Data Set with Inverse 1D-DCT.

Reconstruction error norm values of the overall PortSimLow data set calculated per data column are given in the bottom sub-window of Figure 5.26. Minimum error norm of 204.19 is obtained in the 115th column and the maximum error norm of 492.96 is obtained in the 122nd column of the PortSimLow data set. As it can be seen from the upper and the middle sub-windows of Figure 5.26, 115th column is reconstructed better than the 122nd column as most of the reconstructed data points of the 115th column are around the y=x line.

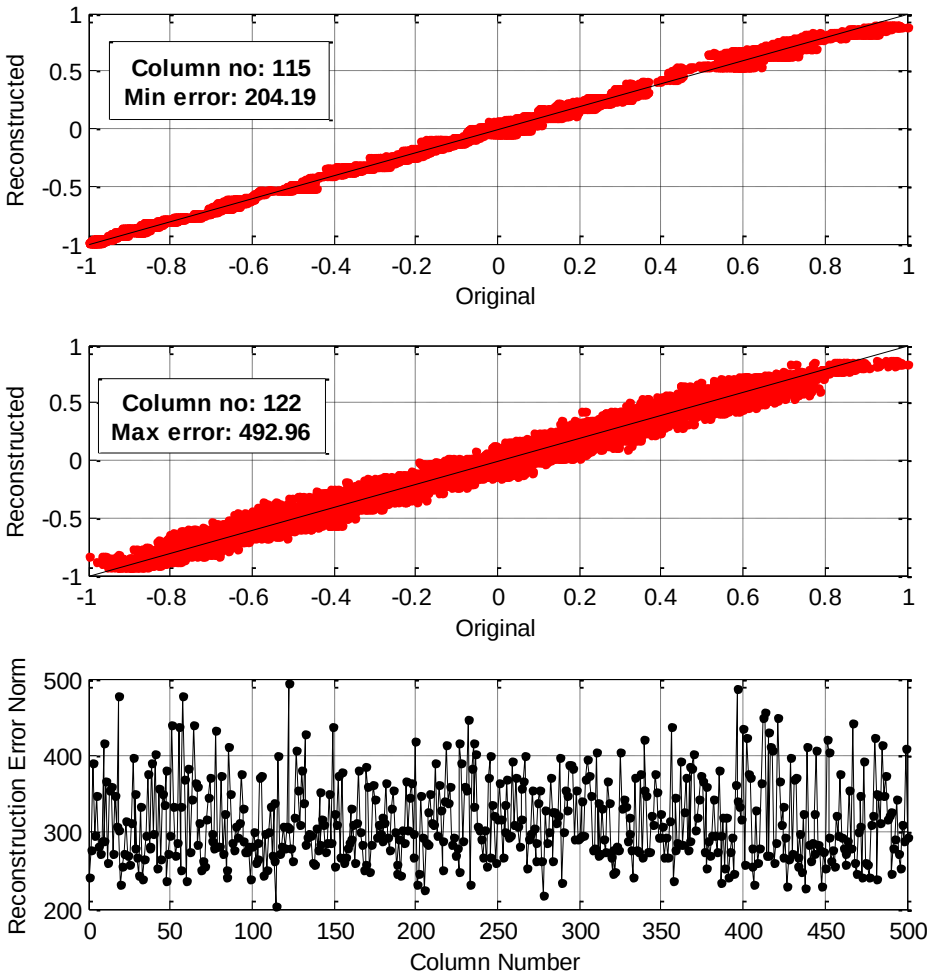

Figure 5.26. Reconstruction Error Norm Values of the PortSimLow Data Set with Inverse 1D-DCT.

5.3.4. Two Dimensional Discrete Cosine Transform of the PortSimLow Data Set

In this section, 2D-DCT is applied to the overall PortSimLow data set (consisting of 500 stock prices), however, for illustration purposes, only the first 16 stock prices of the PortSimLow data set are presented. The first 16 stock prices were given before in scaled format in Figure 5.18.

Transform coefficients of the stock prices of the PortSimLow data set produced with 2D-DCT before and after thresholding, in which the percentile value is taken as 99.5%, are shown in Figure 5.27 and Figure 5.28. As it can be seen from Figure 5.27, most of the transform coefficients after about the first 100 are very small. Thus, large majority of them become zero in thresholding step. It is seen that the first few transform coefficients in the

first column of the coefficients matrix are the largest in magnitude having the most important information content than the high-frequency components exhibiting better energy compaction property than 1D-DCT. Furthermore, it should also be stated that 2D-DCT coefficients of the PortSimHigh data set are much smaller than those of the PortSimLow data set, revealing that smaller coefficients are obtained when highly correlated data set is used. Consequently, these small coefficients can be discarded in quantization prior to encoding without significant distortion while achieving maximum amount of compression.

Semilog-log (upper sub-window of Figure 5.29) and log-log (bottom sub-window of Figure 5.29) plots of the sorted absolute values of the 2D-DCT coefficients of the overall PortSimLow data set, which are padded into a vector of size 5000000, and the threshold limit of 0.5622 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 5.29. The coefficients below the threshold limit (4975000 coefficients), which are identified clearly in the log-log plot, are set to zero for compression, in other words 99.5% of the transform coefficients become zero. The number of nonzero coefficients kept is 25000, which is only 0.5% of the number of data points.

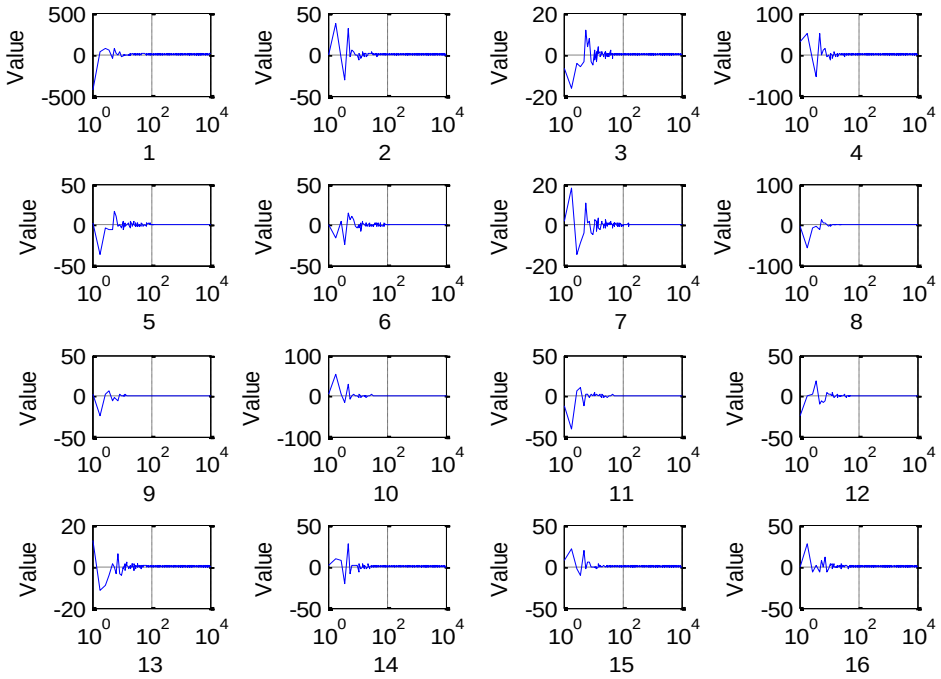

Figure 5.27. 2D-DCT Coefficients of the PortSimLow Data Set.

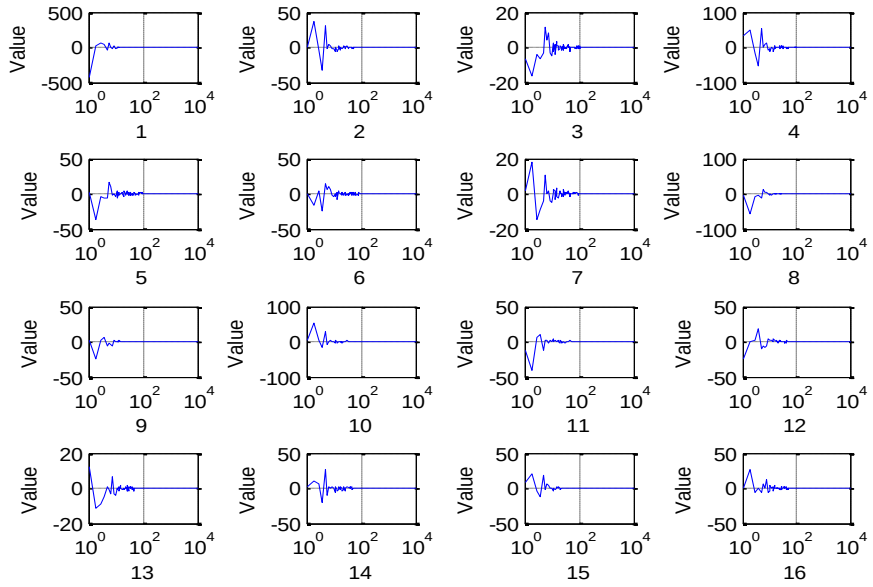

Figure 5.28. 2D-DCT Coefficients of the PortSimLow Data Set after Thresholding with α=99.5 %.

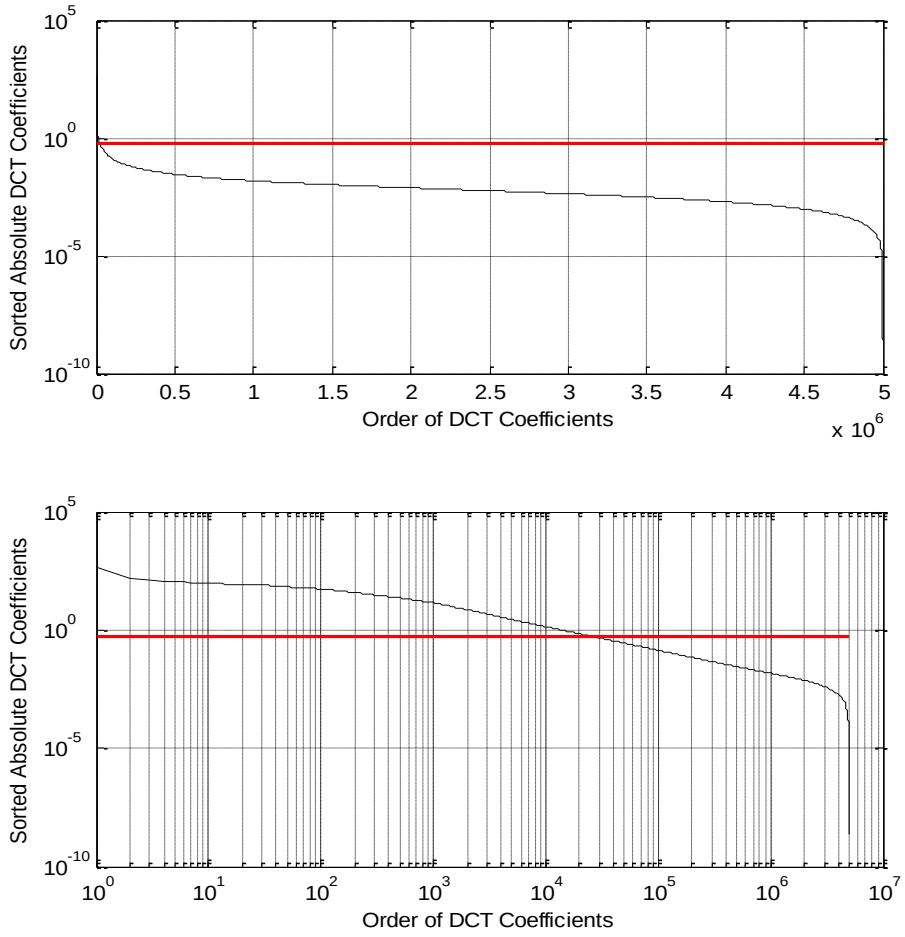

Figure 5.29. Semilog-log and Log-log Plots of Sorted Absolute 2D-DCT Coefficients of the Overall PortSimLow Data Set for α=99.5 %.

Sizes of the ZIP files of the original and encoded data sets are compared as it is seen from Figure 5.30, the ZIP file of the original data set is nearly 51.1 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.93 MB. Thus, compression can be increased 54.9 times by applying 2D-DCT technique and taking the percentile value as 99.5% in thresholding step (similar to the 1D-DCT results).

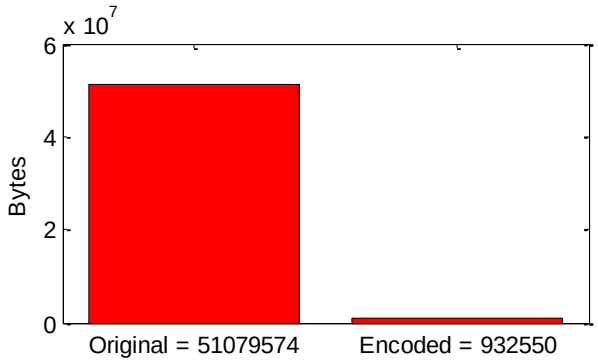

Figure 5.30. ZIP Compression Comparison of the Original and Encoded Overall PortSimLow Data Set for α=99.5 % with 2D-DCT.

Reconstructed data generated with 2D-iDCT (inverse DCT) and their overlay with the originals are shown in Figure 5.31 and Figure 5.32 respectively. It is seen that original and reconstructed data sets are overlapped as illustrated in Figure 5.32 concluding that there is not any significant loss in the decoded data set.

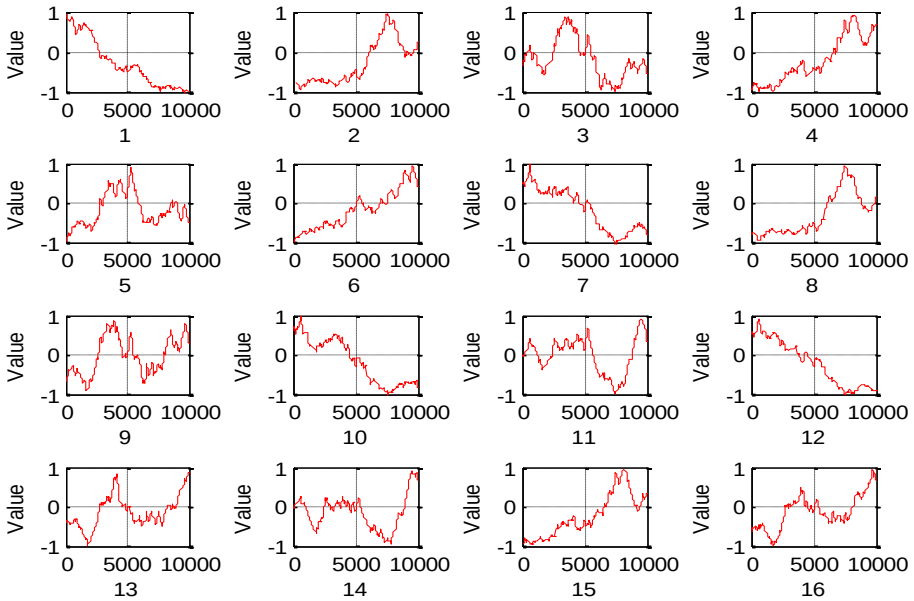

Figure 5.31. Reconstructed PortSimLow Data Set with Inverse 2D-DCT.

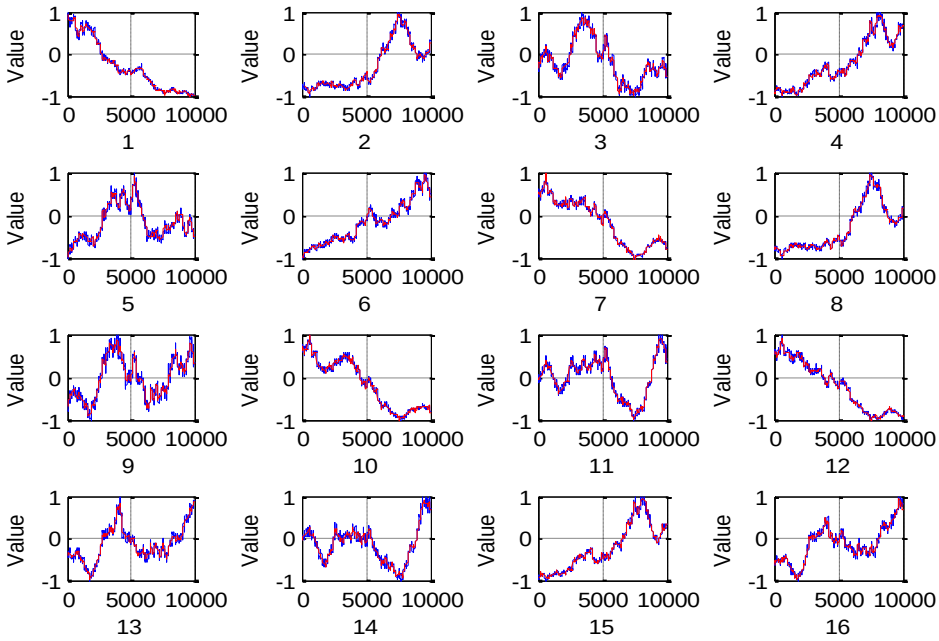

Figure 5.32. Original and Reconstructed Signals of the PortSimLow Data Set with Inverse 2D-DCT.

In Figure 5.33, the original and reconstructed signals produced with 2D-iDCT are plotted around the y=x line. It can be said that almost each of the reconstructed signals is similar to the original ones as most of the data points are located around the y=x line.

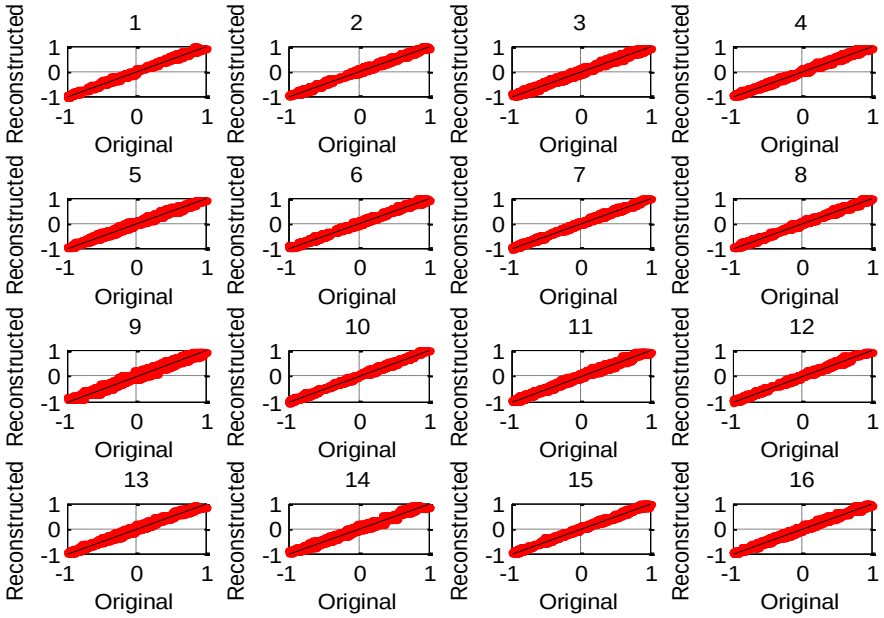

Figure 5.33. Reconstructed versus Original Signals of the PortSimLow Data Set with Inverse 2D-DCT.

Error norms between original and reconstructed data sets are calculated per data column. Reconstruction error norm values of the overall PortSimLow data set are given in the bottom sub-window of Figure 5.34. Minimum error norm of 199.94 is obtained in the 482nd column and the maximum error norm of 641.03 is obtained in the 122nd column of the PortSimLow data set after applying 2D-DCT. It can be concluded that 2D-DCT technique is not superior to 1D-DCT when they are applied to the less correlated PortSimLow data set as compression ratios produced in these two methods are almost the same and smaller distortion is produced in the latter technique.

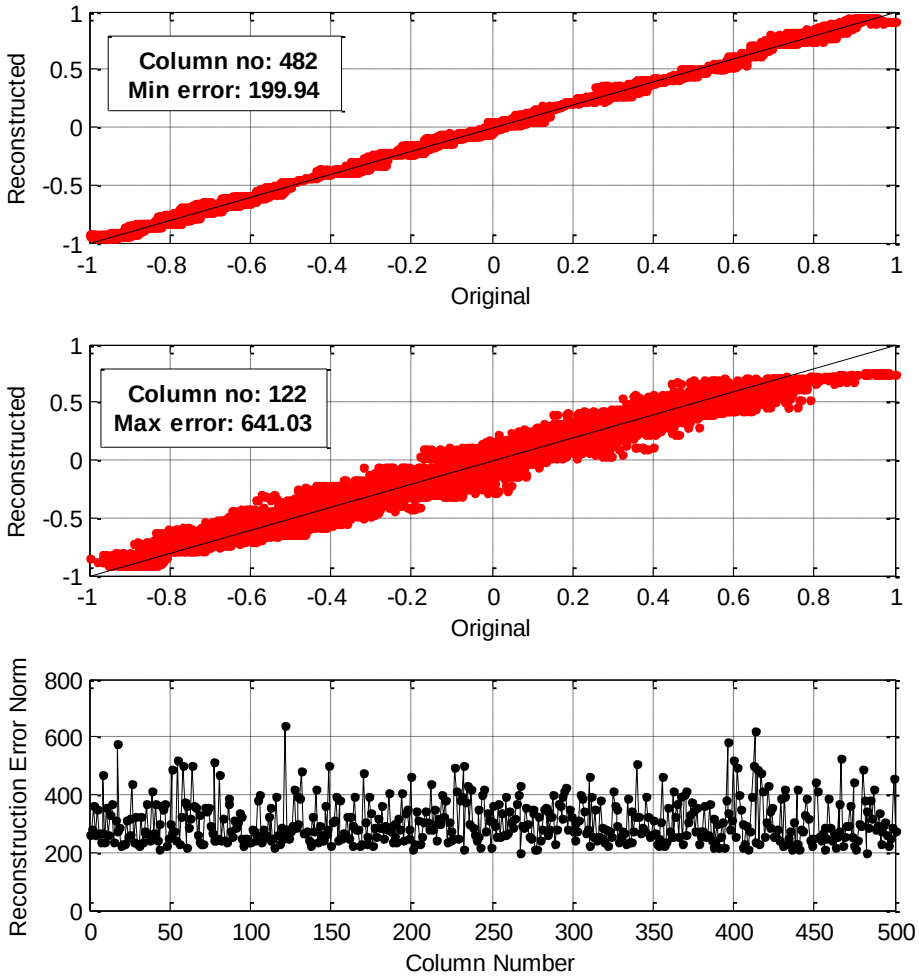

Figure 5.34. Reconstruction Error Norm Values of the PortSimLow Data Set with Inverse 2D-DCT.

5.3.5. One Dimensional Discrete Cosine Transform of the SELDI-TOF MS Data Set

In this section, 1D-DCT is applied to the overall SELDI-TOF MS data set (consisting of six ovarian cancer samples). The complete scaled intensities of the SELDI-TOF MS data set containing baseline noise and sparse peaks are shown in Figure 5.35.

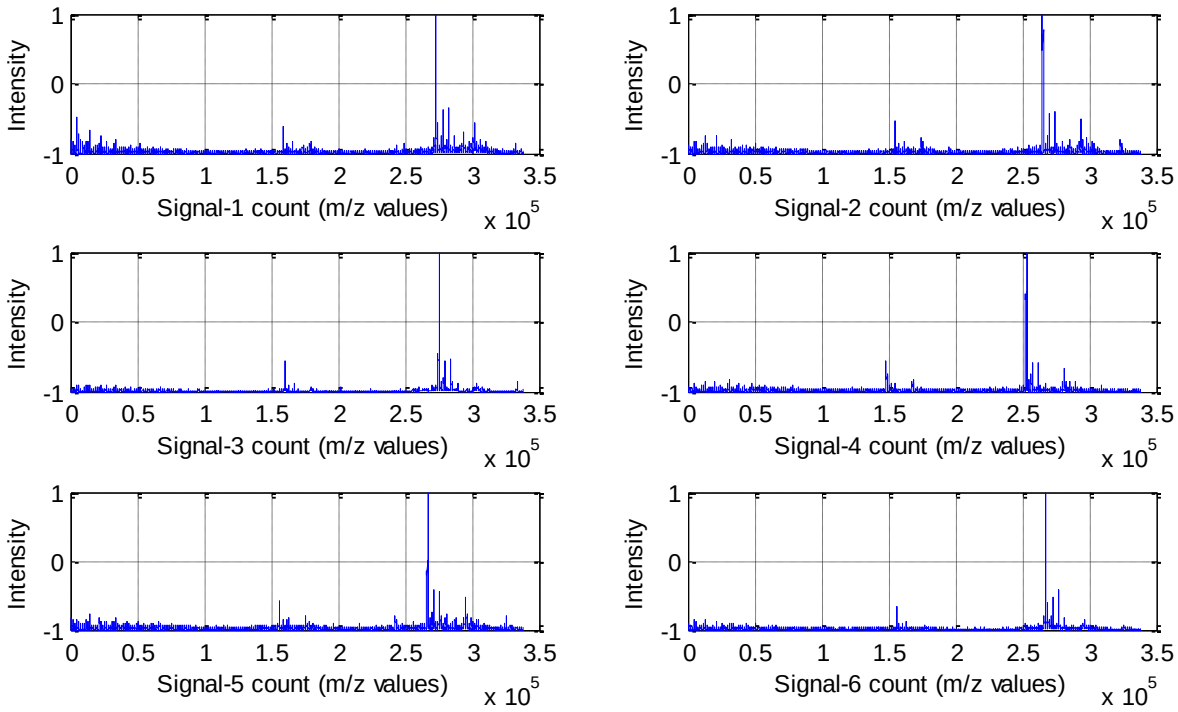

Figure 5.35. Scaled Intensities of the SELDI-TOF MS Data Set.

Transform coefficients of the scaled intensities of the SELDI-TOF MS data set obtained with 1D-DCT before and after thresholding, in which the percentile value is taken as 99.5%, are shown in Figure 5.36 and Figure 5.37. It is seen that the first few transform coefficients in each column of the coefficients matrix are large in magnitude having the most important information content. In thresholding step, these coefficients remain unchanged, whereas most of the high-frequency components become zero. In addition, baseline noise effect is observed in transform coefficients (especially in cancer samples one and two) as illustrated in Figure 5.36 whereas these coefficients are zeroed after thresholding concluding that 1D-DCT also provides de-noising. Hence, higher compression levels are expected as noise removal is achieved besides dimensionality reduction.

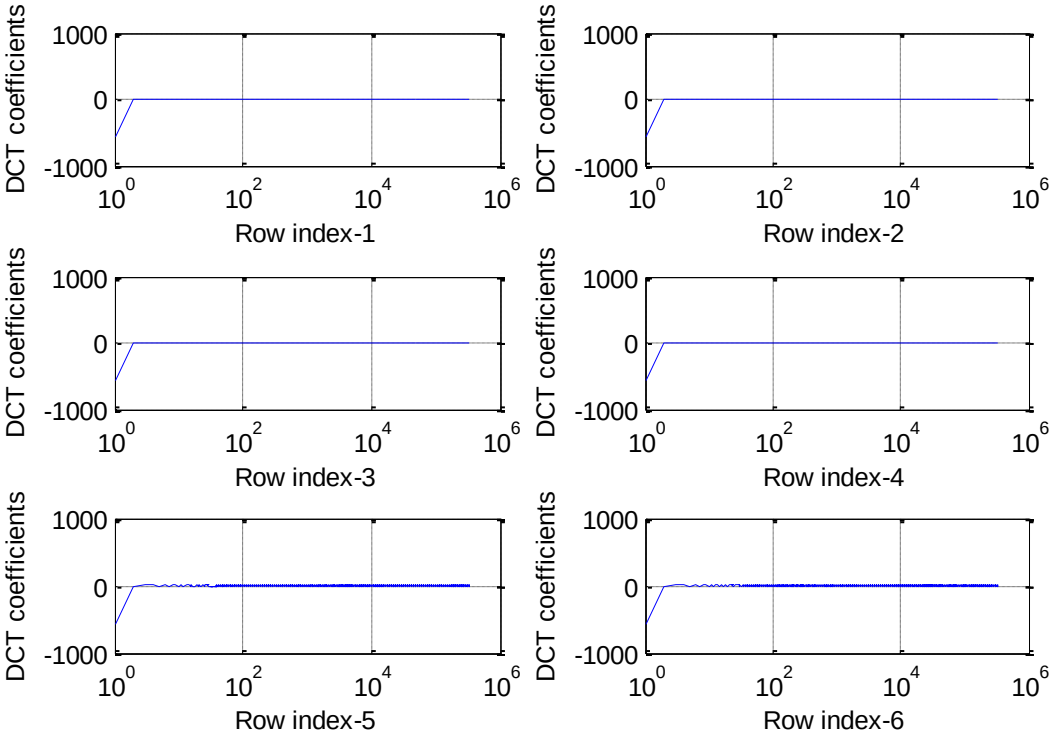

Figure 5.36. 1D-DCT Coefficients of the SELDI-TOF MS Data Set.

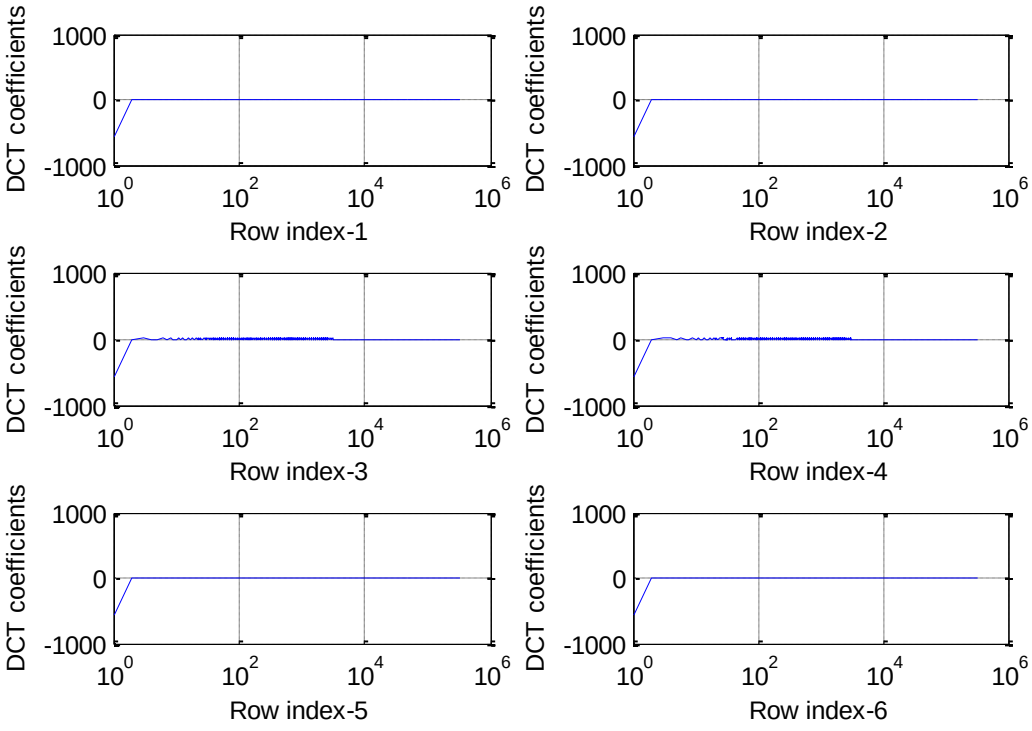

Figure 5.37. 1D-DCT Coefficients of the SELDI-TOF MS Data Set after Thresholding with α=99.5 %.

Semilog-log (upper sub-window of Figure 5.38) and log-log (bottom sub-window of Figure 5.38) plots of the sorted absolute values of the 1D-DCT coefficients of the overall SELDI-TOF MS data set, which are padded into a vector of size 2027928 and the threshold limit of 0.2101 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 5.38. Transform coefficients above and below the threshold limit can be identified clearly in the log-log plot. Transform coefficients below the threshold limit (2017788 coefficients) are set to zero for compression. The number of nonzero coefficients kept is 10140, which is only 0.5% of the number of data points.

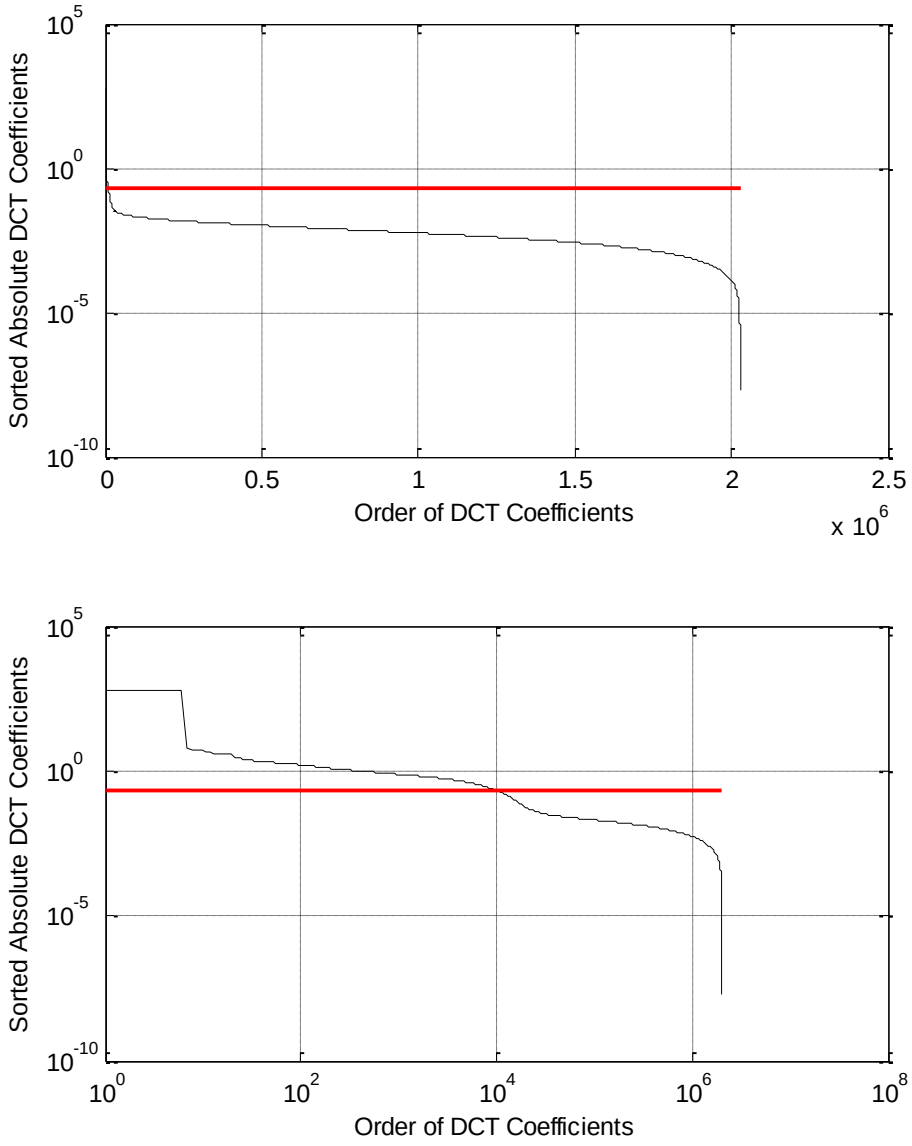

Figure 5.38. Semilog-log and Log-log Plots of Sorted Absolute 1D-DCT Coefficients of the Overall SELDI-TOF MS Data Set for α=99.5 %.

Sizes of the ZIP files of the original and encoded data sets are compared as it is seen from Figure 5.39, the ZIP file of the original data set is nearly 20.7 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.35 MB. Thus, ZIP compression can be increased 58.6 times by applying 1D-DCT technique and taking the percentile value as 99.5% in the thresholding step.

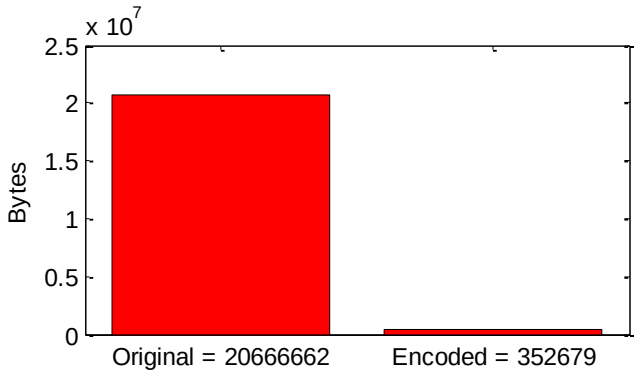

Figure 5.39. ZIP Compression Comparison of the Original and Encoded Overall SELDI-TOF MS Data Set for α=99.5 % with 1D-DCT.

Reconstructed data generated with 1D-iDCT (inverse DCT) and their overlay with the originals are shown in Figure 5.40 and Figure 5.41 respectively. It is seen that original and reconstructed data sets are similar as illustrated in Figure 5.41. However, there are some distortions observed in the magnitudes of the major peaks. Furthermore, it can be seen that reconstructed signals are de-noised, in other words removal of the irrelevant data is achieved.

The original and reconstructed signals produced with 1D-iDCT are plotted around the y=x line in Figure 5.42. It can be stated that scaled intensities having the magnitudes over 0.5 and the magnitudes between [-1,-0.5] cannot be reconstructed thoroughly (observed in all of the six cancer samples) due to the sharp peaks occurred in these regions.

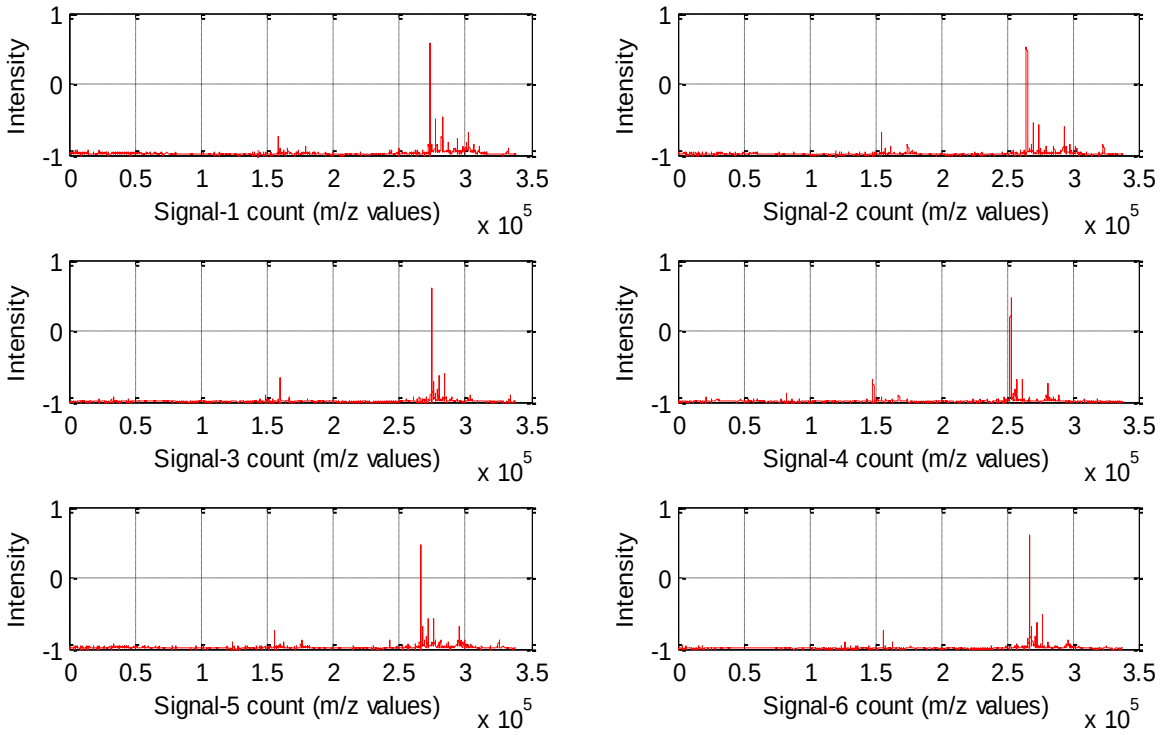

Figure 5.40. Reconstructed SELDI-TOF MS Data Set with Inverse 1D-DCT.

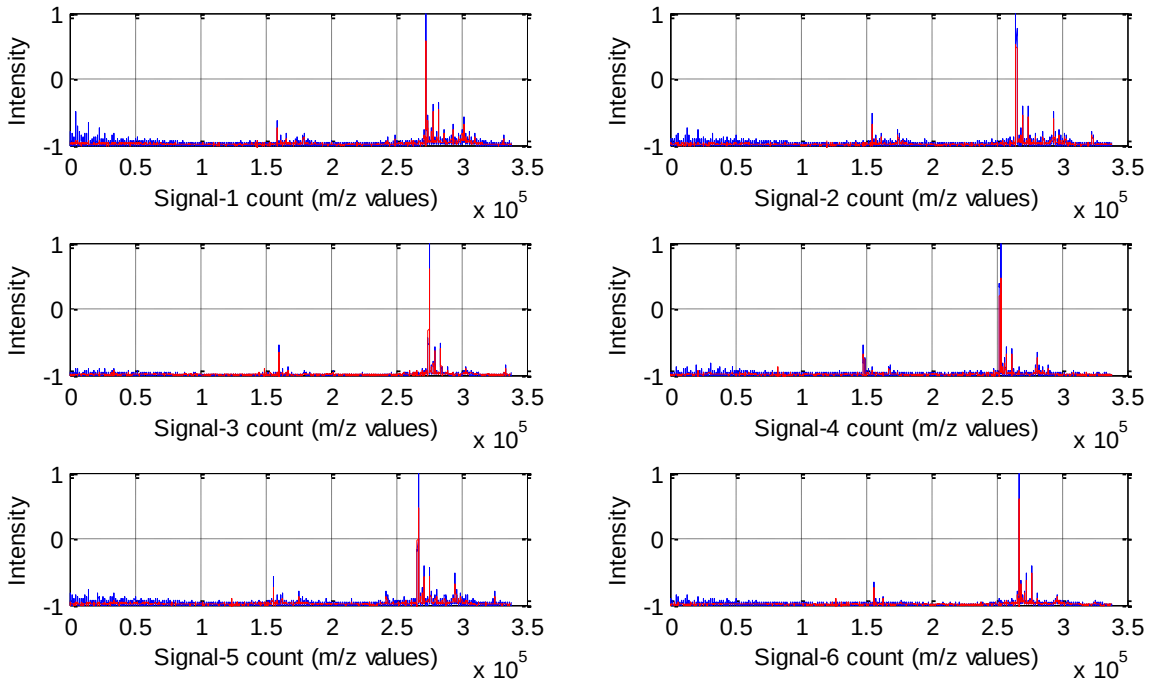

Figure 5.41. Original and Reconstructed Signals of the SELDI-TOF MS Data Set with Inverse 1D-DCT.

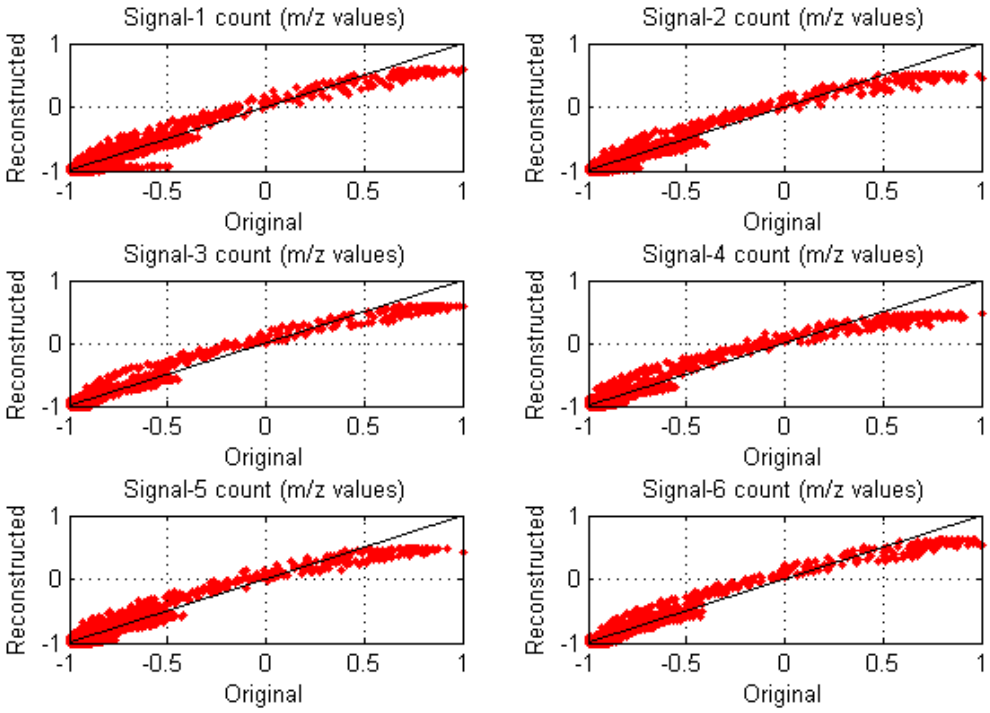

Figure 5.42. Reconstructed versus Original Signals of the SELDI-TOF MS Data Set with Inverse 1D-DCT.

Error norms between original and reconstructed data sets are calculated per data column. Reconstruction error norm values of the overall SELDI-TOF MS data set are given in the bottom sub-window of Figure 5.43. Minimum error norm of 2029.2 is obtained in the third column and the maximum error norm of 3447.8 is obtained in the fifth column of the SELDI-TOF MS data set after applying 1D-DCT. As it can be seen from the upper and the middle sub-windows of Figure 5.43, third column is reconstructed better than the fifth column as most of the reconstructed data points of the third column are located around the y=x line. Even so, it can be concluded that reconstruction error norms are generally quite large in magnitude as SELDI-TOF MS data set contains sharp peaks and baseline noise.

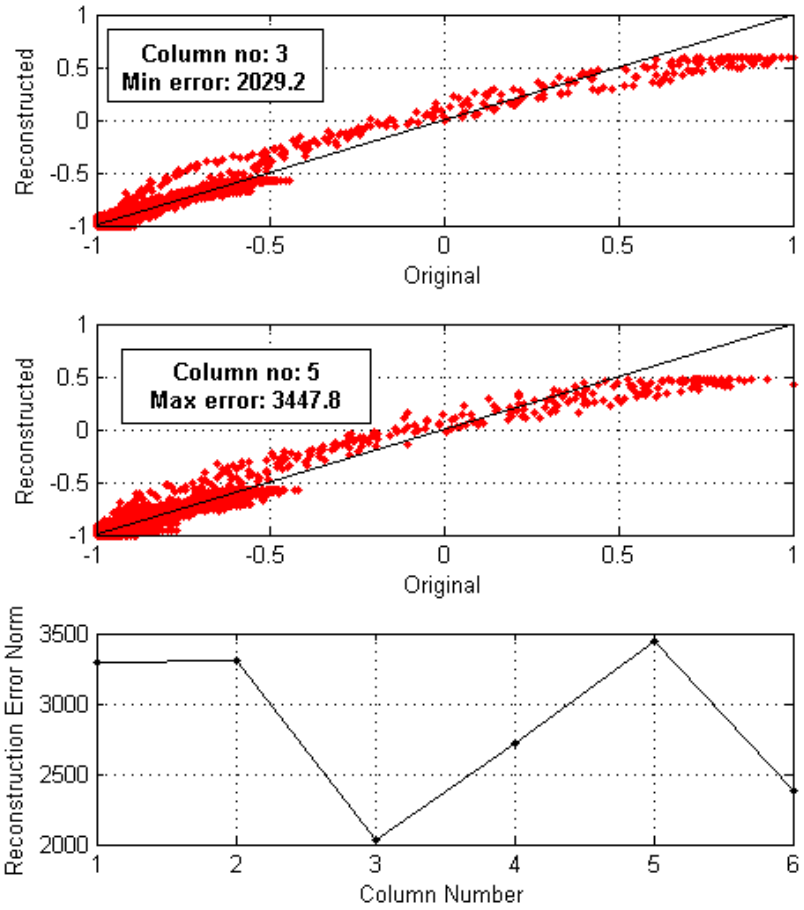

Figure 5.43. Reconstruction Error Norm Values of the SELDI-TOF MS Data Set with Inverse 1D-DCT.

5.3.6. Two Dimensional Discrete Cosine Transform of the SELDI-TOF MS Data Set

In this section, 2D-DCT is applied to the overall SELDI-TOF MS data set (consisting of six ovarian cancer samples). The complete scaled intensities of the SELDI-TOF MS data set were given before in the previous section in scaled format with Figure 5.35.

Transform coefficients of the scaled intensities of the SELDI-TOF MS data set obtained with 2D-DCT before and after thresholding, in which the percentile value is taken as 99.5%, are shown in Figure 5.44 and Figure 5.45. It is seen that the first few transform coefficients (with large magnitudes) in the first column of the coefficients matrix store the most important information content than the high frequency components exhibiting better energy compaction property than 1D-DCT. High-frequency coefficients which are close to zero (after about first 10000 rows in all columns) become zero in thresholding. The effect

of the baseline noise in the original data set is also observed in the columns of the coefficients matrix.

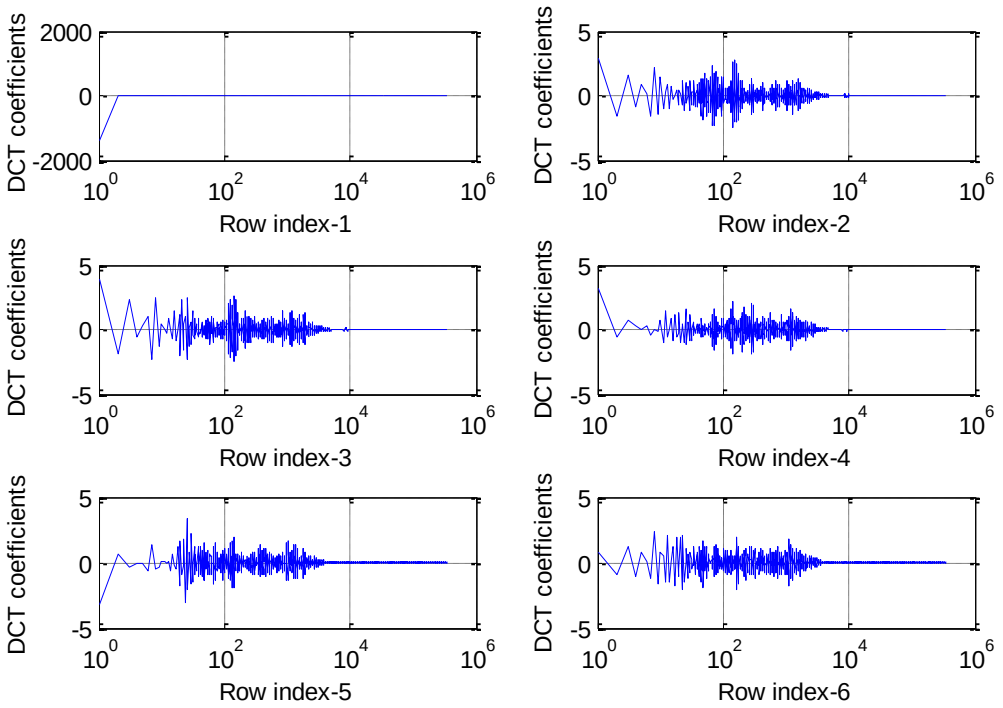

Figure 5.44. 2D-DCT Coefficients of the SELDI-TOF MS Data Set.

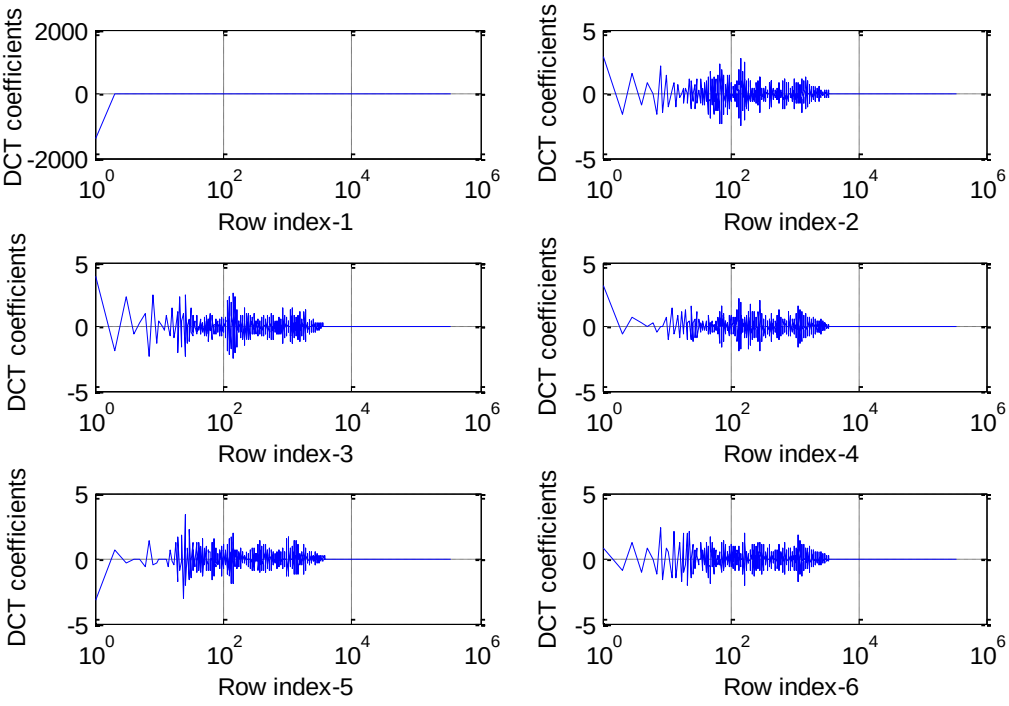

Figure 5.45. 2D-DCT Coefficients of the SELDI-TOF MS Data Set after Thresholding with α=99.5 %.

Semilog-log (upper sub-window of Figure 5.46) and log-log (bottom sub-window of Figure 5.46) plots of the sorted absolute values of the 2D-DCT coefficients of the overall SELDI-TOF MS data set, which are padded into a vector of size 2027928 and the threshold limit of 0.1694 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 5.46. The coefficients below the threshold limit (2017788 coefficients), which are identified clearly in the log-log plot, are set to zero for compression, in other words 99.5% of the transform coefficients become zero. The number of nonzero coefficients kept is 10140, which is only 0.5% of the number of data points. In addition, the threshold limit is slightly smaller than that found with 1D-DCT, which was 0.2101 as mentioned in Section 5.3.5, since 2D-DCT coefficients are rather small.

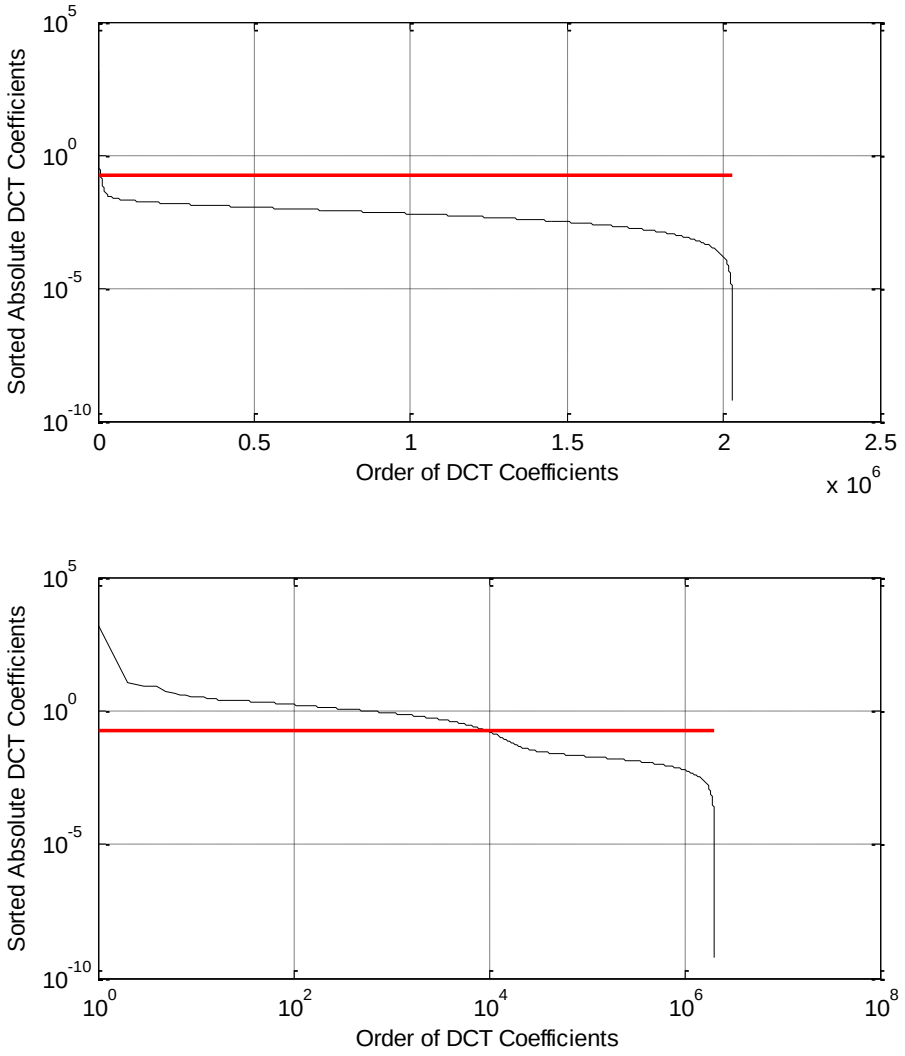

Figure 5.46. Semilog-log and Log-log Plots of Sorted Absolute 2D-DCT Coefficients of the Overall SELDI-TOF MS Data Set for α=99.5 %.

Sizes of the ZIP files of the original and encoded data sets are compared as it is seen from Figure 5.47, the ZIP file of the original data set is nearly 20.7 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.35 MB. Thus, ZIP compression can be increased 58.4 times (similar to 1D-DCT results) by applying 2D-DCT technique and taking the percentile value as 99.5% in the thresholding step.

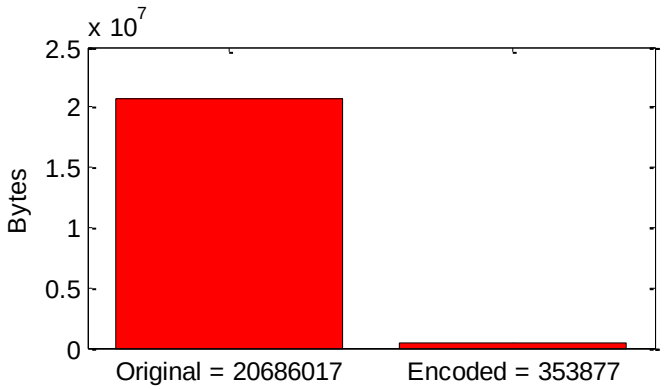

Figure 5.47. ZIP Compression Comparison of the Original and Encoded Overall SELDI-TOF MS Data Set for α=99.5 % with 2D-DCT.

Reconstructed data are generated with 2D-iDCT (inverse DCT). The reconstructed data and their overlay with the originals are shown in Figure 5.48 and Figure 5.49 respectively. It is seen that original and reconstructed data sets are overlapped as illustrated in Figure 5.49. However, there are some small distortions observed in reconstructed peaks (observed in all of the six cancer samples).

In Figure 5.50, the original and reconstructed signals obtained with 2D-iDCT are plotted around the y=x line. It can be stated that the scaled intensities having the magnitudes over 0.5 and magnitudes between [-1,-0.5] cannot be reconstructed thoroughly (similar to 1D-DCT results) due to the peaks observed in these regions since corresponding reconstructed data points are quite far from the y=x line.

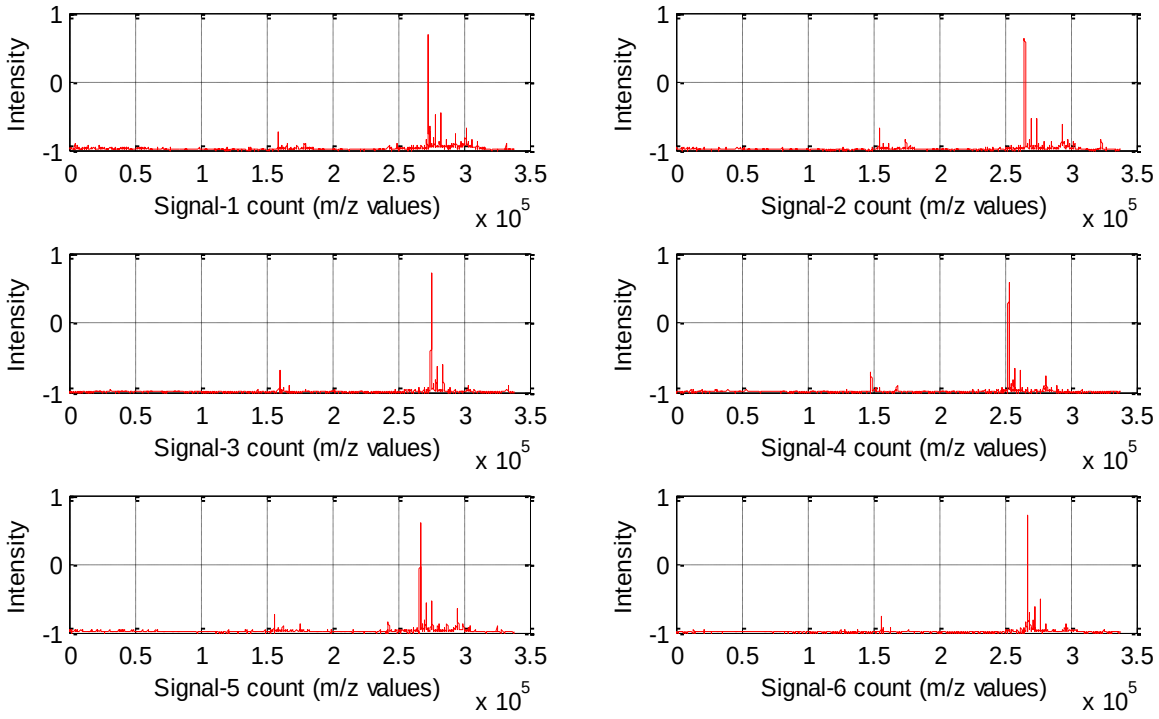

Figure 5.48. Reconstructed SELDI-TOF MS Data Set with Inverse 2D-DCT.

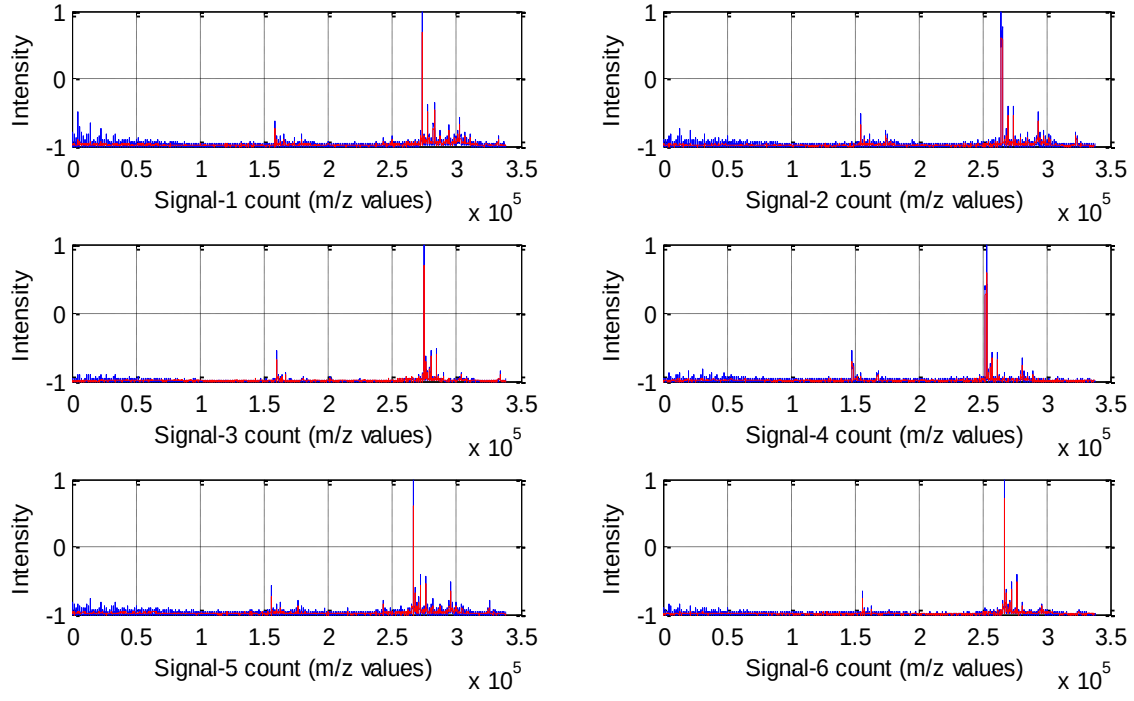

Figure 5.49. Original and Reconstructed Signals of the SELDI-TOF MS Data Set with Inverse 2D-DCT.

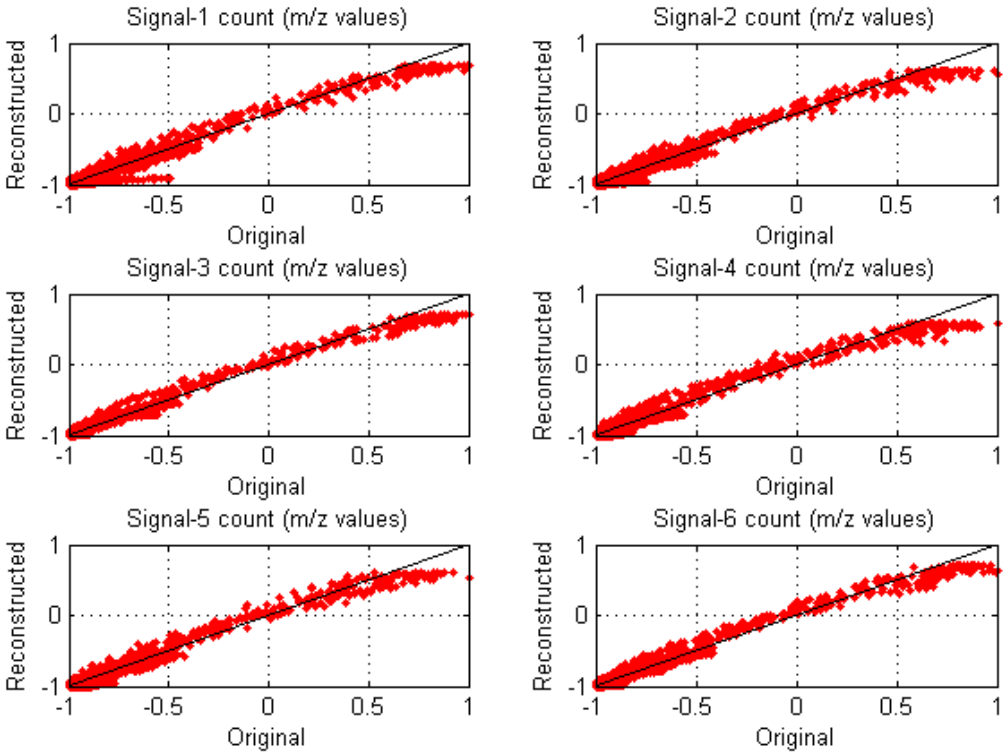

Figure 5.50. Reconstructed versus Original Signals of the SELDI-TOF MS Data Set with Inverse 2D-DCT.

Reconstruction error norm values of the overall SELDI-TOF MS data set calculated per data column are given in the bottom sub-window of Figure 5.51. Minimum error norm of 1965.2 is obtained in the third column and the maximum error norm of 3325.1 is obtained in the fifth column of the SELDI-TOF MS data set after applying 2D-DCT. As it can be seen from the upper and the middle sub-windows of Figure 5.51, third column is reconstructed better than the fifth column as most of the reconstructed data points of the third column are located around the y=x line. It can also be stated that slightly smaller reconstruction error norms are produced with 2D-DCT than 1D-DCT although these two techniques give almost the same compression ratios, concluding that 2D-DCT can be the preferred technique when highly uncorrelated SELDI-TOF MS data set is used.

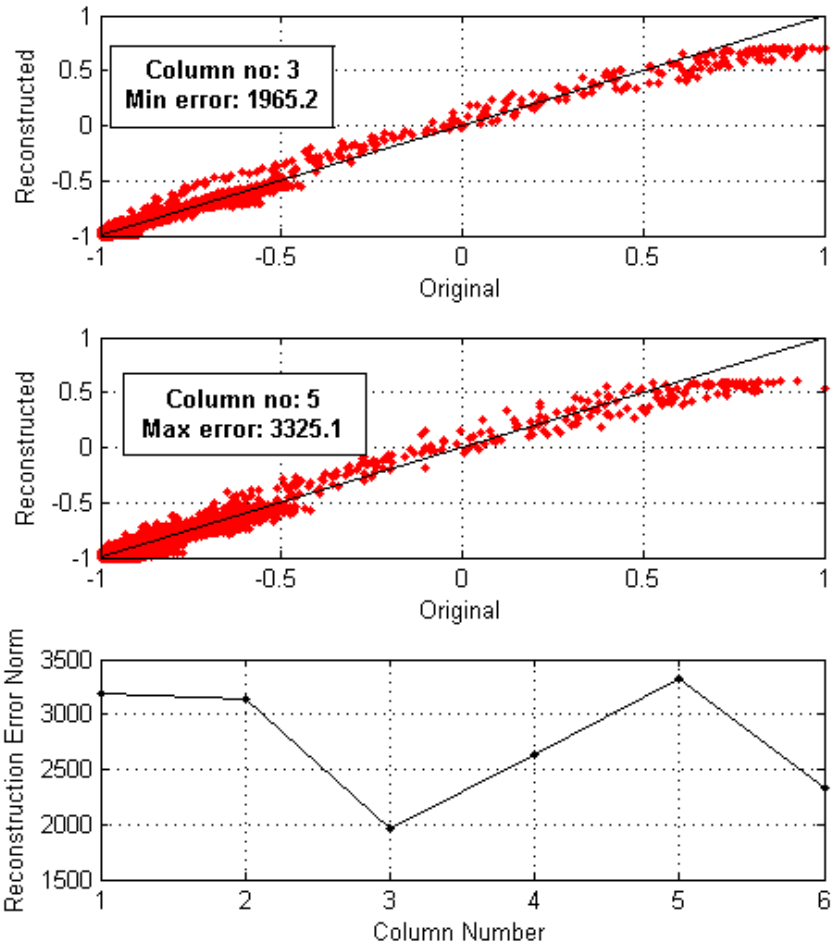

Figure 5.51. Reconstruction Error Norm Values of the SELDI-TOF MS Data Set with Inverse 2D-DCT.

5.3.7. One Dimensional Discrete Cosine Transform of the TEP Data Set

In this section, 1D-DCT is applied to the overall TEP data set (consisting of 41 measurements). Overall output signals of the TEP including all of the 41 measured variables are shown in Figure 5.52. Each process measurement exhibits different features such as upward/downward shifts (observed in measurements one and four) and decreasing/increasing trends (observed in measurements 28 and 34) occurred due to the consecutive fault disturbances introduced.

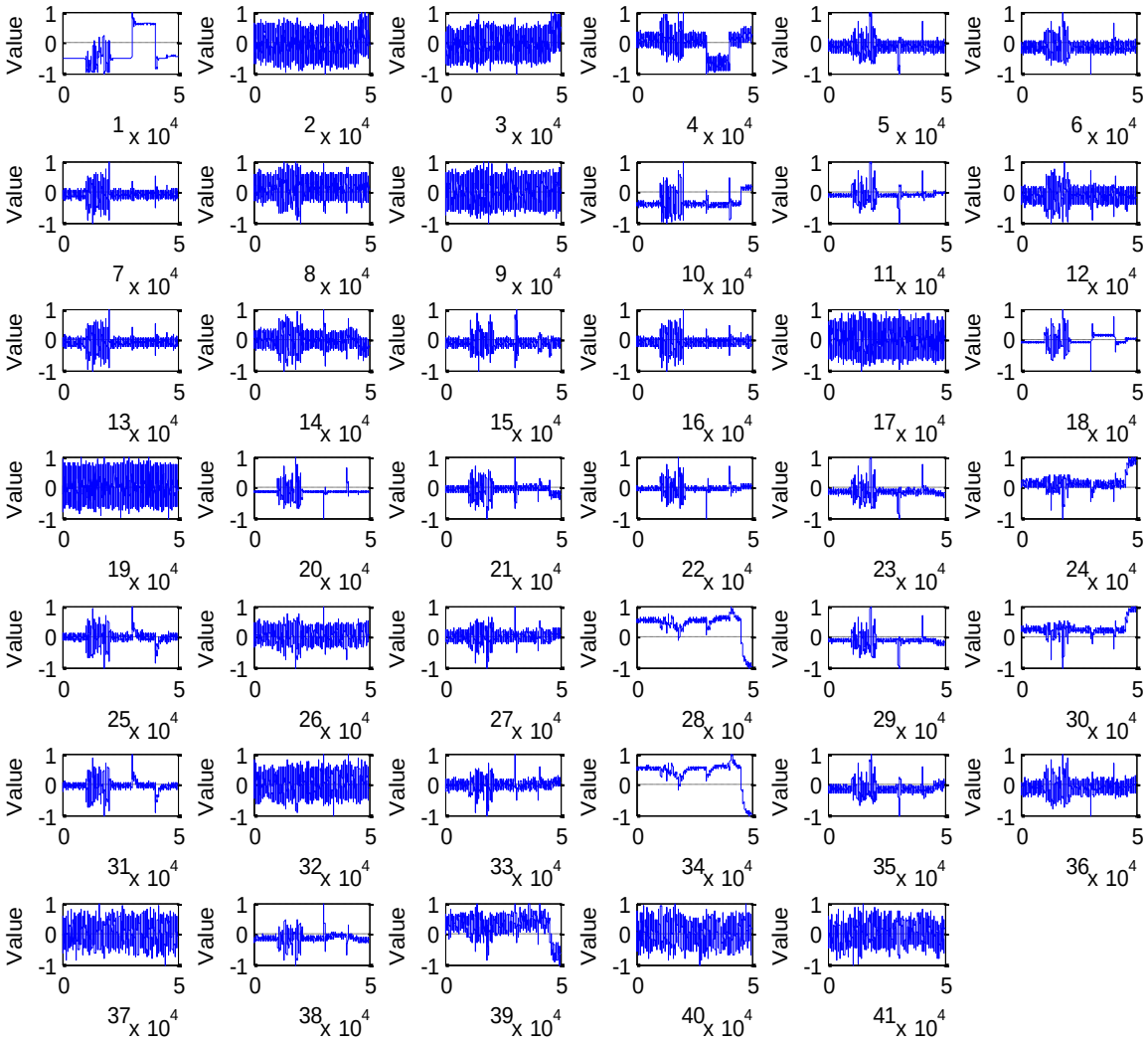

Figure 5.52. Complete Output Signals of the TEP Data Set as a result of Three Consecutive Fault Disturbances in Scaled Format.

Transform coefficients of the output signals of the TEP obtained with 1D-DCT before and after thresholding, in which the percentile value is taken as 99.5%, are shown in Figure 5.53 and Figure 5.54. Only the first 400 of the 50001 transform coefficients are given to observe the coefficient magnitudes in detail. It is seen that transform coefficients do not die out exponentially towards zero (measurements nine, 37, 40 and 41), they persist to be significant throughout the vector. Thus, thresholding method is applied instead of zero padding. After thresholding, most of the coefficients with small amplitudes become zero (measurements one, 10, 30 and 34), whereas the coefficients out of the threshold limits (measurements 25, 27, 40 and 41) are stored as illustrated in Figure 5.54.

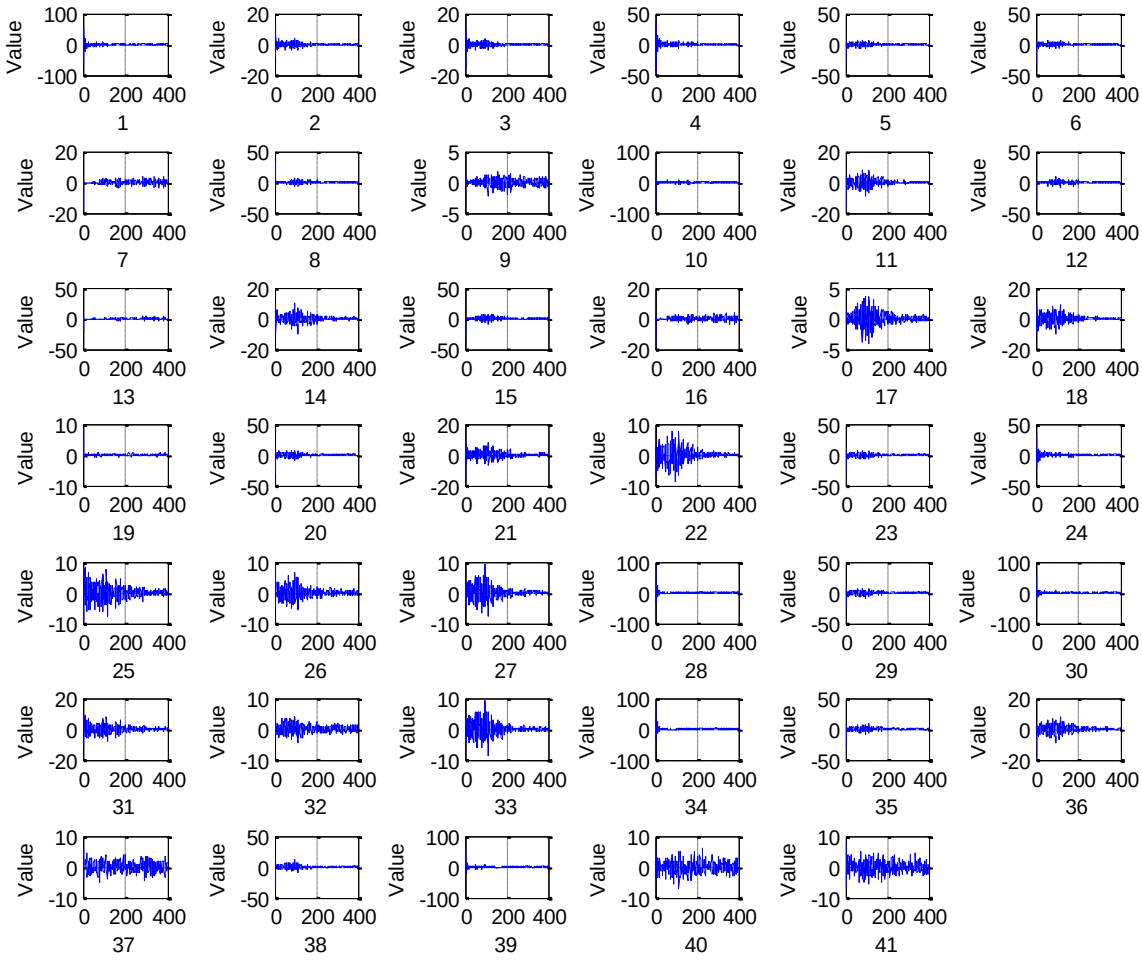

Figure 5.53. First 400 1D-DCT Coefficients of the Complete Output Signals of the TEP Data Set.

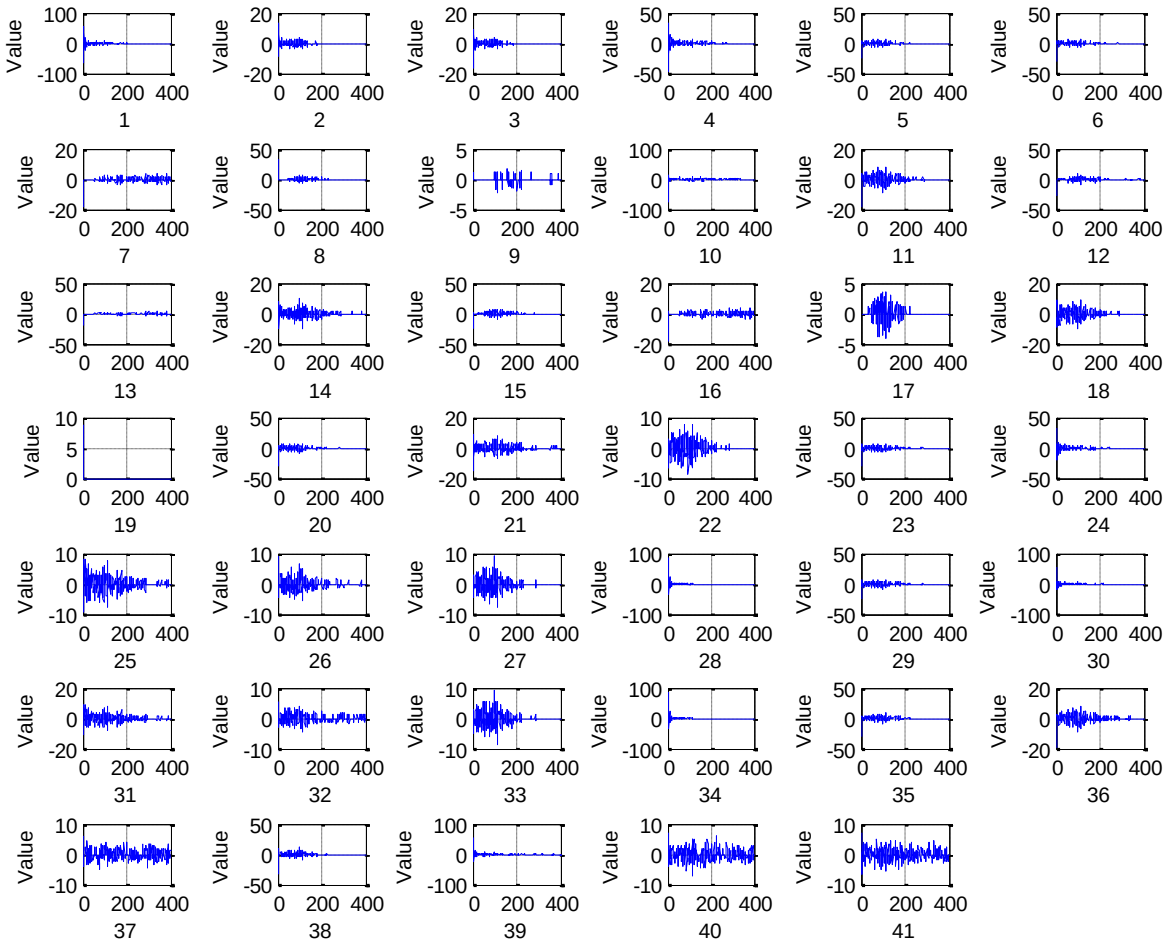

Figure 5.54. First 400 1D-DCT Coefficients of the Complete Output Signals of the TEP Data Set after Thresholding with α=99.5 %.

Semilog-log (upper sub-window of Figure 5.55) and log-log (bottom sub-window of Figure 5.55) plots of the sorted absolute values of the 1D-DCT coefficients of all 41 columns, which are padded into a vector of size 2050041 and the threshold limit of 0.967 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 5.55. Transform coefficients below the threshold limit (2039791 coefficients), which can be identified clearly in the log-log plot, are set to zero for compression. The number of nonzero coefficients kept is 10250, in other words 0.5% of the transform coefficients remain unchanged. Error in reconstruction step will increase as the number of zeroed transform coefficients increases as mentioned in Chapter 2.

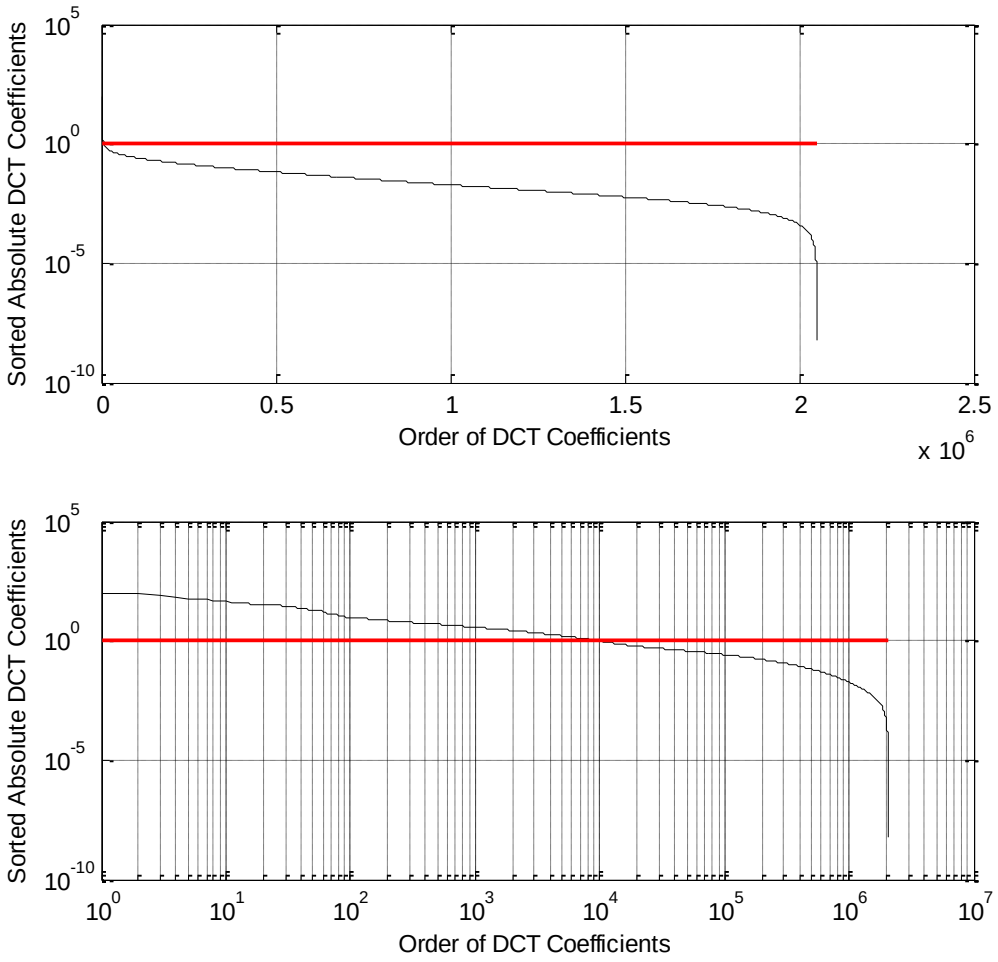

Figure 5.55. Semilog-log and Log-log Plots of Sorted Absolute 1D-DCT Coefficients of the Overall TEP Data Set for α=99.5 %.

Sizes of the ZIP files of the original and encoded data sets are compared as it is seen from Figure 5.56, the ZIP file of the original data set is nearly 21 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.38 MB. Thus, ZIP compression can be increased 55.5 times by applying 1D-DCT technique and taking the percentile value as 99.5% in the thresholding step. Large compression is expected as the filtered transform coefficients consist of repeating patterns (zeros) which are favored in lossless compression algorithms, such as the ones used in ZIP software.

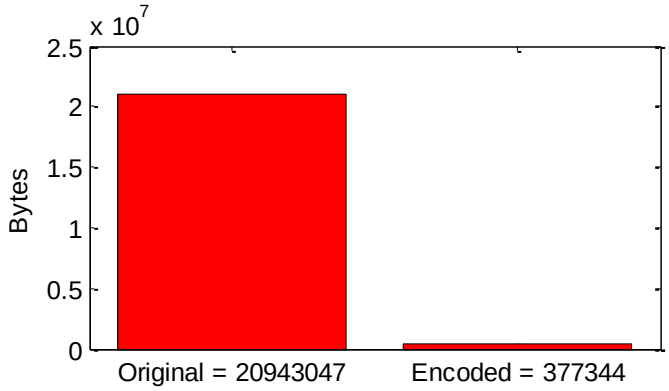

Figure 5.56. ZIP Compression Comparison of the Original and Encoded Overall TEP Data Set for α=99.5 % with 1D-DCT.

Reconstructed data generated with 1D-iDCT (inverse DCT) and their overlay with the originals are shown in Figure 5.57 and Figure 5.58 respectively. The major process features including important peak points, upward/downward shifts (observed in measurements one and four) and decreasing/increasing trends (observed in measurements 28 and 34) occurred due to the consecutive fault disturbances are reconstructed thoroughly as illustrated in Figure 5.57. In addition, noise in measured data makes compression difficult. For this reason, noise removal, in other words loss of the irrelevant data, is one of the major advantages of the dimensionality reduction technique, DCT (due to its decorrelation property); improving compression. It can be seen from Figure 5.58 that reconstructed values of the measurements consisting of almost pure noise (measurements nine and 19) are smoothened around a straight line. Except these noisy measurements, the amplitudes of the reconstructed data set are almost same as those of the output signals of the TEP data set concluding that there is not any significant loss in information content.

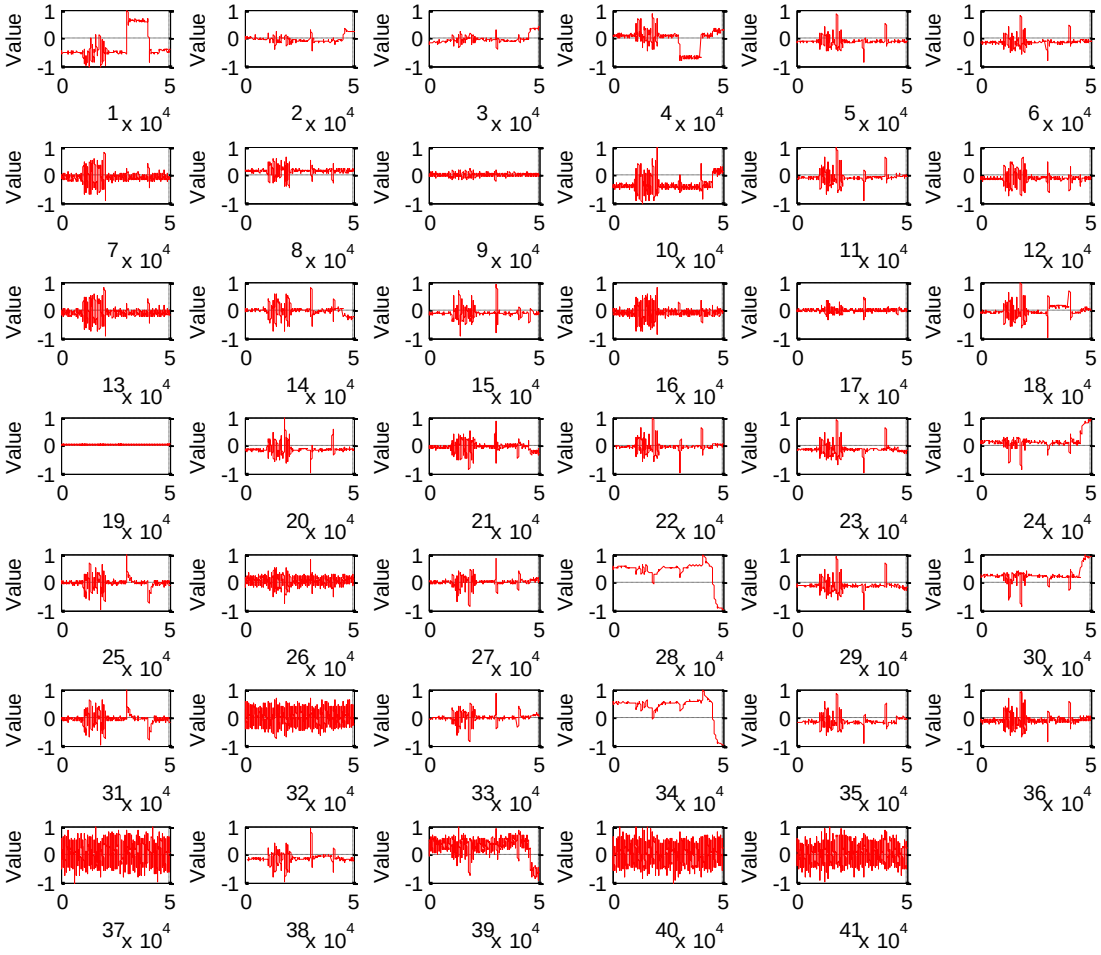

Figure 5.57. Reconstructed TEP Data Set with Inverse 1D-DCT.

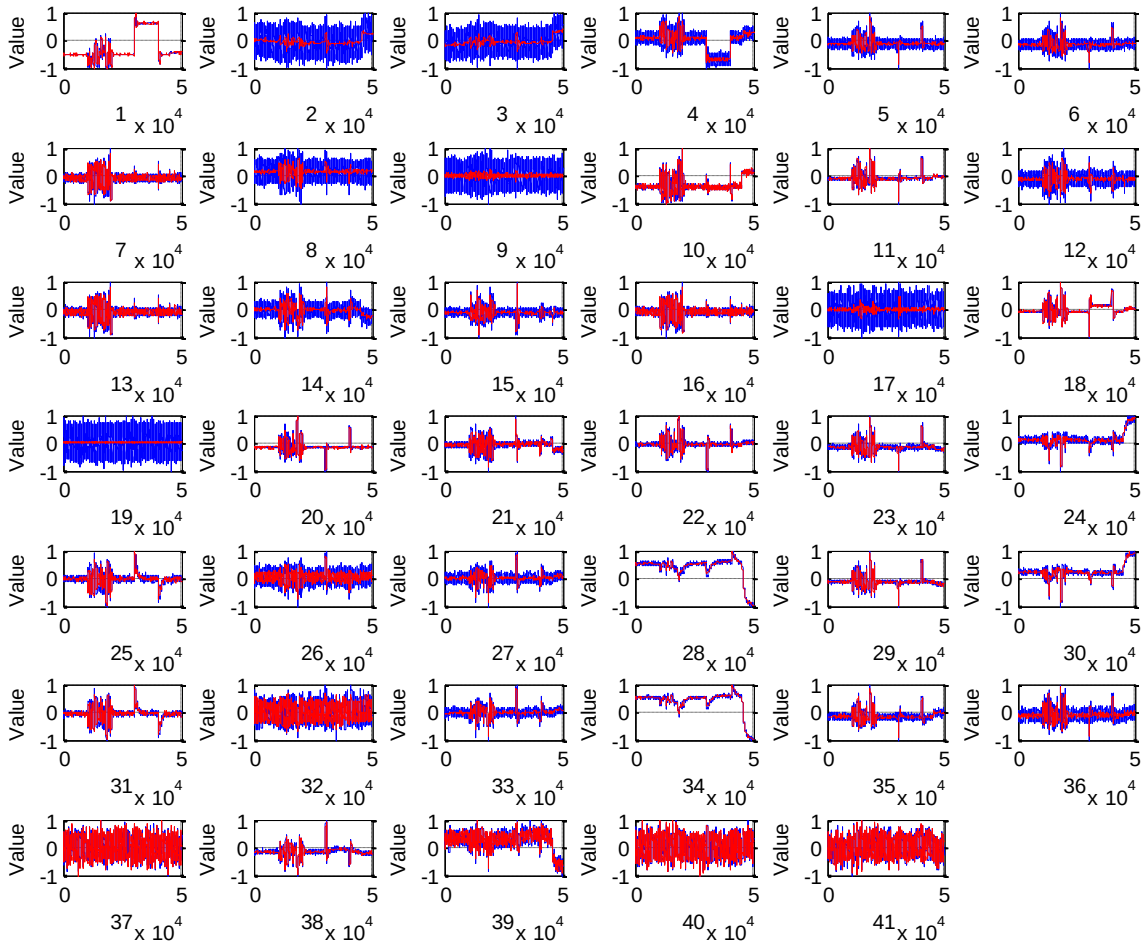

Figure 5.58. Original and Reconstructed Signals of the TEP Data Set with Inverse 1D-DCT.

In Figure 5.59, the original and reconstructed signals are plotted around the y=x line. Reconstructed data points around the y=x line indicates that reconstructed signals are very close to the original signals (measurements one, 11 and 34) whereas, reconstructed data points around the y=0 line shows that they are completely different from the original signals (measurements two, three and 19).

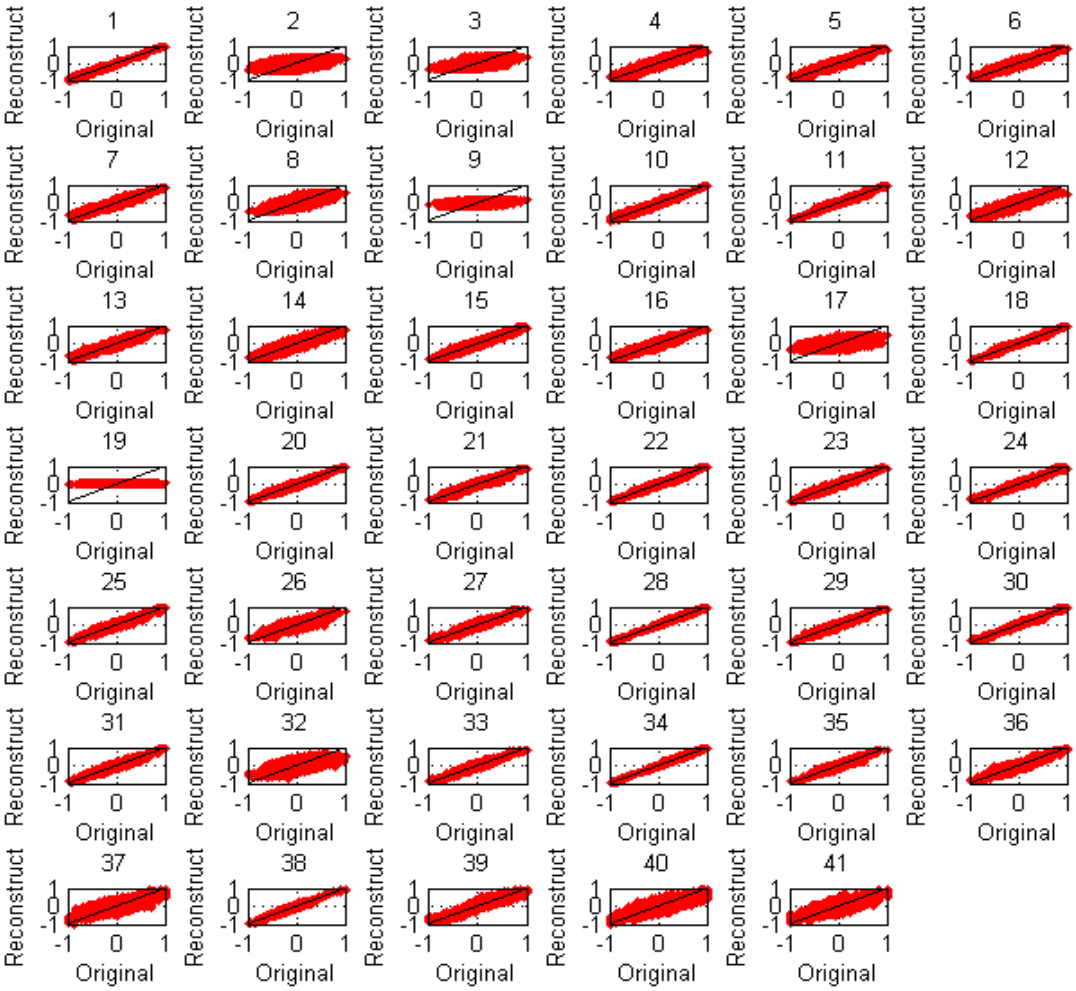

Figure 5.59. Reconstructed versus Original Signals of the TEP Data Set with Inverse 1D-DCT.

Error norms between original and reconstructed data sets are calculated per data column. Minimum error norm of 1283.5 is obtained in the 34th column of the TEP data set, the measurement showing decreasing trend with less noise. As it can be seen from the upper sub-window of Figure 5.60, reconstructed data points cluster around the y=x line meaning that reconstructed signals are very close to the original signals. On the other hand, the maximum error norm of 9669.5 is obtained in the 19th column of the TEP data set. Actually, it can be said that the information loss is appropriate for this measurement which consists of almost pure noise, as noise removal is achieved providing better compression. Reconstructed data points are located on the y=0 line as illustrated in the middle sub-window of Figure 5.60, in other words, reconstructed signals are completely different from the original signals. Furthermore, reconstruction error norm values of each column of the TEP data set are given in the bottom sub-window of Figure 5.60.

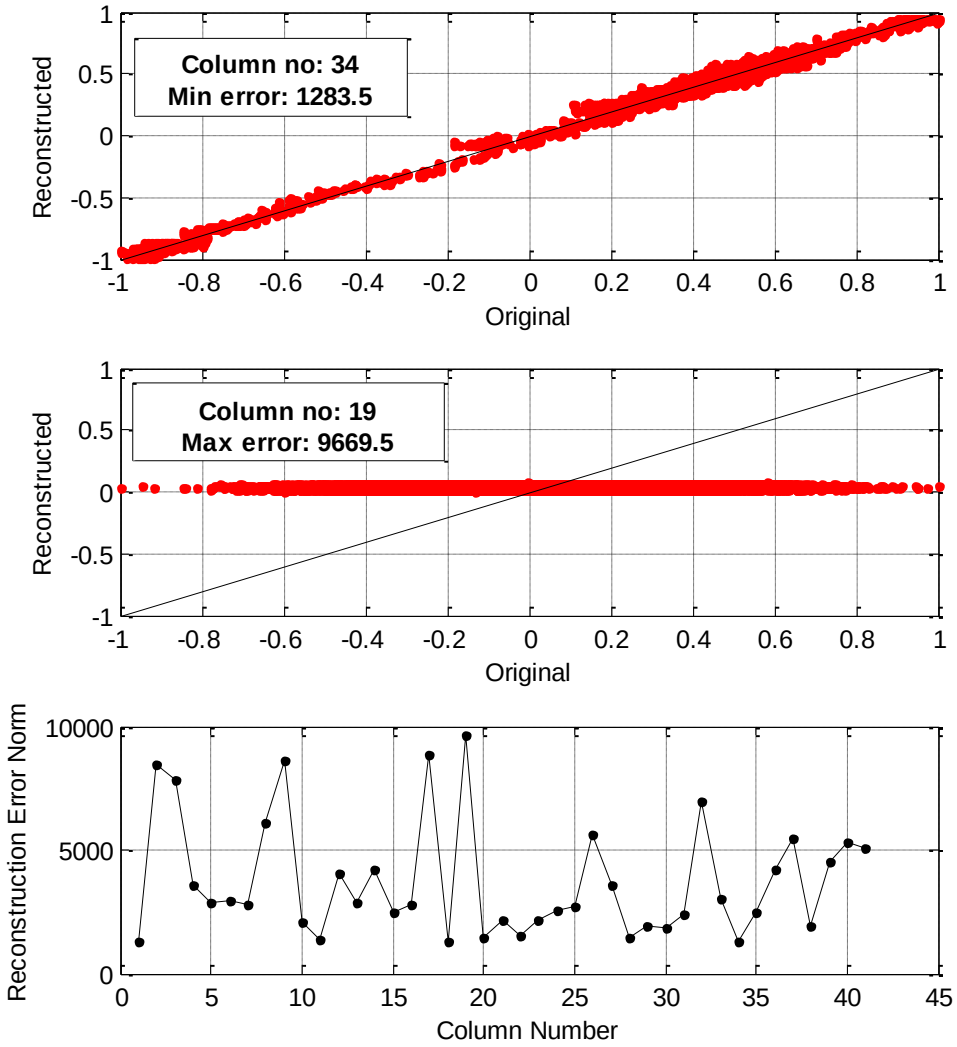

Figure 5.60. Reconstruction Error Norm Values of the TEP Data Set with Inverse 1D-DCT.

5.3.8. Two Dimensional Discrete Cosine Transform of the TEP Data Set

In this section, 2D-DCT is applied to the overall TEP data set (consisting of 41 measurements). The complete output signals of the TEP data set were given before in the previous section in scaled format with Figure 5.52.

Transform coefficients of the output signals of the TEP obtained with 2D-DCT before and after thresholding, in which the percentile value is taken as 99.5%, are shown in Figure 5.61 and Figure 5.62. Only the first 400 of the 50001 transform coefficients are given to be able to observe the magnitudes of the coefficients in detail.

Unlike PortSimHigh and SELDI-TOF MS data sets, the first few coefficients of the first column of the coefficients matrix do not store the most important information content, in other words 2D-DCT does not exhibit good energy compaction property when TEP data set is used. Furthermore, transform coefficients of the TEP data set do not die out exponentially towards zero (measurements nine, 37, 40 and 41), they persist to be significant throughout the vector as shown in Figure 5.61. After thresholding, high-frequency coefficients with small amplitudes become zero (measurements 27, 38 and 40), whereas the coefficients out of the threshold limits (measurements three, 11 and 24) are kept as illustrated in Figure 5.62.

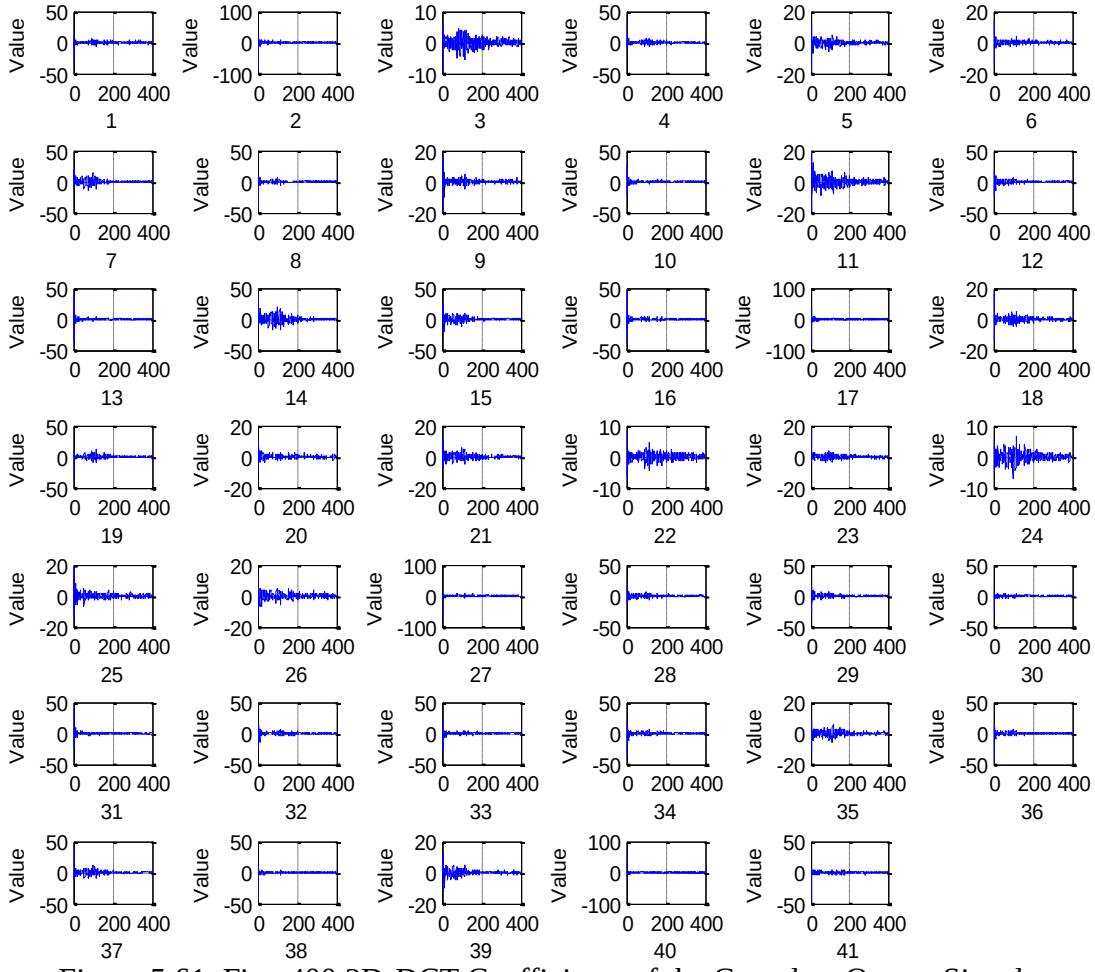

Figure 5.61. First 400 2D-DCT Coefficients of the Complete Output Signals of the TEP Data Set.

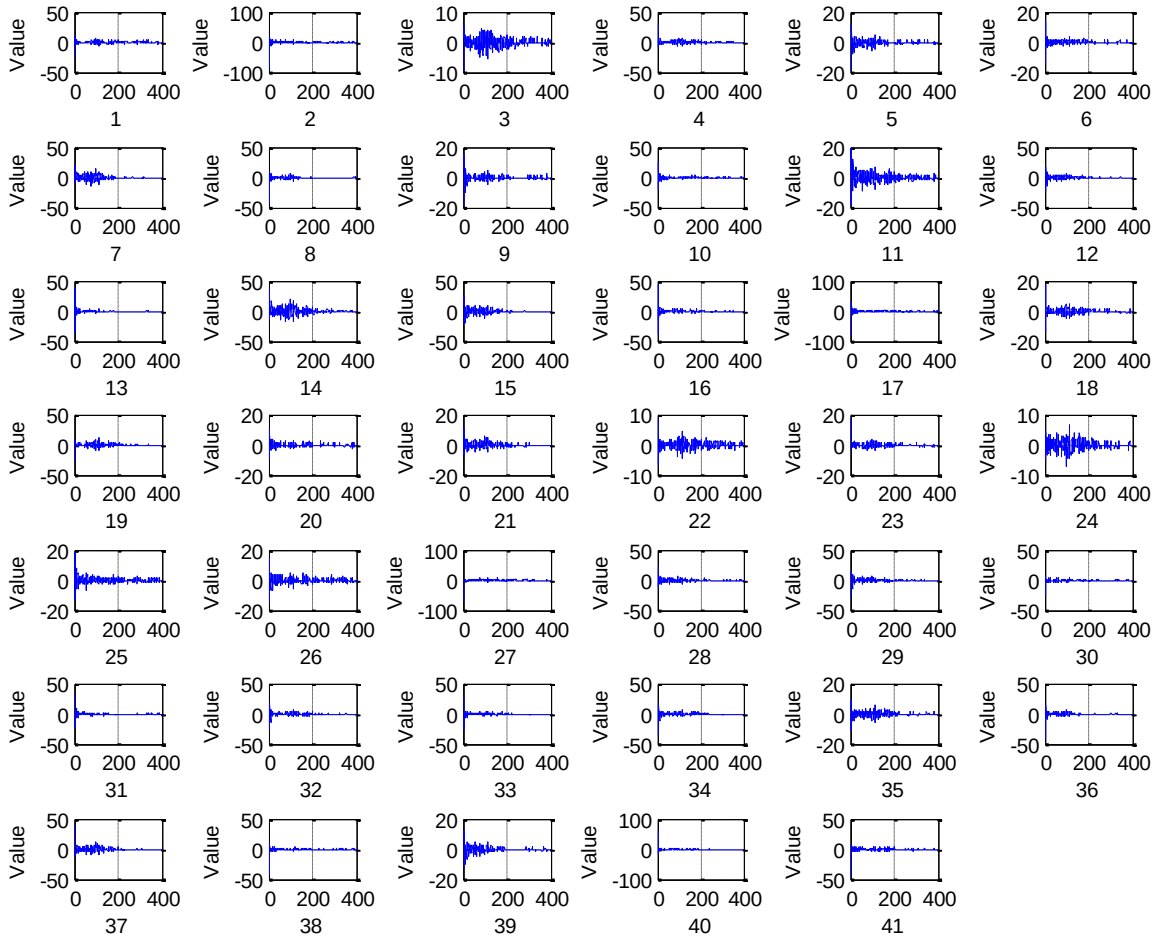

Figure 5.62. First 400 2D-DCT Coefficients of the Complete Output Signals of the TEP Data Set after Thresholding with α=99.5 %.

Semilog-log (upper sub-window of Figure 5.63) and log-log (bottom sub-window of Figure 5.63) plots of the sorted absolute values of the 2D-DCT coefficients of the overall TEP data set, which are padded into a vector of size 2050041 and the threshold limit of 0.879 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 5.63. The coefficients below the threshold limit (2039791 coefficients), which are identified clearly in the log-log plot, are set to zero for compression, in other words 99.5% of the transform coefficients become zero. The number of nonzero coefficients kept is 10250, which is only 0.5% of the number of data points. Although the threshold limit specified in 2D-DCT is smaller than that specified in 1D-DCT, which was 0.967 as mentioned in Section 5.3.7, the number of zeroed transform coefficients in 2D-DCT and 1D-DCT are same due to the fact that 2D-DCT coefficients are smaller than 1D-DCT coefficients.

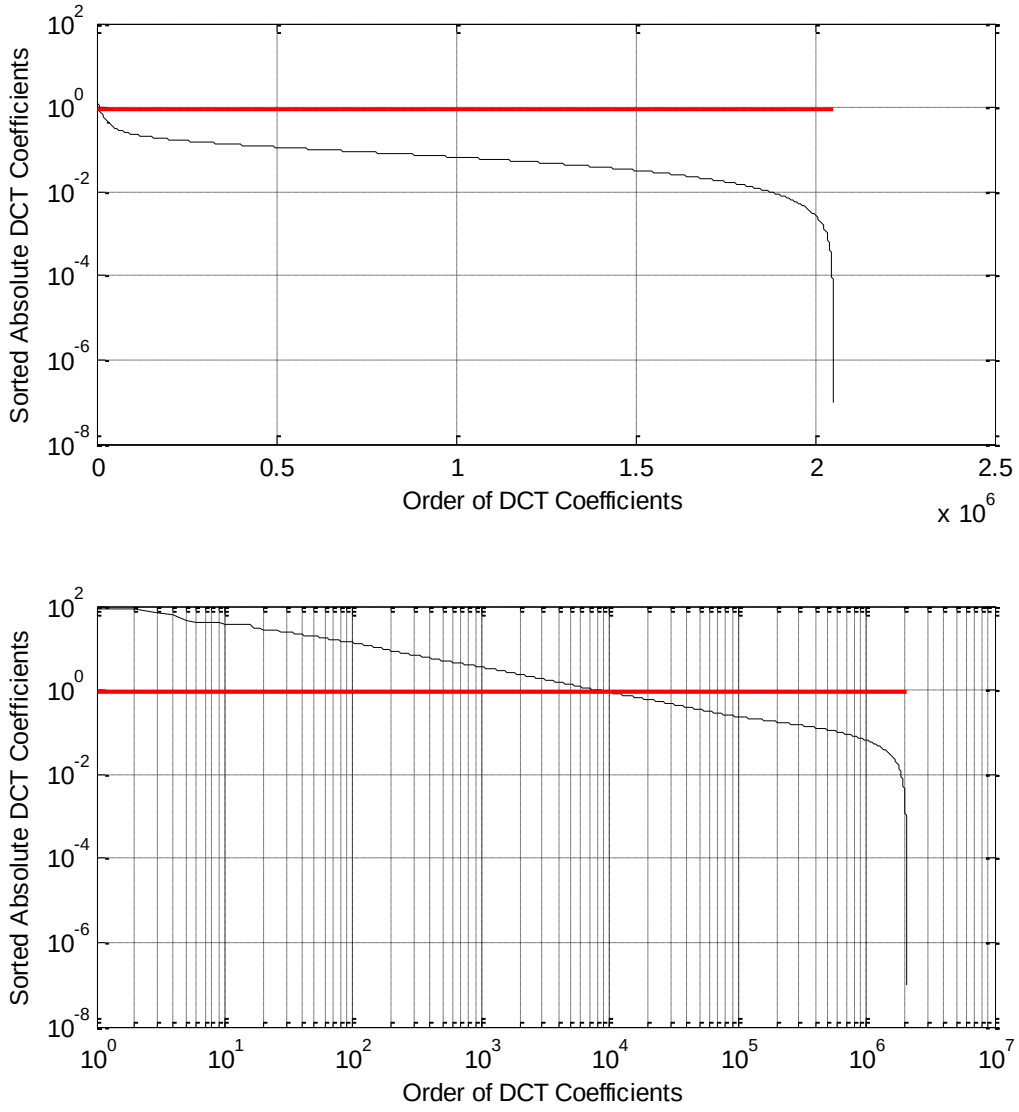

Figure 5.63. Semilog-log and Log-log Plots of Sorted Absolute 2D-DCT Coefficients of the Overall TEP Data Set for α=99.5 %.

Sizes of the ZIP files of the original and encoded data sets are compared as it is seen from Figure 5.64, the ZIP file of the original data set is nearly 21 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.38 MB. Thus, ZIP compression can be increased 54.5 times by applying 2D-DCT technique and taking the percentile value as 99.5% in the thresholding step. It can also be stated that compression ratio obtained in 1D-DCT is slightly larger than that obtained in 2D-DCT.

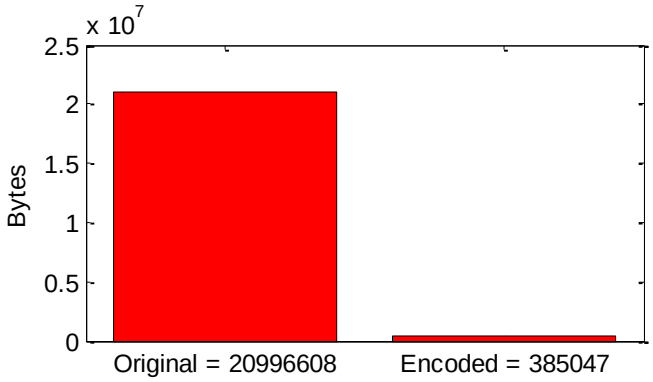

Figure 5.64. ZIP Compression Comparison of the Original and Encoded Overall TEP Data Set for α=99.5 % with 2D-DCT.

Reconstructed data generated with 2D-iDCT (inverse DCT) and their overlay with the originals are shown in Figure 5.65 and Figure 5.66 respectively. Like 1D-DCT, important features such as upward/downward shifts (observed in measurements one and four) and decreasing/increasing trends (observed in measurements 28 and 34) occurred due to disturbances are reconstructed perfectly as illustrated in Figure 5.65. However, a little distortion is observed in signal amplitudes (measurements 32, 37 and 40) unlike 1D-DCT. Thus, higher reconstruction error norms are expected. Noise removal is also achieved as it can be seen from Figure 5.66 that reconstructed values of the measurements consisting of almost pure noise (measurements nine and 19) are smoothened around a straight line.

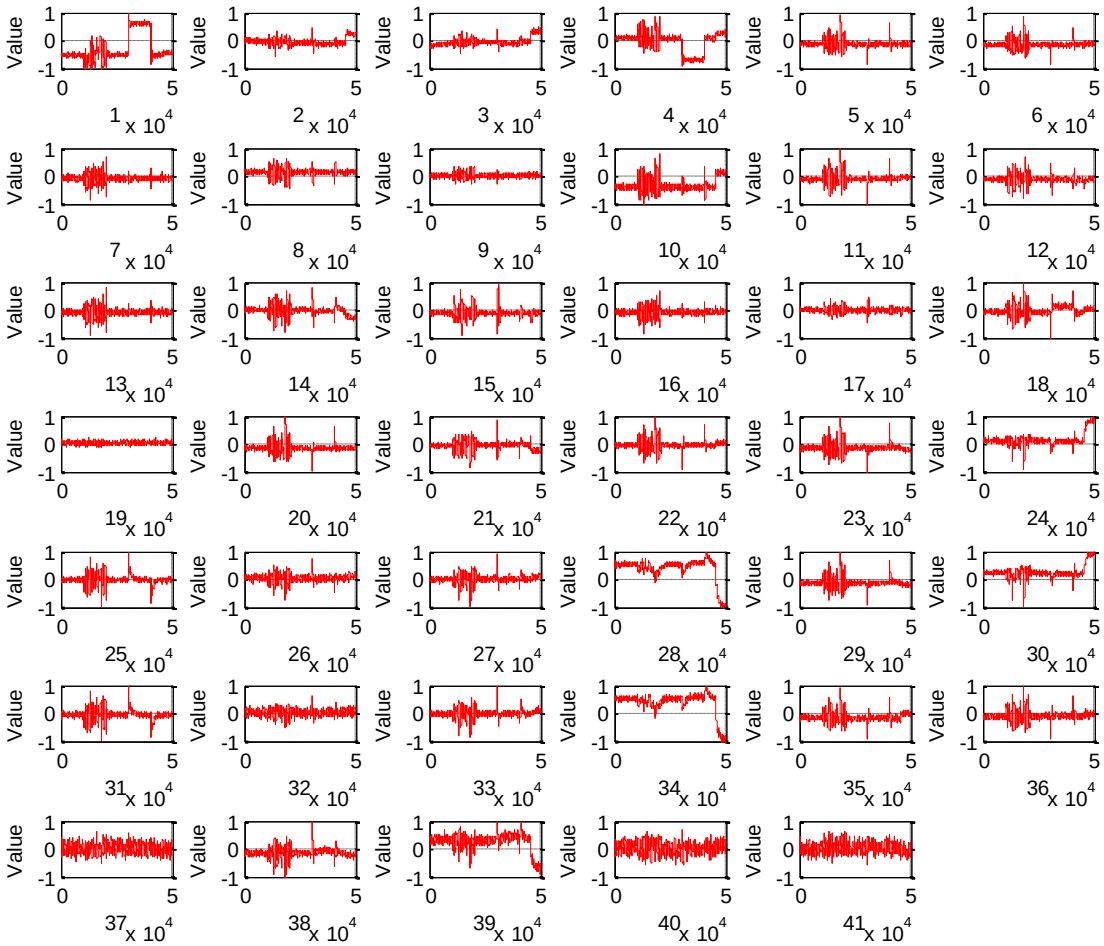

Figure 5.65. Reconstructed TEP Data Set with Inverse 2D-DCT.

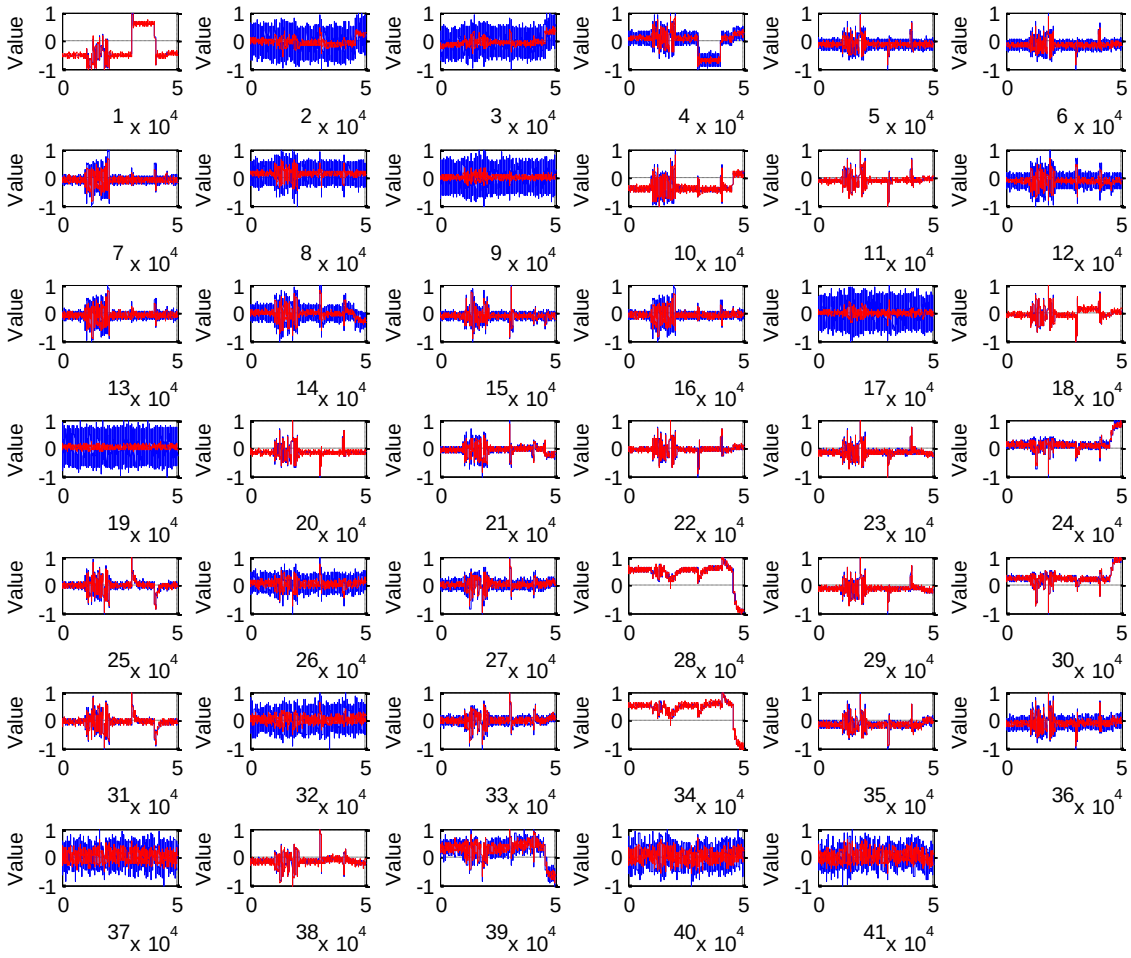

Figure 5.66. Original and Reconstructed Signals of the TEP Data Set with Inverse 2D-DCT.

In Figure 5.67, the original and reconstructed signals obtained with 2D-iDCT are plotted around the y=x line. Reconstructed data points around the y=x line indicates that original signals are reconstructed perfectly (measurements one, 34 and 38) whereas, reconstructed data points around the y=0 line shows that they are completely different from the original signals (measurements two, nine and 19).

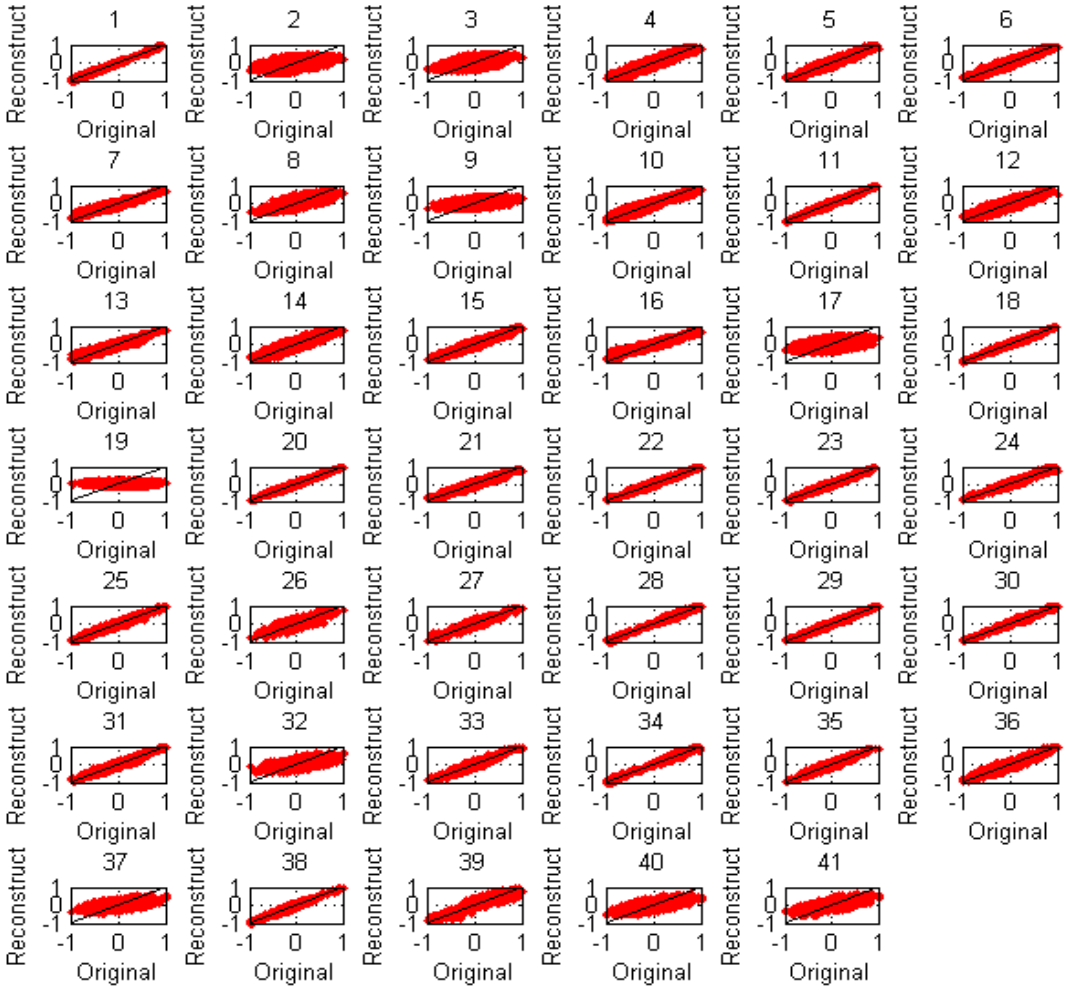

Figure 5.67. Reconstructed versus Original Signals of the TEP Data Set with Inverse 2D-DCT.

Reconstruction error norms between original and reconstructed data sets calculated per data column are given in the bottom sub-window of Figure 5.68. Minimum error norm of 1572.3 is obtained in the first column of the TEP data set, the measurement showing upward shift with less noise. Thus, reconstructed data points cluster around the y=x line stating that reconstructed signals are almost identical with the original signals as illustrated in upper sub-window of Figure 5.68. On the other hand, the maximum error norm of 9742.5 is obtained in the 19th column of the TEP data set (similar to 1D-DCT), measurement containing almost pure noise. Reconstructed data points are scattered around the y=0 line meaning that reconstructed signals are different from the original signals as shown in the middle sub-window of Figure 5.68. It can be said that the information loss is appropriate as irrelevant data are removed. It can be concluded that 1D-DCT should be the preferred technique when highly uncorrelated TEP data set is used as smaller

reconstruction error norm values and slightly higher compression ratios are generated with 1D-DCT.

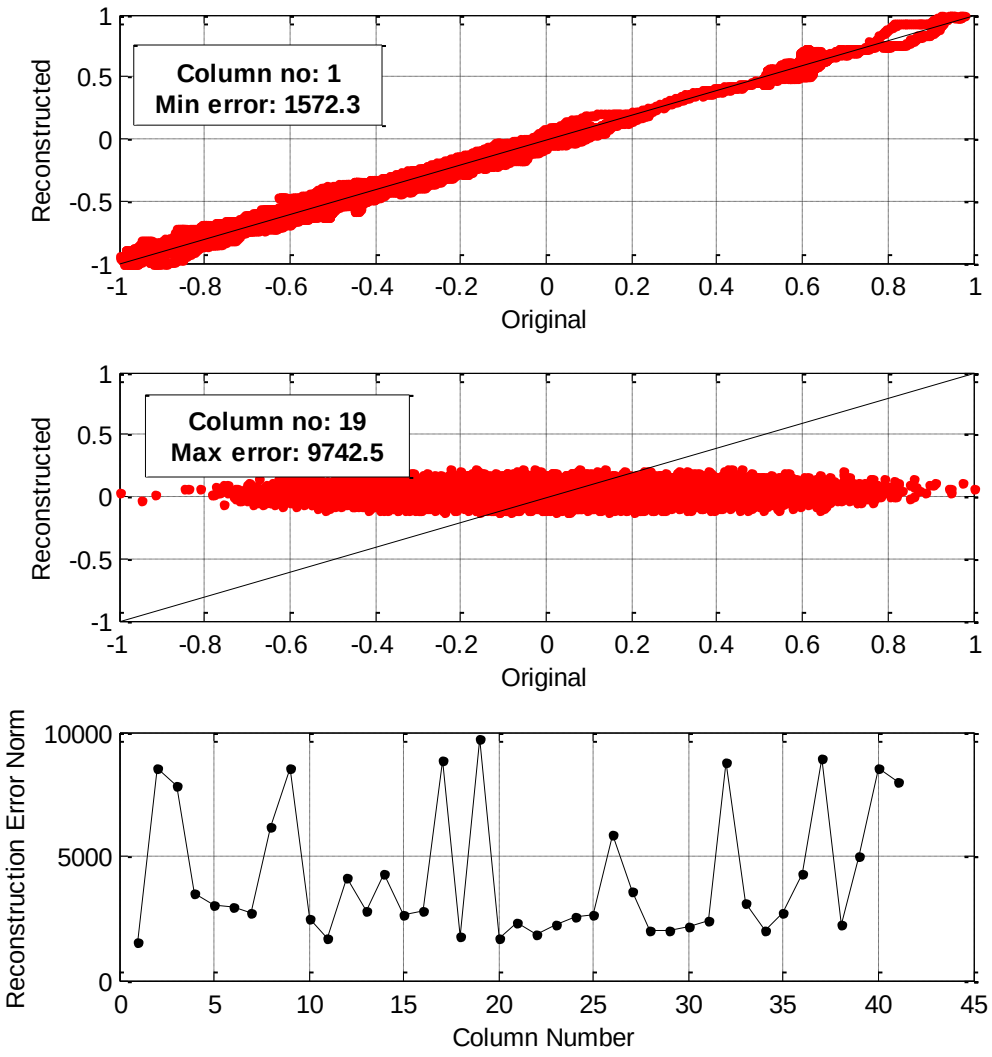

Figure 5.68. Reconstruction Error Norm Values of the TEP Data Set with Inverse 2D-DCT.

5.3.9. Comparison of One Dimensional and Two Dimensional Discrete Cosine Transform Methods

In this section, 1D-DCT and 2D-DCT techniques are compared by using the overall data sets (PortSimHigh, PortSimLow, SELDI-TOF MS and TEP) mentioned in Chapter 3 for 10 different percentile values of the frequency distribution of the transform coefficients in the [15%-99.8%] range. The effect of the percentile values used in thresholding step on compression is measured in terms of compression ratio, mean error norm, % relative global

error and % relative maximum error as shown in Figure 5.69. % Relative global error of a data set is calculated by first applying Equation 5.6 to each data column and then taking the mean of the results and % relative maximum error of a data set is calculated by dividing the maximum reconstruction error norm to the sum of the absolute values of each original data in the column which has the maximum error norm. In addition, ratios of compression ratio to mean error norm are computed to determine the optimum percentile level.

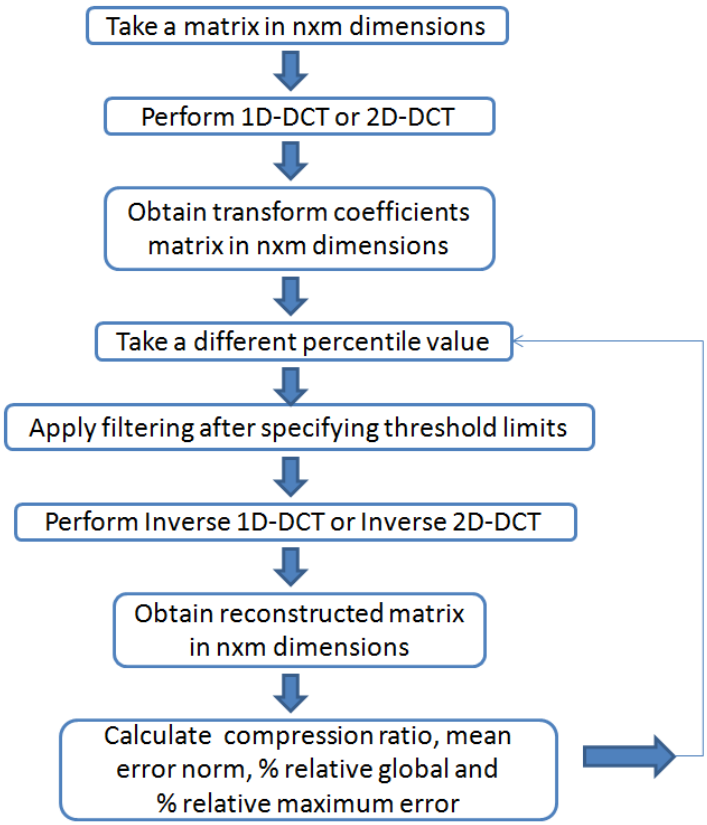

Figure 5.69. The Procedure used in Data Compression via DCT Technique

Measuring the Effect of the Percentile Value on Compression.

In addition, to be able to compare the DCT technique with the hybrid method consisting of PAA and quantization mentioned in Chapter 4, 1D-DCT and 2D-DCT methods are applied to the 50th, second and 30th columns of the PortSimHigh, SELDI-TOF MS and TEP data sets, respectively.

Compression ratios and mean error norms calculated for the overall PortSimHigh data set for 10 different percentile values are given in Figure 5.70. As the percentile value increases from 90% to 99.8%, the compression ratio increases almost seven times (approximately from 10 to 70) as shown in Figure 5.70. Thus it can be said that the maximum compression levels are obtained with the percentile values higher than 90% used in DCT methods. The important point is to obtain the maximum compression ratio while minimizing the reconstruction error norm. It can be said that compression ratios of the PortSimHigh data set obtained with 1D-DCT and 2D-DCT are the same. However, mean error norm values between reconstructed and original signals obtained with 1D-DCT are much higher than those obtained with 2D-DCT, especially for large percentile values used in thresholding step as illustrated in Figure 5.70. In addition, the value of the mean error norm obtained with 2D-DCT at the percentile level of 99.5% can be maintained with 1D-DCT at the percentile level of 60%. However, the achieved compression ratio becomes approximately one instead of 60.

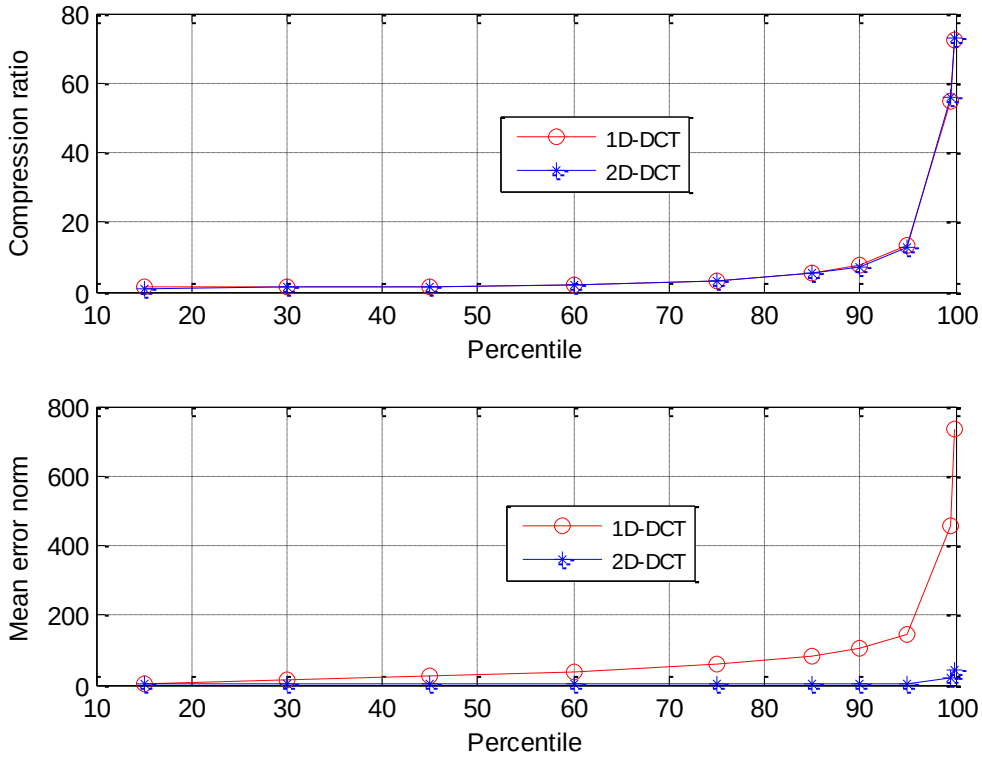

Figure 5.70. Compression Ratio and Mean Error Norm versus Thresholding Percentile for the PortSimHigh Data Set with DCT.

% Relative global and % relative maximum errors calculated for the overall PortSimHigh data set for 10 different percentile values are given in Figure 5.71. % Relative global and % relative maximum errors obtained with 1D-DCT are much larger than those obtained with 2D-DCT as it can be seen from Figure 5.71.

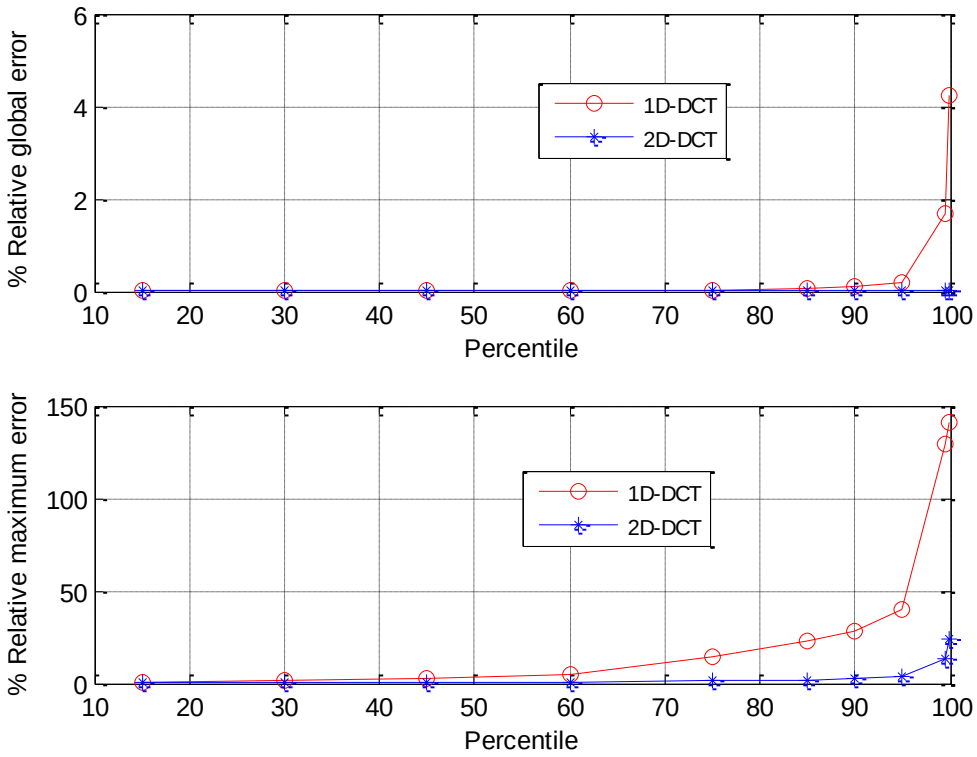

Figure 5.71. % Relative Global and % Relative Maximum Error versus Thresholding Percentile for the PortSimHigh Data Set with DCT.

Figure 5.72 is given to visually locate the optimum percentile value used in thresholding step. However, as the percentile values increase, the ratios of compression ratio to mean error norm decrease sharply for 2D-DCT, thus it is difficult to specify a reasonable optimum percentile value as high compression ratios are obtained with large percentiles. On the other hand, it can be concluded that 2D-DCT gives better results than 1D-DCT for the PortSimHigh data set. Also, the percentile value does not show a noticeable effect on the ratio of compression ratio to mean error norm for 1D-DCT as illustrated in Figure 5.72.

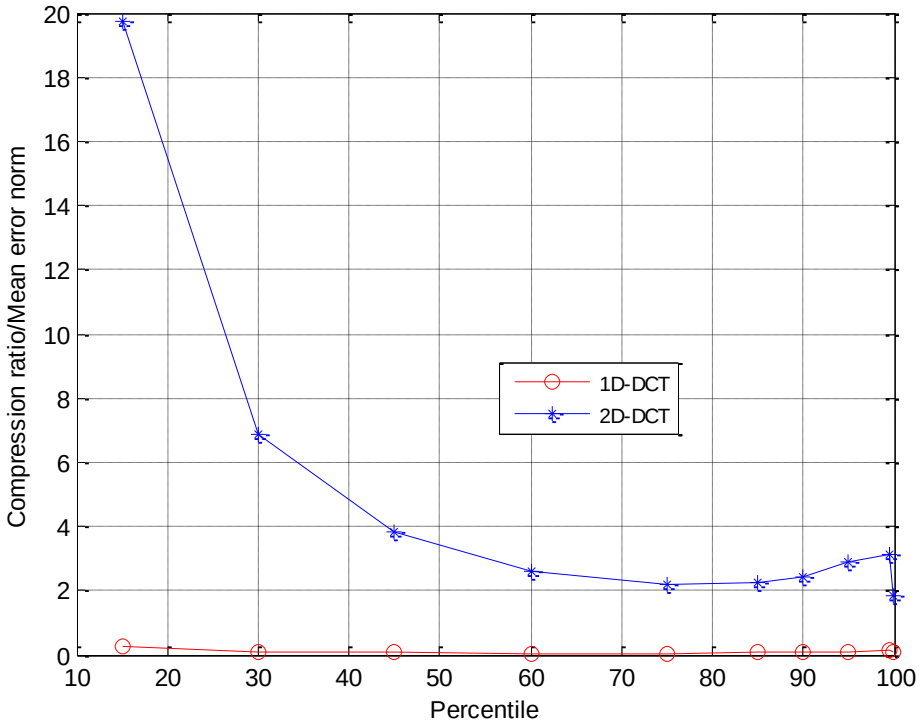

Figure 5.72. Compression Ratio/Mean Error Norm versus Thresholding Percentile for the PortSimHigh Data Set with DCT.

In Chapter 4, the optimum frame size used in the PAA and the optimum number of digits kept after decimal in quantization had been determined so as to maximize the ratio of compression ratio to error norm. The optimum frame size was 120 with one-digit quantization for the 50th column of the PortSimHigh data set, yielding compression ratio of 55 with the error norm close to 500, as shown in Figure 4.20. Same compression level is obtained with the percentile value of 99.5% with the same error norm by applying DCT technique. 1D-DCT and 2D-DCT results are the same when they are applied to single data column, as illustrated in Figure 5.73. Thus, it can be concluded that the hybrid method consisting of PAA and quantization applied in Chapter 4 is as effective as the DCT technique for the highly correlated PortSimHigh data set.

Compression ratios and mean error norms calculated for the overall PortSimLow data set for 10 different percentile values are given in Figure 5.74. Both compression ratio and error norm values obtained with 1D-DCT and 2D-DCT are almost same as shown in Figure 5.74. As far as the characteristics of the data sets mentioned in Chapter 3 are concerned, it can be concluded that 2D-DCT is the preferred method for the data sets with

high correlation among its columns, whereas 2D-DCT has not any superiority over 1D-DCT for the data sets having low correlation among its columns.

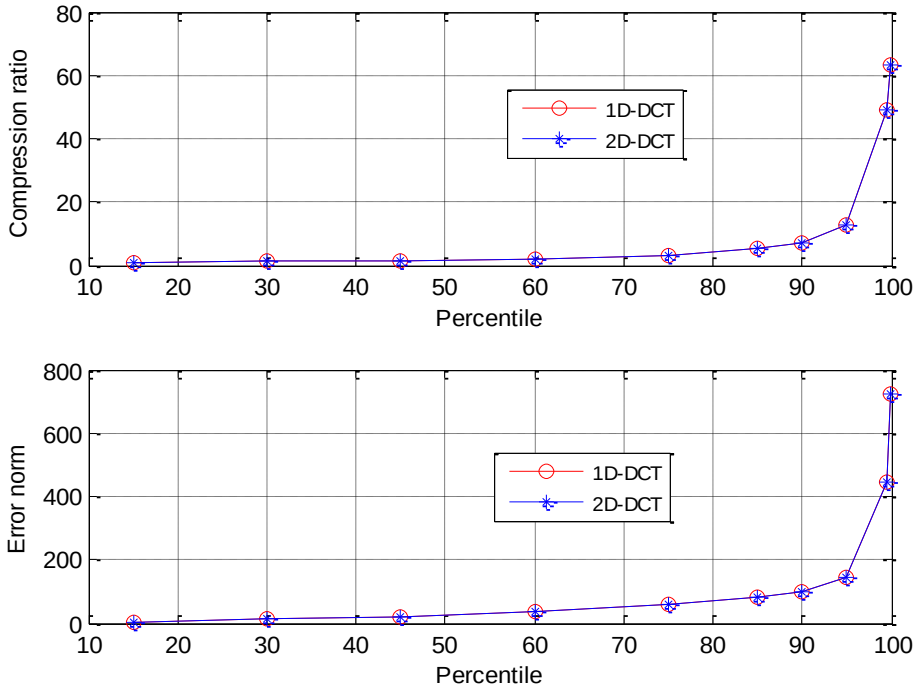

Figure 5.73. Compression Ratio and Error Norm versus Thresholding Percentile for the 50th column of the PortSimHigh Data Set with DCT.

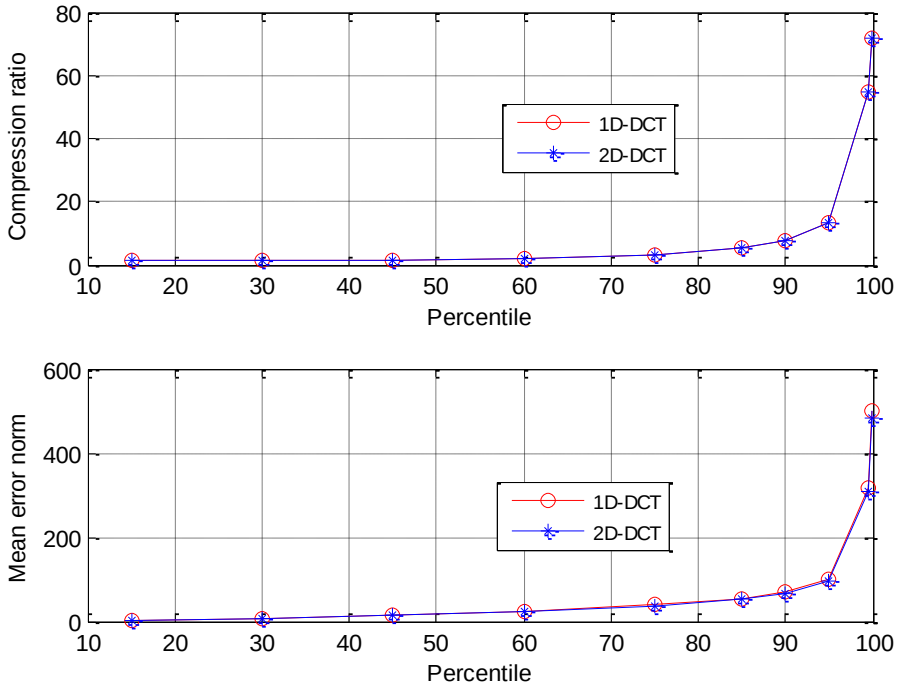

Figure 5.74. Compression Ratio and Mean Error Norm versus Thresholding Percentile for the PortSimLow Data Set with DCT.

% Relative global and % relative maximum errors calculated for the overall PortSimLow data set for 10 different percentile values are given in Figure 5.75. % Relative global errors obtained with 1D-DCT and 2D-DCT are nearly the same, whereas % relative maximum error obtained with 2D-DCT is slightly larger than that obtained with 1D-DCT as it can be seen from Figure 5.75. Thus, it can be stated that localized measure of error is more distinctive than the overall measure.

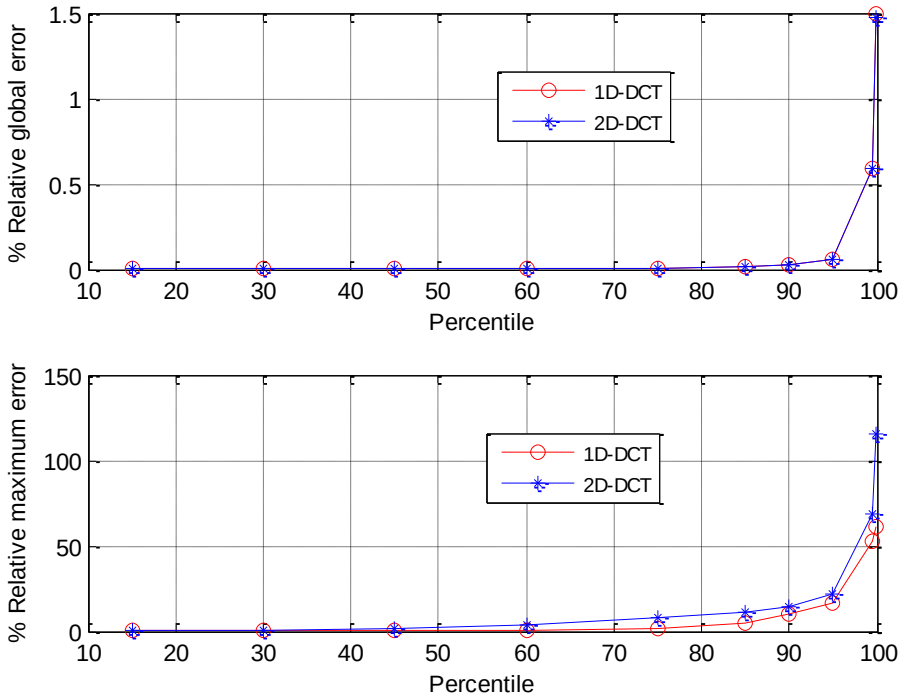

Figure 5.75. % Relative Global and % Relative Maximum Error versus Thresholding Percentile for the PortSimLow Data Set with DCT.

The ratio of compression ratio to error norm values computed for the overall PortSimLow data set for 10 different percentile values are given in Figure 5.76 to determine the optimum percentile level. The ratio of compression ratio to error norm values calculated with 1D-DCT and 2D-DCT methods are almost overlapped as illustrated in Figure 5.76. Optimum percentile value cannot be determined since the ratios decrease as the percentile value increases. The curves are similar to the 2D-DCT curve of the overall PortSimHigh data set illustrated in Figure 5.72. Whereas, the ratio of compression ratio to error norm values for 2D-DCT applied to the PortSimHigh data set are much larger than those for both DCT techniques applied to the PortSimLow data set. As a result, it can be

stated that 2D-DCT is a very efficient method for highly correlated data sets, whereas 2D-DCT may not be the appropriate compression technique for the less correlated data sets.

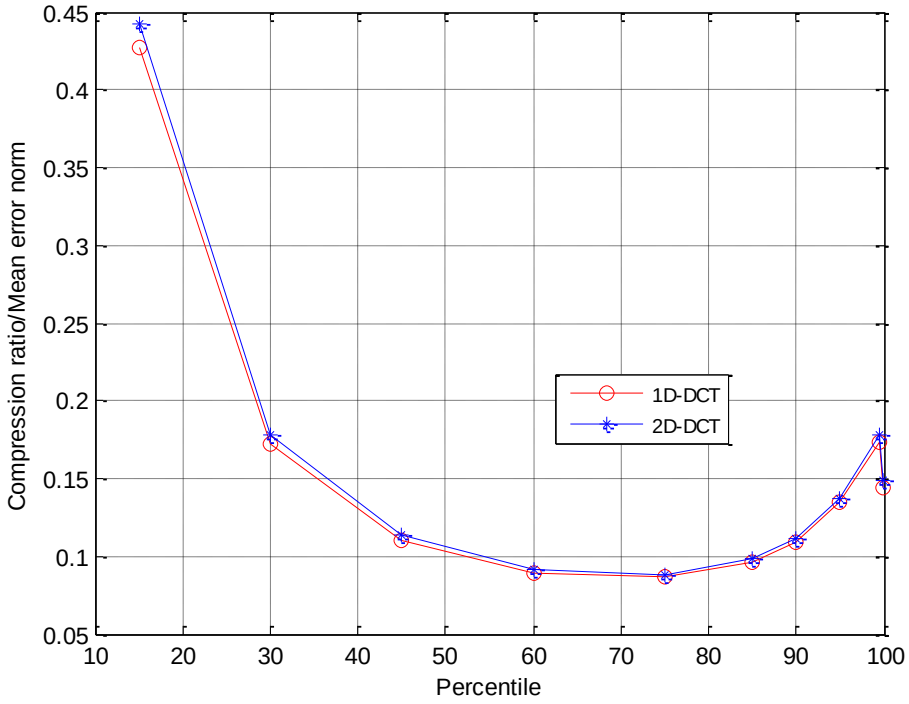

Figure 5.76. Compression Ratio/Mean Error Norm versus Thresholding Percentile for the PortSimLow Data Set with DCT.

Compression ratios and mean error norms calculated for the overall SELDI-TOF MS data set for 10 different percentile values are given in Figure 5.77. 1D-DCT and 2D-DCT methods give the same results for the SELDI-TOF MS data set as shown in Figure 5.77 similar to the PortSimLow data set. As the SELDI-TOF MS data set is highly uncorrelated, it is expected that 2D-DCT method will not give better results than 1D-DCT as mentioned before.

% Relative global and % relative maximum errors calculated for the overall SELDI-TOF MS data set for 10 different percentile values are given in Figure 5.78. % Relative global and % relative maximum errors are too small as illustrated in Figure 5.78 when compared with large mean error norms shown in Figure 5.77. This situation may be related to the data having sharp peaks with large amplitudes. In addition, % relative maximum error obtained with 2D-DCT is slightly larger than that obtained with 1D-DCT. According to the high mean error norms, it is expected that there will be a significant distortion in the

reconstructed SELDI-TOF MS data set (especially the regions where sharp peaks are observed) as mentioned in Sections 5.3.5 and 5.3.6.

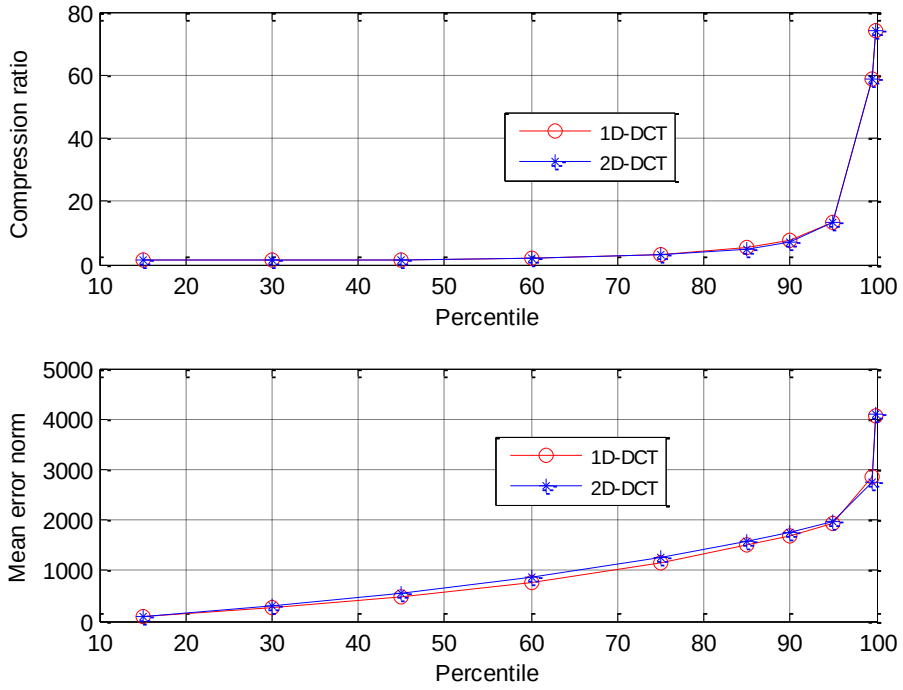

Figure 5.77. Compression Ratio and Mean Error Norm versus Thresholding Percentile for the SELDI-TOF MS Data Set with DCT.

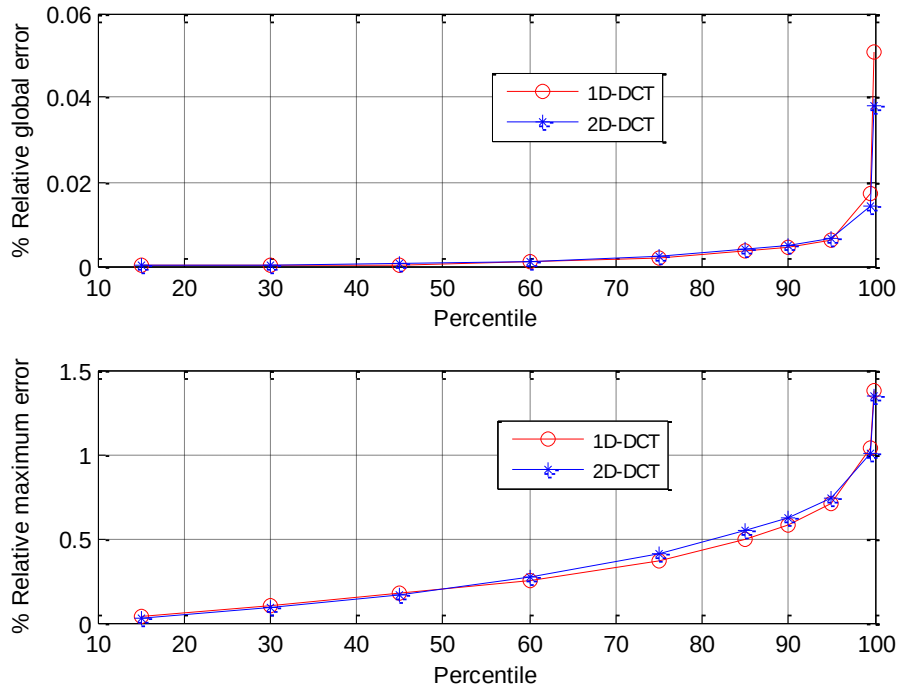

Figure 5.78. % Relative Global and % Relative Maximum Error versus Thresholding Percentile for the SELDI-TOF MS Data Set with DCT.

The ratio of compression ratio to error norm values computed for the overall SELDI-TOF MS data set for 10 different percentile values are given in Figure 5.79. Unlike PortSimHigh and PortSimLow data sets, the optimum percentile value can be specified as 99.5% (the isolated maximum) for the SELDI-TOF MS data set as illustrated in Figure 5.79. The reason may be related with the sparse sharp peaks in SELDI-TOF MS data set as mentioned in Chapter 3. In addition, it can be said that 1D-DCT generally gives slightly better results than 2D-DCT for the highly uncorrelated SELDI-TOF MS data set including baseline noise. However, for the percentile level 99.5%, 2D-DCT is the preferred technique as mentioned in Section 5.3.6, revealing that the gap between these two techniques narrows as the percentile values increase.

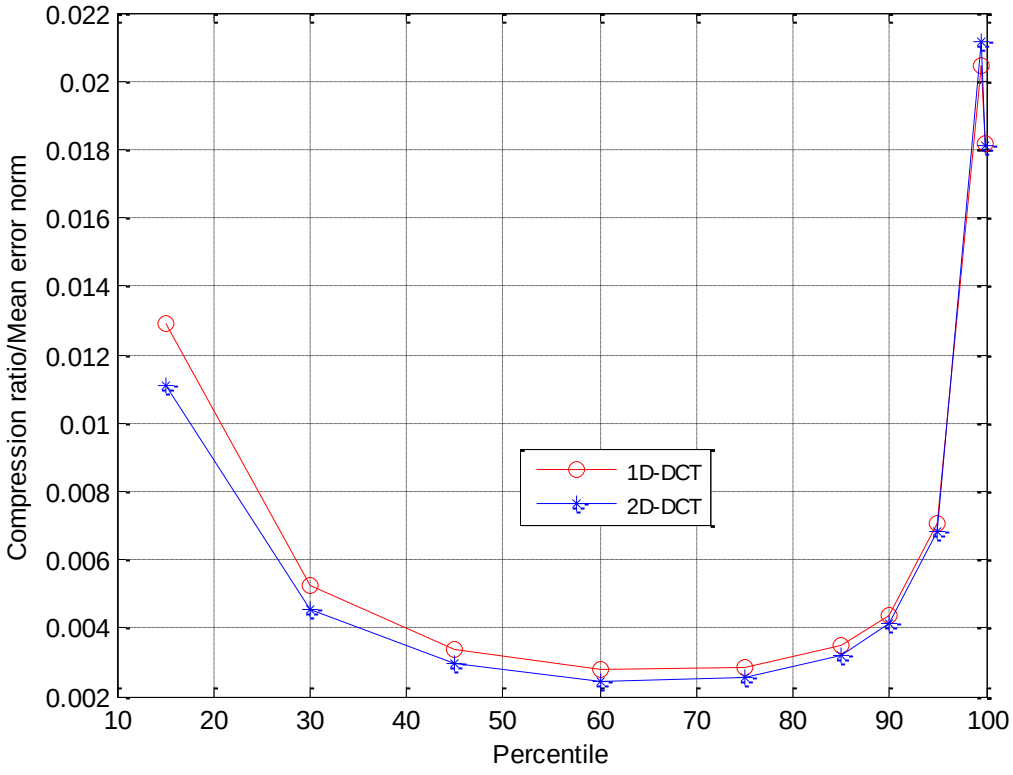

Figure 5.79. Compression Ratio/Mean Error Norm versus Thresholding Percentile for the SELDI-TOF MS Data Set with DCT.

In Chapter 4, the optimum frame size used in the PAA and the optimum number of digits kept after decimal in quantization had been determined as 150 frames and three-digit quantization for the second column of the SELDI-TOF MS data set respectively, yielding compression ratio as 9.5 with the error norm close to 4000 as shown in Figure 4.21. However, by applying DCT technique, compression ratio can be obtained as 70 with the

same error norm at a percentile level of 99.8% as shown in Figure 5.80. Thus, it can be concluded that DCT technique gives superior results, increasing the compression ratio nearly seven times, than the hybrid method consisting of PAA and quantization as applied in Chapter 4 for the highly uncorrelated SELDI-TOF MS data set consisting of sparse peaks.

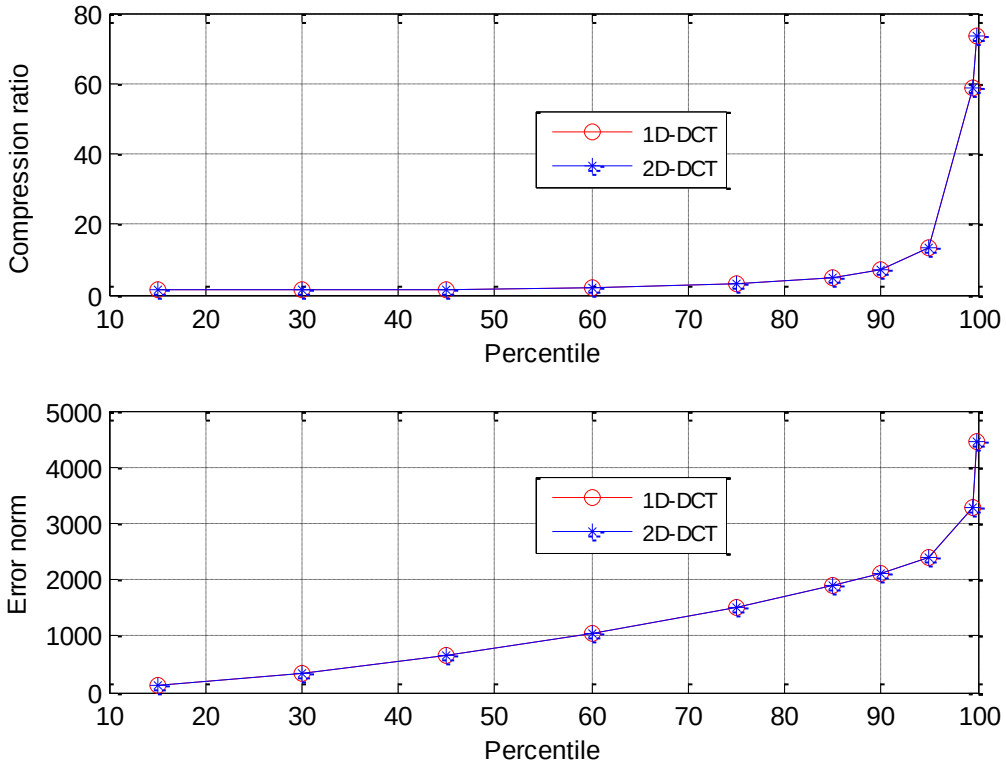

Figure 5.80. Compression Ratio and Error Norm versus Thresholding Percentile for the Second Column of SELDI-TOF MS Data Set with DCT.

Compression ratios and mean error norms calculated for the overall TEP data set for 10 different percentile values are given in Figure 5.81. Unlike PortSimHigh data set, mean error norm values obtained with 1D-DCT are much smaller than those obtained with 2D-DCT for the TEP data set, although these two techniques give almost the same compression ratios as shown in Figure 5.81. However, there is not much difference between mean error norms obtained at percentile levels higher than 95%. In addition, the value of the mean error norm obtained with 1D-DCT at the percentile level of 95% can be maintained with 2D-DCT at the percentile level of 75%. However, the achieved compression ratio becomes approximately five instead of 15.

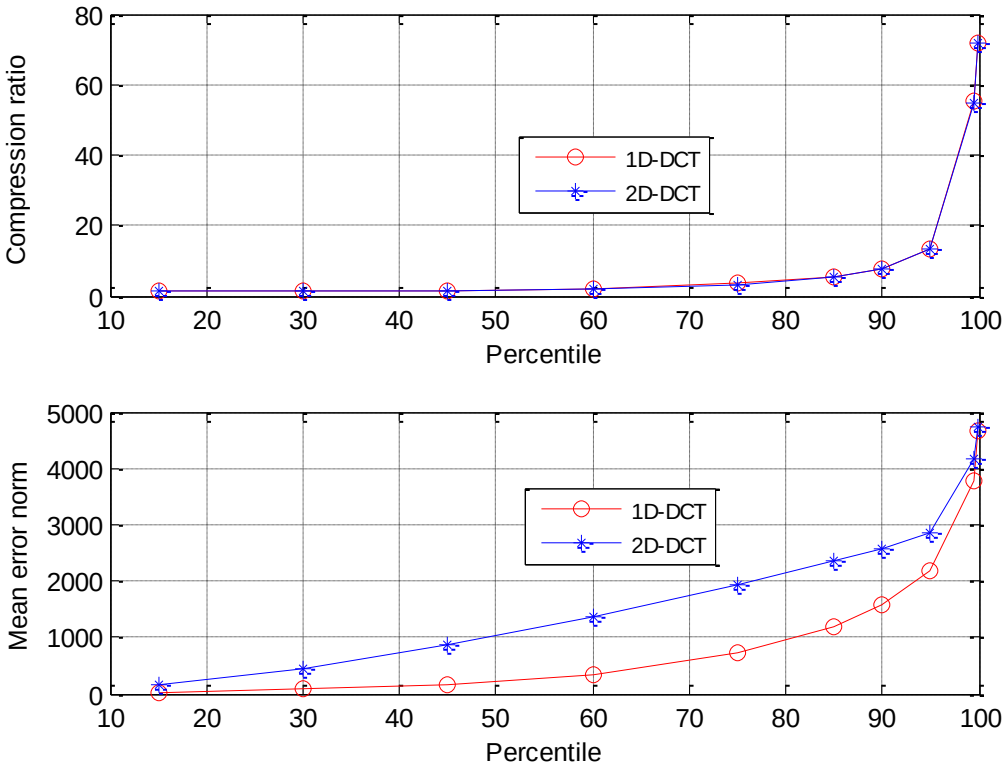

Figure 5.81. Compression Ratio and Mean Error Norm versus Thresholding Percentile for the TEP Data Set with DCT.

% Relative global and % relative maximum errors calculated for the overall TEP data set for 10 different percentile values are given in Figure 5.82. % Relative global and % relative maximum errors obtained with 2D-DCT are larger than those obtained with 1D-DCT as illustrated in Figure 5.82. Besides, maximum % relative errors obtained with the TEP data set are the largest of the four data sets, consistent with the largest mean error norms. The reason is that TEP data set is highly uncorrelated including too much noisy measurements with level jumps. Compression of this type of data sets is very difficult, thus the efficacy of the DCT technique decreases.

The ratio of compression ratio to error norm values computed for the overall TEP data set for 10 different percentile values are given in Figure 5.83. It is seen that 1D-DCT should be the preferred technique for the TEP data set as the ratios of the compression ratio to error norm are higher than those obtained with 2D-DCT as shown in Figure 5.83. However, the gap between these two techniques narrows as the percentile values increase.

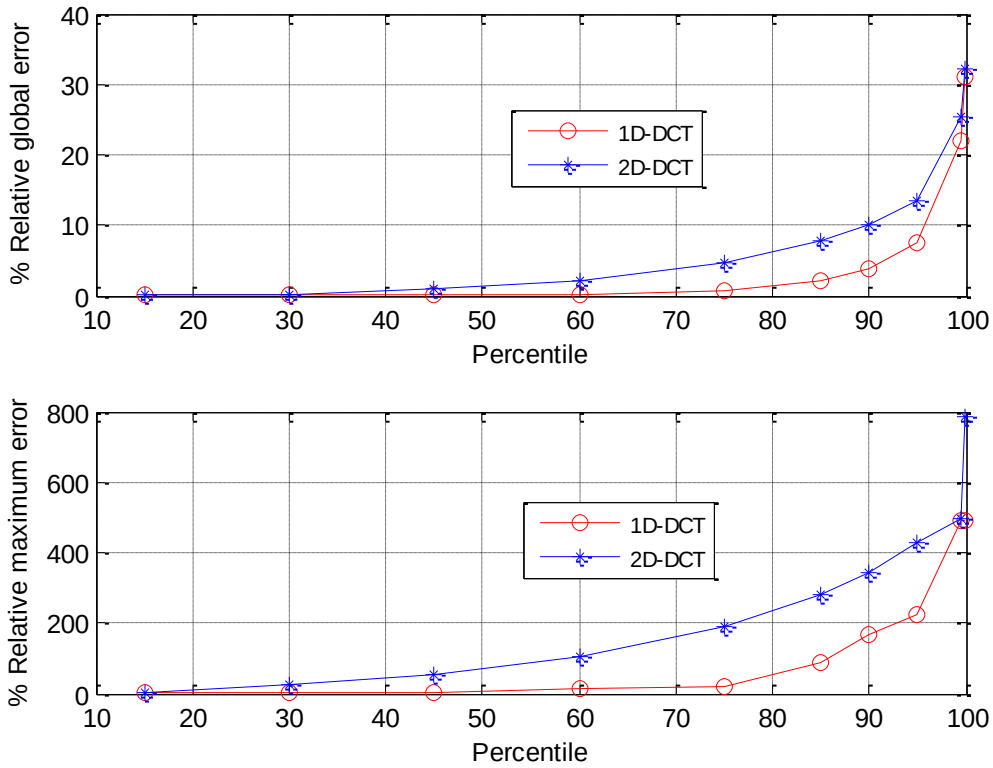

Figure 5.82. % Relative Global and % Relative Maximum Error versus Thresholding Percentile for the TEP Data Set with DCT.

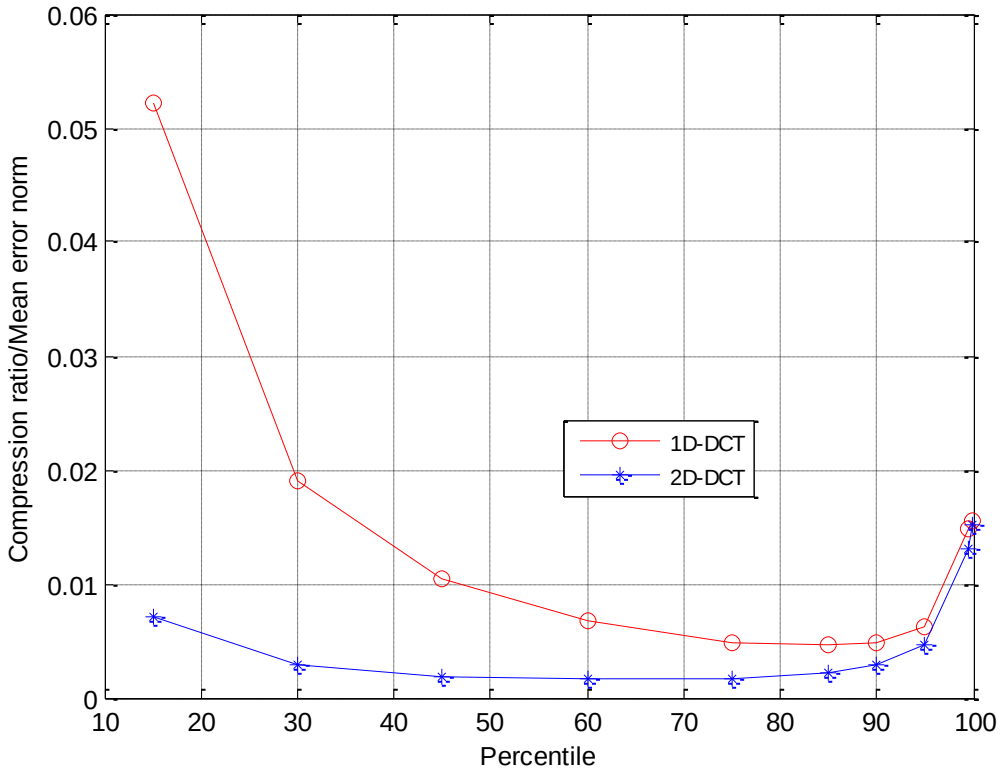

Figure 5.83. Compression Ratio/Mean Error Norm versus Thresholding Percentile for the TEP Data Set with DCT.

In Chapter 4, the optimum frame size used in the PAA and the optimum number of digits kept after decimal in quantization had been determined as 150 frames and one-digit quantization for the 30th column of the TEP data set respectively, yielding compression ratio as 12 with the error norm close to 3000 as shown in Figure 4.22. However, by applying DCT technique, compression ratio can be obtained as 70 with the same error norm at the percentile level of 99.8% as illustrated in Figure 5.84. Thus, it can be concluded that DCT technique gives superior results, increasing the compression ratio nearly six times, than the hybrid method consisting of PAA and quantization for the highly uncorrelated TEP data set which consists of trendy signals with almost pure noise and level jumps.

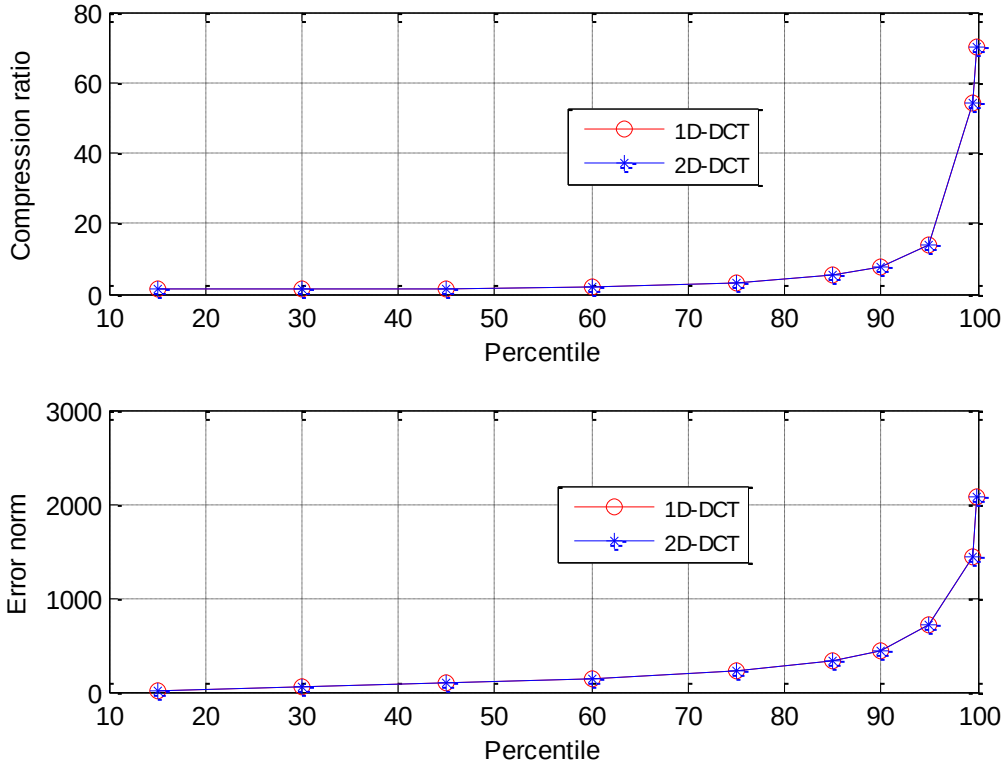

Figure 5.84. Compression Ratio and Error Norm versus Thresholding Percentile for the 30th column of the TEP Data Set with DCT.

To sum up, DCT is the easiest data compression method achieving high compression by performing reversible mapping from time to frequency domain while exhibiting excellent decorrelation and energy compaction properties. The important process features such as upward/downward shifts and decreasing/increasing trends except sharp peaks can be reconstructed thoroughly by IDCT without any significant distortion in information

content. Measurements consisting of almost pure noise do not contain information, and noise removal is well achieved by DCT, besides providing better compression.

2D-DCT technique is the preferred technique for the highly correlated data sets. It can also be stated that for the data sets consisting of almost pure noise with level jumps, higher compression levels can be obtained with the DCT method instead of the hybrid method consisting of PAA and quantization mentioned in Chapter 4. Nevertheless, 2D-DCT may not be the appropriate compression technique for the highly uncorrelated data sets as the efficacy of this method decreases a lot.

As the percentile values used in thresholding step increase, compression ratio, mean error norm, % relative global error and % relative maximum error values increase steadily. Mean error norms calculated for the TEP and SELDI-TOF MS data sets are much higher (nearly five times) than those of the PortSimHigh and PortSimLow data sets for the same compression level of 80 due to their high noise content with sudden changes. Thus, it is more difficult to reconstruct these two data sets by the DCT without any significant loss. However, % relative global and % relative maximum errors calculated for the SELDI-TOF MS data set are much smaller than those of the three data sets for the compression level of 80. This situation may be explained with the presence of sharp peaks (signals with large amplitudes) in SELDI-TOF MS data set.

As the percentile values used in thresholding step increase, both compression ratio and mean error norm values increase steadily. However, high compression levels cannot be achieved with percentile values less than 90% in DCT method. High compression ratios can be obtained by further filtering the transform coefficients in thresholding step. The amount of distortion and compression can be adjusted via proper setting of the threshold limits as explained in Chapter 2. The important point is to adjust these limits beyond which amount of distortion cannot be identified in filtering.

The Discrete Wavelet Transform (DWT) will be studied in the following chapter in order to reduce reconstruction error norms further while maximizing reduction ratios.

6. DATA COMPRESSION VIA DISCRETE WAVELET TRANSFORM

Discrete Wavelet Transform (DWT) is generally used in signal processing to remove undesirable noisy data which are short-lived high frequency signals (Blelloch, 2010). DWT is gaining popularity in various applications such as on-line data compression, data rectification, pattern-matching and image processing. DWT analyzes a signal both in time and frequency domain by multi-scale representation whereas DCT works only in frequency domain. Hence, DWT can be used to identify the signal frequencies in a specific time interval, for instance DWT can detect the instant of a localized change in a signal whereas DCT cannot (Salomon, 2008). Thus, DWT is superior to DCT in analyzing non-stationary signals containing spikes or discontinuities (Pu, 2006).

DWT is useful in revealing signal trends hidden in noisy measurements by dividing the original data set into different frequency components which are called “sub-bands”. Filters allowing certain sub-bands to pass are called “band-pass filters”, for instance, a low-pass filter (LPF) allows the low-frequency components (approximations), whereas a high-pass filter (HPF) allows the high-frequency components (details) as illustrated in Figure 6.1 (Pu, 2006).

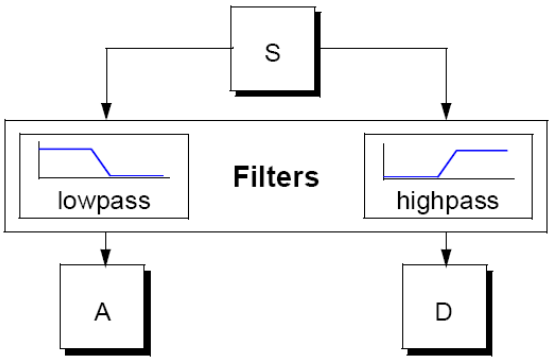

Figure 6.1. One-Stage Filtering of a Signal (Mathworks, 2011).

If the original signal (S) consists of 1000 data points, at the end of the filtering process, each resulting signal (A and D) will have 1000 data points, a total of 2000 data points are produced, in other words the number of data points is doubled. Hence, downsampling is applied to these signal components producing approximation coefficients

(cA) and detail coefficients (cD) each having 500 values, namely a total of 1000 data points will be produced as shown in Figure 6.2.

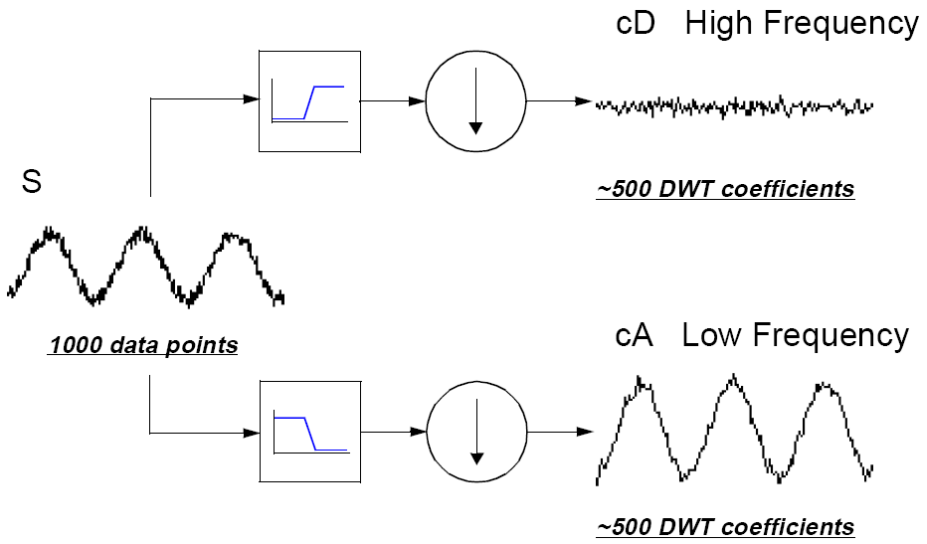

Figure 6.2. Filtering and Downsampling of a Signal Producing DWT Coefficients (Mathworks, 2011).

The scheme of the wavelet transform including decomposition and reconstruction filters is shown in Figure 6.3. The choice of reconstruction filters is important for perfect reconstruction of the original signal from the approximation and detail coefficients (Mathworks, 2011).

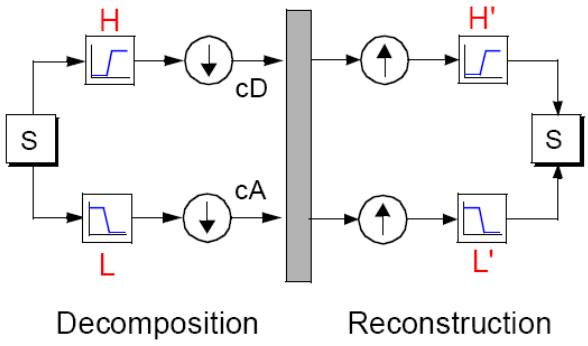

Figure 6.3. Decomposition and Reconstruction Filters (Mathworks, 2011).

Multilevel wavelet decomposition tree is presented in Figure 6.4 and Figure 6.5 for the original signal S, where cA's are the approximation coefficients and cD's are the detail coefficients. Approximation coefficients are the coarse representation and detail

coefficients are the small representation of the original data set. Multilevel decomposition is generally preferred to get rid of noisy measurements since the first levels of decomposition eliminate noise. As the level used in decomposition increases, the number of noisy data eliminated also increases. Thus, smaller reconstruction error norms with higher compression ratios are expected.

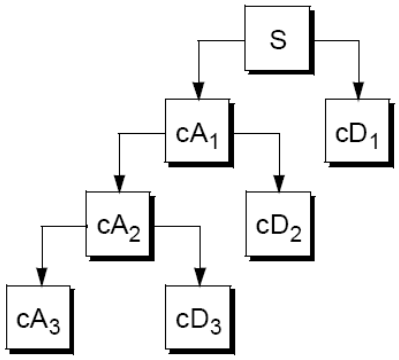

Figure 6.4. Multilevel Wavelet Decomposition Tree (Mathworks, 2011).

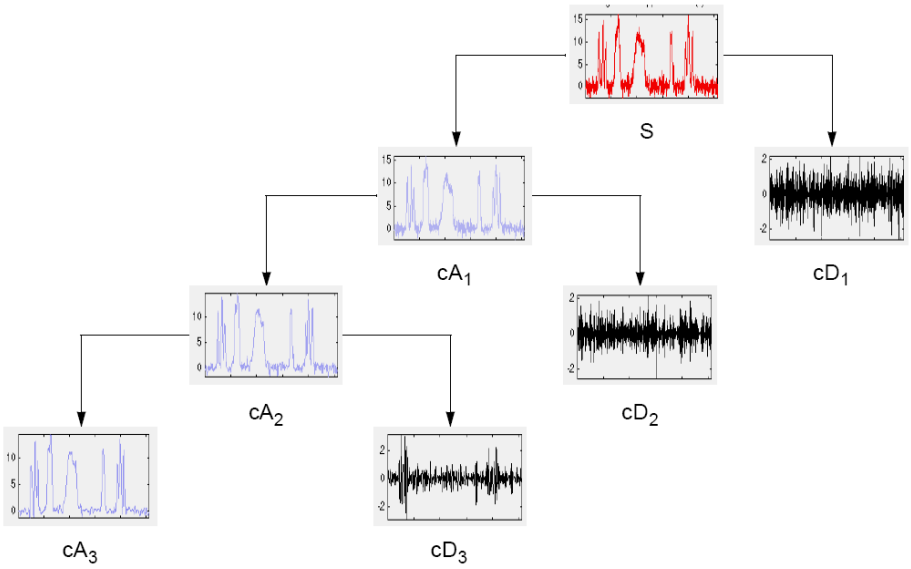

Figure 6.5. Detailed Multilevel Wavelet Decomposition Tree (Mathworks, 2011).

With regard to Figure 6.4 and Figure 6.5, in a three-level wavelet decomposition, original signal (S) is the sum of the approximation at level three (A_3) and the details at levels three, two and one (D_3, D_2, D_1), as illustrated in Figure 6.6. Approximation and detail components are firstly reconstructed from the wavelet coefficients by applying the inverse wavelet transform before they are assembled to reproduce the original signal.

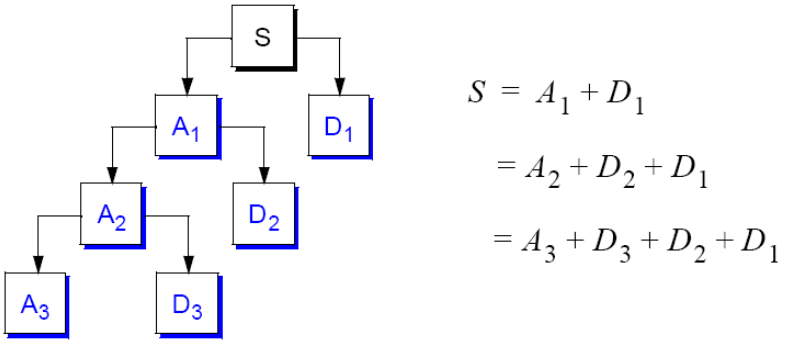

Figure 6.6. Reconstructed Signal Components (Mathworks, 2011).

In MATLAB, the third-level approximation coefficients (cA_3) and the first three levels of detail coefficients (cD_1, cD_2, cD_3) produced in the third-level wavelet decomposition are assembled into one vector, C as illustrated in Figure 6.7.

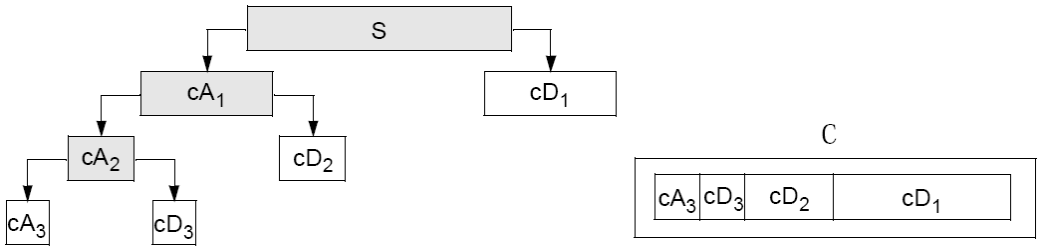

Figure 6.7. Third-Level Decomposition Coefficients (Mathworks, 2011).

Length of the signal's wavelet coefficients decreases by half in each decomposition level, where approximation and detail coefficients are produced. In MATLAB, these coefficients are assembled into the 'coefs' vector and lengths of the coefficients and the 'coefs' vector are stored in the 'longs' vector as illustrated in Figure 6.8.

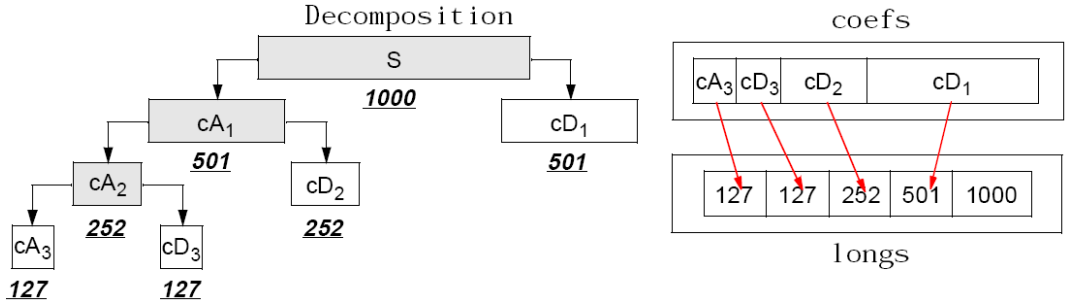

Figure 6.8. Third-Level Decomposition Coefficients and Their Lengths (Mathworks, 2011).

In DWT filtering/de-noising, the signal is first decomposed into detail coefficients (differences) representing the high-frequency components including noisy measurements and approximation coefficients (averages) representing the low-frequency components at multiple levels of resolution. Then, the detail coefficients below certain threshold are set to zero. Thus, noise removal is achieved by quantizing the differences during compression (Bakshi and Stephanopoulos, 1996). These approximation and detail coefficients are encoded separately. Finally, the original signal is reconstructed by applying the inverse wavelet transform over both the detail and approximation coefficients before combining them. Information loss will be more towards the high-frequency region as most of the detail coefficients representing this region are eliminated.

6.1. One Dimensional Discrete Wavelet Transform

DWT does not have a single set of basis functions, unlike DCT. The family of basis functions are scaled and translated versions of a mother wavelet function such as the Haar, the Daubechies family and the Symlet family of orthogonal wavelet functions (Blelloch, 2010). The efficacy of the DWT depends directly on the preferred mother wavelet function (Bakshi and Stephanopoulos, 1996).

Wavelets are irregular and often non-symmetrical having a limited duration. They are suited to localized changes (Watson *et al.*, 1998). Wavelets can detect overall trend of a signal. However, detecting discontinuity is difficult in presence of noise.

Haar wavelets are the most described wavelets, but the least used although their simple form resembling a step function is easy to illustrate, which are derived from the following mother function, ψ(t) (Singhal and Seborg, 2005);

$$\psi(t) = \begin{cases} 1: & 0 \leq t < 1/2 \\ -1: & 1/2 \leq t < 1 \\ 0: & \text{otherwise} \end{cases} \tag{6.1}$$

Daubechies and Symlet family wavelets are denoted by dbN and symN respectively where N is the order. Haar, Daubechies 4 (db4) and Symlet 4 (sym4) wavelets are given in Figure 6.9. Haar wavelet is discontinuous representing the same wavelet as Daubechies 1

(db1) and generally used for smooth data sets whereas Daubechies and Symlet wavelets are generally preferred for noisy data sets (Benouaret *et al.*, 2012).

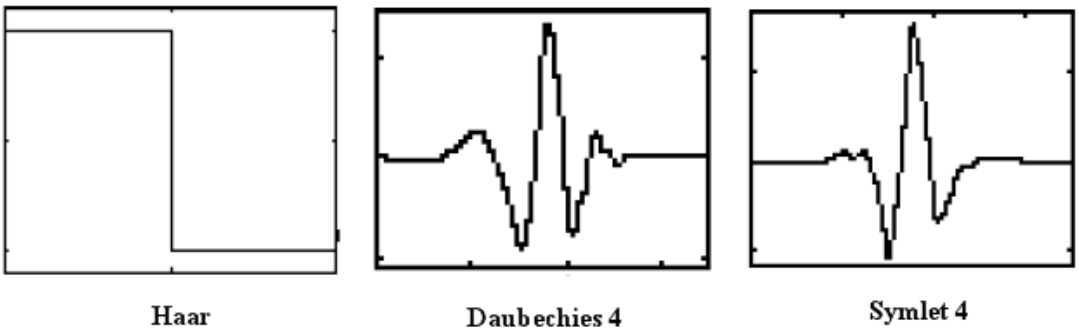

Figure 6.9. Examples of Types of Wavelets (Mathworks, 2011).

The integral wavelet transform of a time-varying signal f(t) is defined as follows (Watson *et al.*, 1998);

$$f'(a,b) = |a|^{-1/2} \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-b}{a}\right) dt \tag{6.2}$$

where a denotes the frequency scale (dilation) at which the signal is decomposed and b denotes its position in time (translation). The wavelet transform f'(a,b) is calculated at the position $b = k/2^j$ and with dilation $a = 2^{-j}$ where j and k are integers. If these dilate and translate components are orthonormal, wavelet function can be represented as follows (Watson *et al.*, 1998);

$$\psi_{j,k}(t) = 2^{j/2}\psi\left(2^j t - k\right) \tag{6.3}$$

The number of wavelet coefficients that are close to zero can be increased by choosing the most appropriate wavelet function ψ(t) to obtain better compression. Furthermore, if a wavelet function has large vanishing moments, then the wavelet coefficients will be small (Chau *et al.*, 2004).

Wavelet coefficients denoted by $g_{j,k}$ are also defined as (Watson *et al.*, 1998);

$$g_{j,k} = f'\left(\frac{1}{2^j}, \frac{k}{2^j}\right) \tag{6.4}$$

1D-DWT is a linear and orthogonal transform, in which a time-varying signal f(t) can be represented as (Watson *et al.*, 1998);

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} g_{j,k} \psi_{j,k}(t) \tag{6.5}$$

The wavelet decomposition coefficients representing the frequency content of the signal f(t) at different times are defined by using an orthonormal wavelet basis as follows (Watson *et al.*, 1998);

$$g_{j,k} = 2^{j/2} \sum_{l=-\infty}^{\infty} f(l) \psi(2^j l - k) \tag{6.6}$$

where f(l) is the l^{th} element of the signal.

6.1.1. Illustration of Multilevel Wavelet Decomposition with One Dimensional Discrete Wavelet Transform

50^{th} column of the PortSimHigh data set will be used here for the illustration of the third-level wavelet decomposition with wavelet type `db1` using the graphical user interface tools (1-D Wavelet Analysis Tool in the Wavelet Toolbox Main Menu) of MATLAB (version R2008b). In addition, MATLAB's `appcoef` and `detcoef` commands will be used for extracting 1D-DWT approximation and detail coefficients respectively.

Original signal (s, of length 10000) is the sum of the approximation at level three (a_3) and the details at levels three, two and one (d_3, d_2, d_1) as illustrated in Figure 6.10. Details are the noisy part of the original signal, it can even be stated that detail at level one is almost pure noise, whereas the overall trend of the signal (increasing/decreasing trends) is revealed in the approximation part. Original signal and its approximation at level three are almost similar as shown in Figure 6.10, in other words detail components can easily be discarded in thresholding step to improve compression without any significant loss in information content. Thus, original signal can be reconstructed by using only approximation at level three.

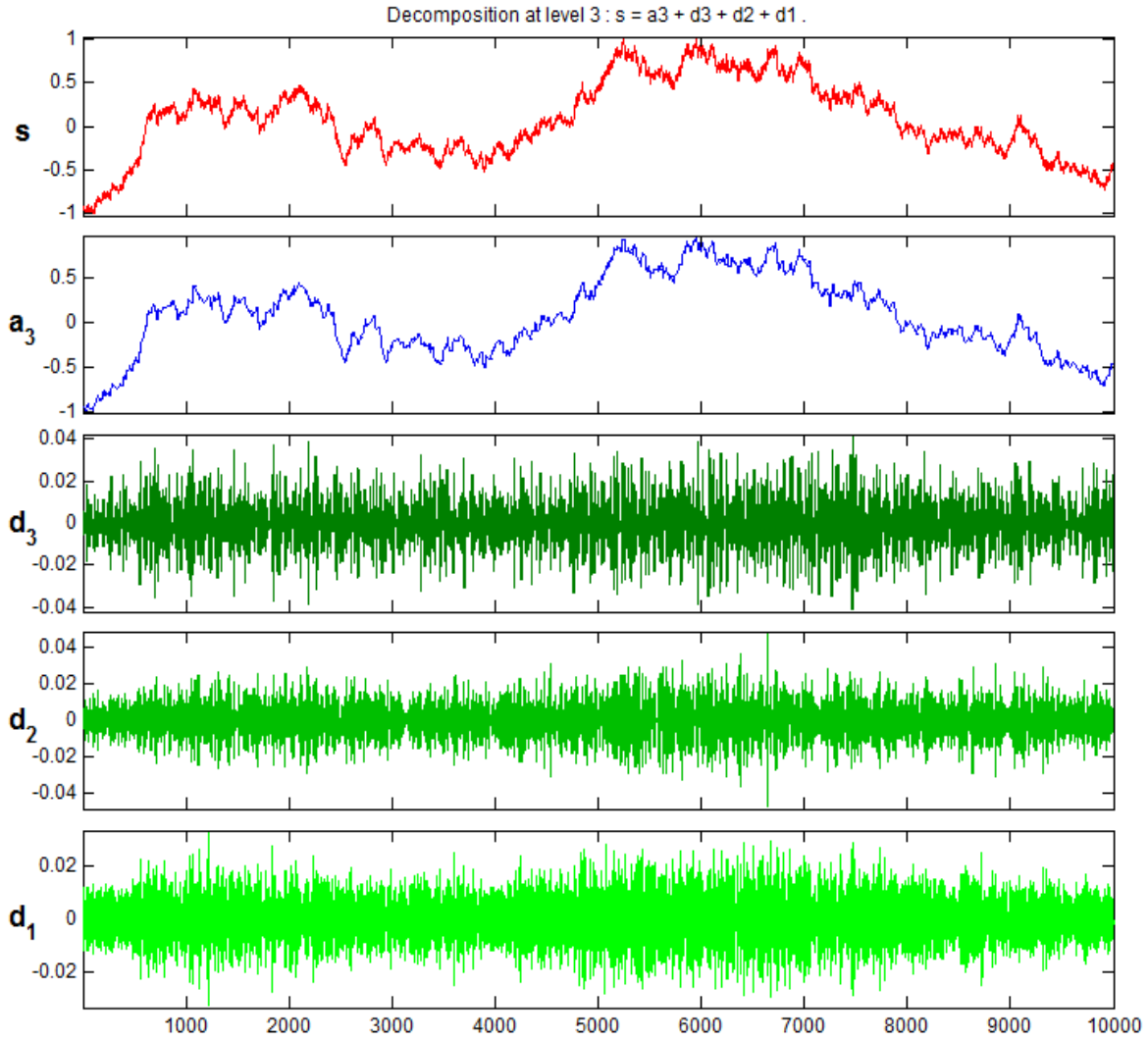

Figure 6.10. Three-Level Decomposition of the 50th Column of the PortSimHigh Data Set with Wavelet Type `db1`.

Wavelet-tree mode of the previous figure is shown in Figure 6.11. The original signal and its reconstructed approximation at level three (de-noised part of the signal which represents its overall trend) are almost same.

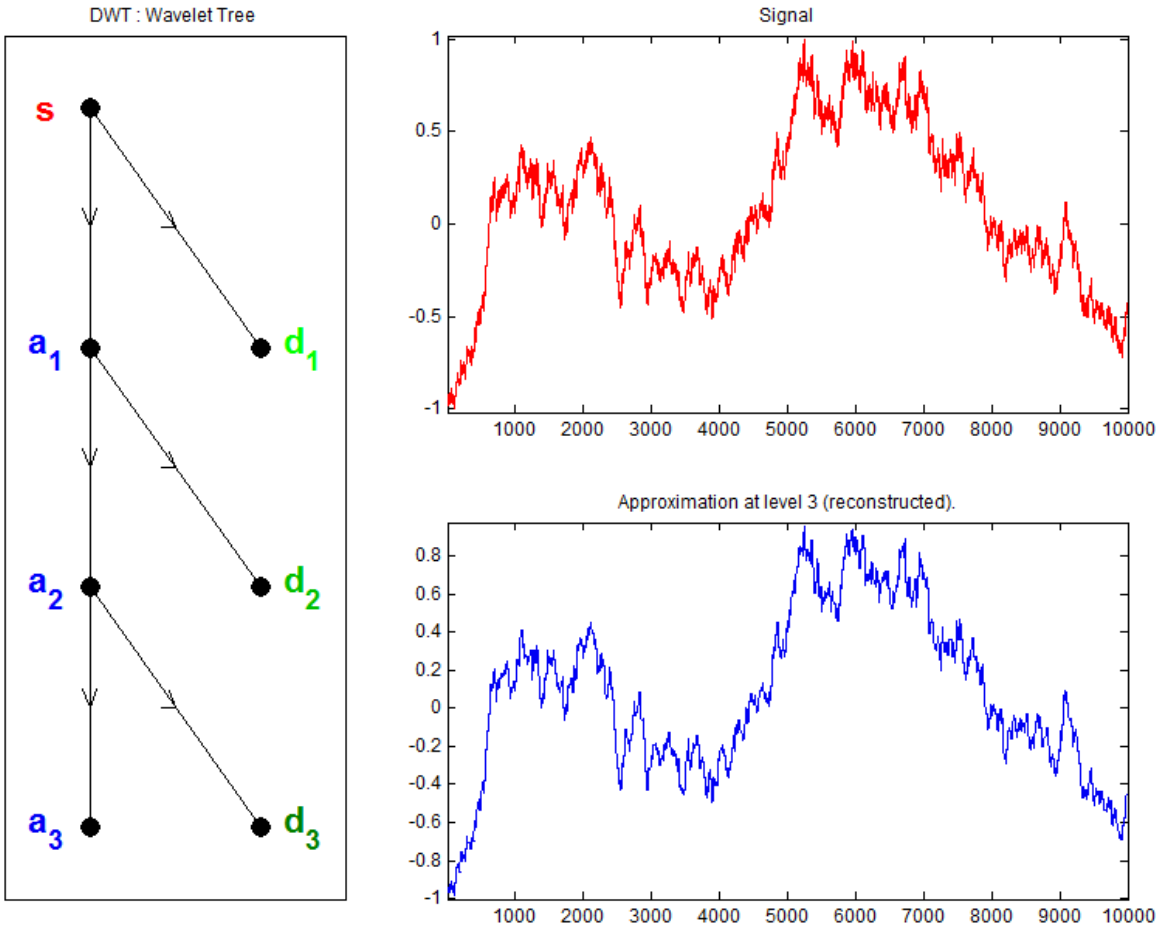

Figure 6.11. Original Signal of the 50th Column of the PortSimHigh Data Set and its Approximation at Level Three with Wavelet Type `db1`.

The original signal having 10000 data points (first sub-window), third-level approximation coefficients (second sub-window) having 1250 data points and detail coefficients from level three to one (third, fourth and fifth sub-windows) having 1250, 2500 and 5000 data points, respectively are shown in Figure 6.12. Magnitudes of the wavelet coefficients increase as the decomposition level also increases, meaning that most important information content is stored in the third-level coefficients. It can be concluded that original signal with dimension 10000 can be reconstructed by using only approximation coefficients at level three with dimension 1250 without any important distortion in the original signal while both dimensionality reduction and de-noising are achieved.

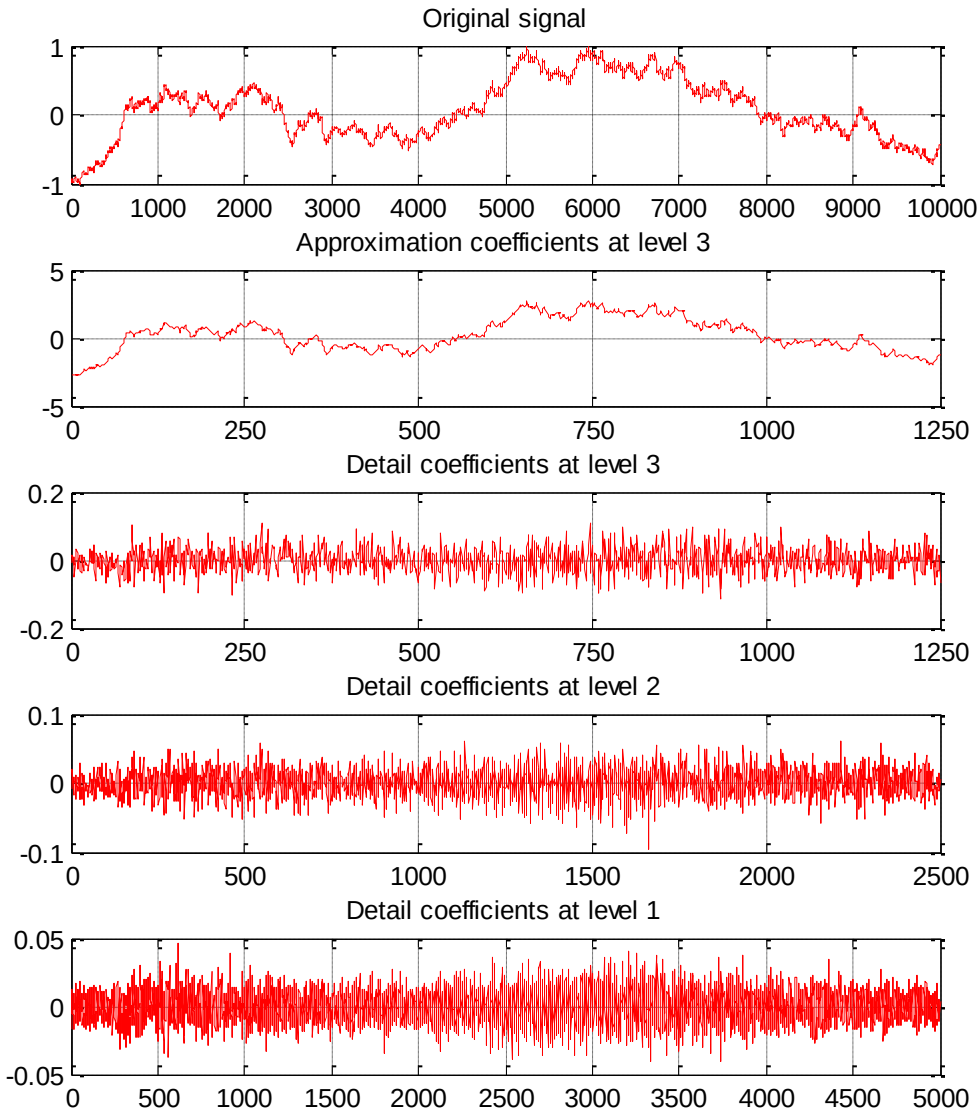

Figure 6.12. Third-Level Decomposition Coefficients of the 50th Column of the PortSimHigh Data Set with Wavelet Type `db1`.

Since the first and second-level detail coefficients are close to zero, they are discarded in thresholding step. In addition, the most largest first 100 third-level detail coefficients out of 1250 ones are selected, in other words, only third-level approximation coefficients with dimension 1250 and third-level detail coefficients with dimension 100 are stored. Synthesized signal is similar to the original signal as it is seen from Figure 6.13 concluding that if the detail coefficients of the first few levels (noisy parts of the signal) are eliminated, there will not be any important distortion in the original signal while high compression ratios are obtained by using de-noised signal components.

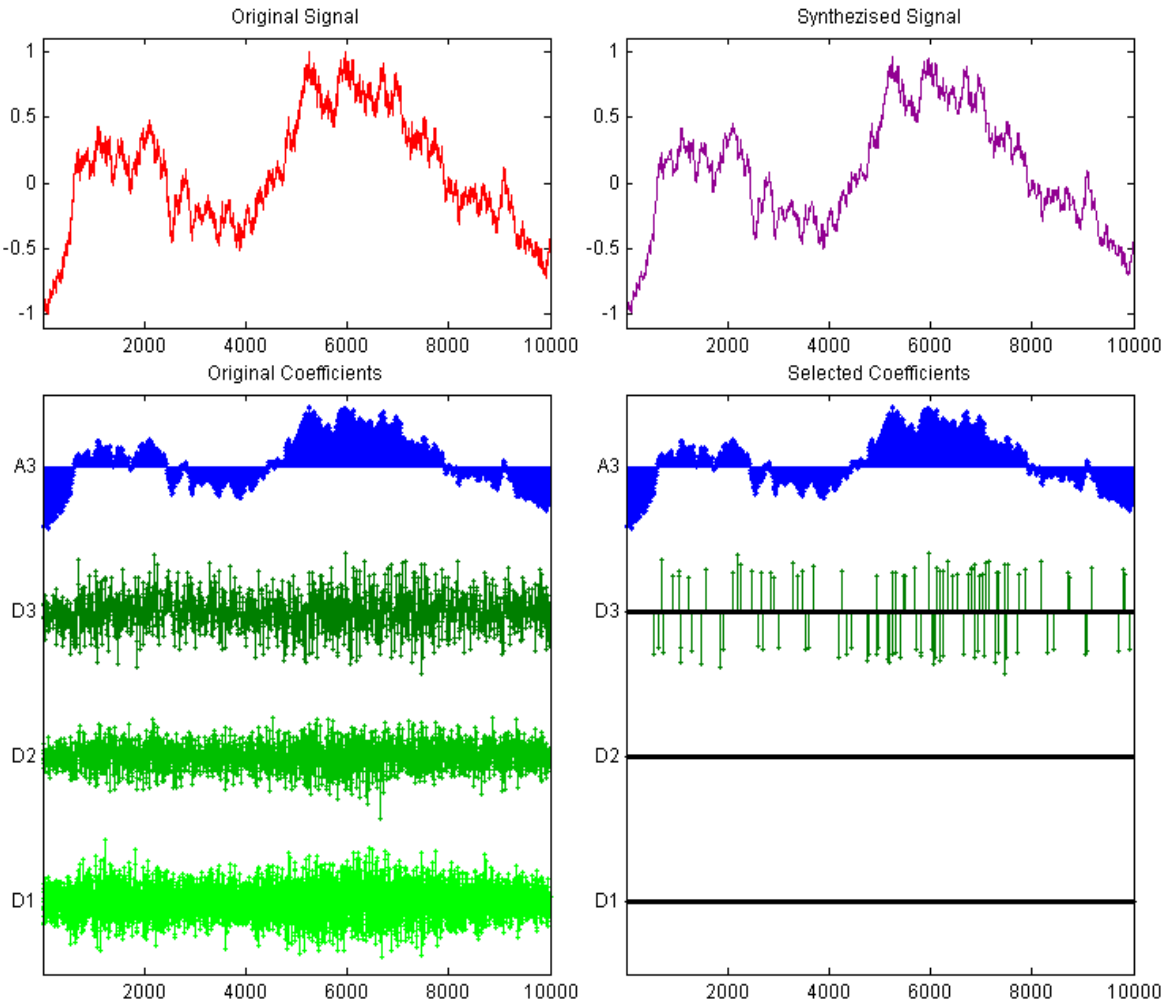

Figure 6.13. Thresholded Third-Level Decomposition Coefficients of the 50th Column of the PortSimHigh Data Set with Wavelet Type `db1`.

Wavelet coefficients of the 50th column of the PortSimHigh data were generated for three-level decomposition with wavelet type `db1` using MATLAB's `wavedec` function. These coefficients are then packed into a single coefficient vector **C** as depicted in Figure 6.7 as **C** = [$\mathbf{cA}_3$ | $\mathbf{cD}_3$ | $\mathbf{cD}_2$ | $\mathbf{cD}_1$]. These packed coefficients before and after thresholding, in which the percentile value is taken as 90%, are shown in the upper and bottom sub-windows of Figure 6.14. It is seen that the number of non-zero coefficients (of size 10000) becomes 1000 after thresholding. Log-log plot of the sorted absolute values of the 1D-DWT coefficients of size 10000 and the threshold limit of 0.3564 denoted by the red horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 90% is given in the middle sub-window of Figure 6.14. The coefficients below the threshold limit (9000 coefficients) are set to zero for compression, in

other words 90% of the transform coefficients become zero. The number of non-zero coefficients kept is 1000, which is only 10% of the number of data points.

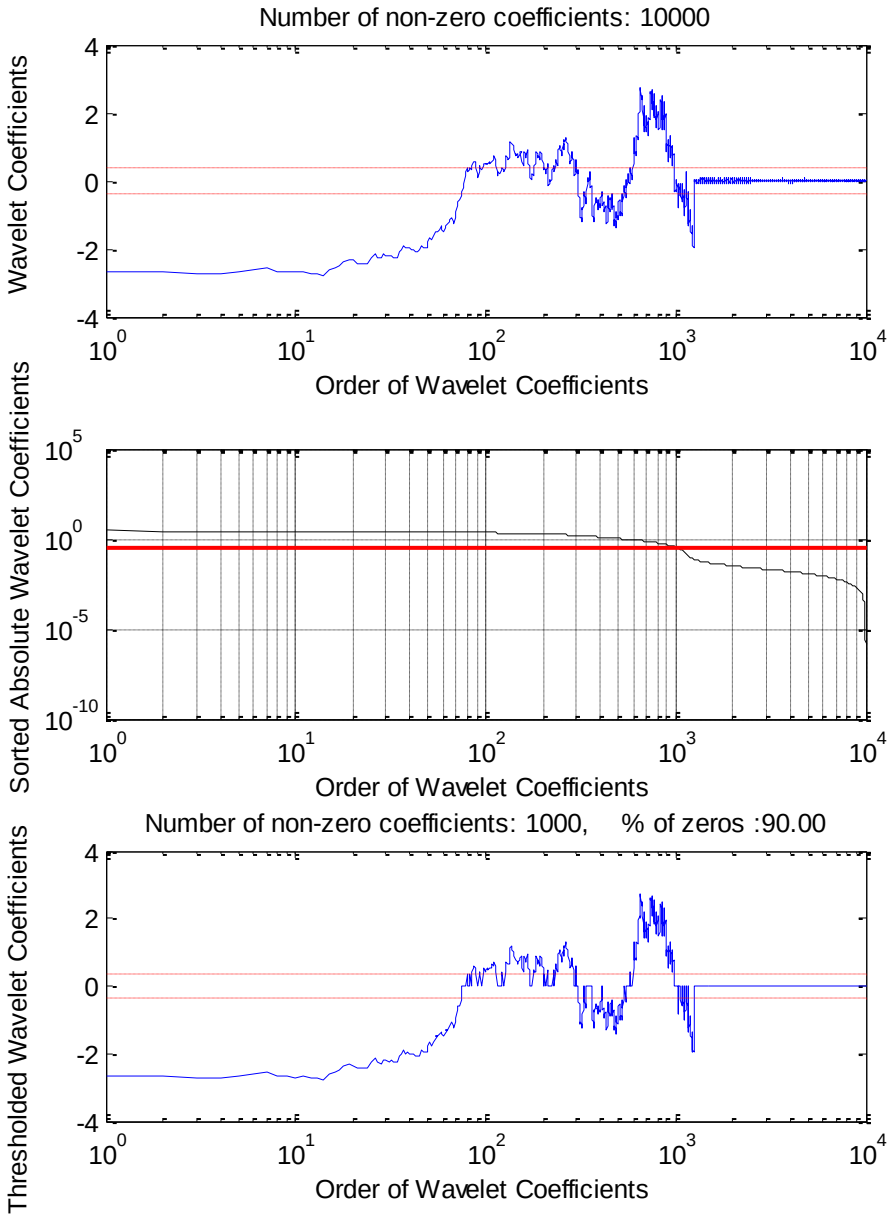

Figure 6.14. Sorted and Thresholded Three-Level Decomposition Coefficients of the 50th Column of the PortSimHigh Data Set with Wavelet Type `db1` for α=90 %.

Reconstructed data generated with inverse DWT and their overlay with the originals are shown in the upper sub-window of Figure 6.15. The major features of the stock prices such as decreasing/increasing trends are reconstructed with some distortion in the sharpest

points leading to higher reconstruction errors (= original - reconstructed) at these regions as illustrated in the bottom sub-window of Figure 6.15.

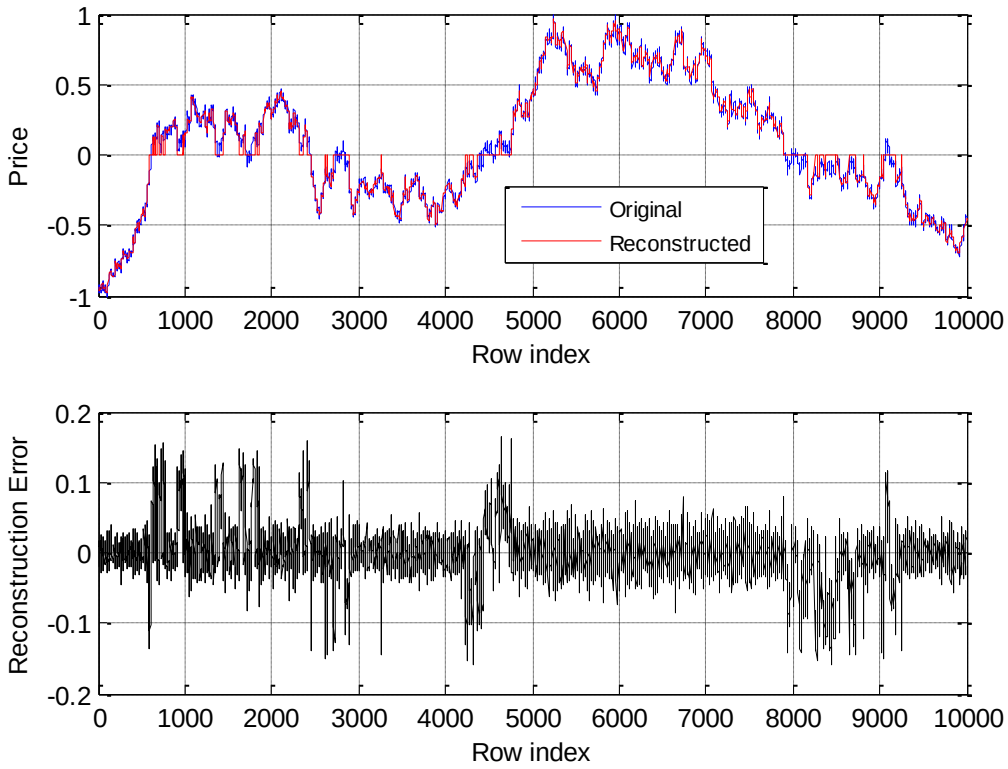

Figure 6.15. Original versus Reconstructed Signals and Reconstruction Errors after Applying Three-Level Decomposition with Wavelet Type `db1` for α=90 %.

6.2. Two Dimensional Discrete Wavelet Transform

Two Dimensional Discrete Wavelet Transform (2D-DWT) is computed by applying 1D-DWT in the horizontal and vertical directions of a data set (Weeks, 2007). Two-dimensional wavelet decomposition tree is presented in Figure 6.16 for the original signal s, where cA's are the approximation coefficients and $cD^{(h)}$, $cD^{(d)}$ and $cD^{(v)}$'s are the horizontal, diagonal and vertical detail coefficients respectively. Horizontal detail coefficients can be thought as the 1D-DWT detail coefficients, whereas diagonal and vertical detail coefficients are generally used for analyzing interactions between multiple series and adjusting the scale in these series respectively (Dillard and Shmueli, 2004).

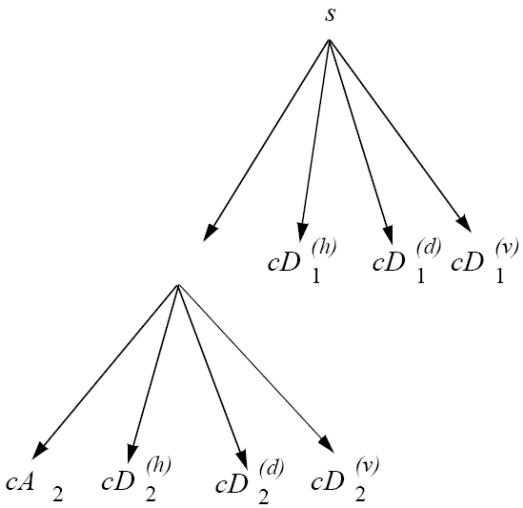

Figure 6.16. Two-Dimensional Wavelet Decomposition Tree (Mathworks, 2011).

In an n-level wavelet decomposition, approximation coefficients at level n (cA_n) and horizontal, vertical and diagonal detail coefficients (cH_n, cV_n, cD_n ,…, cH_1, cV_1, cD_1) at each level are assembled into the 'coefs' vector consisting of 3n+1 sections and sizes of these coefficients (cA_n, cV_n ,…, cV_1) and the original signal (X) are stored in the 'sizes' matrix with dimension (n+2)×2 in MATLAB as illustrated in Figure 6.17 (Mathworks, 2011). The number of data rows in the signal's wavelet coefficients decreases by half in each decomposition level, where approximation and detail (horizontal, vertical and diagonal) coefficients are produced. It should also be mentioned that the total number of 2D-DWT coefficients (later to be compressed) and the number of original data points can be different from each other unlike 1D-DWT.

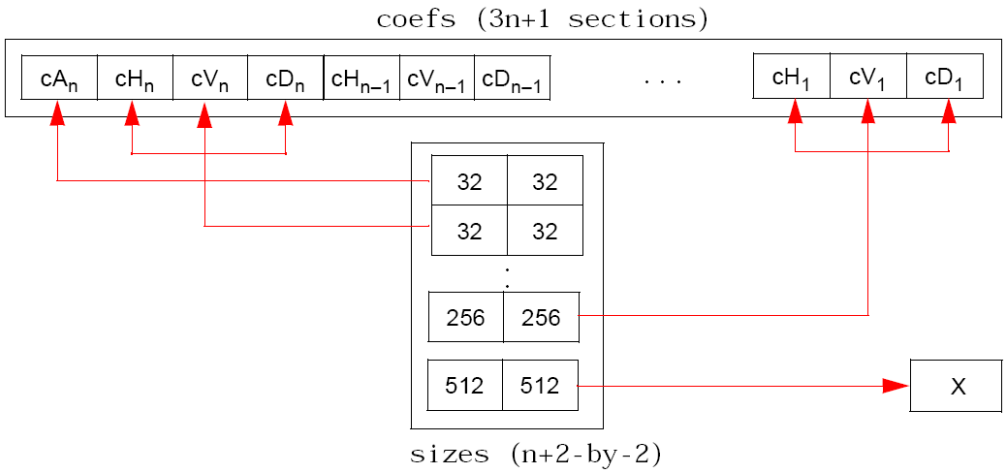

Figure 6.17. n-Level Decomposition Coefficients and Their Lengths (Mathworks, 2011).

In wavelet analysis, the scaling function, ϕ, and the wavelet function, ψ, represent approximation and detail components of a signal, respectively having the following properties (Mathworks, 2011);

$$\int \psi(x)dx = 0 \tag{6.7}$$

$$\int \phi(x)dx = 1 \tag{6.8}$$

In two-dimensional analysis, one scaling function $\phi(x,y)$ and three wavelets $\psi_V(x,y)$, $\psi_H(x,y)$ and $\psi_D(x,y)$ in three orientations (vertical, horizontal and diagonal) are defined as follows (Mathworks, 2011);

$$\begin{aligned} \phi(x,y) &= \phi(x)\phi(y) \\ \psi_V(x,y) &= \phi(x)\psi(y) \\ \psi_H(x,y) &= \psi(x)\phi(y) \\ \psi_D(x,y) &= \psi(x)\psi(y) \end{aligned} \tag{6.9}$$

Approximation coefficients denoted by $W_\phi(j_0,m,n)$ at level j_0 and detail coefficients denoted by $W_\psi(j,m,n)$ at each level are defined as follows (Liu, 2010);

$$W_\phi(j_{0,m,n}) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\phi_{j_0,m,n}(x,y) \tag{6.10}$$

$$W_\psi^i(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\phi_{j,m,n}^i(x,y) \tag{6.11}$$

$$\text{where } i = \{H,V,D\}$$

$$\text{for } \begin{cases} 0 \le m \le M-1 \\ 0 \le n \le N-1 \end{cases}$$

Finally, the data set f(x,y) can be represented as (Liu, 2010);

$$\begin{aligned} f(x,y) = {} & \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\phi(j_0,m,n)\phi_{j_0,m,n}(x,y) \\ & + \frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_\psi^i(j,m,n)\phi_{j,m,n}^i(x,y) \end{aligned} \tag{6.12}$$

The 2D-DWT is employed mostly in image processing. Approximation (A1) and horizontal (H1), vertical (V1) and diagonal (D1) detail components generated by one-step decomposition of a benchmark image are presented in Figure 6.18. It is seen that the image can be represented thoroughly by using only its approximation, whereas details consisting of the high-frequency components are not adequate to describe the image. Thus, all of the detail parts can be eliminated in order to improve compression without any detectable distortion in the image.

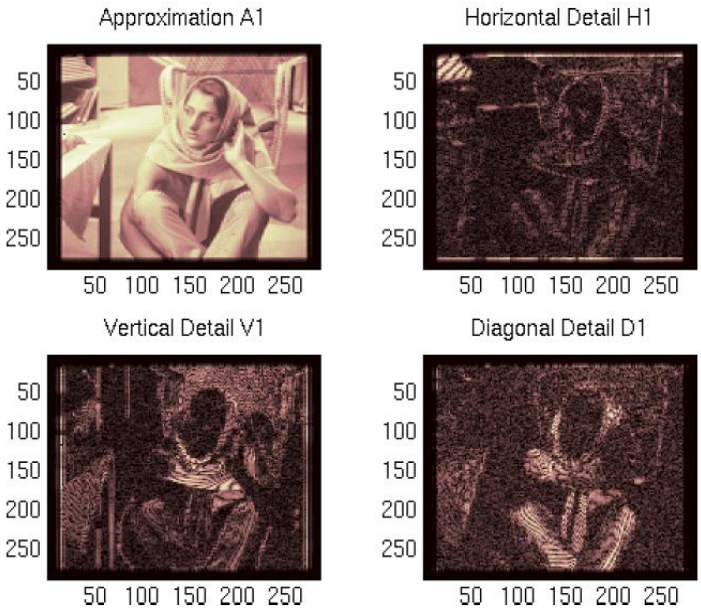

Figure 6.18. One-Step Decomposition of an Image (Mathworks, 2011).

Approximation (A1, A2) and horizontal (H1, H2), vertical (V1, V2) and diagonal (D1, D2) details of an image generated by second-level decomposition are presented in Figure 6.19. It can be stated that the image can be synthesized by using only its second-level approximation without any deterioration in the image quality (compared to the original image or its first-level approximation).

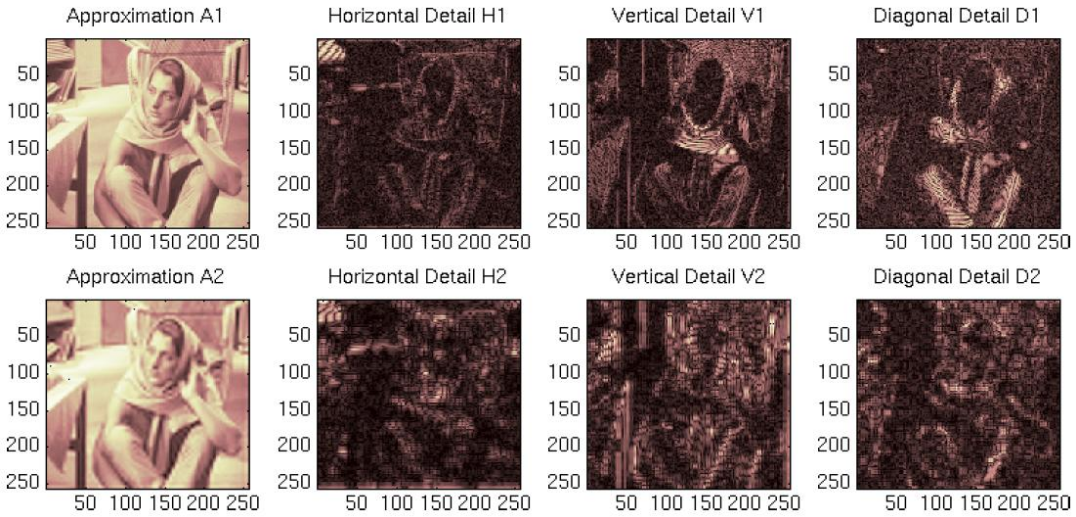

Figure 6.19. Two-Level Decomposition of an Image (Mathworks, 2011).

6.2.1. Illustration of Multilevel Wavelet Decomposition with Two Dimensional Discrete Wavelet Transform

Three inversely correlated columns of the PortSimLow data set (500th, 253th and 18th columns) and three columns of the PortSimHigh data set (350th, 400th and 15th columns) will be used here for the illustration of the three-level wavelet decomposition with wavelet type `db1` using MATLAB (version R2008b). MATLAB's `appcoef2` and `detcoef2` commands will be used for extracting the 2D-DWT approximation and detail coefficients. In addition, MATLAB's `wrcoef2` command will be used for reconstructing approximation and detail components of the original data set. The graphical user interface tools (2-D Wavelet Analysis Tool in the Wavelet Toolbox Main Menu) of MATLAB cannot be used for analyzing the multiple data series as these tools are implemented for image processing applications.

Original signals of the PortSimLow data set of size 10000×3 are the sum of the approximation at level three (A3), two (A2) and one (A1) and horizontal, vertical and diagonal details at levels three (H3, V3, D3), two (H2, V2, D2) and one (H1, V1, D1) as illustrated in Figure 6.20. It is observed that the most prominent patterns in the original series are preserved in the approximations and vertical details at level one (A1 and V1). However, horizontal and diagonal details at each level contain most of the high-frequency components, in other words they can safely be eliminated in thresholding step to improve

compression. Thus, the original signals can be synthesized by using only the approximations and vertical details at level one without any significant loss in the information content while both dimensionality reduction and de-noising are achieved.

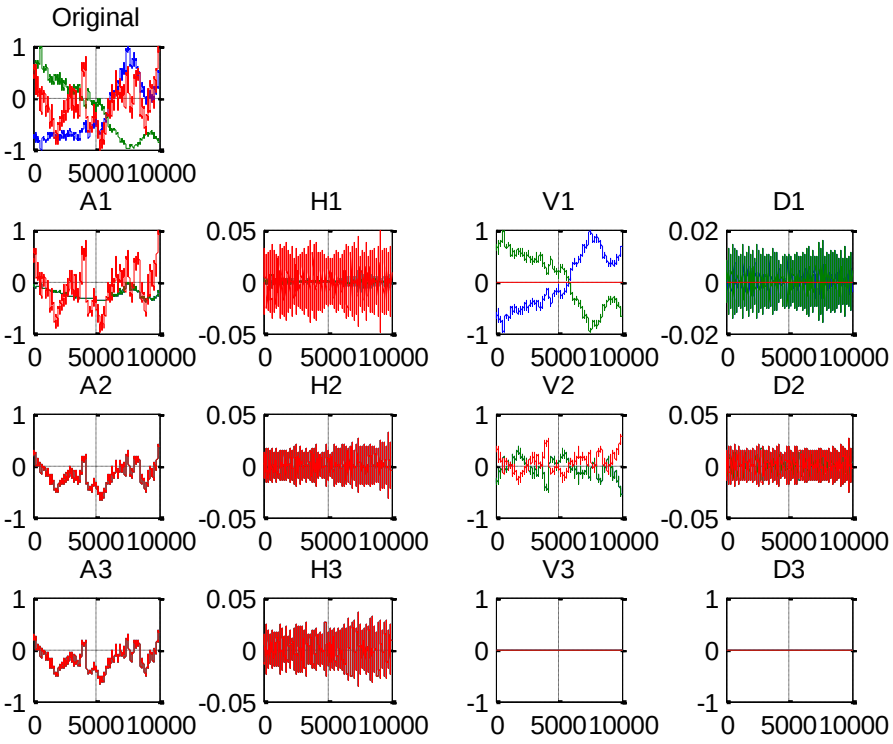

Figure 6.20. Three-Level Decomposition of the PortSimLow Data Set of Size 10000×3 with Wavelet Type `db1`.

The original signals each having 10000 data rows, approximation and horizontal, vertical and diagonal detail coefficients at level three (cA3, cH3, cV3, cD3), two (cA2, cH2, cV2, cD2) and one (cA1, cH1, cV1, cD1) having 1250, 2500 and 5000 data rows respectively are shown in Figure 6.21. Magnitudes of the wavelet coefficients increase as the level in decomposition increases, meaning that most important information content is stored in the third-level coefficients. It is also observed that the approximation and vertical detail coefficients are the largest in magnitude, and thus, the diagonal and horizontal detail coefficients can be eliminated safely without any significant distortion. It can also be stated that for an n-level decomposition, n+1 series may be used in order to generate non-zero vertical and diagonal detail coefficients in the deepest levels.

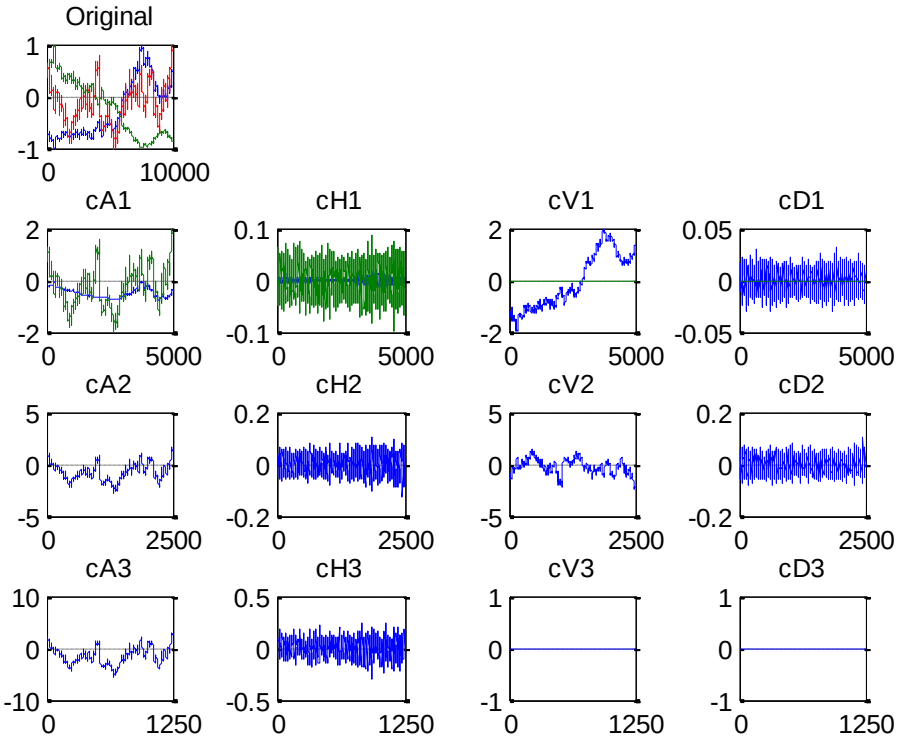

Figure 6.21. Three-Level Decomposition Coefficients of the PortSimLow Data Set of Size 10000×3 with Wavelet Type `db1`.

Wavelet coefficients of the PortSimLow data set of size 10000×3 were generated for three-level decomposition with wavelet type `db1` using MATLAB's `wavedec2` function. These coefficients are then concatenated into a single coefficient vector **C** as shown in Figure 6.17, where **C** = [$\mathbf{cA}_n$ | $\mathbf{cH}_n$ | $\mathbf{cV}_n$ | $\mathbf{cD}_n$ |…| $\mathbf{cH}_1$ | $\mathbf{cV}_1$ | $\mathbf{cD}_1$]. These packed coefficients before and after thresholding, where the percentile value is taken as 90%, are shown in the upper and bottom sub-windows of Figure 6.22. It is seen that the number of non-zero coefficients (of size 30000) becomes 4250 after thresholding. Log-log plot of the sorted absolute values of the 2D-DWT coefficients of size 42500 (the number of coefficients is higher than the number of data points of 30000) and the threshold limit of 0.8933 denoted by the red horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 90% is given in the middle sub-window of Figure 6.22. The coefficients below the threshold limit (38250 coefficients) are set to zero for compression, in other words 90% of the transform coefficients become zero. It can also be stated that all of the horizontal and diagonal detail coefficients are set to zero as they are smaller than the specified threshold limit as it can be seen from Figure 6.21.

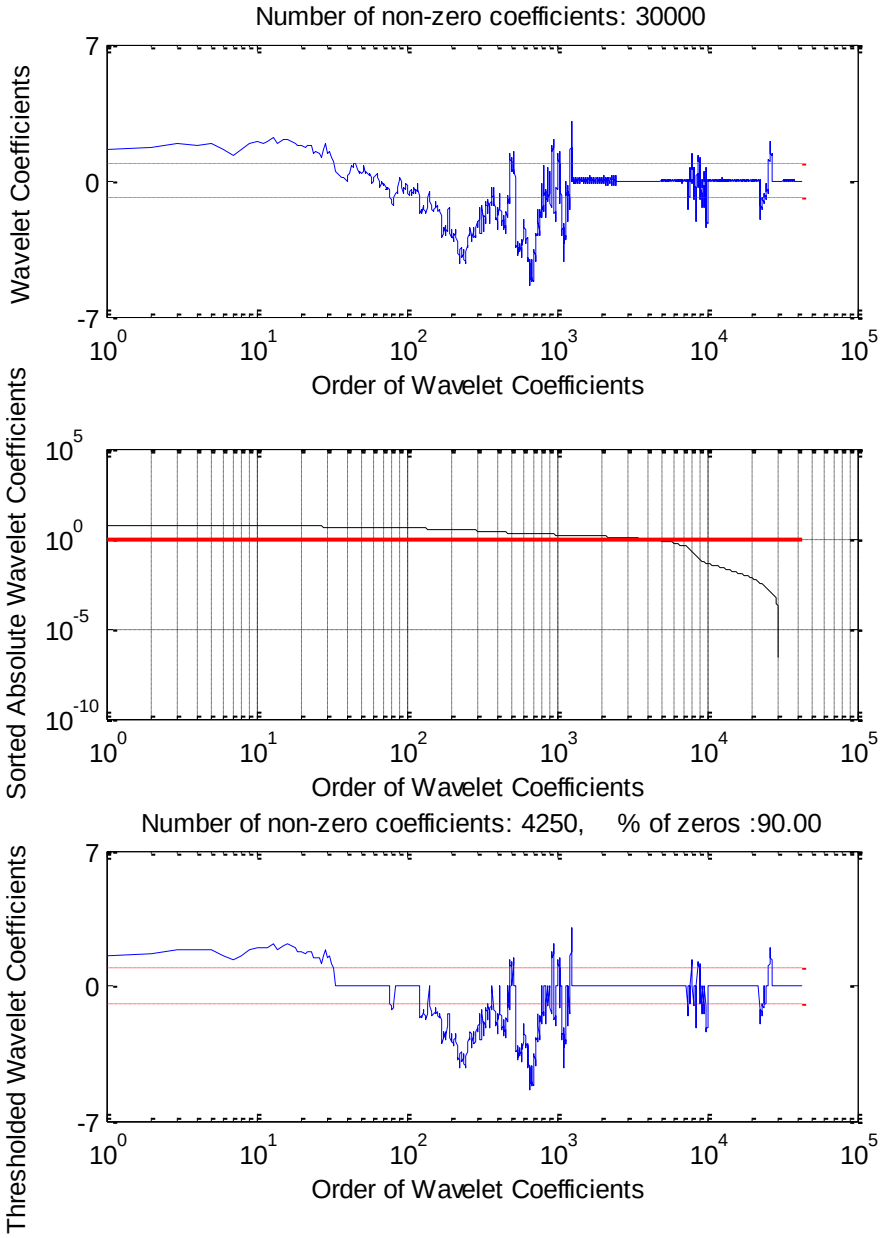

Figure 6.22. Sorted and Thresholded Three-Level Decomposition Coefficients of the PortSimLow Data Set of Size 10000×3 with Wavelet Type `db1` for α=90 %.

Reconstructed data series generated with inverse 2D-DWT and their overlay with the originals are shown in the sub-windows of Figure 6.23 on the left side. The prominent patterns of the stock prices are reconstructed by using only the largest approximations and vertical details with some distortion leading to reconstruction errors (= original - reconstructed) as illustrated in the sub-windows of Figure 6.23 on the right side. It is seen that minimum reconstruction error is generated in the 18th column of the PortSimLow data set.

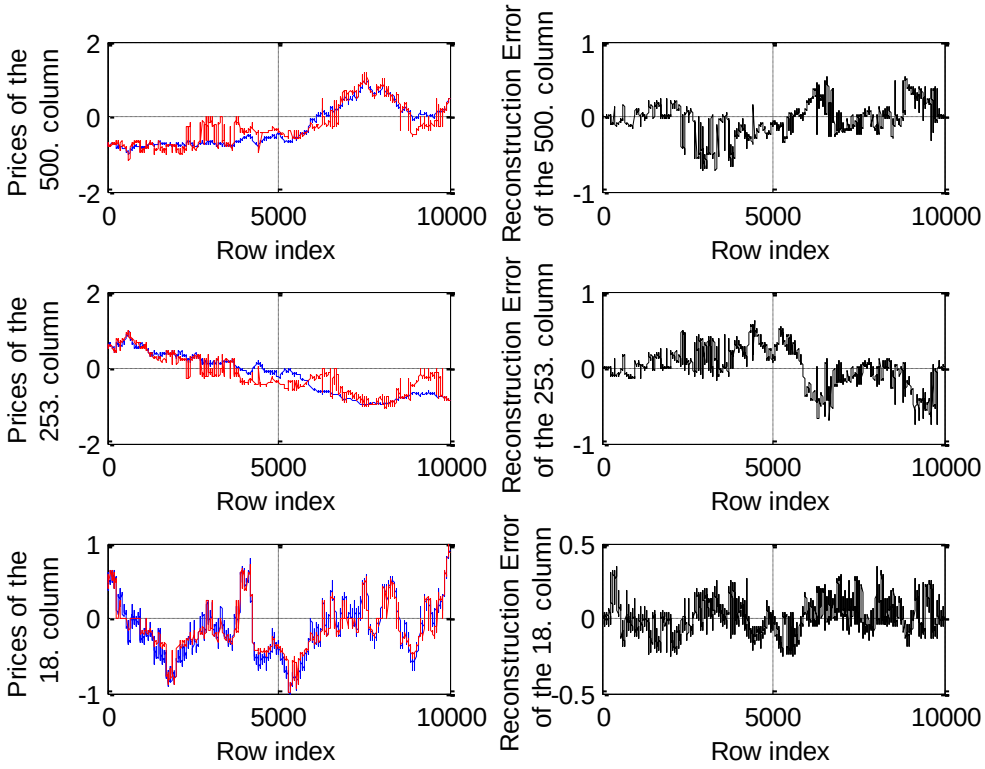

Figure 6.23. Original versus Reconstructed Signals and Reconstruction Errors of the PortSimLow Data Set of Size 10000×3 for α=90 %.

Original signals of the PortSimHigh data set of size 10000×3 (series are highly correlated that they overlap as shown in Figure 6.24) are the sum of the approximation at level three (A3), two (A2) and one (A1) and horizontal, vertical and diagonal details at levels three (H3, V3, D3), two (H2, V2, D2) and one (H1, V1, D1) as illustrated in Figure 6.24. It is observed that approximations at each level are similar to the original series, whereas horizontal, vertical and diagonal details at each level contain the most noisy parts, in other words they can simply be eliminated in thresholding step to improve compression. Thus, the original signal can be reconstructed by using only the approximations in any level without any significant loss in the information content.

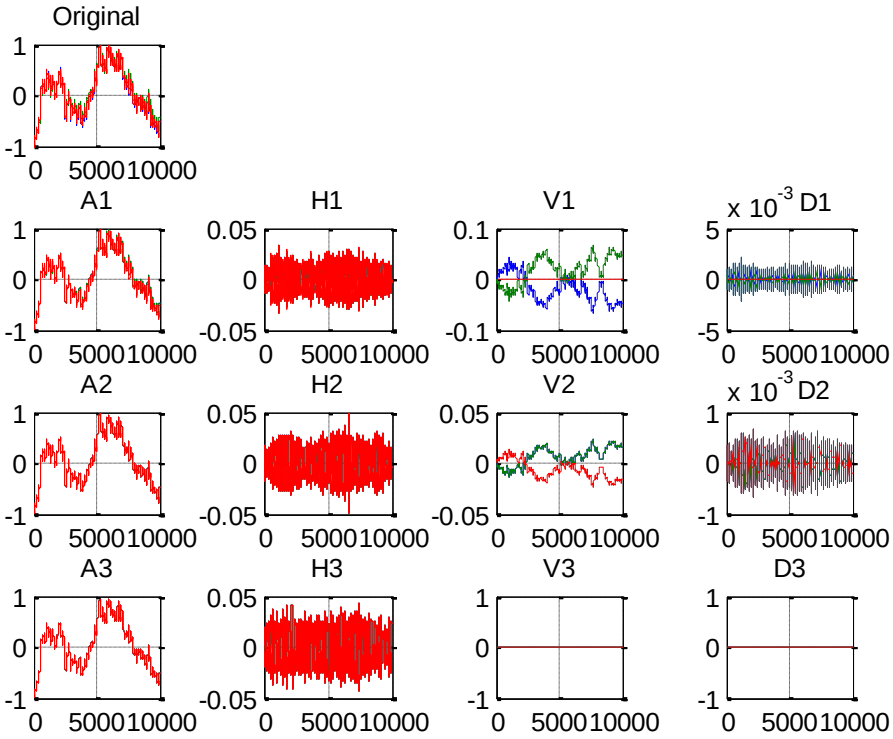

Figure 6.24. Three-Level Decomposition of the PortSimHigh Data Set of Size 10000×3 with Wavelet Type `db1`.

The original signals each having 10000 data rows, approximation and horizontal, vertical and diagonal detail coefficients at level three (cA3, cH3, cV3, cD3), two (cA2, cH2, cV2, cD2) and one (cA1, cH1, cV1, cD1) having 1250, 2500 and 5000 data rows respectively are shown in Figure 6.25. Magnitudes of the wavelet coefficients increase through the deepest levels, revealing that the most important information content is stored in the third-level approximation coefficients. Also, original series having a total of 30000 data points can be synthesized by using only the approximation coefficients at level three with dimension 1250 without any deterioration while both dimensionality reduction and de-noising are achieved simultaneously. In addition, it is observed that vertical and diagonal detail coefficients of the PortSimHigh data set are much smaller than those of the PortSimLow data set as the inversely correlated series of the PortSimLow data set have different scales, requiring a proper scale adjustment represented by the vertical detail coefficients. Moreover, it is observed that differential changes represented by the diagonal detail coefficients are more pronounced in these inversely correlated series. Furthermore, these small detail coefficients of the highly correlated series can simply be set to zero in the thresholding step in order to maximize compression while perfect reconstruction is retained.

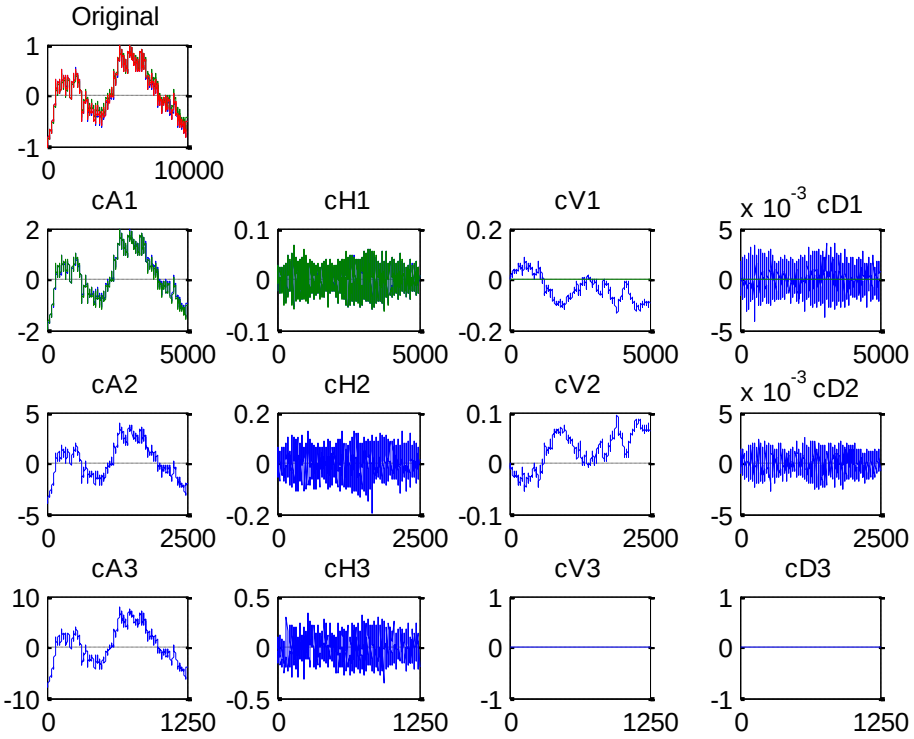

Figure 6.25. Three-Level Decomposition Coefficients of the PortSimHigh Data Set of Size 10000×3 with Wavelet Type `db1`.

Wavelet coefficients of the PortSimHigh data set of size 10000×3 generated for three-level decomposition with wavelet type `db1` using MATLAB's `wavedec2` function are concatenated into a single coefficient vector **C** as shown in Figure 6.17, where **C** = [$\mathbf{cA}_n$ | $\mathbf{cH}_n$ | $\mathbf{cV}_n$ | $\mathbf{cD}_n$ |…| $\mathbf{cH}_1$ | $\mathbf{cV}_1$ | $\mathbf{cD}_1$]. These packed coefficients before and after thresholding, in which the percentile value is taken as 97%, are shown in the upper and bottom sub-windows of Figure 6.26. It is seen that the number of non-zero coefficients (of size 30000) becomes 1275 after thresholding. Log-log plot of the sorted absolute values of the 2D-DWT coefficients of size 42500 (the number of coefficients is higher than the number of data points of 30000) and the threshold limit of 0.2155 denoted by the red horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 97% is given in the middle sub-window of Figure 6.26. The coefficients below the threshold limit (41225 coefficients) are set to zero for compression, in other words 97% of the transform coefficients become zero. The number of non-zero coefficients kept is 1275, which is only 3% of the number of coefficients. It can also be stated that the horizontal detail coefficients at the first two levels and all of the vertical and diagonal detail coefficients are set to zero as they are smaller than the specified threshold limit as illustrated in Figure 6.25.

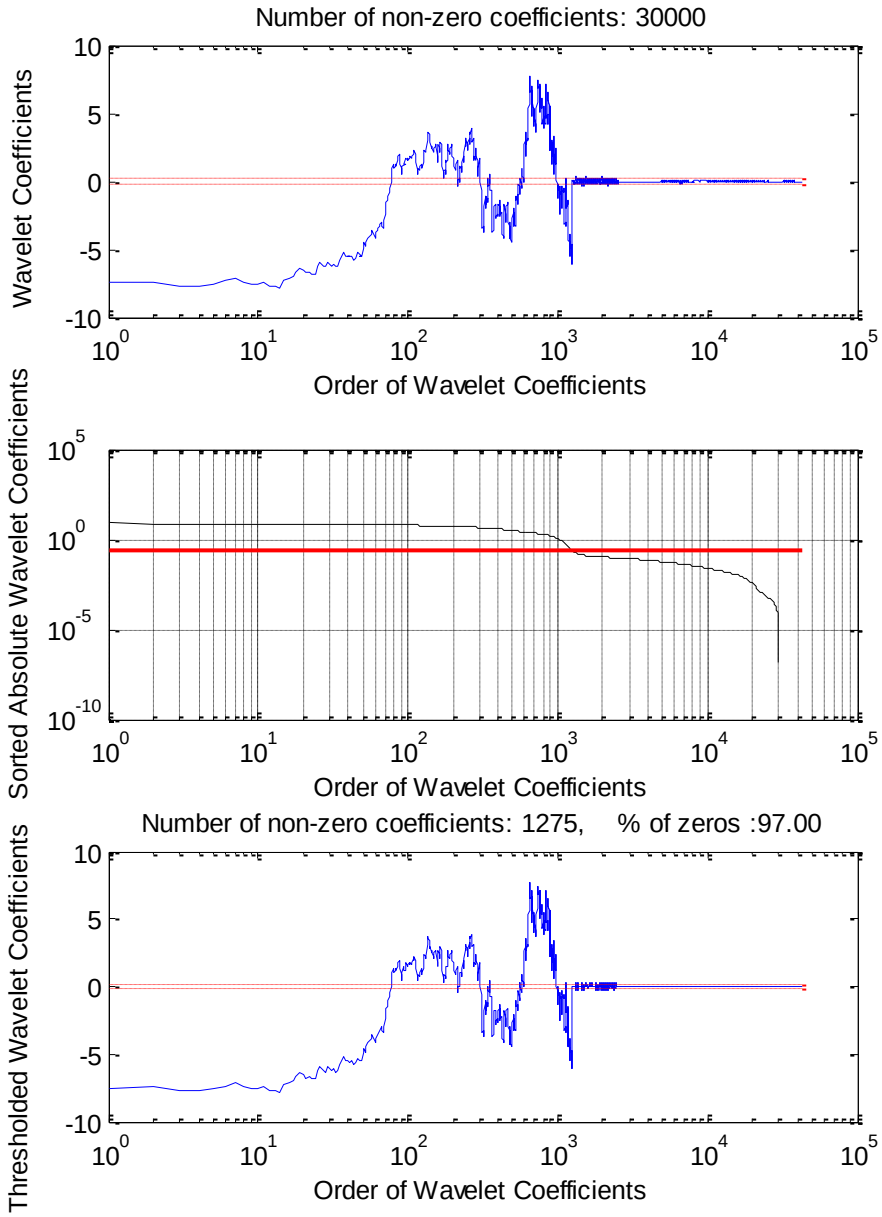

Figure 6.26. Sorted and Thresholded Three-Level Decomposition Coefficients of the PortSimHigh Data Set of Size 10000×3 with Wavelet Type `db1` for α=97 %.

Reconstructed data series generated with inverse 2D-DWT and their overlay with the originals are shown in the sub-windows of Figure 6.27 on the left side. The major features of the stock prices are reconstructed by using only the largest horizontal details at level three and approximations without any significant distortion leading to small reconstruction errors (= original - reconstructed) as illustrated in the sub-windows of Figure 6.27 on the right side. It is observed that PortSimHigh data series are reconstructed better than the PortSimLow data series although the percentile value used in thresholding of the former

data set is larger. This almost perfect reconstruction is due to high correlations among the PortSimHigh data series.

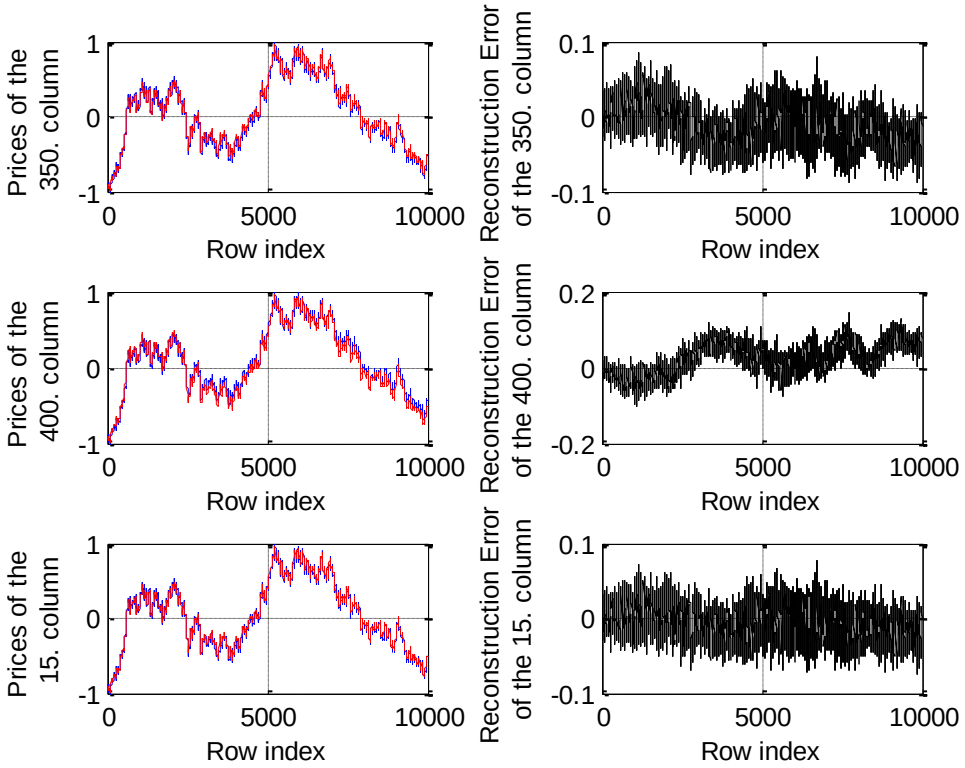

Figure 6.27. Original versus Reconstructed Signals and Reconstruction Errors of the PortSimHigh Data Set of Size 10000×3 for α=97 %.

6.3. Applications of One Dimensional and Two Dimensional Discrete Wavelet Transforms

In this section, data compression and lossy reconstruction will be studied by using the 1D-DWT and 2D-DWT compression techniques, the thresholding method as a lossy compression step and ZIP as the lossless encoding algorithm using the data sets PortSimHigh, PortSimLow, SELDI-TOF MS and TEP mentioned in Chapter 3 for 10 different percentile values of the frequency distribution of the transform coefficients in the [15%-99.8%] range by using wavelet types `db1` for the PortSimHigh and PortSimLow data sets, `db4` for the SELDI-TOF MS data set and `sym4` for the TEP data set at different decomposition levels. The efficacy of the DWT depends directly on the selected mother wavelet function. Hence, `db1` is used for smoother data sets (PortSimHigh and

PortSimLow), whereas `db4` and `sym4` are preferred for noisy data sets (SELDI-TOF MS and TEP).

Detailed 1D-DWT and 2D-DWT analyses are given for each of the data sets for the percentile value 99.5%. However, the figures of the transform coefficients before and after thresholding are not presented as all of the approximation and detail coefficients at each decomposition level cannot be demonstrated in a compact form. Examples of 1D-DWT and 2D-DWT coefficients can be seen in Sections 6.1.1 and 6.2.1, respectively. Furthermore, the total number of coefficients and the number of original data points can be different from each other as discussed in Section 6.2. In addition, the decomposition level in DWT is selected so as to yield the same compression level generated by DCT at the percentile value 99.5%. Consequently, reconstruction error norms produced in DCT and DWT can be compared at the same compression level. Furthermore, the effect of the percentile values used in thresholding step on compression is measured in terms of compression ratio, mean error norm, % relative global error and % relative maximum error. Ratios of compression ratio to mean error norm are also computed to determine the optimum percentile level.

The computations were done in MATLAB (version R2008b) using MATLAB's `wavedec`, `wavedec2`, `waverec` and `waverec2` commands from the Wavelet Toolbox for multi-level 1D and 2D wavelet decomposition and reconstruction and MATLAB's internal `zip` command is used as the lossless compression algorithm.

6.3.1. One Dimensional Discrete Wavelet Transform of the PortSimHigh Data Set

In this section, 1D-DWT with 10-level decomposition and the wavelet type `db1` is applied to the overall PortSimHigh data set (consisting of 500 stock prices) and for illustration purposes, only the first 16 stock prices of the PortSimHigh data set are presented. The first 16 stock prices were given before in Chapter 5 in scaled format with Figure 5.1.

Semilog-log (upper sub-window of Figure 6.28) and log-log (bottom sub-window of Figure 6.28) plots of the sorted absolute values of the 1D-DWT coefficients of the overall

PortSimHigh data set which are padded into a vector of size 5002000 (the number of coefficients is higher than the number of data points of 5000000, unlike 1D-DCT) and the threshold limit of 1.1315 denoted by the red horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 6.28. Transform coefficients above and below the threshold limit can be identified clearly in the log-log plot. The coefficients below the threshold limit (4976990 coefficients) are set to zero for compression, in other words 99.5% of the transform coefficients become zero. The number of nonzero coefficients kept is 25010, which is only 0.5% of the number of data points.

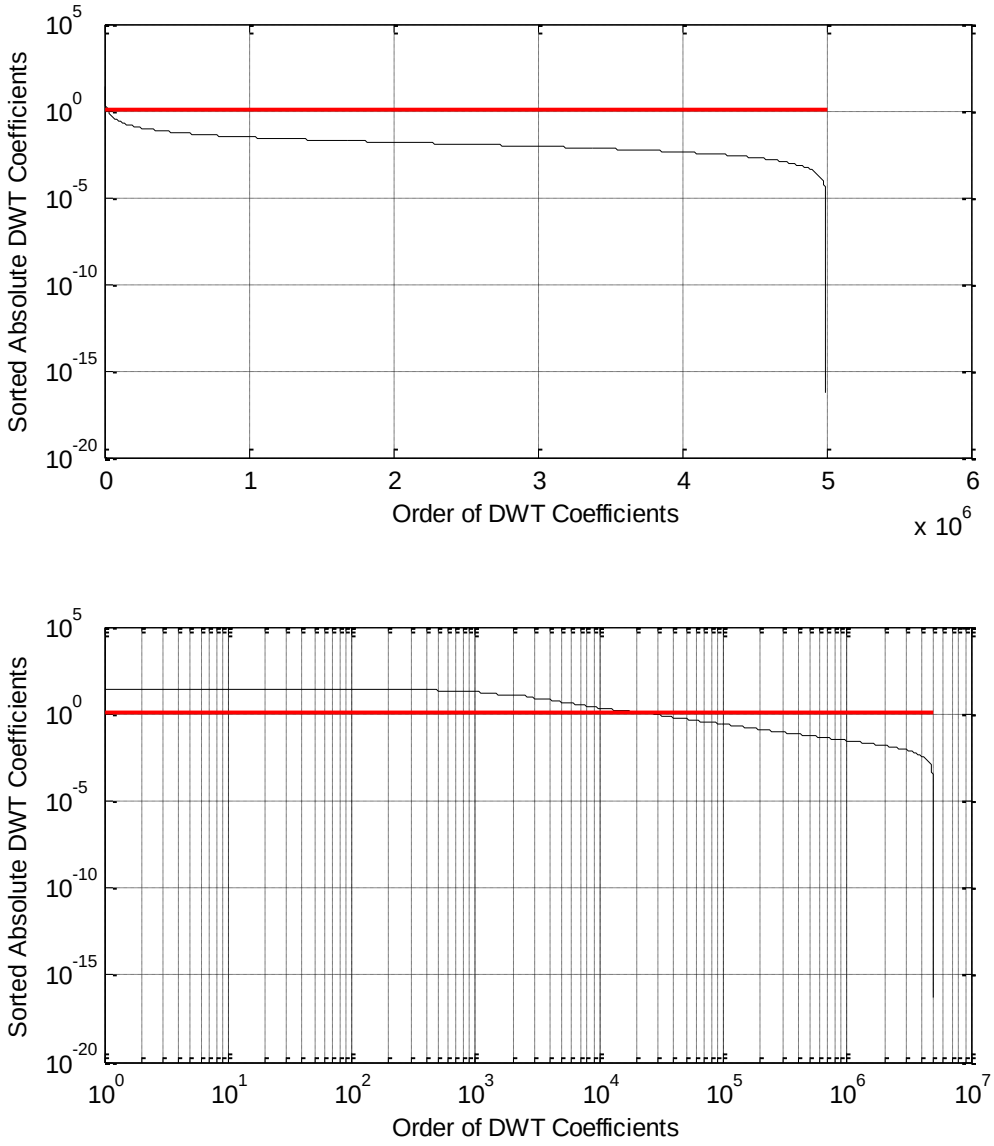

Figure 6.28. Semilog-log and Log-log Plots of Sorted Absolute 1D-DWT Coefficients of the Overall PortSimHigh Data Set for α=99.5 %.

The sizes of the ZIP files of the original and encoded data sets are compared in Figure 6.29. It is seen that the ZIP file of the overall PortSimHigh data set is nearly 51.1 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.93 MB. Thus, compression can be increased 54.9 times by applying 1D-DWT technique and taking the percentile value as 99.5% in thresholding step (similar to the 1D-DCT result).

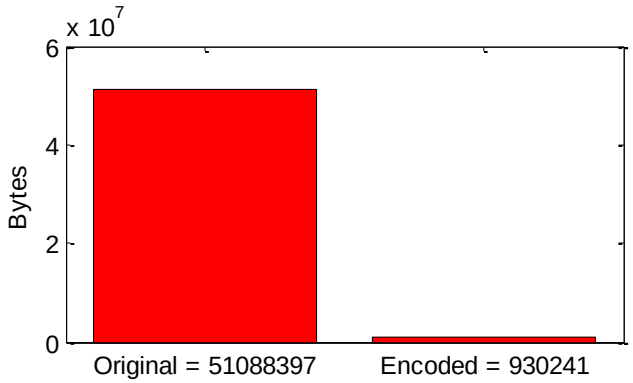

Figure 6.29. ZIP Compression Comparison of the Original and Encoded Overall PortSimHigh Data Set for α=99.5 % with 1D-DWT.

Stepwise reconstructed data generated with 1D-iDWT (inverse DWT) and their overlay with the originals are shown in Figure 6.30 and Figure 6.31 respectively. The prominent features of the stock prices such as decreasing/increasing trends are reconstructed without any significant loss, whereas the sharpest features are smoothened as most of the detail coefficients keeping the high-frequency information are eliminated as illustrated in Figure 6.31.

The original and reconstructed signals are plotted around the y=x line in Figure 6.32. It can be stated that reconstructed signals are similar to the original signals as reconstructed data points are scattered around the y=x line (all of the 16 example columns).

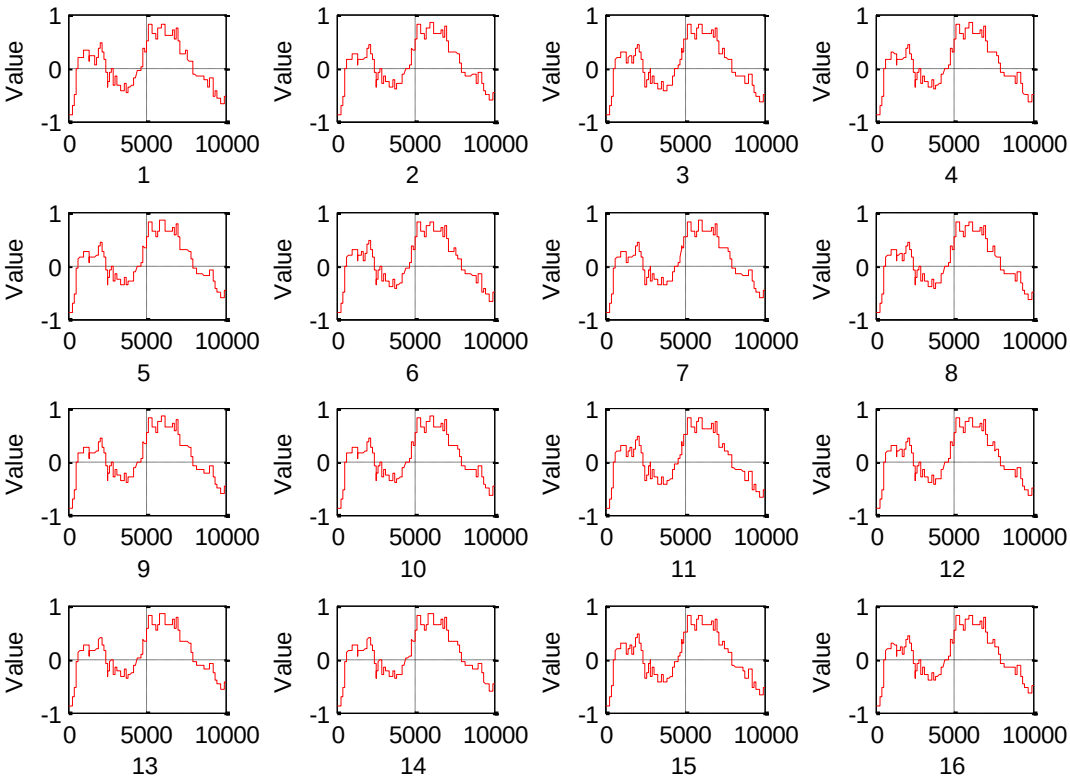

Figure 6.30. Reconstructed PortSimHigh Data Set with Inverse 1D-DWT.

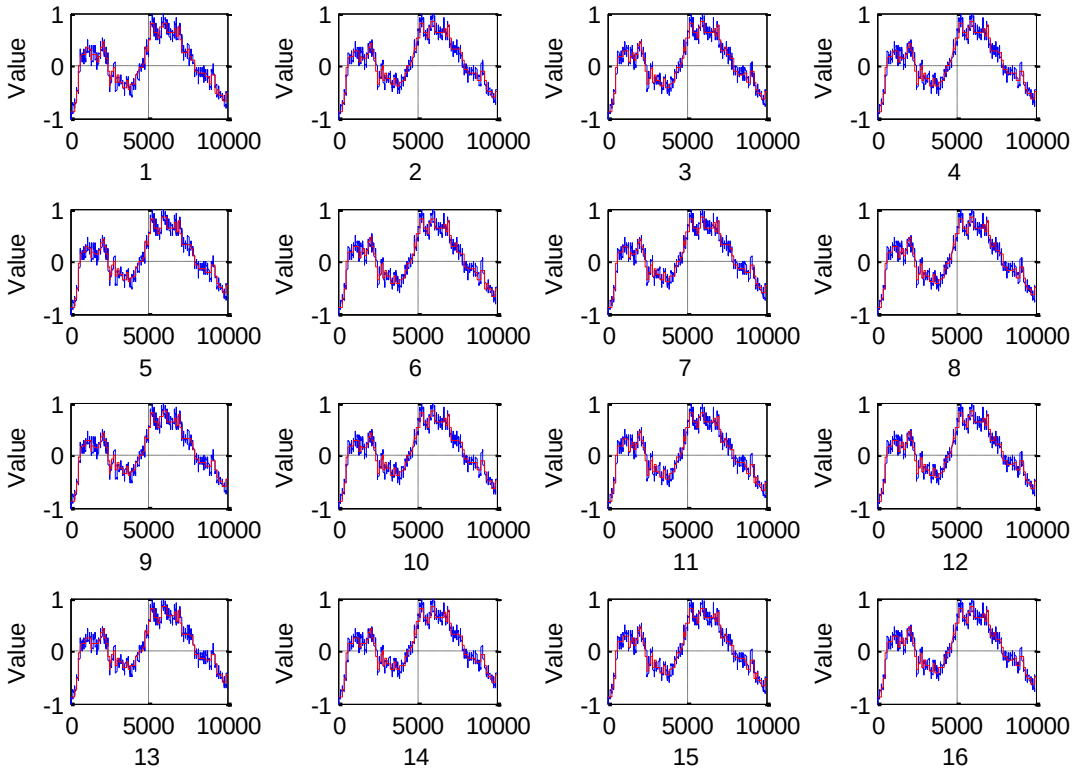

Figure 6.31. Original and Reconstructed Signals of the PortSimHigh Data Set with Inverse 1D-DWT.

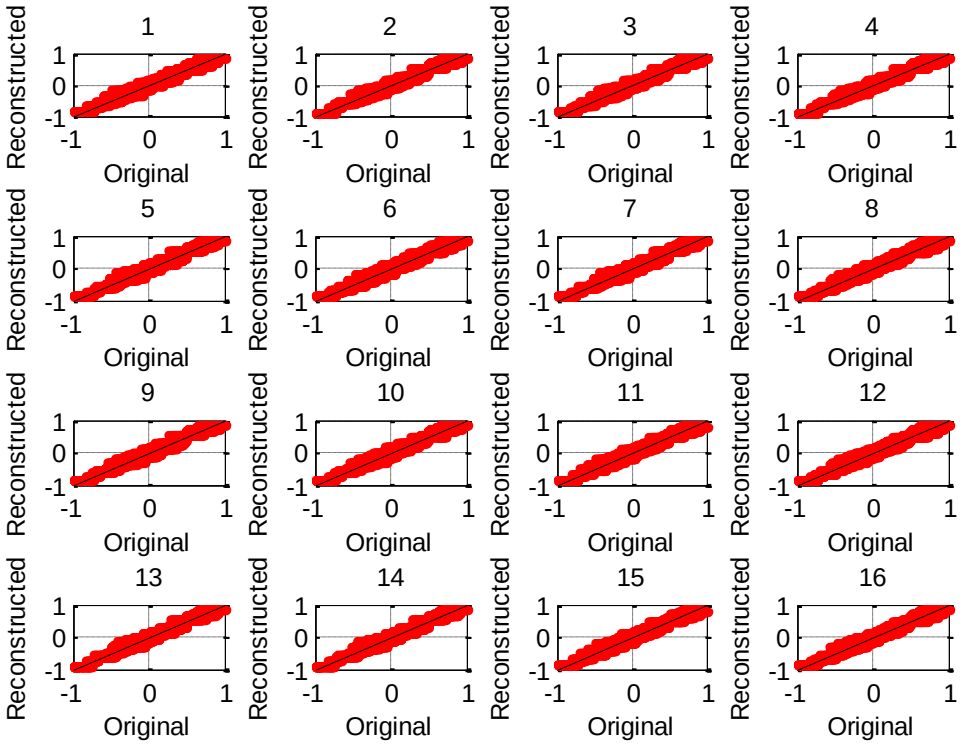

Figure 6.32. Reconstructed versus Original Signals of the PortSimHigh Data Set with Inverse 1D-DWT.

Error norms between original and reconstructed data sets are calculated per data column. Minimum error norm of 576.88 is obtained in the 441th column and the maximum error norm of 617.85 is obtained in the 463th column of the PortSimHigh data set. As it can be seen from both the upper and the middle sub-windows of Figure 6.33, most of the reconstructed data points are located around the y=x line meaning that reconstructed signals are close to the original signals providing small error norms. Furthermore, reconstruction error norm values of each of the 500 columns of the PortSimHigh data set are given in the bottom sub-window of Figure 6.33. It can be concluded that, for PortSimHigh data set, slightly larger reconstruction error norms are produced with 1D-DWT at the same compression level of 55 as compared to 1D-DCT studied in Section 5.3.1 of Chapter 5.

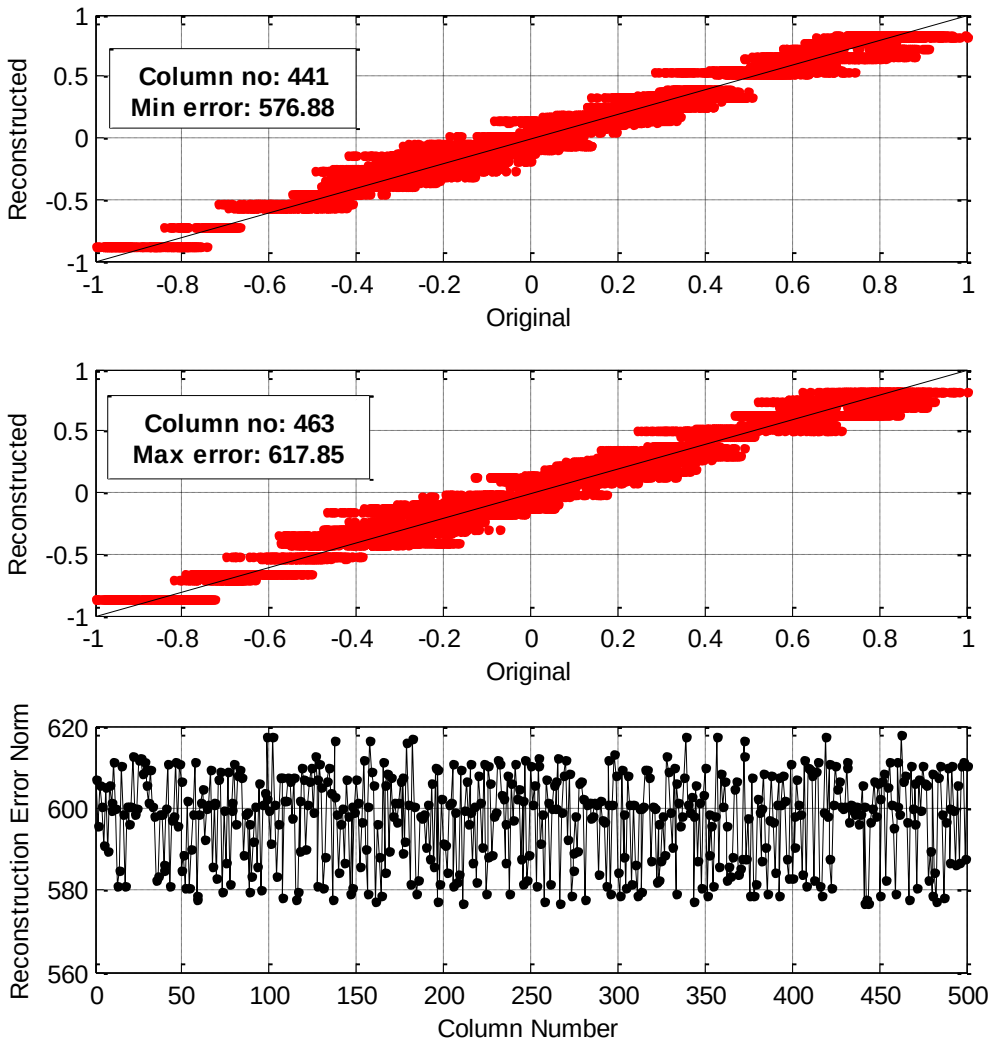

Figure 6.33. Reconstruction Error Norm Values of the PortSimHigh Data Set with Inverse 1D-DWT.

6.3.2. Two Dimensional Discrete Wavelet Transform of the PortSimHigh Data Set

In this section, 2D-DWT with 10-level decomposition and the wavelet type `db1` is applied to the overall PortSimHigh data set (consisting of 500 stock prices), however, for illustration purposes, only the related figures of the first 16 stock prices of the PortSimHigh data set are presented.

Semilog-log (upper sub-window of Figure 6.34) and log-log (bottom sub-window of Figure 6.34) plots of the sorted absolute values of the 2D-DWT coefficients of the overall PortSimHigh data set which are concatenated into a vector of size 5003830 (the number of coefficients is higher than the number of data points of 5000000 unlike 2D-DCT) and the

threshold limit of 0.2003 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 6.34. The coefficients below the threshold limit (4978811 coefficients) which are identified clearly in the log-log plot, are set to zero for compression, in other words 99.5% of the transform coefficients become zero. The number of nonzero coefficients kept is 25019, which is only 0.5% of the number of data points. In addition, the threshold limit is much smaller than that found with 1D-DWT, that is 1.1315, as 2D-DWT coefficients produced for the highly correlated data set are smaller.

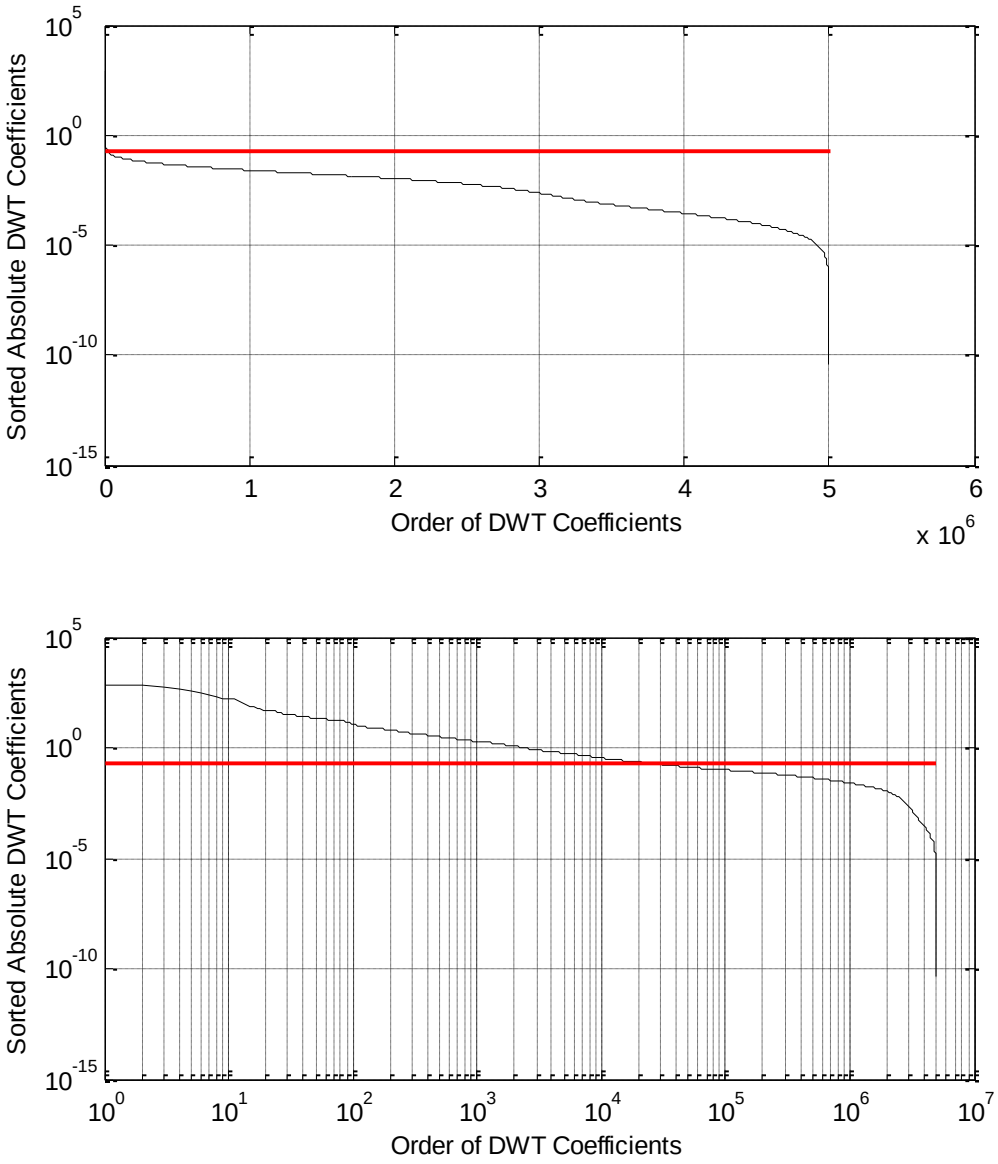

Figure 6.34. Semilog-log and Log-log Plots of Sorted Absolute 2D-DWT Coefficients of the Overall PortSimHigh Data Set for α=99.5 %.

Sizes of the ZIP files of the original and encoded data sets are compared in Figure 6.35. The ZIP file of the overall PortSimHigh data set is nearly 49.9 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.84 MB. Thus, compression can be increased 59.3 times by applying 2D-DWT technique and taking the percentile value as 99.5% in thresholding step. It can also be stated that the compression ratio obtained with 2D-DWT is slightly larger than that obtained with 1D-DWT, which was 54.9 as mentioned in Section 6.3.1. In addition, higher compression is yielded with 2D-DWT compared to 2D-DCT with which the compression ratio had been calculated as 55.7 in Section 5.3.2.

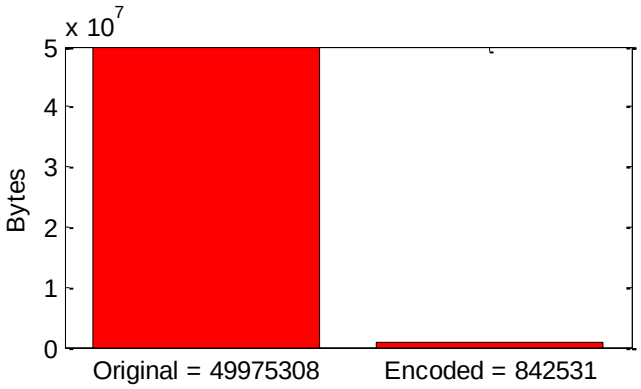

Figure 6.35. ZIP Compression Comparison of the Original and Encoded Overall PortSimHigh Data Set for α=99.5 % with 2D-DWT.

Reconstructed data generated with 2D-iDWT (inverse DWT) and their overlay with the originals are shown in Figure 6.36 and Figure 6.37 respectively. It is seen that original and reconstructed data are overlapped as illustrated in Figure 6.37 concluding that there is almost no distortion in the reconstructed data set. It can also be stated that better reconstruction is obtained with 2D-DWT compared to 1D-DWT, where the reconstructed data were stepwise. The 2D-DWT technique is therefore more appropriate for the highly correlated PortSimHigh data set.

The original and reconstructed signals generated with 2D-iDWT are plotted around the y=x line as shown in Figure 6.38. It is observed that all of the reconstructed data points are located around the y=x line representing that perfect reconstruction is retained in all of the representative 16 stock prices.

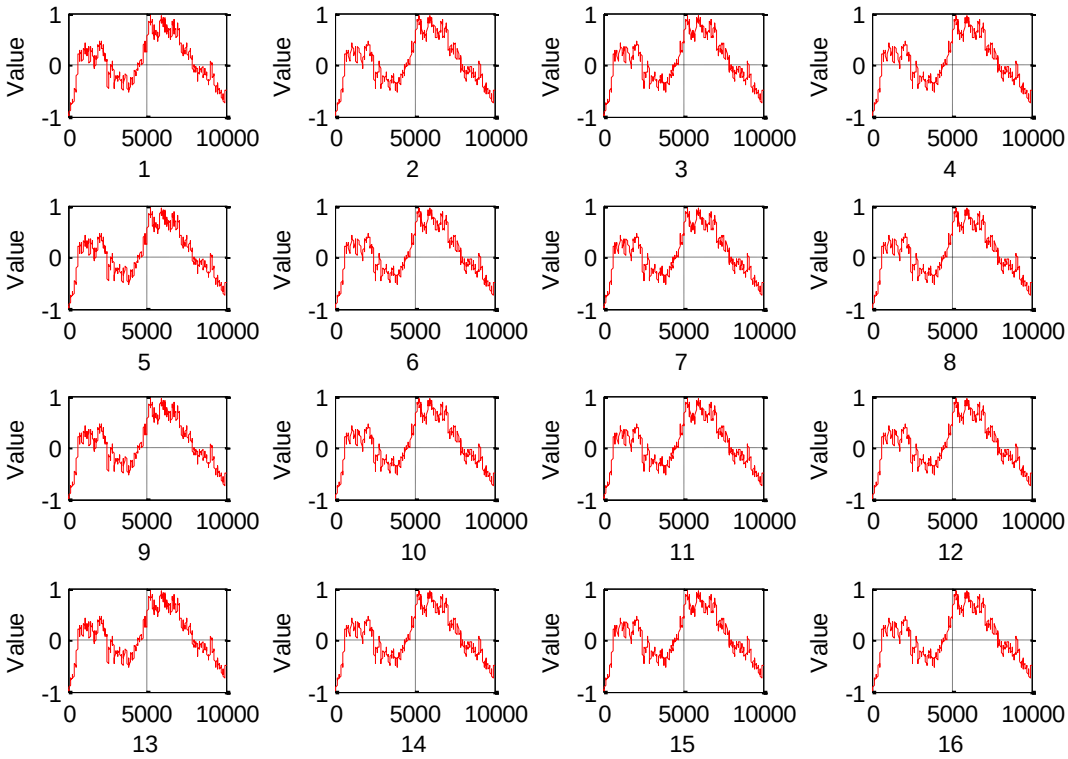

Figure 6.36. Reconstructed PortSimHigh Data Set with Inverse 2D-DWT.

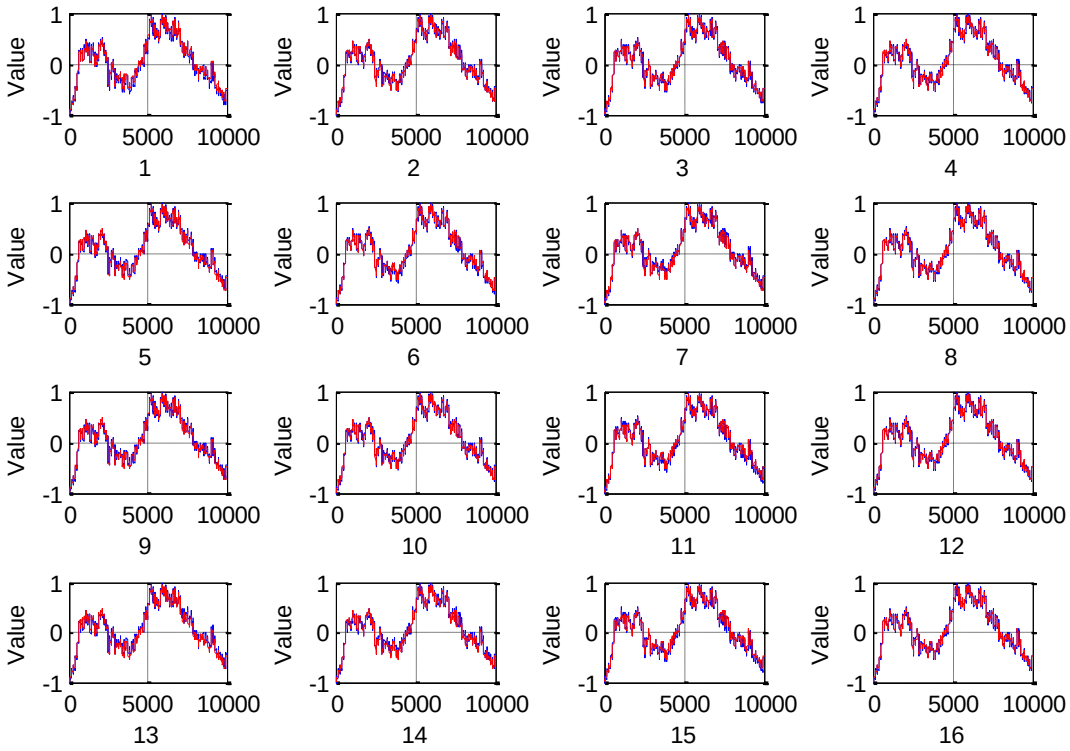

Figure 6.37. Original and Reconstructed Signals of the PortSimHigh Data Set with Inverse 2D-DWT.

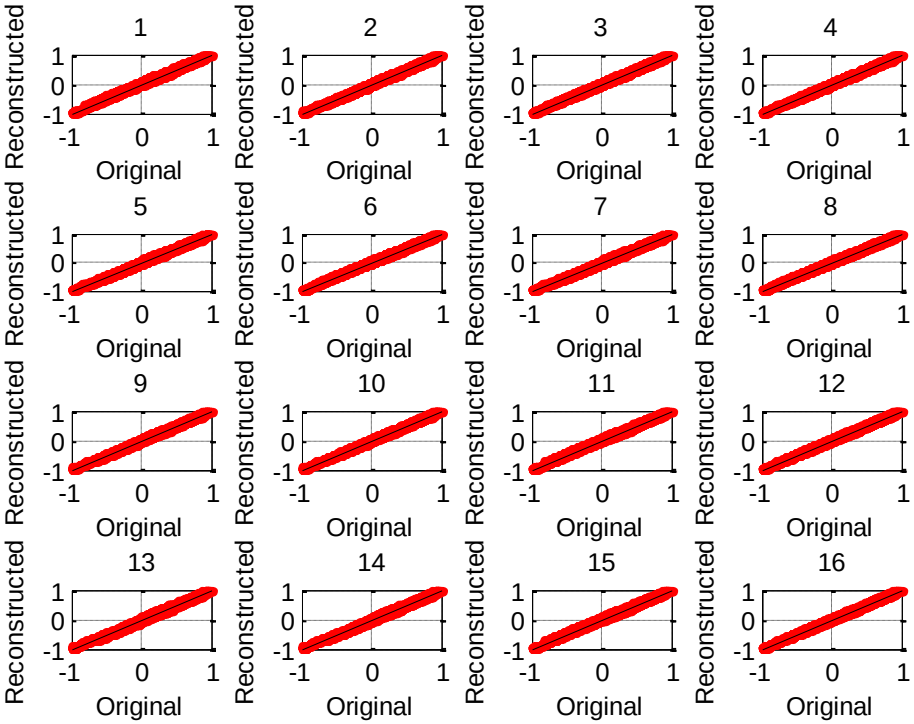

Figure 6.38. Reconstructed versus Original Signals of the PortSimHigh Data Set with Inverse 2D-DWT.

Reconstruction error norm values of the overall PortSimHigh data set calculated per data column are given in the bottom sub-window of Figure 6.39. Minimum error norm of 139.33 is obtained in the 60th column and the maximum error norm of 432.25 is obtained in the 62nd column of the PortSimHigh data set after applying 2D-DWT. As it can be seen from both the upper and the middle sub-windows of Figure 6.39, original signals are decoded thoroughly as reconstructed data points are scattered around the y=x line. It should also be mentioned that smaller reconstruction error norms (nearly three-fold smaller than those produced with 1D-DWT) and slightly higher compression ratios are generated with 2D-DWT, concluding that this technique is clearly superior to 1D-DWT when they are applied to the highly correlated PortSimHigh data set. However, 2D-DCT technique yielded much smaller reconstruction error norms (nearly 10-fold smaller than those produced with 2D-DWT) although compression levels were slightly smaller as demonstrated in Section 5.3.2. To sum up, 2D-DWT provides the highest compression, while 2D-DCT generates more perfect reconstruction when PortSimHigh data set is used.

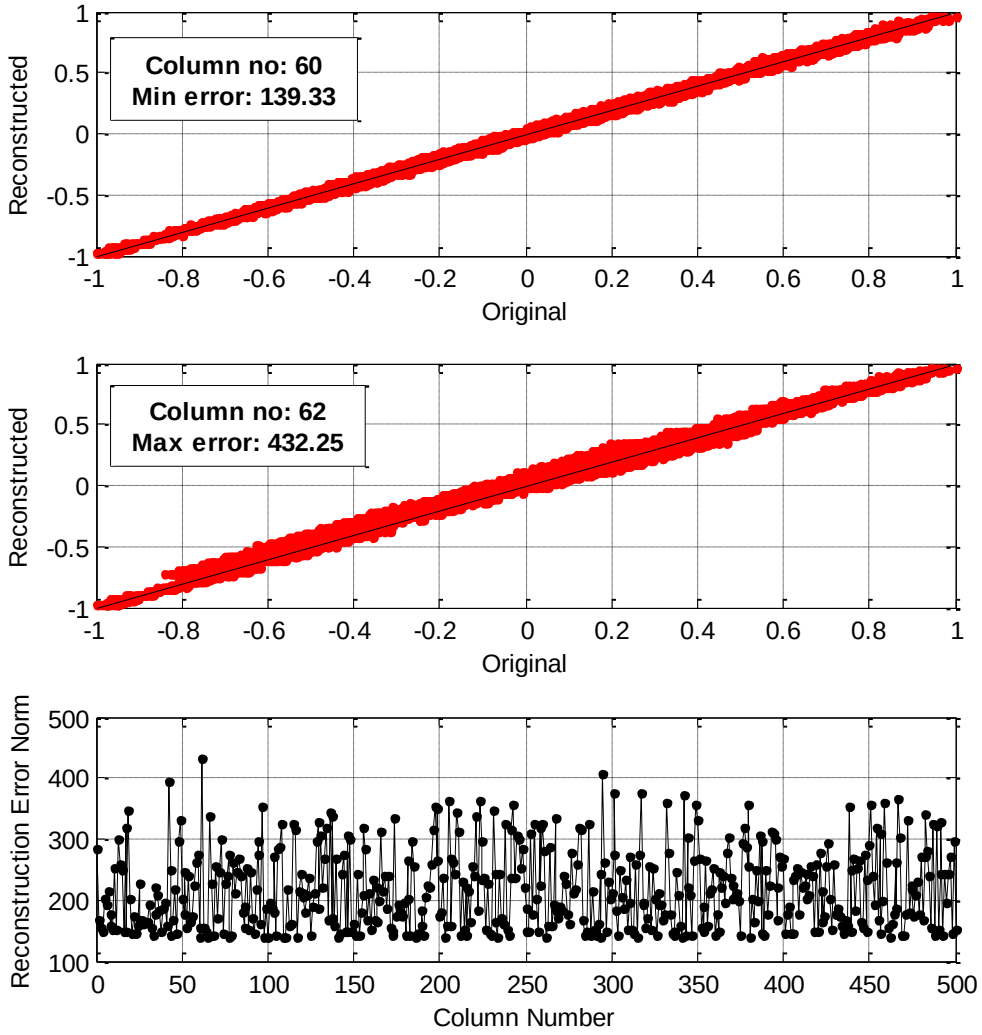

Figure 6.39. Reconstruction Error Norm Values of the PortSimHigh Data Set with Inverse 2D-DWT.

6.3.3. One Dimensional Discrete Wavelet Transform of the PortSimLow Data Set

In this section, 1D-DWT with 10-level decomposition and the wavelet type `db1` is applied to the overall PortSimLow data set (consisting of 500 stock prices), however, for illustration purposes, only the first 16 stock prices of the PortSimLow data set are presented. The first 16 stock prices were given before in Chapter 5 in scaled format with Figure 5.18.

Semilog-log (upper sub-window of Figure 6.40) and log-log (bottom sub-window of Figure 6.40) plots of the sorted absolute values of the 1D-DWT coefficients of the overall PortSimLow data set, which are padded into a vector of size 5002000 (the number of

coefficients is higher than the number of data points of 5000000, unlike 1D-DCT), and the threshold limit of 0.7485 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 6.40. Transform coefficients below the threshold limit (99.5% of the transform coefficients) which are identified clearly in the log-log plot, are set to zero and the number of nonzero coefficients kept is 25010, which is only 0.5% of the number of data points.

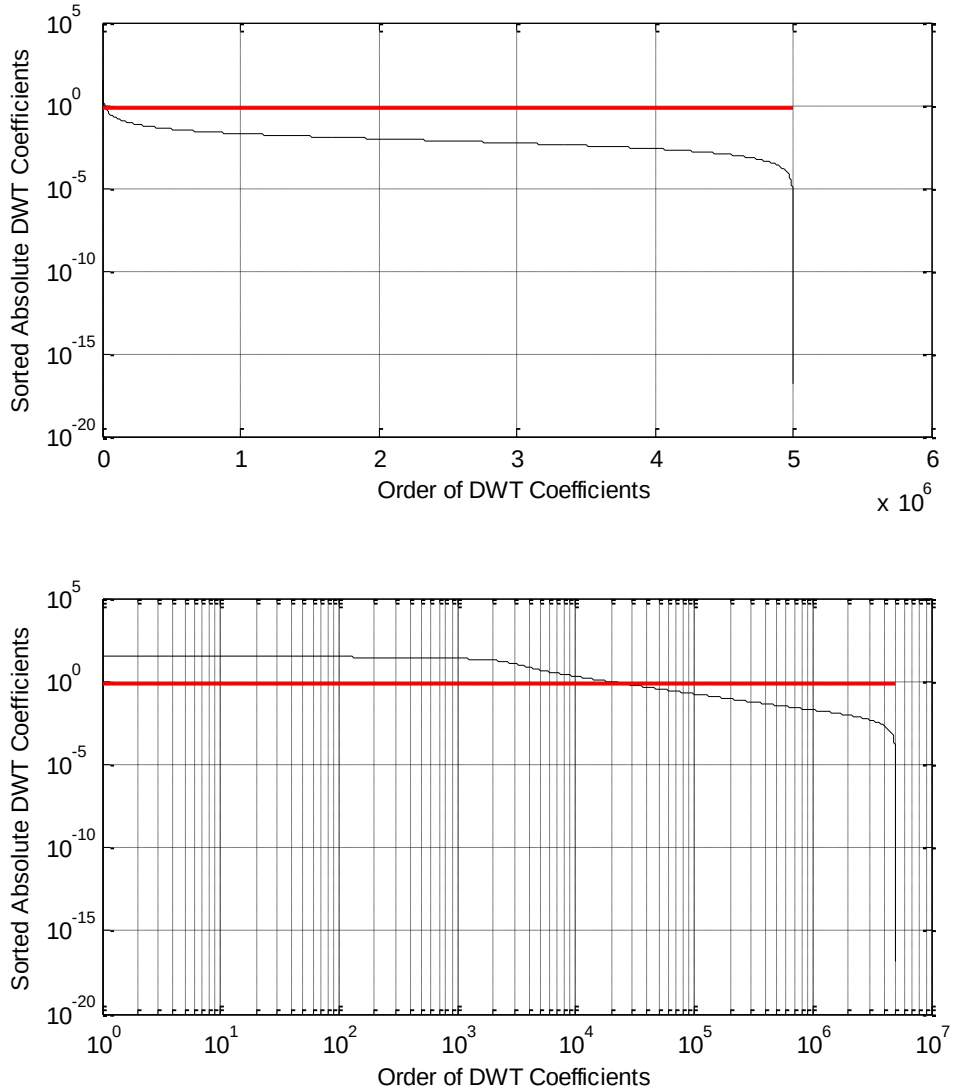

Figure 6.40. Semilog-log and Log-log Plots of Sorted Absolute 1D-DWT Coefficients of the Overall PortSimLow Data Set for α=99.5 %.

Sizes of the ZIP files of the original and encoded data sets are compared in Figure 6.41. It is seen that the ZIP file of the overall PortSimLow data set is nearly 50.9 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.93 MB improving

compression 54.6 times with 1D-DWT technique and taking the percentile value as 99.5% in thresholding step (same compression level generated with 1D-DCT).

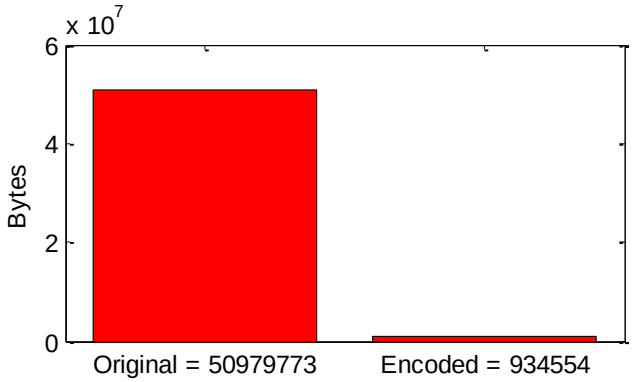

Figure 6.41. ZIP Compression Comparison of the Original and Encoded Overall PortSimLow Data Set for α=99.5 % with 1D-DWT.

Reconstructed data generated with 1D-iDWT (inverse DWT) and their overlay with the originals are shown in Figure 6.42 and Figure 6.43 respectively. It is seen that the major features of the stock prices including peaks and decreasing/increasing trends in the original data set are reconstructed accurately as shown in Figure 6.43 concluding that wavelet transform performs well in reconstructing sudden changes without any significant distortion although the decoded data set is stepwise.

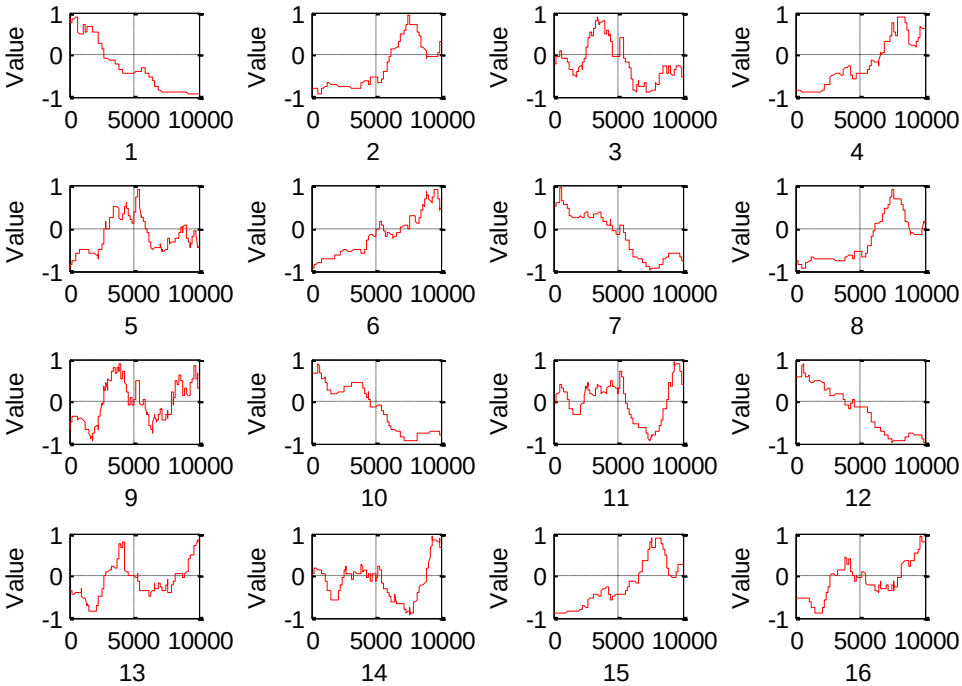

Figure 6.42. Reconstructed PortSimLow Data Set with Inverse 1D-DWT.

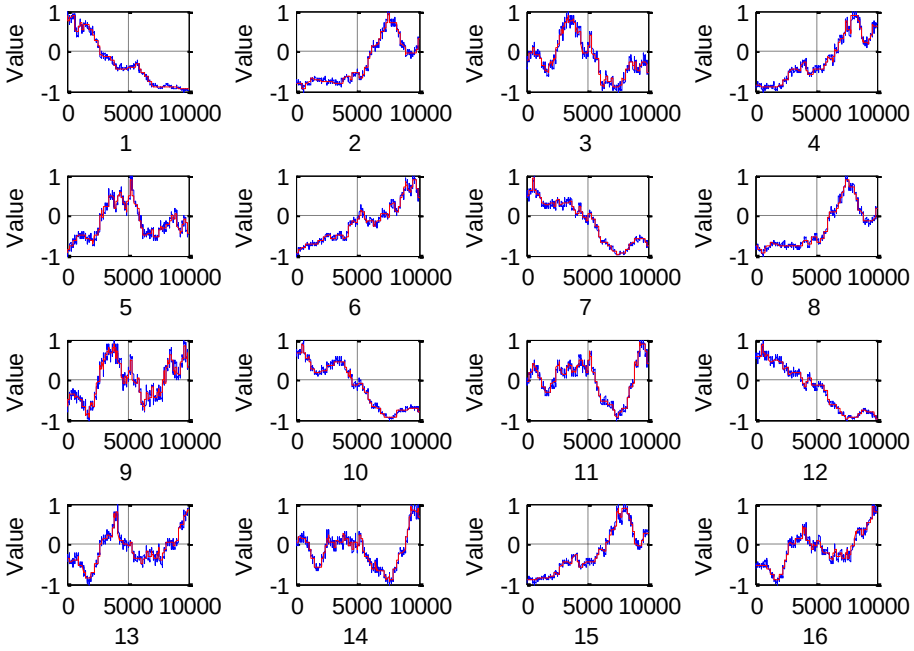

Figure 6.43. Original and Reconstructed Signals of the PortSimLow Data Set with Inverse 1D-DWT.

The original and reconstructed signals generated with 1D-iDWT are plotted around the y=x line in Figure 6.44. Reconstructed data points located around the y=x line reveals that all of the representative 16 stock prices are decoded without any deterioration.

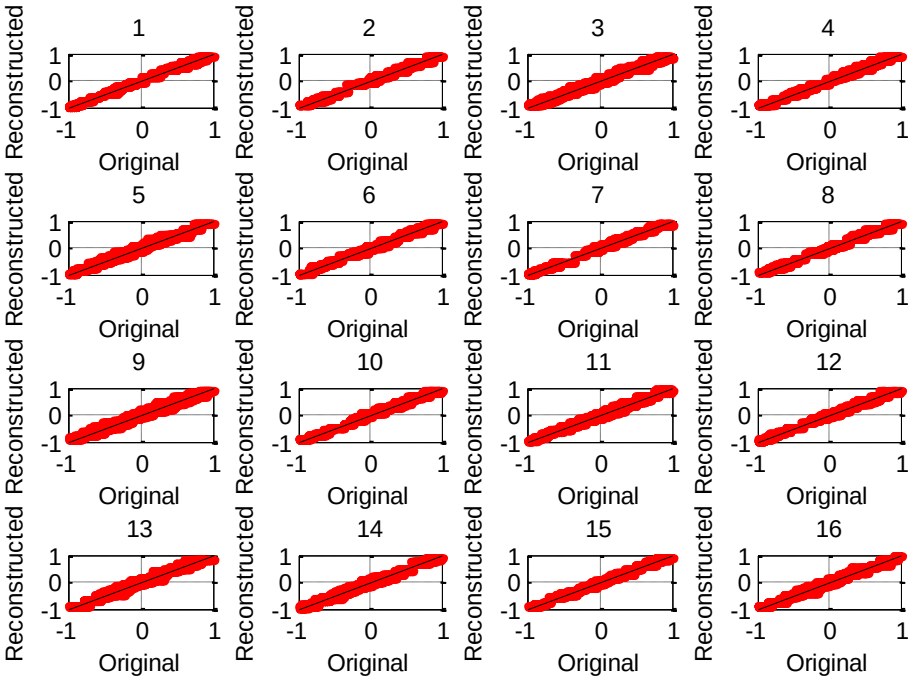

Figure 6.44. Reconstructed versus Original Signals of the PortSimLow Data Set with Inverse 1D-DWT.

Reconstruction error norm values of the overall PortSimLow data set calculated per data column are given in the bottom sub-window of Figure 6.45. Minimum error norm of 278.45 is obtained in the 20th column and the maximum error norm of 650.63 is obtained in the 122nd column of the PortSimLow data set. As it can be seen from the upper and the middle sub-windows of Figure 6.45, 20th column is reconstructed better than the 122nd column as most of the reconstructed data points of the 20th column are located around the y=x line. It can also be stated that slightly smaller reconstruction error norms were produced with 1D-DCT at the same compression level of 54.6 as demonstrated in Section 5.3.3.

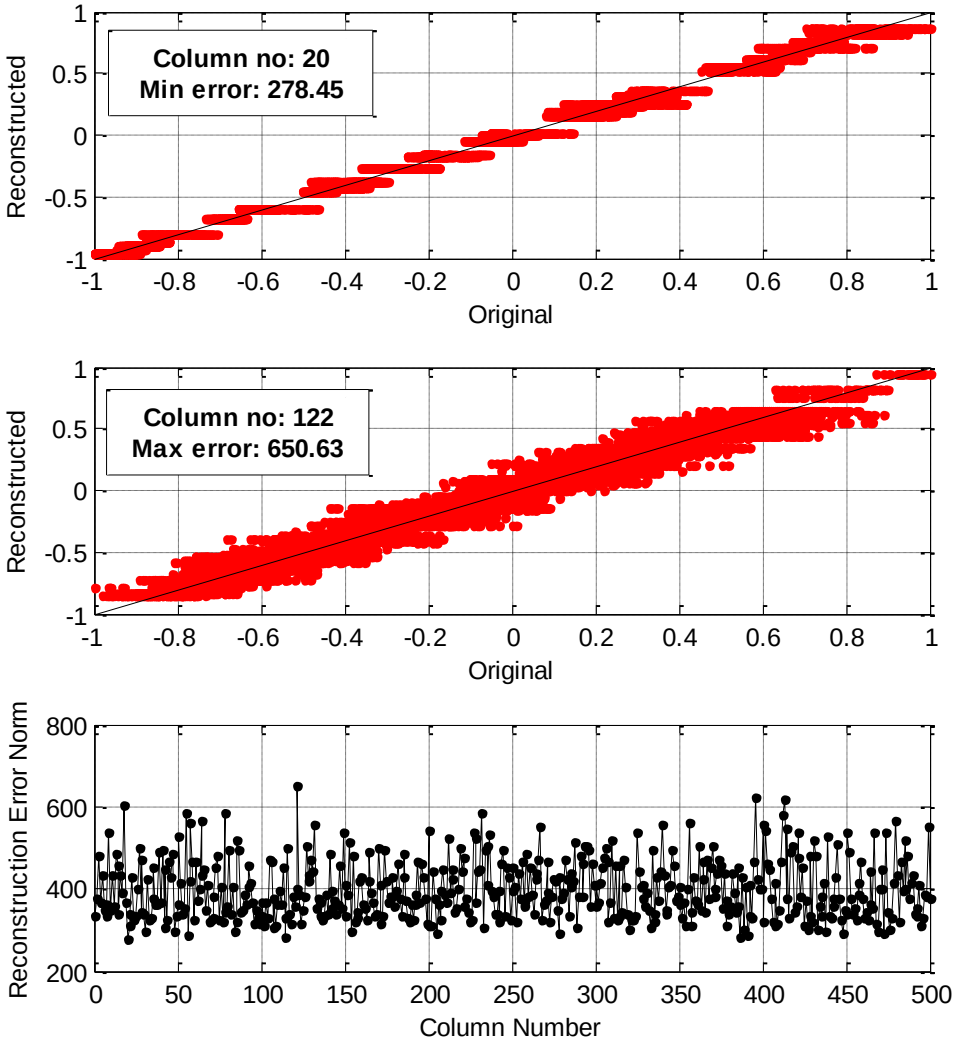

Figure 6.45. Reconstruction Error Norm Values of the PortSimLow Data Set with Inverse 1D-DWT.

6.3.4. Two Dimensional Discrete Wavelet Transform of the PortSimLow Data Set

In this section, 2D-DWT with 10-level decomposition and the wavelet type `db1` is applied to the overall PortSimLow data set (consisting of 500 stock prices), however, for illustration purposes, only the first 16 stock prices of the PortSimLow data set are presented.

Semilog-log (upper sub-window of Figure 6.46) and log-log (bottom sub-window of Figure 6.46) plots of the sorted absolute values of the 2D-DWT coefficients of the overall PortSimLow data set, which are concatenated into a vector of size 5003830 (the number of coefficients is higher than the number of data points of 5000000, unlike 2D-DCT), and the threshold limit of 2.3662 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 6.46. The coefficients below the threshold limit (4978811 coefficients), which are identified clearly in the log-log plot, are set to zero for compression, in other words 99.5% of the transform coefficients become zero. The number of nonzero coefficients kept is 25019, which is only 0.5% of the number of data points. In addition, the threshold limit is much larger than that found with 1D-DWT, that is 0.7485. Hence, higher reconstruction error norms are expected with 2D-DWT for the less correlated PortSimLow data set.

Sizes of the ZIP files of the original and encoded data sets are compared as it is seen from Figure 6.47, the ZIP file of the original data set is nearly 50.5 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.84 MB. Thus, compression can be increased 60.4 times by applying 2D-DWT technique and taking the percentile value as 99.5% in thresholding step. It can also be stated that the compression ratio obtained with 2D-DWT is larger than that obtained with 1D-DWT, which was 54.6 as mentioned in Section 6.3.3. In addition, higher compression is yielded with 2D-DWT compared to 2D-DCT with which compression ratio had been calculated as 54.9 in Section 5.3.4.

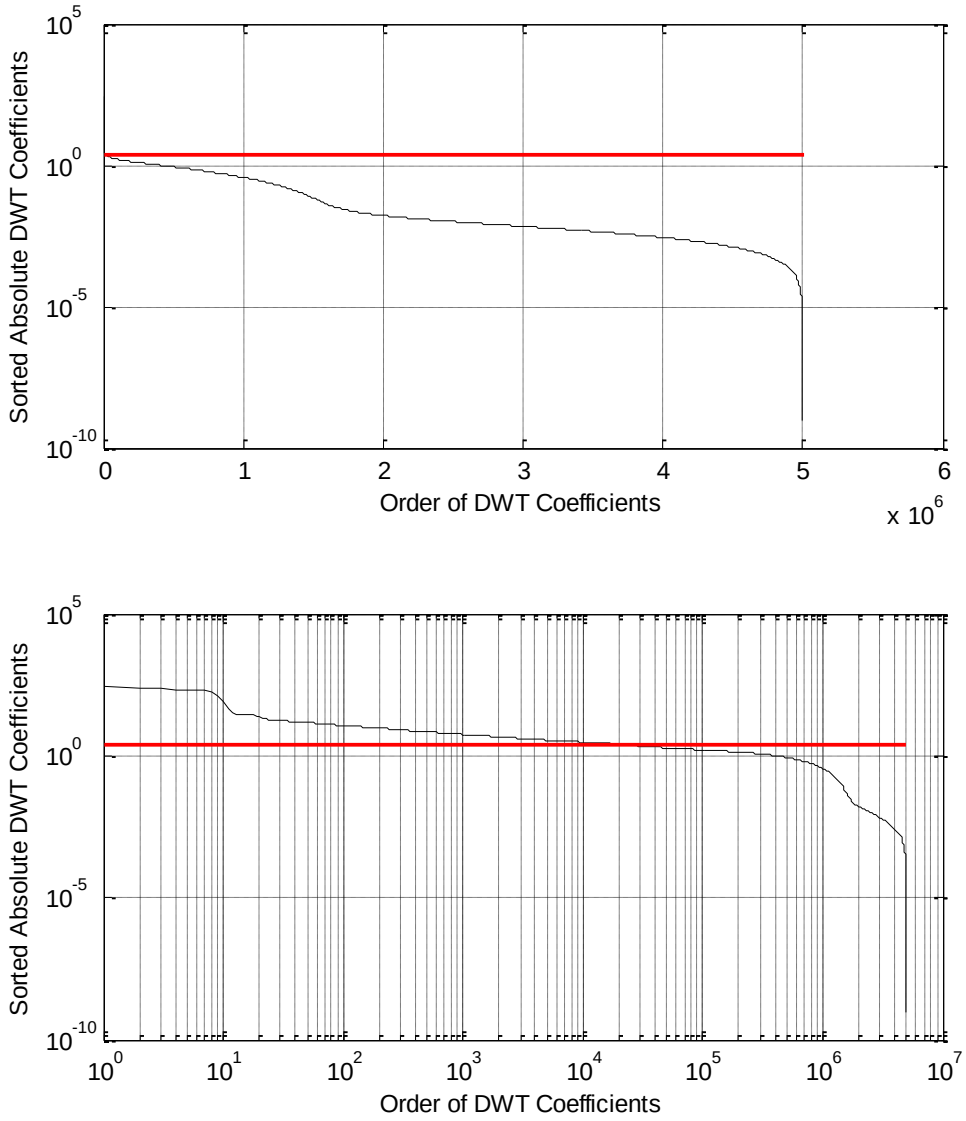

Figure 6.46. Semilog-log and Log-log Plots of Sorted Absolute 2D-DWT Coefficients of the Overall PortSimLow Data Set for α=99.5 %.

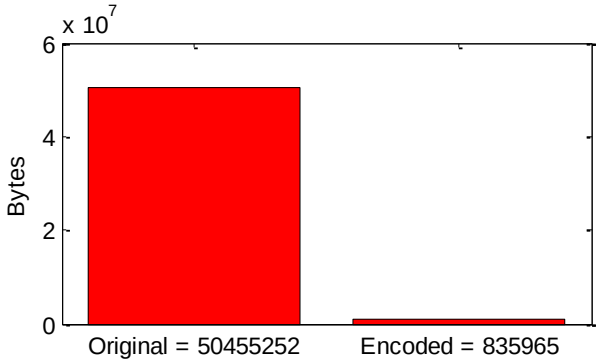

Figure 6.47. ZIP Compression Comparison of the Original and Encoded Overall PortSimLow Data Set for α=99.5 % with 2D-DWT.

Reconstructed data generated with 2D-iDWT (inverse DWT) and their overlay with the originals are shown in Figure 6.48 and Figure 6.49 respectively. It is seen that reconstructed data cannot represent the original data accurately as illustrated in Figure 6.49 concluding that there is a significant loss in the decoded data set (except prices of the 13th and 16th stocks). This situation may stem from the discarded approximation coefficients representing the major trends in the original data. It is obvious that much better reconstruction was obtained with 1D-DWT compared to 2D-DWT concluding that 2D-DWT technique is not adequate for the reconstruction of the less correlated PortSimLow data set.

In Figure 6.50, the original and reconstructed signals produced with 2D-iDWT are plotted around the y=x line. It can be said that almost each of the reconstructed signals is completely different from the original ones (except prices of the 13th and 16th stocks) as most of the data points are located around the y=0 line.

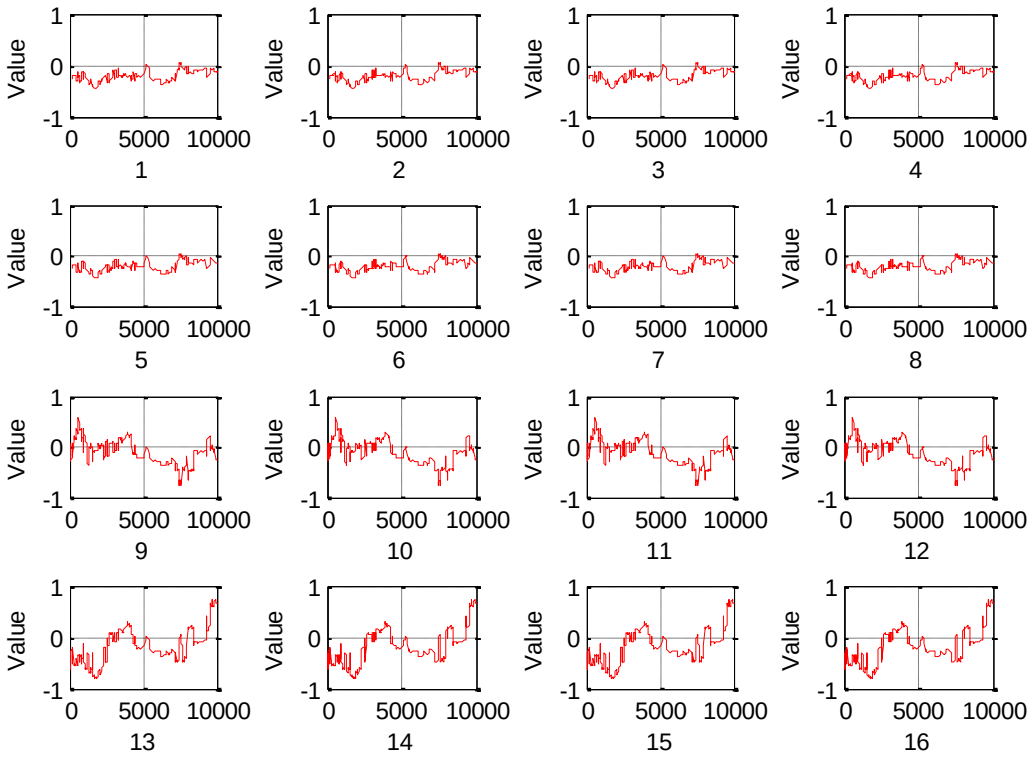

Figure 6.48. Reconstructed PortSimLow Data Set with Inverse 2D-DWT.

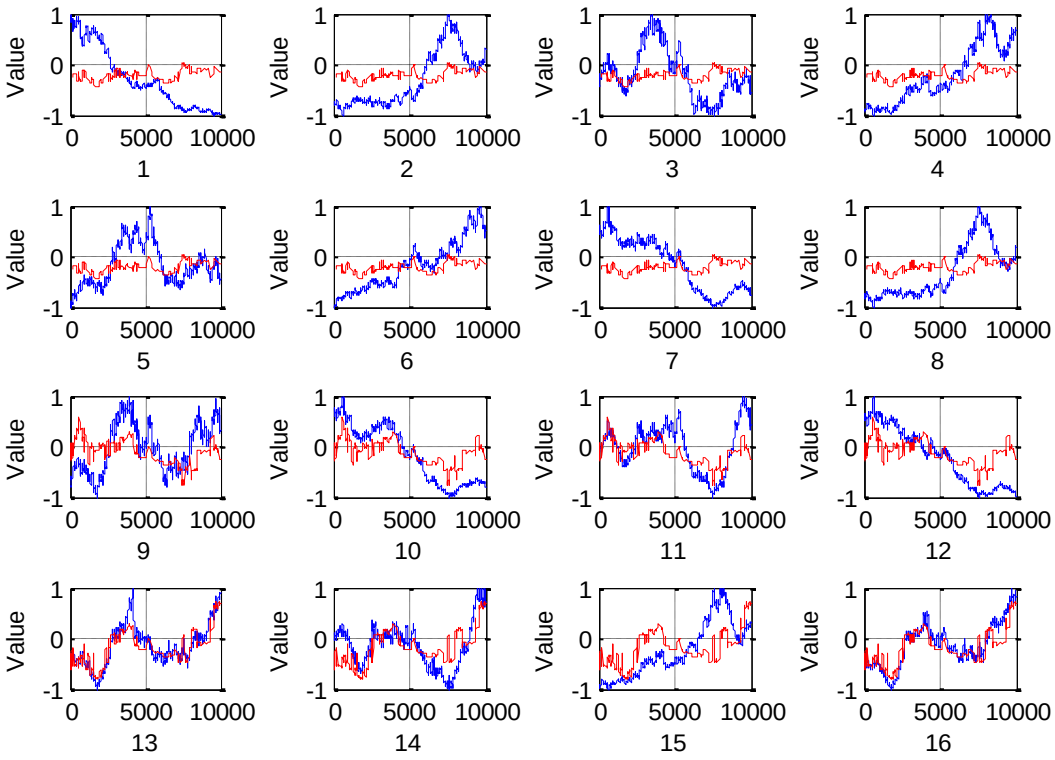

Figure 6.49. Original and Reconstructed Signals of the PortSimLow Data Set with Inverse 2D-DWT.

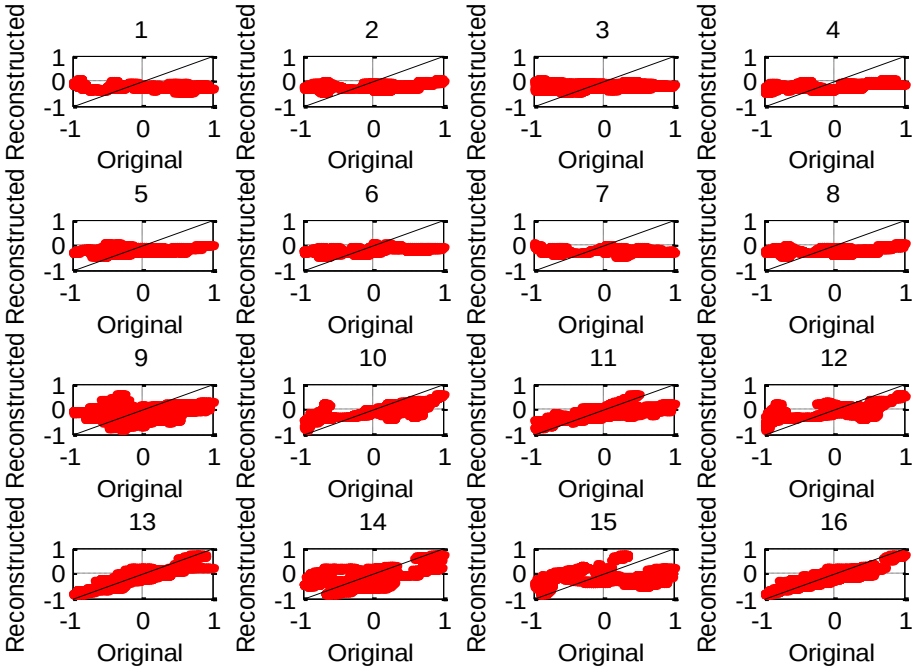

Figure 6.50. Reconstructed versus Original Signals of the PortSimLow Data Set with Inverse 2D-DWT.

Error norms between original and reconstructed data sets are calculated per data column. Reconstruction error norm values of the overall PortSimLow data set are given in the bottom sub-window of Figure 6.51. Minimum error norm of 1290.5 is obtained in the 138th column and the maximum error norm of 7651.3 is obtained in the 461th column of the PortSimLow data set after applying 2D-DWT. It can be concluded that 1D-DWT performs well in reconstruction, whereas 2D-DWT achieves the maximum compression although this technique fails in reconstruction (error norms are nearly 10-fold higher than those produced with 1D-DWT) when less correlated PortSimLow data set is used. Actually, it was decided that 2D-DWT was superior to 1D-DWT in both compression and reconstruction when highly correlated PortSimHigh data set was used in Section 6.3.2. Therefore, it can be said that the efficacy of the 2D-DWT technique depends on the characteristics of the data set.

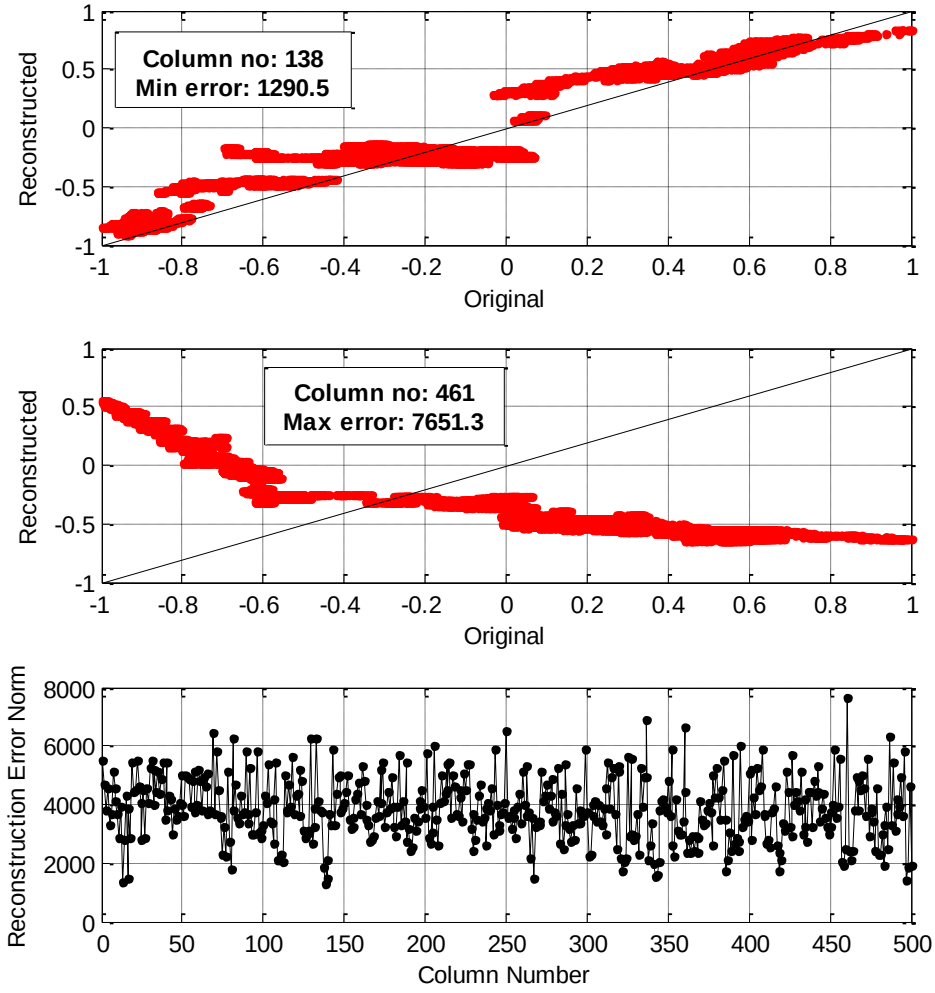

Figure 6.51. Reconstruction Error Norm Values of the PortSimLow Data Set with Inverse 2D-DWT.

6.3.5. One Dimensional Discrete Wavelet Transform of the SELDI-TOF MS Data Set

In this section, 1D-DWT with eight-level decomposition and the wavelet type `db4` is applied to the overall SELDI-TOF MS data set (consisting of six ovarian cancer samples). The complete scaled intensities were given before in Chapter 5 with Figure 5.35.

Semilog-log (upper sub-window of Figure 6.52) and log-log (bottom sub-window of Figure 6.52) plots of the sorted absolute values of the 1D-DWT coefficients of the overall SELDI-TOF MS data set, which are padded into a vector of size 2028234 (the number of coefficients is higher than the number of data points of 2027928, unlike 1D-DCT), and the threshold limit of 0.0815 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 6.52. Transform coefficients below the threshold limit (99.5% of the transform coefficients) which are identified clearly in the log-log plot, are set to zero and the number of nonzero coefficients kept is 10141, which is only 0.5% of the number of data points.

Sizes of the ZIP files of the original and encoded data sets are compared in Figure 6.53. It is seen that the ZIP file of the overall SELDI-TOF MS data set is nearly 20.6 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.34 MB improving compression 60.1 times with 1D-DWT technique and taking the percentile value as 99.5% in thresholding step (slightly higher than the compression level generated with 1D-DCT, that was 58.6, as mentioned in Section 5.3.5).

Reconstructed data generated with 1D-iDWT (inverse DWT) and their overlay with the originals are shown in Figure 6.54 and Figure 6.55 respectively. It is seen that perfect reconstruction is retained (observed in all of the six cancer samples) while de-noising is also achieved as illustrated in Figure 6.55. It should also be mentioned that there were some distortions in the magnitudes of the major peaks in the decoded SELDI-TOF MS data set with 1D-iDCT, whereas these sudden changes can be reconstructed thoroughly with 1D-iDWT.

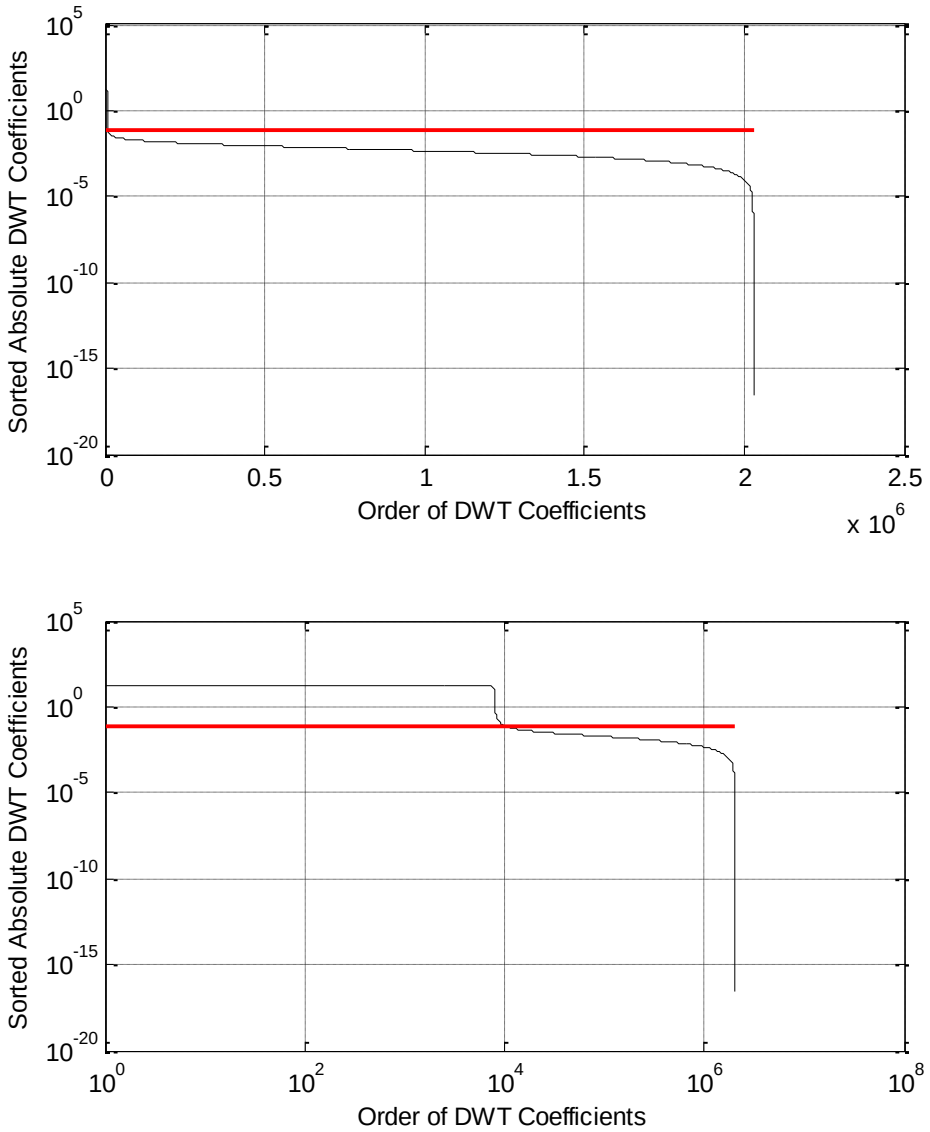

Figure 6.52. Semilog-log and Log-log Plots of Sorted Absolute 1D-DWT Coefficients of the Overall SELDI-TOF MS Data Set for α=99.5 %.

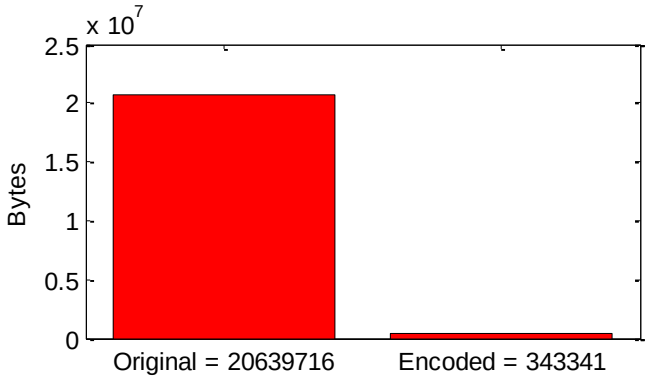

Figure 6.53. ZIP Compression Comparison of the Original and Encoded Overall SELDI-TOF MS Data Set for α=99.5 % with 1D-DWT.

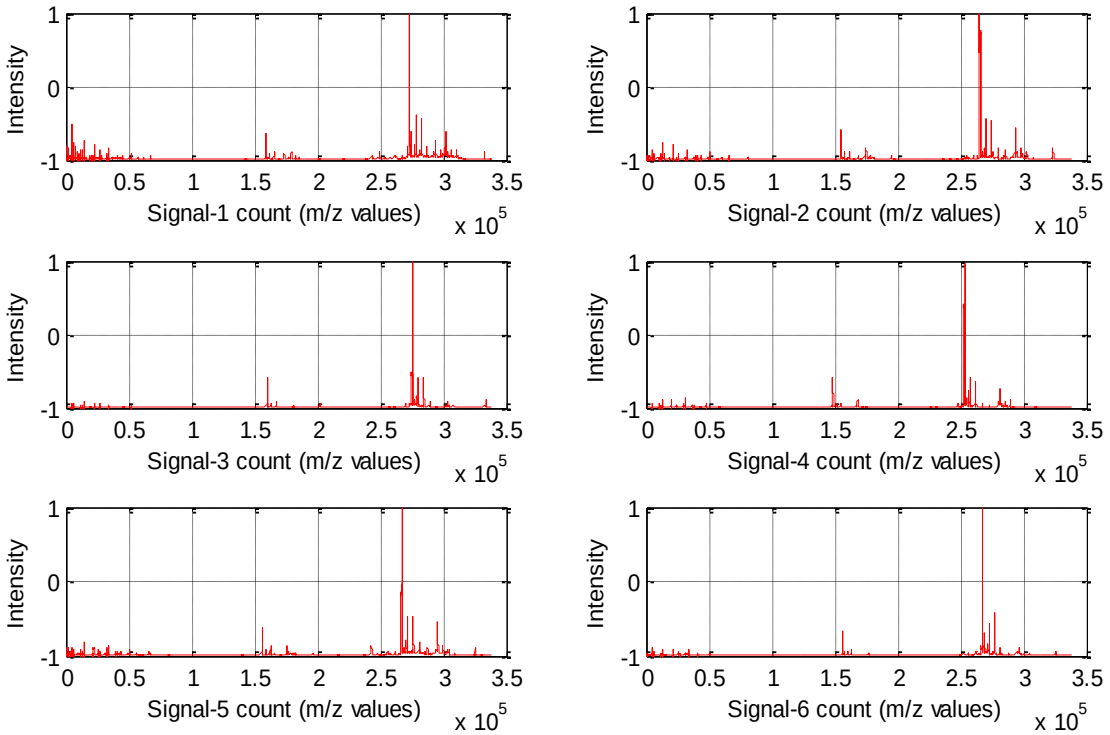

Figure 6.54. Reconstructed SELDI-TOF MS Data Set with Inverse 1D-DWT.

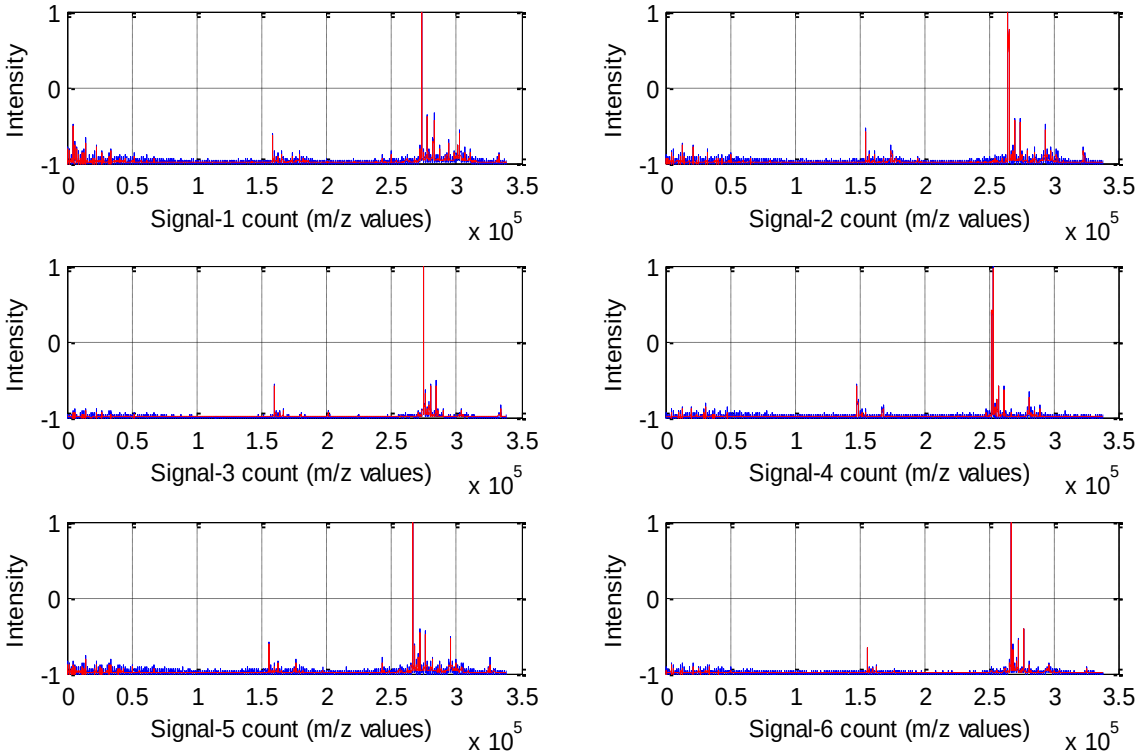

Figure 6.55. Original and Reconstructed Signals of the SELDI-TOF MS Data Set with Inverse 1D-DWT.

The original and reconstructed signals generated with 1D-iDWT are plotted around the y=x line in Figure 6.56. Original signals are decoded without any detectable distortion (observed in all of the six cancer samples) as reconstructed data points are located on the y=x line. However, scaled intensities having the magnitudes over 0.5 and the magnitudes between [-1,-0.5] (regions where sharp peaks occurred) could not have been reconstructed thoroughly with 1D-iDCT as illustrated in Section 5.3.5.

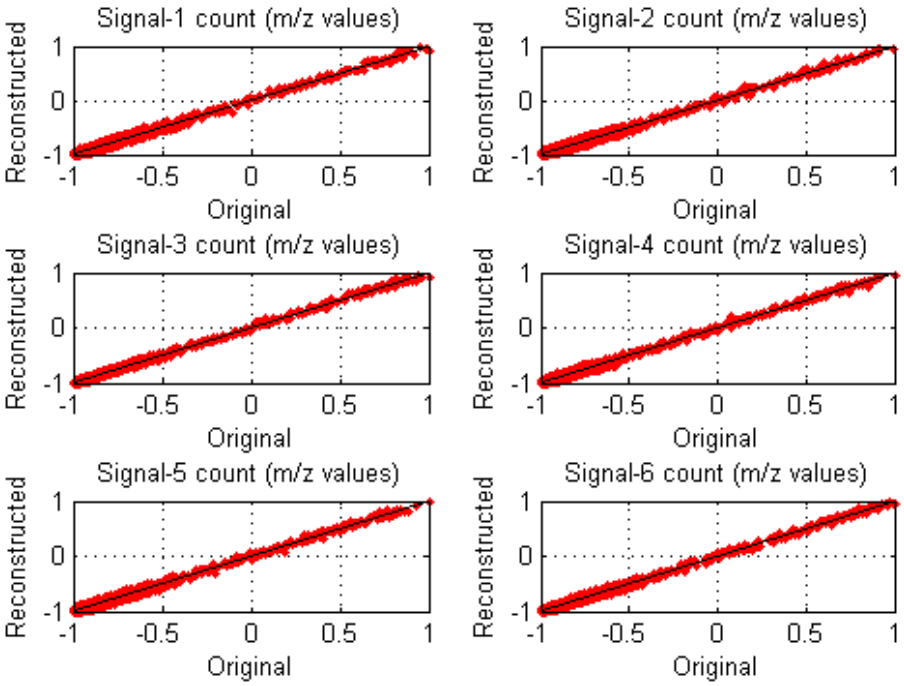

Figure 6.56. Reconstructed versus Original Signals of the SELDI-TOF MS Data Set with Inverse 1D-DWT.

Reconstruction error norm values of the overall SELDI-TOF MS data set calculated per data column are given in the bottom sub-window of Figure 6.57. Minimum error norm of 1128 is obtained in the third column and the maximum error norm of 2885.9 is obtained in the fifth column of the SELDI-TOF MS data set. It can also be stated that smaller reconstruction error norms are generated with 1D-DWT compared to 1D-DCT at the same compression level of 60 as DWT is superior to DCT in analyzing noisy data sets containing localized changes.

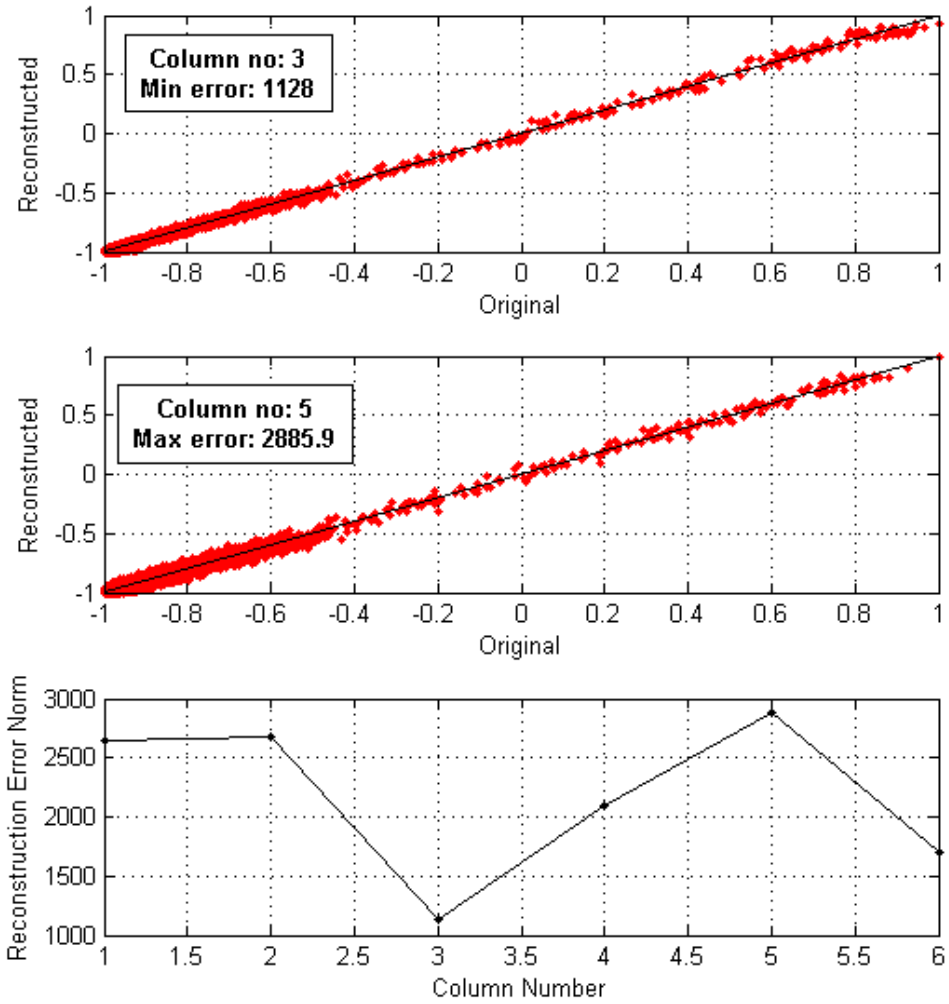

Figure 6.57. Reconstruction Error Norm Values of the SELDI-TOF MS Data Set with Inverse 1D-DWT.

6.3.6. Two Dimensional Discrete Wavelet Transform of the SELDI-TOF MS Data Set

In this section, 2D-DWT with eight-level decomposition and the wavelet type `db4` is applied to the overall SELDI-TOF MS data set (consisting of six ovarian cancer samples).

Semilog-log (upper sub-window of Figure 6.58) and log-log (bottom sub-window of Figure 6.58) plots of the sorted absolute values of the 2D-DWT coefficients of the overall SELDI-TOF MS data set, which are concatenated into a vector of size 6068778 (the number of coefficients is nearly three-fold higher than the number of data points of 2027928, unlike 2D-DCT), and the threshold limit of 0.2510 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in Figure 6.58. The coefficients below the threshold limit

(6038434 coefficients), which are identified clearly in the log-log plot, are set to zero for compression, in other words 99.5% of the transform coefficients become zero. The number of nonzero coefficients kept is 30343, which is only 0.5% of the number of data points. In addition, the threshold limit is much larger than that found with 1D-DWT, that is 0.0815. Hence, higher reconstruction error norms are expected with 2D-DWT for the SELDI-TOF MS data set.

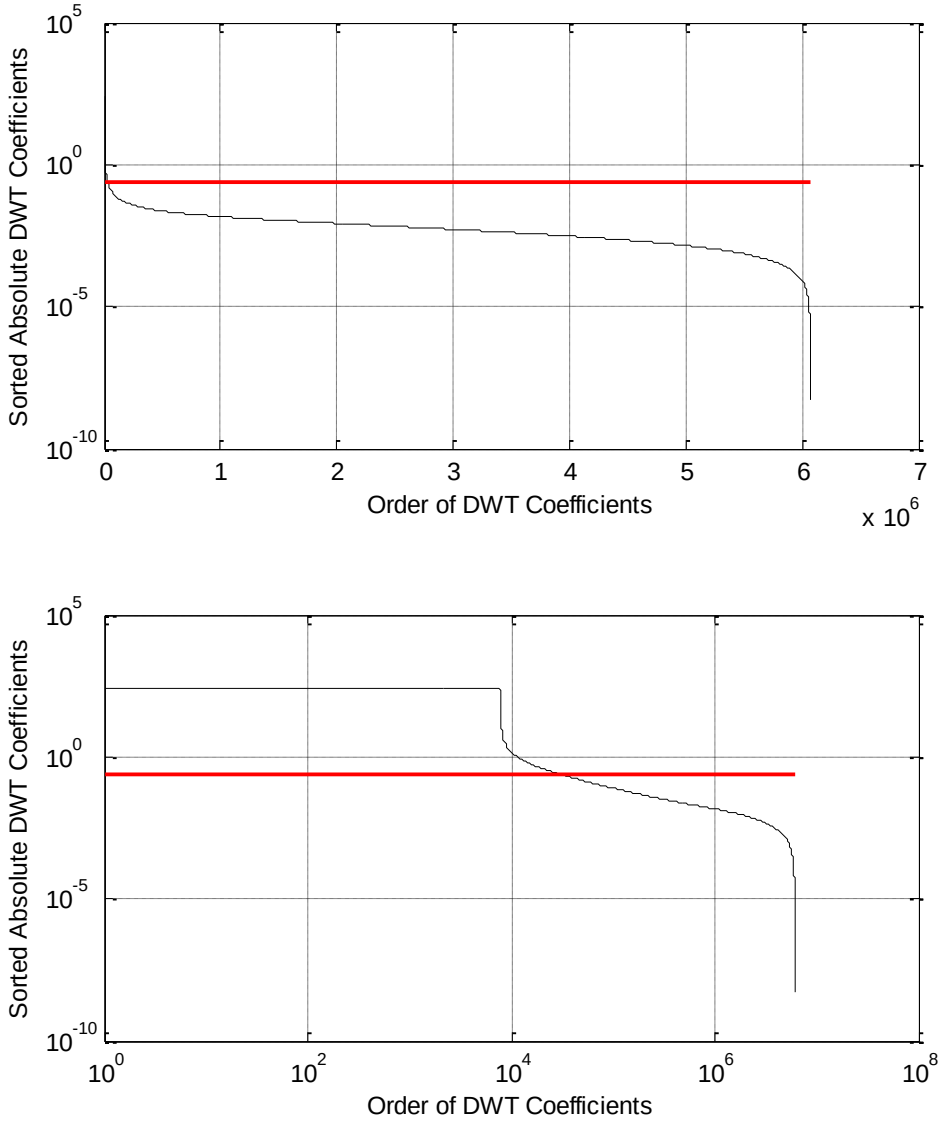

Figure 6.58. Semilog-log and Log-log Plots of Sorted Absolute 2D-DWT Coefficients of the Overall SELDI-TOF MS Data Set for α=99.5 %.

Sizes of the ZIP files of the original and encoded data sets are compared as it is seen from Figure 6.59, the ZIP file of the original data set is nearly 61.7 MB (nearly three-fold higher than the ZIP file in Section 6.3.5 as the number of 2D-DWT coefficients to be

compressed is much larger than the number of 1D-DWT coefficients), whereas the ZIP file of the filtered transformed data set is nearly 1.1 MB. Thus, compression can be increased 59.6 times by applying 2D-DWT technique and taking the percentile value as 99.5% in thresholding step. It can be concluded that there is not much difference between the compression ratios obtained with 1D-DWT and 2D-DWT. However, it is obvious that 2D-DWT is not the appropriate technique for the data sets having large number of data rows as the number of wavelet coefficients (approximation and detail coefficients in three orientations as mentioned in Section 6.2) increases in each decomposition level resulting a dimensionality increase which is not favored in data compression applications.

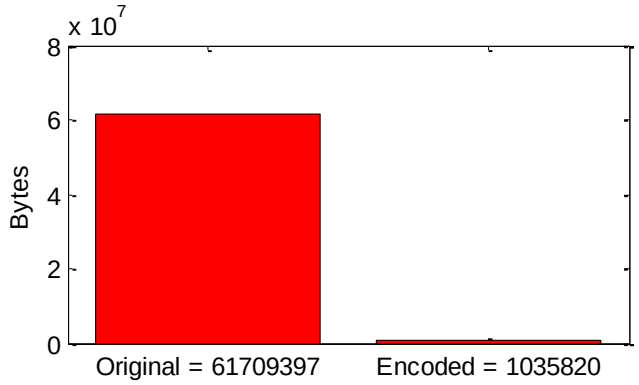

Figure 6.59. ZIP Compression Comparison of the Original and Encoded Overall SELDI-TOF MS Data Set for α=99.5 % with 2D-DWT.

Reconstructed data generated with 2D-iDWT (inverse DWT) and their overlay with the originals are shown in Figure 6.60 and Figure 6.61 respectively. It is seen that reconstructed data set is similar to the original data set as illustrated in Figure 6.61 concluding that there is not a detectable loss in the decoded signals of the six cancer samples. It can also be stated that more perfect reconstruction is generated with 1D-iDWT compared to 2D-iDWT.

In Figure 6.62, the original and reconstructed signals produced with 2D-iDWT are plotted around the y=x line. It can be said that almost each of the reconstructed signals is scattered around the y=x line indicating that synthesized data set is similar to the original data set.

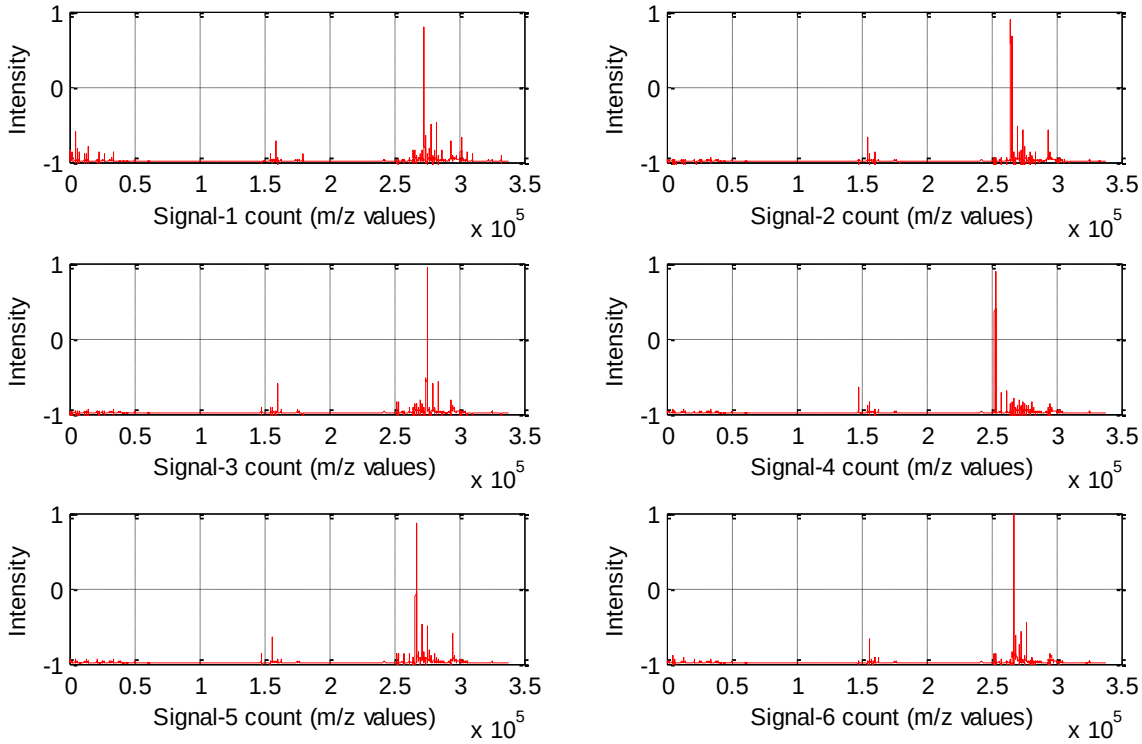

Figure 6.60. Reconstructed SELDI-TOF MS Data Set with Inverse 2D-DWT.

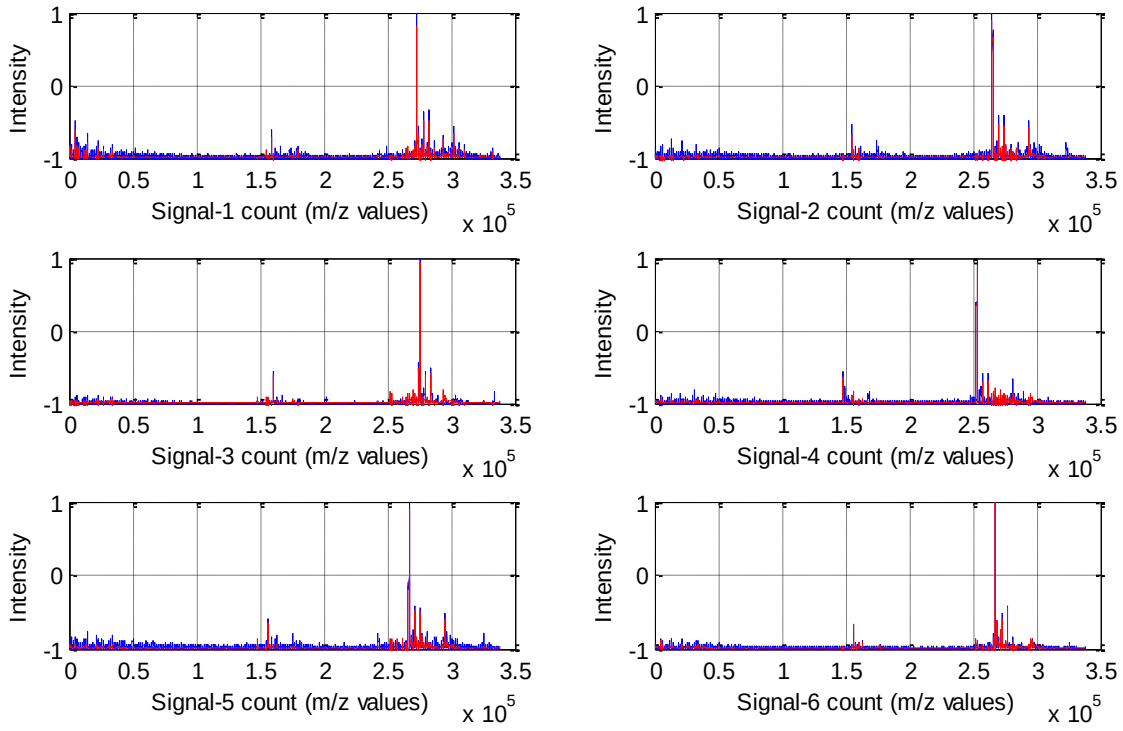

Figure 6.61. Original and Reconstructed Signals of the SELDI-TOF MS Data Set with Inverse 2D-DWT.

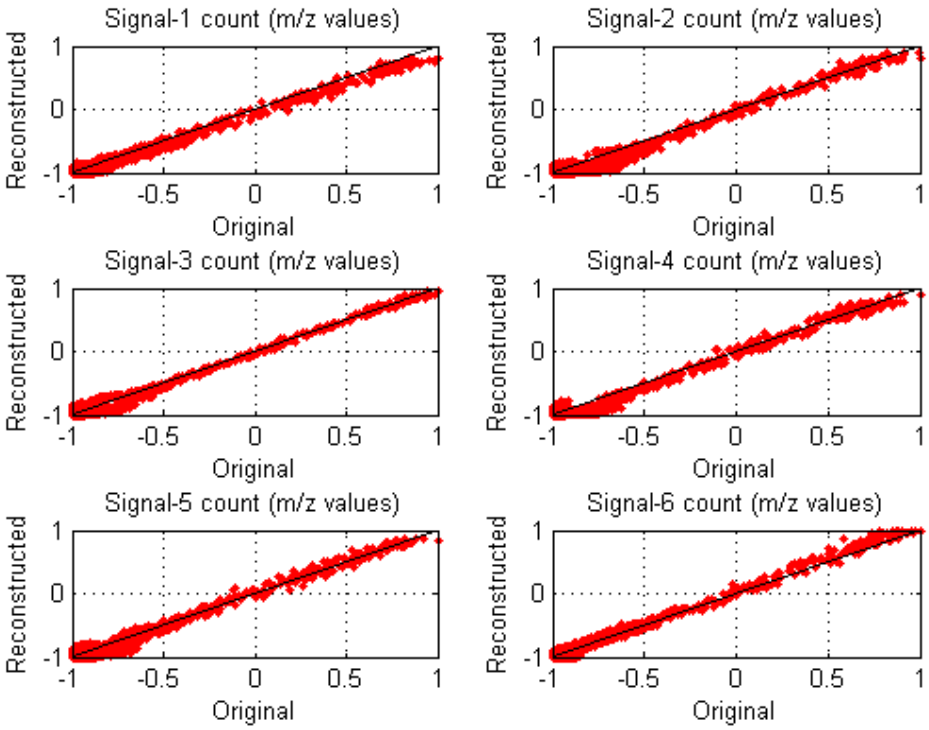

Figure 6.62. Reconstructed versus Original Signals of the SELDI-TOF MS Data Set with Inverse 2D-DWT.

Reconstruction error norm values of the overall SELDI-TOF MS data set are given in the bottom sub-window of Figure 6.63. Minimum error norm of 2045.9 is obtained in the sixth column and the maximum error norm of 4066.6 is obtained in the first column of the SELDI-TOF MS data set after applying 2D-DWT. It can be concluded that 1D-DWT performs well in reconstruction compared to 2D-DWT (higher error norms are produced with 2D-DWT) although these two techniques provide the same compression level when highly uncorrelated SELDI-TOF MS data set is used. It should also be stated that 2D-DWT is not the appropriate method for data series having a large number of data rows as the number of wavelet coefficients increases in each decomposition level leading to a significant increase in dimension, and thus, making compression less viable.

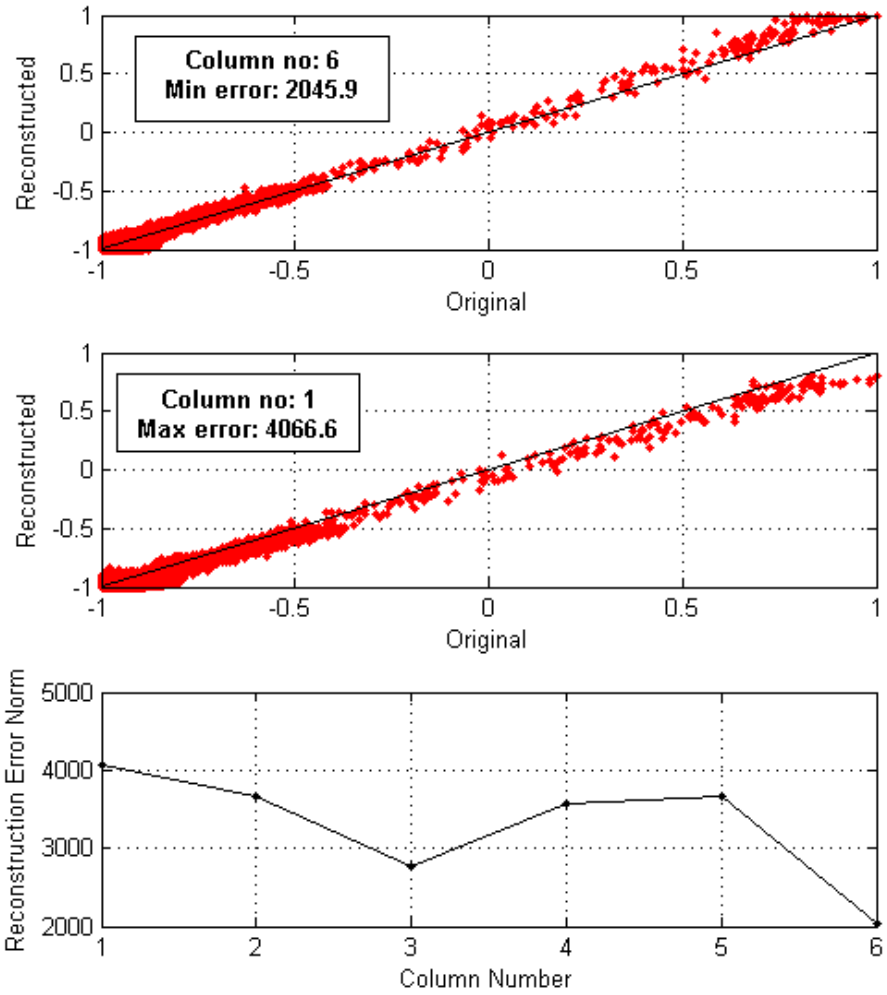

Figure 6.63. Reconstruction Error Norm Values of the SELDI-TOF MS Data Set with Inverse 2D-DWT.

6.3.7. One Dimensional Discrete Wavelet Transform of the TEP Data Set

In this section, 1D-DWT with four-level decomposition and the wavelet type `sym4` is applied to the overall TEP data set (consisting of 41 measurements). Overall output signals of the TEP including all of the 41 measured variables were given before in Chapter 5 in scaled format with Figure 5.52.

Semilog-log (upper sub-window of Figure 6.64) and log-log (bottom sub-window of Figure 6.64) plots of the sorted absolute values of the 1D-DWT coefficients of the overall TEP data set, which are padded into a vector of size 2051107 (the number of coefficients is higher than the number of data points of 2050041, unlike 1D-DCT), and the threshold limit of 2.0796 denoted by the horizontal line, specified by taking the percentile of the frequency

distribution of the transform coefficients as 99.5% are given in Figure 6.64. Transform coefficients below the threshold limit (99.5% of the transform coefficients) which are identified clearly in the log-log plot, are set to zero and the number of nonzero coefficients kept is 10256, which is only 0.5% of the number of data points.

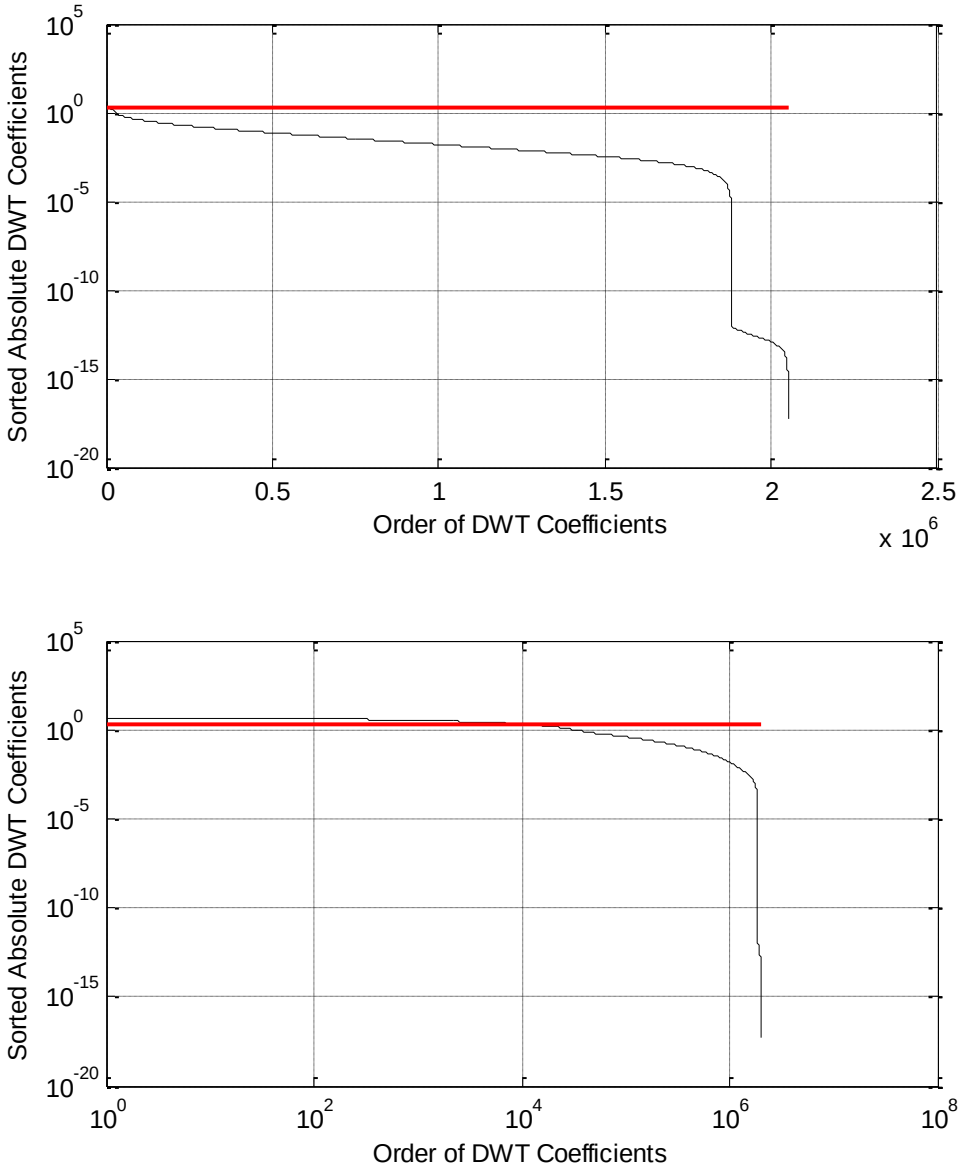

Figure 6.64. Semilog-log and Log-log Plots of Sorted Absolute 1D-DWT Coefficients of the Overall TEP Data Set for α=99.5 %.

Sizes of the ZIP files of the original and encoded data sets are compared in Figure 6.65. It is seen that the ZIP file of the overall TEP data set is nearly 20 MB, whereas the ZIP file of the filtered transformed data set is nearly 0.36 MB improving compression 55.4 times with 1D-DWT technique and taking the percentile value as 99.5% in thresholding

step (corresponding to the same compression level generated with 1D-DCT as mentioned in Section 5.3.7).

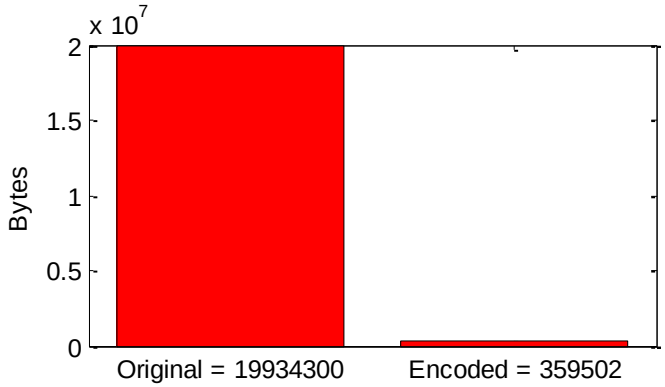

Figure 6.65. ZIP Compression Comparison of the Original and Encoded Overall TEP Data Set for α=99.5 % with 1D-DWT.

Reconstructed data generated with 1D-iDWT (inverse DWT) and their overlay with the originals are shown in Figure 6.66 and Figure 6.67 respectively. It can be mentioned that reconstructed data cannot represent the original data accurately concluding that 1D-iDWT fails in reconstructing the highly uncorrelated TEP data set containing level jumps and noisy measurements. Only a few process events such as important peak points, upward/downward shifts (observed in measurements one and four) and decreasing/increasing trends (observed in measurements 28, 34 and 39) occurred due to the consecutive fault disturbances can be followed from the reconstructed data set as illustrated in Figure 6.67. However, there was not any deterioration in information content in the decoded TEP data set with 1D-iDCT while the loss of the irrelevant data was also achieved in noisy measurements as mentioned in Section 5.3.7.

The original and reconstructed signals generated with 1D-iDWT are plotted around the y=x line in Figure 6.68. It is seen that original signals cannot be reconstructed accurately with 1D-iDWT, whereas reconstructed signals (except measurements consisting of almost pure noise) produced with 1D-iDCT were very close to the original signals as illustrated in Section 5.3.7.

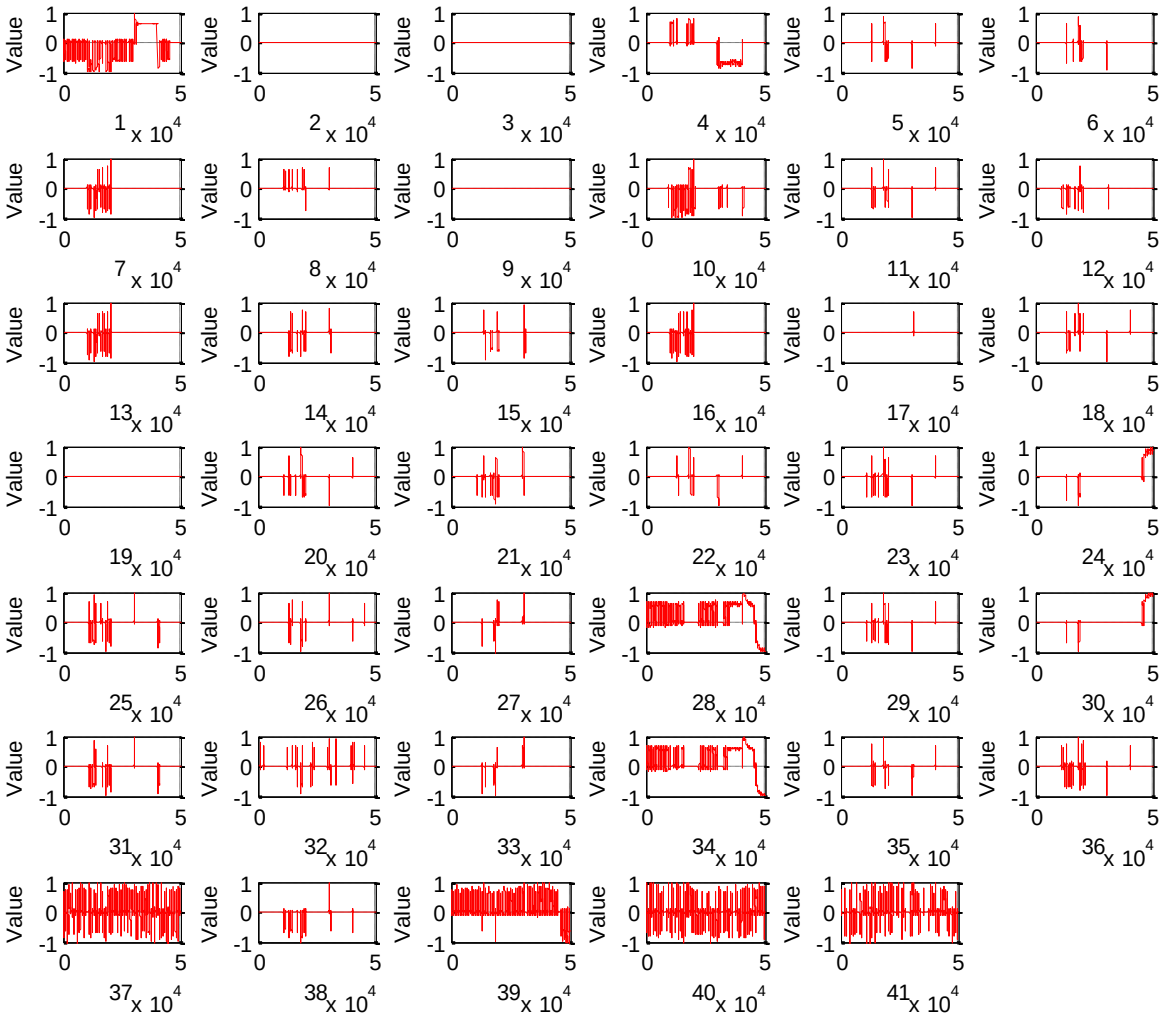

Figure 6.66. Reconstructed TEP Data Set with Inverse 1D-DWT.

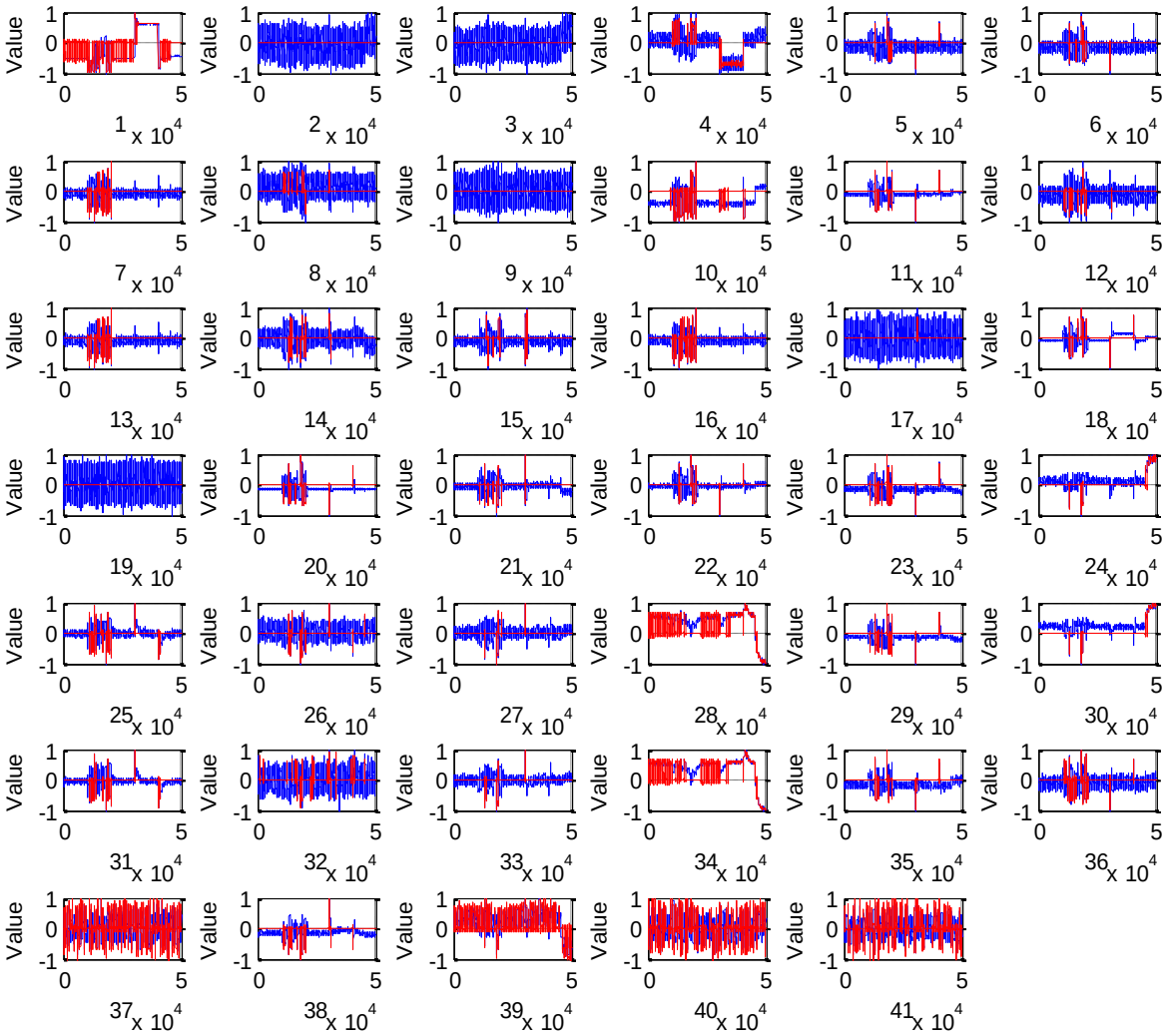

Figure 6.67. Original and Reconstructed Signals of the TEP Data Set with Inverse 1D-DWT.

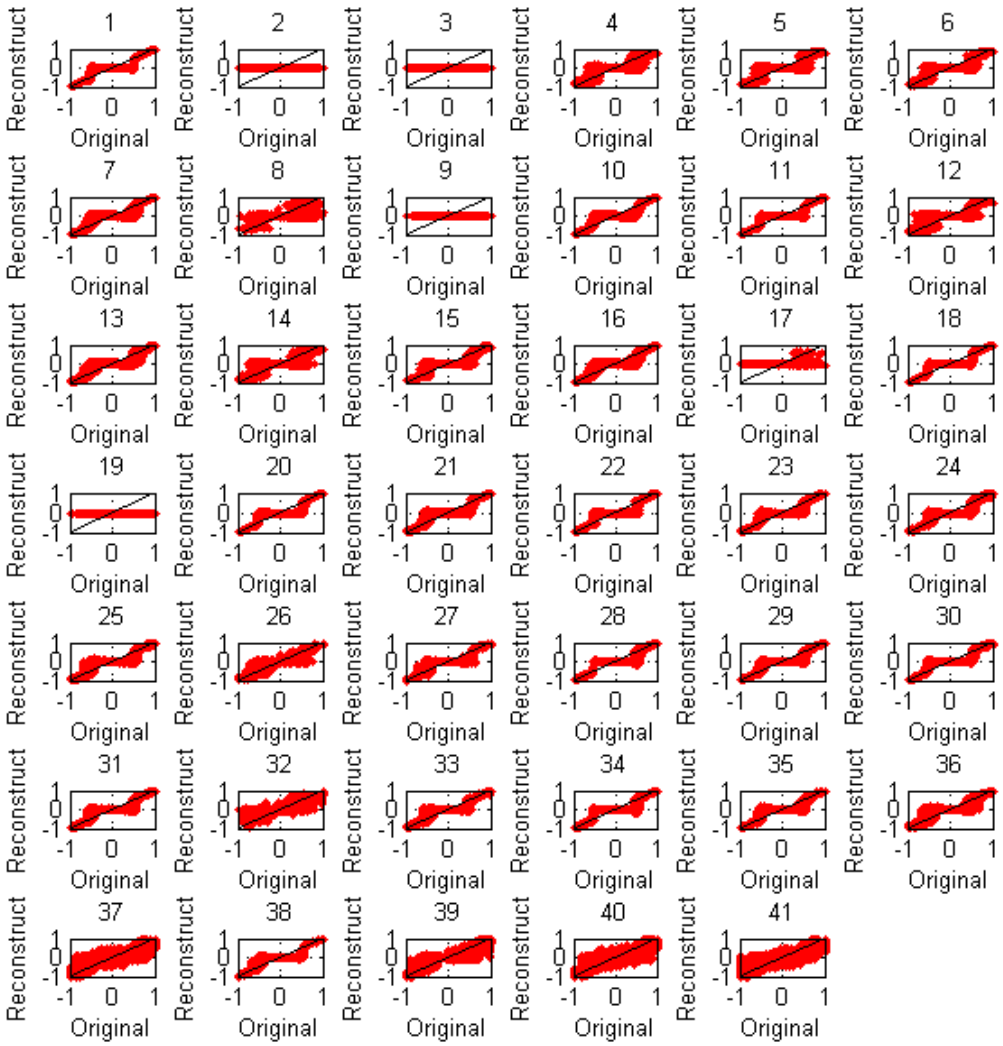

Figure 6.68. Reconstructed versus Original Signals of the TEP Data Set with Inverse 1D-DWT.

Reconstruction error norm values of the overall TEP data set calculated per data column are given in the bottom sub-window of Figure 6.69. Minimum error norm of 4234.7 is obtained in the 22nd column and the maximum error norm of 15805 is obtained in the 10th column of the TEP data set. It can be concluded that higher reconstruction error norms are generated with 1D-DWT compared to 1D-DCT (error norms are nearly two-fold higher than those generated with 1D-DCT) at the same compression level of 55.5. Actually, it was decided that 1D-DWT performs well in reconstruction (especially reconstructing sharp peaks) compared to 1D-DCT when highly uncorrelated SELDI-TOF MS data set was used in Section 6.3.5 as DWT is superior to DCT in analyzing noisy data sets containing localized changes. However, DCT should be the preferred technique when highly uncorrelated TEP data set containing level jumps and noisy measurements is used.

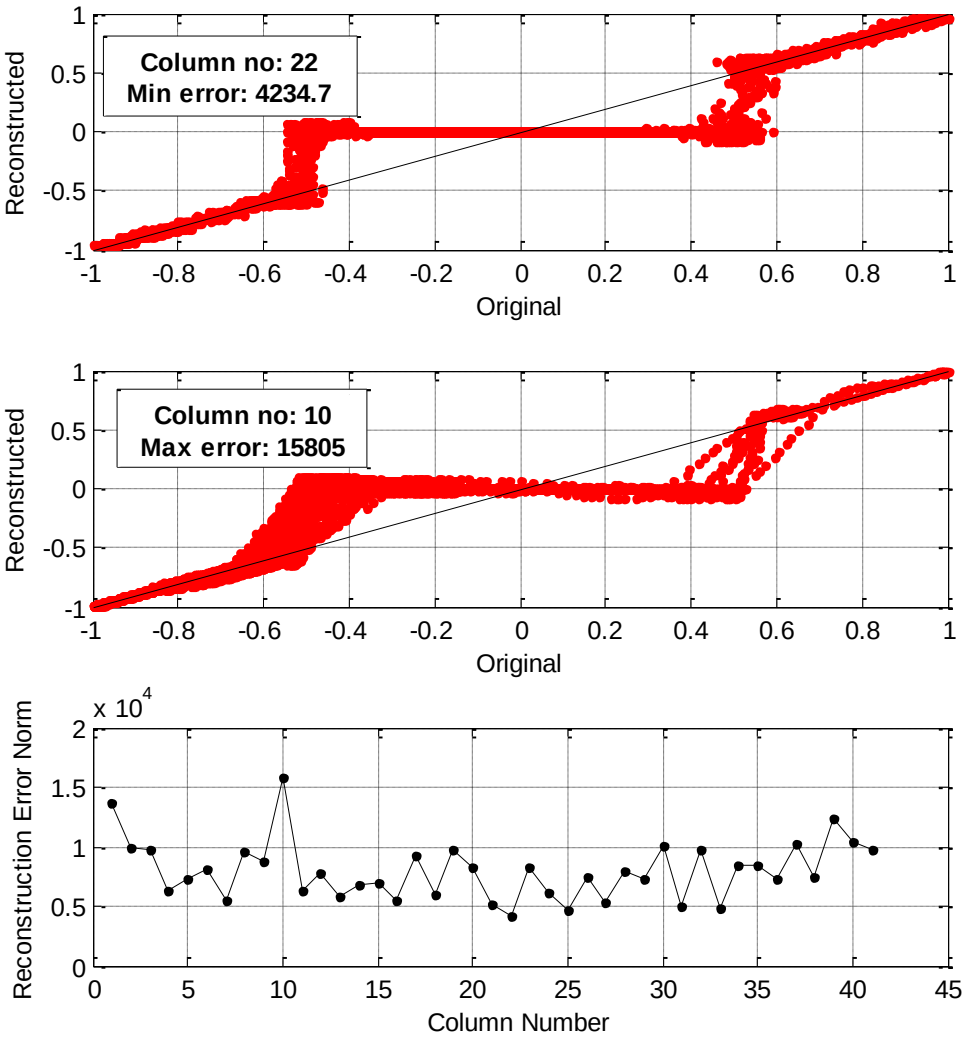

Figure 6.69. Reconstruction Error Norm Values of the TEP Data Set with Inverse 1D-DWT.

6.3.8. Two Dimensional Discrete Wavelet Transform of the TEP Data Set

In this section, 2D-DWT with four-level decomposition and the wavelet type `sym4` is applied to the overall TEP data set (consisting of 41 measurements).

Semilog-log (upper sub-window of Figure 6.70) and log-log (bottom sub-window of Figure 6.70) plots of the sorted absolute values of the 2D-DWT coefficients of the overall TEP data set, which are concatenated into a vector of size 2682177 (the number of coefficients is much higher than the number of data points of 2050041, unlike 2D-DCT), and the threshold limit of 2.1153 denoted by the horizontal line, specified by taking the percentile of the frequency distribution of the transform coefficients as 99.5% are given in

Figure 6.70. The coefficients below the threshold limit (2668766 coefficients), which are identified clearly in the log-log plot, are set to zero for compression, in other words 99.5% of the transform coefficients become zero. The number of nonzero coefficients kept is 13411, which is only 0.5% of the number of data points. In addition, the threshold limit is slightly larger than that found with 1D-DWT, that is 2.0796. Hence, higher reconstruction error norms are expected with 2D-DWT for the TEP data set.

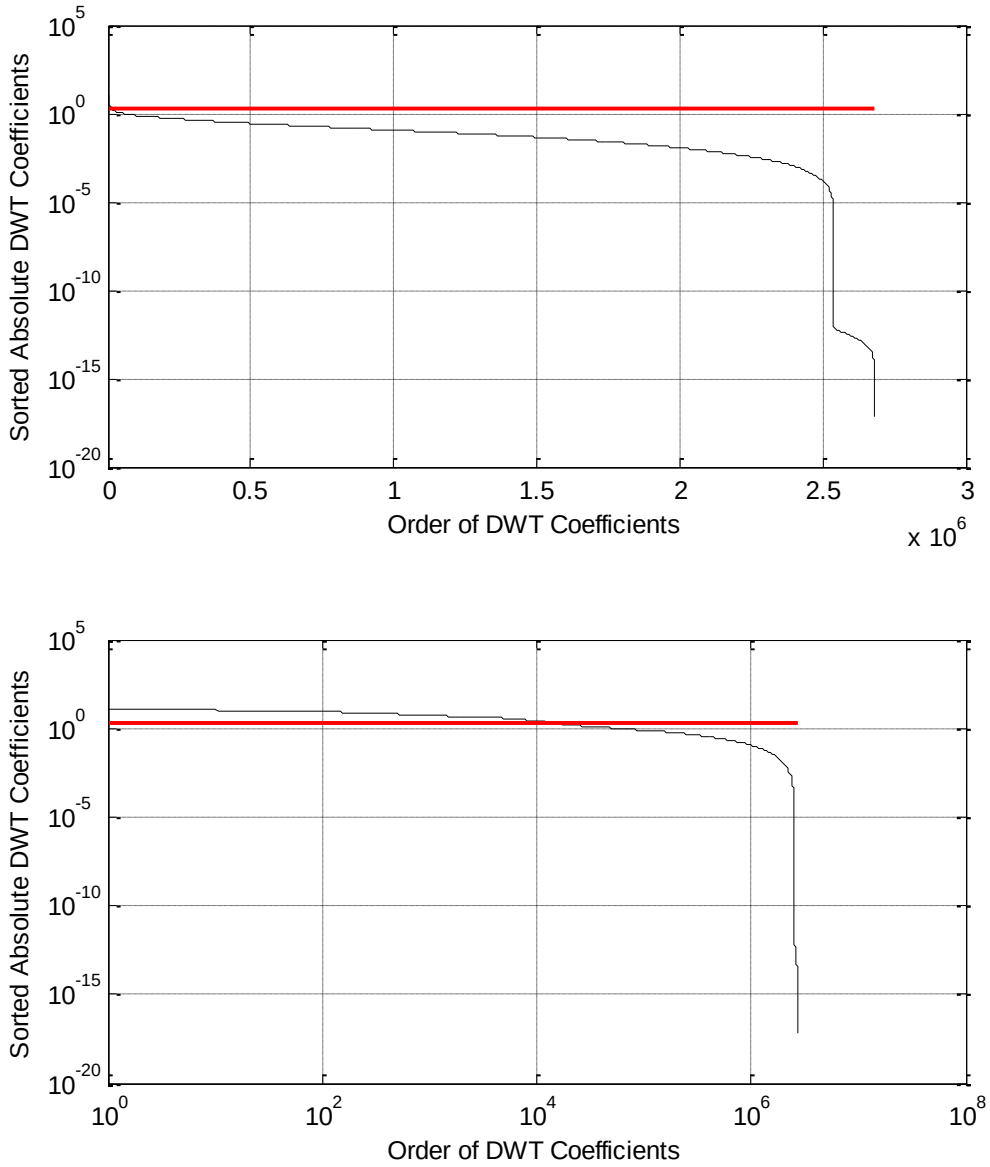

Figure 6.70. Semilog-log and Log-log Plots of Sorted Absolute 2D-DWT Coefficients of the Overall TEP Data Set for α=99.5 %.

Sizes of the ZIP files of the original and encoded data sets are compared as it is seen from Figure 6.71, the ZIP file of the original data set is nearly 26.1 MB (size of the ZIP file

is higher than that generated with 1D-DWT as mentioned in Section 6.3.7 as the number of 2D-DWT coefficients to be compressed is much larger than the number of 1D-DWT coefficients), whereas the ZIP file of the filtered transformed data set is nearly 0.44 MB. Thus, compression can be increased 58.8 times by applying 2D-DWT technique and taking the percentile value as 99.5% in thresholding step. It can be concluded that slightly higher compression levels are generated with 2D-DWT compared to 1D-DWT although 2D-DWT produces a large number of wavelet coefficients for large data series, and thus, resulting an undesirable dimensionality increase.

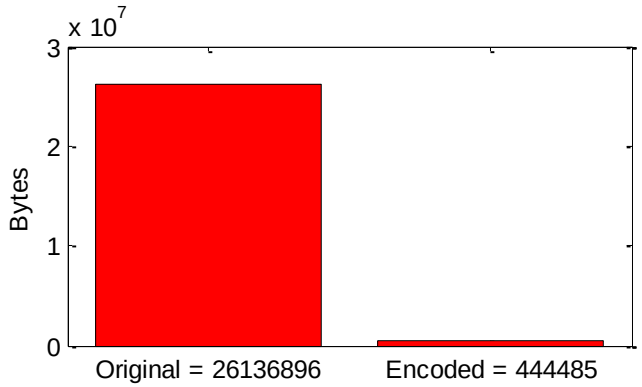

Figure 6.71. ZIP Compression Comparison of the Original and Encoded Overall TEP Data Set for α=99.5 % with 2D-DWT.

Reconstructed data generated with 2D-iDWT (inverse DWT) and their overlay with the originals are shown in Figure 6.72 and Figure 6.73 respectively. It is observed that none of the process events (upward/downward shifts and decreasing/increasing trends) can be monitored from the reconstructed data set as illustrated in Figure 6.73 concluding that there is a significant deterioration in each process measurement. It can also be stated that reconstruction is worsened with 2D-iDWT compared to 1D-iDWT.

In Figure 6.74, the original and reconstructed signals produced with 2D-iDWT are plotted around the y=x line. It can be stated that synthesized data set is completely different from the original data set as almost each of the reconstructed signals is located around the y=0 line.

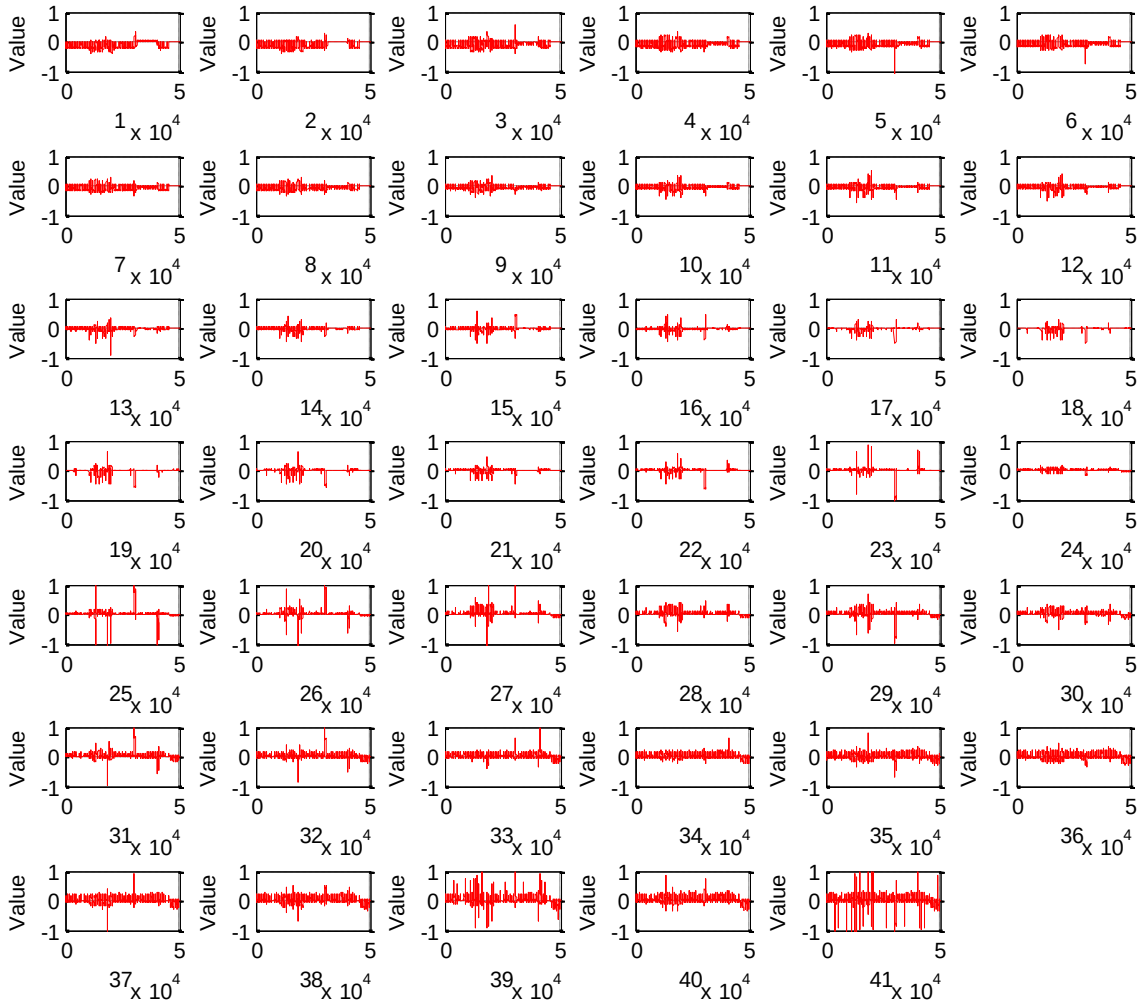

Figure 6.72. Reconstructed TEP Data Set with Inverse 2D-DWT.

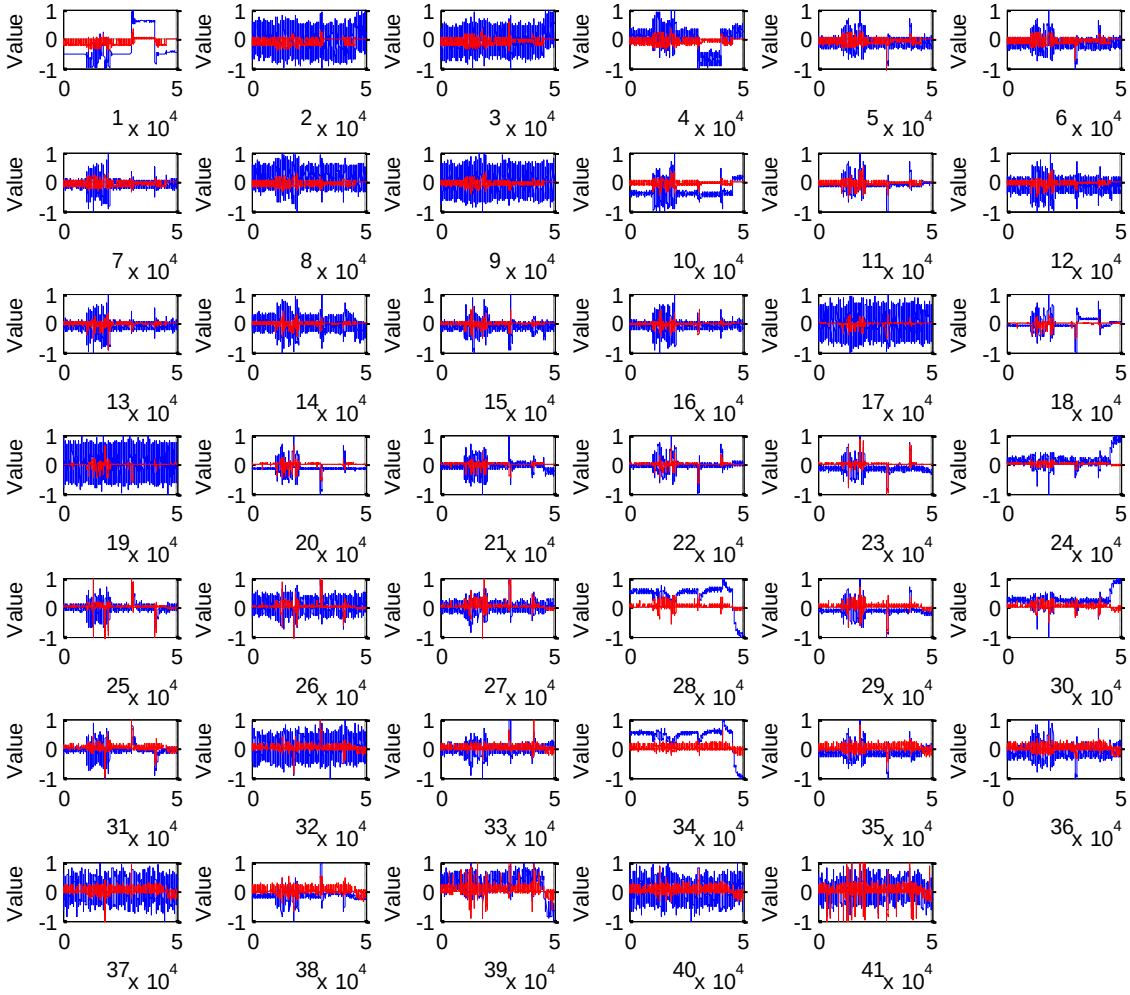

Figure 6.73. Original and Reconstructed Signals of the TEP Data Set with Inverse 2D-DWT.

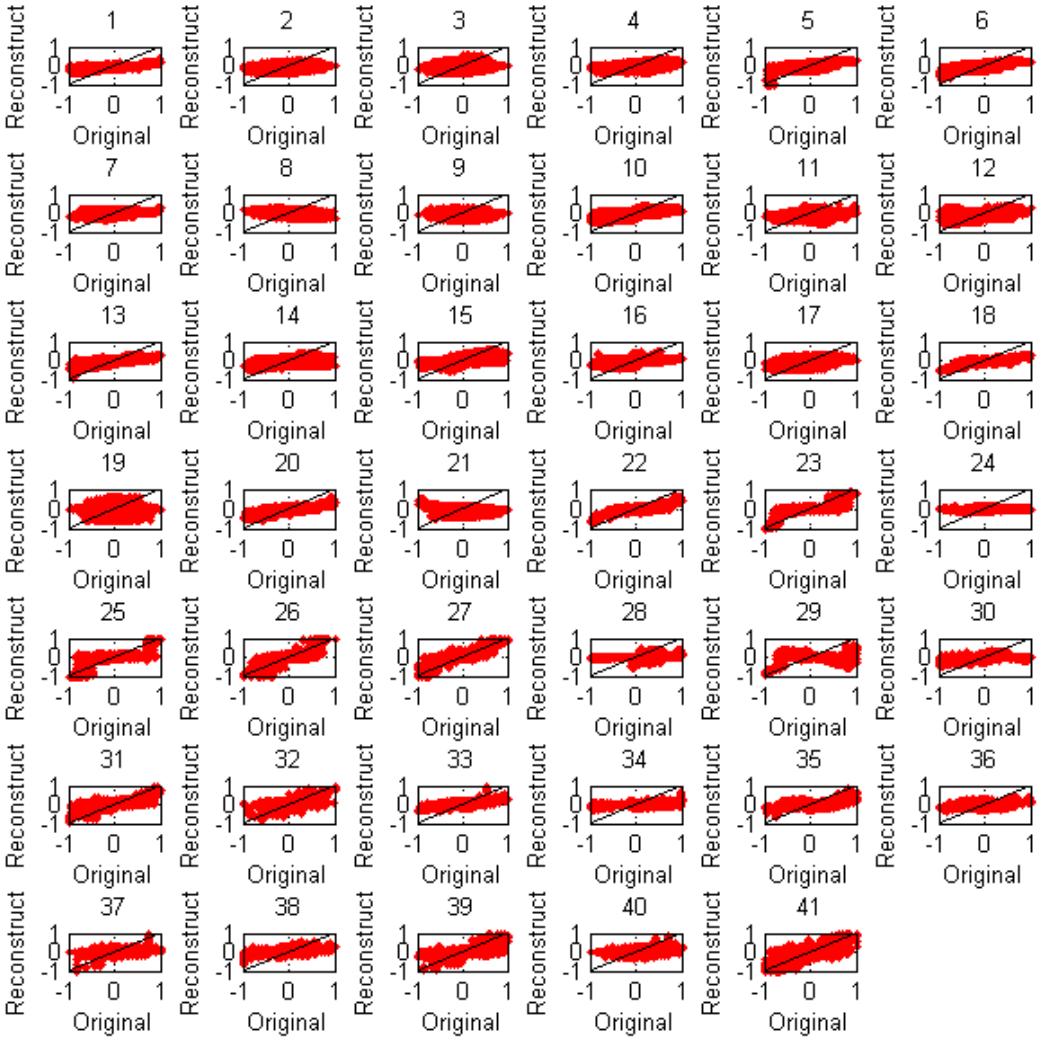

Figure 6.74. Reconstructed versus Original Signals of the TEP Data Set with Inverse 2D-DWT.

Reconstruction error norm values of the overall TEP data set are given in the bottom sub-window of Figure 6.75. Minimum error norm of 4813 is obtained in the seventh column and the maximum error norm of 25747 is obtained in the 34th column of the TEP data set after applying 2D-DWT. It can be concluded that both 1D-DWT and 2D-DWT fail in reconstruction while 2D-DWT generates slightly higher compression levels when highly uncorrelated TEP data set is used.

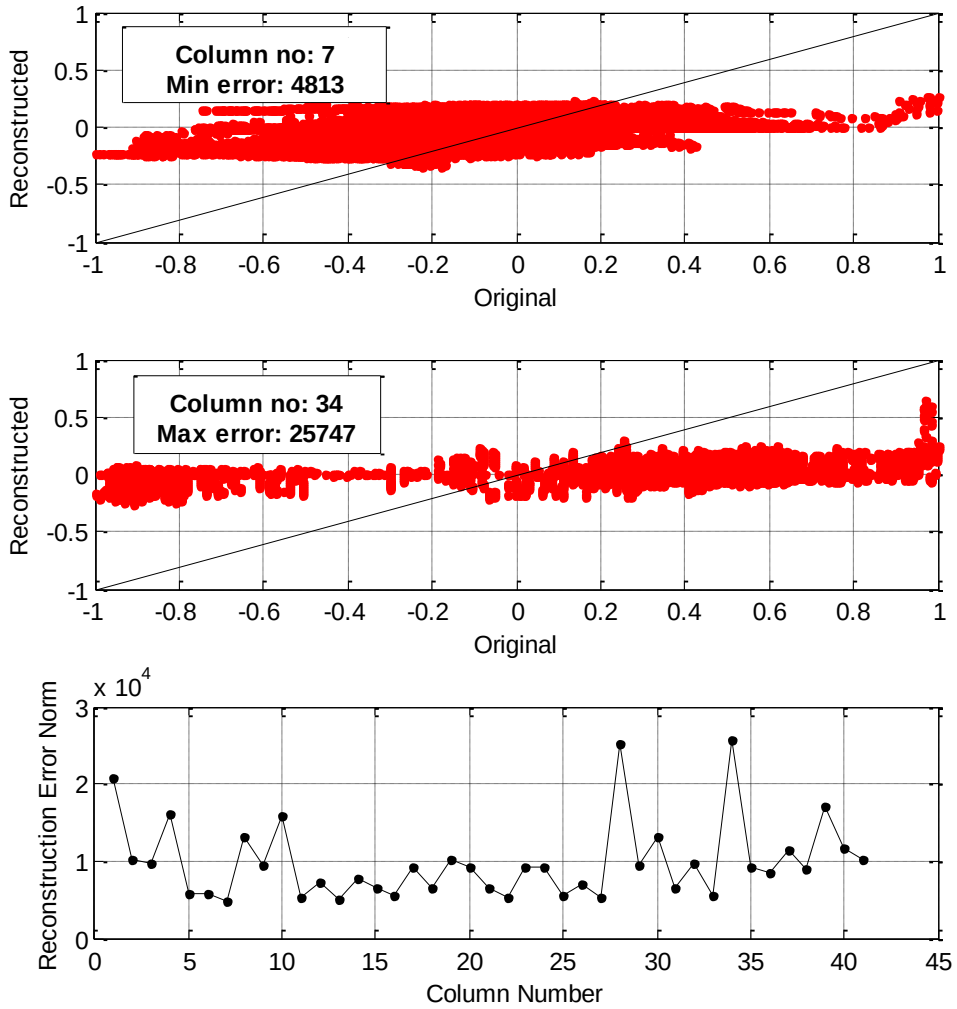

Figure 6.75. Reconstruction Error Norm Values of the TEP Data Set with Inverse 2D-DWT.

6.3.9. Comparison of One Dimensional and Two Dimensional Discrete Wavelet Transform Methods

In this section, 1D-DWT and 2D-DWT techniques are compared by using the overall data sets (PortSimHigh, PortSimLow, SELDI-TOF MS and TEP) mentioned in Chapter 3 for 10 different percentile values of the frequency distribution of the transform coefficients in the [15%-99.8%] range by using wavelet types `db1` with 10-level decomposition for the PortSimHigh and PortSimLow data sets, `db4` with eight-level decomposition for the SELDI-TOF MS data set and `sym4` with four-level decomposition for the TEP data set. The effect of the percentile values used in thresholding step on compression is measured in terms of compression ratio, mean error norm, % relative global error and % relative

maximum error. % Relative global error and % relative maximum error of a data set were defined before in Chapter 5 with Equation 5.8 and Equation 5.9. In addition, ratios of compression ratio to mean error norm are calculated to determine the optimum percentile level.

Compression ratios and mean error norms calculated for the overall PortSimHigh data set for 10 different percentile values are presented in Figure 6.76. It can be stated that slightly higher compression levels and much smaller mean error norm values are generated with 2D-DWT compared to 1D-DWT at thresholding percentiles higher than 95% as illustrated in Figure 6.76. Consequently, 2D-DWT should be the preferred technique for the highly correlated PortSimHigh data set as minimum distortion is obtained while maximum compression is achieved.

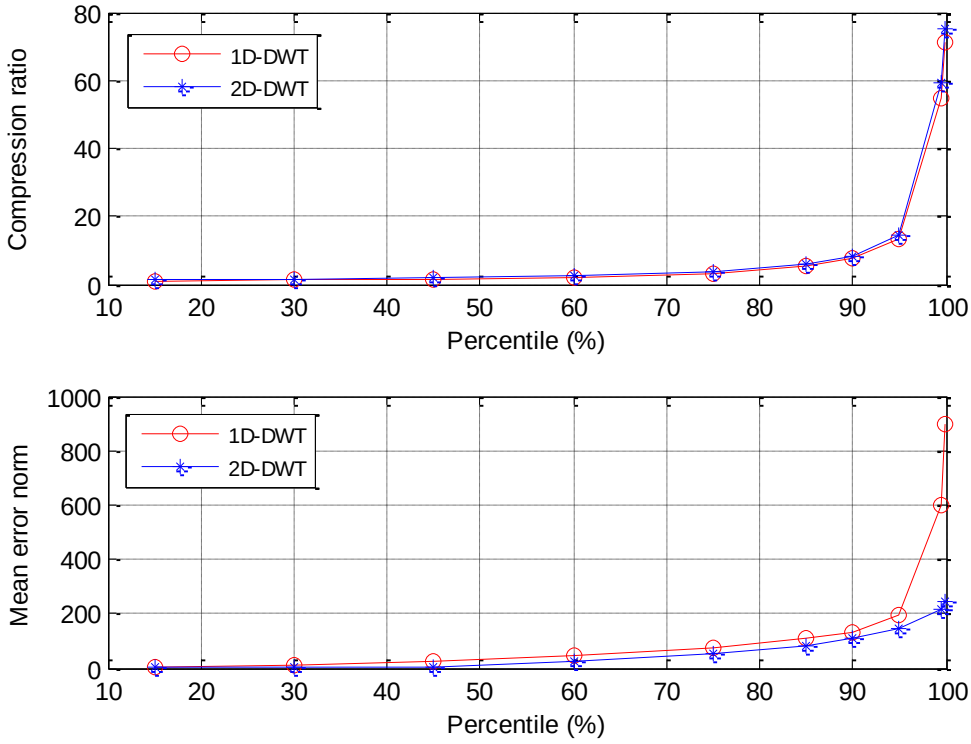

Figure 6.76. Compression Ratio and Mean Error Norm versus Thresholding Percentile for the PortSimHigh Data Set with DWT.

% Relative global and % relative maximum errors calculated for the overall PortSimHigh data set for 10 different percentile values are given in Figure 6.77. Both % relative global and % relative maximum errors obtained with 1D-DWT are much larger

than those obtained with 2D-DWT for percentile values higher than 95% as it can be seen from Figure 6.77.

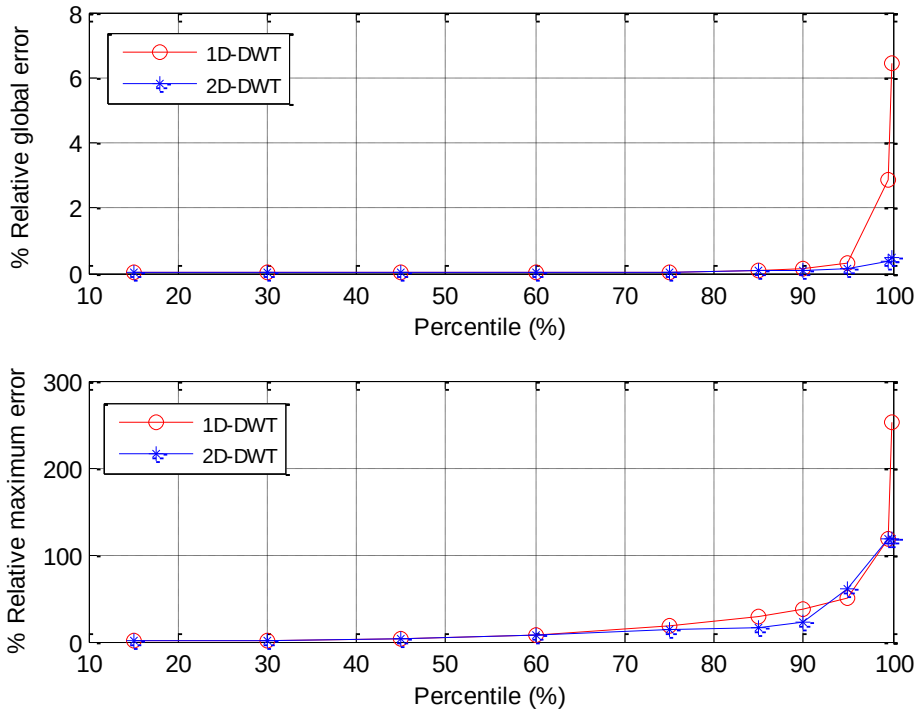

Figure 6.77. % Relative Global and % Relative Maximum Error versus Thresholding Percentile for the PortSimHigh Data Set with DWT.

Figure 6.78 is given to determine the optimum percentile value used in thresholding step. However, a reasonable optimum percentile value cannot be determined as the ratios of compression ratio to mean error norm decrease sharply for 2D-DWT as the percentile values increase. Nevertheless, it can be stated that 2D-DWT technique is more efficient for the PortSimHigh data set as the 2D-DWT curve is above the 1D-DWT curve for the percentiles higher than 95%.

Compression ratios and mean error norms calculated for the overall PortSimLow data set for 10 different percentile values are given in Figure 6.79. Slightly higher compression levels and higher mean error norm values are generated with 2D-DWT compared to 1D-DWT for large thresholding percentiles as shown in Figure 6.79. It can be concluded that 1D-DWT technique is more appropriate for the less correlated data sets while achieving dimensionality reduction without any significant distortion in reconstruction.

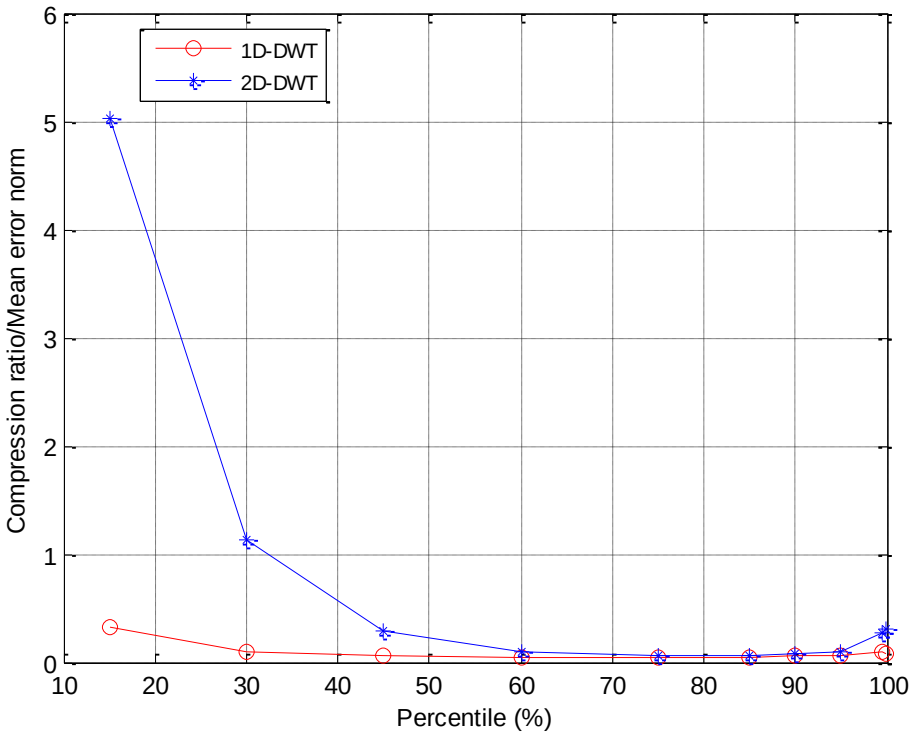

Figure 6.78. Compression Ratio/Mean Error Norm versus Thresholding Percentile for the PortSimHigh Data Set with DWT.

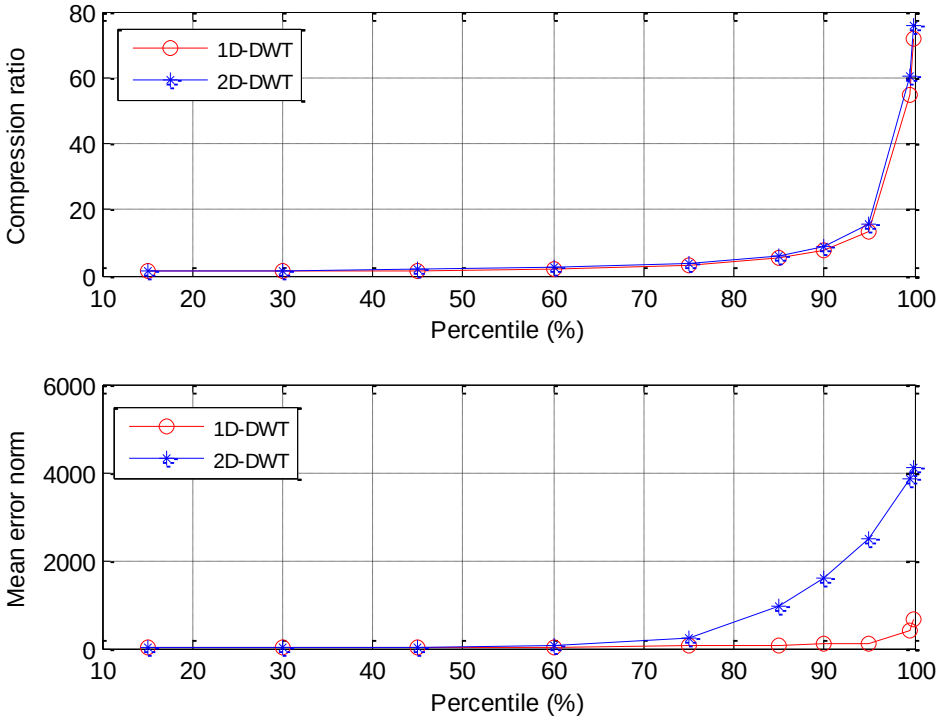

Figure 6.79. Compression Ratio and Mean Error Norm versus Thresholding Percentile for the PortSimLow Data Set with DWT.

% Relative global and % relative maximum errors calculated for the overall PortSimLow data set for 10 different percentile values are given in Figure 6.80. Both % relative global and % relative maximum errors obtained with 2D-DWT are much larger than those obtained with 1D-DWT for thresholding percentiles higher than 70% concluding that perfect reconstruction cannot be retained by 2D-DWT for the less correlated PortSimLow data set.

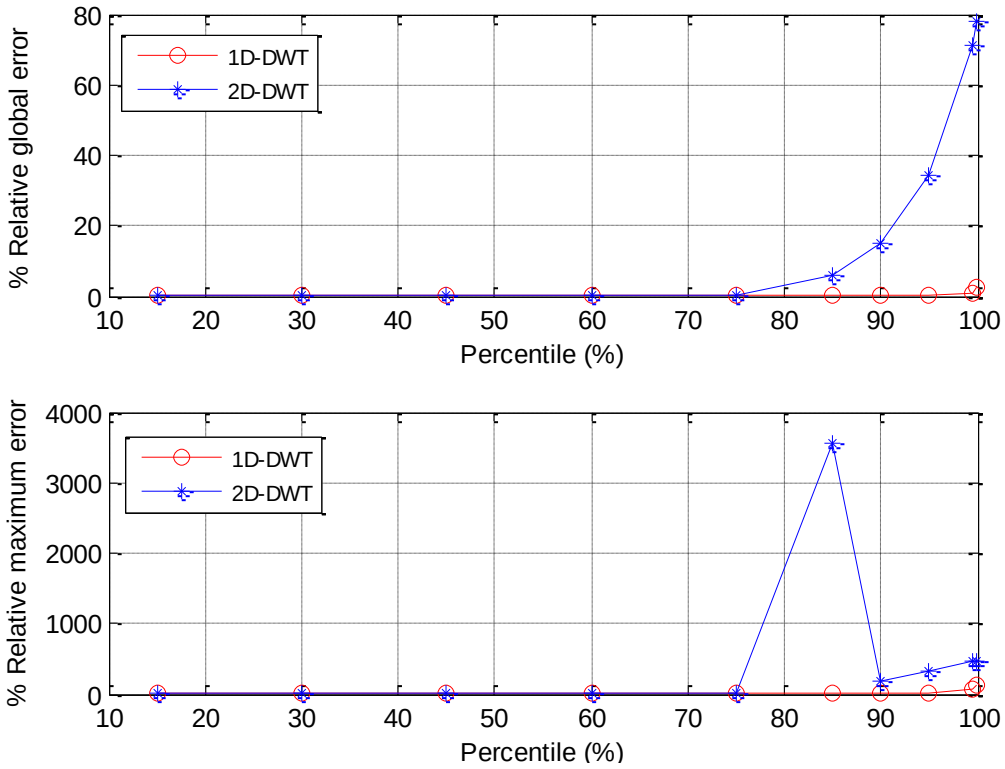

Figure 6.80. % Relative Global and % Relative Maximum Error versus Thresholding Percentile for the PortSimLow Data Set with DWT.

The ratio of compression ratio to mean error norm values computed for the overall PortSimLow data set for 10 different percentile values are given in Figure 6.81 to visually locate the optimum percentile level. However, optimum percentile value cannot be determined since the ratio of compression ratio to mean error norm values decrease as thresholding percentiles increase. Nevertheless, it can be stated that 1D-DWT is a more efficient method for the PortSimLow data set as 1D-DWT curve is above the 2D-DWT curve at each thresholding percentile.

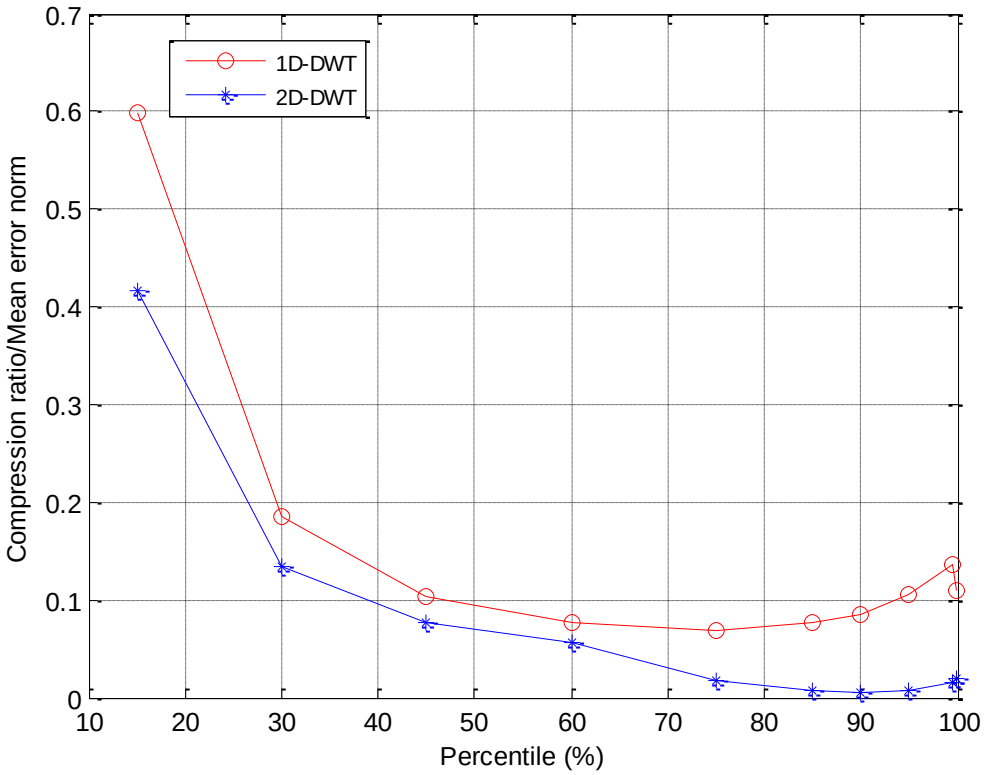

Figure 6.81. Compression Ratio/Mean Error Norm versus Thresholding Percentile for the PortSimLow Data Set with DWT.

Compression ratios and mean error norms calculated for the overall SELDI-TOF MS data set for 10 different percentile values are presented in Figure 6.82. It can be stated that 1D-DWT and 2D-DWT methods give almost the same compression levels and mean error norms. However, the mean error norm produced with 1D-DWT increases sharply at the percentile value of 99.8%.

% Relative global and % relative maximum errors calculated for the overall SELDI-TOF MS data set for 10 different percentile values are given in Figure 6.83. Both % relative global and % relative maximum errors obtained with 1D-DWT and 2D-DWT are similar at each thresholding percentile except the percentile of 99.8% where % relative errors produced with 1D-DWT increase sharply (nearly 50-fold increase in % relative global and nearly 80-fold increase in % relative maximum errors).

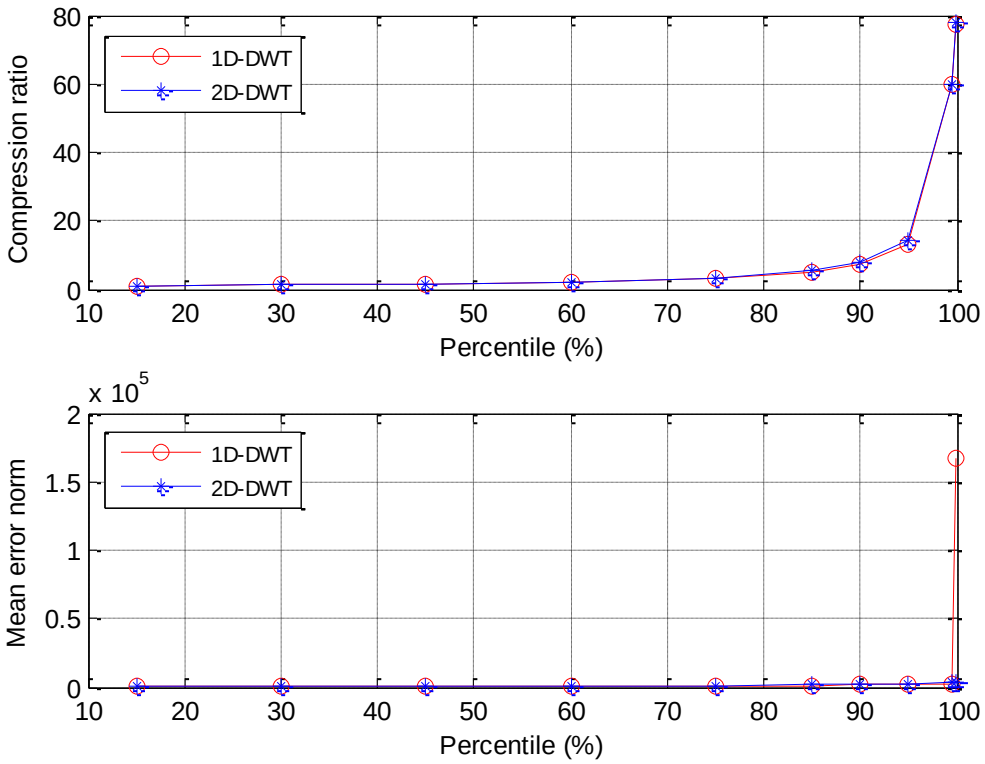

Figure 6.82. Compression Ratio and Mean Error Norm versus Thresholding Percentile for the SELDI-TOF MS Data Set with DWT.

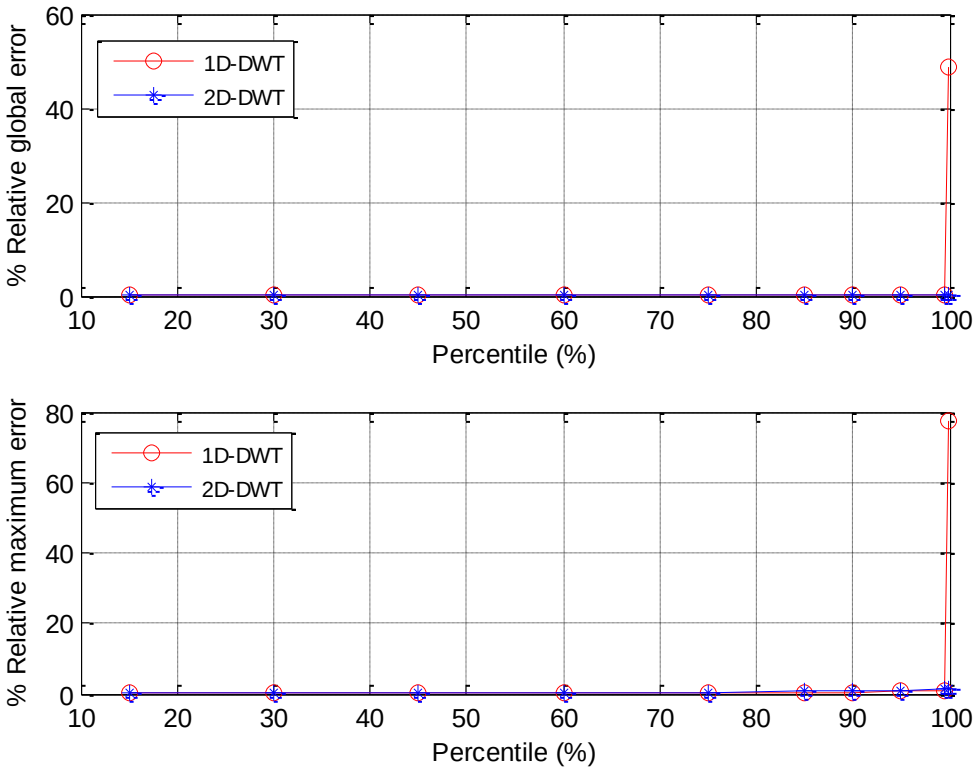

Figure 6.83. % Relative Global and % Relative Maximum Error versus Thresholding Percentile for the SELDI-TOF MS Data Set with DWT.

The ratio of compression ratio to error norm values computed for the overall SELDI-TOF MS data set for 10 different percentile values are given in Figure 6.84. Unlike PortSimHigh and PortSimLow data sets, the optimum percentile value can be specified as 99.5% (the isolated maximum produced with 1D-DWT) for the SELDI-TOF MS data set as shown in Figure 6.84.

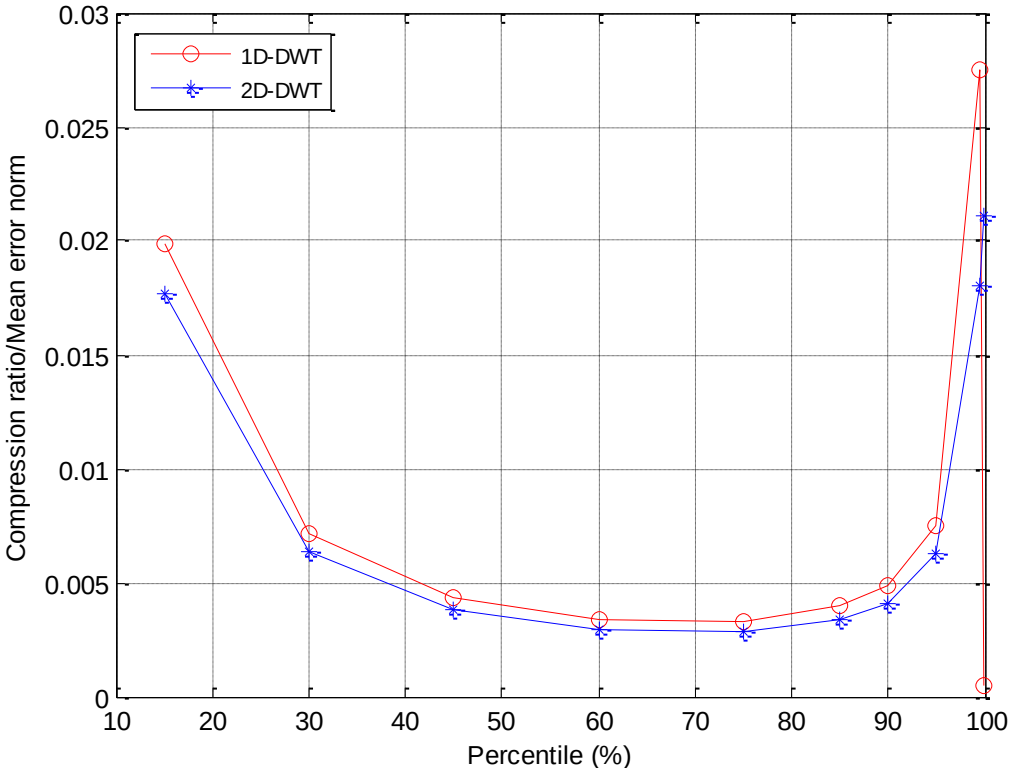

Figure 6.84. Compression Ratio/Mean Error Norm versus Thresholding Percentile for the SELDI-TOF MS Data Set with DWT.

Compression ratios and mean error norms calculated for the overall TEP data set for 10 different percentile values are given in Figure 6.85. Higher mean error norm values are generated with 2D-DWT compared to 1D-DWT for the TEP data set, although these two techniques give almost the same compression ratios as shown in Figure 6.85. However, the gap between the mean error norm values produced with 1D-DWT and 2D-DWT narrows at percentile levels higher than 99.5%. In addition, the value of the mean error norm obtained with 1D-DWT at the percentile level of 95% can be maintained with 2D-DWT at the percentile level of 75%, and thus, the corresponding compression ratio decreases nearly three-fold.

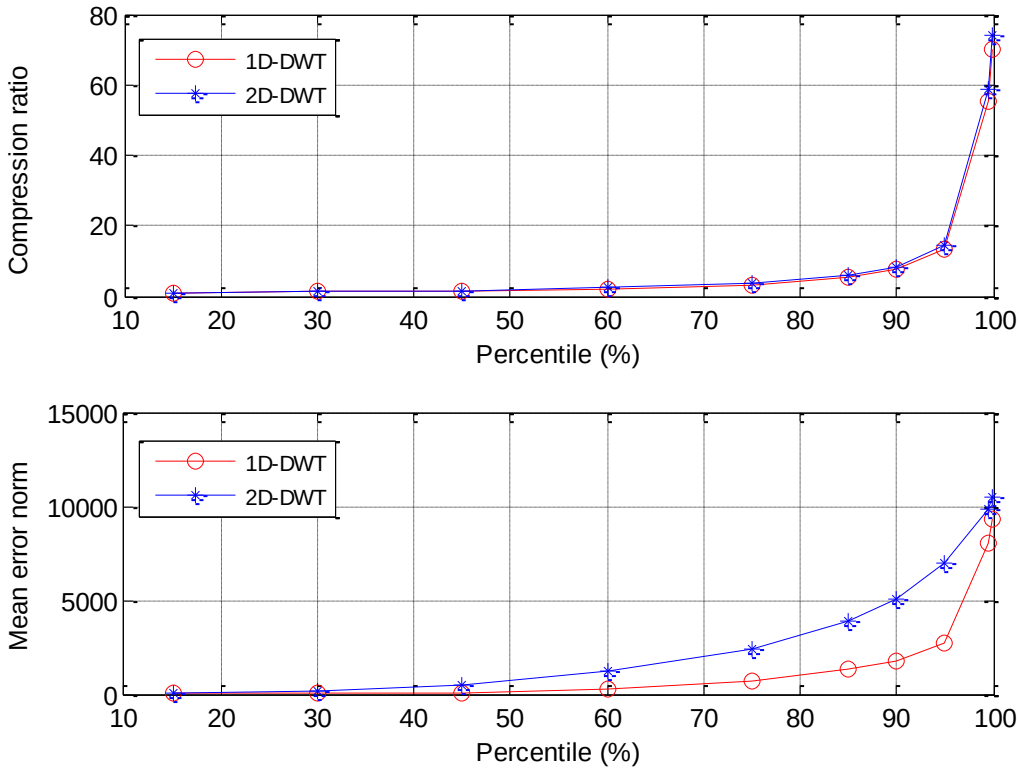

Figure 6.85. Compression Ratio and Mean Error Norm versus Thresholding Percentile for the TEP Data Set with DWT.

% Relative global and % relative maximum errors calculated for the overall TEP data set for 10 different percentile values are given in Figure 6.86. Larger % relative global and % relative maximum errors are produced with 2D-DWT compared to 1D-DWT as illustrated in Figure 6.86.

The ratio of compression ratio to error norm values computed for the overall TEP data set for 10 different percentile values are presented in Figure 6.87. Optimum percentile levels cannot be determined as the ratio of compression ratio to error norm values decrease at higher percentiles. Nevertheless, it can be concluded that slightly better results can be generated with 1D-DWT compared to 2D-DWT for large thresholding percentiles.

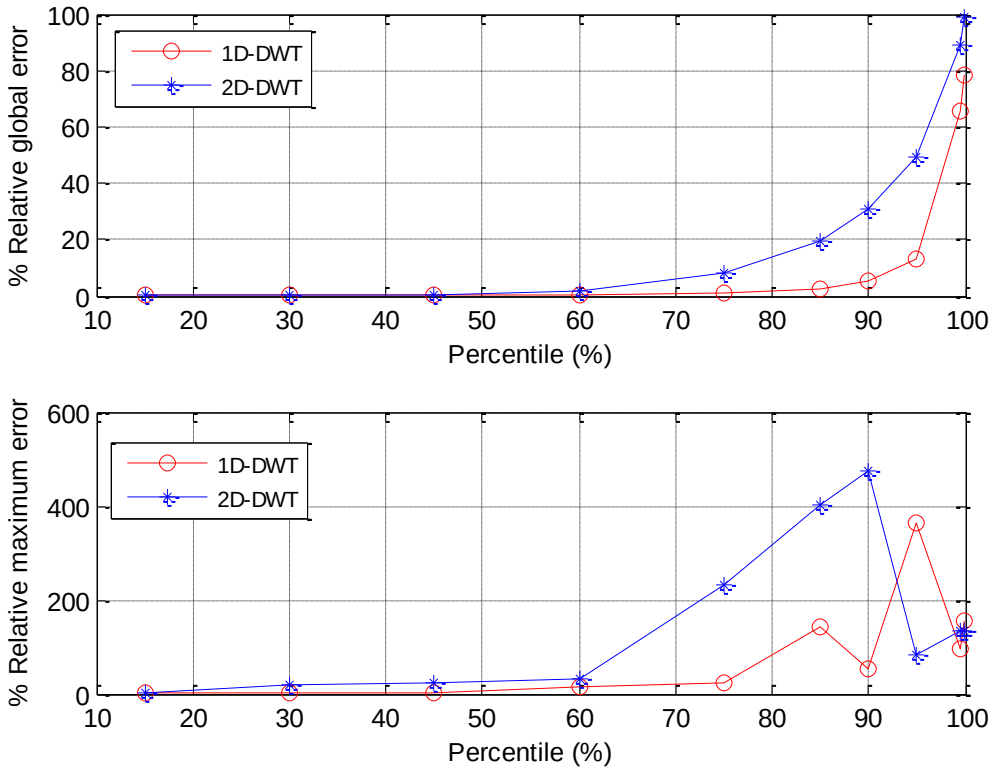

Figure 6.86. % Relative Global and % Relative Maximum Error versus Thresholding Percentile for the TEP Data Set with DWT.

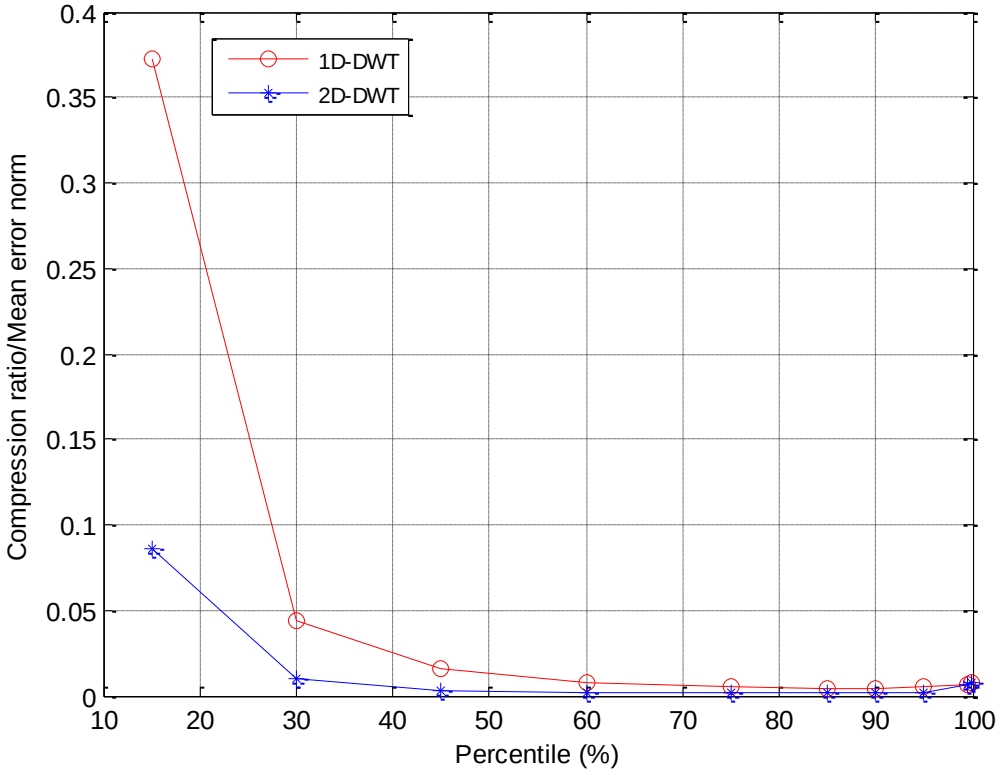

Figure 6.87. Compression Ratio/Mean Error Norm versus Thresholding Percentile for the TEP Data Set with DWT.

To sum up, multilevel decomposition is generally preferred in DWT to eliminate noisy measurements, and thus, providing small reconstruction error norms. Magnitudes of the wavelet coefficients also increase as the level in decomposition increases, revealing that the most important information content is stored in the deepest level coefficients. In addition, the most prominent patterns in the original series are preserved in the approximations, whereas the high-frequency components are kept in details. Compression is achieved by truncating wavelet coefficients below a certain threshold limit. Therefore, the sharpest features will be smoothened if the detail coefficients are eliminated, whereas there will be a significant deterioration in the reconstructed data set if some of the approximation coefficients are discarded.

The efficacy of the DWT depends directly on the selected mother wavelet function. Hence, `db1` is used for smoother data sets (PortSimHigh and PortSimLow), whereas `db4` and `sym4` are preferred for noisy data sets (SELDI-TOF MS and TEP). Furthermore, the decomposition level in DWT is selected so as to yield the same compression level generated by DCT at the percentile value 99.5%, and thus, reconstruction error norms produced in DCT and DWT can be compared at the same compression level.

High compression levels cannot be achieved with percentile values less than 90% in both DCT and DWT methods, whereas the quality of reconstructed signal can be deteriorated at higher threshold values. Consequently, thresholding percentile should be selected so as to maximize compression while preserving major information content. As the percentile values used in thresholding step increase, compression ratio, mean error norm, % relative global error and % relative maximum error values increase steadily. Mean error norms, % relative global and % relative maximum errors calculated for the TEP data set are much higher than those of the SELDI-TOF MS, PortSimHigh and PortSimLow data sets for the same compression level of 80 due to its high noise content with level jumps. Furthermore, it can be stated that the efficacy of the DWT method increases when highly correlated PortSimHigh data set is used as the highest compression ratio to mean error norm ratios are generated.

More perfect reconstruction is obtained with 1D-DWT compared to 2D-DWT although these two techniques provide almost the same compression levels for

PortSimLow, SELDI-TOF MS and TEP data sets. However, 2D-DWT is superior to 1D-DWT in both compression and reconstruction when highly correlated PortSimHigh data set is used. Furthermore, 2D-DWT is not the appropriate technique for the data sets having large number of data rows such as SELDI-TOF MS data set as the number of wavelet coefficients (approximation and detail coefficients in three orientations (horizontal, vertical and diagonal) as mentioned in Section 6.2) increases in each decomposition level resulting a dimensionality increase which is not favored in data compression applications. Therefore, it can be said that the efficacy of the 2D-DWT technique depends on the characteristics of the data set.

Both 1D-DWT and 2D-DWT fail in reconstruction of the highly uncorrelated TEP data set containing level jumps and noisy measurements, and thus, 1D-DCT should be the preferred technique for the TEP data set. In addition, 2D-DCT generates more perfect reconstruction compared to 2D-DWT although 2D-DWT provides the highest compression when highly correlated PortSimHigh data set is used. Furthermore, 1D-DCT produces slightly smaller reconstruction error norms compared to 1D-DWT at the same compression level when less correlated PortSimLow data set is used. However, better reconstruction can be retained with 1D-DWT (especially reconstructing sharp peaks) compared to 1D-DCT at the same compression level when highly uncorrelated SELDI-TOF MS data set is used, and confirming the claim that DWT is superior to DCT in analyzing noisy data sets containing localized changes as DWT analyzes a signal both in time and frequency domain, whereas DCT works only in frequency domain.

7. TWO DIMENSIONAL COMPRESSION OF ONE DIMENSIONAL DATA VIA TRAJECTORY MATRIX APPROACH

The Singular Spectrum Analysis (SSA) is generally used for analyzing and forecasting time series with complex components. The main idea of SSA is to apply principal component analysis to the "trajectory matrix" composed from the original time series (Moskvina and Zhigljavsky, 2003). The SSA technique consists of two stages; decomposition and reconstruction. At the first stage, time series are decomposed into small number of time series so that oscillatory components can be identified, then at the second stage, original time series are reconstructed (Hassani *et al.*, 2009). In this chapter, the interest will only be in the construction of the trajectory matrix for its use in two dimensional compression of one dimensional data.

$\mathbf{m}_T$ is a real-valued nonzero time series of sufficient length T;

$$\boldsymbol{m}_T = (m_1, \dots, m_T) \tag{7.1}$$

Defining

$$K = T - L + 1 \tag{7.2}$$

where L is the window length with the assumption $L \leq T/2$

The Trajectory Matrix (TM) is defined as;

$$\boldsymbol{N} = [\boldsymbol{n}_1, \dots, \boldsymbol{n}_K] = \left(n_{i,j}\right)_{i,j=1}^{L,K} \tag{7.3}$$

$$= \begin{pmatrix} m_1 & m_2 & m_3 & \cdots & m_K \\ m_2 & m_3 & m_4 & \dots & m_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_L & m_{L+1} & m_{L+2} & \dots & m_T \end{pmatrix}$$

N is a Hankel matrix, meaning that its $(i,j)^{th}$ entries depend only on the sum (i+j), in other words;

$$N_{i,j} = N_{i-1,j+1} \tag{7.4}$$

After obtaining the TM, the original vector, $\mathbf{m}_T$, having T rows becomes a multivariate data with L characteristics and K observations. Columns of the trajectory matrix lie in a space R^L (Hassani *et al.*, 2009).

SSA method requires the selection of some parameters but the choice of these parameters should depend on the characteristics of the original series. According to Moskvina and Zhigljavsky (2003), a general rule is to choose T reasonably large. However if T is too small, then any change in the time series can be missed. The window length L is the single parameter that should be selected at the decomposition stage. Golyandina *et al.* (2001) proposed that there are some suggestions in the literature for selecting parameters such as keeping the ratio L'/T' fixed, where L' is the window length for the subseries of length T'=T/k, where k is the number of subseries.

In this chapter, the transformation of one dimensional data into two dimensional data for compression and reconstruction is studied by composing a trajectory matrix and then analyzing the effects of L/K ratio and length of T using the PortSimHigh and Tennessee-Eastman Plant (TEP) data sets via two dimensional Discrete Cosine Transform (2D-DCT) method.

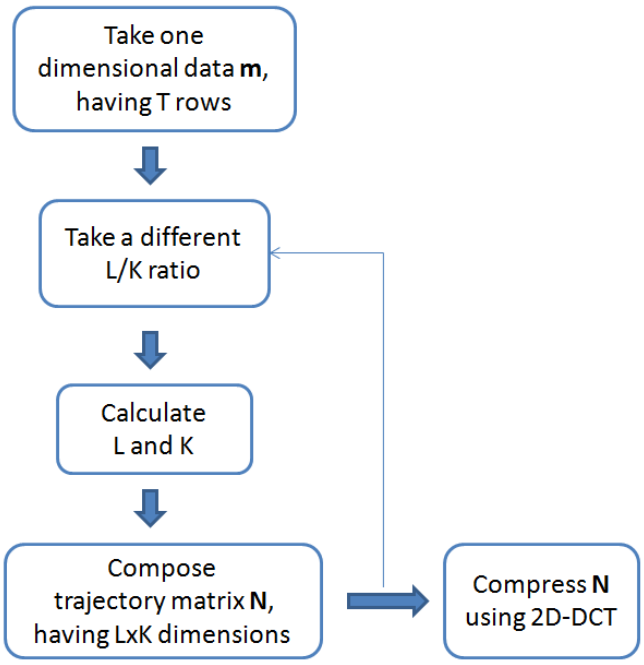

Figure 7.1. The Procedure used in the 2D-DCT Compression of One Dimensional Data via Trajectory Matrix.

At the α-% level of 99.5, 2D-DCT is applied by using 50^{th} and 30^{th} columns of the PortSimHigh and TEP data sets respectively at three different T lengths; 100, 500 and 1500. The L/K ratio is also varied in the [0.1-4.5] range.

Compression ratios of both the original data vector and its TM for different L/K values are given in Figure 7.2. Since one dimensional data compression is independent of the L/K ratio, compression ratio (at the same α-% level of 99.5) of the original data is shown as constant via the thick horizontal line. The reduction ratio increases nearly four times when one dimensional data is transformed into two dimensional data. This is mainly due to the high correlation occurred between columns of the composed TM. It can be said that varying the L/K ratio does not show a certain effect on the compression ratio, whereas the error norms increase steadily as L/K ratio increases. It should also be stated that the highest error norms are obtained in the case of one dimensional data compression.

As far as Figure 7.2 and Figure 7.3 are concerned, it is seen that error norms of the reconstructed TMs composed from the TEP data are much higher than those of the PortSimHigh data due to the fact that there is a higher correlation (as demonstrated in Chapter 3) between the columns of PortSimHigh data leading to a better compression/reconstruction. However, there is not much difference in the compression ratios between these two different data sets.

As the length of T is increased from 500 to 1500 in the TEP data, it is seen from Figure 7.4 that compression ratios of both the original data and TMs increase, but the error norms also increase. However, as the length of T is decreased from 500 to 100 in the same data, both the compression ratios and error norms decrease. For instance, the maximum compression ratio drops from 53% to nearly 38%, however it is obviously seen from Figure 7.5 that, the optimum L/K value can be set in the range [1.0-1.5] leading to the result that when window length L is greater than K, in which case the higher compression can be obtained.

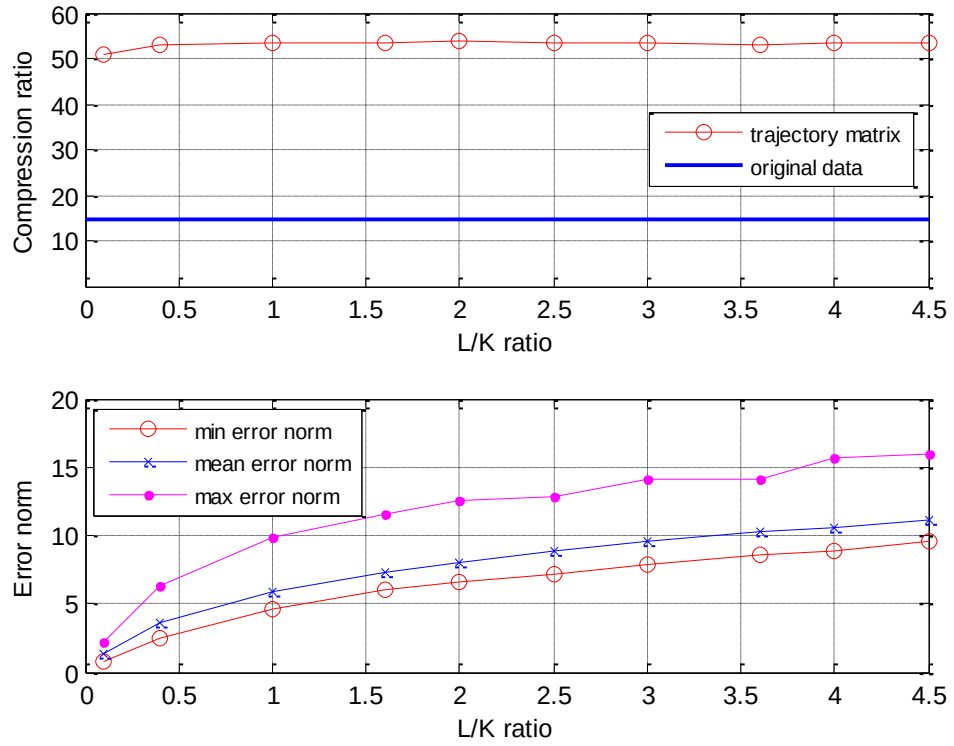

Figure 7.2. Compression Ratio and Error Norm versus L/K ratio using 50th column of the PortSimHigh Data Set (T=500).

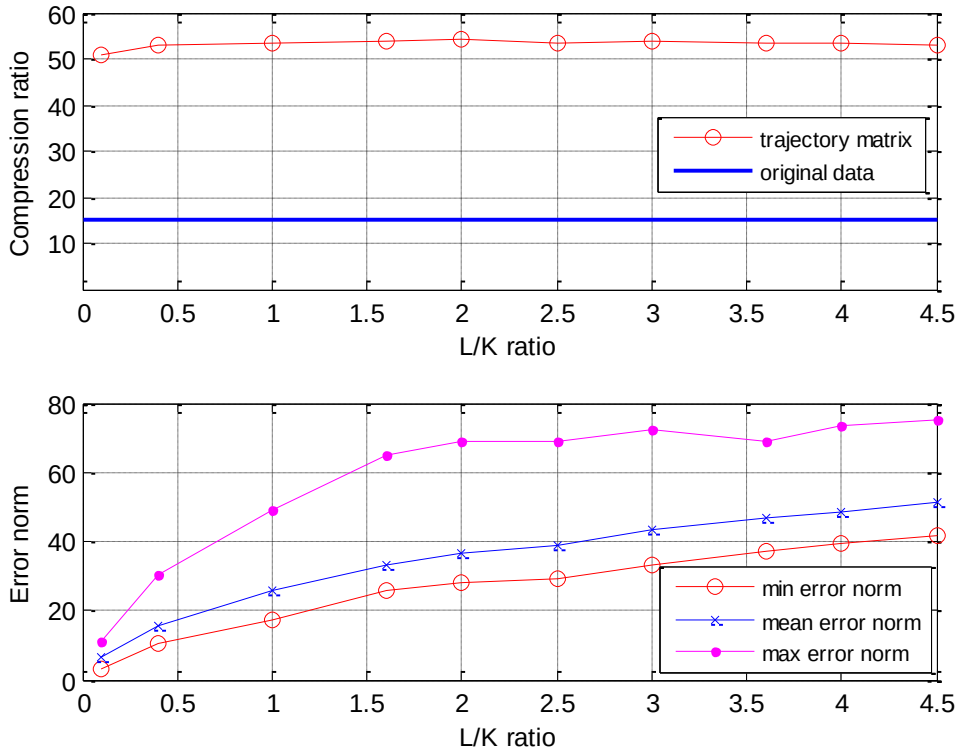

Figure 7.3. Compression Ratio and Error Norm versus L/K ratio using 30th column of the TEP Data Set (T=500).

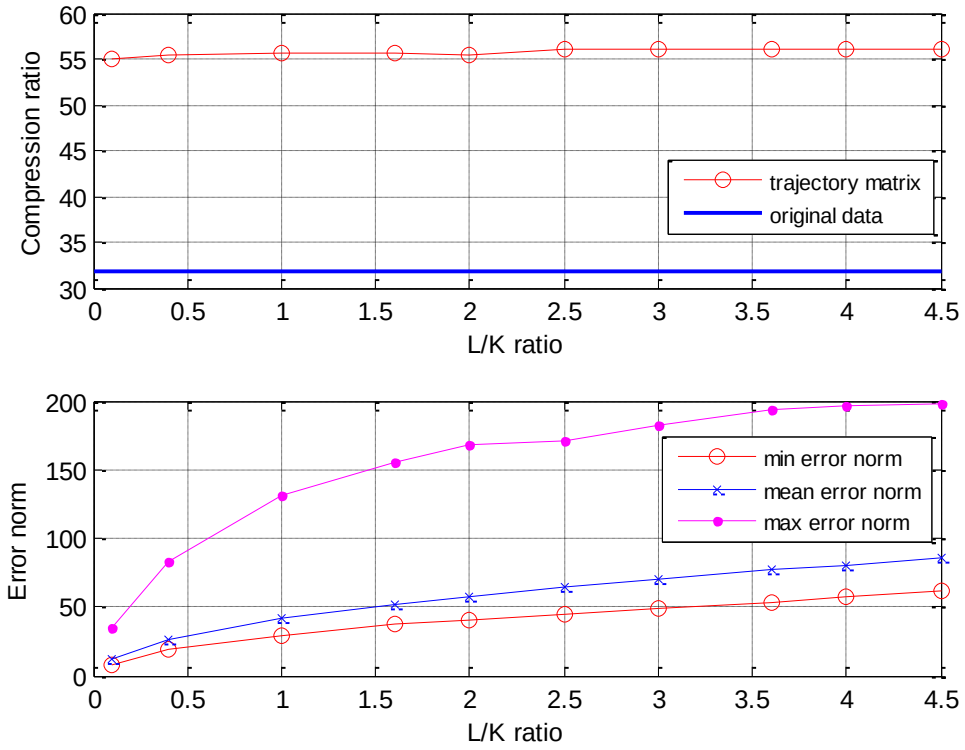

Figure 7.4. Compression Ratio and Error Norm versus L/K ratio using 30th column of the TEP Data Set (T=1500).

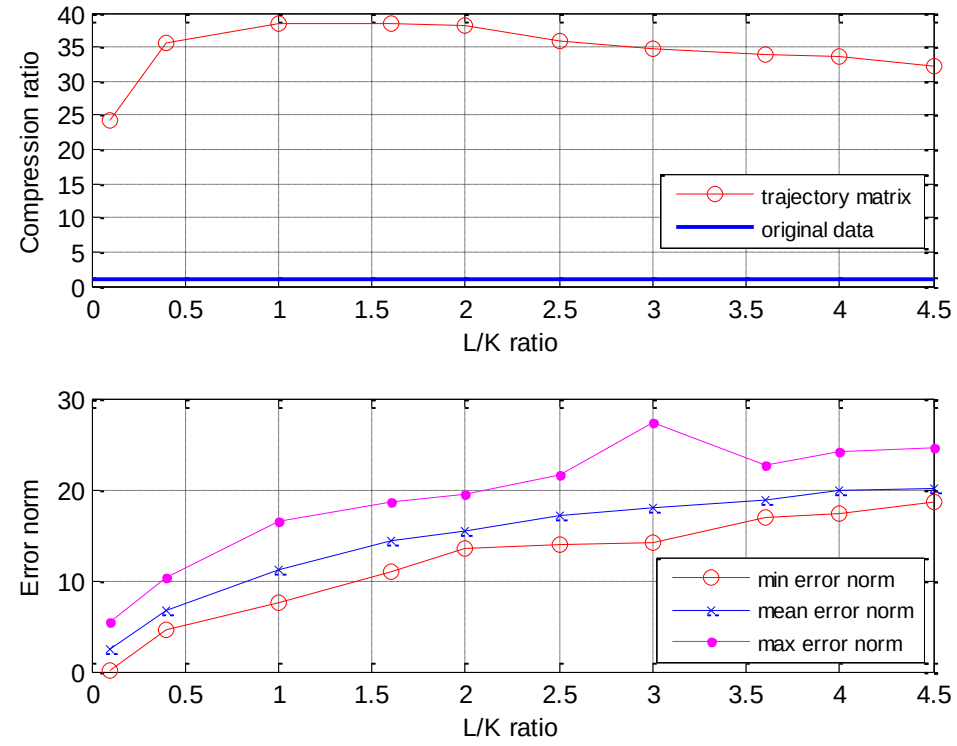

Figure 7.5. Compression Ratio and Error Norm versus L/K ratio using 30th column of the TEP Data Set (T=100).

8. CONCLUSIONS AND RECOMMENDATIONS

In this thesis work, the lossy/lossless data compression and reconstruction were investigated by using the dimensionality reduction techniques Piecewise Aggregate Approximation (PAA), One Dimensional and Two Dimensional Discrete Cosine Transform (1D-DCT and 2D-DCT) and One Dimensional and Two Dimensional Discrete Wavelet Transform (1D-DWT and 2D-DWT), including the thresholding method as a lossy compression step and ZIP as the lossless encoding algorithm using the data sets PortSimHigh, PortSimLow, SELDI-TOF MS and TEP, the properties of which were presented in Chapter 3. These techniques were compared by measuring compression ratio, reconstruction error norm, % relative global error and % relative maximum error for different α-% thresholding levels. All of the computations were performed in MATLAB.

8.1. Conclusions

In Chapter 2, filtering methods, applied to transform coefficients to generate higher compression, were studied. It can be stated that thresholding method should be preferred for data series having high-frequency content, whereas zero padding method is more suitable for smoother data series since their transform coefficients die out exponentially towards almost zero. It is important to adjust threshold limits in lossy quantization step as the reconstruction error norm increases with the number of discarded transform coefficients.

In Chapter 4, the simplest dimensionality reduction technique Piecewise Aggregate Approximation (PAA), in which high compression ratios are achieved by keeping only the segment coordinates, was studied. However, the segmented data set composed by the PAA cannot be reconstructed like other transformed data sets (such as those obtained via DCT or DWT), and thus, the PAA is a lossy compression method and the amount of distortion increases as the number of segments used in PAA decreases. Compression ratios can further be improved by the quantization technique due to discarded digits that are negligible and that will not cause any significant increase in error norms. Quantization step becomes more effective as the number of frames used in PAA increases.

As the number of segments decreases, entropies (information content) of the segmented data also decrease since original data set is represented by fewer bits. However, compression ratios and error norms increase steadily as the number of segments decreases. Reduction ratios of the segmented data set composed from both the TEP and SELDI-TOF MS data are much lower than those of the PortSimHigh data due to their high noise content with sudden changes. It is concluded that PAA is not an appropriate method for noisy data sets especially the ones containing frequent peaks.

In Chapter 5, Discrete Cosine Transform (DCT) was investigated to generate higher compression ratios while minimizing information loss in reconstruction. DCT performs reversible mapping from time to frequency domain while exhibiting excellent decorrelation and energy compaction properties providing both compression and noise removal. Compression is achieved by truncating transform coefficients below a certain threshold limit. The original data set can be reconstructed thoroughly by the inverse DCT without any significant distortion in information content.

High compression levels cannot be achieved with percentile values less than 90% in both DCT and DWT methods, whereas the quality of reconstructed signal can be deteriorated at higher threshold values. Consequently, thresholding percentile should be selected so as to maximize compression while preserving major information content.

Mean error norms calculated for the highly uncorrelated TEP and SELDI-TOF MS data sets were much higher (nearly five times) than those of the PortSimHigh and PortSimLow data sets for the same compression level of 80 due to their high noise content with sudden changes. As the compression of this type of data sets is very difficult, the efficacy of the DCT technique decreases a lot. It can also be stated that for the data sets consisting of almost pure noise with level jumps, higher compression levels can be produced with the DCT method instead of the hybrid method consisting of PAA and quantization. In addition, 2D-DCT is the preferred technique for the highly correlated data sets.

In Chapter 6, Discrete Wavelet Transform (DWT) was investigated to increase compression ratios further while reducing reconstruction error norms. Hierarchical

structure in DWT is favored in eliminating noisy measurements providing smaller reconstruction error norms while dimensionality reduction and de-noising are achieved simultaneously. As the decomposition level increases, both compression ratios and reconstruction error norms decrease. Furthermore, the most important information content is stored in the deepest level coefficients. In addition, the most prominent patterns in the original series are preserved in the approximations, whereas the high-frequency components are kept in the details. Compression is achieved by truncating wavelet coefficients below a certain threshold limit. Therefore, successive approximations become less noisy as more detail coefficients are filtered. The original signal is reconstructed by applying the inverse DWT over both the detail and approximation coefficients before combining them.

DWT does not have a single set of basis functions, unlike DCT. The family of basis functions are scaled and translated versions of a mother wavelet function which strongly affects the efficacy of the DWT. Hence, `db1` was used for smoother data sets (PortSimHigh and PortSimLow), whereas `db4` and `sym4` were preferred for noisy data sets (SELDI-TOF MS and TEP). Furthermore, the decomposition level in DWT was selected so as to yield the same compression level generated by DCT at the percentile value 99.5%, and thus, reconstruction error norms produced in DCT and DWT could be compared at the same compression level.

As the percentile values used in thresholding step increase, compression ratio, mean error norm, % relative global error and % relative maximum error values increase steadily. Mean error norms, % relative global and % relative maximum errors calculated for the TEP data set are much higher than those of the SELDI-TOF MS, PortSimHigh and PortSimLow data sets for the same compression level of 80 due to its high noise content with level jumps.

2D-DWT was superior to 1D-DWT in both compression and reconstruction when highly correlated PortSimHigh data set was used. However, 2D-DWT was not favored for PortSimLow, SELDI-TOF MS and TEP data sets as more perfect reconstruction was obtained with 1D-DWT although these two techniques provided almost the same compression levels. Furthermore, 2D-DWT should not be the preferred technique for the

data sets having large number of data rows, such as SELDI-TOF MS data set, since the number of wavelet coefficients (approximation and detail coefficients in three orientations; horizontal, vertical and diagonal) increases in each decomposition level, resulting a significant increase in dimension, and thus, making compression less viable.

To conclude, 1D-DCT is the most appropriate compression technique for the highly uncorrelated TEP data set containing level jumps and noisy measurements as both 1D-DWT and 2D-DWT fail in reconstruction of the TEP data set, unless the thresholding level is decreased further at the expense of less compression. In addition, 2D-DWT provides the highest compression and 2D-DCT generates the best reconstruction when highly correlated PortSimHigh data set is used. Furthermore, slightly smaller reconstruction error norms are obtained with 1D-DCT compared to 1D-DWT at the same compression level when less correlated PortSimLow data set is used. However, better reconstruction is retained with 1D-DWT (especially in reconstructing sharp peaks) compared to 1D-DCT at the same compression level when highly uncorrelated SELDI-TOF MS data set containing baseline noise is used. DWT is advantageous for the non-stationary and noisy data sets with sudden changes, whereas DCT is more suitable for smooth and random-walk-type data sets.

In Chapter 7, the transformation of one dimensional data into two dimensional data for compression and reconstruction was studied by composing a trajectory matrix and then using the 2D-DCT method. After obtaining the trajectory matrix, the original vector having T rows becomes a multivariate data with L characteristics and K observations. It was concluded that compression ratios increase and error norms decrease when one dimensional data is transformed into two dimensional data due to the high correlation occurred between columns of the composed trajectory matrix. Furthermore, higher compression can be obtained in the case of window length L is greater than K. As the L/K ratio increases, error norms increase steadily. In addition, as the length of T decreases, both the compression ratios and error norms decrease.

To sum up, the efficacy of the compression methods depends on the data characteristics such as smoothness, correlation among columns, presence of peaks, noise content and presence of level jumps as illustrated in Table 8.1 where n/a stands for not

applicable since the efficacy of 1D-DCT, 1D-DWT and PAA does not depend on the correlation property of the data sets.

Table 8.1. Efficacy of the Compression Methods

	Compression Methods				
Data Characteristics	PAA with Quantization	1D-DCT	2D-DCT	1D-DWT	2D-DWT
Smooth • PortSimHigh • PortSimLow	▲▲▲	▲▲▲	▲▲▲	▲▲▲	▲▲
High peak content • SELDI-TOF MS	▲	▲▲	▲▲	▲▲▲	▲
Noisy • TEP • SELDI-TOF MS	▲	▲▲	▲▲	▲▲	▲
High jump content • TEP	▲	▲▲	▲	▲	▲
Correlated • PortSimHigh	n/a	n/a	▲▲▲	n/a	▲▲▲
Uncorrelated • PortSimLow • TEP • SELDI-TOF MS	n/a	n/a	▲	n/a	▲
▲▲▲: very efficient, ▲▲: efficient, ▲: inefficient					

8.2. Recommendations for Future Work

In this thesis work, it is stated that in process monitoring, the key features of the original data set can easily be followed visually from the segmented data set composed by the PAA. However, PAA is an irreversible technique, unlike DCT and DWT, and thus, important data points may sometimes be overlooked depending on the segment size. Furthermore, algorithms used in DCT and DWT consist of the combination of

transformation, filtering, encoding and reconstruction steps, whereas the algorithm used in PAA consists of only segmentation and encoding steps. Consequently, it takes less CPU time to compose the segmented data set by PAA than the reconstruction of data sets by DCT and DWT. In addition, DWT performs best in reconstructing the non-stationary and noisy data sets containing frequent peaks although DCT and DWT almost give the same degree of compression. To conclude, DWT is the preferred technique in many applications such as process monitoring, fault detection and signal processing. As a result of the experience gained in this thesis work and in the light of the conclusions mentioned above there may be a few recommendations for future work.

Data compression and reconstruction can be improved by using the following hybrid methods that capitalize on advantageous properties of two techniques; the DCT followed by quantization, the DWT followed by quantization, the DCT followed by the PAA and the DWT followed by the DCT. In the first and second methods, transform coefficients can be quantized before the encoding step in order to improve compression ratios further without any significant increase in reconstruction error norms. In the third method, PAA can be applied to the DCT coefficients to increase compression. However, desired reconstruction may not be retained depending on the segment size used in PAA. In the fourth method, DCT can be applied to the wavelet transform coefficients to improve the compression performance.

It is often the case that the stored data are retrieved from time to time for various tasks such as pattern recognition, classification and fault detection. In transform compression methods, compression is achieved by the combination of transformation and filtering steps. Thus, the transform method and the thresholding limit in filtering step should be selected so as to maintain the major features of the raw data in the retrieved data (decompressed data). For instance, it can be investigated that up to what thresholding percentile, the raw data and the retrieved data show the same operational fault disturbances. In the same manner, it can be studied that beyond which thresholding percentile, the plant operator cannot distinguish faulty and normal operating conditions visually in the retrieved and the reconstructed data.

APPENDIX A: MATLAB CODES USED

A.1. Matlab Code used in Piecewise Aggregate Approximation followed by Quantization

```
% using 50th column of the PortSimHigh data set
load PortSimHigh.txt;
D=PortSimHigh(:,50);
[TotR TotC]=size(D);
D=[mapminmax(D')]';
% calculating Shannon entropy of the original data
EntropyD=EntropyUA(D);

fid=fopen('SAVE_ORIGINAL_DATA.txt', 'wt');
fprintf(fid,[repmat('%30.20f ',1,52) '\n'],D);
zip('SAVE_ORIGINAL_DATA.zip','SAVE_ORIGINAL_DATA.txt');
fclose(fid);

% f is the segment size
f=[15,30,45,60,75,90,105,120,135,150];

for iLOOP=1:length(f)
      % composing the segmented data
      [CS,CD]=UAMovAvgDecimate(D, f(iLOOP));
      % calculating Shannon entropy of the segmented data
      Entropy(iLOOP)=EntropyUA(CD);

      % applying quantization
      ndigits=1;
      Q=quant(CS,10^-ndigits);
      ndigits2=3;
      Z=quant(CS,10^-ndigits2);
      A=quant(CD,10^-ndigits2);

      fid=fopen('SAVE_DCTCOEF_QUANTIZED.txt', 'wt');
      fprintf(fid,[repmat('%30.20f ',1,52) '\n'],CS);
      zip('SAVE_DCTCOEF_QUANTIZED.zip','SAVE_DCTCOEF_QUANTIZED.txt');
```

```
        fclose(fid);

        ORIGINAL_DATA=dir('SAVE_ORIGINAL_DATA.zip');
        BYTES_ORIGINAL_DATA=ORIGINAL_DATA.bytes
        ENCODED_DATA=dir('SAVE_DCTCOEF_QUANTIZED.zip');
        BYTES_ENCODED_DATA=ENCODED_DATA.bytes
        PerCentReduction=(1-BYTES_ENCODED_DATA/BYTES_ORIGINAL_DATA)*100
        RatioReduction(iLOOP)=BYTES_ORIGINAL_DATA/BYTES_ENCODED_DATA
        ERROR_NORM(iLOOP)=sum(abs(D-CS),1);

        fid=fopen('SAVE_QDCTCOEF_QUANTIZED.txt', 'wt');
        fprintf(fid,[repmat('%30.20f ',1,52) '\n'],Q);
        zip('SAVE_QDCTCOEF_QUANTIZED.zip','SAVE_QDCTCOEF_QUANTIZED.txt');
        fclose(fid);

        ENCODED_QDATA=dir('SAVE_QDCTCOEF_QUANTIZED.zip');
        BYTES_ENCODED_QDATA=ENCODED_QDATA.bytes
        PerCentReductionQ=(1-BYTES_ENCODED_QDATA/BYTES_ORIGINAL_DATA)*100
        RatioReductionQ(iLOOP)=BYTES_ORIGINAL_DATA/BYTES_ENCODED_QDATA
        ERROR_NORMQ(iLOOP)=sum(abs(D-Q),1);

        fid=fopen('SAVE_ZDCTCOEF_QUANTIZED.txt', 'wt');
        fprintf(fid,[repmat('%30.20f ',1,52) '\n'],Z);
        zip('SAVE_ZDCTCOEF_QUANTIZED.zip','SAVE_ZDCTCOEF_QUANTIZED.txt');
        fclose(fid);

        ENCODED_ZDATA=dir('SAVE_ZDCTCOEF_QUANTIZED.zip');
        BYTES_ENCODED_ZDATA=ENCODED_ZDATA.bytes
        PerCentReductionZ=(1-BYTES_ENCODED_ZDATA/BYTES_ORIGINAL_DATA)*100
        RatioReductionZ(iLOOP)=BYTES_ORIGINAL_DATA/BYTES_ENCODED_ZDATA
        ERROR_NORMZ(iLOOP)=sum(abs(D-Z),1);
        % calculating Shannon entropy of the quantized segmented data
        EntropyA(iLOOP)=EntropyUA(A);
end

% compression ratio/error norm vs number of segments plots
P=RatioReduction./ERROR_NORM;
R=RatioReductionQ./ERROR_NORMQ;
S=RatioReductionZ./ERROR_NORMZ;
```

```
plot(f(1,:),P(1,:),'rx-');grid on;hold on;
plot(f(1,:),R(1,:),'bo-');grid on;hold on;
plot(f(1,:),S(1,:),'m*-');grid on;hold on;
xlabel('Number of segments','FontSize',10)
ylabel('Compression ratio/Error norm','FontSize',10);
legend('Segmented data (ndigits=15)','Quantized data
(ndigits=1)','Quantized data (ndigits=3)');
% compression ratio vs number of segments plots
subplot(3,1,1)
plot(f(1,:),RatioReduction(1,:),'rx-');grid on;hold on;
plot(f(1,:),RatioReductionQ(1,:),'bo-');grid on;hold on;
plot(f(1,:),RatioReductionZ(1,:),'m*-');grid on;hold on;
xlabel('Number of segments','FontSize',10)
ylabel('Compression ratio','FontSize',10);
legend('Segmented data (ndigits=15)','Quantized data
(ndigits=1)','Quantized data (ndigits=3)');
% error norm vs number of segments plots
subplot(3,1,2)
plot(f(1,:),ERROR_NORM(1,:),'rx-');grid on;hold on;
plot(f(1,:),ERROR_NORMQ(1,:),'bo-');grid on;hold on;
plot(f(1,:),ERROR_NORMZ(1,:),'m*-');grid on;hold on;
xlabel('Number of segments','FontSize',10)
ylabel('Error norm','FontSize',10);
legend('Segmented data (ndigits=15)','Quantized data
(ndigits=1)','Quantized data (ndigits=3)');
% entropy vs number of segments plots
subplot(3,1,3)
plot(f(1,:),Entropy(1,:),'rx-');grid on;hold on;
plot(f(1,:),EntropyA(1,:),'m*-');grid on;hold on;
line([0 150],[EntropyD EntropyD],'Color','b','LineWidth',2);grid on;hold
on;
xlabel('Number of segments','FontSize',10)
ylabel('Entropy','FontSize',10);
legend('Segmented data (ndigits=15)','Quantized data
(ndigits=3)','Original data');
```

function: UAMovAvgDecimate

```
function [CS, CD]=UAMovAvgDecimate(P, f)
      % f : fraction of the number of data points to be represented,
```

```
      %     every f th, CS will show f linear (horizontal) segments
      [nr,nc]=size(P);
      nf=ceil(nr/f);
      ic=ceil(nr/nf);

      for k=1:nc
            for i=1:ic
            sumC=0;
            k0=(nf*(i-1)+1);
            k1=min(nf*i,nr);
                  for j=k0:k1
                  sumC=sumC+P(j,k);
                  end
            L=k1-k0+1;
            CS(k0:k1,k)=sumC/L;
            CD(i,k)=sumC/L;
            end
      end
end
```

function: EntropyUA

```
function S=EntropyUA(x)
      %-- Entropy Calculation (matrix)
      % The entropy of a sequence X = { x_1  x_2  x_3 ... x_N }
      % is defined as  H(X) = sum_1_to_N ( p_i log( p_i ) )
      % where p_i = | x_i | / || X ||
      % and || X || = sum_1_to_N ( | x_i | ).
      absx=abs(x);
      px=absx./repmat(sum(absx),size(x,1),1);
      S=-sum(px.*log2(px));
end
```

A.2. Matlab Code used in One Dimensional and Two Dimensional Discrete Cosine Transform

```
% using overall PortSimHigh data set
load PortSimHigh.txt;
```

```
D=PortSimHigh(:,1:500);
[TotR TotC]=size(D);
D=[mapminmax(D')]';
% discrete cosine transform coefficients
DCTD2=dct2(D);
DCTD1=dct(D);

% pct is the thresholding percentile
pct=[15, 30, 45, 60, 75, 85, 90, 95, 99.5, 99.8];

fid=fopen('SAVE_ORIGINAL_DATA1.txt', 'wt');
fprintf(fid,[repmat('%30.20f ',1,52) '\n'],DCTD1);
zip('SAVE_ORIGINAL_DATA1.zip','SAVE_ORIGINAL_DATA1.txt');
fclose(fid);

fid=fopen('SAVE_ORIGINAL_DATA2.txt', 'wt');
fprintf(fid,[repmat('%30.20f ',1,52) '\n'],DCTD2);
zip('SAVE_ORIGINAL_DATA2.zip','SAVE_ORIGINAL_DATA2.txt');
fclose(fid);

LDCT1=reshape(DCTD1,TotR*TotC,1);
LDCT2=reshape(DCTD2,TotR*TotC,1);
for iLOOP=1:length(pct)
      p1=prctile(abs(LDCT1),pct(iLOOP));
      p2=prctile(abs(LDCT2),pct(iLOOP));

      % thresholding discrete cosine transform coefficients
      DCTD1(find(abs(DCTD1)<p1))=0;
      DCTD2(find(abs(DCTD2)<p2))=0;

      fid=fopen('SAVE_DCTCOEF_FILTERED1.txt', 'wt');
      fprintf(fid,[repmat('%30.20f ',1,52) '\n'],DCTD1);
      zip('SAVE_DCTCOEF_FILTERED1.zip','SAVE_DCTCOEF_FILTERED1.txt');
      fclose(fid);

      fid=fopen('SAVE_DCTCOEF_FILTERED2.txt', 'wt');
      fprintf(fid,[repmat('%30.20f ',1,52) '\n'],DCTD2);
      zip('SAVE_DCTCOEF_FILTERED2.zip','SAVE_DCTCOEF_FILTERED2.txt');
      fclose(fid);
```

```
ORIGINAL_DATA1=dir('SAVE_ORIGINAL_DATA1.zip');
BYTES_ORIGINAL_DATA1=ORIGINAL_DATA1.bytes
ENCODED_DATA1=dir('SAVE_DCTCOEF_FILTERED1.zip');
BYTES_ENCODED_DATA1=ENCODED_DATA1.bytes
PerCentReduction1=(1-BYTES_ENCODED_DATA1/BYTES_ORIGINAL_DATA1)*100
RatioReduction1(iLOOP)=BYTES_ORIGINAL_DATA1/BYTES_ENCODED_DATA1

ORIGINAL_DATA2=dir('SAVE_ORIGINAL_DATA2.zip');
BYTES_ORIGINAL_DATA2=ORIGINAL_DATA2.bytes
ENCODED_DATA2=dir('SAVE_DCTCOEF_FILTERED2.zip');
BYTES_ENCODED_DATA2=ENCODED_DATA2.bytes
PerCentReduction2=(1-BYTES_ENCODED_DATA2/BYTES_ORIGINAL_DATA2)*100
RatioReduction2(iLOOP)=BYTES_ORIGINAL_DATA2/BYTES_ENCODED_DATA2

% reconstruction
IDCTD1=idct(DCTD1);
IDCTD2=idct2(DCTD2);

ERROR_NORM1=sum(abs(D-IDCTD1),1);
ERROR_NORM2=sum(abs(D-IDCTD2),1);

Mean1(iLOOP)=mean(ERROR_NORM1);
Mean2(iLOOP)=mean(ERROR_NORM2);

Relativeglobal_error1(iLOOP)=
mean(100*(sum((D-IDCTD1).^2))./sum(D.^2));
Relativeglobal_error2(iLOOP)=
mean(100*(sum((D-IDCTD2).^2))./sum(D.^2));

iErrMax1=find(max(ERROR_NORM1)==ERROR_NORM1);
iErrMin1=find(min(ERROR_NORM1)==ERROR_NORM1);
MINERROR_NORM1(iLOOP)=ERROR_NORM1(iErrMin1)
MAXERROR_NORM1(iLOOP)=ERROR_NORM1(iErrMax1)

iErrMax2=find(max(ERROR_NORM2)==ERROR_NORM2);
iErrMin2=find(min(ERROR_NORM2)==ERROR_NORM2);
MINERROR_NORM2(iLOOP)=ERROR_NORM2(iErrMin2)
MAXERROR_NORM2(iLOOP)=ERROR_NORM2(iErrMax2)
MaxD1=abs(sum(D(:,iErrMax1)));
```

```
        Relativemax_error1(iLOOP)=100*MAXERROR_NORM1(iLOOP)/MaxD1;
        MaxD2=abs(sum(D(:,iErrMax2)));
        Relativemax_error2(iLOOP)=100*MAXERROR_NORM2(iLOOP)/MaxD2;
end

% compression ratio vs percentile plots
subplot(2,1,1)
plot(pct(1,:),RatioReduction1(1,:),'ro-');grid on;hold on;
plot(pct(1,:),RatioReduction2(1,:),'bo-')
xlabel('Percentile','FontSize',10)
ylabel('Compression ratio','FontSize',10)
legend('1D-DCT','2D-DCT')
% mean error norm vs percentile plots
subplot(2,1,2)
plot(pct(1,:),Mean1(1,:),'ro-');grid on;hold on;
plot(pct(1,:),Mean2(1,:),'bo-');grid on;hold on;
xlabel('Percentile','FontSize',10)
ylabel('Mean error norm','FontSize',10)
legend('1D-DCT','2D-DCT')

figure(2)
% % relative global error vs percentile plots
subplot(2,1,1)
plot(pct(1,:),Relativeglobal_error1(1,:),'ro-');grid on;hold on;
plot(pct(1,:),Relativeglobal_error2(1,:),'bo-');grid on;hold on;
xlabel('Percentile','FontSize',10)
ylabel('% Relative global error','FontSize',10)
legend('1D-DCT','2D-DCT')
% % relative maximum error vs percentile plots
subplot(2,1,2)
plot(pct(1,:),Relativemax_error1(1,:),'ro-');grid on;hold on;
plot(pct(1,:),Relativemax_error2(1,:),'bo-');grid on;hold on;
xlabel('Percentile','FontSize',10)
ylabel('% Relative maximum error','FontSize',10)
legend('1D-DCT','2D-DCT')
% compression ratio/mean error norm vs percentile plots
plot(pct(1,:),RatioReduction1(1,:)./Mean1(1,:),'ro-');grid on;hold on;
plot(pct(1,:),RatioReduction2(1,:)./Mean2(1,:),'bo-');grid on;hold on;
xlabel('Percentile','FontSize',10)
ylabel('Compression ratio/Mean error norm','FontSize',10)
```

```
legend('1D-DCT','2D-DCT')
fclose('all');
```

A.3. Matlab Code used in One Dimensional and Two Dimensional Discrete Wavelet Transform

```
% using overall TEP data set
load simout.mat;
D=simout(:,1:41);
[TotR TotC]=size(D);
D=[mapminmax(D')]';
Wlet='sym4';
WLevel=4;
% one dimensional discrete wavelet transform coefficients
for k=1:TotC
      [C,L1]=wavedec(D(:,k),WLevel,Wlet);
      C1(:,k)=C;
end
% two dimensional discrete wavelet transform coefficients
[C2,L2]=wavedec2(D,WLevel,Wlet);

[TotR1 TotC1]=size(C1);
[TotR2 TotC2]=size(C2);

% pct is the thresholding percentile
pct=[15, 30, 45, 60, 75, 85, 90, 95, 99.5, 99.8];

fid=fopen('SAVE_ORIGINAL_DATA1.txt', 'wt');
fprintf(fid,[repmat('%30.20f ',1,52) '\n'],C1);
zip('SAVE_ORIGINAL_DATA1.zip','SAVE_ORIGINAL_DATA1.txt');
fclose(fid);

fid=fopen('SAVE_ORIGINAL_DATA2.txt', 'wt');
fprintf(fid,[repmat('%30.20f ',1,52) '\n'],C2);
zip('SAVE_ORIGINAL_DATA2.zip','SAVE_ORIGINAL_DATA2.txt');
fclose(fid);

LDCT1=reshape(C1,TotR1*TotC1,1);
```

```
LDCT2=reshape(C2,TotR2*TotC2,1);

for iLOOP=1:length(pct)
      p1=prctile(abs(LDCT1),pct(iLOOP));
      p2=prctile(abs(LDCT2),pct(iLOOP));
      % thresholding discrete wavelet transform coefficients
      C1(find(abs(C1)<p1))=0;
      C2(find(abs(C2)<p2))=0;

      fid=fopen('SAVE_DWTCOEF_FILTERED1.txt', 'wt');
      fprintf(fid,[repmat('%30.20f ',1,52) '\n'],C1);
      zip('SAVE_DWTCOEF_FILTERED1.zip','SAVE_DWTCOEF_FILTERED1.txt');
      fclose(fid);

      fid=fopen('SAVE_DWTCOEF_FILTERED2.txt', 'wt');
      fprintf(fid,[repmat('%30.20f ',1,52) '\n'],C2);
      zip('SAVE_DWTCOEF_FILTERED2.zip','SAVE_DWTCOEF_FILTERED2.txt');
      fclose(fid);

      ORIGINAL_DATA1=dir('SAVE_ORIGINAL_DATA1.zip');
      BYTES_ORIGINAL_DATA1=ORIGINAL_DATA1.bytes
      ENCODED_DATA1=dir('SAVE_DWTCOEF_FILTERED1.zip');
      BYTES_ENCODED_DATA1=ENCODED_DATA1.bytes
      PerCentReduction1=(1-BYTES_ENCODED_DATA1/BYTES_ORIGINAL_DATA1)*100
      RatioReduction1(iLOOP)=BYTES_ORIGINAL_DATA1/BYTES_ENCODED_DATA1

      ORIGINAL_DATA2=dir('SAVE_ORIGINAL_DATA2.zip');
      BYTES_ORIGINAL_DATA2=ORIGINAL_DATA2.bytes
      ENCODED_DATA2=dir('SAVE_DWTCOEF_FILTERED2.zip');
      BYTES_ENCODED_DATA2=ENCODED_DATA2.bytes
      PerCentReduction2=(1-BYTES_ENCODED_DATA2/BYTES_ORIGINAL_DATA2)*100
      RatioReduction2(iLOOP)=BYTES_ORIGINAL_DATA2/BYTES_ENCODED_DATA2

      % reconstruction
      for k=1:TotC
            C=C1(:,k);
            A1(:,k)=waverec(C,L1,Wlet);
      end
      A2=waverec2(C2,L2,Wlet);
```

```
        ERROR_NORM1=sum(abs(D-A1),1);
        ERROR_NORM2=sum(abs(D-A2),1);

        Mean1(iLOOP)=mean(ERROR_NORM1);
        Mean2(iLOOP)=mean(ERROR_NORM2);

        Relativeglobal_error1(iLOOP)=mean(100*(sum((D-A1).^2))./sum(D.^2));
        Relativeglobal_error2(iLOOP)=mean(100*(sum((D-A2).^2))./sum(D.^2));

        iErrMax1=find(max(ERROR_NORM1)==ERROR_NORM1);
        iErrMin1=find(min(ERROR_NORM1)==ERROR_NORM1);
        MINERROR_NORM1(iLOOP)=ERROR_NORM1(iErrMin1)
        MAXERROR_NORM1(iLOOP)=ERROR_NORM1(iErrMax1)

        iErrMax2=find(max(ERROR_NORM2)==ERROR_NORM2);
        iErrMin2=find(min(ERROR_NORM2)==ERROR_NORM2);
        MINERROR_NORM2(iLOOP)=ERROR_NORM2(iErrMin2)
        MAXERROR_NORM2(iLOOP)=ERROR_NORM2(iErrMax2)

        MaxD1=abs(sum(D(:,iErrMax1)));
        Relativemax_error1(iLOOP)=100*MAXERROR_NORM1(iLOOP)/MaxD1;
        MaxD2=abs(sum(D(:,iErrMax2)));
        Relativemax_error2(iLOOP)=100*MAXERROR_NORM2(iLOOP)/MaxD2;
end

% compression ratio vs percentile plots
subplot(2,1,1)
plot(pct(1,:),RatioReduction1(1,:),'ro-');grid on;hold on;
plot(pct(1,:),RatioReduction2(1,:),'b*-')
xlabel('Percentile (%)','FontSize',10)
ylabel('Compression ratio','FontSize',10)
legend('1D-DWT','2D-DWT')
% mean error norm vs percentile plots
subplot(2,1,2)
plot(pct(1,:),Mean1(1,:),'ro-');grid on;hold on;
plot(pct(1,:),Mean2(1,:),'b*-');grid on;hold on;
xlabel('Percentile (%)','FontSize',10)
ylabel('Mean error norm','FontSize',10)
legend('1D-DWT','2D-DWT')
```

```
figure(2)
% % relative global error vs percentile plots
subplot(2,1,1)
plot(pct(1,:),Relativeglobal_error1(1,:),'ro-');grid on;hold on;
plot(pct(1,:),Relativeglobal_error2(1,:),'b*-');grid on;hold on;
xlabel('Percentile (%)','FontSize',10)
ylabel('% Relative global error','FontSize',10)
legend('1D-DWT','2D-DWT')
% % relative maximum error vs percentile plots
subplot(2,1,2)
plot(pct(1,:),Relativemax_error1(1,:),'ro-');grid on;hold on;
plot(pct(1,:),Relativemax_error2(1,:),'b*-');grid on;hold on;
xlabel('Percentile (%)','FontSize',10)
ylabel('% Relative maximum error','FontSize',10)
legend('1D-DWT','2D-DWT')
% compression ratio/mean error norm vs percentile plots
plot(pct(1,:),RatioReduction1(1,:)./Mean1(1,:),'ro-');grid on;hold on;
plot(pct(1,:),RatioReduction2(1,:)./Mean2(1,:),'b*-');grid on;hold on;
xlabel('Percentile (%)','FontSize',10)
ylabel('Compression ratio/Mean error norm','FontSize',10)
legend('1D-DWT','2D-DWT')
fclose('all');
```

A.4. Matlab Code used in the Trajectory Matrix Construction and Two Dimensional Discrete Cosine Transform

```
% using 30th column of the TEP data set
load simout.mat;
A=simout(1:100,30);
[TotR TotC]=size(A);
A=[mapminmax(A')]';
pct=99.5;
% discrete cosine transform coefficients
DCTD1=dct2(A);

fid=fopen('SAVE_ORIGINAL_DATA.txt', 'wt');
fprintf(fid,[repmat('%30.20f ',1,52) '\n'],DCTD1);
zip('SAVE_ORIGINAL_DATA.zip','SAVE_ORIGINAL_DATA.txt');
fclose(fid);
```

```
LDCT1=reshape(DCTD1,TotR*TotC,1);
p1=prctile(abs(LDCT1),pct)
% thresholding discrete cosine transform coefficients
DCTD1(find(abs(DCTD1)<p1))=0;

fid=fopen('SAVE_DCTCOEF_FILTERED.txt', 'wt');
fprintf(fid,[repmat('%30.20f ',1,52) '\n'],DCTD1);
zip('SAVE_DCTCOEF_FILTERED.zip','SAVE_DCTCOEF_FILTERED.txt');
fclose(fid);

ORIGINAL_DATA=dir('SAVE_ORIGINAL_DATA.zip');
BYTES_ORIGINAL_DATA=ORIGINAL_DATA.bytes
ENCODED_DATA=dir('SAVE_DCTCOEF_FILTERED.zip');
BYTES_ENCODED_DATA=ENCODED_DATA.bytes
PerCentReduction1=(1-BYTES_ENCODED_DATA/BYTES_ORIGINAL_DATA)*100
RatioReduction1=BYTES_ORIGINAL_DATA/BYTES_ENCODED_DATA

% reconstruction
IDCTD1=idct2(DCTD1);
ERROR_NORM1=sum(abs(A-IDCTD1),1);
Mean1=mean(ERROR_NORM1);

% composing trajectory matrix
N=TotR;
LK_ratio=[0.1, 0.4, 1, 1.6, 2, 2.5, 3, 3.6, 4, 4.5];
for iLOOP=1:length(LK_ratio)
      L=floor((LK_ratio(iLOOP))*(N+1)/(1+LK_ratio(iLOOP)))
      K=N-L+1
      D=zeros(L,K);
      for i=1:K
            D(1:L,i)=A(i:L+i-1);
      end

      % discrete cosine transform coefficients
      DCTD=dct2(D);

      fid=fopen('SAVE_ORIGINAL_DATA.txt', 'wt');
      fprintf(fid,[repmat('%30.20f ',1,52) '\n'],DCTD);
      zip('SAVE_ORIGINAL_DATA.zip','SAVE_ORIGINAL_DATA.txt');
      fclose(fid);
```

```
        LDCT=reshape(DCTD,K*L,1);
        p=prctile(abs(LDCT),pct)
        % thresholding discrete cosine transform coefficients
        DCTD(find(abs(DCTD)<p))=0;

        fid=fopen('SAVE_DCTCOEF_FILTERED.txt', 'wt');
        fprintf(fid,[repmat('%30.20f ',1,52) '\n'],DCTD);
        zip('SAVE_DCTCOEF_FILTERED.zip','SAVE_DCTCOEF_FILTERED.txt');
        fclose(fid);

        ORIGINAL_DATA=dir('SAVE_ORIGINAL_DATA.zip');
        BYTES_ORIGINAL_DATA=ORIGINAL_DATA.bytes
        ENCODED_DATA=dir('SAVE_DCTCOEF_FILTERED.zip');
        BYTES_ENCODED_DATA=ENCODED_DATA.bytes
        PerCentReduction(iLOOP)=
        (1-BYTES_ENCODED_DATA/BYTES_ORIGINAL_DATA)*100
        RatioReduction(iLOOP)=BYTES_ORIGINAL_DATA/BYTES_ENCODED_DATA

        % reconstruction
        IDCTD=idct2(DCTD);
        ERROR_NORM=sum(abs(D-IDCTD),1);
        Mean(iLOOP)=mean(ERROR_NORM);

        iErrMax=find(max(ERROR_NORM)==ERROR_NORM)
        iErrMin=find(min(ERROR_NORM)==ERROR_NORM)
        MINERROR_NORM(iLOOP)=ERROR_NORM(iErrMin)
        MAXERROR_NORM(iLOOP)=ERROR_NORM(iErrMax)
end

% compression ratio vs L/K ratio plots
subplot(2,1,1)
plot(LK_ratio(1,:),RatioReduction(1,:),'ro-');grid on;hold on;
line([0 4.5],[RatioReduction1 RatioReduction1],'Color','b','LineWidth',2)
grid on;hold on;
xlabel('L/K ratio','FontSize',10)
ylabel('Compression ratio','FontSize',10)
legend('trajectory matrix','original data')
% error norm vs L/K ratio plots
subplot(2,1,2)
```

```
plot(LK_ratio(1,:),MINERROR_NORM(1,:),'ro-');grid on;hold on;
plot(LK_ratio(1,:),Mean(1,:),'bx-');grid on;hold on;
plot(LK_ratio(1,:),MAXERROR_NORM(1,:),'m.-');grid on;hold on;
xlabel('L/K ratio','FontSize',10)
ylabel('Error norm','FontSize',10)
legend('min error norm','mean error norm','max error norm')
```

REFERENCES

Bakshi, B.R. and G. Stephanopoulos, 1996, "Compression of Chemical Process Data by Functional Approximation and Feature Extraction", *AIChE Journal 1996,* Vol. 42, No. 2, pp. 477-492.

Benouaret, M., A. Sahour and S. Harize, 2012, "Real time implementation of a signal denoising approach based on eight-bits DWT", *International Journal of Electronics and Communications,* In Press, Corrected Proof, http://dx.doi.org/10.1016/j.aeue.2012.04.001, accessed at May 2012.

Blelloch, G.E., 2010, *Introduction to Data Compression*, Carnegie Mellon University, http://www.cs.cmu.edu/~guyb/realworld/compression.pdf, accessed at January 2012.

Bristol, E.H., 1990, "Swinging Door Trending: Adaptive Trend Recording?", *Advances in Instrumentation and Control*; *Instrument Society of America: Research Triangle Park, NC, 1990,* Vol. 45, pp. 749-754.

Chau, F., Y. Liang, J. Gao and X. Shao, 2004, *Chemometrics From Basics to Wavelet Transform*, John Wiley & Sons, Inc., New York, NY, USA.

Chun-Lin, Liu, 2010, *A Tutorial of the Wavelet Transform*, http://disp.ee.ntu.edu.tw/tutorial/WaveletTutorial.pdf, accessed at April 2012.

Conradie, A.v.E. and C. Aldrich, 2005, "Development of Neurocontrollers with Evolutionary Reinforcement Learning", *Computers & Chemical Engineering 2005,* Vol. 30, pp. 1-17.

Dillard, B. and G. Shmueli, 2004, "Simultaneous Analysis of Multiple Time Series Using Two-Dimensional Wavelets", Department of Decision and Information Technologies, University of Maryland, College Park, MD, pp. 1-19.

Downs, J.J. and E.F. Vogel, 1993, "A Plant-Wide Industrial Process Control Problem", *Computers Chemical Engineering 1993,* Vol. 17, No. 3, pp. 245-255.

Ge, Z. and Z. Song, 2007, "Process Monitoring Based on Independent Component Analysis Principal Component Analysis (ICA-PCA) and Similarity Factors", *Industrial Engineering Chemistry Research 2007,* Vol. 46, pp. 2054-2063.

Golyandina, N., V. Nekrutkin and A. Zhigljavsky, 2001, *Analysis of Time Series Structure: SSA and Related Techniques,* Chapman & Hall/CRC, Florida, USA.

Hale, J.C. and H.L. Sellars, 1981, "Historical Data Recording For Process Computers", *Chemical Engineering Progress 1981,* Vol. 77, No. 11, pp. 38-43.

Hassani, H., S. Heravi and A. Zhigljavsky, 2009, "Forecasting European industrial production with singular spectrum analysis", *International Journal of Forecasting 2009,* Vol. 25, pp. 103-118.

Imtiaz, S.A., M.A. Choudhury and S.L. Shah, 2007, "Building Multivariate Models from Compressed Data", *Industrial Engineering Chemistry Research 2007,* Vol. 46, pp. 481-491.

Jockenhövel, T., L.T. Biegler and A. Wachter, 2003, "Dynamic optimization of the Tennessee Eastman process using the OptControlCentre", *Computers and Chemical Engineering 2003,* Vol. 27, pp. 1513-1531.

Keogh, E., K. Chakrabarti, M. Pazzani and S. Mehrotra, 2000, "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases", *Journal of Knowledge and Information Systems*.

Khayam, S.A., 2003, *The Discrete Cosine Transform (DCT): Theory and Application*, Michigan State University ECE 802-602: Information Theory and Coding Seminar 1, http://www.wisnet.seecs.nust.edu.pk/people/~khayam/pdf/DCT_TR802.pdf, accessed at January 2012.

Liu, Y., 2012, "Dimensionality reduction and main component extraction of mass spectrometry cancer data", *Knowledge-Based Systems, 2012,* Vol. 26, pp. 207-215.

Lkhagva, B., Y. Suzuki and K. Kawagoe, 2006, "Extended SAX: Extension of Symbolic Aggregate Approximation for Financial Time Series Data Representation", *DEWS, 2006, 4A-i8*.

Lu, N., Y. Yang, F. Gao and F. Wang, 2004, "Multirate dynamic inferential modeling for multivariable processes", *Chemical Engineering Science, 2004,* Vol. 59, pp. 855-864.

Mah, R.S.H., A.C. Tamhane, S.H. Tung and A.N. Patel, 1995, "Process Trending with Piecewise Linear Smoothing", *Computers and Chemical Engineering 1995,* Vol. 19, pp. 129-137.

McAvoy, T.J., 1999, "Synthesis of Plantwide Control Systems Using Optimization", *Industrial Engineering Chemistry Research 1999,* Vol. 38, pp. 2984-2994.

McAvoy, T. and N. Ye, 1994, "Base control for the Tennessee Eastman Problem", *Computers and Chemical Engineering 1994,* Vol. 18, No. 5, pp. 383-413.

Misiti, M., Y. Misiti, G. Oppenheim and J.M. Poggi, 2011, *Wavelet Toolbox for use with MATLAB*, MathWorks, Inc., USA.

Misra, M., S.J. Qin, S. Kumar and D. Seeman, 2000, "On-Line Data Compression and Error Analysis Using Wavelet Technology", *AIChE Journal, 2000,* Vol. 46, No. 1, pp. 119-132.

Mitra, S. and T. Acharya, 2003, *Data Mining Multimedia, Soft Computing and Bioinformatics*, John Wiley & Sons, Inc., New York, NY, USA.

Moskvina, V.G. and A. Zhigljavsky, 2003, "An algorithm based on singular spectrum analysis for change-point detection", *Communication in Statistics - Simulation and Computation, 2003,* Vol. 32, No. 4, pp. 319-352.

Nelson, M. and J.L. Gailly, 1996, *The Data Compression Book*, Second edition, M&T Books, New York, NY, USA.

Pu, I.M., 2006, *Fundamental Data Compression*, Elsevier, Britain.

Rao, K.R. and P.C. Yip, 2001, *The Transform and Data Compression Handbook*, CRC Press LLC, Florida, USA.

Ricardez-Sandoval, L.A., H.M. Budman and P.L. Douglas, 2009, "Simultaneous design and control of chemical processes with application to the Tennessee Eastman process", *Journal of Process Control, 2009,* Vol. 19, pp. 1377-1391.

Ricker, N.L. and J.H. Lee, 1995, "Nonlinear Model Predictive Control of the Tennessee Eastman Challenge Process", *Computers and Chemical Engineering 1995,* Vol. 19, No. 9, pp. 961-981.

Salomon, D., 2007, *Data Compression, The Complete Reference*, Fourth edition, Springer-Verlag London Limited, Britain.

Salomon, D., 2008, *A Concise Introduction to Data Compression*, Springer-Verlag London Limited, Britain.

Sayood, K., 2003, *Lossless Compression Handbook*, Elsevier Science, California, USA.

Sayood, K., 2006, *Introduction to Data Compression*, Third edition, Elsevier Inc., San Francisco, USA.

Singhal, A. and D.E. Seborg, 2005, "Effect of Data Compression on Pattern Matching in Historical Data", *Industrial Engineering Chemistry Research 2005*, Vol. 44, pp. 3203-3212.

Stark, H., 2005, *Wavelets and Signal Processing*, Springer-Verlag Berlin Heidelberg, Netherlands.

Thornhill, N.F., S. Choudhury and S.L. Shah, 2004, "The impact of compression on data-driven process analyses", *Journal of Process Control, 2004,* Vol. 14, pp. 389-398.

Vedam, H., V. Venkatasubramanian and M. Bhalodia, 1998, "A B-Spline based Method for Data Compression, Process Monitoring and Diagnosis", *Computers Chemical Engineering 1998,* Vol. 22, Suppl., pp. S827-S830.

Watson, M.J., A. Liakopoulos, D. Brzakovic and C. Georgakis, 1998, "A Practical Assessment of Process Data Compression Techniques", *Industrial Engineering Chemistry Research 1998,* Vol. 37, pp. 267-274.

Weeks, M., 2007, *Digital Signal Processing Using MATLAB and Wavelets*, Infinity Science Press LLC, Massachusetts, USA.

Zerkaoui, S., F. Druaux, E. Leclercq and D. Lefebvre, 2010, "Indirect neural control for plant-wide systems: Application to the Tennessee Eastman Challenge Process", *Computers and Chemical Engineering, 2010,* Vol. 34, pp. 232-243.

Zhao, S.J., J. Zhang and Y.M. Xu, 2004, "Monitoring of Processes with Multiple Operating Modes through Multiple Principle Component Analysis Models", *Industrial Engineering Chemistry Research 2004,* Vol. 43, pp. 7025-7035.