BAŞKENT UNIVERSITY
INSTITUTE OF SCIENCE AND ENGINEERING

USE OF INTELLIGENT METHODS IN HARMONIC ANALYSIS OF POWER SYSTEMS

EMRE ÖNER TARTAN

MASTER OF THESIS
2011

This thesis, titled: “Use of Intelligent Methods in Harmonic Analysis of Power Systems”, has been approved in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONICS ENGINEERING, by our jury, on 24/08/2011.

Chairman (Supervisor) :

(Asst. Prof. Dr. Hamit ERDEM)

Member :

(Asst. Prof. Dr. Mustafa DOĞAN)

Member :

(Asst. Prof. Dr. Metin Yıldız)

APPROVAL

…./…./2011

Prof. Dr. Emin AKATA

Director

Institute of Science and Engineering

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Assist. Prof. Dr. Hamit Erdem for his guidance, advices and encouragement.

I am grateful to my parents for their patience and great support.

ABSTRACT

USE OF INTELLIGENT METHODS IN HARMONIC ANALYSIS OF POWER SYSTEMS

Emre Öner TARTAN

Başkent University Institute of Science and Engineering

Department of Electrical and Electronics Engineering

In the last decades, with the increasing use of non-linear electronic equipments in power systems, harmonic pollution has become an important issue in power quality. To prevent problems due to the harmonics and to improve the quality of the delivered energy, estimation of harmonic parameters magnitude and phase angle, is an important task. Conventionally Fourier Transform based algorithms are the most commonly used techniques for harmonic estimation, however these techniques have certain limitations and drawbacks. On the other hand, recently several alternative algorithms utilizing intelligent methods have been proposed for harmonic estimation. In this thesis, three hybrid algorithms, composed by combining Least Squares Method with evolutionary computation algorithms, are applied in harmonic estimation. These algorithms, by utilizing the feature that harmonic estimation problem is linear in amplitude and nonlinear in phase, use evolutionary computation algorithms for phase angle estimation and Least Squares Method for amplitude estimation. Two of the hybrid algorithms using Genetic Algorithm and Particle Swarm Optimization for phase angle estimation, were introduced in literature previously. In this thesis a novel hybrid algorithm which uses Differential Evolution for phase angle estimation, instead of Genetic Algorithm or Particle Swarm Optimization, is presented. The applications are realized in simulation environment and the results of the algorithms are compared.

KEY WORDS: Harmonic Estimation, Least Squares Method, Genetic Algorithms Particle Swarm Optimization, Differential Evolution

Supervisor: Assist. Prof. Dr. Hamit ERDEM, Başkent University, Department of Electrical and Electronics Engineering

ÖZ

GÜÇ SİSTEMLERİNİN HARMONİK ANALİZİNDE AKILLI YÖNTEMLERİN UYGULANMASI

Emre Öner TARTAN
Başkent Üniversitesi Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Son on yıllarda, güç sistemlerinde lineer olmayan elektronik cihazların artan kullanımıyla birlikte harmonik kirliliği güç kalitesinde önemli bir konu haline gelmiştir. Harmoniklere bağlı sorunları önlemek ve iletilen enerjinin kalitesini iyileştirmek için, harmonik parametreleri olan faz açısı ve genliğin kestirimi önemli bir görevdir. Geleneksel olarak, harmonik kestiriminde Fourier Dönüşümü tabanlı algoritmalar en çok kullanılan tekniklerdir ancak bu tekniklerin belirli kısıtlamaları ve kusurları vardır. Öte yandan, son zamanlarda harmonik kestirimi için akıllı yöntemleri kullanan çeşitli alternatif algoritmalar önerilmiştir. Bu tezde, En Küçük Kareler Yöntemi ile evrimsel hesaplama algoritmalarının birleştirilmesiyle oluşturulan üç hibrid algoritma harmonik kestiriminde uygulanmıştır. Bu algoritmalar, harmonik kestirimi probleminin genlikte doğrusal olması ve fazda doğrusal olmaması özelliğinden faydalanarak, faz açısının kestirimi için evrimsel hesaplama algoritmalarını ve genlik kestirimi için En Küçük Kareler Yöntemini kullanmaktadırlar. Bu hibrid algoritmalardan, faz açısı kestiriminde Genetik Algoritma ve Parçacık Sürü Optimizasyonunu kullanan ikisi daha önce literatürde tanıtılmıştır. Bu tezde Genetik Algoritma veya Parçacık Sürü Optimizasyonu yerine faz açısının kestiriminde Türevsel Evrimi kullanan yeni bir hibrid algoritma sunulmuştur. Uygulamalar simülasyon ortamında gerçekleştirilmiş ve algoritmaların sonuçları karşılaştırılmıştır.

ANAHTAR SÖZCÜKLER: Harmonik Kestirimi, En Küçük Kareler Yöntemi, Genetik Algoritmalar, Parçacık Sürü Optimizasyonu, Türevsel Evrim

Danışman: Yrd. Doç. Dr. Hamit ERDEM, Başkent Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

GA	Genetic Algorithm
PSO	Particle Swarm Optimization
PC	Passive Congregation
DE	Differential Evolution
FFT	Fast Fourrier Transform
DFT	Discrete Fourier Transform
LS	Least Squares
SNR	Signal to Noise Ratio
SLG	Single- line-to-ground
$u_{i,g}$,	Trial vector
$x_{i,g}$	Target vector
$v_{i,g}$.	Mutant vector
P_c	Crossover probability
P_m	Mutation probability
ε_1	Error index for distorted waveform
ε_2	Error index for fundamental component
ε_3	Proposed new error index for fundamental component

1. INTRODUCTION

With the increasing use of power electronic components within the distribution system, power quality has become an important concern of electric utilities and end-users. Power quality covers all aspects of power system engineering from transmission and distribution level analyses to end-user problems [1]. In the last years, the wide usage of nonlinear loads such as adjustable speed drives; electronically ballasted lighting; and the power supplies of every computer, copier, and fax machine and much of the telecom equipment used in modern offices has resulted in increasing injection of harmonics to the power line. These have detrimental effects including communication interference, loss of reliability, increased operation costs, equipment overheating, machine transformer and capacitor failures, and inaccurate power metering [1]. Therefore, harmonic distortion has drawn much attention and has been considered as one of the most important problems in power quality [1]. To cancel the harmonic distortion, the undesired harmonic components of current or voltage should be detected, then the harmonics of equal magnitude but opposite phase should be injected. Hence, to prevent problems due to the harmonics and to improve the quality of the delivered energy, measurement and estimation of harmonic parameters, magnitude and phase angle, has become an important task.

Conventionally Fourier Transform based algorithms are the most commonly used techniques for harmonic estimation. However, Fourier transform based algorithms has certain limitations in harmonic analysis. Circumstances are encountered in practice for which the Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT) cannot be relied on to achieve valid harmonic component identification where there are existing noise signals which are not integer multiples of the supply frequency [2]. Another approach utilizes Kalman Filtering-based technique to estimate the harmonic components. This approach is accurate, but requires the correct definition of the state equations, measurement equations and covariance matrices [3;4], in other words a priori knowledge.

On the other hand alternatively several methods utilising intelligent methods have been proposed for harmonic estimation. Intelligent methods is a broad term, covering a range of computing techniques that have emerged from research into artificial intelligence. It includes symbolic approaches and numerical approaches such as neural networks, biologically inspired algorithms, fuzzy logic and hybrid of different methods [5]. Among intelligent methods neural networks (NN) have been popular and applied in several studies for harmonic estimation [2;6-15]. However, NNs require training procedure and optimal network structure design for each different case, therefore this approach lacks adaptability and flexibility. Another intelligent method approach utilizes evolutionary computation algorithms [16-19]. The recent studies also combine least squares method (LS) and propose hybrid algorithms for harmonic estimation [17-19].

Evolutionary computation algorithms are population-based stochastic optimization algorithms. Unlike gradient based methods which take step in the direction of the most negative slope of error surface and can fall at a local optimum in multi modal problems, evolutionary algorithms provide efficient multi-search in problem domain by individuals composing their population. Thus these algorithms are unlikely to be trapped in local minima and efficient in optimization of nondifferentiable, nonlinear and multimodal objective functions.

Genetic Algorithms (GA) are a class of evolutionary computation algorithms inspired by the process of natural evolution developed by Holland [20]. Bettayeb and Qidwai [17] combined genetic algorithm with least squares method to estimate phase angles and amplitudes of harmonics accurately at the same time and to provide improvement in convergence time as compared to using the ordinary GA [16]. The proposed hybrid algorithm takes advantage that the harmonic estimation problem is linear in amplitude and nonlinear in phase. Algorithm iterates between linear least squares amplitude estimation and the nonlinear GA-based phase estimation**.**

Particle Swarm Optimization (PSO) is a newer evolutionary computation algorithm, more specifically member of swarm intelligent techniques, inspired by social behavior of bird flocking or fish schooling, discovered by Kennedy and Eberhart [21] in 1995. Lu et al. [19] proposed using a variant of PSO, particle swarm optimization with passive congration (PSOPC), instead of genetic algorithm for the estimation of phase angles and the least squares method for estimation of amplitudes. This approach is proposed to achive an improved performance over the conventional GA and FFT schemes even in the presence of noise. Also PSO algorithm needs few parameters to be adjusted, hence is easy to apply and adapt in different applications.

Differential Evolution (DE) is one of the most prominent new generation evolutionary computation algorithms, proposed by Storn and Price [22], to exhibit consistent and reliable performance in nonlinear and multimodal environment. DE has shown good performance on many real-world problems and on the majority of the numerical benchmark problems as well. In the last years, DE has been widely applied to many optimization problems [23] and great number of differential evolution publications in scientific journals and conference proceedings are seen. In comparative studies DE has proven its superiority as the best performing algorithm over other evolutionary algorithms for many problems [24]. In these studies, it is shown that DE is reliable, robust and efficient optimization algorithm. Besides, one major advantage of DE is the ease of the application. DE needs few parameters to be adjusted and has a simple vector based iterative algorithm.

The motivation of this thesis is, to develop a method that is easy to apply, adaptable to different cases and accurate, and to achieve an better performance than the applied hybrid algorithms, present and apply a novel hybrid algorithm that uses DE for phase angle estimation and least squares method for amplitude estimation.

Organization of this thesis is as follows

Chapter 2 gives introduction to power quality. Various power quality problems, reasons and effects are discussed briefly.

In Chapter 3, harmonics, sources and effects of harmonic distortion are discussed. Conventional method for harmonic analysis and the drawbacks of this method are described.

In Chapter 4, Least Squares Method and Genetic Algorithms are explained. Structural property of harmonic estimation problem and hybrid algorithm approach for harmonic estimation is described. LS-GA based hybrid algorithm is given.

In Chapter 5, Particle Swarm Optimization and passive congregation concept is explained. LS-PSOPC based hybrid algorithm for harmonic estimation is described.

In Chapter 6, Differential Evolution is described and a novel LS-DE based hybrid algorithm is presented.

In chapter 7, applications of three hybrid algorithms are described and simulation results are given.

In chapter 8, by analyzing simulation results, the proposed novel LS-DE based hybrid algorithm is discussed and compared with LS-GA based and LS-PSOPC based hybrid algorithms and the final conclusion has been given.

In simulations considering the majority of nonlinear loads produce harmonics that are odd multiples of the fundamental frequency and for the purpose of comparison, the same sample distorted wave used in previous studies [17;18;19], is generated and used in simulations of the algorithms. This sample distorted signal includes 5rd, 7th, 11th and 13th harmonics, which emerge at the terminal of the load bus with six-pulse full-wave bridge rectifier. The simulations are performed in MATLAB environment, for noisy and non-noisy conditions.

2. POWER QUALITY

2.1 Introduction to Power Quality

Power quality is a broad term having different definitions, generally meant to express the quality of voltage or the quality of current and can be defined as: the measure, analysis, and improvement of the bus voltage to maintain a sinusoidal waveform at rated voltage and frequency [1]. A simpler and perhaps more general definition might state: “Power quality is a set of electrical boundaries that allows a piece of equipment to function in its intended manner without significant loss of performance or life expectancy” [31]. Power quality affects all connected electrical and electronic equipment, distribution and transmission lines and any power problem manifested in voltage, current, or frequency deviations results in failure or misoperation of customer equipment [25]. The dependence of modern life upon the continuous supply of electrical energy makes system reliability and power quality topics of utmost importance in electric power system area. Power quality has economic impacts on utilities, their customers and suppliers of load equipment. It has not only direct importance for economic losses of utilities and industrial consumers but also for service quality for the end users. Distortion sources of power quality problems can be divided into four categories [1]: unpredictable events, the electric utility, the customer, and the manufacturer.

Both electric utilities and end users agree that more than 60% of power quality problems are generated by natural and unpredictable events. As another category, there are three main sources of poor power quality related to utilities: the point of supply generation, the transmission system and the distribution system. Customer loads also generate a considerable portion of power quality problems in today's power systems and harmonic distortion draws attention as one of the most important end-user related problems. Lastly, two main sources of poor power quality related to manufacturing ragulations are due to standards and equipment sensitivity. The lack of standards for testing, certification, installation, use of electronic equipment and appliances, and proliferation of sensitive electronic equipments are major causes of power quality [1].

2.2 Power Quality Problems

There are different classifications for power quality issues, each using a specific property to categorize the problem. Some of them classify the events as "steady-state" and "non-steady-state" phenomena. In some regulations (e.g., ANSI C84.1 [26]) the most important factor is the duration of the event. Other guidelines (e.g., IEEE-519 [27]) use the wave shape (duration and magnitude) of each event to classify power quality problems. Other standards (e.g., IEC [28]) use the frequency range of the event for the classification. the principal phenomena causing electromagnetic disturbances according to IEC classifications. IEEE standards use several additional terms (as compared with IEC terminology) to classify power quality events. Table 2.1 provides information about categories and characteristics of electromagnetic phenomena defined by IEEE-1159 [29].

2.2.1 Transients

Power system transients are undesirable, fast and short-duration events that produce distortions. Their characteristics and waveforms depend on the mechanism of generation and the network parameters (e.g., resistance, inductance, and capacitance) at the point of interest. "Surge" is often considered synonymous with transient. Transients are usually classified into two categories: impulsive and oscillatory

An impulsive transient is a sudden frequency change in the steady-state condition of voltage, current, or both that is unidirectional in polarity. The most common cause of impulsive transients is a lightning current surge.

An oscillatory transient is a sudden frequency change in the steady-state condition of voltage, current, or both that includes both positive and negative polarity values. Oscillatory transients occur for different reasons in power systems such as appliance switching, capacitor bank switching, fastacting overcurrent protective devices, and ferroresonance. Impulsive and oscillatory transients are shown in figure 2.1.

Table 2.1 Categories and typical characteristics of power system electromagnetic phenomena [29]

Categories		Typical spectral content	Typical duration	Typical voltage magnitude
1. Transient	1.1. Impulsive			
	• nanosecond	5 ns rise	<50 ns	
	• microsecond	1 μs rise	50 ns–1 ms	
	• millisecond	0.1 ms rise	>1 ms	
	1.2. Oscillatory			
	• low frequency	<5 kHz	0.3–50 ms	0–4 pu
	• medium frequency	5–500 kHz	20 μs	0–8 pu
	• high frequency	0.5–5 MHz	5 μs	0–4 pu
2. Short-duration variation	2.1. Instantaneous			
	• interruption		0.5–30 cycles	<0.1 pu
	• sag		0.5–30 cycles	0.1–0.9 pu
	• swell		0.5–30 cycles	1.1–1.8 pu
	2.2. Momentary			
	• interruption		0.5 cycle–3 s	<0.1 pu
	• sag		30 cycles–3 s	0.1–0.9 pu
	• swell		30 cycles–3 s	1.1–1.4 pu
	2.3. Temporary			
	• interruption		3 s–1 min	<0.1 pu
	• sag		3 s–1 min	0.1–0.9 pu
	• swell		3 s–1 min	1.1–1.2 pu
3. Long-duration variation	3.1. Sustained interruption		>1 min	0.0 pu
	3.2. Undervoltage		>1 min	0.8–0.9 pu
	3.3. Overvoltage		>1 min	1.1–1.2 pu
4. Voltage imbalance			steady state	0.5–2%
5. Waveform distortion	5.1. DC offset		steady state	0–0.1%
	5.2. Harmonics	0–100th	steady state	0–20%
	5.3. Interharmonics	0–6 kHz	steady state	0–2%
	5.4. Notching		steady state	
	5.5. Noise	Broadband	steady state	0–1%
6. Voltage fluctuation		<25 Hz	intermittent	0.1–7%
7. Power frequency variations			<10 s	

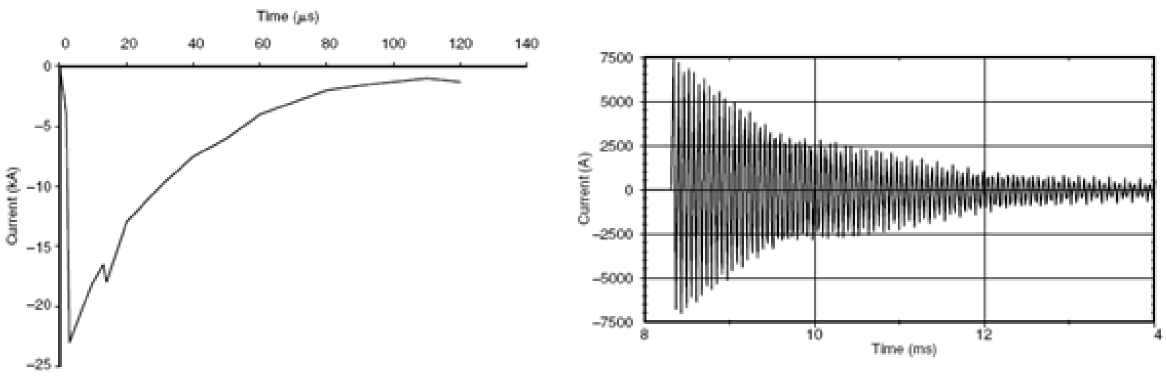

Figure 2.1 (a) Lighting stroke current impulsive transient (b) Oscillator transient current caused by back-to-back capacitor switching [29]

2.2.2 Long-duration voltage variations

Long-duration variations encompass root-mean-square (rms) deviations at power frequencies for longer than 1 min. Long-duration variations can be either overvoltages or undervoltages. Overvoltages and undervoltages generally are not the result of system faults, but are caused by load variations on the system and system switching operations.

2.2.2.1 Overvoltage

An overvoltage is an increase in the rms ac voltage greater than 110 percent at the power frequency for a duration longer than 1 min. Overvoltages are usually the result of load switching (e.g., switching off a large load or energizing a capacitor bank). The overvoltages result because either the system is too weak for the desired voltage regulation or voltage controls are inadequate. Incorrect tap settings on transformers can also result in system overvoltages.

2.2.2.2 Undervoltage

An undervoltage is a decrease in the rms ac voltage to less than 90 percent at the power frequency for a duration longer than 1 min. Undervoltages are the result of switching events that are the opposite of the events that cause overvoltages. A load switching on or a capacitor bank switching off can cause an undervoltage until voltage regulation equipment on the system can bring the voltage back to within tolerances. Overloaded circuits can result in undervoltages also.

2.2.3 Short-duration voltage variations

This category encompasses the IEC category of voltage dips and short interruptions. Each type of variation can be designated as instantaneous, momentary, or temporary, depending on its duration as defined in Table 2.1.

2.2.3.1 Interruption

An interruption occurs when the supply voltage or load current decreases to less than 0.1 pu for a period of time not exceeding 1 min. Interruptions can be the result of power system faults, equipment failures, and control malfunctions.

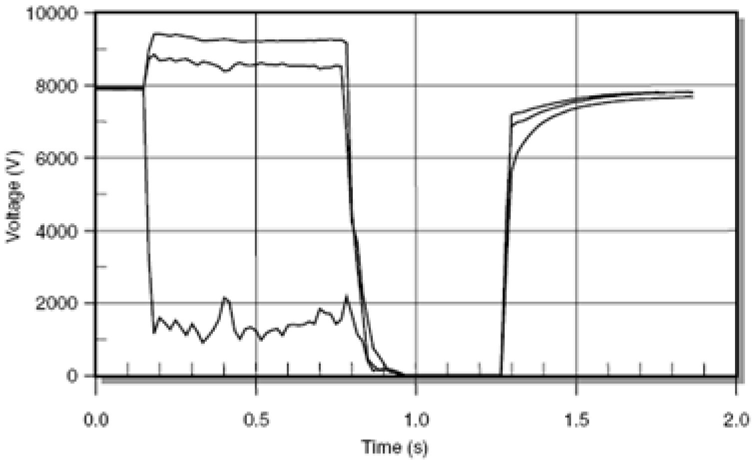

Figure 2.2 Three-phase rms voltages for a momentary interruption due to a fault and subsequent recloser operation [25]

2.2.3.2 Sags (dips)

A sag is a decrease to between 0.1 and 0.9 pu in rms voltage or current at the power frequency for durations from 0.5 cycle to 1 min. Voltage sags are usually associated with system faults but can also be caused by energization of heavy loads or starting of large motors. Figure 2.6 shows a typical voltage sag that can be associated with a single- line-to-ground (SLG) fault on another feeder from the same substation and Figure 2.3 illustrates the effect of a large motor starting.

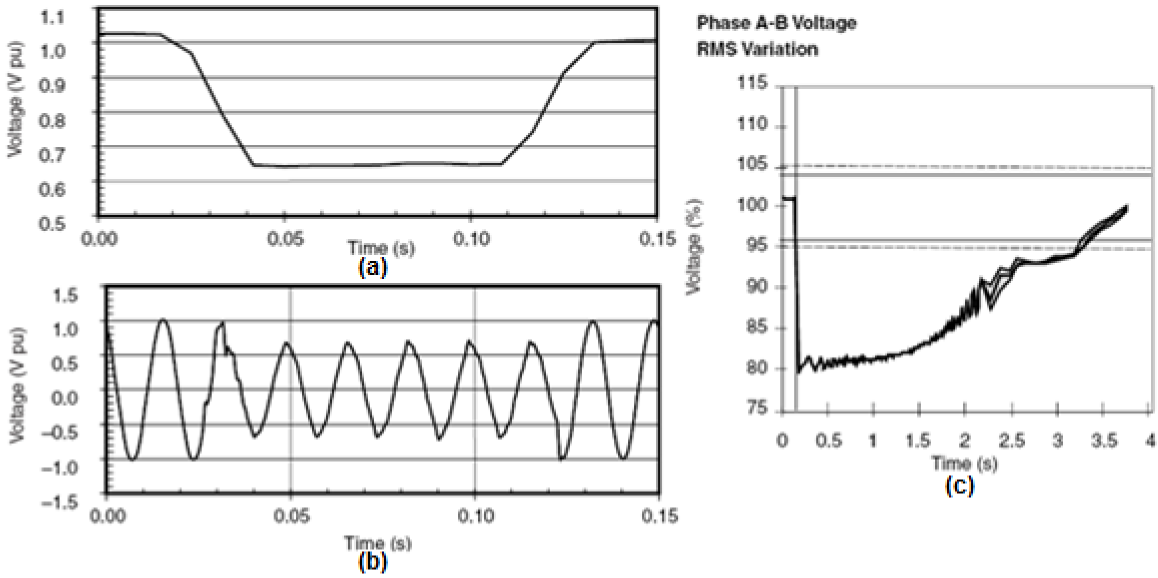

Figure 2.3 (a) RMS waveform for vol. sag caused by SLG fault (b) Vol. sag waveform caused by SLG fault (c) Vol.sag caused by motor starting [25]

2.2.3.3 Swells

A swell is defined as an increase to between 1.1 and 1.8 pu in rms voltage or current at the power frequency for durations from 0.5 cycle to 1 min. As with sags, swells are usually associated with system fault conditions, but they are not as common as voltage sags. One way that a swell can occur is from the temporary voltage rise on the unfaulted phases during an SLG fault., Figure 2.4 illustrates a voltage swell caused by an SLG fault. Swells can also be caused by switching off a large load or energizing a large capacitor bank.

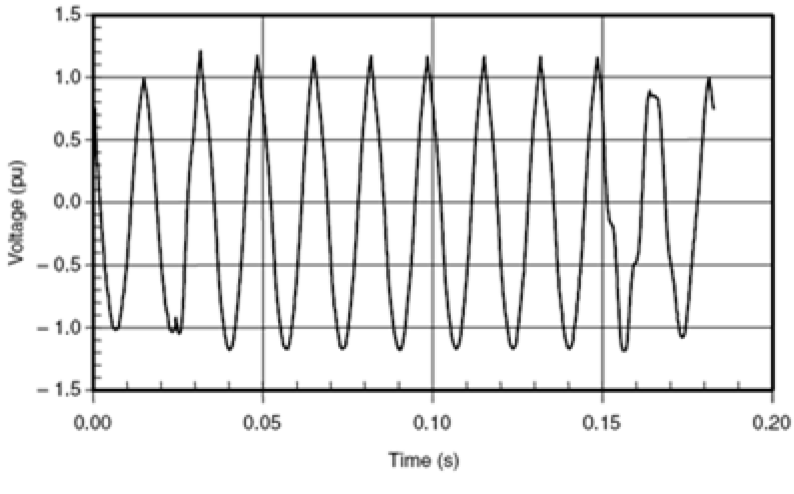

Figure 2.4 Instantaneous voltage swell caused by an SLG fault [25]

2.2.4 Voltage imbalance

Voltage imbalance (also called voltage unbalance) is sometimes defined as the maximum deviation from the average of the three-phase voltages or currents, divided by the average of the three-phase voltages or currents, expressed in percent. Voltage imbalance trend for a residential feeder is shown in figure 2.5.

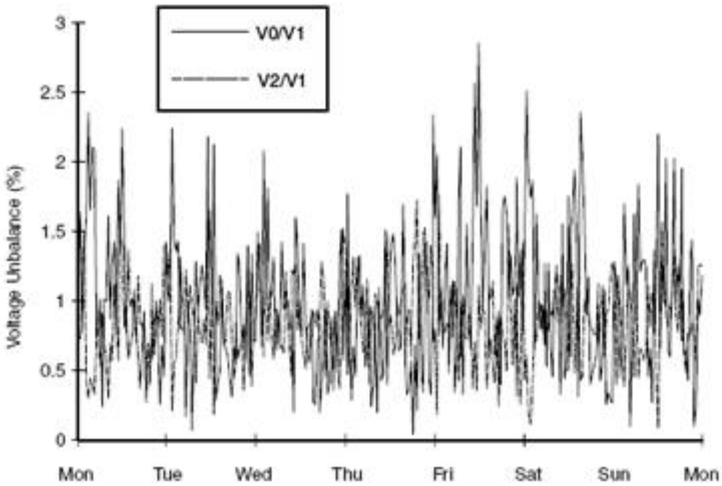

Figure 2.5 Voltage imbalance trend for a residential feeder [29]

2.2.5 Waveform distortion

Waveform distortion is defined as a steady-state deviation from an ideal sine wave of power frequency principally characterized by the spectral content of the deviation. There are five primary types of waveform distortion:

- DC offset
- Interharmonics
- Notching
- Noise
- Harmonics

The next chapter is devoted to discussion of harmonics.

2.2.5.1 DC offset

The presence of a dc voltage or current in an ac power system is termed *dc offset.* This can occur as the result of a geomagnetic disturbance or asymmetry of electronic power converters. Incandescent light bulb life extenders, for example, may consist of diodes that reduce the rms voltage supplied to the light bulb by half-wave rectification. Direct current in ac networks can have a detrimental effect by biasing transformer cores so they saturate in normal operation. This causes additional heating and loss of transformer life. Direct current may also cause the electrolytic erosion of grounding electrodes and other connectors.

2.2.5.2 Interharmonics

The frequency of interharmonics are not integer multiples of the fundamental frequency. Interharmonics appear as discrete frequencies or as a band spectrum. Main sources of interharmonic waveforms are static frequency converters, cycloconverters, induction motors, arcing devices, and computers. Interharmonics cause flicker, low-frequency torques, additional temperature rise in induction machines , and malfunctioning of protective (under-frequency) relays. It is generally the result of frequency conversion and is often not constant; it varies with load. Such interharmonic currents can excite quite severe resonances on the power system as the varying interharmonic frequency becomes coincident with natural frequencies of the system. They have been shown to affect power-line-carrier signaling and induce visual flicker in fluorescent and other arc lighting as well as in computer display devices.

2.2.5.3 Notching

Notching is a periodic voltage disturbance caused by the normal operation of power electronic devices when current is commutated from one phase to another.. The notching appears in the line voltage waveform during normal operation of power electronic devices when the current commutates from one phase to another. During this notching period, there exists a momentary short - circuit between the two commutating phases, reducing the line voltage; the voltage reduction is limited only by the system impedance. measurement equipment normally used for harmonic analysis. Figure 2.6 shows an example of voltage notching from a three-phase converter that produces continuous dc current.

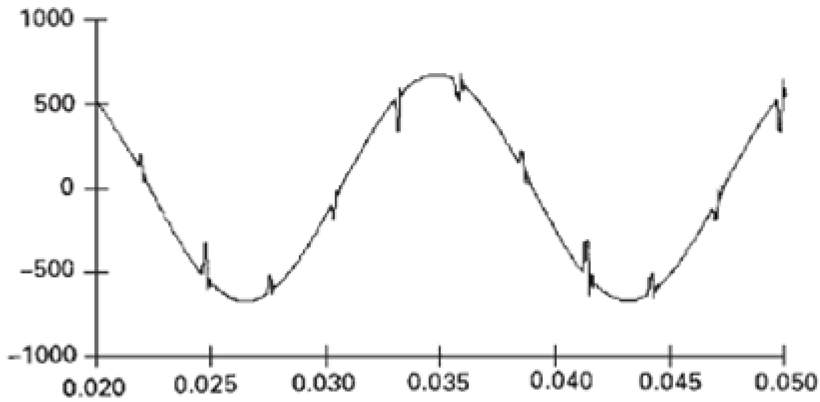

Figure 2.6 Example of voltage notching caused by converter operation [29]

2.2.5.4 <u>Noise</u>

Noise is defined as unwanted electrical signals with broadband spectral content lower than 200 kHz superimposed on the power system voltage or current in phase conductors, or found on neutral conductors or signal lines. Electric noise may result from faulty connections in transmission or distribution systems, arc furnaces, electrical furnaces, power electronic devices, control circuits, welding equipment, loads with solid-state rectifiers, improper grounding, turning off capacitor banks, adjustable-speed drives, corona, and broadband power line (BPL) communication circuits. The problem can be mitigated by using filters, line conditioners, and dedicated lines or transformers. Electric noise impacts electronic devices such as microcomputers and programmable controllers.

2.2.6 Voltage fluctuation and flicker

Voltage fluctuations are systemic variations of the voltage envelope or random voltage changes, the magnitude of which does not normally exceed specified voltage ranges (e.g., 0.9 to 1.1 pu as defined by ANSI C84.1-1982). Voltage fluctuations are divided into two categories:

- step-voltage changes, regular or irregular in time, and
- cyclic or random voltage changes produced by variations in the load impedances

Voltage fluctuations degrade the performance of the equipment and cause instability of the internal voltages and currents of electronic equipment. However, voltage fluctuations less than 10% do not affect electronic equipment. The main causes of voltage fluctuation are pulsed-power output, resistance welders, start-up of drives, arc furnaces, drives with rapidly changing loads, and rolling mills. Loads that can exhibit continuous, rapid variations in the load current magnitude can cause voltage variations that are often referred to as flicker.

2.2.7 Power frequency variations

Power frequency variations are defined as the deviation of the power system fundamental frequency from it specified nominal value (e.g., 50 or 60 Hz). The power system frequency is directly related to the rotational speed of the generators supplying the system. There are slight variations in frequency as the dynamic balance between load and generation changes. The size of the frequency shift and its duration depend on the load characteristics and the response of the generation control system to load changes.

3. HARMONICS

Harmonics are sinusoidal voltages or currents having frequencies that are integer multiples of the frequency at which the supply system is designed to operate (termed the fundamental frequency; usually 50 or 60 Hz) [29].

Harmonics according to their harmonic number will have frequencies:

$f_h = (h)$ x *fundamental frequency*

The component with *h* = 1 is called the fundamental component.

Using the Fourier series, any voltage or current waveform could be reproduced from the fundamental frequency component and the sum of the harmonic components as

$$V(t) = a_0 + \sum_{k=1}^{\infty} V_h \sin(h2\pi ft + \phi_h) \tag{3.1}$$

where,

a_0: dc component

V_h : peak voltage level

f : fundamental frequency

ϕ_h: phase angle

Figure 3.1 shows an ideal 50-Hz waveform with a peak value of 100, which can be taken as one per unit. Likewise, it also portrays waveforms of amplitudes (1/7), (1/5), and (1/3) per unit and frequencies seven, five, and three times the fundamental frequency, respectively. This behavior showing harmonic components of decreasing amplitude often following an inverse law with harmonic order is typical in power systems.

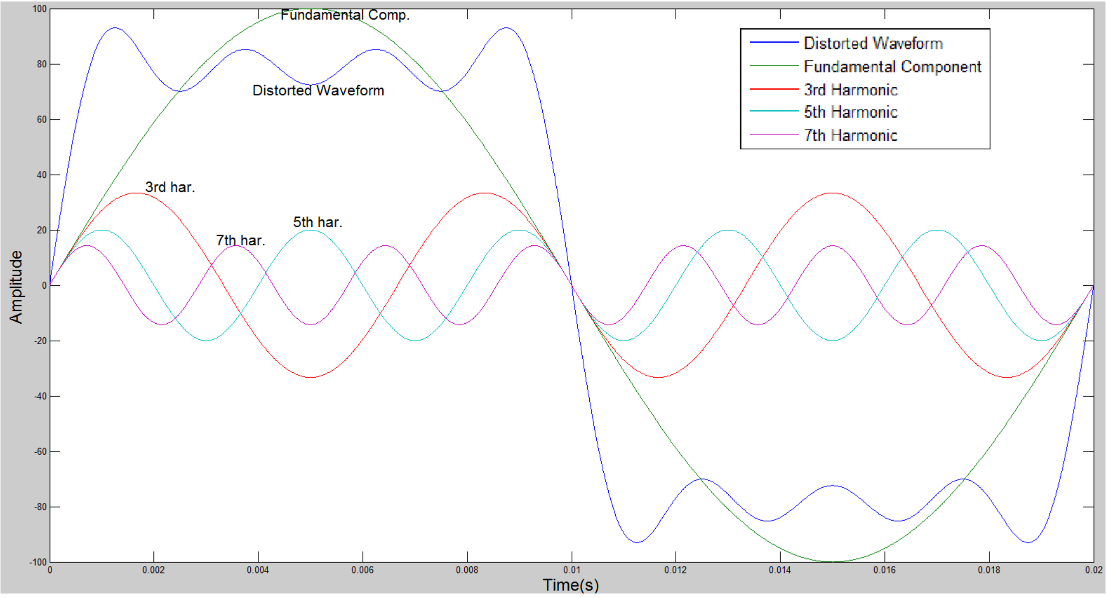

Figure 3.1 50 Hz-fundamental component, 3rd ,5th,7th harmonics and the resulting distorted waveform

3.1 Harmonic Distortion

Ideally, an electricity supply should invariably show a perfectly sinusoidal voltage signal at every customer location. However, for a number of reasons, utilities often find it hard to preserve such desirable conditions. Waveform distortion, due to the harmonics is named as harmonic distortion (figure 3.1) and it constitutes at present one of the main concerns for engineers in the several stages of energy utilization within the power industry [30]. The increasing use of nonlinear loads in industry is keeping harmonic distortion in distribution networks on the rise.

Nonlinear loads are loads in which the current waveform does not resemble the applied voltage waveform due to a number of reasons, for example, the use of electronic switches that conduct load current only during a fraction of the power frequency period. When a nonlinear load is supplied from a supply voltage of 60-Hz or 50-Hz frequency, it draws currents at more than one frequency, resulting in a distorted current waveform. The majority of nonlinear loads produce harmonics that are odd multiples of the fundamental frequency. Certain conditions need to exist for production of even harmonics [31].

Due to the power system impedance, any current (or voltage) harmonic will result in the generation and propagation of voltage (or current) harmonics and affects the entire power system. Figure 3.2 illustrates the impact of current harmonics generated by a nonlinear load on a typical power system with linear loads [1].

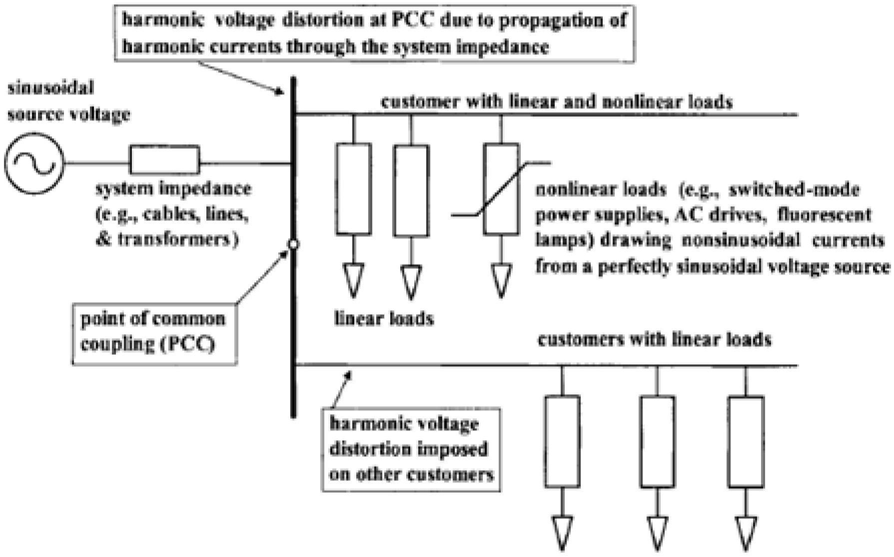

Figure 3.2 Propagation of harmonics (generated by a nonlinear load) in power systems [1]

3.2 Sources of Harmonics

Among the sources of harmonic voltages and currents in power systems three groups of equipment can be distinguished [32]:

- magnetic core equipment, like transformers, electric motors, generators, etc.
- arc furnaces, arc welders, high-pressure discharge lamps, etc.
- electronic and power electronic equipment.

3.2.1 Transformers

The relationship between the primary voltage and current – shown in Figure 3.3 as a magnetization curve – is strongly non-linear and hence its location within the saturation region causes distortion of the magnetizing current (Figure 3.3).

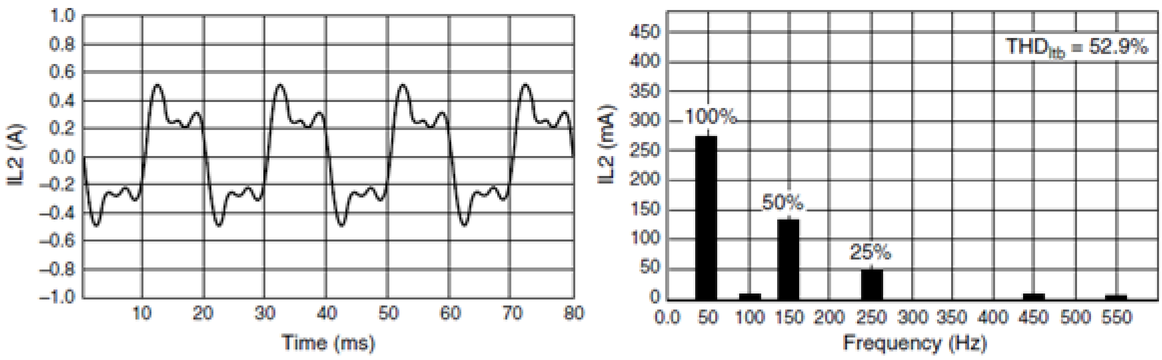

Figure 3.3 An example of a transformer-distorted magnetizing current and its harmonic spectrum [32]

3.2.2 Arc furnaces

Distortion of arc furnace currents and in consequence also of voltages is an important issue because of their common use and large individual powers [32]. Moreover, arc furnaces are presently operated at a lower power factor than in the past. One of the consequences of this, as well as more stringent requirements regarding reactive power compensation, is the increasing rated power of the compensating capacitors. This results in lowering the resonant frequency. As the amplitudes of high harmonics are of significant value in this range of the spectrum, a magnification of the supply voltage harmonics may occur. Conditions for arc discharge change in subsequent phases of the heat. The highest level of current distortion occurs during the melting phase, whereas it is much lower in the other phases (air refining and refining). A typical amplitude spectrum of the current (during the melting and refining) is shown in Figure 3.4.

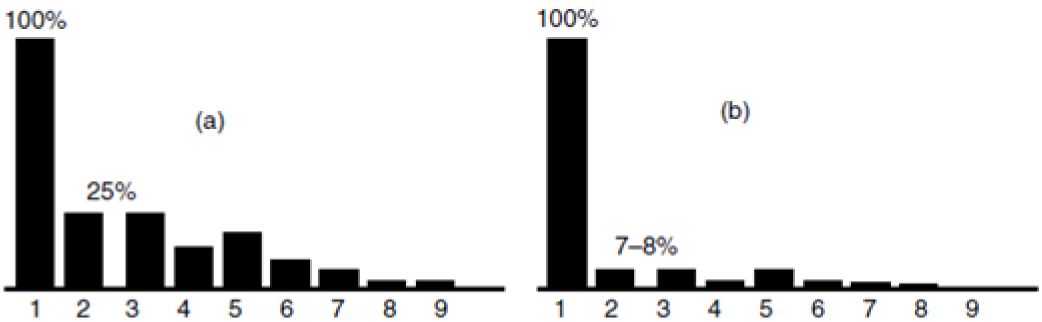

Figure 3.4 Typical harmonic spectrum of an arc furnace current (a) during melting and (b) during refining [32]

3.2.3 Rotating machines

The distribution of the armature windings and the presence of slots in the machines cause spatial harmonics in them [33]. These in turn produce time harmonics in the induced voltages, which appear at the terminals. Most of the power station generators are wye-connected. In such machines, triplen harmonic voltages do not appear in line-to-line voltages. Also, triplen harmonics can be eliminated even in phase-to-neutral voltages by using two-third pitch winding. Usually, the most significant harmonics to be minimized by the use of fractional pitch windings are the fifth and seventh. Higher harmonics than the ninth are so small that they require little attention except in rare cases.

3.2.4 Fluorescent lighting

Fluorescent lights are discharge lamps; thus they require a ballast to provide a high initial voltage to initiate the discharge for the electric current to flow between two electrodes in the fluorescent tube [25]. Once the discharge is established, the voltage decreases as the arc current increases. It is essentially a short circuit between the two electrodes, and the ballast has to quickly reduce the current to a level to maintain the specified lumen output. Thus, a ballast is also a current-limiting device in lighting applications.

3.2.5 Converters

The increasing use of the power conditioners in which parameters like voltage and frequency are varied to adapt to specific industrial and commercial processes has made power converters the most widespread source of harmonics in distribution systems [30]. Electronic switching helps the task to rectify 50-/60-Hz AC into DC power. In DC applications, the voltage is varied through adjusting the firing angle of the electronic switching device. Basically, in the rectifying process, current is allowed to pass through semiconductor devices during only a fraction of the fundamental frequency cycle, for which power converters are often regarded as energy-saving devices. If energy is to be used as AC but at a different frequency, the DC output from the converter is passed through an electronic switching inverter that brings the DC power back to AC. Large power converters like those used in the metal smelter industry and in HVDC transmission systems. Medium-size power converters like those used in the manufacturing industry for motor speed control and in the railway industry Small power rectifiers used in residential entertaining devices, including TV sets and personal computers. Battery chargers are another example of small power converters.

3.3 Effects of Harmonics

The undesirable effects of the harmonics produced by the aforementioned loads are listed as follows [33]:

1. Capacitors: These may draw excessive current and prematurely fail from increased dielectric loss and heating. Also, under resonance conditions, considerably higher voltages and currents can be observed than would be the case without resonance.
2. Power Cables: In systems with resonant conditions, cables may be subjected to voltage stress and corona, which can lead to dielectric (insulation) failure. Further harmonic currents can cause heating.
3. Telephone Interference: Harmonics can interfere with telecommunication systems, especially noise on telephone lines
4. Rotating Equipment (Motors and Generators): Harmonic voltages and currents contribute to increased copper and iron losses, leading to the heating of machines and thus reducing their efficiency
5. Transformers: Harmonic currents increase copper losses and stray load losses, and harmonic voltages cause an increase in iron losses. Higher frequency harmonics increase losses because they are dependent on frequency, but in general higher harmonics are smaller in magnitude. Further, harmonics are responsible for increased audible noise
6. Electronic Equipment: Computers and allied equipment such as programmable controllers frequently require ac sources that have no more than a 5% harmonic voltage distortion factor, with the largest single harmonic below 3% of the fundamental voltage. Higher levels of harmonics result in erratic functioning or

malfunctioning of the equipment. Hence, many medical instruments are provided with line conditioners

7. Metering: Induction disk devices, such as watthour meters, can give erroneous readings in systems with severe distortion.

8. Relaying: As with other equipment, switchgears also can experience increased losses due to harmonics.

3.4 Harmonic Analysis

3.4.1 Fourier series

In 1822 J.B.J. Fourier [34] postulated that any continuous function repetitive in an interval T can be represented by the summation of a d.c. component, a fundamental sinusoidal component and a series of higher-order sinusoidal components (called harmonics) at frequencies which are integer multiples of the fundamental frequency. Harmonic analysis is then the process of calculating the magnitudes and phases of the fundamental and higher-order harmonics of the periodic waveform [34]. The resulting series, known as the Fourier series, establishes a relationship between a time-domain function and that function in the frequency domain. In practice, data is often available in the form of a sampled time function, represented by a time series of amplitudes, separated by fixed time intervals of limited duration. When dealing with such data, a modification of the Fourier transform, the discrete Fourier transform (DFT), is used. The implementation of the DFT by means of the so-called Fast Fourier transform (FFT) forms the basis of most modern spectral and harmonic analysis systems [35].

By definition, a periodic function, $f(t)$, is that where $f(t) = f(t + T)$. This function can be represented by a trigonometric series of elements consisting of a DC component and other elements with frequencies comprising the fundamental component and its integer multiple frequencies. This applies if the following so-called Dirichlet conditions are met:

- If a discontinuous function, $f(t)$ has a finite number of discontinuities over the period T

- If $f(t)$ has a finite mean value over the period T
- If $f(t)$ has a finite number of positive and negative maximum values

The expression for the trigonometric series $f(t)$ is as follows:

$$g(t) = \frac{a_0}{2} + \sum_{h=1}^{\infty} \left[a_h \cos(hw_o t) + b_h \sin(hw_o t) \right] \tag{3.2}$$

where $w_o = 2\pi / T$

Eq.(3.2) can be simplified which yields:

$$g(t) = c_0 + \sum_{h=1}^{\infty} c_h \sin(hw_o t + \phi_h) \tag{3.3}$$

where

$c_o = \frac{a_0}{2}$, $c_h = \sqrt{a_h^{\ 2} + b_h^{\ 2}}$, and $\phi_h = \arctan(\frac{a_h}{b_h})$

Equation (3.2) is known as a Fourier series and it describes a periodic function made up of the contribution of sinusoidal functions of different frequencies.

(hw_o) h th order harmonic of the periodic function

c_0 magnitude of the DC component

c_h and ϕ_h magnitude and phase angle of the h th harmonic component

By Euler's equation, Equation (3.3) can be represented in a complex form as:

$$g(t) = \sum_{h=1}^{\infty} c_h e^{jhw_0 t} \tag{3.4}$$

where $h = 0, \mp 1, \mp 2, \ldots$

$$c_h = \frac{1}{T} \int_{-T/2}^{T/2} g(t) e^{-jhw_0 t} dt \tag{3.5}$$

3.4.1.1 Orthogonal functions

A set of functions, ϕ_i, defined in $a \le x \le b$ is called orthogonal (or unitary, if complex) if it satisfies the following condition:

$$\int_a^b \phi_i(x)\phi_j^*(x)dx = K_i\delta_{ij} \tag{3.6}$$

Where $\delta_{ij} = 1$ for $i = j$, and $\delta = 0$ for $i \ne j$, and * is the complex conjugate.

It can also be shown that the functions:

$$\{1, \cos(w_0t), ... \sin(w_ot), ..., \cos(hw_ot), ..., \sin(hw_0t), ... \} \tag{3.7}$$

for which the following conditions are valid:

$$\int_{-T/2}^{T/2} \cos kx \cos lx dx = \begin{cases} 0, k \ne l \\ \pi, k = l \end{cases} \tag{3.8}$$

$$\int_{-T/2}^{T/2} \sin kx \sin lx dx = \begin{cases} 0, k \ne l \\ \pi, k = l \end{cases} \tag{3.9}$$

$$\int_{-T/2}^{T/2} \cos kx \sin lx dx = 0 \quad (k = 1,2,3,\ldots), \tag{3.10}$$

$$\int_{-T/2}^{T/2} \cos kx dx = 0 \quad (k = 1,2,3,\ldots), \tag{3.11}$$

$$\int_{-T/2}^{T/2} \sin kx dx = 0 \quad (k = 1,2,3,\ldots), \tag{3.12}$$

$$\int_{-T/2}^{T/2} 1 dx = 2\pi \tag{3.13}$$

are a set of orthogonal functions. From equation (3.7) to equation (3.13), it is seen that the integral over period $(-\pi$ to $\pi)$ of the product of any two sine and cosine functions is zero.

3.4.1.2 Fourier coefficients

Integrating equation (3.2) and applying the orthogonal fuctions (equation 3.8 through equation 3.13), we obtain the Fourier coefficients as follows:

$$a_0 = \frac{2}{T}\int_{-T/2}^{T/2} g(t)dt, \qquad (3.14)$$

$$a_h = \frac{2}{T}\int_{-T/2}^{T/2} g(t)\cos(hw_0 t)dt, \text{ and,} \qquad (3.15)$$

$$b_h = \frac{2}{T}\int_{-T/2}^{T/2} g(t)\sin(hw_0 t)dt \qquad (3.16)$$

where $h = 1,2,..\infty$

3.4.2 Fourier transform

Fourier analysis, when applied to a continuous, periodic signal in the time domain, yields a series of discrete frequency components in the frequency domain. By allowing the integration period to extend to infinity, the spacing between the harmonic frequencies, w, tends to zero and the Fourier coefficients, c_n, of equation (3.5) become a continuous function, such that

$$G(f) = \int_{-\infty}^{\infty} g(t)e^{-j2f\pi T}dt \qquad (3.17)$$

The expression for the time domain function $x(t)$, which is also continuous and of infinite duration, in terms of $G(f)$ is then

$$g(t) = \int_{-\infty}^{\infty} G(f)e^{j2f\pi T}df \qquad (3.18)$$

$G(f)$ is known as the spectral density function of *x(t)*. In general, $G(f)$ is complex and can be written as

$$G(f) = \text{Re } G(f) + j \text{ Im } G(f) \tag{3.19}$$

The real part of $G(f)$ is obtained from

$$\text{Re } G(f) = \frac{1}{2}[G(f) + G(-f)] = \int_{-\infty}^{\infty} g(t)\cos 2\pi ftdt \tag{3.20}$$

Similarly, fort he imaginary part of $G(f)$

$$\text{Im } G(f) = \frac{1}{2} j[G(f) - G(-f)] = -\int_{-\infty}^{\infty} g(t)\sin 2\pi fTdt \tag{3.21}$$

The amplitude spectrum of the frequency signal is obtained from

$$|G(f)| = \left[(\text{Re}\, X(f))^2 + (\text{Im}\, G(f))^2\right]^{1/2} \tag{3.22}$$

The phase spectrum is

$$\phi(f) = \tan^{-1}\left[\frac{\text{Im}\, X(f)}{\text{Re}\, X(f)}\right] \tag{3.23}$$

Using equations (3.19) to (3.23), the inverse Fourier transform can be expressed in terms of the magnitude and phase spectra components:

$$g(t) = \int_{-\infty}^{\infty} |X(f)| \cos[2\pi fT - \phi(f)]df \tag{3.24}$$

3.4.3 Discrete Fourier transform

In the case where the frequency domain spectrum is a sampled function, as well asthe time domain function, we obtain a Fourier transform pair made up of discrete components:

$$G(f_k) = 1/N \sum_{n=0}^{N-1} g(t_n) e^{-j2\pi kn/N} \quad \text{and,} \tag{3.25}$$

$$g(t_n) = \sum_{k=0}^{N-1} G(f_k) e^{j2\pi kn/N} \tag{3.26}$$

Both the time domain function and the frequency domain spectrum are assumed periodic, with a total of *N* samples per period. It is in this discrete form that the Fourier transform is most suited to numerical evaluation by digital computation. If equation (3.25) is rewritten as

$$G(f_k) = 1/N \sum_{n=0}^{N-1} g(t_n) W^{kn} \tag{3.27}$$

where $W = e^{-j2\pi/N}$

Over all the frequency components, equation (3.27) becomes a matrix equation.

$$\begin{bmatrix} G(f_0) \\ G(f_1) \\ . \\ . \\ . \\ G(f_k) \\ . \\ . \\ . \\ G(f_{N-1}) \end{bmatrix} = 1/N \begin{bmatrix} 1 & 1 & . & . & . & 1 & . & . & 1 \\ 1 & W & . & . & . & W^k & . & . & W^{N-1} \\ . & . & . & & & & & & . \\ . & . & & . & & & & & . \\ . & . & & & . & & & & . \\ 1 & W^k & . & . & . & W^{k^2} & & & W^{k(N-1)} \\ . & . & & & & . & & & . \\ . & . & & & & . & & & . \\ 1 & W^{N-1} & . & . & . & W^{(N-1)k} & & & W^{(N-1)^2} \end{bmatrix} \cdot \begin{bmatrix} g(t_0) \\ g(t_1) \\ . \\ . \\ . \\ g(t_k) \\ . \\ . \\ g(t_{N-1}) \end{bmatrix} \tag{3.28}$$

or in a condensed form

$$[G(f_k)] = 1/N[W^{kn}][x(t_n)] \tag{3.29}$$

In these equations, $G(f_k)$ is a vector representing the N components of the function in the frequency domain, while [$x(tn)$] is a vector representing the N samples of the function in the time domain.

Calculation of the N frequency components from the N time samples, therefore, requires a total of N^2 complex multiplications to implement in the above form. Each element in the matrix [W^{kn}] represents a unit vector with a clockwise rotation of $2n\pi / N$ (n = 0,1,2,…,(N-1)) introduced between successive components. Depending on the value of N, a number of these elements are the same.

3.4.4 Fast Fourier transform

For large values of N, the computational time and cost of executing the N^2 complex multiplications of the DFT can become prohibitive. Instead, a calculation procedure known as the FFT, which takes advantage of the similarity of many of the elements in the matrix [W^{kn}], produces the same frequency components using only N/2 log2N multiplications to execute the solution of equation (3.16). Thus, for the case N = 1024 = 210, there is a saving in computation time by a factor of over 200. This is achieved by factorising the [W^{kn}] matrix of equation (3.27) into $\log_2 N$ individual or factor matrices such that there are only two non-zero elements in each row of these matrices, one of which is always unity.

Thus, when multiplying by any factor matrix only N operations are required.
The reduction in the number of multiplications required, to $(N/2)\log_2 N$, is obtained by recognising that

$$W^{N/2} = -W^0 \tag{3.30}$$

$$W^{(N+2)/2} = -W^1 \tag{3.31}$$

To obtain the factor matrices, it is first necessary to re-order the rows of the full matrix. If rows are denoted by a binary representation, then the re-ordering is by bit reversal.

3.4.5 Drawbacks of FFT based techniques in harmonic estimation

The Fourier transform based techniques are the most widely utilised signal processing tools in power system harmonic analysis. However, three problems, aliasing, leakage and the picket-fence effect are the main drawbacks of this approach.

3.4.5.1 Picket fence effect

If the analyzed waveform has frequencies which are integral numbers of the original record length T (or observation time), the FFT will yield the appropriate amplitudes at the appropriate frequencies and zero at the others. Thus, ideally, the sampling frequency is

$$f_s = \text{desired number of points x fundamental frequency component} \qquad (3.32)$$

The picket-fence effect occurs if the analyzed waveform includes a frequency which is not one of the discrete frequencies (an integer times the fundamental) [36]. A frequency lying between the nth and the (n + 1)st harmonics where (n + 1) < N/2, affects primarily the magnitudes of the nth and the (n + 1)st harmonics and secondarily the magnitude of all other harmonics. Also, this frequency can cause leakage which in turn may cause pseudoaliasing.

3.4.5.2 Aliasing

"Aliasing" is the phenomenon due to which high frequency components of a time function can translate into low frequencies if the sampling rate is too low. This is shown in figure by showing a relatively high frequency and relatively low frequency that share identical sample points. This uncertainty can be removed by demanding that sampling rate high enough for the highest frequency present to be sampled at least twice during each cycle [36]. Briefly, sampling theorem states that the sampling frequency must be at least twice the highest frequency contained in the original signal for a correct transfer of information to the sampled system.

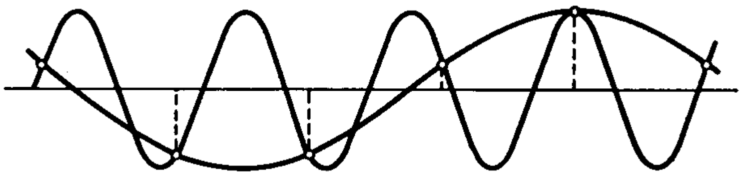

Figure 3.5 Aliasing Effect

3.4.5.3 Spectral leakage

For an accurate spectral measurement, it is not sufficient to use proper signal acquisition techniques to have a nicely scaled, single-sided spectrum. Spectral leakage is the result of an assumption in the FFT algorithm that the time record is exactly repeated throughout all time and that signals contained in a time record are thus periodic at intervals that correspond to the length of the time record. If the time record has a nonintegral number of cycles, this assumption is violated and spectral leakage occurs. Another way of looking at this case is that the nonintegral cycle frequency component of the signal does not correspond exactly to one of the spectrum frequency lines [37].

4. A HYBRID LEAST SQUARES-GA BASED ALGORITHM FOR HARMONIC ESTIMATION

4.1 Least Squares Method

Least Squares Method was first described by Carl Friedrich Gauss around 1794 [38]. Least Squares is a standard approach to minimize the sum of squared vertical distances between the observed responses in the dataset and the responses predicted by the linear approximation.

For the given data z(t) least square estimation can be written in the form:

$$z(t) = h_1(t)A_1 + h_2(t)A_2 + ... + h_n(t)A_n = \phi^T(t)A \tag{4.1}$$

where

z= observed variable

$a_1,\ldots, a_n$ = unknown parameters

$h_1,\ldots, h_n$ = known functions that may depend on other known variables

The variables $h_1,\ldots, h_n$ are called regression variables or regressors and the model of equation (4.1) is called a regression model, where

$$\phi^T(t) = [h_1(t) \quad h_2(t) \quad \cdots \quad h_n(t)] \tag{4.2}$$

$$A^T = [A_1 \quad A_2 \quad \cdots \quad A_n] \tag{4.3}$$

Equation (4.1) can bee written as

$$z(t) = \phi^T(t)_{1xn} A_{nx1} \tag{4.4}$$

If N measurements are taken then for the *k*th measurement the equation can be written as

$$z(k) = [h_1(k) \quad h_2(k) \quad \cdots \quad h_n(k)] \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \tag{4.5}$$

or $\quad Z = H.A \quad$ (4.6)

where Z = N x 1 vector and A = N x n matrix and $$H = \begin{bmatrix} h_1(1) & h_2(1) & \cdots & h_n(1) \\ h_1(2) & h_1(2) & \cdots & h_n(2) \\ \vdots & \vdots & \vdots & \vdots \\ h_1(N) & h_2(N) & \cdots & h_n(N) \end{bmatrix} \tag{4.7}$$

$$H = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ \vdots \\ \phi^T(N) \end{bmatrix} \tag{4.8}$$

To obtain the least squares estimate minimizing *J*

$$J = \frac{1}{N}\sum_{k=1}^{N} e^2(k) = \frac{1}{N}\sum_{k=1}^{N}(z(k) - H^T(k).A)^2 \tag{4.9}$$

becomes in vector form as

$$J = \frac{1}{N}(Z - H\hat{A})^T(Z - H\hat{A}) = \frac{1}{N}[Z^T Z - Z^T H\hat{A} - \hat{A}^T H^T Z + \hat{A}^T H^T H\hat{A}] \tag{4.10}$$

$$(Z^T H\hat{A})^T = \hat{A}H^T Z = Z^T H\hat{A} \tag{4.11}$$

Therefore,

$$J = \frac{1}{N}\left[Z^T Z - \hat{A}^T H^T Z - \hat{A}H^T Z + \hat{A}^T H^T H\hat{A}\right] \tag{4.12}$$

Setting to zero the derivative of *J* with respect to $\hat{A}$

$$\frac{\partial Z}{\partial \hat{A}} = -2H^T Z + 2H^T H\hat{A} = 0 \tag{4.13}$$

$$H^T Z = H^T H\hat{A} \tag{4.14}$$

Finally becomes

$$\hat{A}, = (H^T H)^{-1} H^T Z \tag{4.15}$$

4.2 Genetic Algorithm

4.2.1 Inspiration

Genetic algorithms are a class of evolutionary algorithms inspired by the process of natural evolution.The idea behind the algorithm lies on the neo-Darwinian paradigm which has been composed of Darwin's classical theory of evolution with Weismann's theory of natural selection and Mendel's concept of genetics.[39] Neo-Darwinism is based on processes of reproduction, mutation, competition and selection. Evolution can be seen as a set of these processes leading to the maintenance or increase of a population's ability, called evolutionary fitness, to survive and reproduce in a specific environment resulting in better generations.[40] The idea of GA as an simulation of the natural evolution dates back to the early 1950s, but it was only later, Holland, one of the contributors in foundation of evolutionary computation introduced the methodology of genetic algorithms in a more formal and tractable way [20].

Genetic algorithms can be represented by a sequence of procedural steps for moving from one population of artificial chromosomes to a new population. Population consists of artificial chromosomes, also named as individuals, which are candidate solutions for optimization problem and each chromosome is encoded in variable domain as a number of genes represented by 0 or 1. GA measures the chromosome's fitness by using an evaluation function to carry out reproduction. In the reproduction process, the crossover operator exchanges parts of two single chromosomes, and mutation operator changes the gene value in some randomly chosen location of the chromosome and a new generation takes place. As a result, after a number of successive reproductions, the less fit chromosomes become extinct, while those bests able to survive gradually come to dominate the population.

The main attractions of GAs lies in the fact that unlike the gradient based methods, do not require the calculation of the derivatives and can effectively explore many regions of the search space simultaneously, rather than a single region.

4.2.2 Algorithm

Steps in the GA are numbered and the details explained as below [42]

Step 1: The problem variable domain is represented as a chromosome of a fixed length, the size of a chromosome population N, the crossover probability p_c and the mutation probability p_m are choosen

Step 2: To measure the performance, or fitness, of an individual chromosome, a fitness function is defined in the problem domain

Step 3: An initial population of chromosomes of size N is generated

Step 4: Fitness of each individual chromosome is calculated

Step 5: A pair of chromosomes is selected for mating from the current population

Step 6: By applying genetic operators-crossover and mutation, a pair of offspring chromosomes is created

Step 7: The created offspring chromosomes is placed in the new population

Step 8: Step 5 is repeated until the size of the new chromosome population becomes equal to the size of the initial population, N

Step 9: The initial (parent) chromosome population is replaced with the new (offspring) population

Step 10: To step 4 is returned and the process is repeated until the termination criterion is satisfied

The flowchart of Genetic Algorithm is given in figure 4.1.

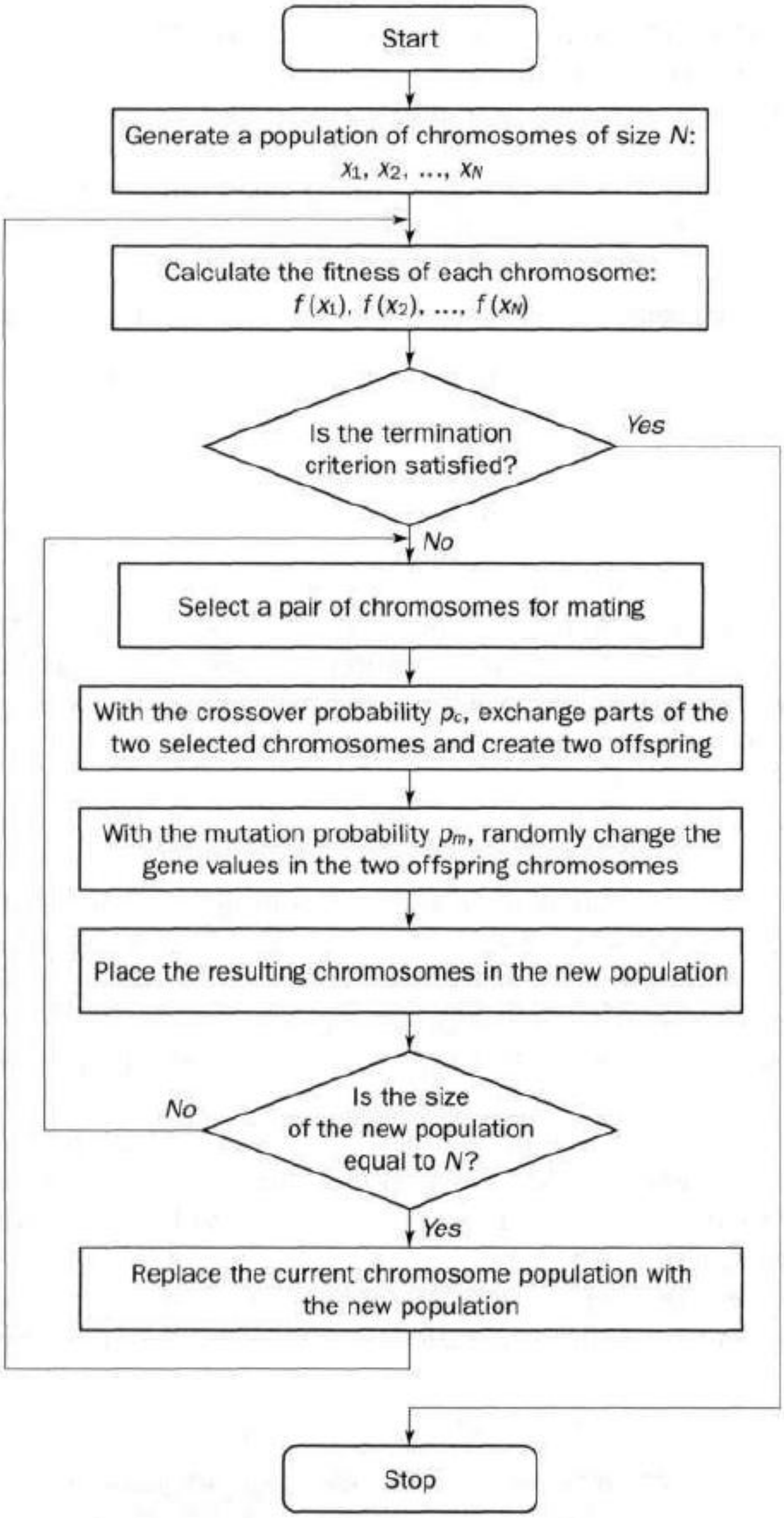

Figure 4.1 Flowchart of Genetic Algorithm [42]

4.2.2.1 Encoding

Typically, GAs encode a continuous parameter, x, as a n integer string of q bits, a_k, k=0,1,…q-1, each of which is a coefficient for a power of 2:

$$x = b_L + \frac{(b_u - b_L)}{2^q - 1} . \sum_{k=0}^{q-1} a_k 2^k \tag{4.16}$$

When decoded, integers are normalized by a factor of 2^{q-1} and multiplied by a b_U-b_L so that values span the range between a parameters's upper and lower bounds, b_U and b_L , respectively. Assuming that equal resources are devoted to each parameter, a vector of D parameters will require l=$q.D$ bits in all.

4.2.2.2 Selection

An initial population of individual structures $P(0)$ is generated (usually randomly) and each individual is evaluated for fitness [43]. Then some of these individuals are selected for mating and copied to the mating buffer $C(t)$. In Holland's original GA, individuals are chosen for mating probabilistically, assigning each individual a probability proportional to its observed performance. The probability of an individual being chosen is a function of its observed fitness. A straightforward way of doing this would be to total the fitness values assigned to all the individuals in the parent population and calculate the probability of any individual being selected by dividing its fitness by the total fitness. Thus, better individuals are given more opportunities to produce offspring. This, one of the most commonly used chromosome selection techniques is called roulette wheel selection [44]. In this technique chromosomes occupy slices in wheel, which have areas proportional to to their fitness ratios. Once a pair of parent chromosomes is selected, the crossover operator is applied. As a result individuals are selected for reproduction on the basis of their fitness,i.e, fitter chromosomes occupy larger areas and hence have the highest likelihood of selection for reproduction. Tournament selection and elitist strategy are the other popular methods for selection.

4.2.2.3 Reproduction

Crossover

The idea behind crossover can be stated as follows: given two individuals who are highly fit, ideally aim is to create a new individual that combines the best features from each. Crossover operates in a way that recombines features at random. It treats these features as building blocks scattered throughout the population and tries to recombine them into better individuals via crossover. Sometimes crossover may combine the worst features from the two parents, in which case these children will not survive for long. But sometimes it will recombine the best features from two good individuals, creating even better individuals, provided these features are compatible. After encoding phase, the representation becomes the classical bitstring representation: individual solutions in the population are represented by binary strings of zeros and ones of length *L*. A GA creates new individuals via crossover by choosing two strings from the parent population, lining them up, and then creating two new individuals by choosing a crossover point where two parent chromosomes break and then exchanging the chromosome parts after that point. As a result, two new offspring are created. Generally a value of 0.7 for the crossover probability is chosen. Crossover operation is shown in figure 4.2.

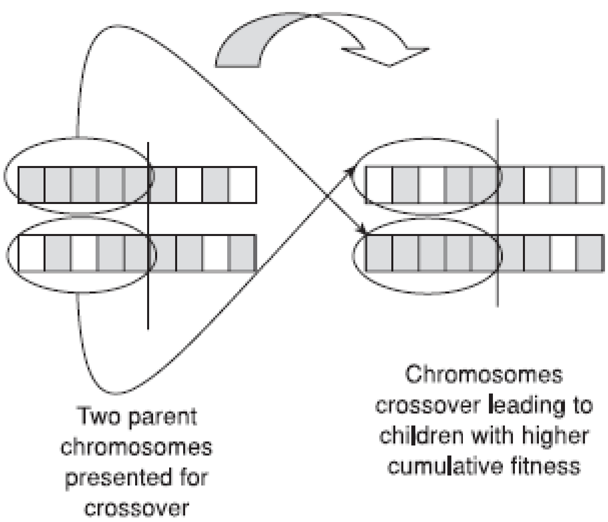

Figure 4.2 Crossover in GA [45]

Mutation

The sequence of selection and crossover operations may stagnate at any homogeneous set of solutions. Under such conditions, all chromosomes are identical, and thus the average fitness of the population cannot be improved. However, the solution might appear to become optimal, or rather locally optimal, only because the search algorithm is not able to proceed any further. Mutation operator's is equivalent to a random search and its role is to gurantee that the search algorithm is not trapped on a local optimum by preventing loss of genetic diversity. The mutation probability is quite low for GAs, typically in the range between 0.001 and 0.01 [42]. The mutation operation is shown in figure 4.3.

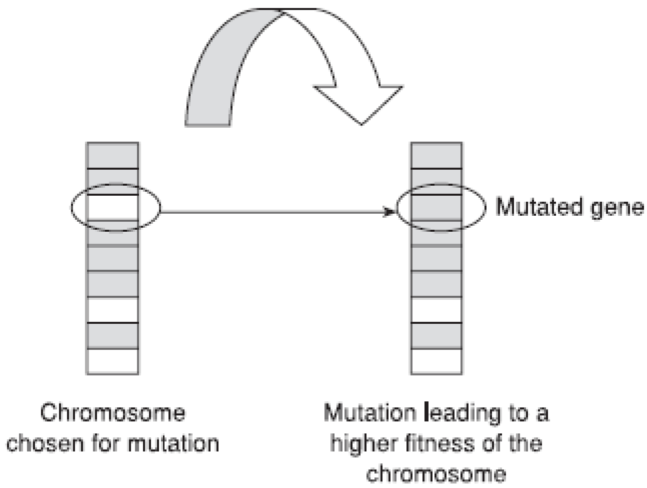

Figure 4.3 Mutation Operation in GA [45]

4.3 A Hybrid Least Squares – GA Based Algorithm For Harmonic Estimation

A signal composed of harmonics is mathematically described as

$$Z(t) = \sum_{n=1}^{N} A_n \sin(w_n t + \phi_n) + v(t) \qquad (4.17)$$

where n=1,2,…,N harmonics represents the order of the harmonic; A_n, w_n, Φ_n are the amplitude, phase angle and angular frequency of the *nt*h harmonic respectively, $w_n = 2\pi f_n$;and v(t) is the additive noise.

The estimated signal model is

$$\hat{Z}(t) = \sum_{n=1}^{N} \hat{A}_n \sin(w_n t + \hat{\phi}_n) \tag{4.18}$$

where $\hat{A}_n$, $\hat{\phi}_n$ are the estimation of A_n, ϕ_n respectively. Thereby, the original signal can be represented as

$$Z(t) = \hat{Z}(t) + r(t) = \sum_{n=1}^{N} \hat{A}_n \sin(w_n t + \hat{\phi}_n) + r(t) \tag{4.19}$$

where r(t) is the estimation residual indicating difference between $Z(t)$ and $\hat{Z}(t)$.

It can be seen that nonlinearity is due to the phase of the sinusoids. However signal is linear in amplitude. A hybrid algorithm proposed by Bettayeb and Qidwai iterating between GA based phase estimation and Least Squares based amplitude estimation is applied in [18]. Once the phases are estimated by GA based estimator, LS method is used for the estimation of amplitudes.

The resulting sampled linear model with K samples of the system with additive noise is given by

$$Z(k) = H(k)A + v(k),\ k = 1,2,\ldots,\mathrm{K} \tag{4.20}$$

where Z(k) is the *k*th sample of the measured values with additive noise *v*(k); $A = [A_1\ A_2 \ldots A_N]^T$ is the vector of the amplitudes to be estimated; H(k) is the system structure matrix given by:

$$H(k) = \begin{bmatrix} \sin(w_1 t_1 + \phi_1) & \sin(w_2 t_1 + \phi_2) & \cdots & \sin(w_n t_1 + \phi_n) \\ \sin(w_1 t_2 + \phi_1) & \sin(w_2 t_2 + \phi_2) & \cdots & \sin(w_n t_2 + \phi_n) \\ \vdots & \vdots & \vdots & \vdots \\ \sin(w_1 t_k + \phi_1) & \sin(w_2 t_k + \phi_2) & \cdots & \sin(w_n t_k + \phi_n) \\ \vdots & \vdots & \vdots & \vdots \\ \sin(w_1 t_K + \phi_1) & \sin(w_2 t_K + \phi_2) & \cdots & \sin(w_n t_K + \phi_n) \end{bmatrix} \tag{4.21}$$

where the phases are estimated from the last GA iteration. The estimation model for this system is

$$\hat{Z}(k) = \hat{H}(k)\hat{A} \tag{4.22}$$

The estimate $\hat{A}$ for the required parameter vector A can be obtained by minimizing the objective function by differentiating with respect to A and setting it to zero, which gives LS estimation algorithm

$$\hat{A}_{LS} = [H^T(k)H(k)]^{-1} H^T(k) Z(k) \tag{4.23}$$

Using the structure matrix with the sampled values constituting $Z(k)$ would give estimates for the amplitudes, which ensures that the estimation of the signal in equation (4.22) is the best in the condition of $\hat{\phi}_n$ .However, the values of $\hat{\phi}_n$ are not the best solution and need to be optimized. Therefore, in the next iteration $\hat{\phi}_n$ is updated using a certain optimization algorithm according to the cost function *J,* which is defined as the total square error between actual sample values and the estimated values of the signal and for the *i*th chromosome is given by

$$J(i) = \sum_{k=1}^{K} \left[Z(k) - \hat{Z}(k) \right]^2 \tag{4.24}$$

The performance index for the GA is calculated using both amplitudes and phases and the cycle is repeated until maximum number of generations reached or convergence condition satisfied.

5. A HYBRID LEAST SQUARES-PSOPC BASED ALGORITHM FOR HARMONIC ESTIMATION

5.1 Particle Swarm Optimization

5.1.1 Inspiration

Particle Swarm Optimization is a population-based stochastic optimization algorithm discovered by Kennedy and Eberhart [46] in 1995. The inspiring concept was the the social model of bird flocking or fish schooling which has collective behaviors of simple individuals interacting with their environment and each other. The theory that individual members of the school can profit from the discoveries and previous experience of all other members of the school during the search of food, suggesting that social sharing of information among conspeciates offers an evolutionary advantage was fundamental to the development of PSO. By syntesizing the cognitive human social behaviour with this hypothesis the PSO model is based on the two following factor

i. The autobiographical memory, which remembers the best previous position of each individual P_i in the swarm
ii. The publicized knowledge, which is the best solution P_g found currently by the population

5.1.2 Algorithm

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithm [47]. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed while it is effective in finding the global optimal solution [48] .In the last years PSO has been applied in many areas such as function optimization, artificial neural network training, fuzzy system control, and other areas where evolutionary computation algorithms can be applied [49].

The *i*th particle at the *k*th iteration has the following two attributes

1) A current position in N dimensional search space
   $X_i^k = (x_1^k, ..., x_n^k, ..., x_N^k)$

where $x_n^k \in [l_n, u_n], 1 \leq n \leq N, l_n$ and u_n are the lower and upper bound fort he *n*th dimension, respectively.

2) A current velocity $V_i^k = v_1^k, \ldots, v_n^k, \ldots, v_N^k$ which is bounded by a maximum velocity $V_{\max}^k = (v_{\max,1}^k, \ldots, v_{\max,2}^k, \ldots, v_{\max,N}^k)$ and a minimum velocity $V_{\min}^k = (v_{\min,1}^k, \ldots, v_{\min,2}^k, \ldots, v_{\min,N}^k)$

If the sum of accelerations would cause the velocity on that dimension to exceed $V_{\max}$,which is a parameter defined by user, then the velocity on that dimension is limited to $V_{\max}$. It determines the resolution, or fineness, with which regions between the present solution and the target(best so far) position are searched. If $V_{\max}$ is too high, particles might flt past good solutions or if $V_{\max}$ is too small, on the hand, particles may not explore sufficiently beyond locally good regions, could become trapped in local optima, unable to move far enough to reach a better position in problem domain [50].

In each iteration of PSO, the swarm is updated by the following equations:

$$V_i^{k+1} = V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^{k)}) \quad (5.1)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (5.2)$$

where P_i is the best previous position of the *i*th particle(also known as *p*best) and r_1 and r_2 are random elements from two uniform sequences in the range (0,1).

The acceleration constants c_1 and c_2 in equation (5.1), also named as cognitive and social parameters respectively, represent the weighting of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions [50]. Small values limit the movement of the particles, while large numbers may cause the particles to diverge.

The effect of considering a random value for acceleration constant helps to create an uneven cycling for the trajectory of the particle when it is searching around the optimal value. Since the acceleration parameter controls the strength of terms,
a small value will lead to a weak effect; therefore, the particles will follow a wide path and they will be pulled back only after a large number of iterations. If the acceleration constant is too high then the steps will be limited by $V_{\max}$ [54].

According to the different definitions of P_g, there are two different versions of PSO [46].

- If P_g is the best position among all the particles in the swarm(also known as *g*best), such a version is called global version
- If P_g is taken from some smaller number of adjacent particles of the population (also known as *l*best), such a version is called the local version

The global version maintains only a single best solution, and each particle moves towards its previous best position and towards the best particle in the whole swarm. The best particle acts as an attractor, pulling all the particles towards it. In the local version, each particle moves towards its previous best position, and also towards the best particle in its restricted neighbourhood and thus maintains multiple attractors. A sub-set of particles is defined for each particle from which the local best particle is selected. Global version is actually a special case of local version in which neighbourhood size is equal to swarm size. P_i and P_g are given by the following equations, respectively:

$$P_i = \begin{Bmatrix} P_i; & if & J(X_i) \geq P_i \\ X_i; & if & J(X_i) \leq P_i \end{Bmatrix} \tag{5.3}$$

$$P_g \in \{P_0, P_1, \ldots, P_m\} \big| J(P_g) = \min(J(P_0), J(P_1), \ldots, J(P_m)) \tag{5.4}$$

where J is the objective function $m \leq M$ and M is the total number of particles.

The standard PSO algorithm with inertia weight is given below and the flowchart is shown in figure 5.1.

Step1: Initialize

(a) Set constants k_{max}, c_1, c_2, v^0_{max}, initialize inertia weight *w* and counter *k*

(b) Randomly initialize particle positions $x_i^0 \in D$ in R^n for *i* = 1,…, *p*

(c) Randomly initialize particle velocities $0 \leq v_i^0 \leq v_{\max}^0$ for *i* = 1,…, *p*

(d) Evaluate fitness values f_i^0 for *i* = 1,…, *p*

(e) Set f_{i_best} for *i*=1,…, *p*. Set f_{g_best} to best f_{i_best}

Step2: Optimize

(a) Update particle velocity vector v_i^{k+1} using equation (5.1)

(b) Update particle velocity vector x_i^{k+1} using equation (5.2)

(c) Evaluate f_i^{k+1} is evaluated using particles' space coordinates x_i^{k+1} for *i* = 1,…,*p*

(d) If $f_i^{k+1} \leq f_{i\_best}$ then $f_{i\_best} = f_i^{k+1}, p_i^{k+1} = x_i^k$ for *i* = 1,…, *p*

(e) If $f_i^{k+1} \leq f_{g\_best}$ then $f_{g\_best} = f_i^{k+1}, p_{g\_best}^{k+1} = x_i^k$ for *i* = 1,…, *p*

(f) If stopping condition is satisfied step3 is processed

(g) All particle velocities v_i^k for i = 1,…, *p* are updated

(h) All particle positions x_i^k for i = 1,…, *p* are updated

(i) If $k = k_{max}$

(j) Increment *k*

(k) Update *w*

(l) Go to 2a

Step3: Terminate

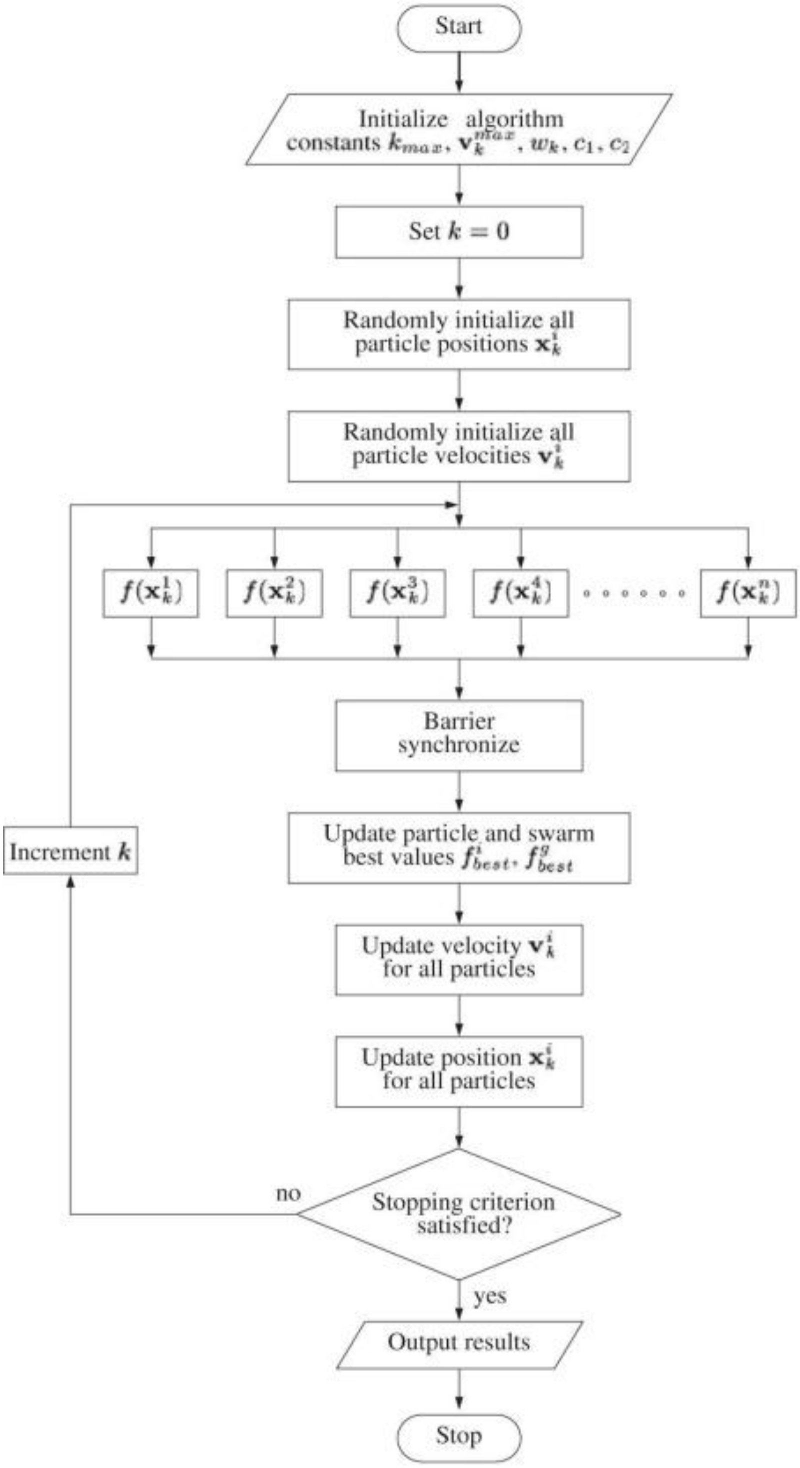

Figure 5.1 Flowchart of PSO Algorithm [55]

5.1.3 A modified particle swarm optimizer

The maximum velocity $V_{\max}$ serves as a constraint to control the global exploration ability of a particle swarm. As a larger $V_{\max}$ facilitates global exploration, a smaller $V_{\max}$ encourages local exploitation.Empirical studies performed on PSO indicate that even when the maximum velocity and acceleration constants are correctly defined, the particles may still diverge, i.e., go to infinity; a phenomena known as "explosion" of the swarm. Two methods are proposed in the literature in order to control this "explosion".

The inertia weight *w*, is introduced in [51], in order to govern how much of the previous velocity should be retained from the previous time step. The inertia weight [51] was set to a constant value, typically in the range of [0.8,1]. A larger inertia weight facilitates global exploration and a smaller inertia wieght tends to facilitate local exploration to fine-tune the current search area. The equation (5.1) with inertia weight becomes

$$V_i^{k+1} = wV_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \tag{5.5}$$

The position update rule remains same as in equation (5.2).

Generally the inertia weight is not kept fixed, but is varied as the algorithm progresses, so as to improve performance [50]. Commonly, a linearly decreasing inertia weight (first introduced by Shi and Eberhart [52], [53]) has produced good results in many applications [54]; however, the main disadvantage of this method is that once the inertia weight is decreased, the swarm loses its ability to search new areas because it is not able to recover its exploration model.

5.2 Particle Swarm Optimization With Passive Congregation

Congregation, is a grouping by social forces, that is the source of attraction is the group itself. Congregation can be classified into passive congregation and social congregation. Passive congregation is an attraction of an individual to other group members but where there is no display of social behavior. Social congregations usually happen in a group where the members are related (sometimes highly related). A variety of inter-individual behaviors are displayed in social congregations, necessitating active information transfer [56]. For example, ants use antennal contacts to transfer information about individual identity or location of resources [57].

In aggregation, which is different from congregation group members can react without direct detection of an incoming signals from the environment, because they can get necessary information from their neighbors [56]. Individuals need to monitor both environment and their immediate surroundings, such as the bearing and speed of their neighbors [56]. Therefore, each individual in an aggregation has a multitude of potential information from other group members that may minimize the chance of missed detection and incorrect interpretations [56]. He et al. [58] proposed that *s*uch information transfer can be employed in the model of passive congregation. Inspired by this, and to keep the model simple and uniform with SPSO, He et al.[58] proposed *a* variant of PSO with passive congregation: in which information can be transferred among individuals of the swarm. The equation (5.5), with proposed model introducing passive congregation becomes

$$V_i^{k+1} = wV_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^{k)} + c_3 r_3 (R_i^k - X_i^k) \tag{5.6}$$

where R_i is a particle selected randomly from the swarm, c_3 the passive congregation coefficient, and r_3 is a uniform random sequence in the range (0,1). The position update is the same as in equation (5.2).

5.3 A Hybrid Least Squares – PSOPC Based Algorithm For Harmonic Estimation

In [26] Lu et. al presented a new algorithm which utilizes the particle swarm optimizer with passive congregation (PSOPC) to estimate the phases of the harmonics, alongside a least-square (LS) method that is used to estimate the amplitudes. Following a similar procedure as in the algorithm introduced in [18], the new algorithm applies a hybrid method iterating between particle swarm optimizer with passive congregation (PSOPC)-based phase estimation and LS-based amplitude estimation.

Amplitude estimation process is performed as explained in chapter 4, utilizing least squares method through equations (4.20) and (4.23). However, in phase estimation stage instead of GA, PSO with passive conregation is used.

Using PSOPC as the optimization scheme, the procedure of the proposed algorithm is depicted as follows

To implement the algorithm, a number of parameters should be initialized, including *N* as the number of phases to be estimated; *P* as the size of the swarm used in PSOPC; *M* as a predefined number of iterations indicating the maximum generation; and ε the cost function.

In the first iteration, the values of $\hat{\phi}_n^1 (n = 1,2,..., N)$ are randomly selected. Consequently, $\hat{H}(k)$ is calculated according to equation (4.21). Afterwards, the amplitude $\hat{A}^1$ is estimated by equation (4.23). Finally, the signal $\hat{Z}^1(k)$ is reconstructed from $\hat{\phi}_n^1$ and $\hat{A}^1$ by the equation (4.22). The error between $\hat{Z}^1(k)$ and $Z\ (k)$ is calculated according to the cost function given in equation (4.24) to direct the search in the next iteration. In the *m*th iteration, the position of the *i*th particle is denoted as $X_i^m = (\hat{\phi}_{i1}^m, \hat{\phi}_{i2}^m,..., \hat{\phi}_{iN}^m)$

In the *(m+1)*th iteration, the particle X_i^{m-1} is updated according to the equation (5.2), which is calculated from $\hat{\phi}_n^m$ and $\hat{A}^m$. The process repeats until the maximum iteration M is reached or the condition $J < \varepsilon$ is satisfied. The detailed pseudocode for the proposed algorithm is listed in table 5.1 and a flowchart is presented in figure 5.2 to describe the estimation process.

Table 5.1 Pseudo Code for LS-PSOPC based algorithm

```
Set k:=1;
Randomly initialize positions and velocities of all particles  Xi=[Φi1, Φi2,…, ΦiN]
 WHILE (the termination conditions are not met)
   FOR (each particle i in the swarm)
     Calculate amplitudes: Calculate the amplitudes of each harmonic , Â^k = [Â1^k ,.., ÂN^k ];
                           using the LS method given by equation(4.20)
     Calculate fitness:    Calculate the fitness value of current particle J(Xi) using equation (4.21)
     Update pbest:         Compare the fitness value of pbest with J (Xi).  If  J(Xi) is better than
                           the fitness value of pbest, then set pbest to  the current position Xi;
     Update gbest:         Find the global best position of the swarm. If J (Xi) is better than the
                           fitness  value of gbest, then gbest is set to the position of the current
                           particle Xi;
         Update Ri:        Randomly select a particle from the swarm as Ri,
         Update velocities: Calculate velocities Vi using Eq.( 5.6). If Vi > Vmax  then Vi = Vmax.
                           If Vi < Vmin then  Vi = Vmin
         Update positions: Calculate positions Xi using equation (5.6)
      END FOR
    Set k:=k+1;
 END WHILE
```

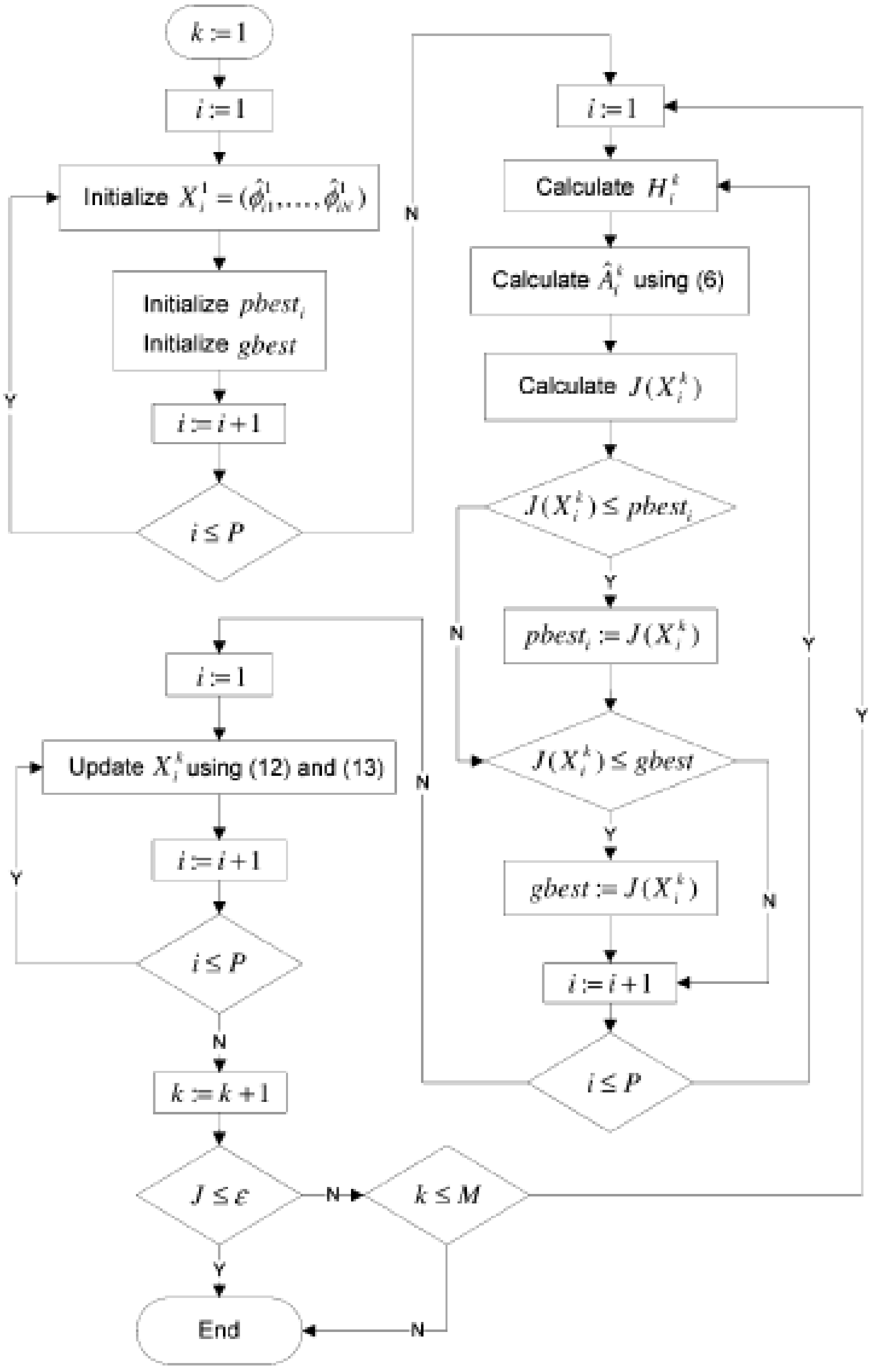

Figure 5.2 Harmonic Estimation Process by PSO[19]

6. A HYBRID LEAST SQUARES-DE BASED ALGORITHM FOR HARMONIC ESTIMATION

Differential evolution (DE) is a simple yet powerful evolutionary algorithm (EA) for global optimization presented by Price and Storn firstly in a technical report [59]. DE proved its performance on benchmark functions in the first contest on evolutionary computation in 1996 [60]. Price and Storn [61;62], by the extensive empirical evidence of DE's robust performance on a wide variety of test functions, introduced DE to a large international audience and many other researchers in optimization became aware of DE's potential. In the last years, there is an increasing interest in differential evolution. DE has been widely applied to many optimization problems and great number of differential evolution publications in scientific journals and conference proceedings are seen [23].

DE has shown good performance on many real-world problems, and on the majority of the numerical benchmark problems as well. In comparative studies DE has proven its superiority as the best performing algorithm over other evolutionary algorithms for many problems[24].

The differential Evolution method briefly consists of three basic steps:

(i) Generation of (large enough) population with N individuals [$x = (x_1, x_2, \ldots, x_m)$] in the m-dimensional space, randomly distributed over the entire domain of the function in question and evaluation of the individuals of the so generated by finding f(x);
(ii) Replacement of this current population by a better fit new population, and
(iii) Repetition of this replacement until satisfactory results are obtained or certain criteria of termination is met [23].

6.1 Population Structure

The current population, symbolized by P_x is composed of vectors, $x_{i,g}$ as

$$P_{x,g} = (x_{i,g}) \quad i = 1,\ldots\ldots,Np,\ g = 1,\ldots,gmax, \tag{6.1}$$

$x_{i,g} = (x_{j,i,g})$, j = 0,1,…,D-1. (6.2)

where *Np* denotes the number of population vectors, *g* defines the generation counter , and *D* the dimensionality, i.e. the number of parameters.

Once initialized,DE mutates randomly chosen vectors to produce an intermediary population

$P_{v,g} = (v_{i,g})$, i= 1,…,Np, g = 1,…, $gmax$, (6.3)

$v_{i,g} = (v_{j,i,g})$, j = 1,…,D (6.4)

Each vector ,n the current population is then recombined with a mutant to produce a trial population, P_u, of *Np* trial vectors, $u_{i,g}$

$P_{u,g} = (u_{i,g})$, i=1,…,Np, g=1,…, g_{max}, (6.5)

$u_{i,g} = (u_{j,i,g})$,j=1,...,D (6.6)

During recombination, trial vectors overwrite the mutant population, so a single array can hold both populations.

6.2 Initialization

Before the population can be initialized, both upper and lower bounds for each parameter must be specified. These 2*D* values can be collected into two, *D*-dimensional initialization vectors, b_L and b_U, for which subscripts L and U indicate the lower and upper bounds of the parameter vectors $x_{j,i}$, respectively. Once initialization bounds have been specified, random number generator, $rand_j[0,1)$, returns a uniformly distributed random number from within range [0,1),i.e., $0 \le rand_j[0,1) < 1$ and assigns each parameter of every vector a value from within the prescribed range.The initialization of the population is realized via

$x_{j,i,0} = rand_j[0,1).(b_{j,U} - b_{j,L}) + b_{j,L}$ (6.7)

6.3 Mutation

Once initialized, DE mutates and recombines the population to produce a population of *Np* trial vectors. In particular, differential mutation adds a scaled, randomly sampled, vector difference to a third vector. Equation (6.8) shows how to combine three different, randomly chosen vectors to create a mutant vector,

$$v_{i,g} = x_{r0,g} + F.(x_{r1,g} - x_{r2,g}) \tag{6.8}$$

The scale factor, $F \in (0,1+)$, is a positive real number that controls the rate at which the population evolves. In classic DE base vector indexes, r_0, r_1, r_2 are randomly chosen vectors index that are different from each others and the target vector index, *i*. In figure 6.1, differential mutation is shown: the weighted differential, $F(x_{r1,g} - x_{r2,g})$ is added to the base vector, $x_{r0,g}$, to produce mutant $v_{i,g}$.

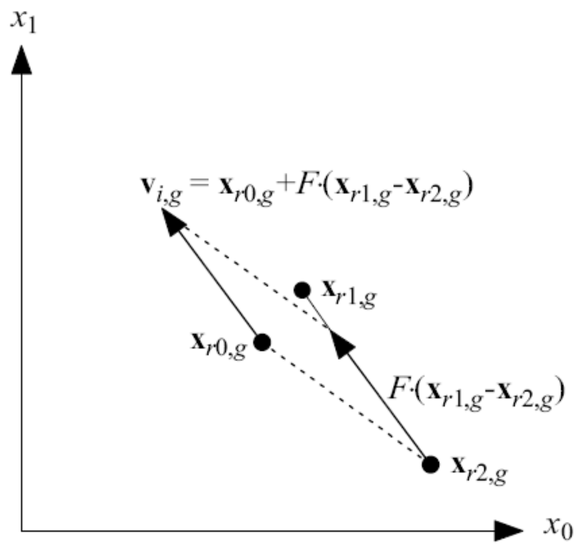

Figure 6.1 Differential Mutation

There are variants of DE represented by the notation[1] DE/x/y/z where x denotes the base vector, y denotes the number of difference vectors used, and z representing the crossover method. The classic DE version which is modeled through Eq.(6.1-6.8) has the notation DE/rand/1/bin.

The DE/ran/1/bin algorithm pits each vector $x_{i,g}$ in the current population against a trial vector $u_{i,g}$ to whose composition it contributes through uniform crossover with a randomly ("/ran/") chosen base vector $x_{r1,g}$ that has been mutated by the addition of a single ("/1/") scaled and randomly chosen difference vector $F(x_{r1,g} - x_{r2,g})$ The appellation "bin" refers to the fact that the number of parameters inherited by the trial vector $u_{i,g}$ from the mutant vector $v_{i,g}$ approximates a binomial distribution. During survivor selection, $u_{i,g}$ replaces $x_{i,g}$ if $f(u_{i,g}) \leq f(x_{i,g})$ *f*; otherwise, $x_{i,g}$ retains its place in the population.

The mutation being used in DE/best/1/bin is given as

$$v_{i,g} = x_{best,g} + F.(x_{r1,g} - x_{r2,g}) \qquad (6.9)$$

6.4 Crossover

To complement the differential mutation search strategy, DE employs uniform crossover as the classic variant of diversity enhancement which mixes parameters of the mutation vector $v_{i,g}$ and the so-called *target vector* $x_{i,g}$ in order to generate the trial vector $u_{i,g}$ as

$$\mathrm{u}_{\mathrm{i,g}} = u_{j,i,g} \begin{cases} v_{j,i,g}, if (rand_j[0,1) \leq C_r or & j = j_{rand} \\ x_{j,i,g} & otherwise \end{cases} \qquad (6.10)$$

The crossover probability, $C_r \in [0,1]$, is a user-defined value that controls the fraction of parameter values that are copied from the mutant. If the random number is less than or equal to *Cr* , the trial parameter is inherited from the mutant $v_{i,g}$; otherwise, the parameter is copied from the vector, $x_{i,g}$. In addition, the trial parameter with randomly chosen index, j_{rand} , is taken from the mutant to ensure that the trial vector does not duplicate $x_{i,g}$. Figure 6.2 plots possible trial vectors that can result from uniformly crossing a mutant vector $v_{i,g}$, with the vector $x_{i,g}$.

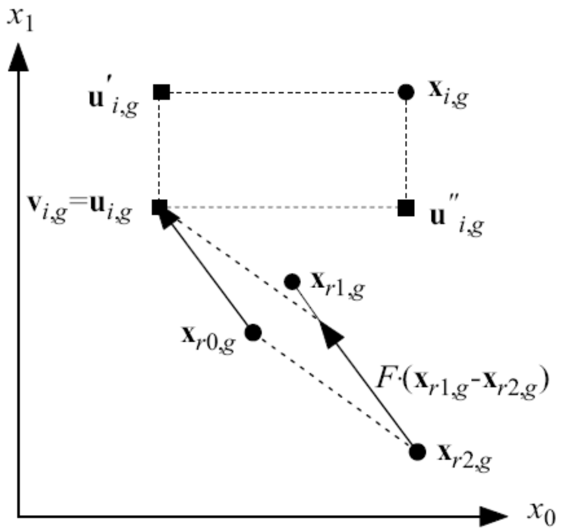

Figure 6.2 The possible additional trial vectors $u'_{i,g}$,$u''_{i,g}$ of $x_{i,g}$ and $v_{i,g}$.

6.5 Selection

DE uses simple one-to-one surviovr selection where the trial vector $\mathbf{u}_{i,g}$ competes against the target vector $\mathbf{x}_{i,g}$. If the trial vector, $\mathbf{u}_{i,g}$, has an equal or lower objective function value than its target vector, $\mathbf{x}_{i,g}$, it replaces the target vector in the next generation; otherwise, the target retains its place in the population for at least one more generation (equation (6.13)).

$$x_{i,g+1} = \begin{cases} u_{i,g}; & if \quad f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g}; & if \quad otherwise. \end{cases} \tag{6.11}$$

Once the new population is installed, the process of mutation, recombination and selection is repeated until the optimum ia located, or a prespecified termination criterion is satisfied, e.g., the number of generations reaches a preset maximum, g_{max}. Figure 6.3 shows the flowchart of DE algorithm and table 6.1 presents the pseudo-code for classic DE.

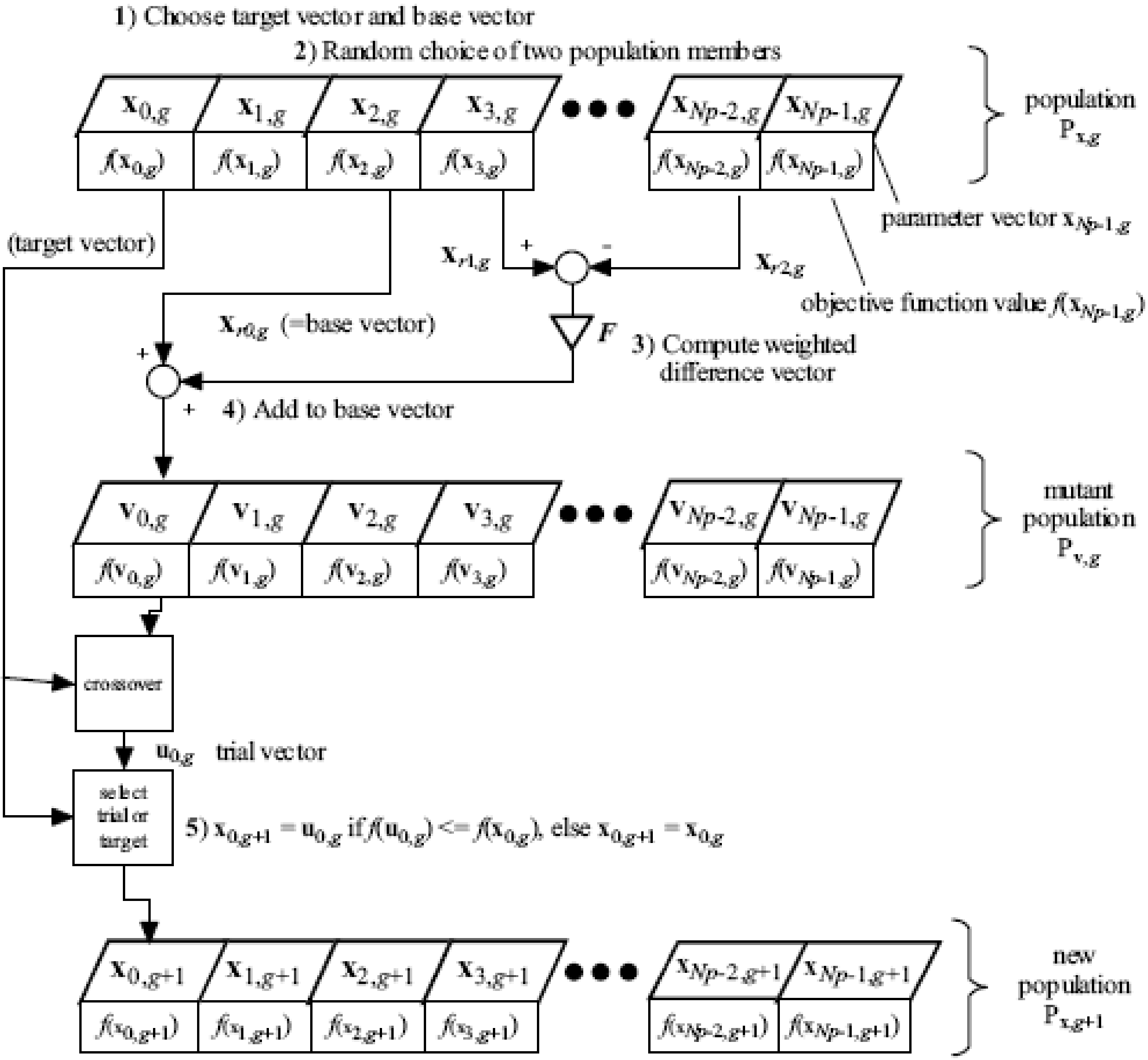

Figure 6.3 Flowchart of Differential Evolution

Table 6.1 Classic algorithm of DE

```
  FOR ( i = 1;  i ≤ Np; i = i+1 )  // Initiliaze population
     FOR ( j = 1; j ≤ D; j = j+1 )  x_{j,i,g}= x_j^lower +U_j(0,1).(x_j^upper - x_j^lower);
     END FOR
     f(i) = f(x_{i,g});   //Evaluate and store f(x_{i,g})
END FOR
FOR ( g = 1; g ≤ G; g = g+1 )   //  Generation loop
    FOR ( i = 1; i ≤ Np; i = i+1 )  //  Generate a trial population
        Jrand_i = floor[R_i(0,1).D]+1;   //  Randomly select a parameter
         DO r1 = floor(R(0,1)_i.Np)+1; WHILE (r1 = i); //Select 3 distinct indices
         DO r2 = floor(R(0,1)_i.Np)+1; WHILE (r2 = i or r2 = r1);
         DO r3 = floor(R(0,1)_i.Np)+1; WHILE (r3 = i or r3 = r1 or r3 = r2);
        FOR ( j=1; j ≤ D; j= j+1;)   // Generate a trial vector
            if (Rj(0,1) ≤  Cr or  j= jrand_i) u_{j,i,g} = v_{j,i,g} = x_{j,r1,g}+F.( x_{j,r2,g}-.x_{j,r3,g});
             else u_{j,i,g}= x_{j,i,g;}
         END FOR
     END FOR
      FOR ( i = 1; i ≤ Np; i = i+1 )   //  Select new population
          if  f(u_{i,g}) ≤  f(x_{i,g})    //  Evaluate trial vector and compare with target vector
          {
              FOR  ( j = 1;  j ≤ Np; j = j+1 ) x_{j,i,g}  = u_{j,i,g};  // Replace inferior target
              END FOR
              f(i) = f(u_{i,g});
           }
        END FOR
END FOR  //  End
```

6.6 Harmonic Estimation

In hybrid algorithms given in chapters 4 and 5, for the estimation of phase angles GA and PSOPC are used as the optimization algorithm. Amplitude estimation by Least Squares Method take part after estimation of phase angles. By following the same iterative procedure, in this thesis, instead of GA and PSOPC, as the optimization algorithm DE is proposed for the estimation of phases.

To apply the algorithm, the parameters, which are common in three population algorithms should be initialized as in the previous algorithms; *D,* the number of phases to be estimated, which is the dimension of the particles, *Np*, the size of the population and *G*, number of maximum generations. The cost function is the same as given in equation (4.24). In the first iteration, as in the previous algorithms the values of vectors,

$$x_{i,g} = (x_{j,i,g}) = (\hat{\phi}_{j,i,g}), \; j = 0,1,\ldots,D\text{-}1. \tag{6.12}$$

are randomly selected. Consequently, $\hat{H}(k)$ is calculated according to equation (4.21). Afterwards, the amplitudes $\hat{A}_{i,g} == [\hat{A}_{j,i,g}]$ are estimated by equation (4.23.). Finally, the signal $\hat{Z}_{i,g}$ is reconstructed from $\hat{\phi}_{i,g}$ and $\hat{A}_{i,g}$ by equation (4.22) The cost function of the vector $x_{i,g}$, $f(x_{i,g})$, which is the error between $\hat{Z}(k)$ and $Z_{i,g}(k)$ is calculated according to equation (4.21). Then for each target vector $x_{i,g}$, $i = 1,2,\ldots, N_p$, a mutant vector $v_{i,g}$ is generated from randomly chosen three vectors which are distinct from each other and *i*th vector. $v_{i,g}$ is produced by adding a scaled, randomly sampled, vector difference to a third vector according to the equation (6.9) and by appliying uniform crossover to the target and mutant vectors, trial population $U_g = [x_{i,g}]$ is obtained. By calculating the cost function of trial population and comparing with target population, the new target vectors $x_{i,g+1}$ are simply updated according to equation (6.13) and new population $P_{x,g+1}$ is obtained. The phases in new target vectors are used in the next iteration in the structure matrix. The process repeats until the maximum iteration G is reached or the condition $f(x_{i,g}) < \varepsilon$ is satisfied. The pseudo code for the proposed algorithm is presented in table 6.2 to describe the estimation process.

Table 6.2 Pseudo-code for the LS-DE based algorithm

Set g:=1 and randomly intialize all target vectors of population P_x, xi = $[x_{j,i}] = [\Phi_{j,i}]$

FOR (each target vector x_i in the target population)

Calculate amplitudes: Calculate the amplitudes of each harmonic , $\hat{A}_{i,g} = [\hat{A}_{j,i,g}];$ using the LS method given by equation(4.20)

Calculate fitness: Calculate the fitness value of current vector $f(x_{i,g})$ using equation (4.21) and store in f(i). Find x_{best} with the best fitness.

END FOR

WHILE (the termination conditions are not met)

FOR (each vector u_i in the trial population)

Select vectors: Randomly select two vectors x_{r1}, x_{r2} from target population such that r_1,r_2 are different from each others, the target vector index,. and the best vector x_{best}.

Generate mutant vector: Generate mutant vector $v_{i,g}$ from x_{best}, x_{r1}, x_{r2} according to the equation (6.9)

Generate trial vector: Generate trial vector $u_{i,g}$ by applying uniform crossover to $v_{i,g}$ and $x_{i,g}$ according to the equation (6.10)

Calculate amplitudes: Calculate the amplitudes of each harmonic , $\hat{A}_{i,g} = [\hat{A}_{j,i,g}];$ using the LS method given by equation(4.23)

Calculate fitness: Calculate the fitness value of current vector, $f(u_{i,g})$ using equation (4.24). Find x_{best} with the best fitness.

Update population: Compare fitness value $f(u_{i,g})$ with f(i). If $f(u_{i,g})$ is better than $f(x_{i,g})$, then replace the target vector with trial vector and update f(i) as $f(u_{i,g})$

END FOR

Set g:=g+1;

END WHILE

7. APPLICATIONS

7.1 Application of Hybrid Least-Squares GA Based Algorithm for Harmonic Estimation

The test signal $Z_0(t)$ in this application, was also used in previous studies [17;18;19]. The test signal is a distorted voltage waveform at the terminals of the load bus for the system in figure 7.1. Figure 7.1 shows the sample system which comprises of a two-bus three-phase system with a full-wave six pulse bridge rectifier at the load bus.

The front-end diode bridge rectifiers of 3-phase, 6-pulse static power convertors (ac-dc), such as those found in variable speed drives, are considered nonlinear because they draw current in a non-sinusoidal manner. The characteristic harmonics are based on the number of rectifiers (pulse number) used in a circuit and can be determined by the following equation:

$$h = (n \times p) \pm 1 \tag{7.1}$$

where:
n = an integer (1, 2, 3, 4, 5 …)
p = number of pulses or rectifiers

Therefore, for 6 pulse rectifier, the characteristic harmonics will be:
h = (1 x 6) ± 1 → 5th & 7th harmonics
h = (2 x 6) ± 1→11th & 13th harmonics
h = (3 x 6) ± 1 →17th & 19th harmonics
h = (4 x 6) ± 1 →23rd & 25th harmonics

Harmonic distortion will be with the predominant harmonics being the 5th and 7th. The 11th, 13th and other higher orders are also present but at lower levels.

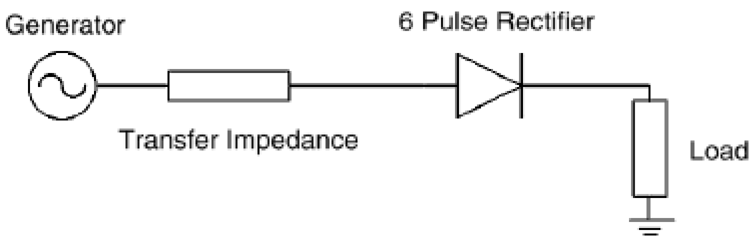

Figure 7.1 Simple power system: a two-bus architecture with six-pulse full-wave bridge rectifier supplying the load [18]

The test signal is mathematically described as

$$Z_0(t) = A_1 * \sin(w_1 t + \phi_1) + A_5 \sin(w_5 t + \phi_5) + A_7 \sin(w_7 t + \phi_7) + A_{11} \sin(w_{11} t + \phi_{11}) + A_{13} \sin(w_{13} t + \phi_{13}) + v(t) \quad (7.2)$$

where v(t) is the white gaussian noise, A_1, A_5, A_7, A_{11}, A_{13} representing the amplitudes and ϕ_1, ϕ_5, ϕ_7, ϕ_{11}, ϕ_{13} representing the phase angles of fundamental component, 5th, 7th, 11th and 13th harmonics respectively.The harmonic content of test signal $Z_0(t)$ is given in table 7.1.

Table 7.1 Harmonic Content of test signal $Z_0(t)$

Harmonic Order	Amplitude (p.u)	Phase Angle (Degrees)
Fundamental (50 Hz)	0.95	-2.02
5th (250 Hz)	0.09	82.1
7th (350 Hz)	0.043	7.9
11th (550 Hz)	0.03	-147.1
13th (650 Hz)	0.033	162.6

Simulation are employed to produce the test signal and sample 64 points per cycle from the distorted 50-Hz voltage waveform. The algorithm is operated under non-noisy and noisy situations. The signal-to-noise ratios (SNRs) are chosen to be 20 dB, 10 dB and 0 dB, respectively. For every situation simulation is repeated 10000 times and average, minimum and maximum error index values are recorded.

7.1.1 Encoding-decoding

In the application, the first step is to represent the problem variables as a chromosome by binary encoding. It is assumed that sample signal contains 5rd, 7th, 11th and 13th harmonics. Thus each chromosome will represent the 5 phase values including the phase of fundamental component. A chromosome becomes a concatenated binary string in which each phase is represented by n binary bits. Each phase is represented by 19 bits as proposed in [18] and consequently lenght of a chromosome which is the number of bits of an individual representing 5 phases, becomes 95.

In the decoding stage chromosomes are partitioned into bit strings representing phase values and decoded to values in problem domain by converting to decimal and scaling. Since a phase represented by 19 bits in a chromosome, it can take values between 0 and 2^{19}-1 decimally.The decoding was given by equation (4.16).

Since the bounds of variable domain are b_L = 0 and b_U = 360 degrees, after converting the each binary string to decimal, it is normalized by a factor $1/(2^{19}-1)$ and multiplied by 360 which maps this interval into problem domain. The decoded value of 19 bit binary string becomes

$$x = \frac{360}{2^{19}-1} . \sum_{k=0}^{n-1} 2^k \qquad (7.3)$$

The precision of the decoding which is half of the difference of two consecutive phase values, will be

$$\frac{360}{2^{19}-1} . \frac{1}{2} = 3.4332\text{x}10^4$$

7.1.2 Fitness function, performance measure and parameter determination

Size of the population N is selected as 50 and number of generations G is selected as 100.

The performance measure fitness function is defined as

$$\text{Fitness}(n) = 1/\sum \text{error}(k)^2 = \frac{1}{\sum_{k=1}^{K} (Z(k) - \hat{Z}(k)\ )^2} \qquad (7.4)$$

where n is the chromosome number, k is the sample number, Z is the original signal and $\hat{Z}$ Is the estimated signal value. According to the equation (7.4) the chromosome which has smaller error will have a greater fitness value, hence bigger chance for selection.

A typical value proposed for the crossover probability P_c is 0.7 which was also used in [18]. It is stated that $P_c = 0.7$ generally gives good results and Pm is usually selected as 0.001 in many applications [42]. However, sometimes a greater mutation may lead to significant improvement in the population fitness, by increasing the randomness to avoid from local optimum. Thus, one way of providing some degree of insurance is to compare results obtained under different rates of mutation [42].

In this application, to ensure the optimal value, P_c is selected experimentally, starting from 0.1, incremented by 0.1 until 1. For every P_c value, algorithm is executed 100 times, each having 100 generations and average fitness values of population is recorded. Performance graph showing mean of the average performances of population for P_c values from 0.6 to 1, over 100 executions is given in figure 7.2. It is seen that, average fitness of population for P_c=0.7, is better than other P_c values. As a result, $P_c = 0.7$ is also verified as an optimal cross probability value as in [18].

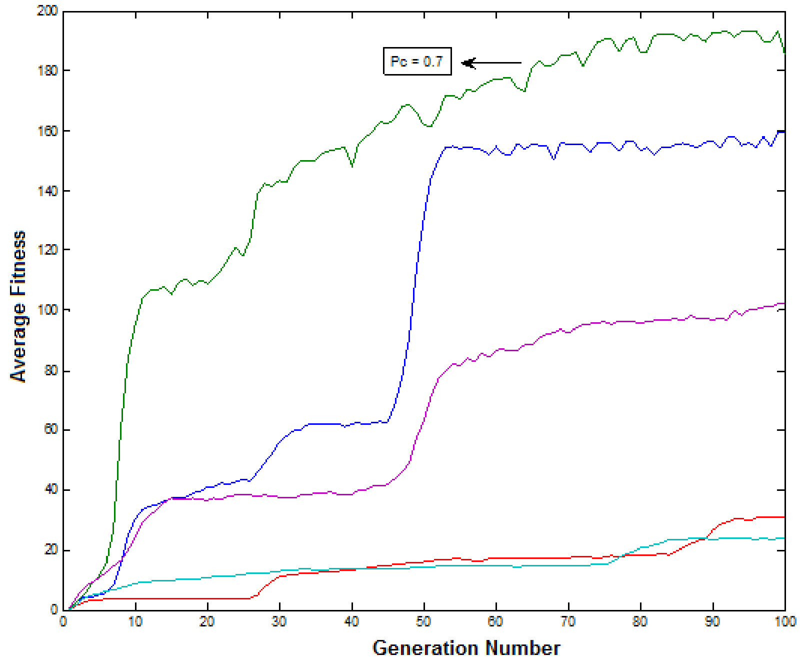

Figure 7.2 Performance graphs for different P_c values (P_m=0.01)

Same procedure is followed for the determination of mutation rate Pm. For P_c=0.7, mutation rate P_m is incremented starting from 0.001, by 0.001 until 0.01 and for each value algorithm executed 100 times. It is observed that, when mutation rate

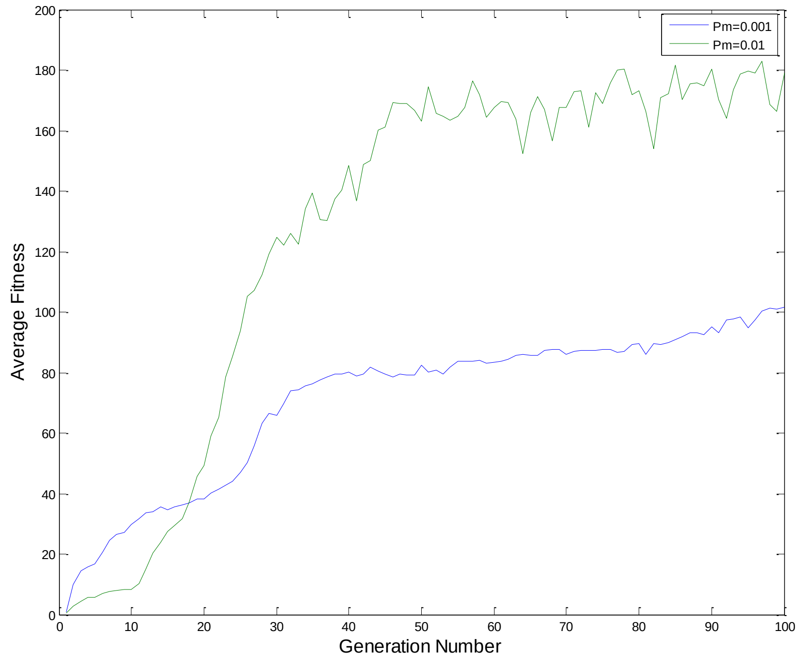

Figure 7.3 Performance graphs for different P_m values (P_c=0.7)

is not introduced or when it is very low, premature convergence may occur which chromosomes begin to converge early on towards solutions which may no longer be valid for later data. Increasing Pm values helps to overcome this problem by increasing genetic diversity. This characteristic is seen from the performance graph (figure 7.3) for the mutation rate, P_m=0.01 and Pm=0.1. Oscillations in performance curve are due to increased mutation rate which increased randomness, equivalently genetic diversity. As proposed in [18] P_m=0. 01 is found to be a suitable mutation rate.

In the simulation studies, as the performance measure, error index of the estimated signal described in [19] is calculated by

$$\varepsilon_1 = \frac{\sum_{k=1}^{K}(Z(k)-\hat{Z}(k))^2}{\sum_{k=1}^{K} Z(k)^2} x100 \qquad (7.5)$$

where k is the sample number and Z(k) is the value of the *k*th sample of actual signal. In [19] error index of fundamental frequency component is given as

$$\varepsilon_2 = \frac{\sum_{k=1}^{K}(S(k)-\hat{Z}_f(k))^2}{\sum_{k=1}^{K} S(k)^2} x100 \qquad (7.6)$$

where S(k) is the value of *k*th sample of the actual fundamental frequency component in the original signal given by

$$S(k) = H_f(k)A_1 \qquad (7.7)$$

where A_1 is the amplitude of fundamental component and $H_f(k)$ is the first column of the structure matrix $H(k)$ given as

$$H_f(k) = \begin{bmatrix} \sin(w_1 t_1 + \phi_1) \\ \sin(w_1 t_2 + \phi_1) \\ \vdots \\ \sin(w_1 t_k + \phi_1) \\ \vdots \\ \sin(w_1 t_K + \phi_1) \end{bmatrix} \tag{7.8}$$

Estimation of fundamental components shown in [18;19] were calculated according to the equation given below

$$\hat{Z}_f(k) = \hat{H}_f(k)\hat{A}_1 \tag{7.9}$$

The obtained results are reconstructed into a signal for the purpose of comparison with the actual fundamental frequency component presented in the original signal. However, from the practical point of view, the fundamental component can be obtained by measuring the level of harmonic current(or voltage) present in the system and injecting currents (or voltages) of opposite polarity to cancel them out. Therefore, for the comparison of estimated and actual fundamental component, in this thesis another error index, is calculated as

$$\varepsilon_3 = \frac{\sum_{k=1}^{K} (S(k) - \hat{S}(k))^2}{\sum_{k=1}^{K} S(k)^2} x100 \tag{7.10}$$

where $\hat{S}(k)$ is obtained by extracting the estimated harmonics from the sampled signal as

$$\hat{S}(k) = Z(k) - \hat{Z}_h(k) = Z(k) - \hat{H}_h(k)\hat{A}_h \tag{7.11}$$

where $\hat{Z}_h(k)$ is the sum of estimated harmonics, $\hat{A}_h = \begin{bmatrix} \hat{A}_5 \\ \hat{A}_7 \\ \hat{A}_{11} \\ \hat{A}_{13} \end{bmatrix}$ represents the estimated amplitudes of harmonics and $\hat{H}_h(k)$ is given by

$$\hat{H}_h(k) = \begin{bmatrix} \sin(w_5 t_1 + \hat{\phi}_5) & \sin(w_7 t_1 + \hat{\phi}_7) & \sin(w_{11} t_1 + \hat{\phi}_{11}) & \sin(w_{13} t_1 + \hat{\phi}_{13}) \\ \sin(w_5 t_2 + \hat{\phi}_5) & \sin(w_7 t_2 + \hat{\phi}_7) & \sin(w_{11} t_2 + \hat{\phi}_{11}) & \sin(w_{13} t_2 + \hat{\phi}_{13}) \\ \vdots & \vdots & \vdots & \vdots \\ \sin(w_5 t_k + \hat{\phi}_5) & \sin(w_7 t_k + \hat{\phi}_7) & \sin(w_{11} t_k + \hat{\phi}_{11}) & \sin(w_{13} t_k + \hat{\phi}_{13}) \\ \vdots & \vdots & \vdots & \vdots \\ \sin(w_5 t_K + \hat{\phi}_5) & \sin(w_7 t_K + \hat{\phi}_7) & \sin(w_{11} t_K + \hat{\phi}_{11}) & \sin(w_{13} t_K + \hat{\phi}_{13}) \end{bmatrix} \tag{7.12}$$

7.1.3 Simulation results

The simulation is repeated 10000 times for the noisy and non-noisy conditions, average, minimum and maximum error index values for each is recorded. For SNR values 0 dB, 10 dB, 20 dB and when there is no noise, $\varepsilon_1, \varepsilon_2, \varepsilon_3$ values are given in table 7.2, 7.3 and 7.4 respectively.

Table 7.2 Average, minimum and maximum ε_1 values for noisy and non-noisy situations in LS-GA based algorithm

Noise Conditions	Error index ε_1		
	Average	Min	Max
No noise	0.0316	2.9951×10^{-8}	0.1881
SNR = 20 dB	0.1702	0.0010	0.7040
SNR = 10dB	0.7561	0.0112	6.1521
SNR = 0dB	4.0917	0.0780	53.8073

Table 7.3 Average, minimum and maximum ε_2 values for noisy and non-noisy situations in LS-GA based algorithm

Noise Conditions	Error index ε_2		
	Average	Min	Max
No noise	0.0211	4.0242×10^{-10}	0.1267
SNR = 20 dB	0.0548	4.1119×10^{-8}	0.5213
SNR = 10dB	0.3561	3.8323×10^{-7}	6.0620
SNR = 0dB	3.4337	1.8931×10^{-5}	53.4942

Actual fundamental component and reconstructed estimations are shown in figure 7.4 for different noise conditions. As seen from figure 7.4, the reconstructed waveform according to the equation (7.9) is a pure sinusoidal. The corresponding ε_2 error index values calculated from equation (7.6) are given in table 7.3.

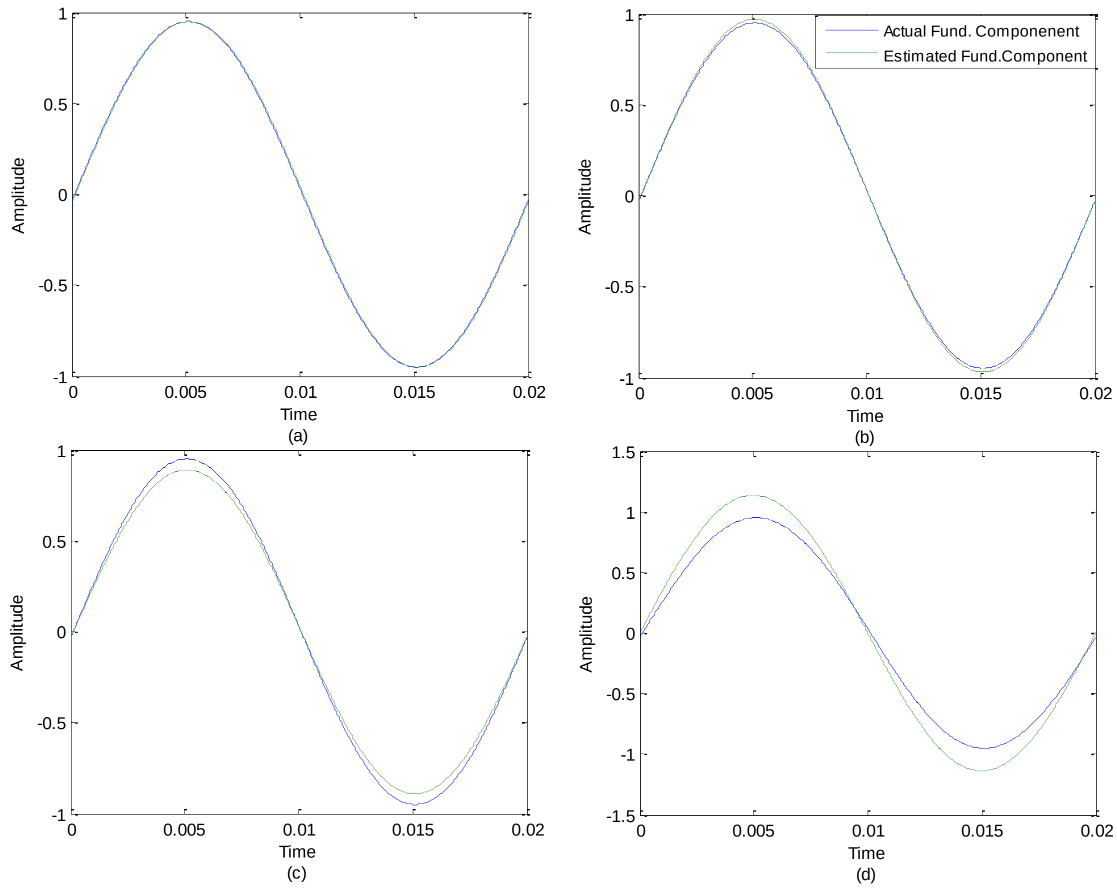

Figure 7.4 Actual and estimated fundamental components by LS-GA based algorithm according to equation (7.9) for (a) No noise (b) SNR = 20 dB (c) SNR = 10 dB (d) SNR = 0 dB

Table 7.4 Average, minimum and maximum ε_3 values for noisy and non-noisy situations in LS-GA based algorithm

Noise Conditions	Error index ε_3		
	Average	Min	Max
No noise	0.0109	3.7896×10^{-9}	0.0624
SNR = 20 dB	0.1204	0.0049	0.5035
SNR = 10dB	0.4102	0.0166	1.3270
SNR = 0dB	0.7109	0.0076	2.9992

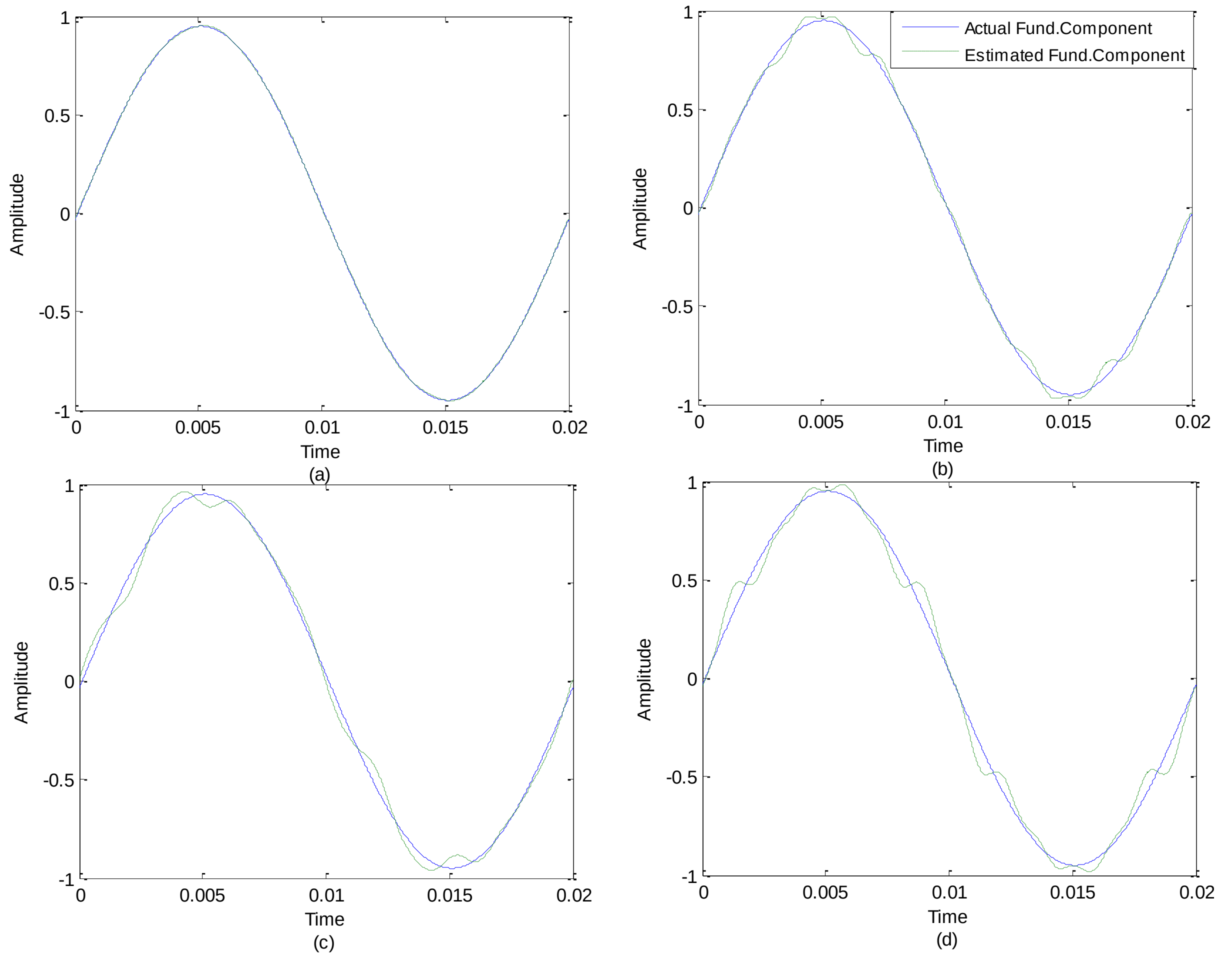

Figure 7.5 Actual and estimated fundamental components by LS-GA based algorithm, according to equation (7.11) for (a) No noise (b) SNR = 20 dB (c) SNR = 10 dB (d) SNR = 0 dB

On the other hand, figure 7.5 shows the estimated fundamental component obtained according to the equation (7.10) and table 7.4 gives the corresponding error index values ε_3 calculated according to the equation (7.10) considering practical concerns. The estimated fundamental waveform in figure 7.5 , is obtained by cancelling the harmonics, therefore, differently from figure 7.4, waveform is not a pure sinusoidal. Consequently, two calculations will result in different estimation errors will be different. Moreover, it is observed that in different noisy -conditions error index values ε_2 and ε_3 do not change in same way such that a lower ε_2 does not always ensure a lower ε_3 or vice versa correspondingly. Therefore, regarding this characteristic, in the following applications, ε_3 which is calculated based on compensation of estimated harmonics, is proposed to be used as an error index for fundamental component instead of ε_2.

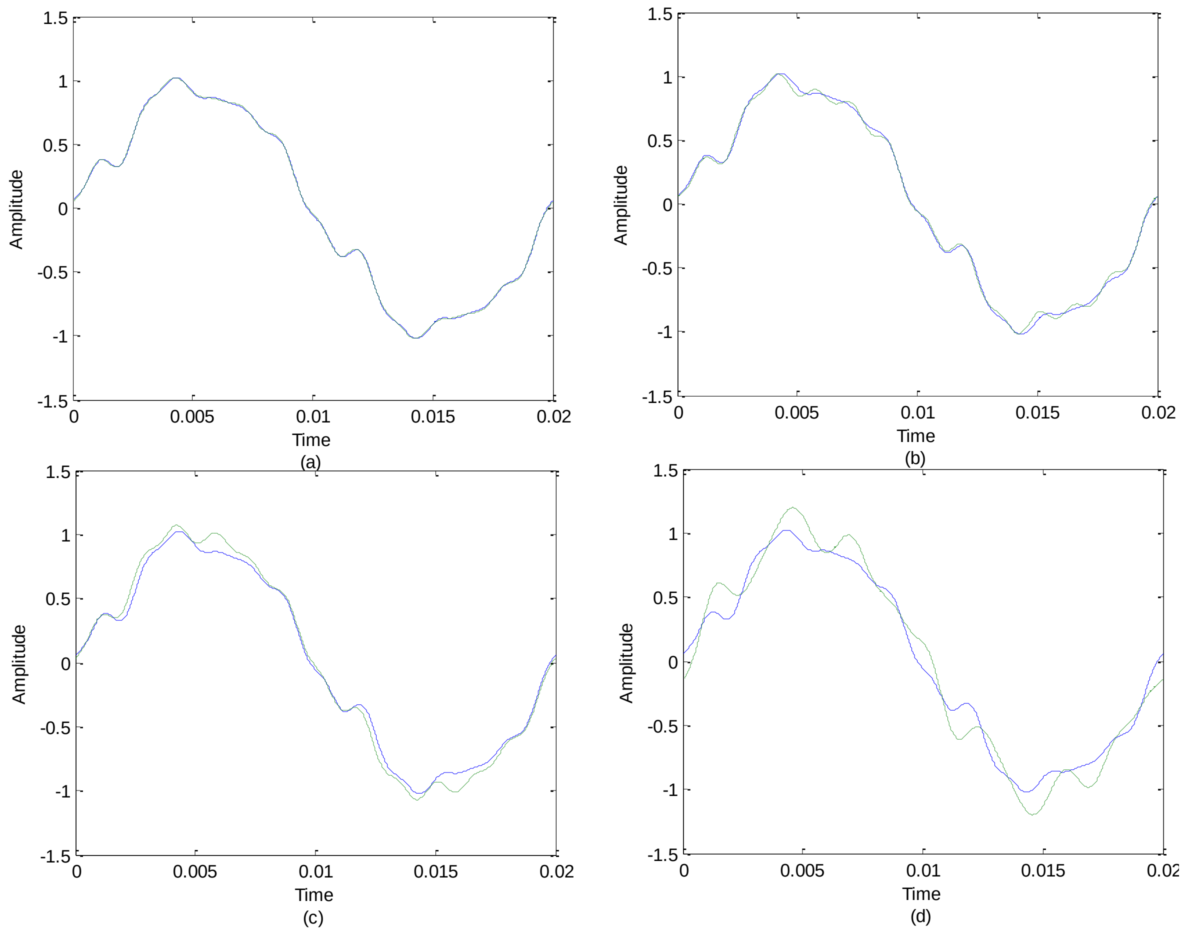

Figure 7.6 Actual and estimated distorted waveforms by LS-GA based algorithm, for (a) No noise (b) SNR = 20 dB (c) SNR = 10 dB (d) SNR = 0 dB

The original distorted waveform $Z_0(t)$ which is sum of all harmonics and fundamental component without noise, and its estimation $\hat{Z}(t)$ reconstructed from equation (4.18) is shown in figure 7.6. The corresponding error index values are given in table 7.2 for different noise conditions and the estimated amplitudes are given in table 7.5.

Table 7.5 Actual and average of estimated amplitudes in LS-GA based algorithm for (a) No noise (b) SNR = 20 dB (c) SNR = 10 dB (d) SNR = 0 dB

Harmonic Order	Actual Amplitudes	Estimated Amplitudes			
		No Noise	20 dB	10 dB	0 dB
Fundamental (50Hz)	0.9500	0.9499	0.9469	0.9509	0.9466
5th (250 Hz)	0.0900	0.0897	0.0900	0.0831	0.0656
7th (350 Hz)	0.0430	0.0429	0.0420	0.0338	0.0309
11th (550 Hz)	0.0300	0.0297	0.0280	0.0223	0.0221
13th (650 Hz)	0.0330	0.0328	0.0314	0.0246	0.0232

7.2 Application of Hybrid Least Squares-PSOPC Based Algorithm For Harmonic Estimation

In this application the test signal Z(t) has the same harmonic content as the used one in part 7.1, given by the equation (7.2) and in table 7.1.

A simulation is employed to produce the test signal and sample 64 points per cycle from the distorted 50-Hz waveform. The algorithm utilizing PSO with passive congregation is operated under non-noisy and noisy situations. The signal-to-noise ratios (SNRs) are chosen to be 20 dB, 10 dB and 0 dB, respectively. For every situation simulation is repeated 10000 times and error index values ε_1 and ε_3 given by equations (7.5) and (7.10), are recorded.

The values of parameters c_1, c_2 and c_3 are determined experimentally by changing one parameter by 0.1 while keeping others fixed. It is verified that setting c_1,c_2 as

0.5 and the passive congregation coefficient as 0.6 gives the best results as proposed in [19]. Maximum number of generations G is set as 100 and number of particles Np is selected as 50. Max velocity is generally limited as the half of the difference of upper and lower boundaries [55]. Thus, following the usual approach maximum velocity for each dimension is limited as $V_{max} = (u_L - b_L)/2 = 180$.

In [52], it is reported that when time varying inertia weight is employed, even better performance can be obtained and in [53] it is experimentally shown on benchmark functions, lineraly decreasing inertia weight improves the performance of PSO. Therefore, a decaying inertia weight starting at 0.9 and ending at 0.4, is used in [53]. However, adaptation of inertia weight may be problem dependent. Linearly decreasing inertia wiegt can be given as

$w=w_i-\alpha.g/G$

where w_i is the inital inertial weight which is generally set as 0.8 or 0.9,

α is the linear decreasing coefficient,

g is the generation number and

G is the total number of generations

By trying different adaptations, it is observed that it is suitable to initialize inertia weight as 0.9 and adapt as w = 0.9 - 0.4g/G. While the genereation number increases, w decreases linearly from 0.9 to 0.5. When fixed inertia weight w=0.9 is used, velocities more oscillate, so for convergence of phase angles, more number of generations is needed. For fixed and time-varying inertia weights, the behaviour of average velocities and average positions of particles are shown in figure 7.7. It is seen that adaptive inertia weight provides better convergence and results in fewer iterations than a fixed inertia weight.

The harmonic estimation procedure follows the same way as in the previous application. After phase estimation, amplitudes are estimated by LS, but for phase estimation instead of GA, PSOPC is utilized.Fitness values of swarm is calculated and phase angles represented by positions of particles are used in the next iteration.

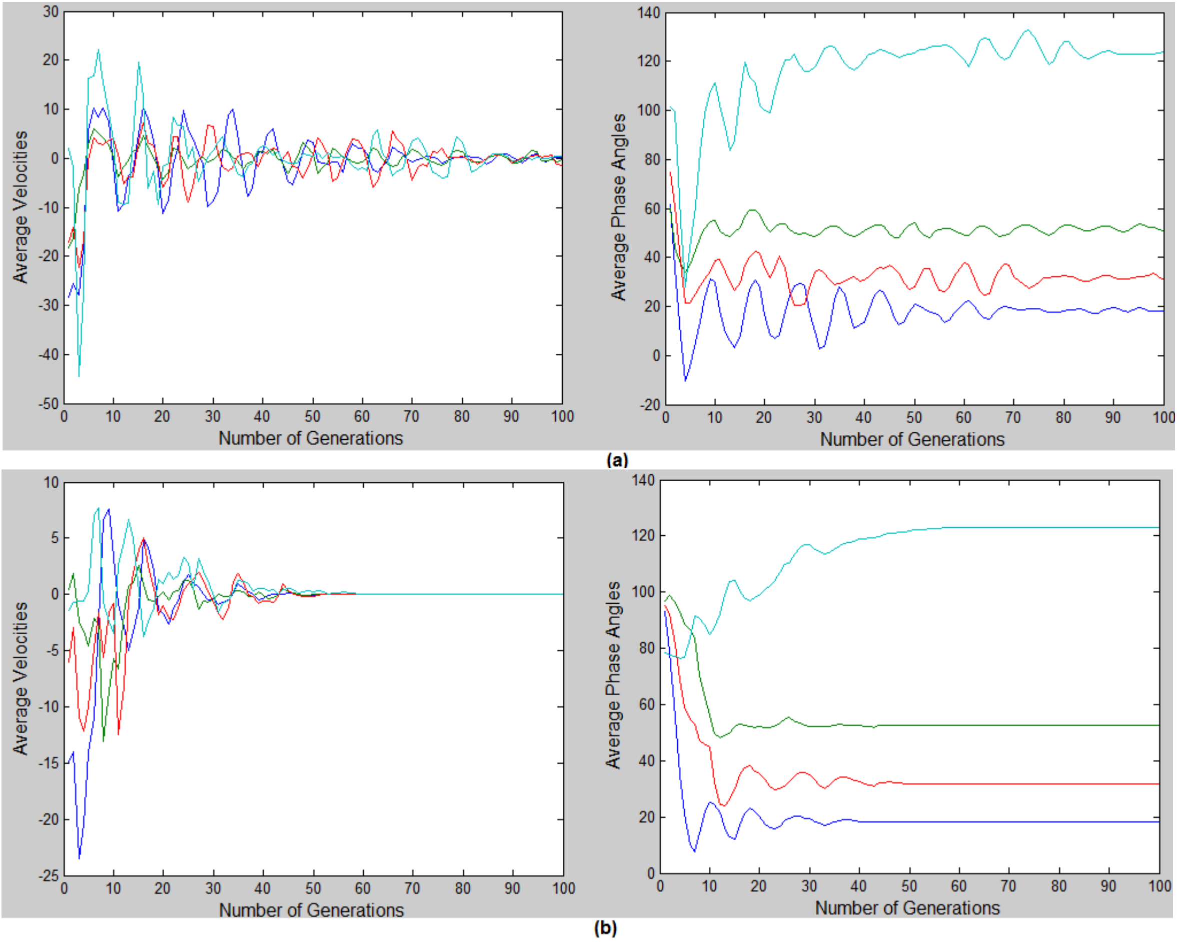

Figure 7.7 Average velocities and positions of population for (a) Fixed inertia weight (b) Time-varying inertia weight

Average, minimum and maximum error index values for 10000 runs of the simulation are given in table 7.3. By comparing the tables 7.2 and 7.6 it can be seen that PSOPC provides more accuracy in approximation of *Z*(*t*), than GA, in all noisy and non-noisy situations, especially when there is no noise. The estimated and actual waveforms are shown in figure 7.8 for average ε_1 values. ε_1 gives a measure of the approximation to the distorted signal Z(*t*). Therefore, in non-noisy condition, ε_1 can be used to compare the algorithms instead of ε_3 equivalently. However, considering practical concerns it is suitable to use ε_3 for comparison of estimations of fundamental component.

Table 7.6 Average, minimum and maximum ε_1 values for noisy and non-noisy conditions in LS- PSOPC based algorithm

Noise Conditions	Error index ε_1		
	Average	Min	Max
No noise	0.0041	2.2194×10^{-11}	0.1848
SNR = 20 dB	0.1449	0.0007	0.6761
SNR = 10dB	0.7190	0.0110	5.9502
SNR = 0dB	3.9593	0.0501	53.7733

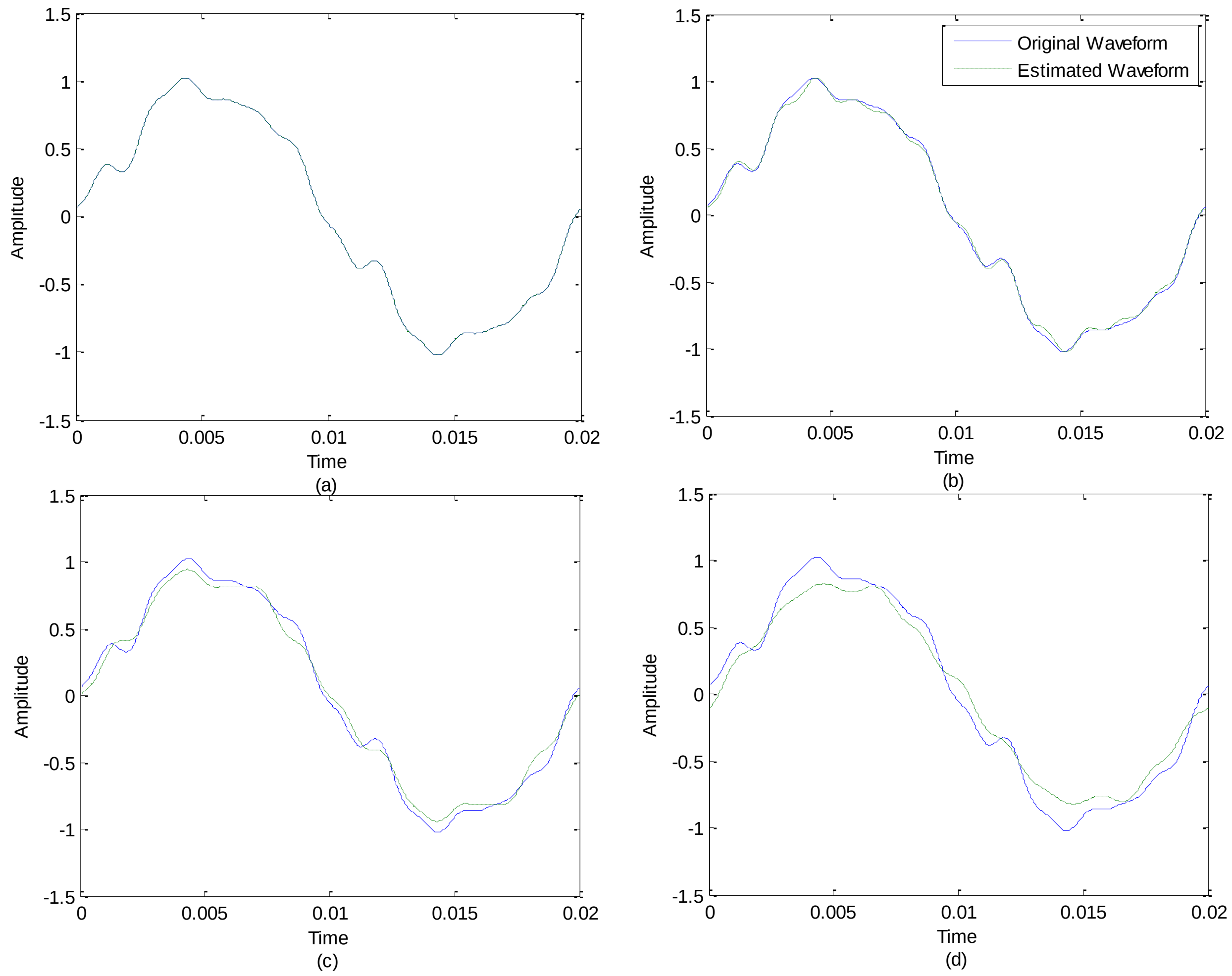

Figure 7.8 Estimated and actual distorted waveforms by LS-PSOPC based algorithm for (a) No noise (b) SNR = 20 dB (c) SNR = 10 dB (d) SNR = 0 dB

Table 7.7 Average, minimum and maximum ε_3 values for noisy and non-noisy conditions in LS-PSOPC based algorithm

Noise Conditions	Error index ε_3		
	Average	Min	Max
No noise	0.0042	1.9389×10^{-11}	0.1905
SNR = 20 dB	0.1119	0.0005	0.5419
SNR = 10dB	0.3928	0.0032	1.3056
SNR = 0dB	0.6084	0.0066	1.8053

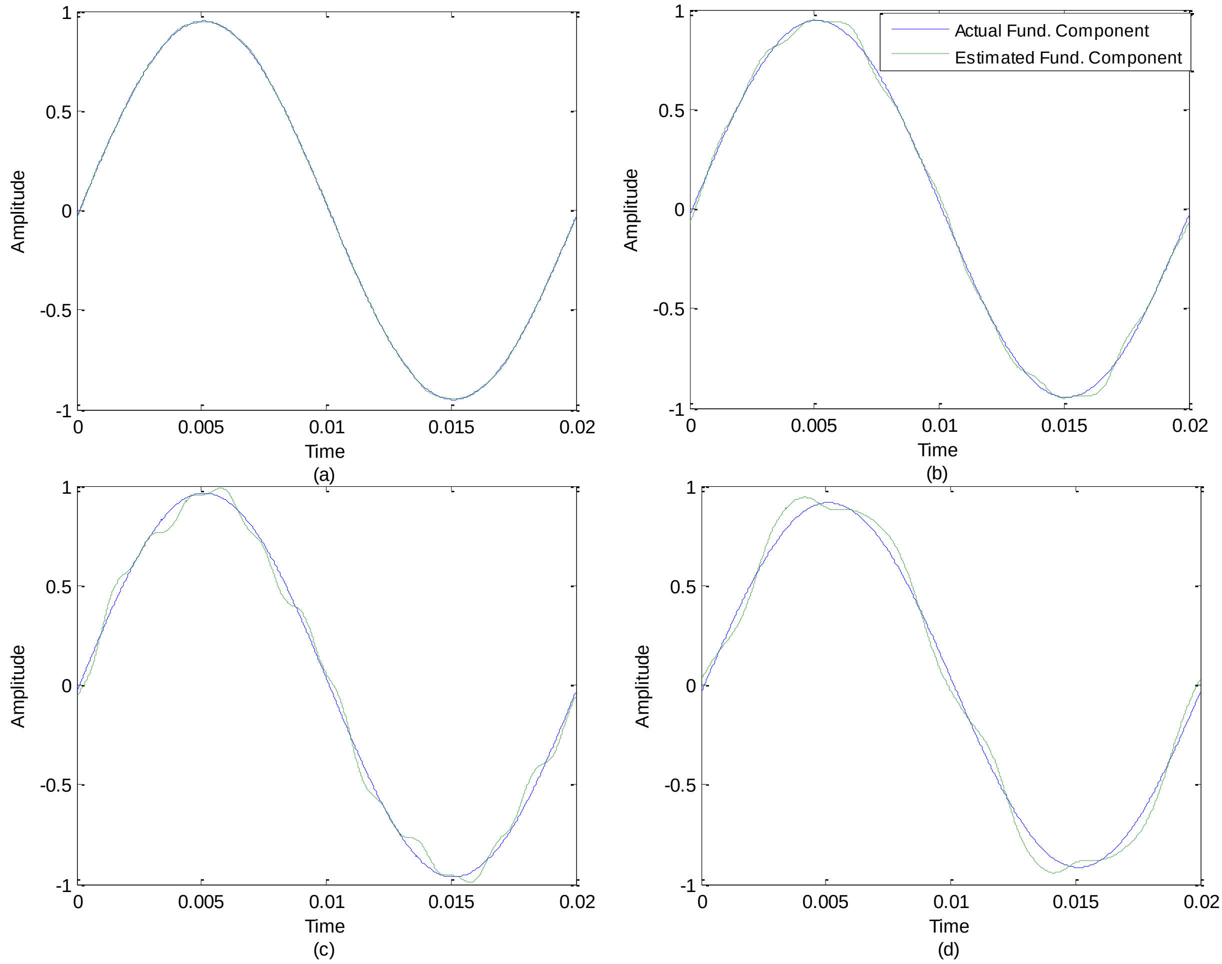

Figure 7.9 Estimated and actual fundamental components by LS-PSOPC based algorithm for (a) No noise (b) SNR = 20 dB (c) SNR = 10 dB (d) SNR = 0 dB

The estimated average amplitudes of LS-PSOPC based algorithm are given in table 7.8. When the results are analysed with the previous application which utilizes GA, it is seen that hybrid algorithm utilizing PSOPC achieves improved estimation accuracy in comparison with GA. Comparing the rates of error index values from tables 7.3 and 7.7, it is seen that PSOPC is significantly better when there is no noise.

Table 7.8 Actual and average of estimated amplitudes in LS-PSOPC based algorithm for (a) No noise (b) SNR = 20 dB (c) SNR = 10 dB (d) SNR = 0 dB

Harmonic Order	Actual Amplitudes	Estimated Amplitudes			
		No Noise	20 dB	10 dB	0 dB
Fundamental (50Hz)	0.9500	0.9500	0.9497	0.9510	0.9466
5th (250 Hz)	0.0900	0.0900	0.0902	0.0832	0.0648
7th (350 Hz)	0.0430	0.0429	0.0420	0.0336	0.0288
11th (550 Hz)	0.0300	0.0296	0.0279	0.0219	0.0197
13th (650 Hz)	0.0330	0.0328	0.0313	0.0243	0.0217

7.3 Application of A Novel Hybrid Least Squares- DE Based Algorithm For Harmonic Estimation

In this application, for the purpose of comparison, as in parts 7.2 and 7.3, the same signal in the equation (7.2), having the harmonic content given in table 7.1, is used. The proposed algorithm follows the iterative procedure as in the previous hybrid algorithms; in the phase estimation, differently, instead of PSOPC and GA, Differential Evolution is utilized as the optimization algorithm.The used model of DE is the DE/best/1/bin which is given aby the equation (6.9). This version differently from standard DE/rand/1/bin model uses the best vector as the base vector in mutation stage. Using DE/best/1/bin model of DE as the optimization scheme, the procedure of the proposed algorithm is depicted as follows

In the first iteration, the target vectors representing the phase angles are initialized randomly in equation (4.21) and estimation of amplitudes is performed by Least

Squares Method as presented by equation (4.23). From estimated signals in equation (4.22) the cost function value of each vector is calculated and recorded. Then each mutant vector is constituted by adding a scaled difference of two random vectors to the best vector. By applying crossover to the target vectors and mutant vectors trial population is obtained. Then, by calculating the cost function of trial population and comparing with target vectors, the new target population is obtained. The phases in new target vectors are used in the next iteration.

In DE, trying to tune the three main control variables F and Cr and finding bounds for their values has been a topic of intensive research [63]. The rule of thumb values for the control variables given by Storn and Price [64]:

1. F → [0.5, 1.0]
2. Cr → [0.8, 1.0]
3. Np → 10.D

are valid for many practical purposes. These values are not strictly defined and still lack generality. Therefore, this does not mean that low values of Cr should always be avoided. Low values of Cr are advantageous for separable functions, since the search concentrates on the axes of the coordinate system as outlined in [23]. Gamperle [63] reported that the control variable settings for F, Cr, and Np can be quite difficult to find, and some objective functions are sensitive to the proper setting. Common parameters used in other population based algorithms are set as

Maximum number of generations, G =100

Size of the population Np=50

Dimension of each vector in the population D=5

For the selection of control parameters Cr and F, as one of the parameters is incremented, starting from 0.1 until 1, the other is kept fixed and for each combination program is executed for 1000 times.From the resulting 10x10 error index matrix it is seen that F=0.4 with Cr=0.7 gives the minimum average value. Estimated distorted waveforms and fundamental components, in different noise conditions are shown in figure 7.10 and figure 7.11 for average ε_1, ε_3 values, respectively.The average, maximum and minimum values of ε_1, ε_3 over 10000 executions are shown in tables 7.9 and 7.10 respectively. As seen from tables DE gives better performance over other algorithms and shows more reliable convergence.

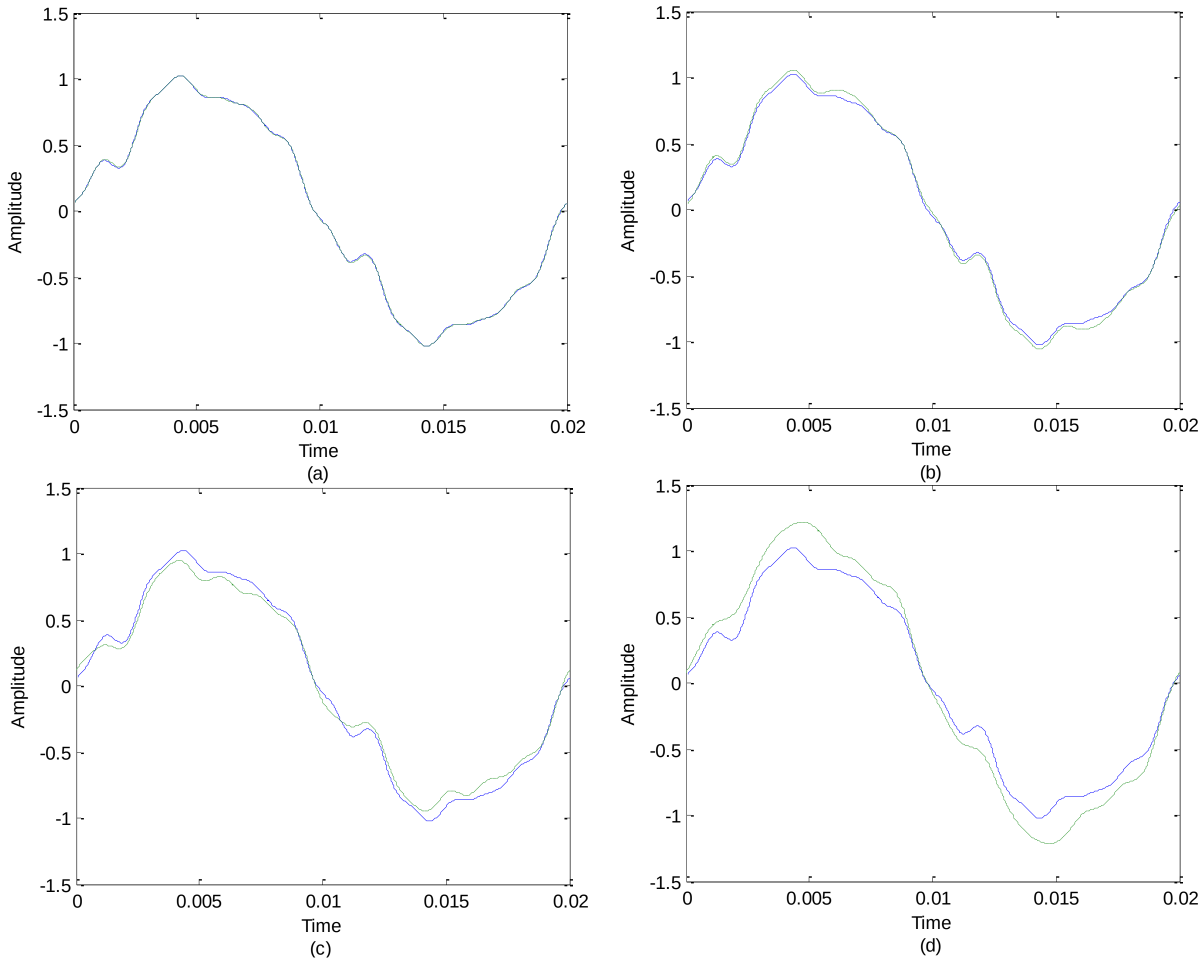

Figure 7.10 Estimated and actual distorted waveforms by in LS-DE based algorithm for (a) No noise (b) SNR = 20 dB (c) SNR = 10 dB (d) SNR = 0 dB

Table 7.9 Average, minimum and maximum ε_3 values for noisy and non-noisy conditions in LS-DE based algorithm

Noise Conditions	Error index ε_1		
	Average	Min	Max
No noise	$2.1974\text{x}10^{-5}$	$2.0189\text{x}10^{-31}$	0.0483
SNR = 20 dB	0.1409	0.0007	0.6760
SNR = 10dB	0.7151	0.0106	5.9489
SNR = 0dB	3.9520	0.0484	53.7733

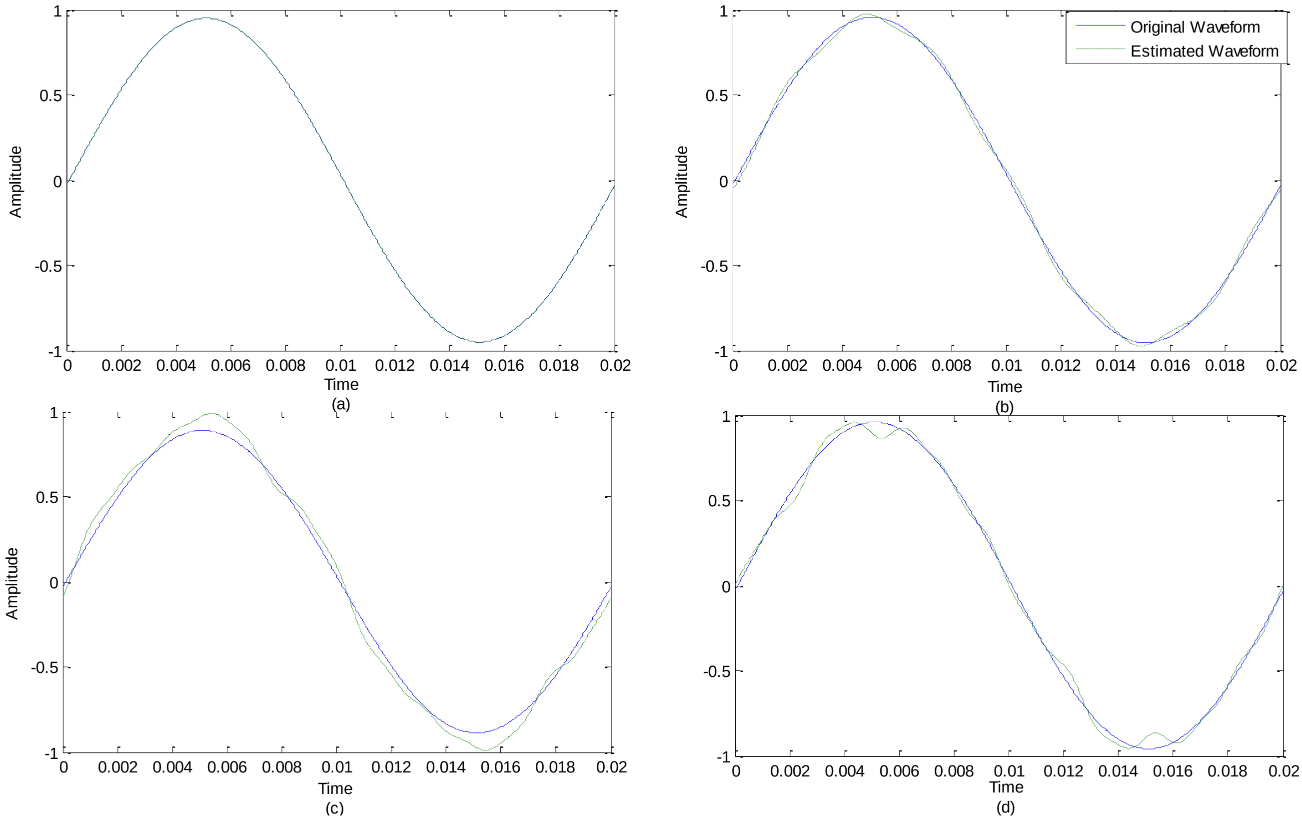

Figure 7.11 Estimated and actual fundamental components by LS-PSOPC based algorithm for (a) No noise (b) SNR = 20 dB (c) SNR = 10 dB (d) SNR = 0 dB

Table 7.10 Average, minimum and maximum ε_3 values for noisy and non-noisy conditions in LS-DE based algorithm

Noise Conditions	Error index ε_3		
	Average	Min	Max
No noise	$2.0814\text{x}10^{-5}$	$1.1595\text{x}10^{-30}$	0.0489
SNR = 20 dB	0.1079	0.0005	0.5412
SNR = 10dB	0.3889	0.0032	1.2814
SNR = 0dB	0.6009	0.0066	1.2997

The average amplitudes for 10000 runs of simulations are given in table 7.11.

Table 7.11 Actual and average of estimated amplitudes in LS-DE based algorithm for (a) No noise (b) SNR = 20 dB (c) SNR = 10 dB (d) SNR = 0 dB

Harmonic Order	Actual Amplitudes	Estimated Amplitudes			
		No Noise	20 dB	10 dB	0 dB
Fundamental (50Hz)	0.9500	0.9500	0.9501	0.9507	0.9574
5th (250 Hz)	0.0900	0.0900	0.0903	0.0823	0.0639
7th (350 Hz)	0.0430	0.0430	0.0421	0.0337	0.0292
11th (550 Hz)	0.0300	0.0300	0.0286	0.0216	0.0196
13th (650 Hz)	0.0330	0.0330	0.0310	0.0246	0.0214

7.4 Comparative Results

The minimum-maximum and average ε_1 values of three algorithms are given in table 7.12 and table 7.13 respectively.

Table 7.12 Minimum and maximum ε_1 values of the algorithms

Noise Conditions	GA		PSOPC		DE	
	Min	Max	Min	Max	Min	Max
No noise	2.9951×10^{-8}	0.1848	2.2363×10^{-14}	0.1881	2.0189×10^{-31}	0.0483
SNR=20dB	0.0010	0.7040	0.0007	0.6761	0.0007	0.06760
SNR=10dB	0.0112	6.1521	0.0110	5.9502	0.0106	5.9489
SNR=0dB	0.0780	53.8073	0.0500	53.7733	0.0484	53.7733

Table 7.13 Average ε_1 values of the algorithms

Noise Conditions	Error index ε_3		
	GA	PSOPC	DE
No noise	0.0316	0.0041	2.1974×10^{-5}
SNR = 20 dB	0.1702	0.1449	0.1409
SNR = 10dB	0.7561	0.7190	0.7151
SNR = 0dB	4.0917	3.9593	3.9520

The minimum-maximum and average ε_3 values of three algorithms are given in table 7.13 and table 7.14 respectively.

Table 7.14 Minimum and maximum ε_3 values of the algorithms

Noise Conditions	GA		PSOPC		DE	
	Min	Max	Min	Max	Min	Max
No noise	3.7896×10^{-9}	0.0624	1.9389×10^{-11}	0.1905	1.1595×10^{-30}	0.0489
SNR=20dB	0.0005	0.5490	0.0005	0.5419	0.0005	0.5412
SNR=10dB	0.0077	1.4814	0.0032	1.3056	0.0032	1.2814
SNR=0dB	0.0128	6.0948	0.0066	1.8053	0.0066	1.2997

Table 7.15 Average ε_3 values of the algorithms

Noise Conditions	Error index ε_3		
	GA	PSOPC	DE
No noise	0.0109	0.0042	2.0814×10^{-5}
SNR = 20 dB	0.1176	0.1119	0.1079
SNR = 10dB	0.4099	0.3928	0.3889
SNR = 0dB	0.7121	0.6084	0.6009

8. CONCLUSION

In this thesis, three hybrid algorithms which use Least-Squares Method for amplitude estimation and evolutionary computation algorithms GA, PSOPC and DE for phase angle estimation are applied respectively. These algorithms utilize the structural property of the signals containing harmonics, which states that the harmonic estimation problem is linear in amplitude and nonlinear in phase. Based on this property, algorithms proceed in an iterative way; after the phase angles are estimated with an evolutionary computation algorithm, amplitudes are simply calculated by Least Squares Method and at each iteration phases are updated according to the cost function to be used in the next iteration. Two of the algorithms which use GA and PSOPC, were studied in papers [18,19] previously. Third novel algorithm utilizing DE/best/1/bin model of DE is presented in this thesis. Simulations are realized in Matlab environment on the generated sample test signal, which was used in [18;19] previously. In three algorithms, common population based algorithm parameters are set to same values; number of generations is set as 100 and population size is set as 50. The algorithms are tested for noisy and non-noisy conditions. For noisy conditions SNR values are chosen as 0 dB, 10 dB and 20 dB respectively. In order to observe convergence properties simulations are ran 10000 times for each condtion. For each noisy situation by adding white gaussian noise to original sample signal, 10000 noisy signals are obtained and the same signals are used in three algorithms for the purpose of comparison.

In all three algorithms, the estimation model is based on the approximation of the distorted signal by minimizing the difference between sampled distorted signal $Z(k) = Z_o(k) + v(k)$ and estimated signal $\hat{Z}(k)$. Thus in non-noisy conditon, for performance measure of estimation of fundamental component, error index of distorted signal, ε_1 , can be used. However, it is observed that in different noisy - conditions a lower ε_1 does not always ensure a lower ε_2 or vice versa correspondingly. Therefore, in this thesis, instead of ε_2 used in [19], ε_3 is defiend as the error index for the fundamental component which is calculated by extraction of estimated harmonics from the test signal and it is proposed to evaluate ε_3 as an performance measure for the fundamental component. Therefore, as a performance measure error index ε_1(equation (7.5)) is used for the estimation of the distorted signal and ε_3(equation (7.10)) is used for the estimation of the fundamental component.

Comparative simulation results of the algorithms, showing the minimum-maximum and average error index values for the distorted signal are given in table 7.12 and table 7.13 respectively. It is seen that GA has the worst performance in all conditions. Average values obtained from 10000 trials for each case, show that DE ensures better convergence than PSOPC in all conditons. When minimum and maximum values obtained from 10000 trials, are considered, PSOPC and DE, when SNR = 20 dB have an equal minimum and when SNR = 0 dB have an equal maximum. All other values of DE is better than PSOPC.

The minimum-maximum and average error index values for the fundamental component are given in table 7.14 and table 7.15, respectively. GA only achieves an equal minimum when SNR = 20 dB, in other conditions has worse performance than PSOPC and DE. When rate of average error index performances are considered, DE has a significant superiority over other algorithms when there is no noise. In high noisy conditions DE has a closer performance to other algorithms, but is still better.Considering the minimum and maximum values, it is seen that in noisy conditions the best values obtained by DE and PSOPC are same, however when the maximum values are considered DE has a lower than PSOPC.

The simulation work is carried out on a PC with a 2.93GHz Intel Core 2 Duo CPU and 2.00-GB RAM. Average time for the execution of the LS-PSOPC based algorithm on the sampled test signal is recorded as 1.1419 seconds. In [19] LS-PSOPC based algorithm was introduced as a more computationally efficient algorithm than LS- GA based algorithm. Simulation results show that the average computation time for LS-DE algorithm is 0.9620 seconds. Therefore, when compared with LS-PSOPC based algorithm, it is seen that LS-DE based algorithm also improves the computation time with %15.75. However due to the computation time, the algorithm is applicable off-line like PSOPC and GA.

As a result,

- For the applied algorithms differentiability or continuity are not necessary. As such, they are flexible and can be adapted to different cases for nonlinear optimization of phase angles in harmonic estimation. The only configuration will be made in the programs for different harmonics, is just to specify the harmonic orders to be estimated
- DE is a simple algorithm, needs few operations and has few control parameters; only Cr and F values are needed to be examined. Hence, it is easy to apply.
- PSOPC has an better performance than GA
- DE outperforms PSOPC and GA in all conditions, significantly when there is no noise
- DE requires less computation time than PSOPC.

In conclusion, in this thesis a new hybrid algorithm which can accurately estimate the amplitudes and phases of the harmonics contained in a voltage or current waveform for PQ monitoring, is presented .The estimation process is iterative and in each iteration, the algorithm first applies DE to estimate the phases, then calculates the amplitudes using the LS. DE is robust, converges fast, easy to apply and adapt, with Least Squares Method to different cases in harmonic estimation. Multiple trials are used to investigate the on-average performance of the algorithms and simulation results show that hybrid method utilizing DE outperforms the other algorithms. Therefore, it is proposed to use LS-DE based hybrid algorithm for harmonic estimation.

REFERENCES

[1] Power Quality in Electrical Machines and Power Systems, EwaldF.Fuchs, Mohommad A.S.Masoum, 2008.

[2] H. C. Lin, "Intelligent neural network-based fast power system harmonic detection," IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 43–52,Feb. 2007.

[3] F. F. Costa, A. J. M. Cardoso, and D. A. Fernandes, "Harmonic analysis based on kalman filtering and prony's method," in Proc. Int. Conf. Power Engineering, Energy Electrical Drives, Setúbal, Portugal, Apr. 12-14, 2007, pp. 696–701.

[4] H. Ma and A. A. Girgis, "Identification and tracking of harmonic sources in a power system using kalman filter," IEEE Trans. Power Del., vol. 11, no. 4, pp. 1659–1665, Oct. 1996.

[5] Hopgood, A.A., Intelligent Systems for Engineers and Scientists CRC Press, 2001.

[6] Scala, M. L., Trovato, M., and Torelli, F.: 'A neural network-based method for voltage security monitoring', IEEE Trans. Power Syst. , 11, (3), pp. 1332-1341, 1996.

[7] Abdeslam D. O., Wira P., Merckl J., Flieller D., Chapuis Y. A., A unified artificial neural network architecture for active power filters, IEEE Transactions on Industrial Electronics, 54(1), p. 61-76, 2007.

[8] J. Mazumdar, R. G. Harley, F. C. Lambert, and G. K. Venayagamoorthy, "Neural network based method for predicting nonlinear load harmonics," IEEE Trans. Power Electron., vol. 22, no. 3, pp. 1036–1045, May 2007.

[9] J.L. Flores, P. Salmerón, "Harmonic detection by using different artificial neural network topologies", International Conference on Renewable Energy and Power Quality (ICREPQ), Vigo (Spain), April 2003.

[10] L. L. Lai , W. L. Chan and A. T. P. So "A two-ANN approach to frequency and harmonic evaluation", Proc. 5th Int. Conf. Artif. Neural Netw., p.245 , 1997.

[11] Hartana, R.K., Gill, G. Richards, "Harmonic source monitoring and identification using neural networks" IEEE Transactions on Power Systems, vol. 5, No. 4.pp.1098-1104, Nov. 1990.

[12] El-Amin and I. Arafah, "Artificial neural networks for power systems harmonic estimation", Proc.8th Int.Conf. Harmonics Quality Power, vol. 2, pp.999-1009, 1998.

[13] Osowski, S, “Neural Network for Estimation of Harmonic Components in a Power System,” IEE Proceedings-C, Vol. 139, No. 2, pp. 129-135, March 1992.

[14] Pecharanin, N., Mitsui, H., and Sone, M.: ‘Harmonic detection using neural network’. Proc. IEEE Intl. Conf. On Neural Networks, Perth, Australia, pp. 923–926, 1995.

[15] A. A. M. Zin, M. Rukonuzzaman, H. Shaibon, and K. I. Lo, “Neural network approach of harmonics detection,” in Proc. Int. Conf. Energy Manage. Power Delivery (EMPD), Singapore, vol. 2, pp. 467–472, Mar. 3–5, 1998.

[16] U. Qidwai and M. Bettayeb, “GA-Based nonlinear harmonic estimation,” IEEE Trans. Power Delivery, Dec. 1998.

[17] Mishra, S.: A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. IEEE Trans. on Evolutionary Computation, vol. 9(1): 61 73, 2005.

[18] M. Bettayeb and U. Qidwai, “A hybrid least square-GA-based algorithm for harmonic estimation,” IEEE Trans. Power Del., vol. 18, no. 2, pp. 377–382, Apr. 2003.

[19] Lu, Z., Ji, T.Y., Tang, W.H., Wu, Q.H.: ‘Optimal harmonic estimation using a particle swarm optimizer’, IEEE Trans. Power Deliv., 23, (2), pp. 1166–1174

[20] Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 2008.

[21] Kennedy, J., Eberhart, R.C.: ‘Particle swarm optimization’. Proc. IEEE Int. Conf. on Neural Networks, Piscataway, USA, vol. 4, pp. 1942–1948, , 27 November–1 December 1995.

[22] Storn, R., Price, K.V.: Differential Evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11(4), 341–359 (1997)

[23] Chakraborty, U.K., ed. , Advances in Differential Evolution, Springer, 2008,

[24] J. Vesterstroem and R. Thomsen, “A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems,” in Proc. IEEE Congr. Evolutionary Computation, Portland, OR, pp. 1980–1987, Jun. 20–23, 2004,

[25] Dugan,R.C., Mc Granaghan,M.F., Santoso,S., Electrical Power Systems Quality, Mc GrawHill, 2003.

[26] Standard ANSI C84.1.

[27] IEEE Standard 519-1992: Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems.

[28] Standard IEC 61000

[29] Standard IEEE-1159

[30] Harmonics and Power Systems, Francisco C. De La Rosa, CRC Press, 2006

[31] C. Sankaran, Power Quality, CRC Press, 2002

[32] Baggini, Angelo: Handbook of power quality. West. Sussex, John Wiley and Sons, 2008.

[33] R. S. Vedam and M. S. Sarma, “Power Quality VAR Compensation in Power Systems,” CRC Press, Boca Raton, 2009

[34] Fourier, J.B.J., Theorie analytique de la chaleur, Paris, 1822.

[35] Arrillaga, J., Watson, Neville R., Wood, Alan R., Smith, B.C., Power System Harmonic Analysis, John Wiley & Sons, New York, 1997

[36] A. Girgis and F. Ham, A qualitative study of pitfalls in FFT. IEEE Trans. Aerospace and Electronic Systems AES-16 4, pp. 434–439, 1980.

[37] G.D. Bergland, "A guided tour of the fast Fourier transform," in Digital Signal Processing, L.R. Rabiner and C.M. Raader, Eds. New York: IEEE Press, pp. 228-239, 1972.

[38] Bretscher, Otto ,Linear Algebra With Applications, 3rd ed.. Upper Saddle River NJ: Prentice Hall, 1995.

[39] Mayr, E., Towards a New Philosophy of Biology: Observations of an Evolutionist.Belknap Press, Cambridge, MA, 1988.

[40] Hartl, D.L. and Clark, A.G., Principles of Population Genetics, 2nd edn. Sinauer,Sunderland, MA, 1989

[41] Goldberg, David E.,The Design of Innovation: Lessons from and for Competent Genetic Algorithms, Addison-Wesley, Reading, MA, 2002.

[42] M. Megnevitsky. Artifical Intelligence: A Guide to Intelligent Systems. New York:Addison-Wesley / Pearson Education Limited, 2002.
[43] K. A. De Jong, Evolutionary Computation: A Unified Approach, the MIT Press 2006.
[44] Goldberg D.E., Genetic Algorithms in Search Optimization and Machine Learning (Reading, MA: Addison-Wesley), 1989.
[45] F. Karray, C. De Silva, Soft Computing and Intelligent Systems Design: Theory, Tools, and Applications, Addison-Wesley Publishing, 2004.
[46] Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. Proceedings of the sixth international symposium on micro machine and human science pp. 39-43. IEEE service center, Piscataway, NJ, Nagoya, Japan, 1995.
[47] http://www.swarmintelligence.org/tutorials.php
[48] Eberhart, R. C. and Shi, Y. Comparison between genetic algorithms and particle swarm optimization. Evolutionary programming vii: proc. 7th ann. conf. on evolutionary conf., Springer-Verlag, Berlin, San Diego, CA., 1998.
[49] Poli, R. "Analysis of the publications on the applications of particle swarm optimisation". Journal of Artificial Evolution and Applications, 2008
[50] Eberhart, R. C., & Shi, Y. (2001, May 27-30). Particle swarm optimisation: Developments, applications, and resources. Proceedings of the IEEE Congress on Evolutionary Computation, Seoul, South Korea (pp. 81-86).
[51] Y. Shi and R. C. Eberhart, “A modified particle swarm optimiser,” in Proc. IEEE Int. Conf. Evolutionary Computation, Piscataway, NJ, pp. 69–73, 1998.
[52] Y. Shi and R. Eberhart, “Parameter selection in particle swarm optimization,” in Proc. 7th Int. Conf. Evol. Program. (EP), pp. 591–600, May 1998.
[53] Y. Shi and R. Eberhart, “Empirical study of particle swarm optimization,” in Proc. IEEE Congr. Evol. Comput., vol. 3, pp. 1945–1950, Jul. 1999.
[54] Y. d. Valle , G. K. Venayagamoorthy , S. Mohagheghi , J.-C. Hernandez and R. G. Harley "Particle swarm optimization: Basic concepts, variants and applications in power systems", IEEE Trans. E Comput, vol. 12, p.171 , 2008.
[55] Schutte JF, Reinbolt JA, Fregly BJ, Haftka RT, George AD. Parallel global optimization with particle swarm algorithm. International Journal for Numerical Methods in Engineering,. 2004.

[56] Parrish, J.K., Hamner,W.M., Animal Groups in Three Dimensions. Cambridge University Press, Cambridge, UK, 1997.

[57] Gordon, D.M., Paul, R.E., Thorpe, K., What is the function of encounter pattern in ant colonies? Anim. Behav. 45, 1083–1100, 1993.

[58] S. He, Q. H.Wu, J. Y.Wen, J. R. Saunders, and R. C. Paton, “A particle swarm optimizer with passive congregation,” BioSyst., vol. 78, no. 1–3, pp. 135–147, Dec. 2004.

[59] Storn R, Price KV, Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI, 1995.

[60] Storn, R., Price, K.V.: Minimizing the real functions of the ICEC 1996 contest by differential evolution. In: Proceedings of the 1996 IEEE international conference on evolutionary computation, Nagoya, Japan, pp. 842–844. IEEE Press, New York 1996.

[61] Price K, Storn R, Differential evolution: a simple evolution strategy for fast optimization. Dr. Dobb’s Journal 22(April):18–24, 1997.

[62] Storn, R., Price, K.V.: Differential Evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11(4), 341–359, 1997.

[63] Price, K., Storn, R., Lampinen, J.: Differential Evolution – A Practical Approach to Global Optimization. Springer, Heidelberg, 2005.

[64] Gamperle, R., Müller, S.D., Koumoutsakos, P.: A Parameter Study for Differential Evolution. In: Grmela, A., Mastorakis, N.E. (eds.) Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, pp. 293–298. WSEAS Press, 2002.