

January 01, 2009

Multi-resolution approach to identification of recurring patterns in process signal

Tarun M. Kothia
Northeastern University

Recommended Citation

Kothia, Tarun M., "Multi-resolution approach to identification of recurring patterns in process signal" (2009). *Engineering Management Master's Theses*. Paper 2. <http://hdl.handle.net/2047/d10017008>

This work is available open access, hosted by Northeastern University.

Multi-resolution Approach to Identification of Recurring Patterns in Process Signal

A Thesis Presented

By

Tarun M Kothia

to

Department of Mechanical and Industrial Engineering

in partial fulfillment of the requirements

for the degree of

Master of Science

in

Engineering Management

Northeastern University

Boston, Massachusetts

January 2009

NORTHEASTERN UNIVERSITY

Graduate School of Engineering

Thesis Title: Multi-resolution Approach to Identification of Recurring Patterns in
Process Signal

Author: Tarun M Kothia

Department: Mechanical and Industrial Engineering Department

Approved for Thesis Requirement of the Master of Science Degree

Thesis Advisor, **Dr. Sagar Kamarthi**

Date

Thesis Advisor **Dr. Ibrahim Zeid**

Date

Chairman of the MIE Department, **Dr. Hameed Metghalchi**

Date

Director of Graduate School, **Dean Yaman Yener**

Date

Copy Deposited in Library

Reference Librarian

Date

1. Abstract

Process control is important for enhancing the quality of manufactured products. Manufacturing processes are generally monitored by observing uniformly sampled process signals collected from application specific sensors and comparing them against known standard patterns. Effective process monitoring and control requires identification of different types of variation, including recurring patterns, in process variables. From the process control view point, any repeating patterns in the process measurements will warrant an investigation into potentially assignable causes. In order to devise an effective process control scheme, a novel universally applicable method for the identification of the repeated occurrence of patterns in process measurements is described in this thesis. First the sampled process signal is decomposed into signals of different resolution using *à trous* translation invariant wavelet transform. Next, a frequency index is assigned to every sampling point of the process signal at every resolution level to improve the pattern recognition. Recurring patterns detected at different resolutions using neighborhood search in Euclidian space. The experimental results show that the method used in this work accurately detects a broader family of recurring patterns even in the presence of noise

2. Acknowledgement

I would like to express my sincere gratitude to my thesis advisors, Professor Sagar Kamarthi and Professor Ibrahim Zeid, for their unremitting guidance and support. Their profound knowledge and expertise in my research topic have been of great value for me. Their direction and support has nourished my research abilities and made me confident to rise after faltering. I feel privileged to have been able to work under their guidance and be associated with research work at Mechanical and Industrial Engineering Department, Northeastern University.

I would like to thank my parents Mr. R.M.S Kothia and Mrs. Sushila Kothia, and my family for giving me all support during my years at Northeastern University. They have taken great pains and have sacrificed to see me prosper in life and studies. My successful completion of MS thesis would not have been possible without their loving support, caring thoughts and wishes, and endless patience.

In the end, I would like to show my gratitude towards all my friends for being there during the lows and the highs of my life.

Tarun M Kothia

3. Table of Contents

1.	ABSTRACT.....	1
2.	ACKNOWLEDGEMENT.....	2
3.	TABLE OF CONTENTS.....	3
	LIST OF FIGURES.....	5
	LIST OF TABLES.....	6
	CHAPTER 1.....	7
	INTRODUCTION.....	7
	<i>1.1 Overview.....</i>	<i>7</i>
	<i>1.2 Problem Statement.....</i>	<i>8</i>
	<i>1.3 Research Objectives.....</i>	<i>8</i>
	<i>1.4 Motivation.....</i>	<i>9</i>
	<i>1.5 Thesis Outline.....</i>	<i>9</i>
	CHAPTER 2.....	10
	A REVIEW OF TIME SERIES AND PATTERN SEARCH.....	10
	<i>2.1 Process Control.....</i>	<i>10</i>
	<i>2.2 Time Series Analysis.....</i>	<i>11</i>
	2.2.1 Time Series Representation.....	11
	2.2.2 Deductive Approach.....	11
	2.2.3 Inductive Approach.....	12
	<i>2.3 Indexing.....</i>	<i>13</i>
	CHAPTER 3.....	17
	A REVIEW OF WAVELET TRANSFORMS.....	17
	<i>3.1 Overview of Wavelet Transforms.....</i>	<i>17</i>
	<i>3.2 Transform Analysis.....</i>	<i>18</i>
	3.2.1 Fourier Transform (FT).....	19
	3.2.2 Short Time Fourier Transform (STFT).....	19
	<i>3.3 Wavelet Transform (WT).....</i>	<i>20</i>
	3.3.1 Continuous Wavelet Transform.....	21
	3.3.2 Discrete Wavelet Transform.....	23
	3.3.3 Multi-Resolution Decomposition:.....	24
	3.3.4 Fast Wavelet Transform.....	26

3.3.5 Translation Invariant Wavelet Transform	29
3.3.6 The <i>à trous</i> Algorithm	29
CHAPTER 4	33
METHOD TO DETECT RECURRING PATTERNS IN PROCESS SIGNAL	33
4.1 <i>Introduction</i>	33
4.2 <i>Preprocessing Stage</i>	33
4.2.1 Computation of Frequency Index	34
4.2.2 Process Signal with Local Frequency Indices	35
4.3 <i>Recurring Pattern Detection</i>	36
4.3.1 Defining a Primary Set of Patterns in the Process Signal	36
4.3.2 Defining Boxes in a Euclidean Space	37
4.3.3 Determining the Minimal Length of the Subsignal	41
4.3.4 Neighborhood Search in the Euclidean Space IR^{2t}	42
4.3.5 Determining the Value of P and Q	44
CHAPTER 5	47
EXPERIMENTATION	47
5.1 <i>Introduction</i>	47
5.2 <i>Experimental Results</i>	47
5.3 <i>Discussion</i>	55
CHAPTER 6	57
CONCLUSION	57
6.1 <i>Summary of Observations</i>	57
6.2 <i>Future Work</i>	58
BIBLIOGRAPHY	59
APPENDIX	66
METHOD TO IDENTIFY RECURRING PATTERNS IN A PROCESS SIGNAL	66
MULTI-SCALE DECOMPOSITION USING A TROUS ALGORITHM	73

List of Figures

Figure 5.1: Original process signal	48
Figure 5.2: First pair of systematic pattern contaminated with additive white noise	48
Figure 5.3: Second pair of systematic pattern contaminated with additive white noise...	49
Figure 5.4: Test process signal of length 256 with inserted systematic patterns	49
Figure 5.5: Multi-resolution analysis of sampled process signal.....	54

List of Tables

Table 5.1: Distribution of parameters in enriched process signal.....	50
Table 5.2: Comparison of the constructed and the detected pairs	52
Table 5.3: Calculated values of P and Q for each decomposed level.....	53
Table 5.4: Detected pairs of systematic subsequences at different resolution levels	55

Chapter 1

Introduction

1.1 Overview

Identification of recurring signal patterns from an ongoing process is important to investigate potentially assignable causes and to take proper corrective measures. A pattern that repeats itself along the process signal is referred to as a recurring pattern.

Pattern search plays an important role in manufacturing process control. Consider a process signal measured at equal intervals using an instrument or sensor attached to a manufacturing process. The important goal of process control in manufacturing is to detect undesirable patterns to avoid certain undesirable process conditions that would adversely affect product quality.

The detection of abnormal patterns provides many direct benefits which include improved product quality, effective operating machine control, and good manufacturing decisions.

Pattern detection methods have been studied extensively in the literature. In this thesis, a novel method is proposed to recognize recurring patterns in a process signal. This method is applied at different scales of process signal decomposed by using *à trous* algorithm.

1.2 Problem Statement

Finding irregular or repetitive patterns in a signal is considered an important tool for detecting formidable changes in a manufacturing process. Most of the existing techniques try to find patterns by comparing them against a given reference pattern. A novel universally applicable method for identifying the repeated occurrence of unknown patterns in process measurements would be useful to devise an effective process control scheme. The method developed in this thesis detects patterns more efficiently by associating frequency index to each amplitude value of the process signal.

1.3 Research Objectives

This thesis presents a method to find recurring signal patterns even in the presence of noise. The original signal is enriched by assigning a frequency index computed via Haar wavelet transform next to every signal measurement. Then the pattern search method is applied to the enriched version of original signal to find recurring patterns. The sampled process signal is decomposed into multi-resolution signal using *a trous* algorithm and the method is applied at different resolutions to find recurring patterns. The steps in the pattern search method are as follows:

- Representing all subsequences (of a certain minimum length) of the process signal as points in a Euclidean space
- Storing the subsequences in special data structures
- Performing an efficient neighborhood search for similar subsequences
- Extending similar subsequences to their full length

1.4 Motivation

The need to discovery of recurring patterns is important in many application areas such as bioinformatics and manufacturing process control. There exists a vast body of literature on efficiently locating known patterns known a priori. In contrast not much work has been done with regard to recognizing recurring pattern of arbitrary shape. Most of the existing methods use only the amplitude values for pattern detection. Sometimes subsequences with approximately equal amplitude but different frequency are detected as recurring patterns. So time-frequency decomposition is used to obtain coefficients to overcome this problem. The input signal may consist of different recurring patterns at different frequencies. This motivated to use multi resolution decomposition to decompose the input signal into different levels and analyze these recurring patterns at various levels.

1.5 Thesis Outline

Chapter 2: It reviews the literature about research in pattern identification in time series.

Chapter 3: It provides an overview of wavelet transforms and multi-resolution analysis.

Chapter 4: It proposes a method for identification of recurring patterns in a signal. The details of method along with steps to find similar patterns are presented.

Chapter 5: It presents the results obtained by applying the proposed algorithms on a simulated data set.

Chapter 6: It presents the conclusion of the work, and offers direction for future work.

Appendix: It lists the source code of the algorithm implemented in MATLAB.

Chapter 2

A Review of Time Series and Pattern Search

2.1 Process Control

Process control helps to maintain quality in manufacturing products. The main goal of process control is to monitor every variation in an observed process variable such as diameter of work piece, machine vibration, temperature, pressure, torque and voltage. Monitoring an industrial process usually involves signal collection, signal analysis, and fault identification. A process is called “in control” if a process variable exhibits a random pattern around a fixed nominal value. It is termed as “out of control” if a systematic or a distinct pattern is found in a quality related variable. If a process is out-of-control, one needs to look for assignable causes to fix the problem. A definite diagnosis is possible only after a systematic pattern reoccurs more than once in a process signal. These patterns in the process signal can be found by a pattern recognition approach.

Manufacturing processes are generally monitored by observing uniformly sampled signals collected from application specific sensors. A uniformly sampled signal from a manufacturing process results in time series data. Searching and detecting unnatural patterns from the resultant time series data using a proper pattern recognition and similarity search method is an important part of process monitoring. This is also referred as time series analysis.

2.2 Time Series Analysis

Time series is a collection of observations made sequentially through time¹. These observations are assumed to possess systematic patterns and random noise. Generally time series analysis is done to obtain insight into phenomenon, to discover repetitive patterns and trend, and to predict future. The presence of random noise makes the pattern discovery difficult. A wavelet transform has denoising property. It has the potential for better discrimination between noise and real data². Time series analysis has many applications apart from process and quality control such as economic forecasting, sales forecasting, workload projections and many, many more.

2.2.1 Time Series Representation

This section gives a brief description about the representation of a time series. Time series partitioning can be done in to two approaches called Deductive approach and Inductive approach³. A deductive approach classifies the data set in time series by matching it with shape descriptors given in advance and inductive approach uses a clustering technique to classify similar data sets in the time series.

2.2.2 Deductive Approach

The commonly used shape descriptors for deductive approach are “linearly increasing,” “convexly increasing,” and “concavely decreasing.” Breaking down a time series into many subsequences using these basic shapes is theoretically easy but the influence of

noise in data leads to large number of tiny segments and many local extrema. So the challenge in this approach is the correct classification of the patterns in the presence of noise. This issue is addressed by using function approximation techniques. The description of time series is extracted from the approximated function rather the original series. The noise is removed by applying a regression technique.

2.2.3 Inductive Approach

In this approach, the given time series is represented partitioned using a clustering technique and the distance between the clusters is calculated. The most commonly used distance measure is Euclidean distance. There are various approaches in the literature. The generally used approaches are given below³:

Clustering of Embedded Subsequences

The content of a given time series is transformed into a number of observations by moving a window of constant length over it. Mostly it is used when the data is being measured uniformly over time. So this embeds a constant number of consecutive values into n -dimensional vector of observations. This is one of the most popular methods used in the past^{4, 5}.

Clustering of Embedded Models

Instead of representing the time series in its original form, it is processed or enriched with additional information. This gives a more abstract representation of the time series, which can be embedded in a vector. Then clustering is performed on these embedded models

instead of on embedded series. In this thesis, a model which contains additional information next to each raw data point is used.

Clustering by Warping Costs

A dynamic time warping technique can be used to locally shrink and stretch the time axis such that a point-to-point similarity of the two time series is reduced^{6, 7}. For any pair of series dynamic time warping yields some costs. This is considered as a distance between time series. This technique is used when other effects like vertical scaling are not significant.

The other approaches include clustering using Markov Models where a hidden Markov model or Markov chain for each subsequence is learned and clustering is done in the resulting probability model.

Symbolization is another way to represent a time series. This is called discrete representation of the time series⁸. The commonly observed problem in these methods is the dimensionality of the subsequences. Another drawback of discretization is that the distance measures defined in symbolic sequences have very little correlation with the true distances between the original “raw” time series⁸.

2.3 Indexing

Majority of work in the literature related to time series pattern search focuses on indexing time series and efficiently finding of known patterns^{9, 10, 11, 12, 13}. There are a variety of algorithms for indexing time series. These algorithms are used for indexing time series to

bring the data into a main storage and thus provide fast information retrieval. Researchers utilize several advanced techniques from algorithm theory including tree structures. Most commonly used tree structure was R-trees for indexing points in multi-dimensional space. Although traditionally R-trees are used to index spatial data, it also allows indexing time series data by their position in feature space^{14, 15}.

Indexing supports efficient retrieval and matching of time series, but dimensionality is an issue in such a method. Various dimensionality reduction methods are applied to reduce the dimensions. For example, transformations such as Discrete Fourier transform (DFT), Discrete Wavelet Transform (DWT), Singular Value Decomposition (SVD), and Piecewise Aggregate Approximation (PAA) are applied to time series to reduce the dimensionality of the feature vectors. An overview of wavelet transforms is given in the next chapter.

Segmentation is another commonly used task in time series analysis. Given a time series containing certain number of data points, a new model is constructed from piecewise segments, which closely approximate the time series. The number of data points in segments is less than the number of points in time series. Segmentation is performed to create a high level representation of the time series that supports indexing, clustering and classification. Segmentation algorithms are grouped into any one of the following categories, namely, sliding windows, top-down, and bottom-up¹⁶.

An algorithm, named PROJECTION, applies random projections to find similar patterns in nucleotide sequences¹⁷. This algorithm is used for time series data after symbolic representation¹⁸.

The basic methodology in pattern recognition is to identify patterns which resemble the target pattern of interest. There exists many methods, which efficiently locate a target pattern of interest in time series^{9, 10, 11, 12, 13, 19}. In contrast, the present work aims to find systematic or unnatural patterns in process signal without the prior knowledge of the pattern shape or length. Finding unknown patterns in a time series is more interesting and challenging. Unknown patterns are referred to as “motifs” because of their close analogy to their discrete counterparts in computation biology^{19, 20}

The pattern search problem can be decomposed into two sub-problems: a supervised learning problem and an unsupervised learning problem. In a supervised learning problem, exemplars of patterns are provided and the time series is labeled according to whether they contain the patterns, whereas in an unsupervised problem the time series is used to generate an approximation to the input. PERUSE is one of the unsupervised algorithm for finding recurring patterns in time series, it was developed by Oates²¹. The goal of this algorithm is to discover the patterns that are more frequently used to generate segments of time series data. The time series is assumed to be generated by repeatedly selecting a pattern according to some distribution and then generating an exemplar of the chosen pattern. Here the window size used is user defined parameter. This thesis deals with the unknown patterns where one doesn't have an idea about their shape and size.

The present algorithm uses an optimal window length calculated by an equation which is explained in chapter 4.

Earlier methods reported in the literature try to detect only a few pre-defined classes of odd patterns by analyzing mostly the global and qualitative information. For example periodic segments of a process signal are detected by means of the Discrete Fourier Transform^{37, 38, 39, 40}. The proposed method here relies on a time-frequency analysis.

Chapter 3

A Review of Wavelet Transforms

3.1 Overview of Wavelet Transforms

Wavelets have become a popular and useful tool for investigation and analysis of a wide variety of problems. It has gained remarkable attention and growth in the recent time. It is applied in several fields from climate analysis to the analysis of financial indices, from heart monitoring to the condition monitoring of rotating machinery, from seismic signal denoising to the denoising of astronomical images and so on²². Brief history of early research on wavelets can be found in Polikar²³.

While the Fourier transform deals with transforming the time domain components to frequency domain and frequency analysis, the wavelet transform deals with scale analysis. The limitation of the Fourier transform to analyze non-stationary signals and to represent the time and frequency domain information at the same time has led to the emergence of the wavelet transforms.

Because of their capability to give detail spatial-frequency information, wavelets can discriminate better between noise and the real data. They are less complex in computation than other transforms. This chapter will present a brief review of various transforms including Fourier transforms and multi-resolution decomposition using wavelets. The *à trous* algorithm is used for wavelet decomposition in this thesis and the advantages for using this algorithm is explained in section 3.3.6.

3.2 Transform Analysis

Engineering problems consists of complex data and the easiest way to simplify and accelerate problem solution is through transform analysis. Transforms still remain the best tools for signal processing applications. Instead of finding the response of a complex system using the given input signal, one can take a simpler approach using transforms. The input signal is broken down into simple signals and then the simple signal is used for analyzing the characteristics and behavior of a complex system.

Fourier analysis was invented in early 1800s, when Joseph Fourier thought that he could superpose *sines* and *cosines* to represent other functions. He discovered that all periodic functions could be expressed as a weighted sum of basic trigonometric functions²⁴. It is represented as the sum of perfectly smooth oscillations. It can be mathematically stated as follows²⁵:

$$f(t) = \frac{1}{2}a_o + \sum_1^{\infty} (a_k \cos kt + b_k \sin kt)$$

$$a_o = \frac{1}{2\pi} \int_0^{2\pi} f(t) d(t)$$

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(t) \cos(kt) dt$$

$$b_k = \frac{1}{\pi} \int_0^{2\pi} f(t) \sin(kt) dt$$

where a_k and b_k are the Fourier coefficients and t the duration of the period of the function $f(t)$. The coefficient a_0 represents the aperiodic fraction of the function. This famous mathematical statement is known as the Fourier series.

3.2.1 Fourier Transform (FT)

Fourier transform is a mathematical tool to transform the given signal from time domain to frequency domain. It is used to decompose or separate a waveform into a sum of sinusoids of different frequencies. For a continuous signal $x(t)$, the FT is given by :

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt$$

where $X(f)$ is the Fourier transform; and $e^{-j2\pi ft} = \cos(2\pi ft) - j \sin(2\pi ft)$ gives the frequency components in $x(t)$. The limitation of FT is that it cannot offer both time and frequency localization of a signal at the same time.

3.2.2 Short Time Fourier Transform (STFT)

To overcome the limitations of Fourier transforms, a revised version of FT called Short Time Fourier Transform is developed by Gabor²⁶. The advantage of STFT is that it uses small time shifted fixed length window $g(t)$ to approximate time-frequency information giving bands of frequencies over a time increment. STFT decomposes a stationary signal $x(t)$ into a two dimensional time-frequency representation $S(\tau, f)$ at different times τ . Thus the FT of windowed signal $x(t)g^*(t-\tau)$ yields STFT as

$$STFT_x(\tau, f) = \int_{-\infty}^{\infty} x(t)g^*(t - \tau)e^{-j2\pi ft} dt$$

The size of the window is important for obtaining accurate results of the STFT because the signal inside a time window is stationary. There is always a trade off between time resolution and frequency resolution in STFT.

3.3 Wavelet Transform (WT)

Wavelet algorithms can process data at various scales or resolutions. Fixed resolution limitation of STFT is resolved by letting the time resolution and frequency resolution vary in order to obtain multi-resolution analysis. WT provides a flexible time-frequency window. The idea of the wavelet transform is first introduced in 1982 by a French geophysical engineer Jean Morlet²⁷. In his analysis, signals consists of high frequency components in short time duration and low frequency components in long time duration. Wavelet can be represented in continuous domain as well as discrete domain which are called as Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT) respectively. Wavelets are computationally less complex than other transforms. They are easy to implement and calculate. It requires less computational time for compression and decompression than the traditional transforms.

3.3.1 Continuous Wavelet Transform

Let $x(t)$ be any square integrable function. Then the CWT or continuous-time wavelet transform of $x(t)$ with respect to a wavelet $\psi(t)$ can be written as:

$$W(a,b) = w(a) \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t-b}{a} \right) dt$$

where a and b are the dilation or scale parameter and translation parameter respectively and $w(a)$ is a weighing function.

The wavelet transform can be thought of as the cross-correlation of the signal with a set of wavelets of various widths. For the purpose of energy conservation, $w(a)$ is set to $1/\sqrt{a}$. This ensures that the wavelets at every scale all have the same energy. The asterisk indicates that the complex conjugate of the wavelet function is used in the transform.

Rewriting the above equation with $w(a) = 1/\sqrt{a}$, one gets

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t-b}{a} \right) dt$$

The normalized wavelet function is written as

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right)$$

where the normalization is in the sense of wavelet energy.

Hence the transform can be expressed as

$$W(a, b) = \int_{-\infty}^{\infty} x(t) \psi_{a,b}^*(t) dt$$

The above can be represented more compactly as an inner product:

$$W(a, b) = \langle x, \psi_{a,b} \rangle$$

The function $x(t)$ can be recovered from its transform by a reconstruction property. This can be called as inverse wavelet transform which helps to recover the original signal by integrating overall scales and locations, a and b . This can be mathematically expressed as follows:

$$x(t) = \frac{1}{C_g} \int_{-\infty}^{\infty} \int_0^{\infty} W(a, b) \psi_{a,b}(t) \frac{da db}{a^2}$$

where C_g is the admissibility constant.

The performance of wavelet transforms is in general better compared to STFT because the former uses varying window lengths. But the uncertainty principle affects the accuracy of the time and the frequency information of a wavelet transform which is also a limitation of STFT.

3.3.2 Discrete Wavelet Transform

In continuous wavelet transform, the original signal is recovered by using continuous integrals. But when certain conditions are met, it is possible to completely reconstruct the original signal using infinite summations of discrete wavelet coefficients. Discrete wavelet transform is the discretized version of the continuous wavelet transform. Discretization is done by using a logarithmic discretization of the scale a , and a is linked to the b with $\Delta b = b_0 a_0^m$. This kind of discretization of the wavelet has the form

$$\psi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}} \psi \left[\frac{t - nb_0 a_0^m}{a_0^m} \right]$$

where integer m controls the wavelet dilation and integer n controls the translation. The scaling and translation parameters a and b are functions of parameter m and n . The Discrete wavelet transform of a continuous signal $x(t)$ can be written as:

$$D_{m,n} = \int_{-\infty}^{\infty} x(t) a_0^{-m/2} \psi(a_0^{-m} t - nb_0) dt$$

where the values $D_{m,n}$ are known as wavelet coefficients or details coefficients. The common choices for discrete wavelet parameters a_0 and b_0 are 2 and 1 respectively. This power of two logarithmic scaling of both the dilation and translation steps results in a dyadic time-scale grid sampling. So substituting this in the above equation, the dyadic grid wavelet:

$$\psi_{m,n}(t) = \frac{1}{\sqrt{2^m}} \psi(2^{-m}t - n)$$

Using the dyadic grid wavelet of equation, DWT can be written as:

$$D_{m,n} = \int_{-\infty}^{\infty} x(t) \psi_{m,n}(t) dt$$

Discrete dyadic grid wavelets are usually chosen to be orthonormal and normalized to have unit energy. DWT are orthogonal which means that the information stored in a wavelet coefficient $D_{m,n}$ is not repeated elsewhere and allows for the complete regeneration of the original signal without redundancy. The important property of discrete wavelet transform which helps in practical usefulness is its Multi-Resolution Analysis (MRA) ^{28, 29, 30}.

3.3.3 Multi-Resolution Decomposition:

Multi-resolution analysis helps us to represent a function or signal on different scales. Successive approximation of the signal is performed to obtain smoother and smoother versions of the original signal. So the basic idea behind multi-resolution analysis is to represent a function as a limit of successive approximation. The scaling function associated used for smoothing of the signal can be given as follows:

$$\phi_{m,n}(t) = 2^{-m/2} \phi(2^{-m}t - n)$$

$$\int_{-\infty}^{\infty} \phi_{0,0}(t) dt = 1$$

where $\phi_{0,0}(t) = \phi(t)$ is called as father wavelet.

Using wavelets, the signals are decomposed into set of coefficients. The approximation coefficients are produced by convolving the signal and the scaling function. It can be expressed as follows:

$$S_{m,n} = \int_{-\infty}^{\infty} x(t) \phi_{m,n}(t) dt$$

The approximation coefficients are simply the weighted averages of the continuous signal $x(t)$ factored by $2^{m/2}$. A continuous approximation of the signal at scale m can be generated by summing a sequence of scaling function at this scale factored by the approximation coefficients as follows:

$$x_m(t) = \sum_{n=-\infty}^{\infty} S_{m,n} \phi_{m,n}(t)$$

where $x_m(t)$ is a smooth version of the signal $x(t)$ at scale index m . Summarizing the discussion a wavelet series representation of the signal $x(t)$ is given by

$$x(t) = \sum_{n=-\infty}^{\infty} S_{m_0,n} \phi_{m_0,n}(t) + \sum_{m=-\infty}^{m_0} \sum_{n=-\infty}^{\infty} D_{m,n} \psi_{m,n}(t)$$

$$S_{m,n} = \int_{-\infty}^{\infty} x(t) \phi_{m,n}(t) dt$$

$$D_{m,n} = \int_{-\infty}^{\infty} x(t) \psi_{m,n}(t) dt$$

Thus signal $x(t)$ is expressed as a combined series expansion consisting of approximation coefficients and the detail coefficients.

3.3.4 Fast Wavelet Transform

This section describes a fast computation of DWT called as Fast wavelet transform. The described integration of convolution of the signal with the wavelet function in previous section is avoided to compute FWT. The method is described in the following paragraphs.

Consider the following scaling equation (also called dilation equation) which describes the scaling function $\phi(t)$ in terms of contracted and shifted versions of itself.

$$\phi(t) = \sum_k C_k \phi(2t - k)$$

where $\phi(2t - k)$ is a contracted scaling function with step k and C_k is the scaling coefficient. The sum of the scaling coefficients and the sum of squares of the scaling

coefficients both must be equal to 2. The scaling coefficients must also have finite length. This condition satisfies what is called as compact support. Thus the wavelet functions with finite number of coefficients N_k can be written as

$$\psi(t) = \sum_k (-1)^k C_{N_k-1-k} \phi(2t-k)$$

The approximation and details coefficients of level $m+1$ can be represented using the scaling coefficients as follows:

$$S_{m+1,n} = \frac{1}{\sqrt{2}} \sum C_{k-2n} S_{m,k}$$

$$D_{m+1,n} = \frac{1}{\sqrt{2}} \sum b_{k-2n} S_{m,k} ; b_k = (-1)^k C_{N_k-1-k}$$

These two equations represent the multi-resolution decomposition algorithm. The first step to compute fast wavelet transform consists of computing these two coefficients. Then the approximation coefficients are combined with the scaling function to regenerate the approximation coefficients of signal at the next level and details coefficients are used to regenerate the detail signal at next level. The vector $(1/\sqrt{2})c_k$ is the low pass filter and $(1/\sqrt{2})b_k$ is the high pass filter. The low and high frequency information of the signal is captured by a low pass filter and a high pass filter respectively. The end product of multi-resolution decomposition differs depending upon the number of levels. A 4-level

decomposition of a signal results in one signal with low frequency components and 4 detail signals with high and intermediate frequencies. The second step is the reconstruction algorithm which can be represented as follows:

$$S_{m-1,n} = \frac{1}{\sqrt{2}} \sum_k C_{n-2k} S_{m,k} + \frac{1}{\sqrt{2}} \sum_k b_{n-2k} D_{m,k}$$

Generally in signal processing, the approximation coefficients at resolution m , $S_{m,n}$, are convolved with the low pass filter by moving the filter along the signal one step at a time. The approximation coefficients are then subsampled. Subsampling is done by choosing every second value to give the approximation coefficient vector at scale $m+1$. This reduces the length of the signal by 2. Similarly the approximation coefficients at resolution m , $S_{m,n}$, are convolved with high pass filter and subsampled to give detail signal coefficients at scale $m+1$. In the next level, the details components are kept and approximation coefficients $S_{m+1,n}$ are again passed through the low pass and high pass filters to obtain $D_{m+2,n}$ and $S_{m+2,n}$. The process is repeated over all scales to obtain the full decomposition of the signal. This process is called the filtering process. This wavelet transform is very efficient from the computational point of view but it is not translation invariant. Subsampling in filtering process can be skipped to obtain a translation invariant version of discrete wavelet transform. Such a type of wavelet transform algorithm is used in this thesis in order to get more complete characteristic of the analyzed signal. The algorithm is described in next section.

3.3.5 Translation Invariant Wavelet Transform

This transform is known by a variety of names including the redundant, stationary, maximal overlap or non-decimated wavelet transform. The normal filtering process is modified by avoiding the subsampling part. This results in a translation invariant decomposition which means that the number of wavelet coefficients generated at each step is equal to the number of signal components. The general idea behind it is that the decimation of signal is not carried out at each step. Since the signal is not decimated, more precise information for the frequency localization is obtained. A number of algorithms are available to compute such a transform. The present work uses the *à trous* algorithm.

3.3.6 The *à trous* Algorithm

The discrete wavelet transform can be implemented without decimation by *à trous* algorithm. The term *à trous* – (means “with holes” in French) was coined by Holschneider *et al.* in reference to the fact that only one in every 2^{j-1} coefficients is non-zero in the filter impulse responses at the j^{th} octave³¹. The details of this algorithm are given in Dutilleux³², Aussem³³, Percival³⁴. A brief description is presented here.

Let $c_j(n)$ be a 1-D signal and let $h(n)$ be the low pass filter and $\varphi(x)$ be the scaling function. Then the undecimated successive approximation of $c(n)$ is given by

$$c_j(n) = \sum_k h(k) f_{j-1}(n - k2^{j-1}); j = 1, \dots, J$$

where $c_o(n) = c(n)$ and j and J are the scale index and the number of scales, respectively. The wavelet coefficients are extracted by using the high-pass filter, $g(n)$; the associated with the wavelet function, $\psi(x)$ is as follows

$$w_j(n) = \sum_k g(k) f_{j-1}(n - k2^{j-1}); j = 1, \dots, J$$

Two dual filters, $\bar{h}(n)$ and $\bar{g}(n)$, are needed for exact reconstruction which must satisfy the Quadrature Mirror Filter (QMF) condition:

$$\bar{h}(n) * h(n) + \bar{g}(n) * g(n) = \delta(n)$$

where $h(n)$ is the unit impulse sequence and “*” denotes the discrete convolution. A simple choice for these dual filters is $\bar{h}(n) = \bar{g}(n) = \delta(n)$. Substituting in the above equation one gets,

$$g(n) = \delta(n) - h(n)$$

So from the above equations, one can obtain wavelet coefficients as

$$w_j(n) = c_{j-1}(n) - c_j(n); j = 1, \dots, J$$

The reconstruction of the original signal can be easily done by

$$c(n) = c_j(n) + \sum_{j=1}^J w_j(n)$$

The low pass filter used here is B_3 spline defined as $\left(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right)$. This is of compact support and is point-symmetric. Since subsampling of signal is not carried out, it is replaced by upsampling the low pass filter. The upsampling of filter is done by inserting zeros between each of the filter's coefficients at each level. It can be represented as follows:

$$h_0, \underbrace{0, \dots, 0}_{2^{j-1}-1 \text{ zeros}}, h_1, \underbrace{0, \dots, 0}_{2^{j-1}-1 \text{ zeros}}, \dots, h_{L-2}, \underbrace{0, \dots, 0}_{2^{j-1}-1 \text{ zeros}}, h_{L-1}$$

where L is the length of the low pass filter and j is number of resolution.

The steps to program the algorithm can be given as follows:

Index n ranges over all data values.

Step 1: Initialize j to 0 and store input signal in $c_j(n)$.

Step 2: Increment i . Carry out discrete convolution of $c_{j-1}(n)$ using the low pass filter h .

The distance between a central pixel and adjacent ones is 2^{j-1} .

Step 3: Obtain wavelet coefficient at level j , $w_j(n) = c_{j-1}(n) - c_j(n)$.

Step 4: If j is less than the number of resolutions, goto step 2.

Step 5: The final set $W = \{w_0, w_1, \dots, w_J, c_J\}$ represents the wavelet transform of the data.

The computational complexity of the above algorithm is $O(n)$ for n -valued input and storage capacity is $O(n^2)$. The advantages of the *à trous* algorithm are given as follows:

- Details of MRA using *à trous* are more consistent across time when compared with the discrete wavelet transform of the same signal.
- Reconstruction is possible and is trivial
- Invariance under translation is completely verified
- The transform is known at each pixel, allowing detecting without any error, and without interpolation
- In two dimensions, the transform is practically isotropic
- Evolution of the transform can be followed from one scale to the next

Chapter 4

Method to Detect Recurring Patterns in Process Signal

4.1 Introduction

The chapter will focus on the proposed method which identifies all repetitive patterns of any structure of any given length. Earlier methods reported in the literature try to detect only a few pre-defined classes of odd patterns by analyzing mostly global and qualitative information. This method follows a significantly different approach to identify odd patterns in process signals. It relies on a time-frequency analysis – an idea which has given promising results for similarity queries in time series databases^{35,36}.

4.2 Preprocessing Stage

Preprocessing or preliminary process is performed on raw data to enhance and to transform raw data into informative input data. Here HAAR wavelet decomposition is employed to preprocess the data and to compute local frequency information. The need in such information and the method to compute the local information is described below.

The importance of local frequency information can be understood from the following two situations. Consider two uniformly sampled identical signals which have completely different amplitudes at any time instance of sampling. If one compares these two signals based on amplitude values, then it is classified as different signals. Similarly if two uniformly sampled signals with different forms but which are approximately equal in amplitude are compared based on amplitude values alone, then it can be classified as equal. To overcome such situations, the local frequency information (computed using

Haar wavelet decomposition) is used along with amplitude values to compare two signal segments. This kind of time-frequency decomposition gives coefficients that have more obvious physical interpretation than those occurring with Discrete Fourier Transforms.

4.2.1 Computation of Frequency Index

Let us consider s_0 to be a sequence taken from a uniformly sampled process signal. Consider the Haar wavelet decomposition of s_0 . It follows that the energies of the sequences d_m in the highest frequency regions, say for m from 1 to 3, are the indicators of the frequency of the discrete signal s_0 . According to this argument, for every signal segment $s_0 \neq 0$ an index is defined as

$$\omega(s_0) = \frac{\sum_{m=1}^3 \|d_m\|^2}{\|s_0\|^2}$$

which reflects the frequency of this discrete signal. In the above definition, only three frequency levels ($d_m, m=1,2,3$) are taken into consideration because it gives empirically the best results. One can take fewer or more levels into account if the specific application demands.

The value of $\omega(s_0)$ is greater for a high frequency sequence s_0 than for a low frequency sequence s_0 .

4.2.2 Process Signal with Local Frequency Indices

Let $(y_1, y_2, \dots, y_n, \dots, y_N)$ be a uniformly sampled process signal. The time instant at which the process signal is sampled is denoted by subscript n . The process signal is extended to a new sequence along with the local frequency indices. The computation of frequency index $\omega(\cdot)$ was explained in the previous section. Thus a new predicative sequence r is obtained by inserting next to each and every y_n a corresponding local frequency index ω_n designed as $\omega_n := \omega(b_n)$, where b_n is the sequence of length $R=8$ around the sample y_n .

$$r = (y_1, \omega_1, y_2, \omega_2, \dots, y_N, \omega_N)$$

$$b_n = (y_{n-(R/2-1)}, y_{n-(R/2-2)}, \dots, y_{n-1}, y_n, y_{n+1}, \dots, y_{n+(R/2-1)}, y_{n+R/2}).$$

It includes 3 ($=R/2-1$) samples to the left and 4 ($=R/2$) samples to the right of y_n . Undefined elements in b_n are substituted with zeros. So any index ω_n can be computed from the Haar wavelet transform from the sequence b_n which is built from sampling values in the neighborhood of y_n .

Thus the sampled process signal is preprocessed using Haar wavelet transform. Raw data is now transformed into useful informative data with frequency indices. This can be used as the input pattern for pattern detection procedure. The input pattern is scaled at different levels using multi-resolution decomposition to detect hidden patterns at various levels.

4.3 Recurring Pattern Detection

The preprocessed process signal contains the information necessary for detecting repetitive signal segments. This sampled signal from manufacturing process is a time series data. There exist many methods which efficiently locate the target pattern of interest in time series^{9, 10, 11, 12, 13}. The present work deals with unknown recurring patterns in a process signal. This makes the problem more complicated and practical. Such unknown patterns are also termed as “motifs” because of their close analogy to their discrete counterparts in computation biology¹⁹. Finding these motifs was based on an algorithm for pattern discovery in DNA sequences.

The present pattern search method can be split into following steps:

1. Defining a primary set of patterns in the process signal
2. Defining Boxes in Euclidean space
3. Determining the minimal length of the subsignal
4. Neighborhood search in the Euclidean Space $IR^{2\ell}$

4.3.1 Defining a Primary Set of Patterns in the Process Signal

From the constructed enriched process signal $r = (y_1, \omega_1, y_2, \omega_2, \dots, y_N, \omega_N)$, all its subsignals of a certain length $2L$ can be represented by the moving windows of the form

$$v_i = (y_i, \omega_i, y_{i+1}, \omega_{i+1}, \dots, y_{i+L-1}, \omega_{i+L-1})$$

Therefore, by detecting similar vectors of length $L = l$ in the set

$$V: = \{v_i: i=1, 2, \dots, N-l+1\}$$

through a proper pattern recognition procedure, it is possible to find all repetitive patterns of arbitrary length L with $L \geq \ell$. Here ℓ is the length of the shortest pattern expected to repeat itself in the original process signal. If some observations in the process signal are missing, this method can be still applied by simply eliminating signal segments v_i with missing observations from the whole set V .

4.3.2 Defining Boxes in a Euclidean Space

Certain intervals of the form $[y_{min}, y_{max})$ and $[\omega_{min}, \omega_{max})$ are chosen as narrow as possible to contain each and every signal measurement y_n ($n = 1, 2, \dots, N$) and index ω_n ($n = 1, 2, \dots, N$), respectively. The ranges $[y_{min}, y_{max})$ and $[\omega_{min}, \omega_{max})$ is divided into P and Q subintervals of equal length, respectively, where P and Q are integers greater than one.

Similarity criterion for the vector in the set V defined with tolerances ε :

Let $v_i=(y_i, \omega_i, y_{i+1}, \omega_{i+1}, \dots, y_{i+L-1}, \omega_{i+L-1})$ and $v_j=(y_j, \omega_j, y_{j+1}, \omega_{j+1}, \dots, y_{j+L-1}, \omega_{j+L-1})$ be two vectors of equal length $2L$ extracted from the sequence r . The vectors v_i and v_j are called similar if

1. $|i-j| \geq L$. This condition checks whether the vectors are “nonoverlapping” and
2. $|y_{i+k} - y_{j+k}| < \varepsilon_1$ as well as $|\omega_{i+k} - \omega_{j+k}| < \varepsilon_2$ for $k = 0, 1, \dots, L-1$

where

$$\varepsilon_1 = \frac{y_{\max} - y_{\min}}{P}$$

$$\varepsilon_2 = \frac{\omega_{\max} - \omega_{\min}}{Q}$$

This similarity criterion helps to recognize identical subsignals even if they are contaminated by noise and/or shifted in time. The integers P and Q determine the sensitivity of the algorithm to find identical patterns. It also determines the minimal subsignal length which is explained later in this chapter. The larger the values of P and Q the less tolerant are the similarity criterion to the presence of noise as well as shift in time. By varying the values of P and Q , the similarity criterion can be focused on frequency domain or amplitude values. If P is set to a small integer, then it is strongly dependent on comparisons in the frequency domain and weakly dependent on amplitude values of the process signal. If P is set to 1, then the similarity criterion is solely determined by frequency domain information. Consider a situation where one is monitoring the vibration of a piece of process equipment and the amplitude of the vibration does not vary much, but the variations in the frequency of the vibration affect the performance of the process. Here choosing a small P is meaningful.

An exhaustive search for extracting all pairs of similar vectors in the sequence r involves examining sequentially every pair of subsequences with a certain minimal length 2ℓ for a match in compliance with the above similarity criterion. A guideline to determine the minimal length ℓ is explained later in the chapter. The objective is to design an algorithm

which is considerably more efficient. For the design of such an algorithm, some useful concepts from the theory of fractals are used.

According to similarity criterion, a new sequence \tilde{r} corresponding to a sequence $r=(y_1, \omega_1, y_2, \omega_2, \dots, y_N, \omega_N)$ as

$$\tilde{r} = (a_1, w_1, a_2, w_2, \dots, a_N, w_N)$$

$$a_n := \left[\frac{y_n - y_{\min}}{\varepsilon_1} \right]$$

$$w_n := \left[\frac{\omega_n - \omega_{\min}}{\varepsilon_2} \right]$$

for $n=1, 2, \dots, N$.

Here, $[.]$ denoted the integer part of a real number x . The elements a_n and w_n of the sequence \tilde{r} satisfy the inequalities

$$0 \leq a_n \leq P - 1 \text{ and } 0 \leq w_n \leq Q - 1$$

Every vector $v_i = (y_i, \omega_i, y_{i+1}, \omega_{i+1}, \dots, y_{i+L-1}, \omega_{i+L-1})$ can be encoded into a vector of nonnegative integers defined by

$$\tilde{v}_i = (a_i, w_i, a_{i+1}, w_{i+1}, \dots, a_{i+L-1}, w_{i+L-1})$$

Each vector \tilde{v}_i identifies a box in the Euclidean space $IR^{2\ell}$. A box represented by one vector is considered to be in the neighborhood of the box represented by another vector if it satisfies the condition given below.

Let $\tilde{v}_i = (a_i, w_i, a_{i+1}, w_{i+1}, \dots, a_{i+L-1}, w_{i+L-1})$ and $\tilde{v}_j = (a_j, w_j, a_{j+1}, w_{j+1}, \dots, a_{j+L-1}, w_{j+L-1})$ be two vectors identifying separate boxes in the Euclidean space. They are said to be neighborhood if

$$|a_{i+k} - a_{j+k}| < 1 \quad \text{and} \quad |w_{i+k} - w_{j+k}| < 1$$

for $k = 0, 1, \dots, \ell - 1$.

After defining boxes in the Euclidean space one can frame a similarity criterion as follows.

Two vectors $v_i, v_j \in V$ are similar if their corresponding boxes indexed by the vectors $\tilde{v}_i = (a_i, w_i, a_{i+1}, w_{i+1}, \dots, a_{i+L-1}, w_{i+L-1})$ and $\tilde{v}_j = (a_j, w_j, a_{j+1}, w_{j+1}, \dots, a_{j+L-1}, w_{j+L-1})$ are either equal or in the neighborhood of each other in the sense of above condition.

If the pattern length ℓ for raw patterns is fixed, the search for pairs of similar vectors in the set V of enriched patterns can begin with identifying vectors in identical or neighboring boxes of the Euclidean space $IR^{2\ell}$.

Useful solutions for a similar problem have been proposed by Theiler⁴¹ and Grassberger⁴² in the context of measuring the so-called correlation dimension of a subset of a Euclidean space. However, due to short comings of these approaches concerning computation time

and storage requirements for higher dimensional spaces, a new method is presented in this thesis.

4.3.3 Determining the Minimal Length of the Subsignal

The dimension 2ℓ of the Euclidean space $\mathbb{R}^{2\ell}$ necessary for the construction of the set V can be fixed from the equation below. This drastically reduces the chance of classifying pairs of random patterns of a sequence r as similar.

Suppose that a sequence $r = (y_1, \omega_1, y_2, \omega_2, \dots, y_N, \omega_N)$ consists of samples from a uniform distribution with values $y_n \in [y_{min}, y_{max})$ and $\omega_n \in [\omega_{min}, \omega_{max})$ for $n = 1, 2, \dots, N$.

If ℓ is chosen such that

$$\ell \geq \frac{2 \log_3 N + 3}{\log_3 PQ - 1}$$

then the probability $P_{2\ell}$ that the set V contains no similar vectors is greater than 98%. The proof for the above theorem is not explained here in detail. The assumptions of this theorem will be only coarsely satisfied by a process signal acquired from a practical application because of the following reasons:

- (a) The random patterns need not follow a uniform distribution
- (b) If the amplitude values of a process signal are uniformly distributed, then the corresponding frequency indices need not be also uniformly distributed.

However, it is hoped that the equation provides a useful guideline for determining the minimal length for practical applications.

4.3.4 Neighborhood Search in the Euclidean Space $\mathbb{R}^{2\ell}$

The search for all possible pairs on similar patterns can be done by following steps:

Step 1: In this step all vectors \tilde{v}_i are constructed. For each vector v_i of V the corresponding vector \tilde{v}_i of box indices is constructed directly from the sequence \check{r} .

Step 2: The vectors constructed in step 1 are stored in a matrix M . It is stored in such a way that the rows of matrix M are filled up with these vectors. Each row represents one vector and can be denoted as $\tilde{v}_{i,k}$ where $i = 1, 2, \dots, N-\ell-1$ and $k = 1, 2, \dots, 2\ell$ are the corresponding row and column of vector \tilde{v}_i in the matrix M respectively. In addition, the row numbers of the matrix M are stored by creating one more column at end. Thus the final matrix M consists of $N-\ell-1$ rows and $2\ell+1$ columns.

Step 3: The resultant matrix from step 2 is rearranged in step 3. The rows of the matrix M are sorted according to the ascending order of the elements in the first column of M .

Step 4: The rearranged matrix is now used to determine all the pairs that have equal or neighboring boxes. Let $g(p)$ denote the index of the vector stored in the p th row. This means that the p th row of the matrix M stores the vector $\tilde{v}_{g(p)}$. The indices $g(p)$ are stored in the last column of M , i.e. $g(p) = M(p, 2\ell+1)$. For each position p of the first column of M all succeeding positions $p+1, p+2, \dots, p+K$ are determined using following condition:

$$|M(p+k, 1) - M(p, 1)| \leq 1$$

for $k = 1, 2, \dots, K$. This set is defined as $I_1 := \{p\} \cup \{p+1, p+2, \dots, p+K\}$ of rows of M , which refers to the vectors $\tilde{v}_{g(p)}, \tilde{v}_{g(p+1)}, \dots, \tilde{v}_{g(p+K)}$. A row of the form $p+k$ ($k = 0, 1, \dots, K$) of the matrix M contains the components of the vector $\tilde{v}_{g(p+k)}$. With respect to the index $g(p)$, each index of the form $g(p+k)$ computed through I_1 with $k = 1, 2, \dots, K$ corresponds to a certain pair of vectors $(\tilde{v}_{g(p)}, \tilde{v}_{g(p+k)})$.

With regard to the second column of the matrix M , the set I_1 can be reduced to

$$I_2 := \{p\} \cup \{p+k : |M(p+k, 2) - M(p, 2)| \leq 1, k = 1, 2, \dots, K\}$$

by directly checking the condition

$$|M(p+k, 2) - M(p, 2)| \leq 1$$

for all row numbers of I_1 . Similarly a further reduction of this set I_2 will result in sequence of sets $I_3, I_4, \dots, I_{2\ell}$. The final set $I_{2\ell}$ of rows corresponds to a set of vector pairs of the form $(\tilde{v}_{g(p)}, \tilde{v}_{g(p+k)})$, which represent equal or neighboring boxes. If during a particular reduction step there remains no vector $\tilde{v}_{g(p+k)}$ that could be in the neighborhood of $\tilde{v}_{g(p)}$, proceed with the next row $p+1$ of M . Since the procedure in this step is performed for each row p of the matrix M , every pair (v_i, v_j) of the vectors that represents

equal or neighboring boxes in $\mathbb{R}^{2\ell}$ is taken into consideration. Consequently, the set W_1 of vector pairs is a superset of all pairs of similar vectors v_i and v_j of V .

$$W_1 := \{(v_i, v_j) : (\tilde{v}_i, \tilde{v}_j) \text{ is a result of step 4, } |i-j| \geq \ell\}$$

Step 6: The set W_1 is reduced to a set W_2 which contains just all pairs of similar vectors of V that satisfy the similarity criteria from section 4.3.2. In addition, every pair (v_i, v_j) of W_2 is ordered such that $j \geq i+\ell$. This set W_2 can be stated as follows

$$W_2 := \{(v_i, v_j) \in W_1 : |y_{i+k} - y_{j+k}| < \varepsilon_1, |\omega_{i+k} - \omega_{j+k}| < \varepsilon_2 \text{ for } 0 \leq k \leq \ell-1; j \geq i+\ell\}$$

Step 7: The sequences v_i and v_j in each pair $(v_i, v_j) \in W_2$ are extended in length to maximal extent. This extension is done both backward and forward as long as the sequences maintain their similarity property defined in previous sessions.

4.3.5 Determining the Value of P and Q

Let y_i and y_j be two arbitrary elements taken from the process signal which are assumed to be generated by adding noise to a true amplitude value y . This can be denoted as follows:

$$y_i = y + \tilde{\eta}_1$$

$$y_j = y + \tilde{\eta}_2$$

where $\tilde{\eta}_1$ and $\tilde{\eta}_2$ are the noise values.

For the absolute value of the difference between two arbitrary noise values $\tilde{\eta}_3$ and $\tilde{\eta}_4$, 80% quantile $Q_{0.8}$ can be represented by

$$|\tilde{\eta}_3 - \tilde{\eta}_4| < Q_{0.8}$$

This also means that

$$|y_i - y_j| = |\tilde{\eta}_1 - \tilde{\eta}_2| < Q_{0.8}$$

So value of P can be selected by the following relation from section 4.3.2,

$$\varepsilon_1 = \frac{y_{\max} - y_{\min}}{P} \approx Q_{0.8}$$

Similarly Q can be selected using the following relation

$$\frac{\omega_{\max} - \omega_{\min}}{Q} = 2\varepsilon(2 + \varepsilon)$$

The choice of ε can be made by the following equation where ω_i and ω_j correspond to related positions in pairs of similar subsequences

$$|\omega_i - \omega_j| \leq 2\varepsilon(2 + \varepsilon)$$

In order that the presented method can detect any repetitions, at least one of the values P or Q needs to be set to a value greater than or equal to two. In order for P to have a value of two or higher, it follows from P equation that the spread of the amplitude values needs to be a multiple of the average disturbance of the amplitude values. The disturbance characteristics that permit us to set $Q \geq 2$ are more restrictive. The average disturbance in a process signal should not exceed around 12% of the amplitude values because for Q equal to 2 in Q equation, ε is 0.12.

Chapter 5

Experimentation

5.1 Introduction

This chapter presents the implementation of the recursive pattern search algorithm described in the previous chapter. The pattern search method and the *à trous* algorithm for multi-resolution decomposition were coded using MATLAB. Source codes are enclosed in the appendix.

5.2 Experimental Results

Datasets from manufacturing process was not readily available; hence the algorithm is implemented on a mathematical simulated signal. A process signal of length $N=256$ is assumed to be collected at the rate of 1Hz is generated using mathematical expressions. The sampled process signal is assumed to follow an approximate uniform distribution when the process is under “in-control” state. Two different pairs of repetitive patterns are inserted in the test signal. Noise is added to test the accuracy of method in presence of noise. It is generated using normal distribution with mean 0 and standard deviation equal to 5% of the amplitude which is then added to every sampling point of these repeating subsignals. The original test signal, repetitive patterns with and without noise, and signal with repetitive patterns are shown in figure 5.1, 5.2, 5.3 and 5.4 respectively.

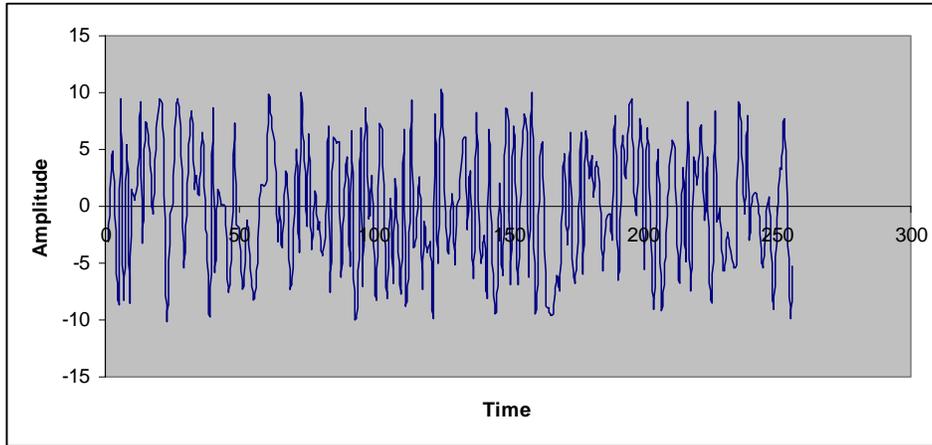


Figure 5.1: Original process signal of length 256 and collected at a sampling rate of 1Hz

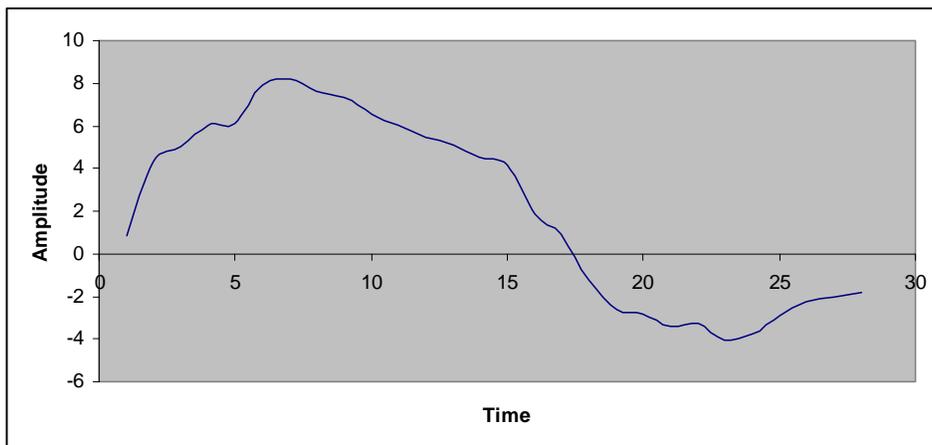
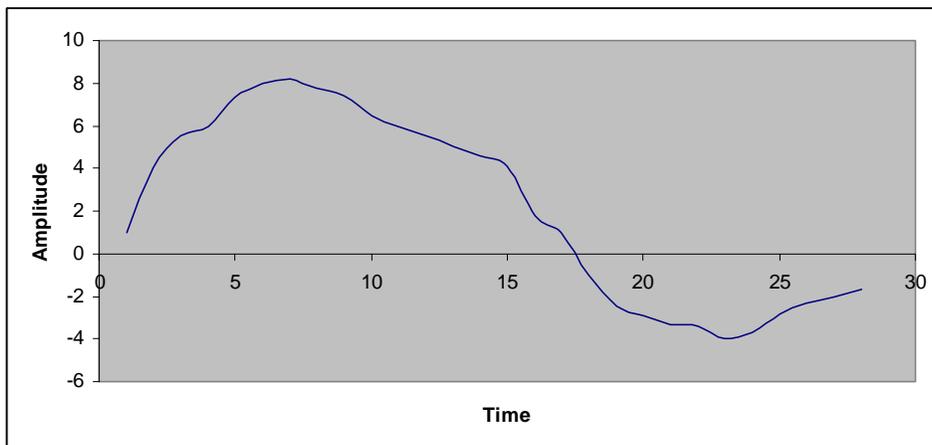


Figure 5.2: First pair of systematic pattern contaminated with additive white noise

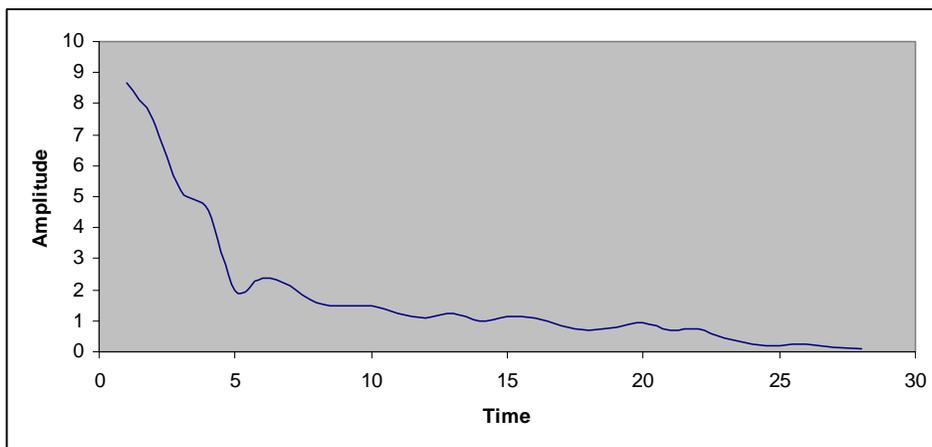
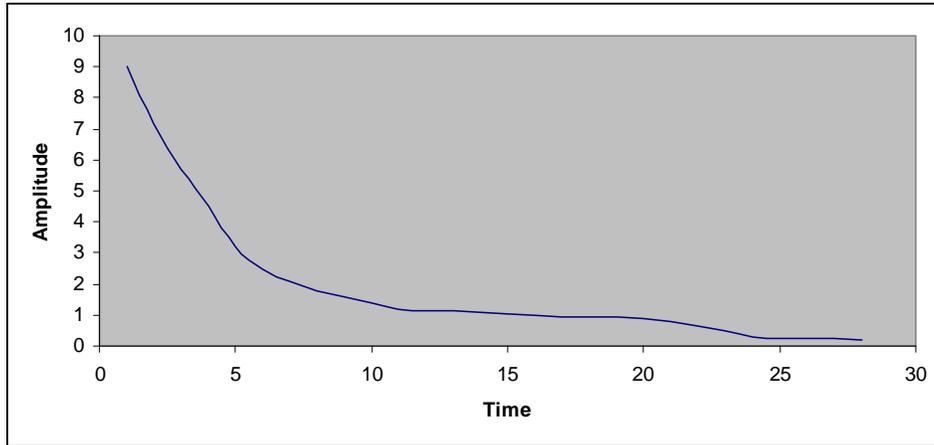


Figure 5.3: Second pair of systematic pattern contaminated with additive white noise

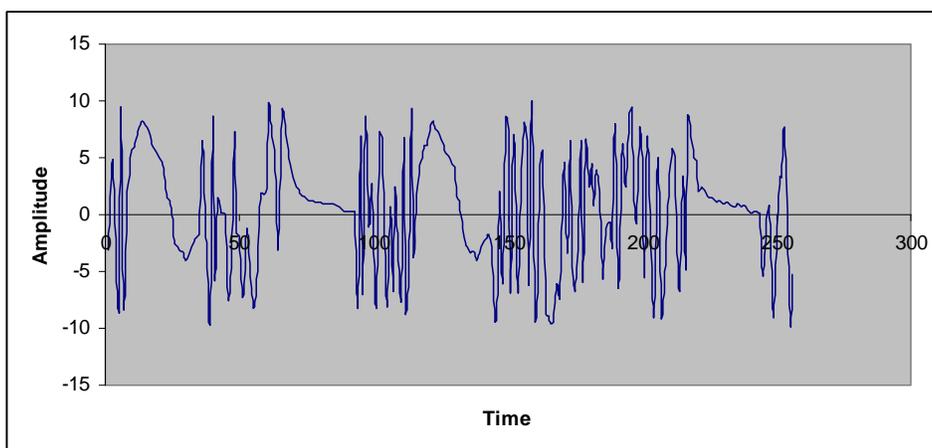


Figure 5.4: Test process signal of length 256 with inserted systematic patterns

The process measurements in between the test subsignals are assumed to be uniformly distributed between the range $[-10, 10]$. The frequency indices computed via Haar wavelet transform are inserted next to the signal samples. The statistics of this enriched signal is given in the table below. These parameters are used to determine the tolerances described in section 4.3.2.

Table 5.1: Distribution of parameters in enriched process signal

Parameter	Value
y_{max}	9.96
y_{min}	-9.84
ω_{max}	1
ω_{min}	0.0064

The values P and Q are determined as follows:

The absolute values of the amplitudes of the systematic patterns are less than 10, as can be seen from the Figure 5.2 and Figure 5.3. So the standard deviation turns out to be 0.5 as per the assumption. Thus noise values of the process signal have the normal distribution $\mathcal{N}(0, 0.5)$. Under this assumption the difference $|\tilde{\eta}_3 - \tilde{\eta}_4|$ of two arbitrary noise values in the process signal have normal distribution $\mathcal{N}(0, \sqrt{0.5})$. Instead of 80% prescribed in the equation, the present work takes 90% to have extra 10% as a factor of safety in determining the repetitive patterns. From the table of the normal distribution

function, the 90% quantile of $\mathcal{N}(0, 1)$ is 1.281. Consequently, the quantile for the distribution of $|\tilde{\eta}_3 - \tilde{\eta}_4|$ turns out to be 0.906 as given below.

$$Q_{0.9} = 1.281 * \sqrt{0.5} \approx 0.905$$

Using this quantile value in equation for the parameter P, one gets

$$\frac{9.96 - (-9.84)}{P} = 0.906$$

which results in $P \approx 21$. From the normal distribution function table, 72.5% quantile of $\mathcal{N}(0, 1)$ is 0.598. With this value, ε turns out to be 0.042. Substituting this in equation for the parameter Q, one gets

$$\frac{1 - 0.0064}{Q} = 2 * 0.042(2 + 0.042)$$

which results in $Q \approx 5$.

Substituting P and Q values, minimal subsignal length ℓ turns out to be 5 according to the equation derived in section 4.3.3. The proposed method coded in MATLAB version 5.1 is run on a Pentium M machine. The results are shown in Table 5.2. Similar subsignals inserted in the process signal are detected and no additional random signals are detected

as recurring patterns. There might be missing parts at the beginning of the computed subsequences which is due to the time window used for computing the frequency index described in section 4.2.1.

Table 5.2: Comparison of the constructed and the detected pairs of systematic subsequences identified by their first and last indices

Pairs of systematic test pattern	Detected systematic pairs
(8 – 35); (116 – 143)	(9 – 27); (117– 135)
(66 – 93); (217 – 244)	(64 – 89); (215 – 240)

To demonstrate the importance of the added local frequency indices, the method is implemented with the same P value but Q is set to 1. This means that the computed local frequency index for every amplitude value is ignored and pattern search is done only with the amplitude values. In addition to the systematic pairs the results showed plenty of pairs that are not similar in shape but had nearly equal in amplitude values. Even small oscillation around the same amplitude is detected as a recurring pattern. This proved the usefulness of computed frequency index information.

The second part of the algorithm is to identify the recurring patterns at different resolution levels. This is performed by decomposing the test process signal using *à trous* algorithm.. The process signal is decomposed into 5 levels; to demonstrate the identification of patterns at different frequency levels. Thus 5 levels of wavelet

coefficients are calculated, with the signal getting coarser at each increasing level. The optimal value of J in section 3.2.6 was set to 5.

The values of P and Q are calculated for each level using the same procedure explained before. Table 5.3 shows the values of P and Q used for each level.

Table 5.3: Calculated values of P and Q for each decomposed level of process signal along with the ranges of coefficients at each level

Level	y_{max}, y_{min}	$\omega_{max}, \omega_{min}$	P	Q
1	10.25, -8.0749	1, 0.4758	20	5
2	3.6576, -3.9703	1, 0.0213	14	3
3	2.0888, -2.8186	1, 0.0138	10	2
4	2.8406, -2.0838	1, 0.00047	9	2
5	1.95, -1.4791	0.9999, 0	9	2

Figure 5.5 shows the multi-resolution analysis of the sampled process signal with its five wavelet coefficients. The low and high frequency component gets separated and thus provides time-frequency analysis.

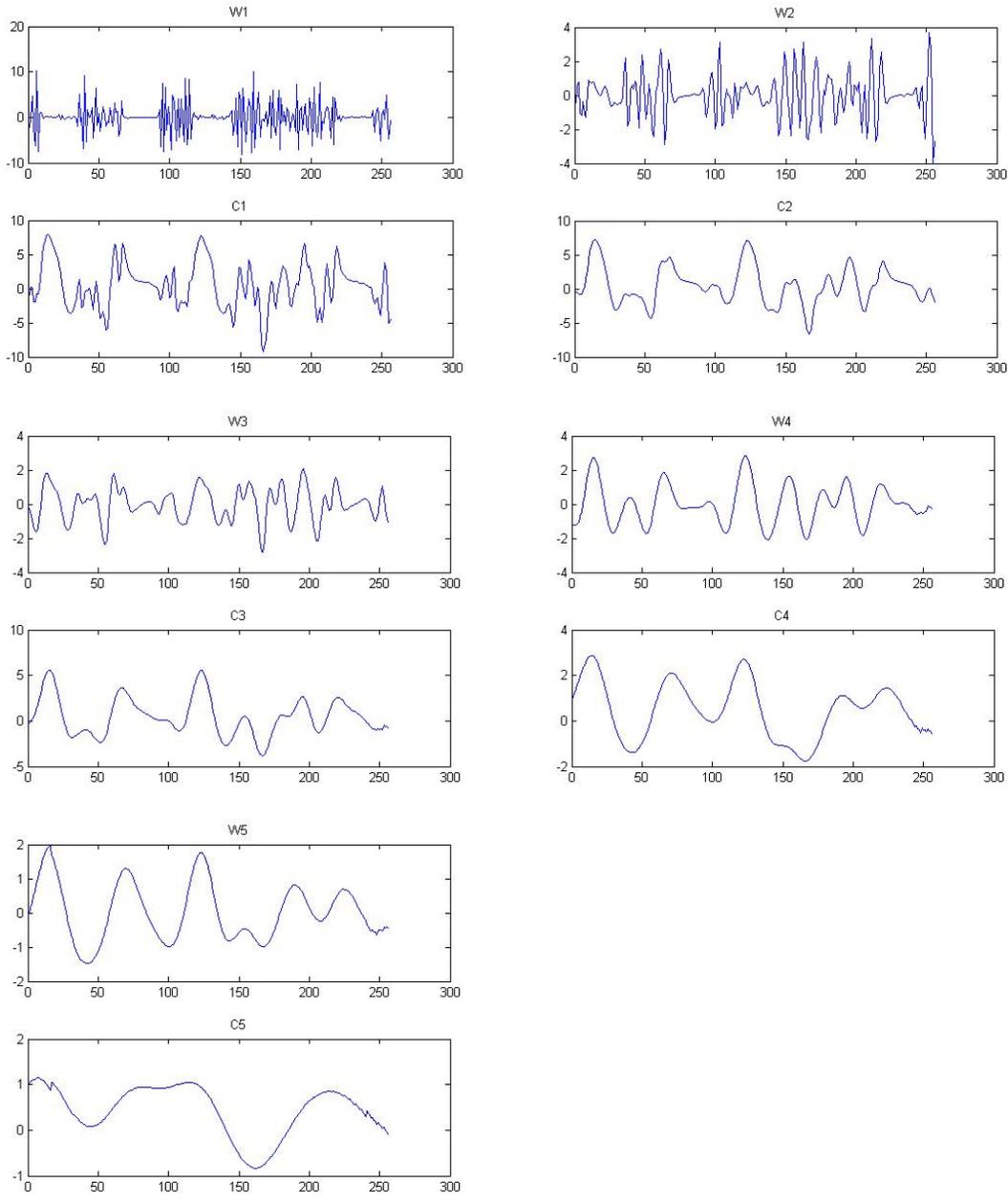


Figure 5.5: Multi-resolution analysis of sampled process signal. W represents detail coefficients and C represents approximation coefficients

Each decomposed level is given as input to the first algorithm and recurring patterns are detected at each level. As the signal resolution goes down, the signal gets smoother. So at

lower scales not many recurring patterns can be found. Temporal patterns at different scales are identified and the detected pairs are shown in Table 5.4.

Table 5.4: Detected pairs of systematic subsequences at different resolution levels identified by their first and last indices

Resolution Level	Pairs of systematic test pattern in initial process signal	Detected systematic pairs
1	(8 – 35); (116 – 143) (66 – 93); (217 – 244)	(5-27); (116-133) (67-88); (220-240)
2		(64-89); (215-240)
3		(14-52); (123-165) (80-113); (220-251)
4		(36-97); (177-235)
5		(17-93); (121-201)

As the the resolution decreases, the length of time-window is increased since the signal gets smoother at higher scales. The minimal subsignal length ℓ is taken in steps of 10 (i.e. 10, 20, 30, ..., N; N is the time-window at the last level).

5.3 Discussion

The results show that the proposed method identifies recurring patterns even in the presence of noise. Using multi-resolution decomposition, more temporal patterns at

different frequency levels are identified. Computation time reduces at lower resolution levels; it means extensive data sets can be handled. There are some redundancies in the output because of the length of time-window chosen to find the frequency index in section 4.1.1 and also the minimal subsignal length ℓ .

Chapter 6

Conclusion

6.1 Summary of Observations

In this thesis, a method for identification of recurring patterns necessary for constructing successful process monitoring techniques has been developed. Using Multi-resolution decomposition, temporal patterns at different frequencies are identified to provide better view of the repetitive patterns in the process signal.

The proposed system can be considered general-purpose in the sense of being able to recognize arbitrary patterns in a process signal along with the time instants of their occurrence without prior knowledge about the process. The previous methods only recognized reference patterns. This greatly assists quality control personnel in tracking down the cause of process irregularities, because the process conditions at the time instants where the repeated pattern occur can be analyzed. The addition of local frequency index proved to provide useful information in improving the performance and accuracy of amplitude-based search methods.

The simulation study with a sampled signal demonstrated the recognition ability of the method. The algorithm used to decompose the signal provided shift invariant versions of signal at different resolutions. The multi-resolution approach allowed identifying broader family of recurring patterns together with time instants of their occurrence. Thus adoption of this approach enhances diagnostic capabilities of process control system and improves the quality of manufactured products.

6.2 Future Work

Future work will focus on using an advanced and a robust distance measure than the Euclidean measure. An alternate distance measure is proposed by J. R Chen⁴³. A more sophisticated method can be developed to determine the value of the parameters P and Q and the minimal subsignal length l . The application of this method can be extended to two-dimensional data as well.. Another possible extension is devising a way to integrate the patterns found at different level to provide integrated pattern detection.

Bibliography

- ¹ C. Chatfield, “*The Analysis of Time Series*”, Chapman and Hall, 6th ed London, 2003
- ² A. Gyaourova, C. Kamath and I.K. Fodor, “*Undecimated wavelet transforms for image de-noising*”, 2002
- ³ F. Höppner, “*Time series abstraction methods - A Survey*”, Workshop on Knowledge Discovery in Databases, Dortmund, pp. 777-786, Sept/Oct, 2002
- ⁴ I. Savnik, G. Lausen, H. P. Kahle, H. Spiecker and S. Hein, “*Algorithm for matching sets of time series*”, International Conference on Principles of Data Mining and Knowledge Discovery, pp. 277–288, 2000
- ⁵ G. Das, K. I. Lin, H. Mannila, G. Renganathan and P. Smyth, “*Rule discovery from time series*”, Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pp. 16–22, AAAI Press, 1998
- ⁶ D. J. Berndt and J. Clifford, “*Finding patterns in time series: A dynamic programming approach*”, pp 229–248, MIT Press, 1996
- ⁷ D. Sankoff and J. B. Kruskal, “*Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*”, Addison Wesley, 1998

⁸ C. S. Daw, C. E. A. Finney and E.R. Tracy, "*Symbolic analysis of experimental data*", Review of Scientific Instruments, 2001

⁹ R. Agrawal, K. I. Lin, H.S. Sawhney, and K. Shim, "*Fast similarity search in the presence of noise, scaling, and translation in time-series databases*", In proceedings of the 21st International Conference on Very Large Databases, Zurich, Switzerland, pp 490-501, Sept, 1995

¹⁰ K. Chan and A. W. Fu, "*Efficient time series matching by wavelets*", In proceedings of the 15th IEEE International Conference on Data Engineering, Sydney, Australia, pp 126-133, Mar 23-26, 1999

¹¹ X. Ge and P. Smyth, "*Deformable Markov model templates for time-series pattern matching*", In proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, pp 81-90, Aug 20-23, 2000

¹² E. Keogh, K. Chakrabarti, M. Pazzani and Mehrotra, "*Dimensionality reduction for fast similarity search in large time series databases*", Journal of Knowledge and Information Systems, pp 263-286, 2000

¹³ B. K. Yi, H. Jagadish and C. Faloutsos, "*Efficient retrieval of similar time sequences under time warping*", IEEE International Conference on Data Engineering, pp 201-208, 1998

¹⁴ D. Gunopulos and G Das, “*Time Series Similarity Measures*”, In Tutorial Notes of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining, pp 243–307, 2000

¹⁵ J. Han, J. Pei, Y. Yin and R. Mao , “*Mining frequent patterns without candidate generation: a frequent-pattern tree approach*”, Data Mining and Knowledge Discovery, v.8 n.1, pp 53 – 87, 2004

¹⁶ E. Keogh and S. Kasetty, “*On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration*”, Data Mining and Knowledge Discovery, v.7 n.4, pp 349 – 371, 2003

¹⁷ M. Tompa and J. Buhler, “*Finding motifs using random projections*”, In proceedings of the 5th International Conference on Computational Molecular Biology, Montreal, Canada, pp 67-74, Apr 22-25, 2001

¹⁸ B. Chiu, E. Keogh and S. Lonardi, “*Probabilistic Discovery of Time Series Motifs*”, In the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, pp 493-498, Aug 24-27, 2003

¹⁹ J. Lin, E. Keogh, P. Patel and S. Lonardi, “*Finding motifs in time series*”, In the 2nd Workshop on Temporal Data Mining, at the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, 2002

²⁰ W. Wilson, P. Berkin, and U. Aickelin , “*Motif detection inspired by immune memory*”, In the Proceedings 6th International Conference, ICARIS, 2007

²¹ T. Oates, “*PERUSE: An Unsupervised Algorithm for Finding Recurring Patterns in Time Series*”, Proc. IEEE, pp 330-337, 2002

²² P. S. Addison, “*The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, medicine and Finance*”, IOP Publishing, 2002

²³ R. Polikar, “*The Story of Wavelets in Physics and Modern Topics in Mechanical and Electrical Engineering*”, World Scientific and Engineering, Society Press, pp 192-197, 1999

²⁴ I. Daubechies, “*Where do Wavelets Come From? - A Personal Point of View*”, Proc. IEEE, v. 84 n. 4, pp 510-513, 1996

²⁵ G. Icke, “*Entropy Based Feature Extraction Methods from Discrete Wavelet Transform Coefficients*”, Masters Thesis, Northeastern University, 2005

²⁶ D. Gabor, “*Theory of Communication*”, Journal of the IEE, v. 93, pp 429-457, 1946

²⁷ L. Debnath, “*Wavelet Transforms and Their Applications*”, Birkhäuser, 2002

²⁸ S. Mallat, “*Multiresolution Approximations and Wavelet Orthonormal Bases of $L_2(\mathbb{R})$* ”,
Trans. American Math. Soc., v. 315, pp 69-87, 1989

²⁹ B. Jawerth and W Sweldens, “*An Overview of Wavelet Based Multiresolution Analysis*”, SIAM Review, v. 36, pp 377-412, 1994

³⁰ S. Mallat, “*A Theory for Multiresolution Signal Decomposition: The Wavelet Representation*”, IEEE Transactions on Pattern Analysis and Machine Intelligence, v.11,
pp 674-693, 1989

³¹ M. Shensa, “*The Discrete Wavelet Transform: Wedding the à Trous and Mallat Algorithms*,” IEEE Transactions on Signal Processing, v. 40, n.10, pp 2464-2482, 1992

³² P. Dutilleul, “*An implementation of the ‘algorithme à trous’ to Compute the Wavelet Transform*”

³³ A. Aussem, J. Campbell and F. Murtagh, “*Wavelet-Based Feature Extraction and Decomposition Strategies for Financial Forecasting*”, To appear in Journal of Computational Intelligence in Finance, 1997

³⁴ D. B. Percival and A. T. Walden, “*Wavelet Methods for Time Series Analysis*”, Cambridge University Press, pp 159-204, 2000

³⁵ Z. R. Struzik and A. Siebe, “*Wavelet transform in similarity paradigm*”, Research and Development in Knowledge Discovery and Data Mining, X. Wu, R. Kotagiri and K. B. Korb, Eds. Berlin, Germany pp 295-309, 1998

³⁶ E. Keogh, “*Fast similarity search in the presence of longitudinal scaling n time series databases*”, Proceedings of the 9th IEEE International Conference on tools with Artificial Intelligence, Newport Beach, CA, pp 578-584, 1997

³⁷ M. Beneke, L. M. Leemis, R. E. Schlegel and B. L. Foote, “*Spectral analysis in quality control: a control chart based on the periodogram*”, Technometrics, v. 30, pp 63--70, Feb, 1988

³⁸ J. D. Spurrier and L. A. Thombs, “*Control charts for detecting cyclical behavior*”, v.32, pp 163-171, May, 1990

³⁹ T.H. Hou and L. Lin, “*Using neural networks for the automatic monitoring and*

recognition of signals in manufacturing processes”, In Design and Implementation of Intelligent Manufacturing Systems - From Expert Systems, Neural Networks, to Fuzzy Logic, H. R. Parsaei and M. Jamshidi, Eds. NJ: Prentice Hall, pp 141-160, 1995

⁴⁰ H. B. Hwang, “*Proper and effective training of a pattern recognizer for cyclic data*”, IIE Transactions, v. 27, pp 746-756, Jun, 1995

⁴¹ J. Theiler, “*Efficient algorithm for estimating the correlation dimension from a set of discrete points*”, Phys. Rev. A, v. 36, pp 4456-4462, Nov, 1987

⁴² P. Grassberger, “*An optimized box-assisted algorithm for fractal dimensions*”, Phys. Lett. A, v. 148, pp 63-68, Jan, 1990

⁴³ J.R. Chen, “*Making subsequence time series clustering meaningful*”, Proceedings of the 5th IEEE International Conference on Data Mining, pp 8, 2006.

Appendix

Method to Identify Recurring Patterns in a Process Signal

```
%% Signal s0 input

clear all;
load ('testsig2.dat')
s0=testsig2;

%%Finding the length of signal

n=length(s0);

%% Creating matrix MAT1 and storing s0

As0=[0 0 0 s0 0 0 0 0];
MAT1=zeros(8,1);

%%MAT1 contains 8 rows and n columns . Moving window of length 8 (for bn) is used
on s0 and those values are stored in MAT1

for i=1:1:n
    MAT1=[MAT1 As0(i:1:i+7)'];
end

%% Computing local frequency index with Haar Wavelet decomposition

w=zeros(1,1);
for i=1:1:n
    [C L]=wavedec(MAT1(:,i+1),3,'haar');
    d1=detcoef(C,L,1);
    d2=detcoef(C,L,2);
    d3=detcoef(C,L,3);
    NUM= sum(d1.*d1)+sum(d2.*d2)+sum(d3.*d3);

    T=MAT1(:,i+1).*MAT1(:,i+1);
    DEN=sum(T);
    w(i)=NUM/DEN;
end

%% Matrix MS0 contains the enriched process signal s0 i.e, s0 along with computed
local frequency index next to each value
```

```

MS0=zeros(1,n*2);
MS0(1:2:n*2)=s0;
MS0(2:2:n*2)=w(1:1:end);
dlmwrite('esignal.dat',MS0,'\t');
nn=size(MS0);
nn=nn(1,2);

%% Finding Max and Min value from the enriched process signal

ymax=max(MS0(1:2:end));
ymin=min(MS0(1:2:end));
wmax=max(MS0(2:2:end));
wmin=min(MS0(2:2:end));
ymax
ymin
wmax
wmin

%%P and Q input from user

P=input('input P :');
Q=input('input Q:');

%% Computing tolerances

e1=(ymax-ymin)./P;
e2=(wmax-wmin)./Q;

%% Calculation of a1 and w1

an=floor((MS0(1:2:end)-ymin)./e1);
wn=floor((MS0(2:2:end)-wmin)./e2);

%% Enriched process signal rn contains non negative integers computed using values a1
and w1

rn(1:2:nn)=an;
rn(2:2:nn)=wn;

%% Finding minimal subsequence length ln

ln=(2*(log10(n)/log10(3))+3)/((log10(P*Q)/log10(3))-1);
ln=ceil(ln);

%% Keeping minimal subsequence length ln as multiple of 10 for multi-resolution level
L

```

```

%ln=L*10;
ln

%% Matrix V contains enriched vectors of length 2*ln

V=zeros(n-ln+1,2*ln);
for pp=1:1:n-ln+1
    V(pp,:)=rn((2*pp-1):1:(2*pp-1)+2*ln-1);
end

%% Matrix R contains vectors of length 2*ln

R=zeros(n-ln+1,2*ln);
for pp1=1:1:n-ln+1
    R(pp1,:)=MS0((2*pp1-1):1:(2*pp1-1)+2*ln-1);
end

[rsize,colsiz] = size(R);
VT=1:rsize;
newR=[R,VT'];

%%PATTERN SEARCH PART

[q,r]=size(V)
T=1:q;
NM=[V,T'];
SM=sortrows(NM,1);
W=zeros(1,1);
%W1=zeros(1,1);
for i=1:q-2
    SM1=SM(i:end-1,:);
    [q1,r1]=size(SM1);
    W={};

    for j=1:r1-1

        if isempty(SM1)
            break
        end
        [q3,r3]=size(SM1);

        I=SM1(1,:);

        for k=2:q3

```

```

difference=abs(SM1(k,j)-SM1(1,j));

if difference<=1
    I=[I;SM1(k,:)];
end

if k==q3

    if numel(I)>r1
        W{j}=I;
        SM1=I;
    else
        W{j}=[];
        %W{j}=zeros(1,128) zeros(1,128)
        SM1=[];
    end

end

end

end

if numel(W)>=r1-1
    W1{i}=W{r1-1};
end

end

%% Making pairs using nchoosek

FINV=[0 0];

count=0;
pair=zeros(1,1);
H=zeros(1,1);
for cc=1:1:size(W1,2)
    H=[0 0];
    %cc=1;
    if isempty(W1{1,cc})
        continue
    else

        pair=nchoosek(W1{1,cc}(:,2*ln+1),2);
        per=pair;
    end
end

```

```

len=size(pair,1);
i=1:1:len;
kk=abs(pair(i,1)-pair(i,2));
pair=[pair kk];
%H=[0 0];
%count=0;
for i=1:1:len
    if pair(i,3)>=ln
        count=count+1;
        H=[H;pair(i,1:2)];
    end
end

%% condition for Y n W

for i=2:1:size(H,1)
    sd=newR(H(i,1),:);
    sd1=newR(H(i,2),:);
    if ((sd(1:2:end-1)-sd1(1:2:end-1)<e1)) &((sd(2:2:end)-sd1(2:2:end)<e2)) &
abs((sd(2*ln+1))- (sd1(2*ln+1)))>=ln
        FINV=[FINV ; H(i,:)];
    end
end
end
end

%% Signal Expansion

FINV=FINV(2:end,:);
ct=0;
alp=zeros(1,1);
bet=zeros(1,1);
alp2=zeros(1,1);
bet2=zeros(1,1);

for i=1:1:size(FINV,1) %

    P=FINV(i,1);
    Q=FINV(i,2);

    %% Backward Extension

    p=P;
    q=Q;
    cc=1;

```

```

%% for the first column
while (cc==1)

    ct=ct+1;
    p=p-1;
    q=q-1;
    if p==0
        p=p+1;
        break;
    end
    if q==0
        q=q+1;
        break;
    end

    sd=R(p,:);
    sd1=R(q,:);

    if (sd(1:2:end)-sd1(1:2:end)<e1) &(sd(2:2:end)-sd1(2:2:end)<e2) &(Q-ct>=P+ln)
        cc=1;
    else
        cc=0;
    end
end%% end of while

alp(i)=p-1;
bet(i)=q-1;

%% Forward Extension
p=P;
q=Q;
ct=0;
cc=1;

%% for the first column
while (cc==1)
    ct=ct+1;
    p=p+1;
    q=q+1;
    if q>=size(R,1)
        q=q-1;
        break;
    end
    if p>=size(R,1)

```

```

        p=p-1;
        break;
    end

    sd=R(p,:);
    sd1=R(q,:);

    if (sd(1:2:end)-sd1(1:2:end)<e1) &(sd(2:2:end)-sd1(2:2:end)<e2) &(Q-ct>=P+ln)
        cc=1;
    else
        cc=0;
    end
end
end
alp2(i)=p-1;
bet2(i)=q-1;
end

%%Arranging the resultant pattern

ALP = [alp;alp2];
BLP = [bet;bet2];
ANSWER=[ALP;BLP];
ANSWER
clear all;

```

Multi-scale decomposition using *a trous* Algorithm

```
%%  
%Low pass filter h  
  
h=[1/16 1/4 3/8 1/4 1/16];  
  
% Number of scales j=5  
j=1;  
c{j}=dlmread('testsig2.dat');  
  
%%  
%%calculating wavelet coefficients  
  
for j=2:1:6  
    L=4*(2^(j-2))+1;  
    K{j}=zeros(1,L);  
  
    %Inserting zeros between low pass filter  
    K{j}(1:2^(j-2):L)=h;  
  
    %Convolution of signal and filter  
    c{j}=conv2(c{j-1},K{j},'same');  
  
    %Wavelet coefficients  
    w{j}=c{j-1}-c{j};  
end  
  
%Storing wavelet coefficients at different levels to be inputted to pattern search  
algorithm  
  
dlmwrite('L1.dat',w{2},'\t');  
dlmwrite('L2.dat',w{3},'\t');  
dlmwrite('L3.dat',w{4},'\t');  
dlmwrite('L4.dat',w{5},'\t');  
dlmwrite('L5.dat',w{6},'\t');  
  
%%  
%Signal reconstruction  
%-----  
residual=c{6};  
for i=1:1:5  
    residual=residual+w{6-(i-1)};
```

```

end

%%
for k=2:1:6
figure;
o=int2str(k);

%Plotting wavelet coefficients
%-----
subplot(2,1,1),plot(w{k});
oo=strcat('W',o-1);
title(oo);

%Plotting detail coefficients
%-----
subplot(2,1,2),plot(c{k});
oo1=strcat('C',o-1)
title(oo1);
end

%%
%Plotting the input signal
%-----
figure;
plot(c{1});
title('Input Signal');

```