MULTI PARTICLE SWARM OPTIMISATION ALGORITHM APPLIED TO SUPERVISORY POWER CONTROL SYSTEMS

by

Abdulhafid Faraj Sallama

Doctor of Philosophy

Department of Electronic and Computer Engineering

College of Engineering, Design and Physical Sciences

Brunel University London

October 2014

Abstract

Power quality problems come in numerous forms (commonly spikes, surges, sags, outages and harmonics) and their resolution can cost from a few hundred to millions of pounds, depending on the size and type of problem experienced by the power network. They are commonly experienced as burnt-out motors, corrupt data on hard drives, unnecessary downtime and increased maintenance costs. In order to minimise such events, the network can be monitored and controlled with a specific control regime to deal with particular faults. This study developed a control and Optimisation system and applied it to the stability of electrical power networks using artificial intelligence techniques.

An intelligent controller was designed to control and optimise simulated models for electrical system power stability. Fuzzy logic controller controlled the power generation, while particle swarm Optimisation (PSO) techniques optimised the system's power quality in normal operation conditions and after faults. Different types of PSO were tested, then a multi-swarm (M-PSO) system was developed to give better Optimisation results in terms of accuracy and convergence speed.. The developed Optimisation algorithm was tested on seven benchmarks and compared to the other types of single PSOs.

The developed controller and Optimisation algorithm was applied to power system stability control. Two power electrical network models were used (with two and four generators), controlled by fuzzy logic controllers tuned using the Optimisation algorithm. The system selected the optimal controller parameters automatically for normal and fault conditions during the operation of the power network. Multi objective cost function was used based on minimising the recovery time, overshoot, and steady state error. A supervisory control layer was introduced to detect and diagnose faults then apply the correct controller parameters. Different fault scenarios were used to test the system performance. The results indicate the great potential of the proposed power system stabiliser as a superior tool compared to conventional control systems.

Table of Contents

Acknowledgements

All praise is due to Allah (Glorified and Exalted is He), without whose immeasurable blessings and favours (with the prayers of parents, spouse sons and friends) none of this could have been possible.

Firstly I would like to express my sincere appreciation and gratitude to all those who made this thesis possible. This work would not have been possible without help, support and strong motivation of my supervisor, Dr Maysam Abbod, who also gave me the opportunity to work with him in his lab research module and his encouragement, guidance and support from the beginning to the end made my work much easier and more enjoyable.

Also, I would like to express my deepest appreciation to my second supervisor, Prof Gareth Taylor for his important advice and guidance. I will also extend my hand to Late Dr Peter Turner who gave me the first opportunity to work with him in the research lab.

I am heartily thankful to Dr Mohamed Darwish for his support and good spirit which have boosted my moral. I don't forget Dr Mohammed Ghazi Gronfula who supported me with his knowledge.

I would also like to thank my colleagues especially Basem Alamri, Khaled Sehil, Yaminidhar Bhavanam, Mohammed Alqarni and Shariq Khan who were join me in productive discussions and pushed me to strive towards my goal, and for all their fascinating discussions about the area and the work undertaken.

Lastly, I thank my family for their support, love and encouragement throughout the period. Also I express my sincere thanks to all my teachers, relatives, colleagues, elders and all those from whom I have learnt and gained knowledge.

Declaration

I certify that the effort in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree. I also certify that the work in this thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been duly acknowledged and referenced.

Signature of Student

Abdulhafid Faraj Sallama

October 2014, London

List of Abbreviation

Abbreviations	Description
AC	Alternating Current
AEM	Abnormal Event Management
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANFIS-PSS	Adaptive Neuro-Fuzzy Inference System- Power System Stabilisers
APSO	Adaptive Particle Swarm Optimization
BN	Big Negative
BP	Big Positive
CLPSO	Comprehensive Learning Particle Swarm Optimization
COPSS	Control And Optimization Of Power System Stabilisation
CPSS	Conventional Power System Stabiliser
CPU	Central Processing Unit
DE	Dispatch Economic
DED	Dynamic Economic Dispatch
DGs	Distributed Generators
DMS-PSO	Dynamic Multi-Swarm Particle Swarm Optimization
DMS-PSO-SHS	DMS-PSO With Sub-Regional Harmony Search
EC	Evolutionary Computation
ED	Economic Dispatch
EP	Evolutionary Programing
ES	Evolution Strategies
FACTS	Flexible Ac Transmission Systems
FDD	Fault Detection And Diagnosis
FLC	Fuzzy Logic Controller
FLC_{AT}	Fuzzy Logic Controller Auto-Trained
FLC_{MT}	Fuzzy Logic Controller Manually Tuned
FPSS	Fuzzy Logic Controller Power System Stabiliser
GA	Genetic Algorithms
GENCOs	Generation Expansion Companies
GEP	Generation Expansion Problem
GPSO	Global Version Of Particle Swarm Optimization
IEEE	Institute Of Electrical And Electronic Engineers
LPSO	Local Version Of Particle Swarm Optimization
LPSSs	Local Power System Stabilisers

MB-PSS	Multiband Power System Stabilizers
MF	Membership Function
M-PSO	Multi-Swarm Particle Swarm Optimization
N	Negative
NF	Neuro Fuzzy Logic
NN	Neural Network
OPF	Optimal Power Flow
OPLF	Optimal Power And Load Flow
P	Positive
PID	Proportional Integral Derivative
PPSO	Parallel Particle Swarm Optimisation
PSO	Particle Swarm Optimization
PSS2B	Power System Stabiliser Type 2B
PSS4B	Power System Stabiliser Type 4B
PSSs	Power System Stabilisers
Smf	S-Shaped Membership Function
SPSO	Sequential Particle Swarm Optimisation
SPSSC	Supervisory Power System Stability Controller
SPSSC	Supervisory Power System Stability Controller
SSMDFLC	Supervisory Self-Monitors And Decision Fuzzy Logic Control
STAs	Stochastic Technique Algorithms
STS	Stochastic Time Series
SVC	Static Var Compensator
TSC	Thyristor Switched Capacitor
TSR	Thyristor Switched Reactor
UPSO	Unified Particle Swarm Optimization
VCC	Voltage/Var Control
WSM	Weighted Sum Method
Z	Zero
Zmf	Z-Shaped Membership Function

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

Power system stabilisers (PSSs) have been utilized for decades to deliver damping to system oscillations. Local, conservatively distributed local PSSs (LPSSs) are deliberated to have fixed parameters gathered from a stabilised model around a certain operating point. Final configurations are prepared by field tests at one or two operating points. A major source of model fluctuation and disturbance can be caused by the inherent nonlinearity in the system. Power systems operate in a changing and highly non-linear environment as a result of change in loads, key operating parameters and generator output. When the system is exposed to a disturbance, the stability of the system will be a subject of the nature of the disturbance as well as the initial operation condition (Kundur et al., 2004).

In this thesis, the term *uncertainty* concerns the lack of precision in modelling of all elements in electrical power grid containing the transformers, the transmission lines, and the loads, in addition to the fluctuation of clear linearized power plant control factors as the operating point fluctuates. LPSS parameters based on any single model are probably not optimal and may limit the linearizing impact. Nonetheless, in the developed system negligible errors can be minimised, particularly if the damping controller is designed on the basis of the robustness standards and the closed-loop system preserves an acceptable performance level. Some modifications should be applied to the system of power system controllers, particularly PSSs utilising Optimisation techniques in the case of disturbances and failures (Snyder *et al.*, 1999) (Klein *et al.*, 1995) . The likelihood of strong coupling among the local modes and the interarea modes would render the tuning of LPSSs for damping all modes nearly unachievable when there is no supervisory level controller. The intelligent system has been utilised to organise multiple local controllers in electrical power system (Guo, Hill, & Wang, 2001).

Intelligent systems are tools and methodologies inspired by nature to solve computationally intensive problems in mathematics that are very important for the progress of current trends in survey and information technology. Artificially intelligent systems these days utilize computers to be easily emulate various

biological metaphors and faculties of human intelligence. They use techniques which combine the symbolic and sub-symbolic systems capable of *developing* human knowledge abilities and intelligence, not merely to do things that humans can do well. Intelligent systems are ideally suited for tasks s9uch as optimisation and search, pattern recognition and adaptation, planning, vagueness management, control and adjustment. In this thesis, the technologies of intelligent systems and their implementation are highlighted by a series of examples.

In the domain of artificial intelligence, neuro fuzzy logic (NF) denotes combinations of fuzzy logic and artificial neural networks. NF systems use a self-learning algorithm derived and inspired by the concept of neural networks to achieve the use of processed data samples (their fuzzy sets and fuzzy rules) (J. Jang, 1993a). However, in this research using this methodology to develop stability control system is integrated with an Optimisation methodology such as particle swarm Optimisation (PSO) (Kennedy & Eberhart, 1995b) to determine the optimum supervisory control parameters. Furthermore, an accurate electrical stability control system element model adapted for fast response, overshooting, fluctuation and to eliminate the steady state error was developed using standard data for Optimisation purposes.

This research explores different types of Particle Swarm Optimisation (PSO) as intelligent Optimisation methodologies for the purpose of emphasis on the application to tune the power stability control elements and hierarchical control systems. Different types of PSOs give different Optimisation results in terms of accuracy and convergence speed. The advantage of the ones characterises in quick convergence and low accuracy can be combined with those that have slow convergence but accurate results. The new algorithm will be a combination of fast and accurate optimisers such that the benefit of both system are utilised. The developed optimiser can be tested on different benchmark.

1.2 Motivations

The motivation for this thesis is two-fold:

First, the need for evolutionary algorithms to optimise nonlinear or change state frequently problems. Scaling the possibility of Optimisation algorithm programs to find the best solutions for that type of complex problems via using computing

systems and high-tech solutions whereby evolutionary algorithms are applied to solve static problems; however, many real-world systems are often continuous or change state frequently. The system state changes necessitate recurrent (sometimes frequent or continuous) re-Optimisation. It has been shown that the Optimisation of the particle swarm can be applied successfully for dynamic control and optimisation systems (Eberhart & Shi, 2001a). Therefore, there is great need for an efficient algorithm that is able to solve real, complex, multi-dimension problems. In fact to develop such an algorithm, several models have to be developed, such as different types of PSO optimisers, and appropriate benchmark functions should be selected to test new algorithm and power system stability control with supervisory control in electrical network.

This thesis presents intelligence algorithms capable of dealing with the intricacy of electrical power system problems using PSOs algorithm methodologies, which are considered to be robust adaptive Optimisation techniques. Furthermore, an effective algorithm based on algorithms portfolio procedure can be developed using different types of PSOs algorithms techniques whereby the communication between these algorithms in the early stages can be considered (the fast and less accurate algorithm can pass its results to the slow and more accurate algorithm, which will consequently benefit from the good results at an early stage).

Second, the electrical power energy market is evolving with the progressive growth of networks and a continuous improvement in performance, rapidly increasing the number of consumers and the critical need for sophisticated control equipment governing generation plants to produce high quality energy in terms of continuity, stability, reliability, flexibility and quick response. Furthermore, the rapid development in the use of the solar panels, wind turbines and other sustainable energy sources, which are mostly turbulent and unpredictable sources, requires advanced control devices.

Steady state stability or power transient is defined as the ability of a power system to control stably without loss of synchronisation between power plant generations after a small or large disturbance. Moreover, voltage stability is the system ability to maintain load voltage magnitudes under steady state conditions within specified operating limits. Since it has become increasingly difficult to obtain power plant sites

in the vicinity of power consumers (due to the lack of space for large power facilities in urban areas and public opposition to power infrastructure), electrical power is now often transported over long distances using large capacity lines. Under these circumstances, voltage stability can be a major problem, as well as transient and steady state stabilities (Abe, Fukunaga, Isono, & Kondo, 1982).

1.3 Aim and Objectives

The overall aim of the thesis is to introduce and design a new controller device working on the principle of neurons fuzzy logic and artificial intelligence, using a new Optimisation algorithm as a novel paradigm in the stability and supervision of electrical network.

The research aim is addressed through the following objectives:

1. To review the area of searching techniques and intelligent Optimisation algorithms.
2. To review different types of PSOs algorithms as stochastic swarm intelligence technique.
3. To review different types of linear and non-linear functions to be used as benchmark.
4. To introduce new combination of multi particle swarm Optimisation paradigm.
5. To review power system stability and the most important devices currently used for this purpose.
6. To apply the new intelligent Optimisation algorithm in power system stability.
7. To scale-up the control system and development a new supervisory stability control system.

1.4 Challenges

There are many technical challenges in developing the control device and supervision of the electrical power system by the use of artificial intelligence technique. These challenges are inherited from the original components of stability control with the following considerations:

- In real-world applications, optimising problems is much more complex because of various constraints, objectives and the size of the search space involved, in relation to different types of controlling.
- There is a need to develop a system that is able to control electrical power system stability in networks with high-quality continuity, stability, reliability, flexibility and quick response.
- Stability control system should handle any phenomenon causing instability in the electrical network using artificial intelligent techniques.
- An efficient electrical network containing fuzzy logic controller model administrating stability can be used for Optimisation purposes.
- There is a need for an efficient algorithm that can solve and optimise the parameters problems of multi controller network.
- A suitable technique should measure the quality of multiple objectives.
- A new PSOs technique that can guarantee feasible solutions is needed.
- A hierarchical control system should detect the system frailer and find immediate solutions using new Optimisation algorithm and artificial intelligent paradigm.

1.5 Contributions to Knowledge

This research developed a multi PSO intelligent Optimisation program. This algorithm includes a combination of different types of PSOs, such as Local Version of Particle Swarm Optimisation (LPSO), Global Version of Particle Swarm Optimisation (GPSO), Dynamic Multi-Swarm Particle Swarm Optimisation DMS-PSO with Sub-regional Harmony Search (DMS-PSO-SHS), Adaptive Particle Swarm Optimisation (APSO) and others. Moreover, an effective paradigm was developed using all previous algorithms based on portfolio methodology in two patterns, parallel and serial. In this manner of communication between different algorithms is considered in the early stages, whereby the fast convergence and less accurate algorithm can pass its results to the slow and more accurate algorithm, which will benefit from the good results at an early stage.

Moreover, a number of different types of benchmark functions evaluate each algorithm individually, and the results can be compared together in addition to the

new optimal algorithm. This analytical technique enables identification of the best paradigm to discover the different crossover effects and a practical manner to avoid the generation of useless solutions in extensive systems.

A fuzzy logic controller model was developed and auto tuned for Optimisation purposes. This model provides a new stability controller type, using NF and PSO procedures as expression of artificial intelligence to optimise the controller scale of factors. Subsequently, an altogether successful model of an advanced control system was developed.

The intelligent supervisory system that is able to control stability in electrical network when facing any disturbances and during normal operation with high efficiency in power quality causes the least trouble for consumers. This system deals with multiple objectives and measures the fitness of each solution that is generated by the system. The system used Weighted Sum Method (WSM) techniques and the hierarchy technique to address the problem of the stability control in hierarchy promised stages.

1.6 Thesis Outline

The work presented in this thesis is organised into seven chapters. Figure 1.1 illustrates the structure of the thesis and relationship to the thesis objectives presented in section 1.3.

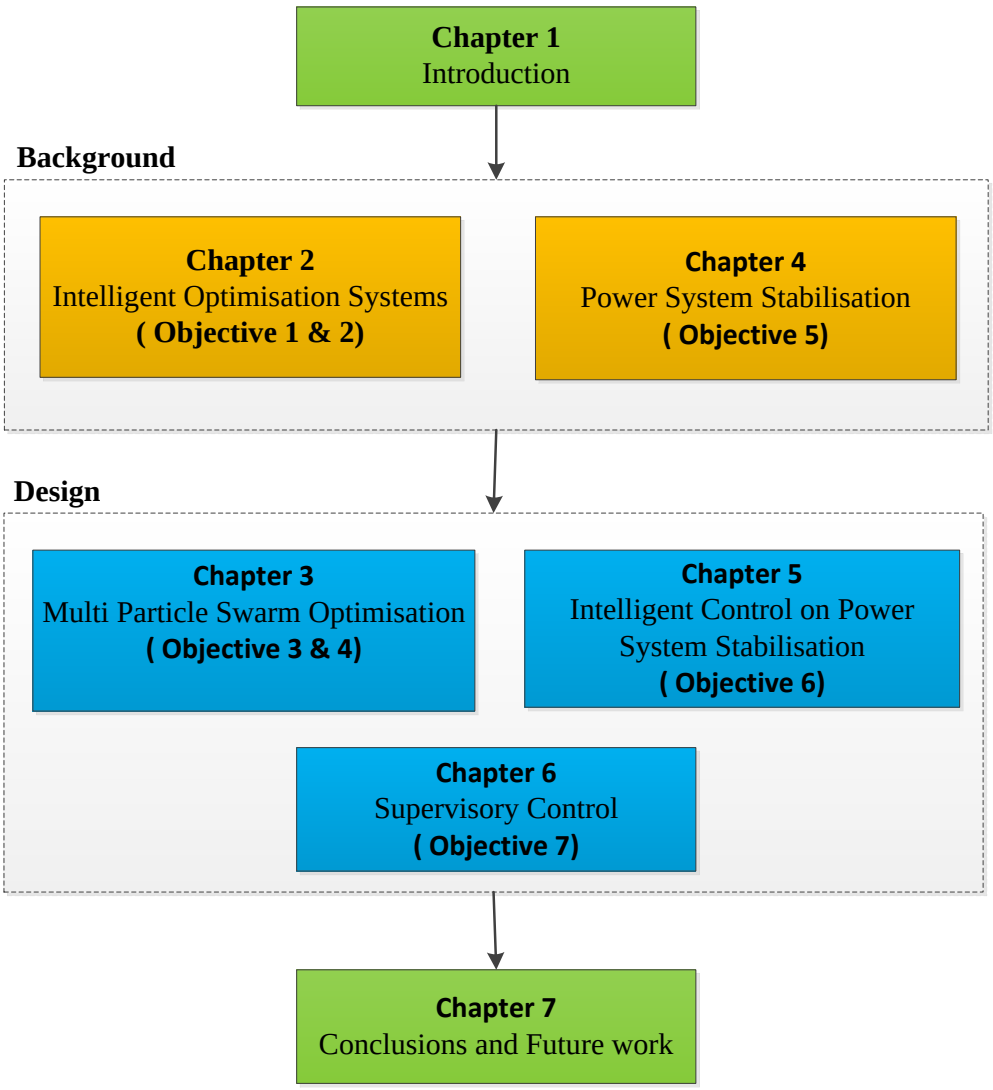

Figure 1.1: Thesis structure.

Following this introductory chapter, the next six chapters contain more detailed information about the theoretical background and technical development of artificial intelligent and power stability devices.

Chapter 2 presents a detailed background about intelligent Optimisation systems and their evolution. It explores how Optimisation tools have become an important part of life to solve constrained and unconstrained continuous and discrete problems, by providing generic algorithms as part of stochastic techniques. The chapter reviews and compares related paradigms intelligent research methods. The chapter concludes with a brief summary and discussion.

Chapter 3 introduces the combination of MPSO and outlines a group of complex functions to be used in benchmark testing. A new method is proposed using different Optimisation schemes which are combinations of different types of PSO and methods used simultaneously, to make use of the characteristics of each individual method to tackle the same problem. The proposed schemes can utilize and share information about the local and global best as well as the swarm population, in specific predefined iteration of all types simultaneously.

Chapter 4 lays the background for power system stabilisation by presenting the resource scheduling problem and the latest conventional device in this area. This chapter presents power system stability by discussing the most important four dynamic phenomena affecting it, namely wave, electromagnetic, electromechanical and thermodynamic phenomena. Furthermore, the relationship between reactive power voltage and stability is explored; this variable is one of the most important targets in the search, as an actual variable that indicates the state of the electrical power grid in terms of stability. Additionally, this chapter, through mathematical analysis, discusses the most important types of power system stability devices to learn the working methods, as well as tuning methods. The most important faults types which affect the power system stability of the electrical grid are then described, allowing with how these problems can be emulated by software programs for the purpose of analysis. Finally, the latest types of conventional power system stabiliser (PSS4B) devices are explained and tested using different conditions and compared to other devices from prior generations.

Chapter 5 applies the intelligent control and Optimisation on the power system stabilisation. Furthermore to describes the current state of the intelligent Control and Optimisation of Power System Stabilisation (COPSS). Furthermore, it explains the fuzzy logic controller and the design and tune stable control system using fuzzy logic controller before the specific approach in neuro fuzzy logic systems. The implementation of ANFIS-PSS controller and training them in different stages is described, involving single- and three-phase training. The new ANFIS-PSS controller response to ground fault in tie line in machines A and B is then explained and the power quality in the network is outlined before the auto tuning of scaling factors using intelligent Optimisation, and the simulation results of rotor speed

deviation on both machines A and B, in addition to comparing the power quality in the network.

Chapter 6 introduces in detail a new supervisory control describing the design and implementation of advanced Supervisory Power System Stability Controller (SPSSC) using neuro-fuzzy system and Matlab S-function tool, whereby the controller is taught from data generated by simulating the system for the optimal control regime. The controller is compared to a multi-band control system which is utilized to stabilize the system for different operating conditions. Simulation results show that the supervisory power system stability controller produced better control action in stabilizing the system for conditions such as: normal, after disturbance in the electrical grid as a result of changing of the plant capacity like switching renewable energy units, high load reduction or in the worst case of fault in operating the system, e.g. phase short circuit to ground. The new controller decreased the settling time and overshoot after disturbances, which means that the system can reach stability in the shortest time with minimum disruption. Such behaviour improves the quality of the provided power to the power grid.

Chapter 7 summarises the thesis aims, major contributions and significant findings. It highlights areas and directions for further research.

1.7 Author's Publications

A number of journal and conference papers related to this thesis have been published at international conferences and in journals, and some recent additional papers are pending acceptance for international conferences and journals, as presented below.

A. Conference papers (Published)

[1] A. Sallama and M. Abbod, "Neuro-Fuzzy System for Power Generation Quality" published on ISGT 2011, December 2011.

[2] A. Sallama, M. Abbod, P. Turner, "Neuro-Fuzzy System for Power Generation Quality Improvements" published on UPEC 2012, Brunel University, September 2012.

[3] A. Sallama, M. Abbod, P. Turner, "Intelligent Control System for Power System Stability" published on ResCon 2012, Control Engineering , Power System Second year of PhD.

[4] A. Sallama, M. Abbod, G Taylor, " Development Intelligent Control System for Power System Stability" published on ResCon 2013, Control Engineering , Power System third year of PhD.

[5] A. Sallama, M. Abbod, P. Turner, "Supervisory Power System Stability Control Using Neuro-Fuzzy System and Particle Swarm Optimisation Algorithm" published on UPEC 2014, Technical University of Cluj-Napoca, Romania, September 2014.

B. Conference papers (Accepted)

[6] B. Alamri, A. Sallama, M. Darwish, "Optimum SHE for cascaded H-Bridge multilevel inverter using: NR-GA-PSO, comparative study" ACDC 2015, The 11th International Conference on AC and DC Power Transmission, 10 - 12 February 2015, Birmingham, UK.

C. Journal papers (Published)

[7] Sallama and M. Abbod, "Applying Sequential Particle Swarm Optimisation Algorithm to Improve Power Generation Quality", *International Journal of Engineering and Technology Innovation*, 2014.

D. Journal papers (Accepted)

[8] Shariq Mahmood Khan, R.Nilavalan, Abdulhafid Sallama, "A Novel Approach for Reliable Route Discovery in Mobile Ad-Hoc Network" Wireless Personal Communications, October 2014.

Chapter 2: Intelligent Optimisation Systems

2.1 Introduction

Optimisation tools are becoming increasingly important in everyday life to solve constrained and unconstrained, continuous and discrete problems, by providing widely algorithms as part of stochastic technique algorithms (STAs), which are very effective in solving standard and large-scale Optimisation problems (Mahfoud, 1995), provided that the problem does not require multiple solutions (e.g. classification problems in machine learning where, in single run, multiple optima and peaks need to be found) (Koper, Wysession, & Wiens, 1999).

2.2 Optimisation Techniques

Optimisation has been an active area of research for several decades. As many real-world Optimisation problems become increasingly complex, better Optimisation algorithms are always needed. Unconstrained Optimisation problems can be formulated as *n* dimensional minimization, thus:

$$Min\ f(x), x = [x_1, x_2, \ldots\ldots, x_n] \tag{2.1}$$

where *n* is the number of the parameters to be optimized.

In case of searching for optimum solutions, using Optimisation techniques there are three main broad classes used to find the solution mentioned by Goldberg (Goldberg, 1990), as shown in Figure 2.1. The following subsections list the different types of Optimisation technique with a short description of each.

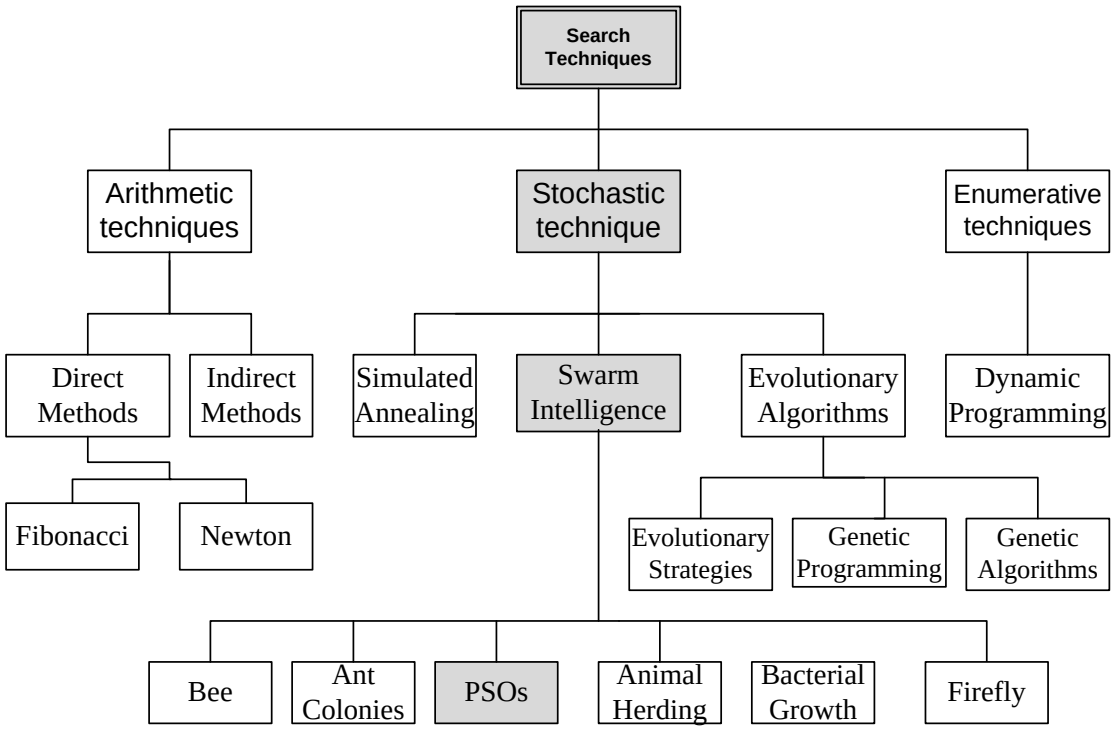

Figure 2.1: Search techniques.

2.2.1 Arithmetic Techniques

Arithmetic techniques can be divided into two types: direct and indirect. Direct ways, such as those of Newton and Fibonacci (Bóna, 2011), seek the greatest "jump" on the search space and evaluate the gradient of a new point approaching the solution. Indirect ways are conducted by solving a set of non-linear equations to search for local extreme values, typically caused by a gradient of the objective function equal to zero, and by restricting itself to points to search for possible solutions (function peaks) with zero slope in all directions.

2.2.2 Enumerative Techniques

This technique is very simple to implement, requiring high significance computation applied on applications with too large a domain space, because every point search to an objective function's at domain space (Ribeiro Filho, Treleaven, & Alippi, 1994). A good example for that technique is a dynamic programming.

2.2.3 Stochastic Search Technique

In this case, search techniques are based on enumerative techniques in addition to using more essential information to guide the random search processing. This is done by three major methods: evolutionary algorithms, intelligent swarm and simulated annealing. Intelligent algorithms use natural selection principles or simulate the natural particle swarm as in bird flocking or fish school phenomena (Kennedy, Kennedy, & Eberhart, 2001). Here, the solutions are improved features of possible solutions throughout generations by biological processes inspired in such techniques. Particle swarm Optimisation (PSO) and genetic algorithms (GA) are good examples for this technique. Simulated annealing used a thermodynamic evolution process to search for possible solutions (Schwefel, 1981).

2.3 Particle Swarm Optimisation (PSO)

Most creatures in nature behave as swarms. The study of artificial life is highly influenced by studies of natural swarm behaviour, which has been mapped and reconfigured in mathematical models which can be processed by computers (Liu & Passino, 2000). PSO mathematically requires only primitive mathematical operators that can be implemented by computer models and code in a few lines. This feature makes it inexpensive in terms of both memory and speed requirements. Furthermore, the PSO has been known as evolutionary computation technique (Kurian, George, Bhat, & Aithal, 2006a) , and have all the features of Evolution Strategies (ES), Genetic Algorithms (GA) and Other evolutionary computation (EC) techniques, such as utilizing some searching points in the solution space, similar to genetic algorithms (Eberhart & Shi, 1998) (Panduro, Brizuela, Balderas, & Acosta, 2009).

A GA system is initialized with a population of random solutions. While GA can handle combinatorial Optimisation problems, PSO has continuous Optimisation problems. In contrast to GA, each single population is also assigned a random speed in PSO, in effect to flying them through the solution hyperspace. Moreover, PSO has been enhanced in the combinatorial Optimisation problems, thus it is possible to simultaneously search the optimal solution in several dimensions, unlike other evolutionary computation techniques (Tandon, El-Mounayri, & Kishawy, 2002).

Several scientists (e.g. Heppner and Grenander, 1990; Reynolds, 1987) studied a particularity of synchronized movement without collision for the members of bird flocks and fish schools. Eberhart and Kennedy (1995) developed new evolutionary computation technique dubbed PSO, and Shi and Eberhart (1996) introduced the concept of inertia weight to the original version of PSO, in order to improve the search during the Optimisation process (Shi & Eberhart, 2001)(Eberhart & Shi, 2001b). Several types of developed PSO subsequently emerged, as explained in this section, in addition to the updated types such as those devised during this research.

2.3.1 Classical PSO Algorithm

PSO developed as a simulation of the bird flocking flow, in two space dimensions (x, y), where (vx) represents the agent velocity in the direction of x-axis, (vy) represents the agent velocity in the direction of y-axis, (x, y) represents the agent's current position and (v_x, v_y) represents the current velocities in two dimensions. From the velocity and position information, the agent can be modified for the new position.

The school of fish and birds flock optimizes a given objective function based on the experiences and every time solution. The particle recognizes this information and an analogy is stored each time in under the name of the local best solution (***pbest***). At the same time in every cycle all particles recognize the best solution for all groups stored each time under the name of the global best solution (***gbest***). Depending on this information, each particle recognizes its performance, and performs in tandem with all other particles in the group. Therefore, each particle tries to adjust its position as shown in Figure 2.2, using the following information:

- Current positions (x, y),
- Current velocities (vx, vy),
- Distance between the current position (**x, y**) and **pbest**
- Distance between the current position (**x, y**) and **gbest**

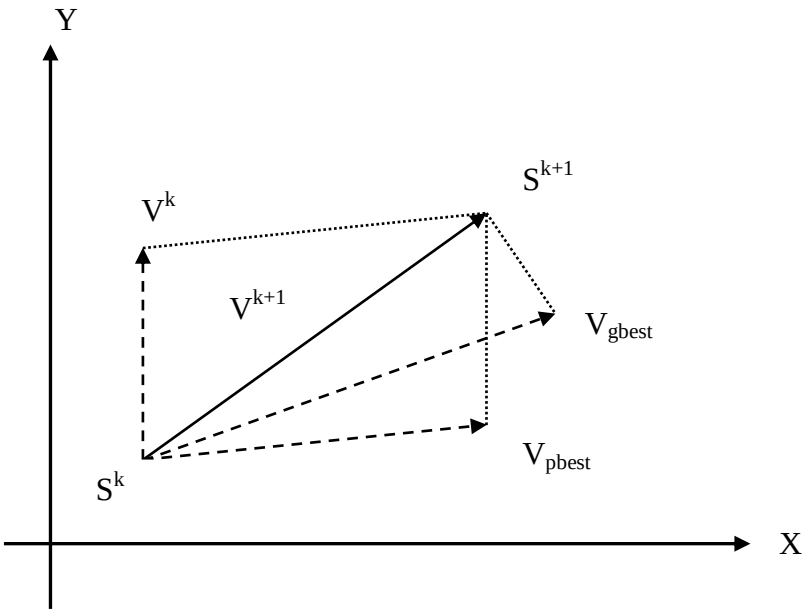

Figure 2.2: Concept of modification of a searching point by PSO.

where

S^k: current searching point.

S^k+1: modified searching point.

V^k: current velocity.

V^{k+1}: modified velocity.

V*pbest*: velocity based on *pbest*.

V*gbest*: velocity based on *gbest*

From the concept of velocity the new position is represented and modified (i.e. the modified value for the current positions). The following equation 2.2 by (Eberhart & Shi, 1998) expresses the modified velocity of each particle:

$$V_i^{k+1} = WV_i^k + c_1 rand_1 \times \left(pbest_i - S_i^k\right) + c_2 rand_2 \times \left(gbest_i - S_i^k\right) \qquad (2.2)$$

where

V_i^{k+1}: velocity of particle *i* at iteration k+1.

V_i^k: velocity of particle *i* at iteration k.

W: inertia function.

C1 and C2: are the acceleration constants.

Rand1 and Rand2: random number between 0 and 1.

S_i^k: current position of particle *i* at iteration *k*.

pbest$_i$: best position of particle ***i***.

gbest: the global best position of the group.

While the inertia weighting function is usually utilized as follows:

$$W = W_{max} - \frac{W_{max} - W_{min}}{iter_{max}} \times iter_i \tag{2.3}$$

where

W_{max}: initial weight,

W_{min}: final weight,

$iter_{max}$: maximum iteration number.

$iter_i$: current iteration number.

Equation 2.2 can be explained as follows. The RHS of consists of three terms, the first of which is the previous velocity of the particle. The second and third terms are utilized to change the velocity of the particle. Without the second and third terms, the agent will keep on "flying" in the same direction until it hits the boundary (i.e. it tries to explore new areas). Therefore, the first term corresponds to the diversification in the search procedure. On the other hand, without the first term, the velocity of the "flying" particle is only determined by using its current position and its best positions *pbest* in history. The particles will try to converge in the *pbests* and/or *gbest*, therefore the terms are corresponding to intensification in the search procedure (Eberhart & Shi, 2001b).

Figure 2.2 illustrates a concept of modification with particles in the solution space. Each particle finds the new position using the integration of velocity vectors. The current position can be modified by the following equation:

$$S_i^{k+1} = S_i^k + V_i^{k+1} \tag{2.4}$$

where

S^{k+1}: modified searching point.

S^k: current searching point.

V^{k+1}: modified velocity.

2.3.2 PSO Algorithm

PSO algorithm comprises a very simple concept, and paradigms are implemented in a few lines of computer code. The general flow chart of PSO is shown in Figure 2.3 and described below:

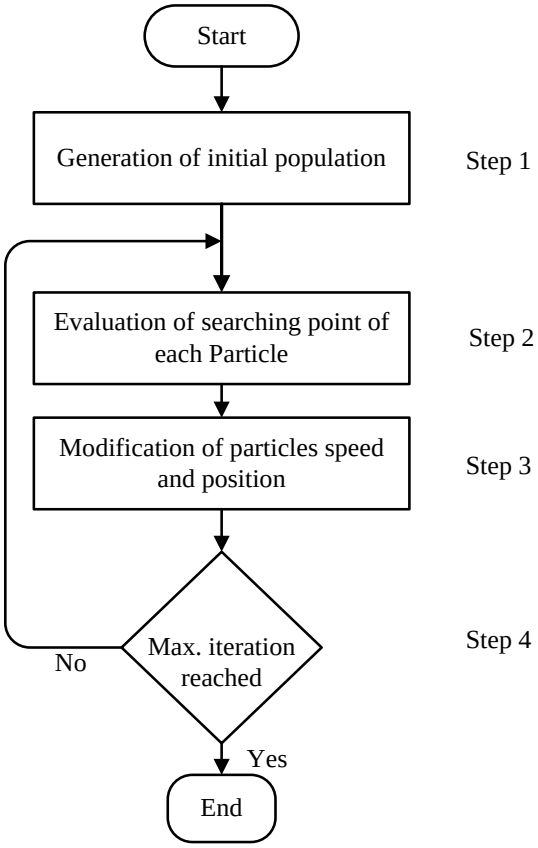

Figure 2.3: General PSO algorithm flowchart.

Step 1: Generate initial condition for each agent. Initial position searching points of particle $\boldsymbol{i}$ at iteration $\boldsymbol{k = 0}$, ($\mathbf{S}_i^{\mathbf{k}}$) and velocities ($\mathbf{V}_i^{\mathbf{k}}$) of each agent are usually generated randomly within the allowable range. The current searching point is set to *pbest* for each agent. The best-evaluated value of *pbest* is set to *gbest* and the agent number with the best value is stored.

Step 2: Evaluation of searching point of each particle. The objective function value is calculated for each particle. If the obtained value is better than the current local best value *pbest* of the particle, the new *pbest* value is replaced by the current value. If the local best value of *pbest* is better than the current global best value *gbest*, then

the new *gbest* is replaced by the best value and the particle number with the best value is stored.

Step 3: Modification of each searching point S^{k+1}. The current searching point S^{k} of each particle updated using previous equations 2.2 and 2.4.

Step 4: Checking the exit condition. The current iteration number reaches the predetermined maximum iteration number, then exit. Otherwise, go to step 2.

The detailed features of the searching procedure of PSO are:

(I). As shown in equations 2.2 and 2.4, PSO can essentially handle continuous Optimisation problem.

(II). Similar to GA, the PSO utilizes several searching points and the searching points gradually get close to the optimal point using their local best pbests and the global best gbest.

(III). The first term of right-hand side at equation 2.2 corresponds to diversification in the search procedure. The second and third terms at RHS of the equation correspond to intensification in the search procedure. This method has a well-balanced mechanism to utilize diversification and intensification in the search procedure efficiently.

(IV). The above concept of PSO can use more than two dimensions in the space. However, the method can be easily applied to n-dimension problem. In other words, PSO can handle continuous Optimisation problems with continuous state variables in an n-dimension solution space.

2.4 PSO Derivatives

As mentioned earlier in this chapter, PSO is one of the most important groupings under swarm intelligence, but it in turn is divided into several types, the most important of which are explained below.

2.4.1 Linear Version of Particle Swarm Optimisation (LPSO)

In the linear version of PSO, the particles are not influenced by random strangers; they are only affected by their topological neighbours. The topological geometry of

the particle swarm remains constant throughout the run. It is sometimes called local PSO (LPSO). LPSO is much like classical PSO, except that rather than use a different random number for each element of the velocity and position vectors, a single scalar is multiplied by each vector, thus:

$$V_{t+1} = wv_t + \emptyset_1 U_{1t}(p_i - x_t) + \emptyset_2 U_{2t}(g_i - x_t) \quad (2.5)$$

where w is the inertia weight, each $\emptyset_{1,2} \approx 2$, and $U_{1,2}$ is a vector of numbers drawn from a standard uniform distribution.

This means that the resultant velocity (and therefore position) is a strictly linear combination of other particle positions. If the particles are all initialized within $f = \{x|Ax = b\}$, then they will always be within f (Mendes, Kennedy, & Neves, 2004).

2.4.2 Global Version of Particle Swarm Optimisation (GPSO)

GPSO first introduced a new parameter called *inertia weight* into the original particle swam optimiser in order to increase the performance of the PSO. Initial GPSO found that inertia weight in the range (0.9, 1.2) on average will have a better performance; that is, it has a bigger chance to find the global optimum within a reasonable number of iterations (Yuhui Shi & Eberhart, 1998). Furthermore, a time decreasing inertia weight is introduced which brings in a significant improvement on the PSO performance as in (Zhan & Zhang, 2008); this research approved that the best value of inertia weighting during the Optimisation should be change from minimum value 0.4 to maximum value 0.9 (Shi & Eberhart, 1998), (Eberhart & Shi, 2001c) linked to this equation:

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{G}\right) \times g \quad (2.6)$$

where

W_{max}: maximum value of inertia weight,

W_{min}: minimum value of inertia weight,

G: maximum iteration number,

g: current iteration number.

In GPSO, each particle's velocity is adjusted according to its personal best and the performance achieved so far within learning from the personal best and the best position achieved so far by the whole population; instead of the local version each particle's velocity is adjusted according to its personal best and the performance achieved so far within its neighbourhood.

2.4.3 Comprehensive Learning Particle Swarm Optimisation (CLPSO)

CLPSO utilizes a novel learning strategy pattern whereby all other particles' historical best information, is used to update a particle's velocity. This strategy enables the diversity of the swarm to be preserved to discourage premature convergence.

In this strategy, the following equation is used to update the velocity:

$$V_i^d \leftarrow w * V_i^d + c^* rand_i^d * \left(pbest_{fi(d)}^d - X_i^d\right) \quad (2.7)$$

where *d* the dimension and *i* the particle

c^**:* are the acceleration constants,

V_i^d *:* the velocity,

X_i^d*:* the position,

W: inertia function.

$f_i = [\, f_i(1),\ f_i(2),\ \ldots . f_i(D)]$: defines of pbest's which particle i should be followed.

$pbest_{fi(d)}^d$: can be the corresponding dimension of any particles.

2.4.4 Dynamic Multi-Swarm PSO with Local Search (DMS-PSO)

The dynamic multi-swarm particle swarm optimiser, where is constructed based on the basic version of PSO and a new neighbourhood topology, is used in this case (J. Liang & Suganthan, 2005):

$$Min_x\, f(x) = [x_1, x_2, \ldots \ldots, x_D] \quad (2.8)$$

where $x \in [x_{min}, x_{max}]$ and D the dimension number to be optimised.

This new neighbourhood structure has two important characteristics:

A. Small Sized Swarms

PSO needs a comparatively smaller population size, particularly to solve the simple problems, as a population with four to six particles can achieve best results. Conversely, the other evolutionary algorithms prefer larger populations, while smaller neighbourhoods yield good results and perform better for complex problems. Hence, in the new version, small neighbourhoods are used. In order to slow down the population's convergence velocity and increase diversity, the DMS-PSO, divides the population into small sized swarms, each of which uses its own members to search for better areas in the search space.

B. Randomly Regrouping Schedule

Swarms of small sizes are looking for the best use of their own historical data, which is easy to converge to a local optimum, as are the property of convergence of the PSOs. In this case, it can keep the same neighbourhood structures, there will be no exchange of information between the swarms, and PSO will co-evolve these swarms in parallel investigation. In order to prevent this, a program is imported into the random rearrangement. Each R generations the population is grouped randomly and searching beings with a new configuration of small swarms. This period is called the group R. Thus, good information will be obtained to be exchanged between each swarm, although the diversity of the population increases. The new neighbourhood structure has more freedom compared to the conventional neighbourhood structure, resulting in better performance for complex multimodal problems.

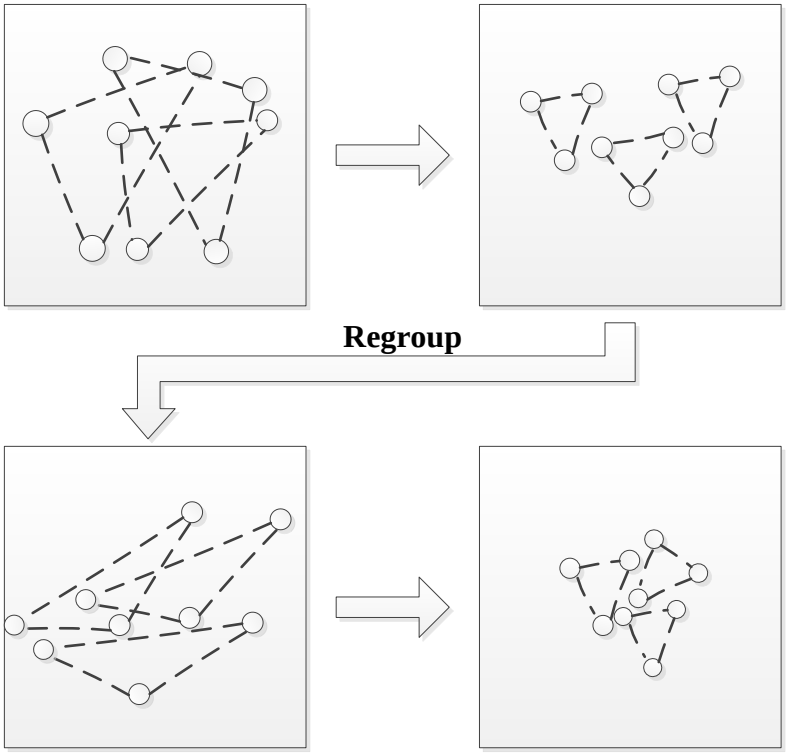

Figure 2.4: DMS-PSOs search (J. Liang & Suganthan, 2005).

Figure 2.4 shows how to regrouping schedule whereby the swarm is divided into three new random swarms with three particles in each one. Subsequently, the three flocks of particles are searching for better solutions individually. During this period, they may converge to the closed a local optimum. Subsequent regrouping leads to redistribution of the population in new swarms, which start searching. This process continues until a stopping criterion is met. The program is scheduling all particles swarms to regroup in new configurations so that each small swarm's search space is enlarged, increasing the chance of finding better solutions using new small swarms (S. Zhao, Liang, Suganthan, & Tasgetiren, 2008) . At the end of the search, in order to perform a better local search, all the simple particles from the each swarm become a GPSO version. The pseudo code of DMS-PSO is given in Figure 2.5.

m: Each swarm's population size
n: Swarms' number
R: Regrouping period
Max_gen: Max generations, stop criterion
Initialize m×n particles (position and velocity)
Divide the population into n swarms randomly,
with m particles in each swarm.
For i=1:0.9×Max_gen
Update each swarm using local version PSO
If mod(i, R)==0,
Regroup the swarms randomly,
End

Figure 2.5: DMS-PSO sequence.

2.4.5 DMS-PSO with Sub-regional Harmony Search (DMS –PSO- SHS)

The dynamic multi-swarm particle swarm optimiser (DMS-PSO) with sub-regional harmony search (SHS) cross-breeds to obtain DMS-PSO-SHS. A modified algorithm called multi-trajectory search (MTS) is widely applied in various selected solutions. Effectively, variety maintaining population diversity puts more dynamic properties of swarms in DMS-PSO. Without crossing operation in high operational characteristics, HS intersection operation converts multiple parents of overall search behaviour for the proposed DMS-PSO-SHS. The entire population of PSO is divided into several sub-swarms, as population individuals into a large number of sub-swarms are often grouped into individual HS population. These sub-swarms are regrouped regularly by various programs and information is exchanged between the particles in the whole swarm. Therefore, diverse existing multi-swarm PSOs or local-version PSOs emerge, and the sub-swarms are small but dynamic, which is useful for a population and is appropriate for harmony research. Moreover, the last selected solutions from external memory are used to improve the diversity swarm.

The equation of the original HS algorithm are specified as the harmony memory and stored on feasible vectors (Lee & Geem, 2005), (Omran & Mahdavi, 2008) as shown in equation 2.9.

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_D^1 \\ x_1^2 & x_2^2 & \dots & x_D^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_D^{HMS} \end{bmatrix} \begin{bmatrix} fitness(x^1) \\ fitness(x^2) \\ \vdots \\ fitness(x^{HMS}) \end{bmatrix} \quad (2.9)$$

where *HMS* is harmony memory size, and the candidate set [x_1, x_2, …, x_{HMS}] are the rules number with *D* is the number of parameters (Geem, 2009).

As explained previously, the DMS-PSO-SHS is the hybridization of DMS-PSO and the regional harmony search (HS), which is based on the current *pbest*s in each sub-swarm after PSO positions are updated. Nearest *pbest* is replaced by better fitness from with new harmony. MTS modified algorithm implements new line search along the dimension one by one. In addition, a method for improving diversity is used to improve the diversity of the swarm with a relatively low frequency and for timely discourage convergence in the right steps during the early search stage. The DMS-PSO-SHS modification with MTS attempts to take advantage of the PSO, HS and MTS to sidestep all particles are found in the lower regions local optimum. DMS PSO-SHS makes the particles several examples teach after swarms are often grouped and have greater harmony research among different sub-populations potential space. The DMS-PSO-SHS rejects the parameters of original HS, which usually need to be adjusted based on the property of the test problems, such as the bandwidth.

2.4.6 Adaptive Particle Swarm Optimisation (APSO)

Adaptive particle swarm optimiser (APSO) provides a better search performance and more efficiency than basic PSO. More importantly, it enables the performance of a global search space at a speed of convergence. The APSO consists of two phases: first, by examining population distribution and particle fitness, a method for the estimation of the evolution status in real-time is carried out on the four stages of identity evolution defined below, including the exploration, exploitation, convergence and jumping out in each generation. Also, its enables the automatic control of inertia weight, acceleration coefficients and other performance parameters to improve search algorithms, efficiency and speed of convergence. Then, it makes a statement of changes elite learning strategy when the state is classified as convergence. The strategy will focus on globally best particle to jump out of the likely local optima (Zhan, Zhang, Li, & Chung, 2009).

A. Population Distribution Information

The most prominent step that has been focused by APSO is based on complex analysis procedures on the population itself. In this section, PSO process and studied the characteristics of the population distribution for the first time to make an evolutionary state estimation approach. The distribution of information can be formulated as shown in Figure 2.6.

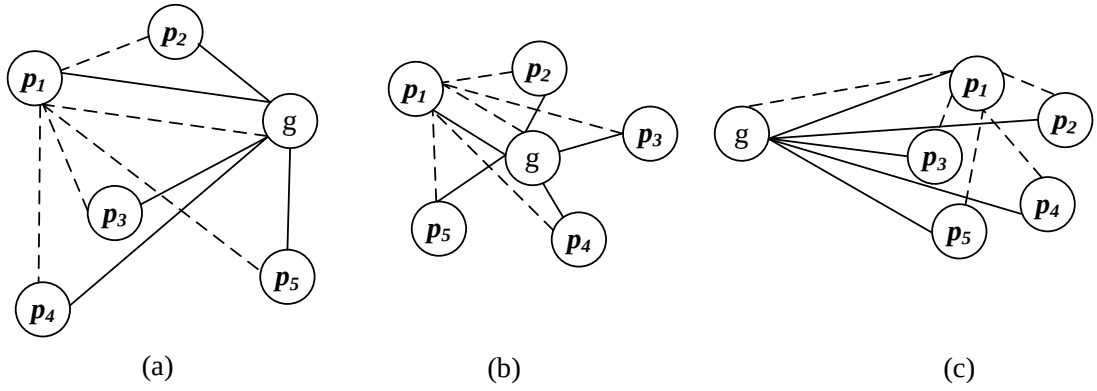

Figure 2.6: APSO population distribution information quantified by evolutionary factor *f*. (Zhan et al., 2009).

By calculation of the average particle distance of each of the other particles, it is reasonable to expect that the average distance between the global best particles with other particles in the state of convergence would be minimal as shown in the Figure 2.6 (a) the distance $d_g \approx d_{pi}$ during exploration, Figure 2.6 (b) the distance $d_g \ll d_{pi}$ during exploiting and Figure 2.6 (c) the distance $d_g \gg d_{pi}$ during the jumping out whereby the *gbest* is inclined to be surrounded by the swarm. In contrast, the average distance would be the maximum distance when jumping out of the state, because the global best is likely to be crowded away from the swarm. Therefore, evolutionary state estimation focuses on the dissemination of information to the population in each generation, as in the following steps:

Step 1: calculate the average distance of each particle *i* for all the other particles. This can be measured using the Euclidian metric equation:

$$d_i = \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} \sqrt{\sum_{k=1}^{D} \left(x_i^k - x_j^k\right)^2} \tag{2.10}$$

where *N* is the population size and *D* is the number of dimensions.

Step 2: indicate the distance *di* of the globally best particle as *dg*, and compare all other *di's*, for the particles to determine the maximum distance *dmax* and minimum distances *dmin* to calculate the evolutionary factor *f* as defined by this equations.

$$f = \frac{d_g - d_{min}}{d_{max} - d_{min}} \tag{2.11}$$

Step 3: classify the f value into four sets (S1, S2, S3 and S4) representing the states of exploration, exploitation, convergence and jumping out Figure 2.7.

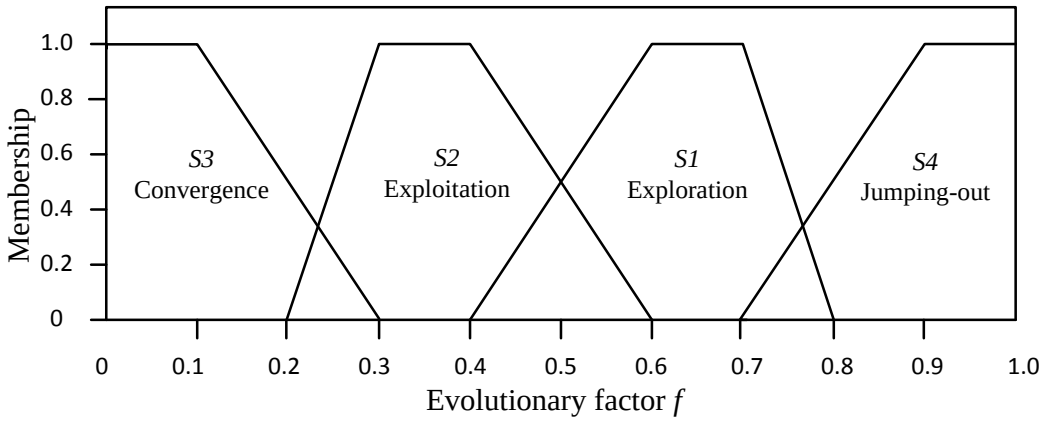

Figure 2.7: Fuzzy membership functions for the four evolutionary states (J. Liang & Suganthan, 2005)

B. Adaptive Control of PSO Parameters

In this phase is to be controlled in PSO parameters (Eberhart & Shi, 2001b) (Ratnaweera, Halgamuge, & Watson, 2004), including the adaptation of the inertia weight, control of the acceleration coefficients and bounds of the acceleration coefficients during the search process, whereby the adaptation of the inertia weight is controlled by the following formula:

$$w(f) = \frac{1}{1 + 1.5e^{-2.6f}} \in [0.4,0.9] \tag{2.12}$$

where the acceleration coefficients and bounds of the acceleration coefficients are controlled by the following equations (Zhan, Xiao, Zhang, & Chen, 2007)

$$|c_i(g+1) - c_i(g)| \leq \delta \tag{2.13}$$

where δ is the acceleration rate and the value is dependent on the experiments revealing that a uniformly generated random value in some research takes the interval (0.05, 0.1).

$$c_i = \frac{c_i}{c_1 + c_2} \times 4.0 \quad (2.14)$$

where C_1 and C_2 are the parameter speed coefficients where take the value [1.5 to 2.5] as interval chosen (Carlisle & Dozier, 2001):

2.4.7 Unified Particle Swarm Optimisation (UPSO)

UPSO associates both the LPSO and GPSO variants and its structure has been proposed as an alternative that combines the properties of exploration and exploitation associated with local and global PSO (K. Parsopoulos & Vrahatis, 2004). Let *Li* (t +1) and *Gi* (t +1) denote update the velocity of the particle x_i in local and global PSO variation respectively (K. E. Parsopoulos & Vrahatis, 2005):

$$G_i(t+1) = x\left[V_i(t) + c_1 rand_1\left(P_i(t) - X_i(t)\right) + c_2 rand_2\left(P_b(t) - X_i(t)\right)\right] \quad (2.15)$$

$$L_i(t+1) = x\left[V_i(t) + c_1 rand_1\left(P_i(t) - X_i(t)\right) + c_2 rand_2\left(P_{b_i}(t) - X_i(t)\right)\right] \quad (2.16)$$

where

t: denotes the iteration number,

b: is the index of the best particle of the whole swarm,

bi: is the index of the best particle in the neighbourhood of Xi (local variant).

The main UPSO scheme is defined by:

$$U_i(t+1) = (1-u)L_i(t+1) + uG_i(t+1) \quad (2.17)$$

$$X_i(t+1) = X_i(t) + U_i(t+1) + uG_i(t+1) \quad (2.18)$$

The parameter u ∈ (0, 1) balances the influence of the local and global search direction in the unified layout, which called the unification factor.

In addition to the previous paradigm, a stochastic parameter to mimic the mutation can also be included in this evolutionary algorithm in equation 2.17 in order to improve UPSO exploration ability (K. E. Parsopoulos & Vrahatis, 2005). Thus, the variant of USPS is based primarily on equation 2.17 and can be written as equation 2.18:

$$U_i(t+1) = (1-u)L_i(t+1) + r_3 u G_i(t+1) \tag{2.19}$$

where $r_3 \sim N(M, \sum)$ is a normally distributed parameter with mean vector M and variance matrix $\sum$ (Matyas, 1965), or

$$U_i(t+1) = r_3(1-u)L_i(t+1) + u G_i(t+1) \tag{2.20}$$

2.5 PSO Application in Power Systems

The applications of particle swarm Optimisation in electric power systems and other optimisers such as GA and ant colony Optimisation has been successfully applied to solve electrical power system problems through Optimisation problems such as optimal power flow, economic dispatch, reactive power dispatch, unit commitment, generation and transmission planning, maintenance scheduling, state estimation, model identification, load forecasting, control, power system reliability and security, generation expansion problem, neural network training and others.

2.5.1 Optimal Power and Load Flow

The optimal power and load flow (OPLF) problem can be solved by implementing PSO as introduced by Abido (Abido, 2002). The aim of using optimal power flow (OPF) is to explore the best state or optimal combination of independent variables power generation cost function by finding the lowest cost. The generator actual power productions are treated like predictor variables with the other predictors measured earlier in the reactive power Optimisation problem. Sophisticated Optimisation problems can be handled with PSO effectively, which has several equality and inequality limitations and both continuous and discrete variables. Zhao *et al.* answered the very complicated OPF Optimisation issue by minimizing a non-stationary multiagent assignment penalty function in another methodology for Optimisation (B. Zhao, Guo, & Cao, 2004). It was found that the PSO solved the

Optimisation of extremely constrained problems in OPF. All penalty values change dynamically according to the system constraints.

In (He *et al.*, 2004) , the concept of passive congregation was incorporated into PSO to solve the OPF problem. This improved convergence characteristics of the traditional hybrid PSO technique to solve the same OPF problem. The load flow study is the most fundamental tool for the analysis and design of electrical power system installations and it is used for the operation of the electricity system scheduling, economic planning, transient stability, voltage stability and emergency studies. The load flow equations contain energy balance equations for both the active and reactive power for each bus. The problem of the flow of feed can be formulated as an Optimisation problem where the goal is to minimize and make the voltage amplitudes and angles difference between the input power and output on any bus (El-Dib, Youssef, El-Metwally, & Osman, 2006) . The cost function can be expressed as follows:

$$f(V,\delta) = \sum_{i} (F_{pi}^2 + F_{qi}^2) + \omega P_l \tag{2.21}$$

and the particle *x* is defined as:

$$x = [V_1, \dots, V_n, \delta_1, \dots, \delta_n] \tag{2.22}$$

where (V_i, δ_i) depends on the magnitude and phase of the bus i voltage, F_{pi} and F_{qi} are determined according to the nonlinear power flow equation, ω is weighting factors and P_l is the real loss.

2.5.2 Economic Dispatch

PSO is utilised for Dispatch Economic (DE), which is realized as the generation unit in the combined process of allocating generation levels, to make the system load fully powered and economical (Victoire & Jeyakumar, 2004). Mathematically, the problem of economic dispatch is to minimize find limitations, respecting the constraints of equality and inequality, as a result of selecting the optimal combination. Economic transmission can be thought of as a static Optimisation problem in which the costs associated with the operation of the change of the generators outputs are not taken into account, and a Dynamic Economic Dispatch

(DED) Optimisation problem considers the costs associated with the change. The ED or DED issues can be expressed with or without smooth cost function. Where ED in a smooth function with costs, operating costs of each generator are represented by a quadratic function as follows:

$$Cost = C = \sum_{i \in I} F_i(P_i) \tag{2.23}$$

where

C is the total cost.

F_i is the generation cost function.

P_i is the generation electrical output.

I is the all generation group.

2.5.3 Reactive Power and Voltage Control

Overall, a grid is activated responsively to sudden changes in the composition, such as lines and suddenly changes in loads. In this case the dynamic system tries to keep the voltage within an allowable range for consumers, which is one of the main tasks of the generation electrical power units. To achieve this objective, power plant operators use advanced control of the synchronous generators, transformers tap setting, and flexible alternating current transmission system devices, as well as adapting the power plants to keep the required amount of reactive power to maintain the bus voltages at the required level. This strategy is used online in order to achieve the suggested reactive power and voltage control power or volt/VAR control (VVC) in short-lived. Essentially, all VVC strategy must ensure that voltage safety limit is to be respected, so that system conditions do not escalate to voltage collapse. There are many conventional techniques for the analysis of emergency assessment (Cutsem & Vournas, 2008). The voltage and VAR control can be optimised as in the following formula (Coath, Al-Dabbagh, & Halgamuge, 2004):

$$Minimise: \sum_{i=1}^{n} Q_{loss_i} \tag{2.24}$$

where *n* is the number of network branch, and Q_{loss_i} is the reactive power at the branch *i*.

2.5.4 Unit Commitment

For the problem of scheduling unit, Miranda and Win (Miranda & Win-Oo, 2006) described the use of the difference in evolutionary particle swarm Optimisation on the generator unit commitment problem of the power generator system programming. In this case, the study was applied to a number of generators with production curves, to determine which generator's producers should work and where, and the level of production, in order to minimize costs, including start-up and operating costs for a particular set of system conditions. The algorithm with improved version called evolutionary particle swarm Optimisation PSO performs better in the movement rule by the memory element. Also, a better result was achieved by PSO-based algorithm compared with other types of GA (Miranda & Win-Oo, 2006; Victoire & Jeyakumar, 2005). The cost function was expressed as:

$$P_{Cmax} = \sum_{i=1}^{N} P_{imax} \;\; ; \;\; P_{Cmin} = \sum_{i=1}^{N} P_{imin} \tag{2.25}$$

where N is the number of units and $i \neq NRU$ (not run unit), $P_{cmax} \geq D_{max}$, and $P_{cmin} \leq D_{min}$ where D_{max} and D_{min} are the maximum and minimum power demands in the whole scheduling prospect.

2.5.5 Generation and Transmission Planning

Electrical production planning determines what, when, where and how to install generating units to supplying electricity to the electrical grid. Mathematically, the planning of generation can be formulated non-linearly with a large-scale Optimisation mixed stochastic problem to minimise the risk and maximise profits, subject to a range of complex constraints of load demand without sacrificing reliability (Wu, Yen, Hou, & Ni, 2004) . In other methods based on PSO as an AI and metaheuristic technique it has been applied to solve problems of electrical production (Kannan, Slochanal, & Padhy, 2005; Wu et al., 2004) . Other research applied the particle swarm Optimisation to the problem of electrical production planning in a competitive milieu concluded that the PSO technique is a genuine and efficient means for generating competitive scheduling problems, especially when the

number of companies that generate or generate strategies increases (Slochanal, Kannan, & Rengaraj, 2004). The cost function is expressed as:

$$Max_{Profet} = q_i[P - C_i] \tag{2.26}$$

$$P = constant - a\left(\sum_{i=1}^{N} q_i\right) \tag{2.27}$$

where a is demand coefficient, q_i is the power in (MW) produced by i-th generation expansion companies (GENCOs), N is the total number of GENCOs, C_i is the cost of i-th GENCOs.

2.5.6 Maintenance Scheduling

Maintenance program is a prevention outage program for electrical production units in power systems during a given period of time, which is a complex problem when the power system includes a number of units with different generation specifications. While it has a number of limitations, these must be considered against its practical benefits and function as a possible solution (Koay & Srinivasan, 2003). It also presents a spawning and selection mechanism with PSO algorithm for solving problems by scheduling the maintenance of multi-generator with many limitations, concluding with the optimisation based on particle swarm obtaining efficient schedules in a reasonable time. The following function is necessary to minimize all parameters if crew and resource constants or the schedule cannot meet the power demand the penalty cost is added to evaluation function (Srinivasan & Malik, 2002).

$$F = \sum_{t=1}^{T}\sum_{x=1}^{X} 168(a_x P_x^2 + b_x P_x + C_x) + \sum_{y=1}^{Y} V_y D + \sum_{t=1}^{T} Penalty\ Cost \tag{2.28}$$

where

X = unit in operation in that week,

Y = unit in maintenance,

T= length of the maintenance planning schedule (week),

P, = generator output (MW) of operation unit,

a_x, b_x, c_x. = fuel cost coefficient,

V_y = maintenance cost per week (£/week),

D = downtime (weeks).

2.5.7 State Estimation

State estimation is generally formulated as a weighted least mean squares problem, and must take into account the asynchrony measurement and error (Naka, Genji, Yura, & Fukuyama, 2003). A hybrid PSO can also be proposed for the distribution of the state estimation taking into account the non-linear characteristics of practical machinery and limited of actual distribution measurements, concluding that estimation method is proposed for practical distribution systems using PSO of the application state (Naka, Genji, Yura, Fukuyama, & Hayashi, 2000). Online estimation of the state is a crucial factor in the electrical distribution control centres, particularly with the introduction of distributed generators (DGs) for the electricity power system. Energy companies must have accurate estimation of the system load and outputs of DGs, while usually limited measures are obtainable by the network. Estimation of the state can be formulated as a weighted average least squares problem (Naka *et al.*, 2003):

$$Minimise \quad J(x) = \sum_{i=1}^{m} w_i (z_i - h_i(x))^2 \tag{2.29}$$

where m is the number of measurement variable, x state variable (active-power loads and active-power output of DGs), w_i weighting factor of measurement variable, z_i measurement value of measurement variable (voltages and currents) i, h_i is the power flow equation of measurement variable.

2.5.8 Model Identification

PSO has been used in order to highlight the disadvantages of common model identification methods for thermal processes, and to present novel identification solutions based on PSO (Liu & He, 2005). Identification solutions based on PSO can overcome the drawbacks of the common model identification method of heat treatment process, and PSO approaches provide the characteristics of ease of realisation and identifying with high accuracy compared to identification by the improvement of the GA. The suggested of optimal fitness function is:

$$f_{itness} = \sum_{t=1}^{M} 100[y(t) - \tilde{y}(t)]^2 \quad (2.30)$$

where $\tilde{y}$ is the output actual process, *y* is the output of identification result, and *M* is the number of sample data.

2.5.9 Load Forecasting

PSO has recently been applied in load forecasting to identify average autoregressive exogenous with exogenous variables of load model. The research (Huang, Huang, & Wang, 2005) presents an alternative technique for determining and estimating model parameters for a global minimum prediction error in an efficient computation time. Accuracy of fit is also improved. The proposed technique is tested on four different power load datasets for practice one day and one week in advance. Load forecasting based on PSO has better prediction capability, with a shorter execution time than evolutionary programing (EP) and conventional stochastic time series (STS) convergence (Huang et al., 2005) (B. Wang et al., 2008). Auto-regressive and moving average with exogenous variables (ARMAX) model can be expressed as:

$$A(q)L(t) = B(q)u(t) + C(q)e(t) \quad (2.31)$$

where *L(t)* is the load demand at time *t*, *u(t)* is the exogenous temperature, *e(t)* is the noise, *A(q)* is the equation of the autoregressive, *B(q)* is the equation of the exogenous input and part *C(q)* is equation of the moving average part.

2.5.10 Neural Network Training

Neural networks are a valuable tool for AI in many areas of electrical systems. PSO is used to train a neural network for transformer safety protection (El-Gallas, El-Hawary, Sallam, & Kalas, 2001), with the goal of developing an intelligent model that is able to distinguish between the internal fault current and magnetisation current flow in transformers. PSO is used to improve the accuracy and the performance of the identification processing time. PSO identifies the optimal weight of a neural network model and controls the stability of the power supply system (Hirata, Ishigame, & Nishigaito, 2002). Where the problem is to find the connecting weight matric *W, V* of neural network (NN), the maximum value function is given by:

$$J(W,V) = max\, F(x,W,V) \tag{2.32}$$

To be minimized as $J(w,v) = 0$, which is non-deferral function, as it only can be solved by PSO, and $F(x)$ is the training function problem.

To determine improvement, the Optimisation problem was formulated as a min-max problem with the objective that the function be of the discontinuous nature and not a differential problem. PSO is integrated with a neural network to identify the safe limits dynamic energy in a surrounding area of power system (Kassabalidis, El-Sharkawi, Marks, Moulin, & Alves da Silva, Alexander P, 2002). The minimized objective function is expressed as:

$$f_1(\overrightarrow{x}) = \left|NN(\overrightarrow{x}) - C\right| \tag{2.33}$$

where $(\overrightarrow{x})$ is the operating point of the power system as the security index of the border were set by the user, *NN* is the security index produced by the neural network.

2.5.11 Other Applications

The PSO was used to solve nonlinear Optimisation problems on many other applications at electrical power system, including the following.

Electric machinery; PSO application in the field of electrical machines has not been extensively studied. However, Emara et al. (Emara, Ammar, Bahgat, & Dorrah, 2003) applied PSO to estimate stator induction motor failure.

1. Power system reliability and security; reliability distribution is defined as adequate supply, which is linked to the presence of adequate system to the load requirements within the limits of the safety system and constraints, which is the capability of the system to overcome disturbances occurring and interference occurring on it (Robinson, 2005).
2. Generation expansion problem; generation expansion problem (GEP) is becoming increasingly important in economic decisions. The application of PSO to solve GEP was reported in (Slochanal et al., 2004), (Sensarma, Rahmani, & Carvalho, 2002), and the application of equation metaheuristic techniques for solving the GEP in (Sensarma et al., 2002), wherein PSO was compared to eight other metaheuristic techniques in terms of success rate and execution time.

2.6 Summary

Many areas in power systems require the solution of one or more non-linear Optimisation problems. Although the analysis may suffer from slow convergence and the curse of dimensionality, heuristics-based swarm intelligence can be a very effective alternative. PSO, which is part of the family of swarm intelligence, is known widely to be effective to solve nonlinear Optimisation problems. As explained in this chapter, the basic principles of PSO and its variants provide a detailed study and applications on the power system that have benefited from the powerful of PSO as an Optimisation technique. For each application, the technical specifications for the implementation of the PSO, such as different types, the formulation of particles (representation of the solution), and effective fitness functions are also discussed.

Chapter 3: Combination of Multi Particle Swarm Optimisation

3.1 Introduction

In this chapter, a new method is proposed using different Optimisation schemes which are combinations of different types of PSO and methods used simultaneously. The idea is to make use of the characteristics of each individual method to tackle the same problem. The proposed schemes can utilize and share information about the local and global best as well as the swarm population, in specific predefined iteration of all types simultaneously. The method uses the good properties of the global search and efficient local search capability. Moreover, the schemes themselves are very simple and easy to implement, as explained in detail below.

As mentioned in Chapter Two, this research concerns a collection of different algorithms running together. The main objective behind this approach is the large variations in the performance of meta-heuristics for the same problem of varying complexity or using different random seeds on the same problem instance (Gomes & Selman, 2001).

This chapter describes the design and implementation of new Optimisation methods called parallel and sequential particle swarm Optimisation (PPSO and SPSO) using different types of PSO, whereby the new algorithm is tested by a collection of empirical benchmark functions that represent uni-model and multi-model as extremely complex systems, which will be explained and recognised by their shapes into two dimensions, with detailed discussion of their complexity. The new algorithms are compared to the most well-known types of PSO algorithms.

The chapter is organized into seven sections. Following this introductory section, a brief summary is given of the combination of empirical benchmark functions, then an explanation of the design and development of parallel and sequential particle swarm Optimisation algorithms is presented. The different types of PSOs and their behaviour with regard to different benchmarks are then compared, followed by the conclusion of the chapter.

3.2 Combination

In this work, the seven types of PSOs (LPSO, GPSO, CLPSO, DMS–PSO, DMS–PSO-SHS, APSO and UPSO) are used to design the multi-particle swarm Optimisation algorithm. In this design, communication between different algorithms are considered at early stages, whereby the fast and less accurate algorithm can pass its results to the slow and more accurate algorithm, which will benefit from the good results at an early stage. An outline of the proposed algorithms used in this paradigm is listed as follows:

- Generate the first swarm randomly
- Measure the fitness of each individual
- Evaluate the swarm
- Ranking operation
- Selection operation
- Go to all optimisers to produce a new generation
- Evaluate and scale the swarm
- Measure the fitness again
- Send the most fit individuals to all optimisers

In this technique, each type of PSO optimiser takes the generation after encoding and modifying the swarm based on its principles. After that, the optimisers work individually, in parallel or in series, depending on the paradigm, to find a good solution during the iterations. The number of iterations is chosen experimentally after several trails. The iterations number that best improves the performance of algorithm is selected. When the optimisers finish the internal iterations, the best individuals are provided to the optimisers by the system while the worse individuals are removed. In the next generation, the optimisers will benefit from this advantage whereby good individuals have been harvested from other optimisers during their search for the optimal result. Figure 3.1 illustrates the general mechanism of the communication between all optimisers.

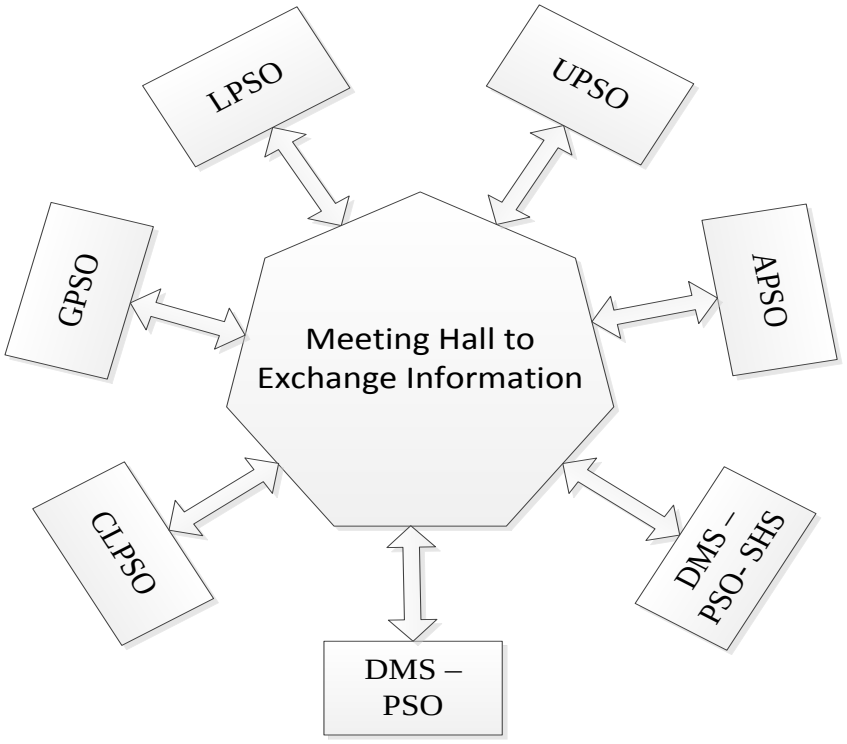

Figure 3.1: Communication between the optimisers.

In this research, after several simulations experiments, the number of iterations internally is adjusted to be five, which renders the best performance for the algorithm by experiment. Hence, after each these iterations, the best generation for each optimiser is sent by the system parallel or serial to the meeting hall, where the orders of individual particles are rearranged based on their fitness. After that the best individuals are chosen to send to the optimisers while the less fit individuals are removed. The number of individuals that are chosen must be equal to the size of the original swarm. Additional details about the two proposed paradigms are given later in this chapter.

3.3 Empirical Benchmark Functions

Further experimental tests with benchmark functions are carried out in this section to validate the evolutionary algorithms techniques and to compare them with others. There are a popular range of benchmark test functions which are commonly used to test the performance of Optimisation algorithms, which are chosen with regard to their particularities, hence the benchmark sets are defined here to include several conventional scenarios, from a simple function with a single minimum to others having a considerable number of local minima of very similar values (RW.ERROR -

Unable to find reference:116; Esquivel & Coello Coello, 2003; Pohlheim, 2007). The seven benchmark functions listed in Table 3.1, presenting four uni-modal functions and three multimodal benchmark functions. All functions are tested on 10 trails and 30 dimensions, according to their bounds.

Table 3.1: Benchmark functions, uni-modal and multi-modal.

Mathematical Equation		Function Name	Search Band	Global Minima	Accepted	D
Uni-modal						
f_1	$\sum_{i=1}^{D} A^{((i-1)/(D-1))} \times x_i^2$	Elliptic	$[-100\ 100]^D$	0	0.01	30
f_2	$\sum_{i=1}^{D} x_i^2$	Sphere	$[-100\ 100]^D$	0	0.01	30
f_3	$\sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	Schwefel	$[-10\ 10]^D$	0	0.01	30
f_4	$\sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	Rosenbrock	$[-10\ 10]^D$	1	100	30
Multimodal						
f_5	$\sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	Rastrigin	$[-5.12\ 5.12]^D$	0	50	30
f_6	$-20exp\left(-0.2\sqrt{1/D \sum_{i=1}^{D} x_i^2}\right) - exp\left(1/D \sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + e$	Ackley	$[-32\ 32]^D$	0	0.01	30
f_7	$1/4000 \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(x_i/\sqrt{i}) + 1$	Griewank	$[-600\ 600]^D$	0	0.01	30

3.3.1 Elliptic Function

An elliptic function is holomorphic function except at poles and has singularities in a finite part of the plane. A periodic function of a real variable is defined by its values on an interval, and satisfies equation 3.1, and the shape of this function for two-dimensional case is shown in Figure 3.2.

$$f_1(x) = \sum_{i=1}^{D} A^{((i-1)/(D-1))} \times x_i^2 \tag{3.1}$$

where *A* is constant and *D* the number of dimensions.

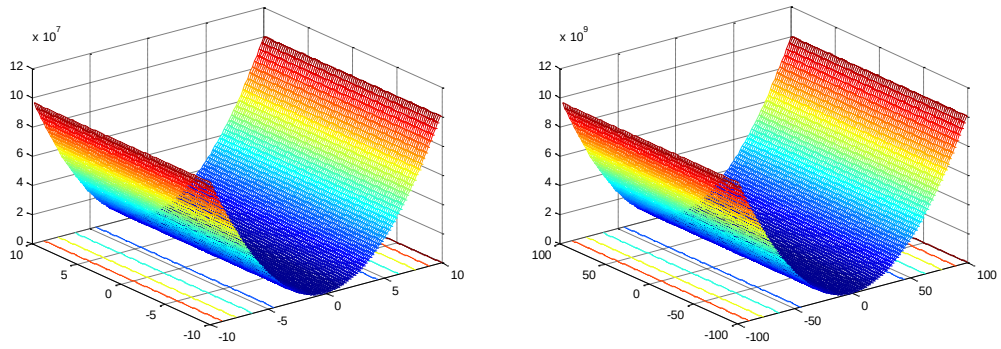

Figure 3.2: Elliptic function.

3.3.2 Sphere Function

The sphere function is commonly used as a test function for Optimisation algorithms, because any smooth function is locally quadratic near its optimum, and thus convergence on the sphere function is a necessary condition for convergence on any smooth function, also known as De Jong's function. The shape of this function for two-dimensional case is shown in Figure 3.3.

$$f_2(x) = \sum_{i=1}^{D} x_i^2 \tag{3.2}$$

where *D* the number of dimensions.

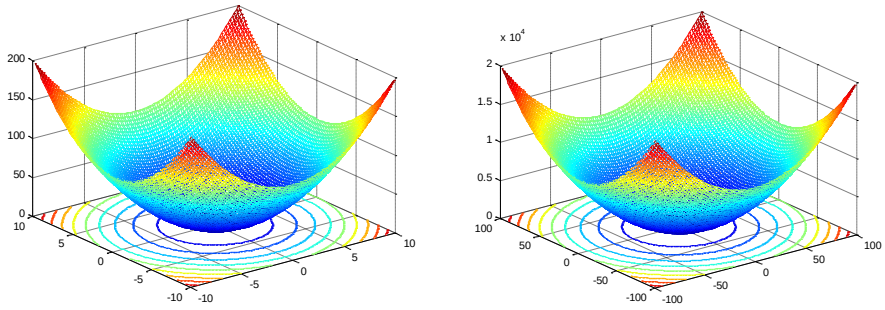

Figure 3.3: Sphere function.

3.3.3 Rastrigin's Function

Rastrigin's function is based on the Sphere function with the addition of cosine modulation in order to produce frequent local minima. Thus, the test function is highly multimodal. However, the location of the minima is regularly distributed. The

function is presented by equation 3.3, and the shape of this function for two-dimension case is shown in Figure 3.4.

$$f_5(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10] \tag{3.3}$$

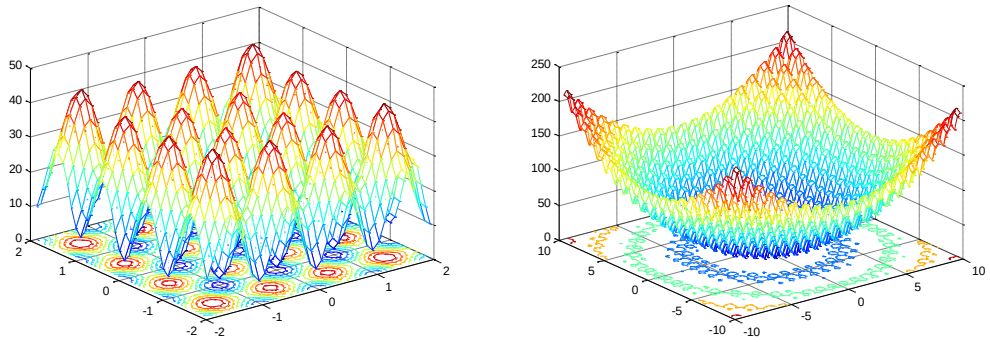

Figure 3.4: Rastrigin's function.

3.3.4 Schwefel's P2.22 Function

Schwefel's function is somewhat easier than Rastrigin's function, and is characterized by a second-best minimum, which is far from the global optimum. The shape of this function for two-dimensional case is shown in Figure 3.5.

$$f_3(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i| \tag{3.4}$$

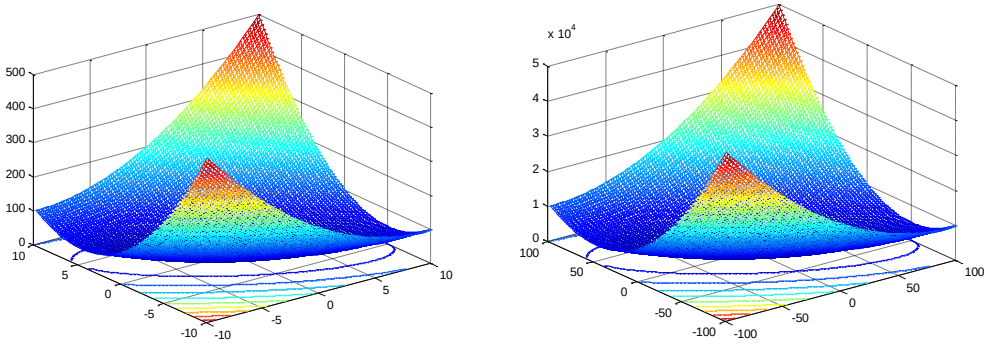

Figure 3.5: Schwefel's P2.22 function.

3.3.5 Rosenbrock's Function

Rosenbrock's function is a classic Optimisation problem, also known as Banana function. The global optimum is inside a long, narrow, parabolic shaped flat valley. To find the valley is simple, however convergence to the global optimum is difficult, hence this problem has been repeatedly used to assess the performance of Optimisation algorithms. The can be presented as in equation 3.5, and the shape of this function for two-dimensional case is shown in Figure 3.6.

$$f_4(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \quad (3.5)$$

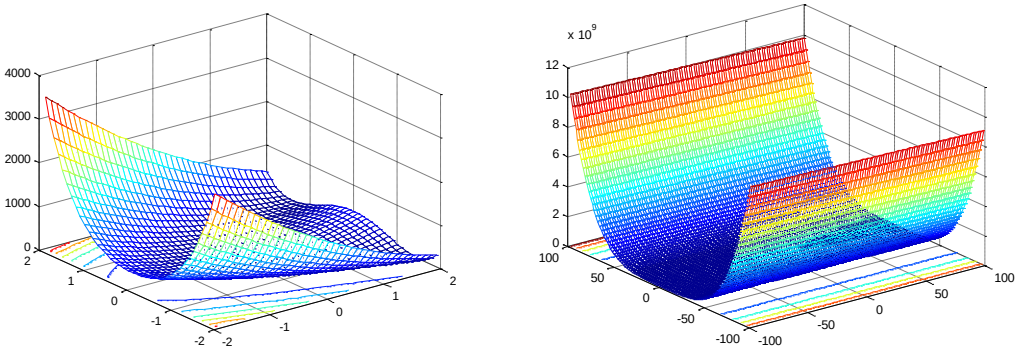

Figure 3.6: Rosenbrock's function.

3.3.6 Ackley's Function

The Ackley's function is an n-dimensional highly multimodal function that has a large number of local minima but only one global minimum. This is a widely used as multimodal test function. Its shape for two-dimensional case is shown in Figure 3.7. The function definition is as follows:

$$f_6(x) = -20exp\left(-0.2\sqrt{1/D\sum_{i=1}^{D} x_i^2}\right) - exp\left(1/D\sum_{i=1}^{D} cos(2\pi x_i)\right) + 20 + e \quad (3.6)$$

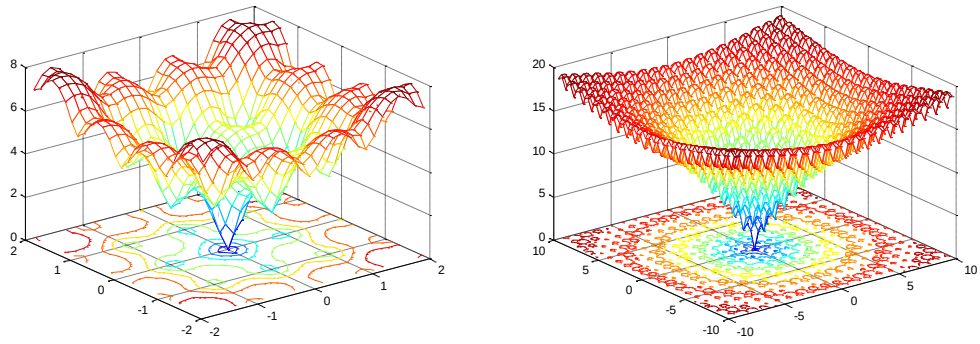

Figure 3.7: Ackley's function.

3.3.7 Griewank's Function

Griewank's function is similar to Rastrigin's function. Although it has many widespread local minima, their locations are regularly distributed. The shape of this function for two-dimensional case is shown in Figure 3.8 and the function definition is as follows:

$$f_7(x) = 1/4000 \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} cos(x_i/\sqrt{i}) + 1 \qquad (3.7)$$

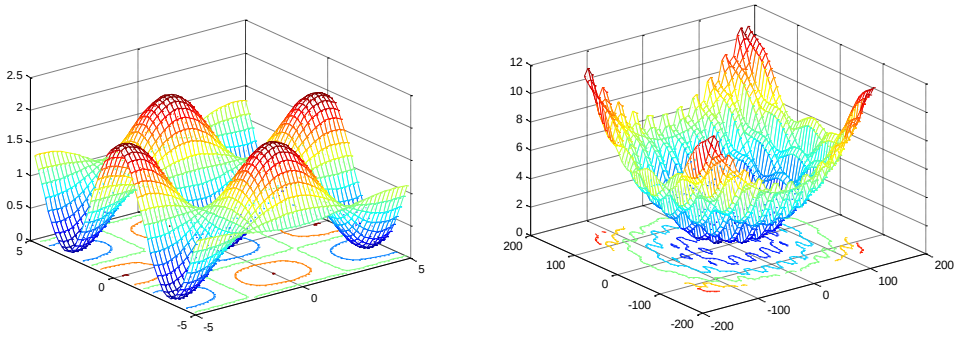

Figure 3.8: Griewank's function.

3.4 Comparisons

This comparison is made for the computation paradigms for seven types of PSO: LPSO, GPSO, CLPSO, DMS–PSO, DMS–PSO-SHS, APSO and UPSO. The comparison is applied to seven types selection of popular benchmarks, where operators of each paradigm are reviewed, focusing on how each algorithm affects it's search behaviour in the problem space. The goals of this process are to provide

additional insights into how each paradigm works, and to suggest ways in which performance might be improved by incorporating features from one paradigm into the others.

3.4.1 Comparison of Algorithms on all Functions

For the purpose of comparison, the mentioned benchmarks were tested under similar scenarios based on the initial conditions: 10,000 iteration (generation), 20 particles, 30 dimensions and mean fitness of 10 trials. The results are presented in the following Figures 3.9-3.15.

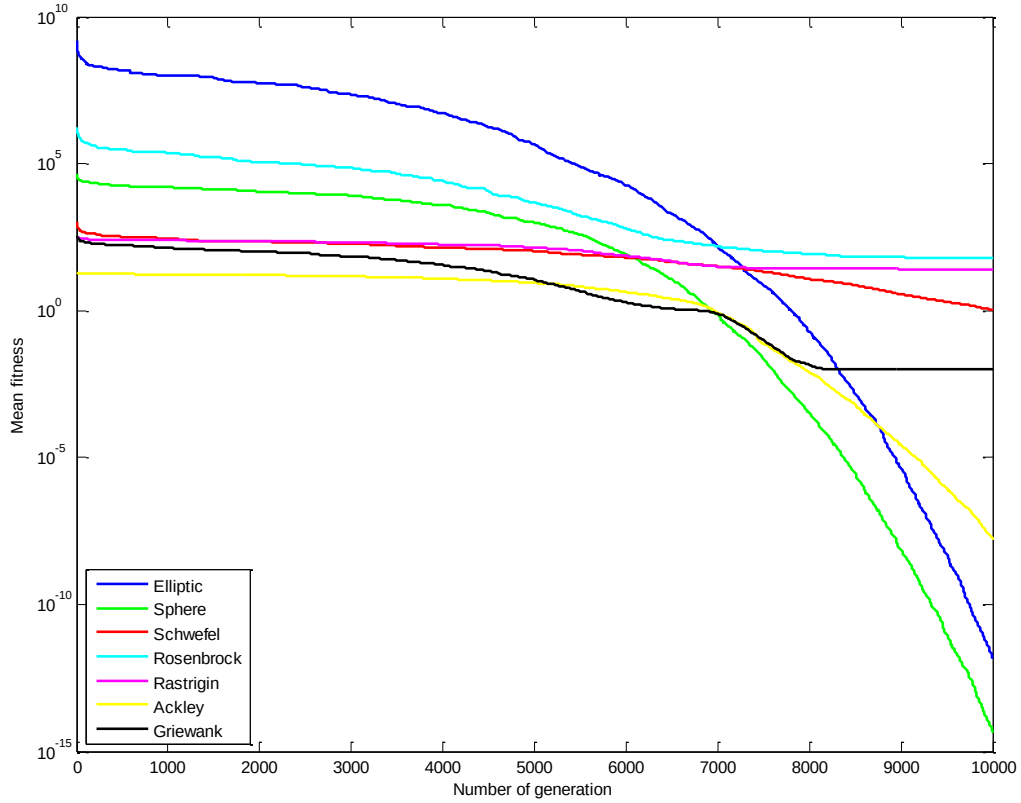

Figure 3.9: Convergence performance of seven test functions on LPSO.

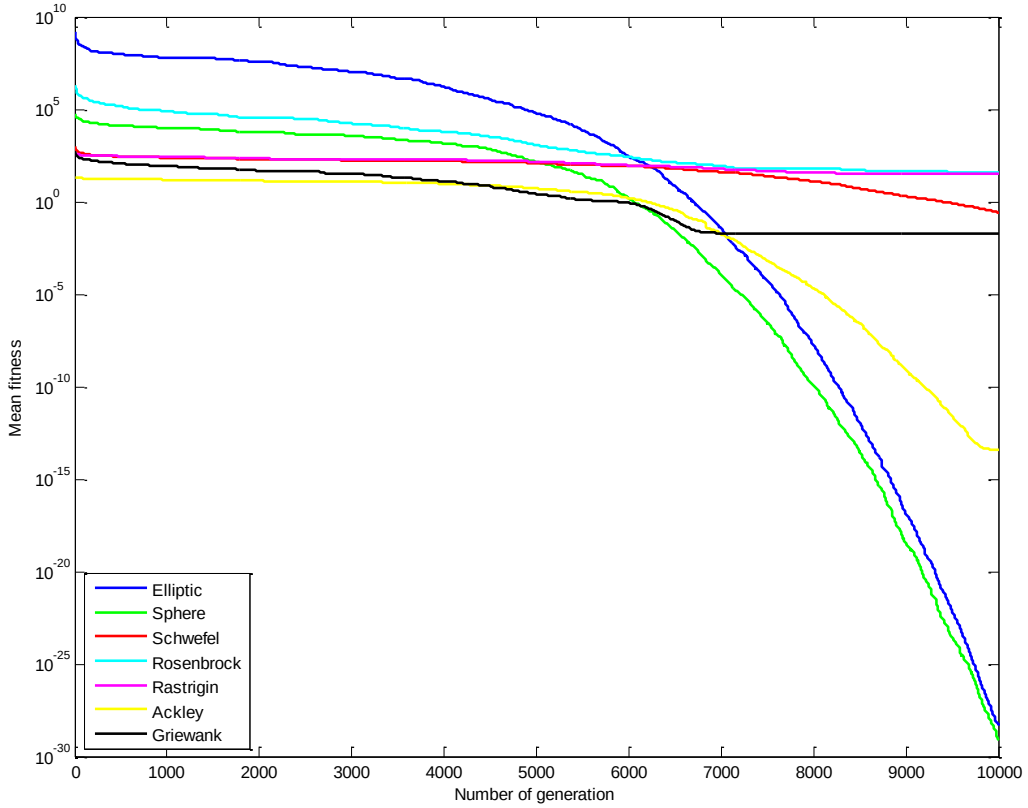

Figure 3.10: Convergence performance of seven test functions on GPSO.

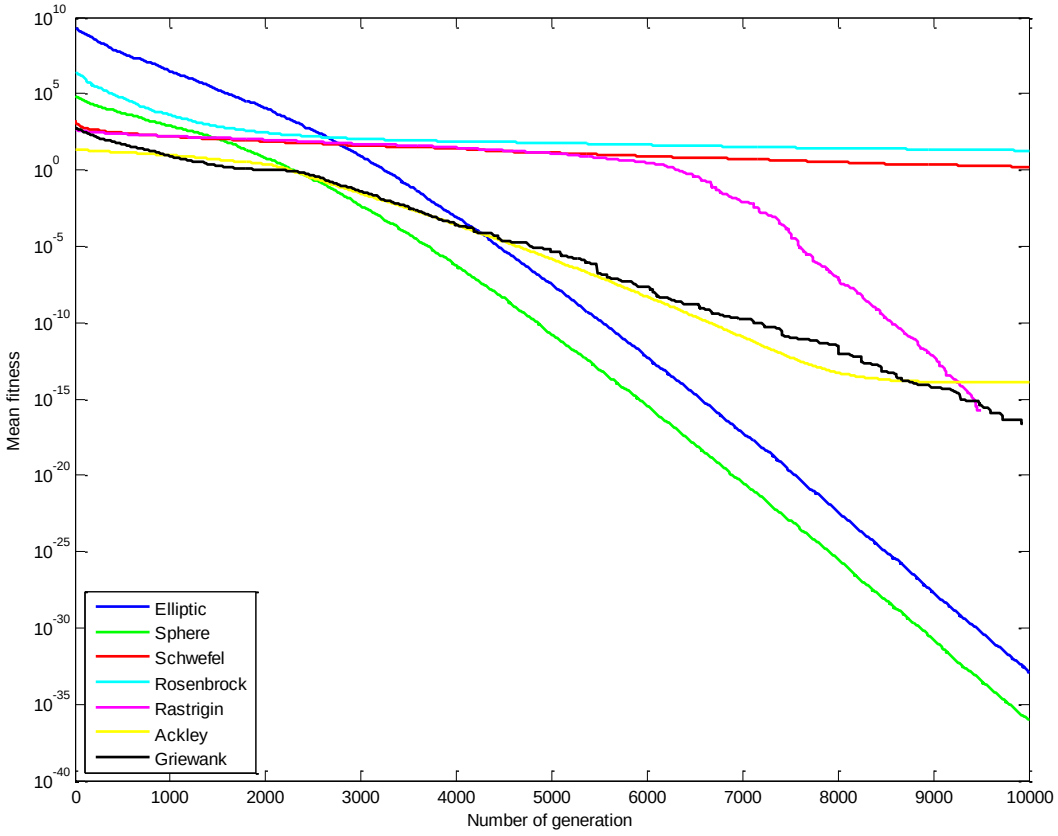

Figure 3.11: Convergence performance of seven test functions on CLPSO.

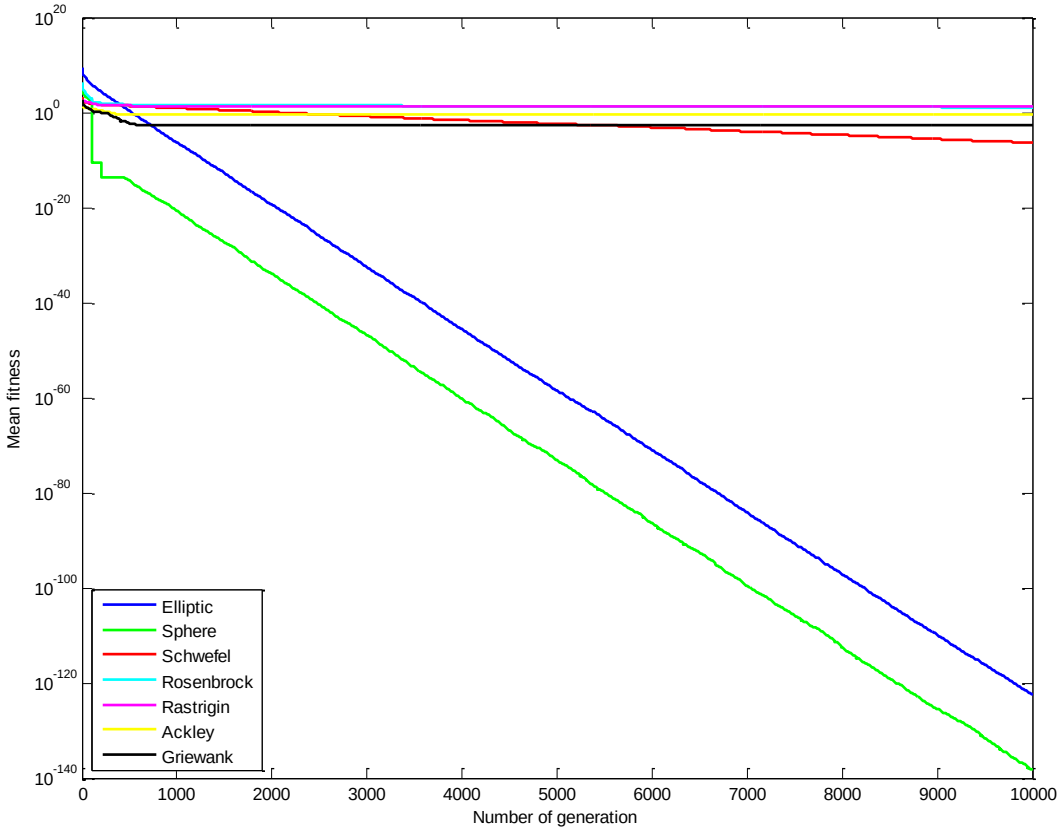

Figure 3.12: Convergence performance of seven test functions on DMS-PSO.

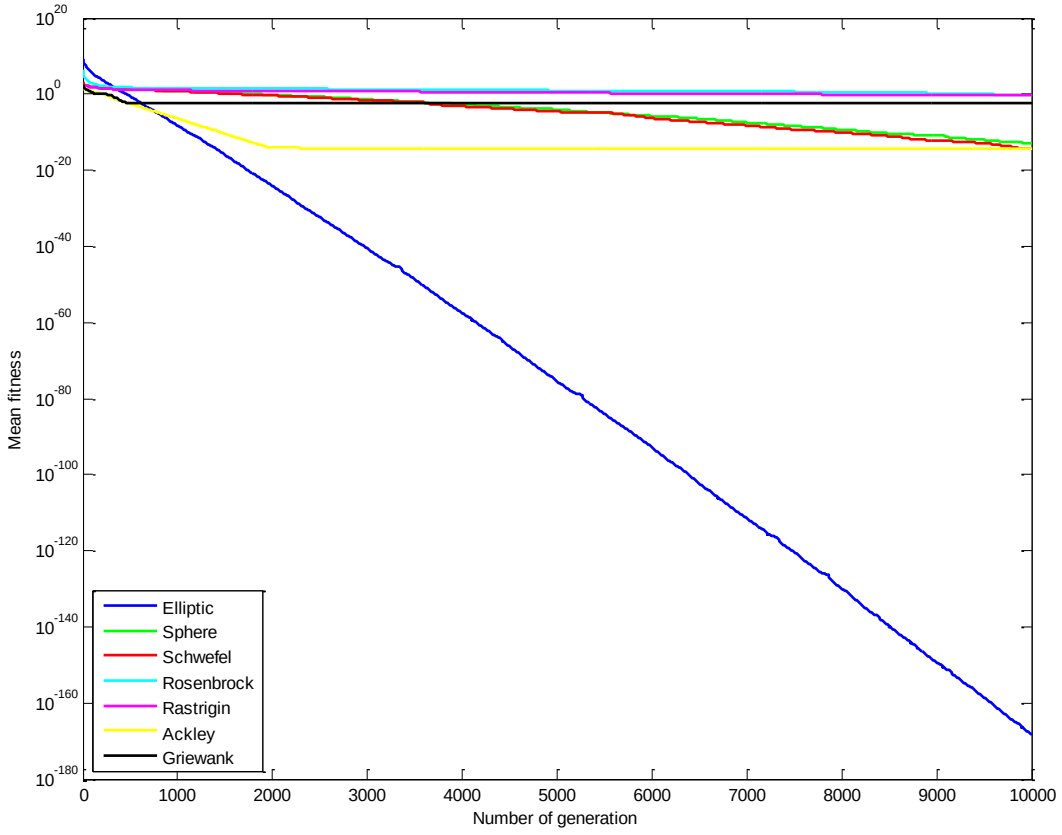

Figure 3.13: Convergence performance of seven test functions on DMS-PSO-SHS.

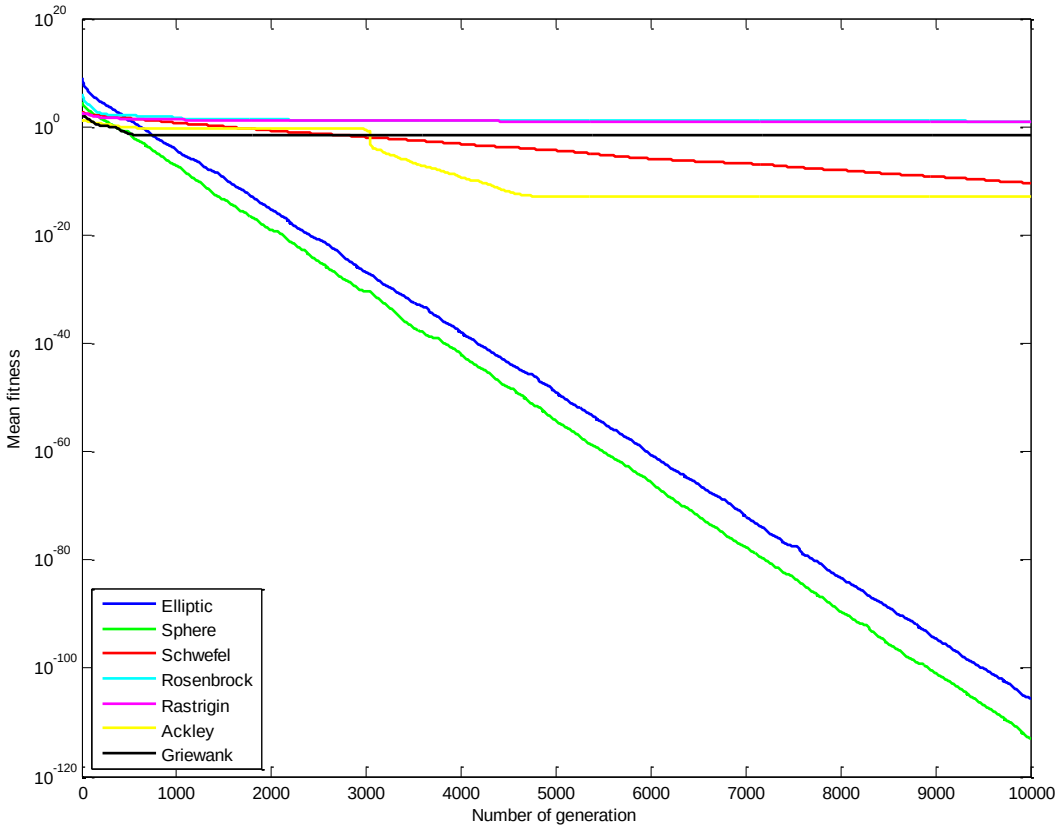

Figure 3.14: Convergence performance of seven test functions on APSO.

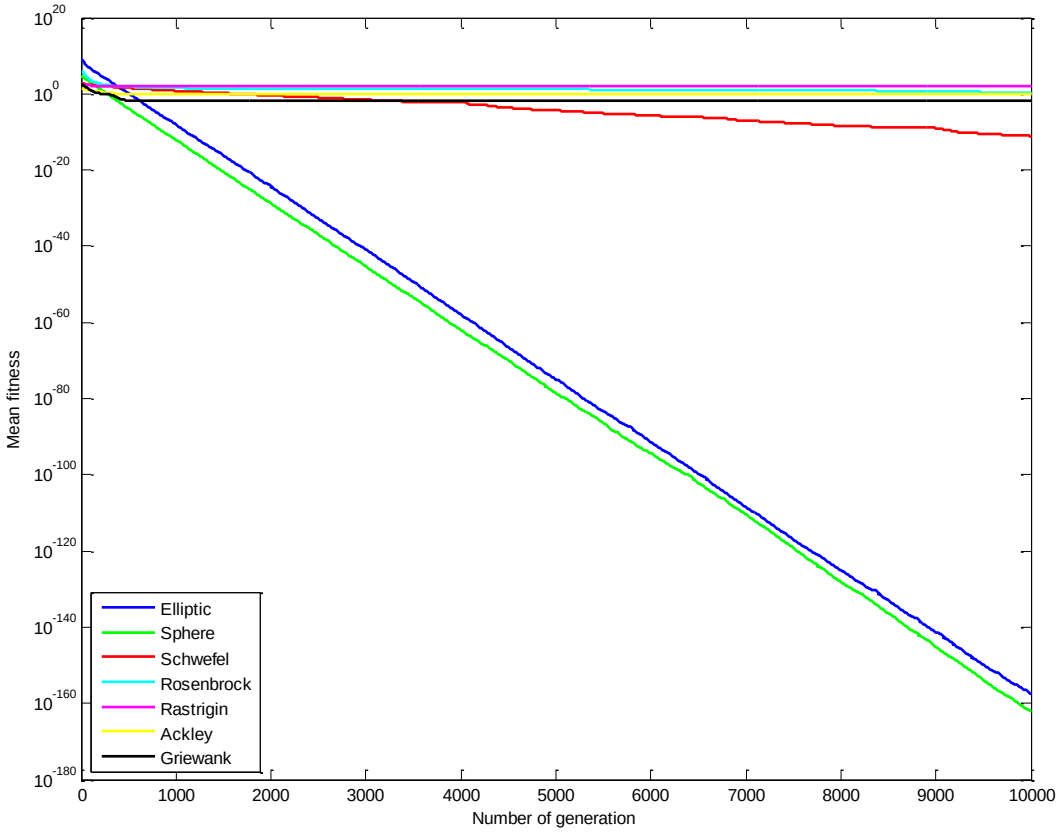

Figure 3.15: Convergence performance of seven test functions on UPSO.

3.4.2 Comparisons of Each Function on all algorithms

The algorithms were tested under the same conditions for the purposes of comparison. The seven algorithms were compared for one function as a benchmark with 10,000 iterations (generation), 20 particles, 30 dimensions and mean fitness of 10 trials, as shown in Table 3.2 and Figures 3.16-3.22.

Table 3.2: Comparison performance of seven PSOs algorithms on seven benchmark functions.

Function	Algorithms	LPSO	GPSO	CLPSO	DMS –PSO	DMS –PSO-SHS	APSO	UPSO
f(1)	Mean	1.5698E-012	5.0916E-029	1.2930E-033	4.1610E-123	**5.5257E-169**	2.9669E-106	3.2869E-158
	Std. Dev.	1.7135E-012	4.5713E-029	8.6152E-034	8.8028E-123	**0**	6.3742E-106	9.6779E-158
f(2)	Mean	4.2524E-015	7.6205E-030	9.8864E-037	5.7091E-139	1.0317E-013	1.2063E-113	**9.6143E-163**
	Std. Dev.	6.5964E-015	2.0462E-029	7.9316E-037	1.7489E-138	1.5867E-013	5.6938E-114	**3.1435E-162**
f(3)	Mean	1.0966	0.2615	1.6034	4.5419E-007	**4.0322E-015**	3.6768E-011	8.4294E-012
	Std. Dev.	0.9239	0.1311	0.4150	2.9708E-007	**3.2000E-015**	3.1343E-011	7.8499E-012
f(4)	Mean	58.9279	39.8112	18.8381	12.7978	**4.618E-001**	9.9841	2.4545
	Std. Dev.	36.8700	32.7277	25.5289	7.2983	**5.716E-001**	7.0320	3.2557
f(5)	Mean	25.7694	31.1422	**0**	4.975E-001	4.975E-001	7.3627	71.1395
	Std. Dev.	4.7008	10.1796	**0**	9.669E-001	9.669E-001	2.6882	12.8298
f(6)	Mean	1.8281E-008	4.0146E-014	1.2079E-014	3.018E-001	**3.9080E-015**	1.4069E-013	1.1224
	Std. Dev.	1.9793E-008	1.4017E-014	2.4841E-015	4.897E-001	**1.1235E-015**	1.6497E-013	1.2553
f(7)	Mean	0.0098	0.0194	**0**	2.0E-003	3.9E-003	2.45E-002	1.06E-002
	Std. Dev.	0.0136	0.0309	**0**	4.2E-003	7.0E-003	3.06E-002	1.04E-002

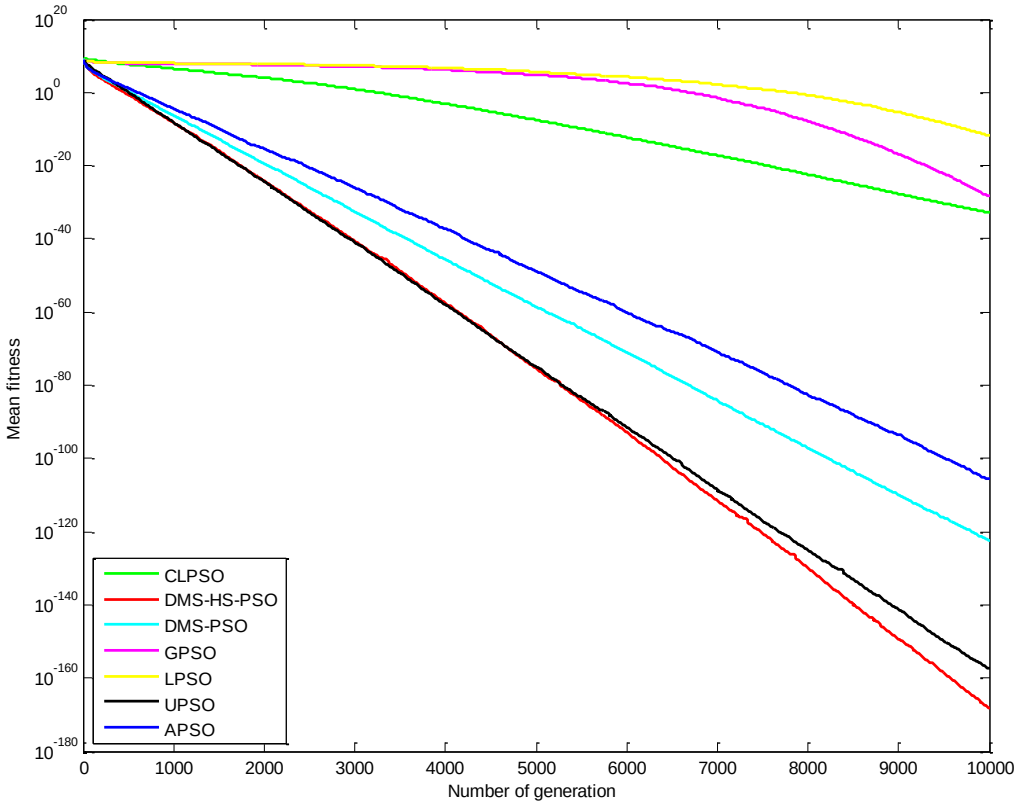

Figure 3.16: Convergence performance of seven PSOs on Elliptic function.

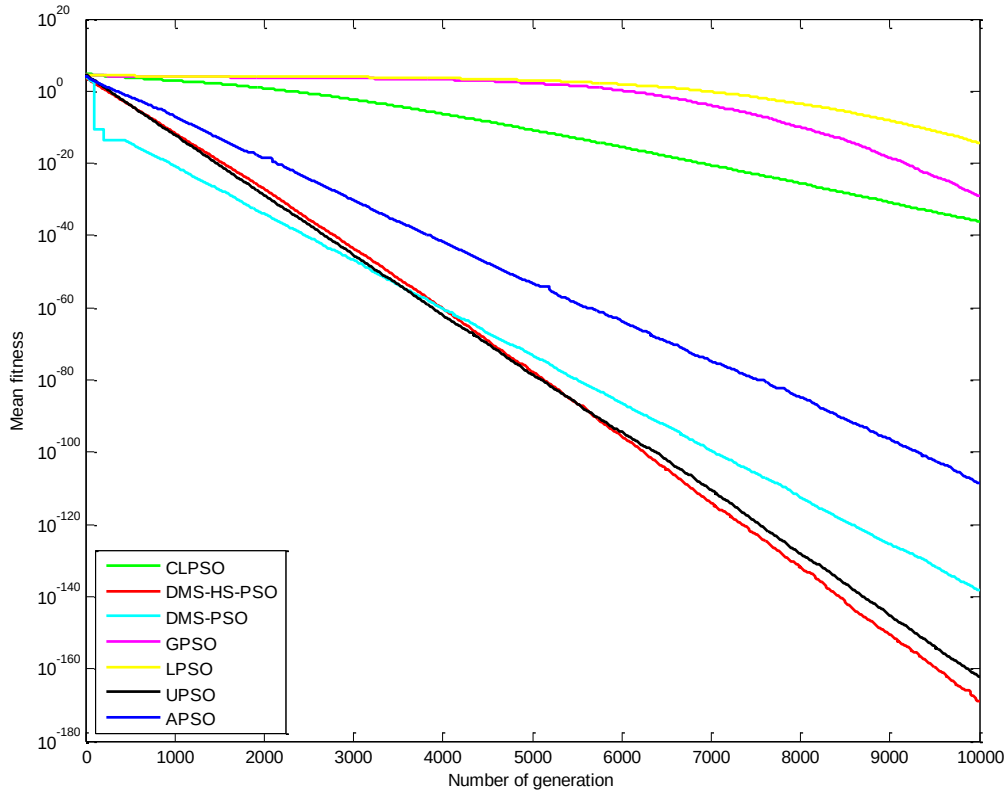

Figure 3.17: Convergence performance of seven PSOs on Sphere function.

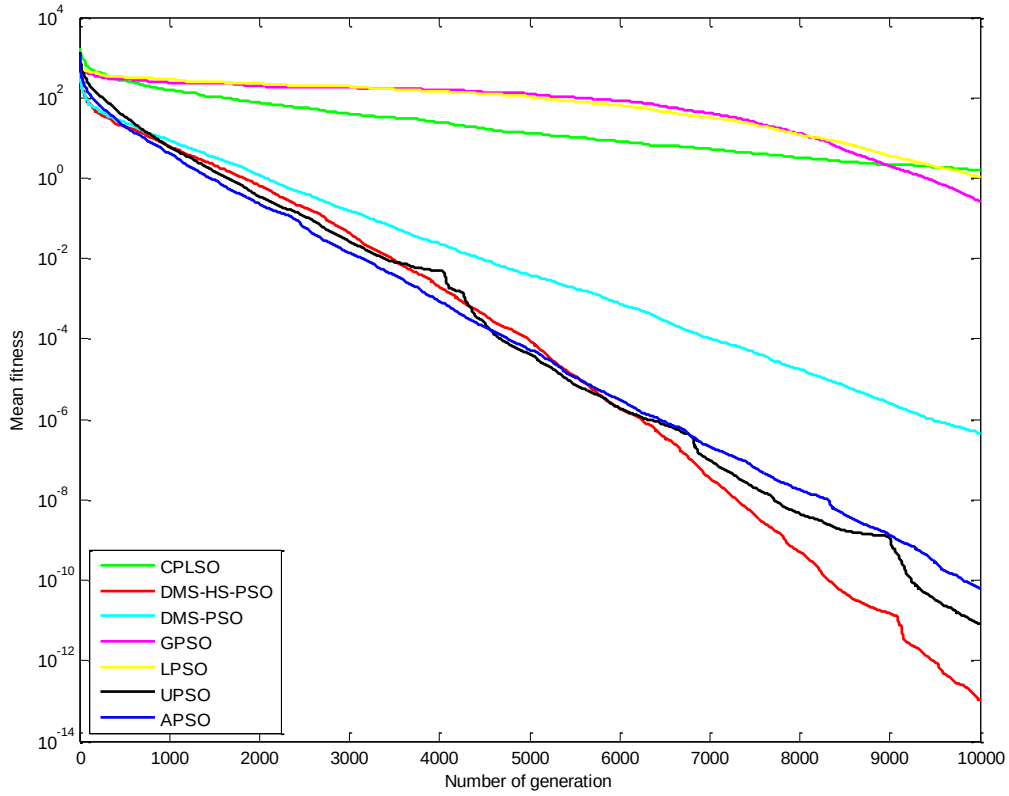

Figure 3.18: Convergence performance of seven PSOs on Schwefel's P2.22 function.

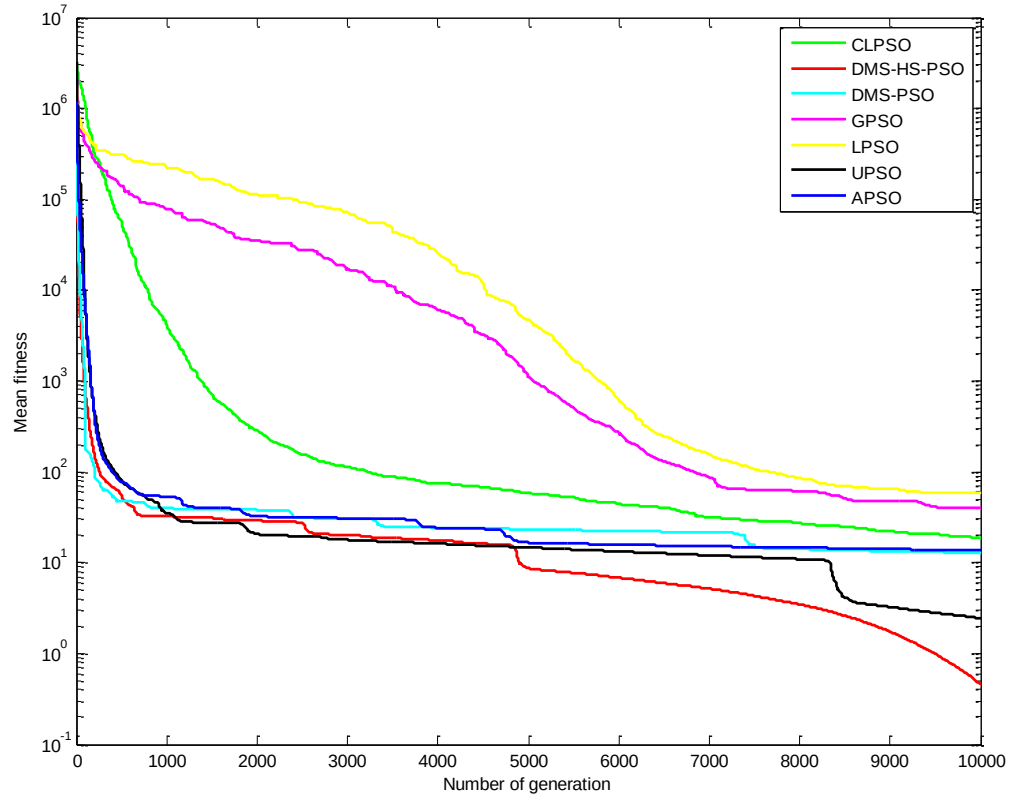

Figure 3.19: Convergence performance of seven PSOs on Rosenbrock's function.

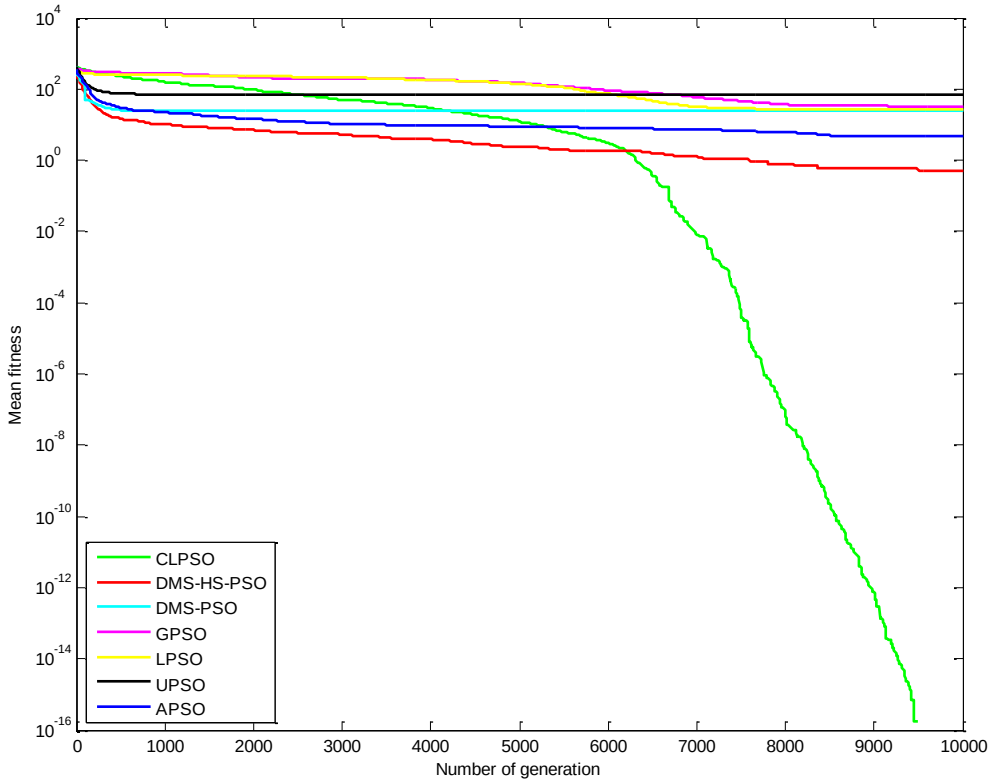

Figure 3.20: Convergence performance of seven PSOs on Rastrigin's function.

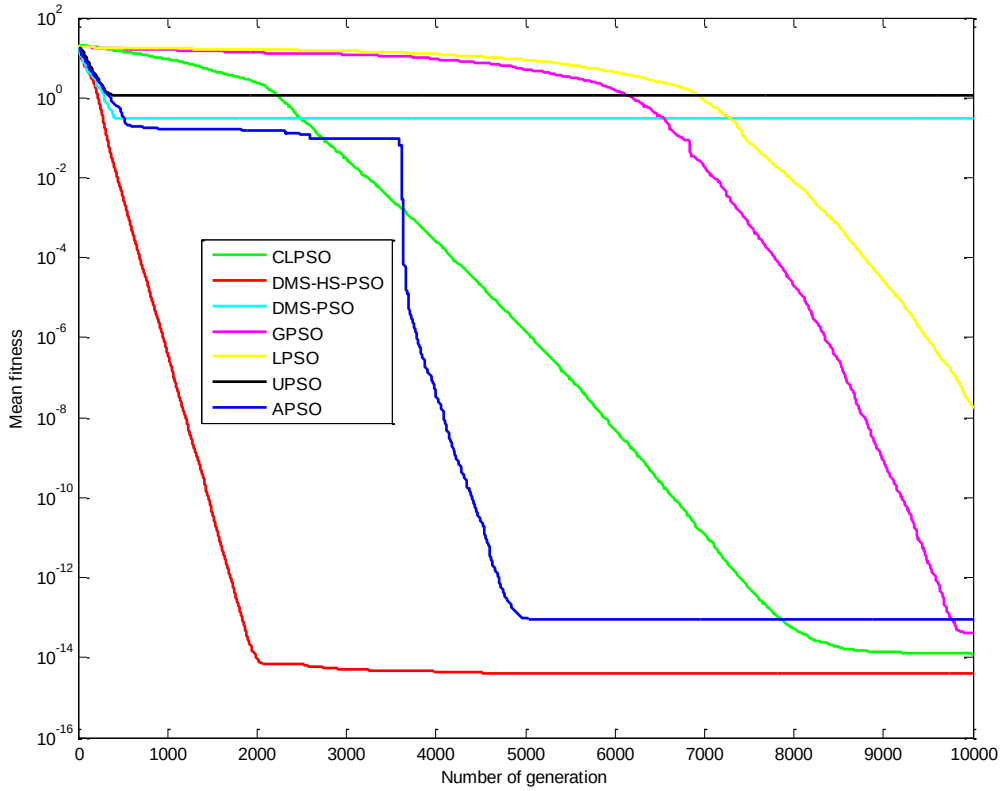

Figure 3.21: Convergence performance of seven PSOs on Ackley's function.

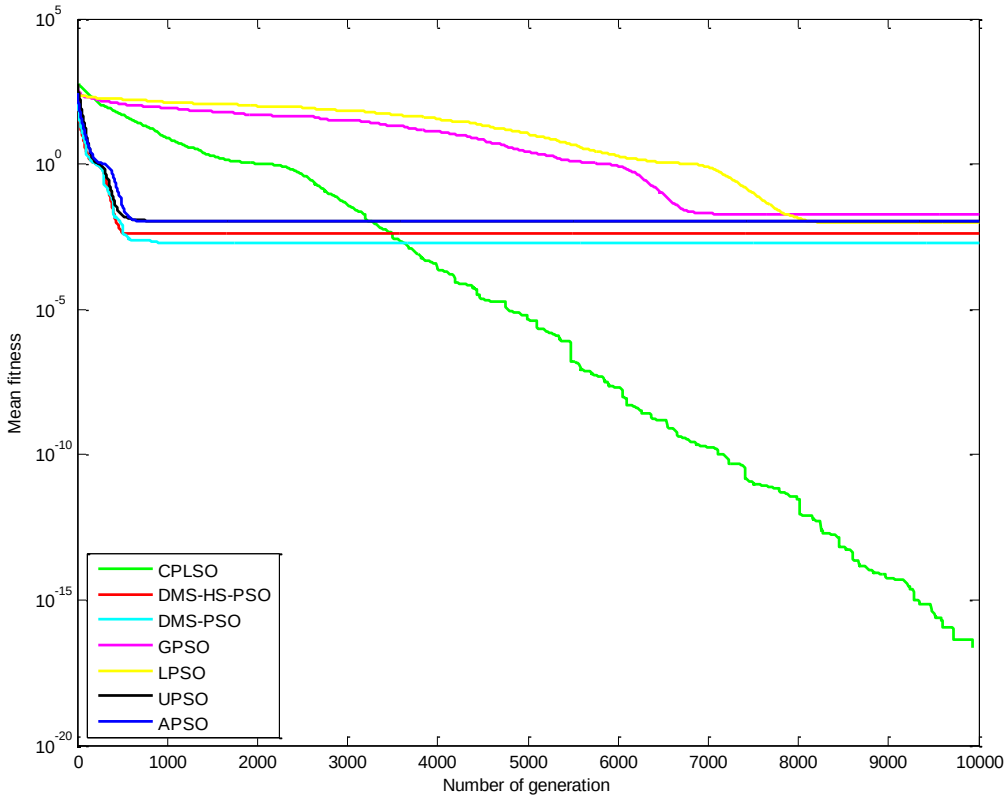

Figure 3.22: Convergence performance of seven PSOs on Griewank's function.

The comparison results illustrate that each algorithm has its advantages and disadvantages. Some of them achieve high performance on the uni-modal function, where they reach the right solution with very small error in a short time. Conversely, the same algorithm does not achieve the same efficiency on the multi-modal function.

For example as shown in Figure 3.23, 3.17, 3.18, 3.19 and 3.21) the best algorithms are DMS-PSO-SHS, UPSO and APSO, which work very well on the following benchmark functions: elliptic, sphere, Schwefel's P2.22, Rosenbrock and Ackely, while the others have varying performance from function to function. Alternatively DMS-PSO-SHS, UPSO and APSO are overdue in performance compared to CLPSO on the Rasrtrigin and Griewank functions, as shown in Figure 3.23 and 3.22).

3.4.3 Algorithms Efficiency Comparisons

Virtual efficiency comparison was undertaken for on all logarithms to test their efficiency in terms of accuracy speed and convergence with the selected benchmark.

A scale of 1 to 7 was used to measure the accuracy and convergence where 1 is the lowest and 7 is the highest, as shown in Table 3.3 and

Table 3.4, respectively.

Table 3.3: Accuracy comparison by rank (1-7: lowest-highest).

Function / Algorithms	*f(1)*	*f(2)*	*f(3)*	*f(4)*	*f(5)*	*f(6)*	*f(7)*	**Av.**
DMS-HS-PSO	7	7	7	7	6	7	5	6.6
UPSO	6	6	6	6	1	2	5	4.6
APSO	4	4	5	4	5	5	4	4.4
DMS-PSO	5	5	4	5	3	3	6	4.4
CLPSO	3	3	1	3	7	6	7	4.3
GPSO	2	2	3	2	4	5	2	2.9
LPSO	1	1	2	1	2	1	1	1.3

Table 3.4: Convergence speed comparison by rank (1-7: lowest-highest).

Function / Algorithms	*f(1)*	*f(2)*	*f(3)*	*f(4)*	*f(5)*	*f(6)*	*f(7)*	***Av.***
MS-HS-PSO	7	5	7	7	7	7	6	6.6
DMS-PSO	6	7	6	6	6	5	7	6.1
UPSO	6	6	4	5	4	4	5	4.9
APSO	5	4	5	4	5	6	4	4.7
CLPSO	3	3	3	3	3	3	3	3
GPSO	2	2	2	2	2	2	2	2
LPSO	1	1	1	1	1	1	1	1

3.5 New Combination of PSOs

After the tests and comparisons were conducted for all types of PSO algorithms by testing each one separately against a different set of benchmarks, and observing the response of each algorithm on all the benchmark functions' tests in terms of convergence and accuracy, a new method containing all previous algorithms was established. Each PSO is involved in processing the same population at each iteration, in two structures parallel and serial.

3.5.1 Parallel Structure (PSOs)

In this stage, the structure is built for all algorithms in parallel, so that all algorithms contend with the same population at the same time, being processed through a specific number of internal iterations. The experiment revealed that the internal iteration number must be between three to seven iterations. In this research the number of iterations was selected as five, which gives the best experimental results in terms of convergence for all problems tested, as shown in Figure 3.23.

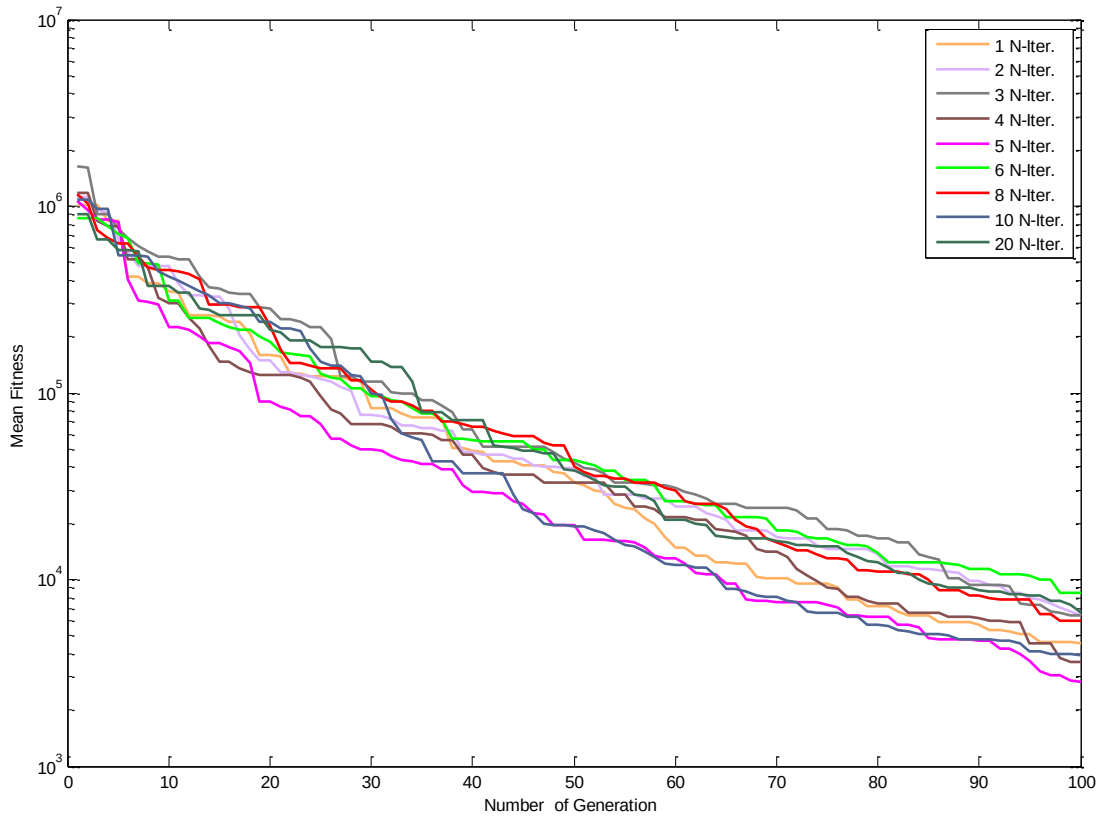

Figure 3.23: Comparison of the number of internal iteration.

In the next phase, the results are collected from all the algorithms and revised to recognize the best of all local best *pbest* and best of all global best *gbest* from all proposed solutions. A new population can select with the same size of the original one by sorted order, including the best of all best particles, as indicated in the flowchart shown in Figure 3.24.

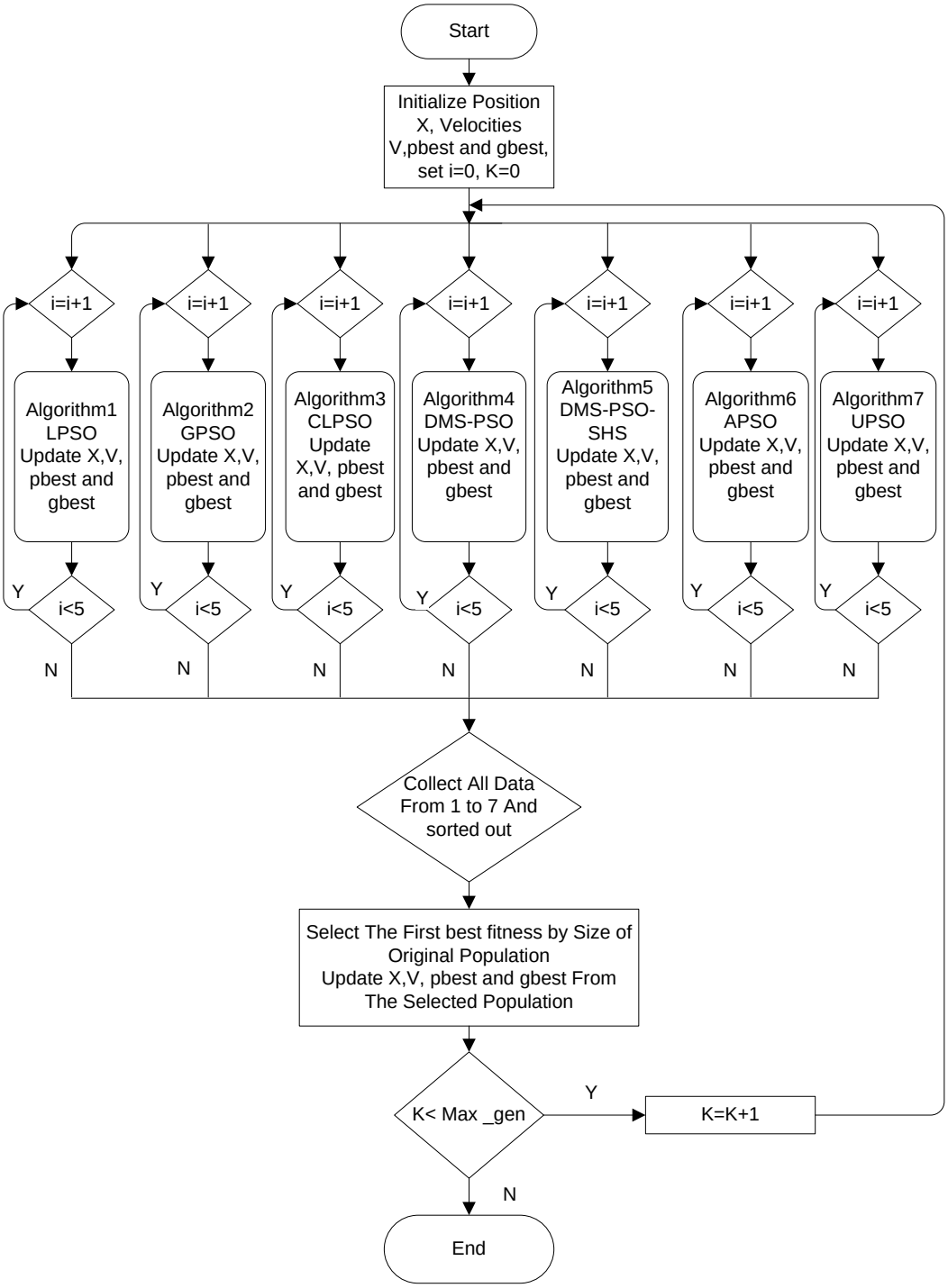

Figure 3.24: Parallel PSO flowchart.

3.5.2 Sequential Structure (PSOs)

For this technique the structure is built for all algorithms in series, as explained in the literature review each type of PSO algorithm has specific characteristic. In this case the algorithm starts with the simple and fast solution first and then the most complex and more accurate in solution subsequent, so all the algorithms are tackling the same

population individually one after another, which is processed with a specific number of internal iterations. From experience it was found that the internal iteration numbers must be among three to seven, as shown in Figure 3.24. Therefore, in this research the number of internal iterations was set to five as a result of the experiments that identified this number as the most efficient in terms of convergence.

Each algorithm received the latest updated information such as *pbest*, *gbest* and the last velocity and position (*v*,*x*) for each particle from the previous algorithm, and the best results obtained were set as initial values for the current algorithm, to be processed as described previously. After that the population delivered to the next algorithm. This was repeated until the seventh algorithm, which then hands over its findings to the first algorithm to complete the cycle until the last iteration. In this case the population is the same size during the whole process, including the best of all best particles in each step. This process is illustrated in the flowchart shown in Figure 3.25.

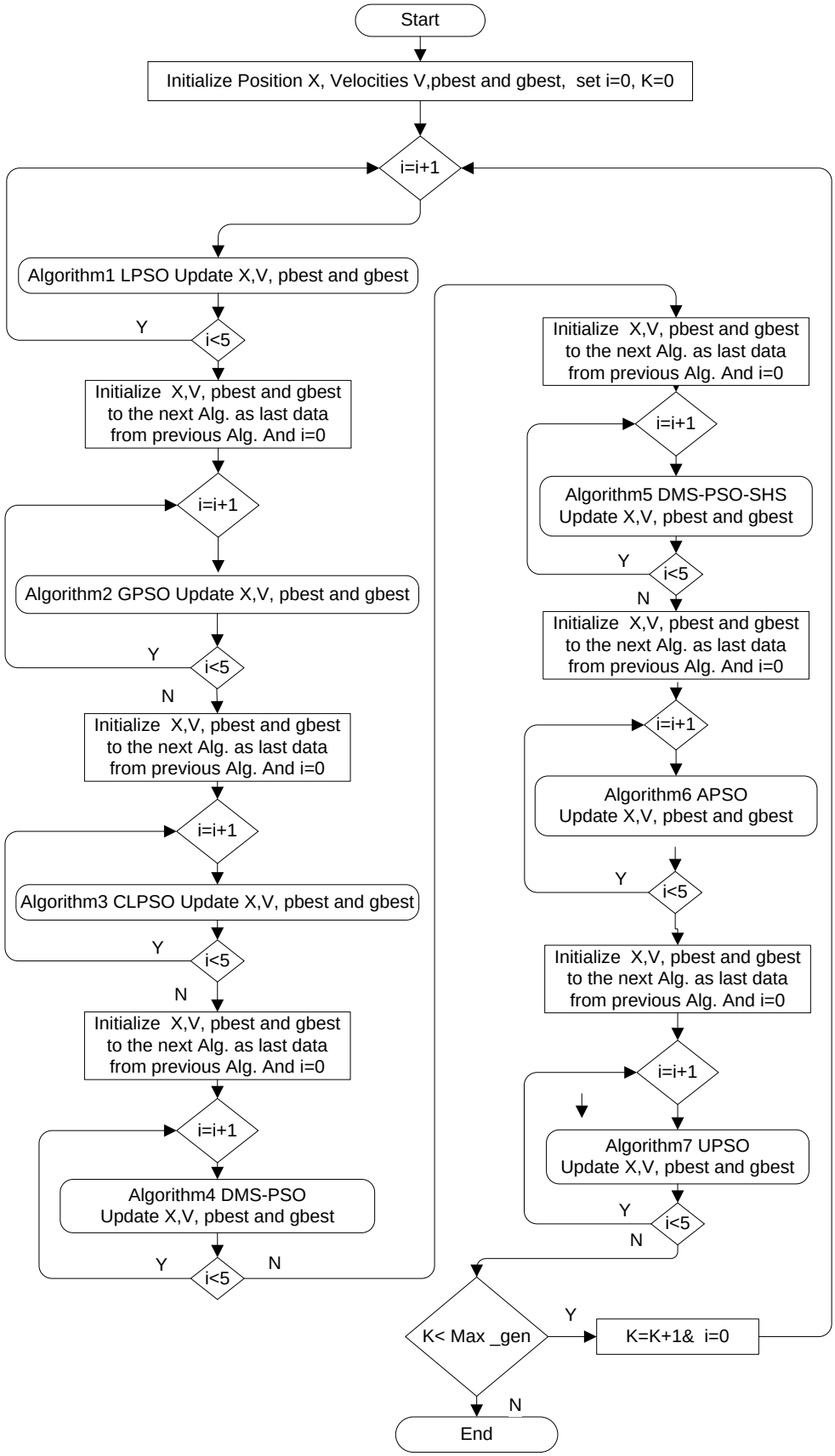

Figure 3.25: Serial PSO flowchart.

3.6 Experiments

All experiments were executed to evaluate the performance of the proposed PSOs algorithms on the same benchmark functions described previously. The performance of PSO algorithms was evaluated in comparison with seven other variations described before. Two PSO versions were used, PPSO and sequential (SPSO). The main objective is to improve PSOs algorithm on wide range of uni-modal and multi-modal functions.

3.6.1 Experimental Setting

Experiments revealed that SPSO achieved better results than PPSO. The test run comparisons are performed for all candidate types of PSOs by the enforcement of each algorithm separately on a different set of benchmarks, as well as the new method of SPSO, and observing the response of each algorithm on all selected benchmark functions in terms of convergence and accuracy. The results shown in Figures 3.26-2.29 illustrate the difference between the SPSO and PPSO in comparison to all other algorithms. In order to cover the two different types of systems modal, four benchmarks were selected: sphere together with Rosenbrock as uni-modal and Rastrigin with Griewank as multimodal function system.

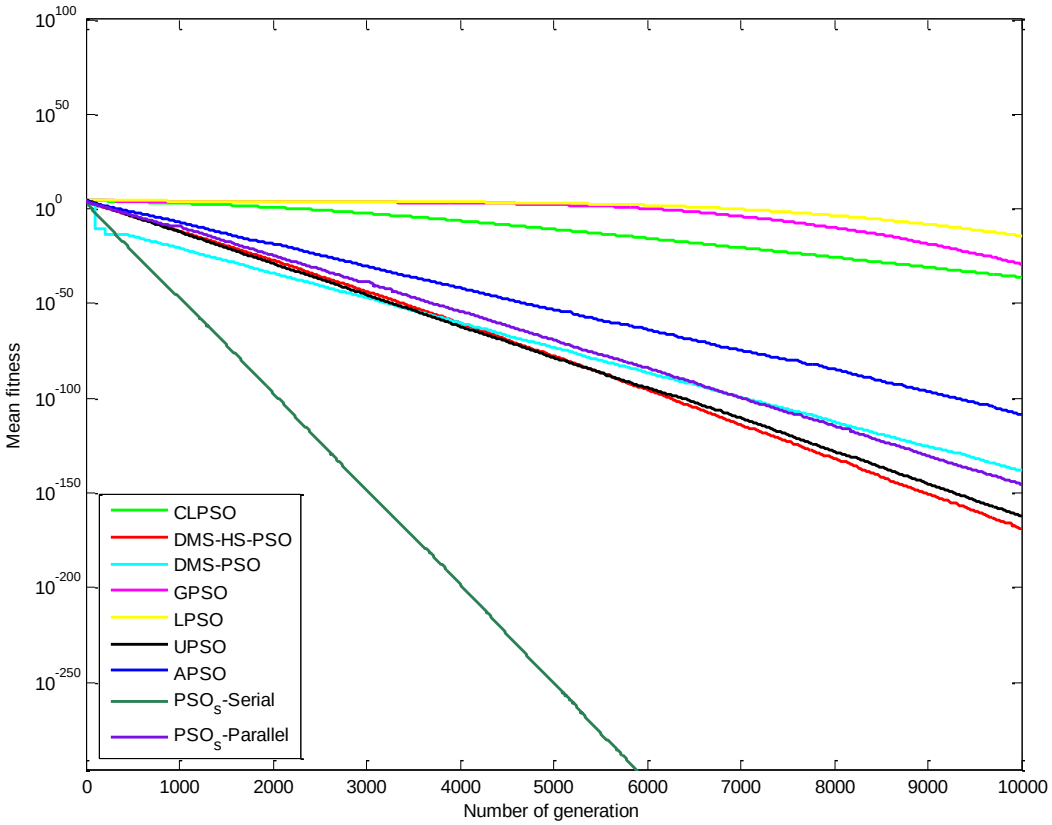

Figure 3.26: Comparison of SPSO and PPSO performances using Sphere algorithm.

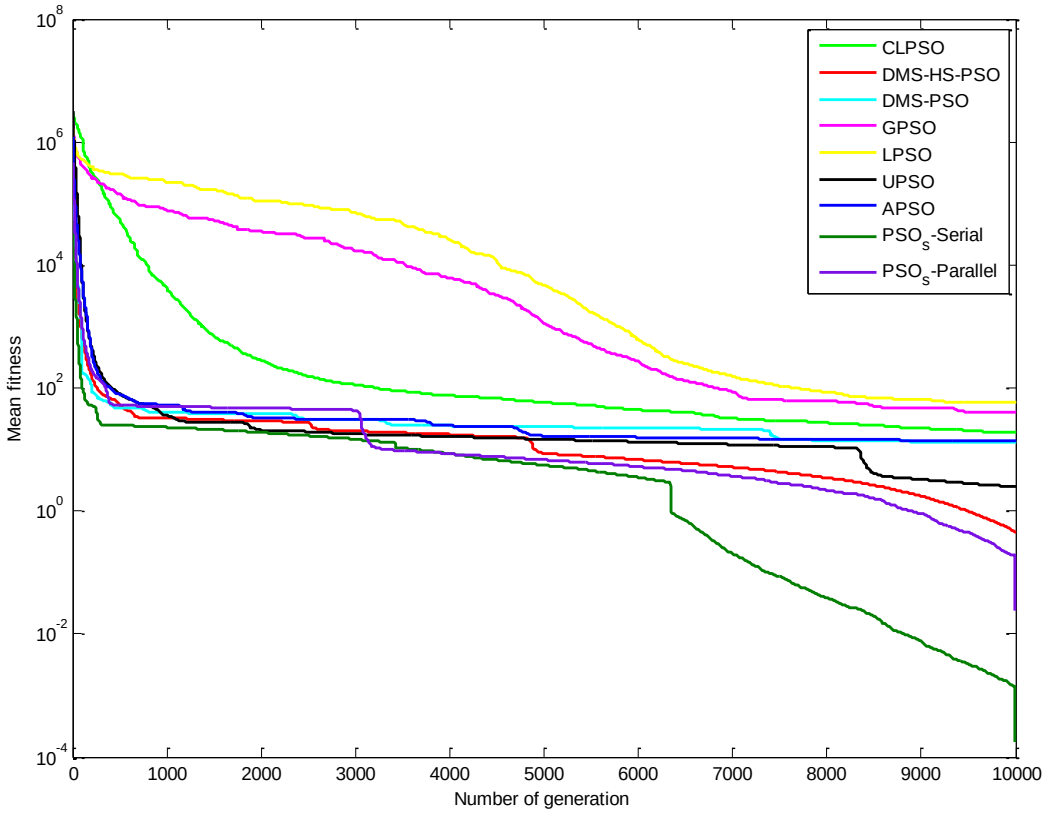

Figure 3.27: Comparison of SPSO and PPSO performances using Rosenbrock's algorithm.

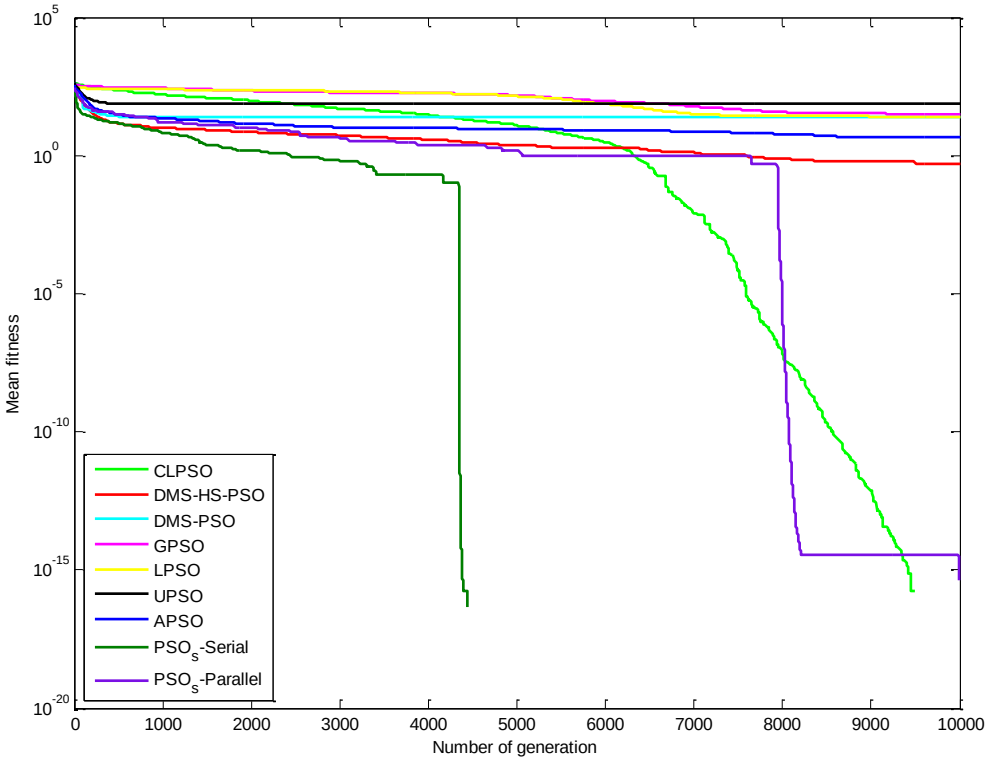

Figure 3.28: Comparison of SPSO and PPSO performances using Rastrigin's algorithm.

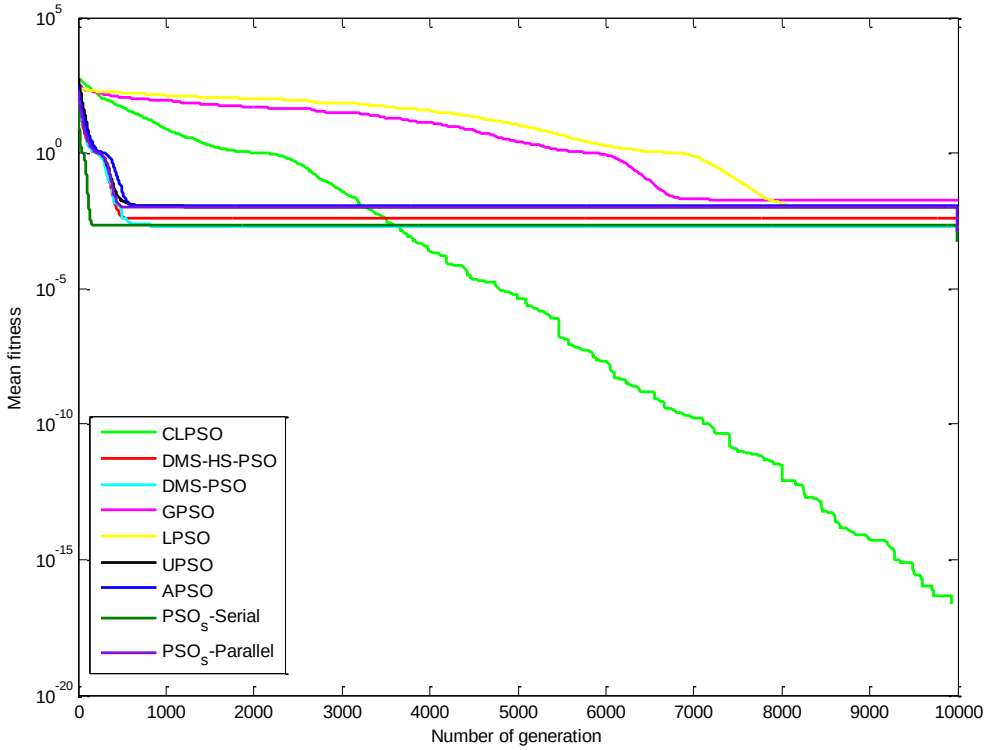

Figure 3.29: Comparison of SPSO and PPSO performances using Griewank's algorithm.

3.6.2 Comparison of New Method Algorithm on all Functions

The test of the new methods was conducted on all benchmarks. The seven algorithms were compared with the new method on one function as a benchmark each time, with 10,000 iterations (generation), 20 particles, 30 dimensions and mean fitness of 10 trials, as shown in Table 3.6 and in Figures 3.30-3.36.

3.6.3 Investigations of the Solution Accuracy

The performance of the result exactness of each PSO recorded in.

Table 3.2 is compared with the SPSO. The results illustrated in Table 3.5 are obtained for 10 independent runs for every algorithm. The boldface text in the table shows the best performance among those acquired by each of the eight contenders. Figures 3.30-3.36 display the graphs of all PSOs indicating the speed of convergence. The results displayed in both Figures 3.30-3.36 and Table 3.6 demonstrate that SPSO offers the best performance on most test functions when solving uni-modal system; it is particularly accurate on functions f_1, and f_2. SPSO also achieves good results when testing the complex multi-modal functions f_3, f_4, f_5 and f_6. Although CLPSO outperforms SPSO and others on f_7 (Griewank's function), the solutions of other functions are worse than those of the SPSO. Furthermore, SPSO successfully jumps out of the local optima on most of the multimodal functions and exceeds all the other algorithms on functions $f_1,\ldots,f_6$, where the global optimum of f_3 (Schwefel's function) is far away from any of the local optima (Yao, Liu, & Lin, 1999), and the globally best solutions of f_5 (continuous/non-continuous Rastrigin's functions) are surrounded by a large number of local optimas (J. J. Liang, Qin, Suganthan, & Baskar, 2006; Yang & Li, 2010). The capability of abstaining from being trapped into the local optima and achieving global optimal solutions on multimodal functions suggests that the SPSO can indeed benefit from the elitist learning strategy with respect to its much improved performance, which was superior to all other algorithms which utilized a similar technique.

3.6.4 Comparisons on the Convergence Speed

The speed in obtaining the global optimum is also a noticeable measuring criterion for measuring the algorithm performance. Results reveal that the SPSO generally

offer a much higher speed, which is measured by either the mean number of function evaluations or by the mean CPU time needed to reach an acceptable solution. The CPU time is important to measure the computational load, as many existing PSO variants have added extra operations that require more computational time. Although the SPSO need to calculate the mean distance between every pair of particles in the swarm, the calculation costs are negligible. Figures 3.30-3.34 illustrate that SPSO offers the highest convergence speed for the right solution in about 6000 iterations on functions f_1, and f_2 (elliptic and sphere) as uni-model system, and achieves very accurate solutions 5.2454×10^{-52} and 1.0349×10^{-10} of complex multimodal functions f_3, f_4 and f_5 (Schwefel's, Rosenbrock and Rastrigin). Although DMS-HS-PSO outperforms the SPSO on the convergences speed with right solution on function f_6 (Ackley) as an error of 0.1×10^{-14} in iteration 2500, the SPSO reaches the same value in 500 iterations, as shown in Figure 3.35. Similarly, despite the fact that CLPSO achieves the nearest results on function f_7 (Griewank), SPSO reaches very good results of a fitness of 1×10^{-3} in a short time, about 200 iteration compared to the same results, where CLPSO achieves this in 4000 iterations, as shown in Figure 3.36.

Table 3.6: Mean and standard deviation results for the 7 benchmark functions using different PSO types.

Function \ Algorithms		LPSO	GPSO	CLPSO	DMS – PSO	DMS – PSO-SHS	APSO	UPSO	SPSO
f(1)	Mean	1.5698E-012	5.0916E-029	1.2930E-033	4.1610E-123	5.5257E-169	2.9669E-106	3.2869E-158	**0 ≤ E-325**
	Std. Dev.	1.7135E-012	4.5713E-029	8.6152E-034	8.8028E-123	0	6.3742E-106	9.6779E-158	**0 ≤ E-325**
f(2)	Mean	4.2524E-015	7.6205E-030	9.8864E-037	5.7091E-139	1.0317E-013	1.2063E-113	9.6143E-163	**0 ≤ E-325**
	Std. Dev.	6.5964E-015	2.0462E-029	7.9316E-037	1.7489E-138	1.5867E-013	5.6938E-114	3.1435E-162	**0 ≤ E-325**
f(3)	Mean	1.0966	0.2615	1.6034	4.5419E-007	4.0322E-015	3.6768E-011	8.4294E-012	**5.2454E-052**
	Std. Dev.	0.9239	0.1311	0.4150	2.9708E-007	3.2000E-015	3.1343E-011	7.8499E-012	**6.9377E-052**
f(4)	Mean	58.9279	39.8112	18.8381	12.7978	4.618E-001	9.9841	2.4545	**1.0349E-010**
	Std. Dev.	36.8700	32.7277	25.5289	7.2983	5.716E-001	7.0320	3.2557	**1.513E-010**
f(5)	Mean	25.7694	31.1422	**0 ≤ E-325**	4.975E-001	4.975E-001	7.3627	71.1395	**0 ≤ E-325**
	Std. Dev.	4.7008	10.1796	**0 ≤ E-325**	9.669E-001	9.669E-001	2.6882	12.8298	**0 ≤ E-325**
f(6)	Mean	1.8281E-008	4.0146E-014	1.2079E-014	3.018E-001	3.9080E-015	1.4069E-013	1.1224	**2.2E-015**
	Std. Dev.	1.9793E-008	1.4017E-014	2.4841E-015	4.897E-001	1.1235E-015	1.6497E-013	1.2553	**5.0E-015**
f(7)	Mean	0.0098	0.0194	**0 ≤ E-325**	2.0E-003	3.9E-003	2.45E-002	1.06E-002	3.553E-015
	Std. Dev.	0.0136	0.0309	**0 ≤ E-325**	4.2E-003	7.0E-003	3.06E-002	1.04E-002	0 ≤ E-325

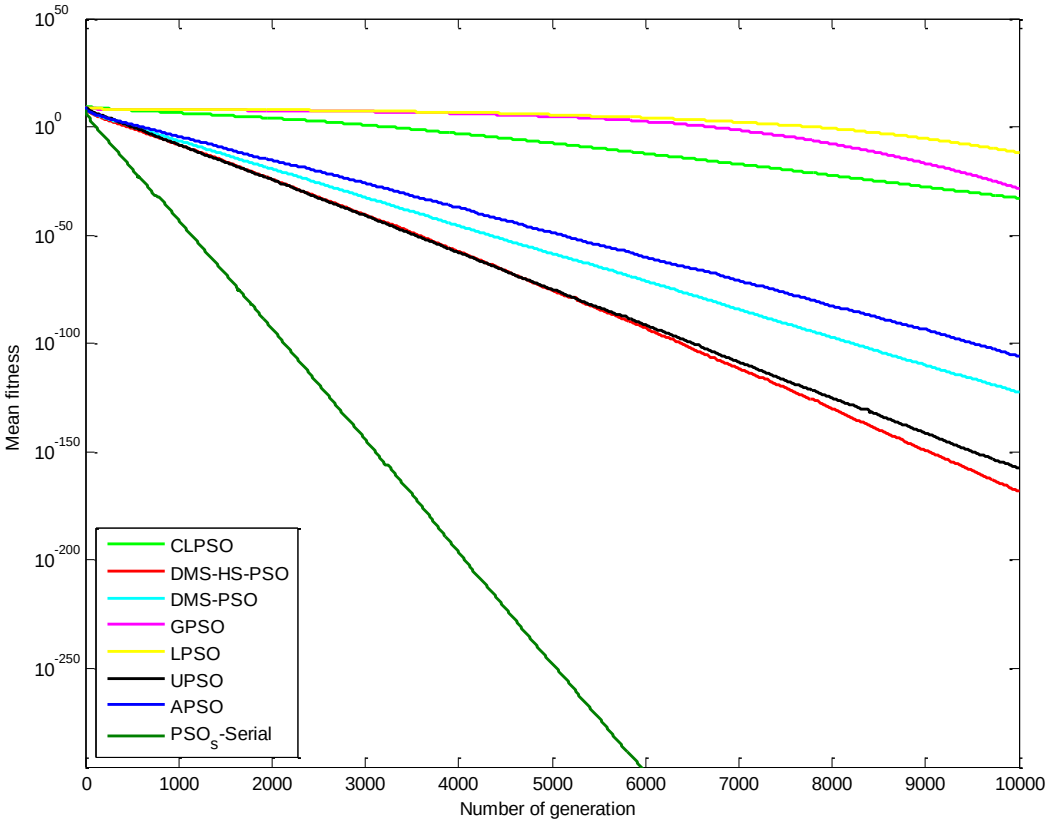

Figure 3.30: Comparison of new method with other algorithms responses on Elliptic function.

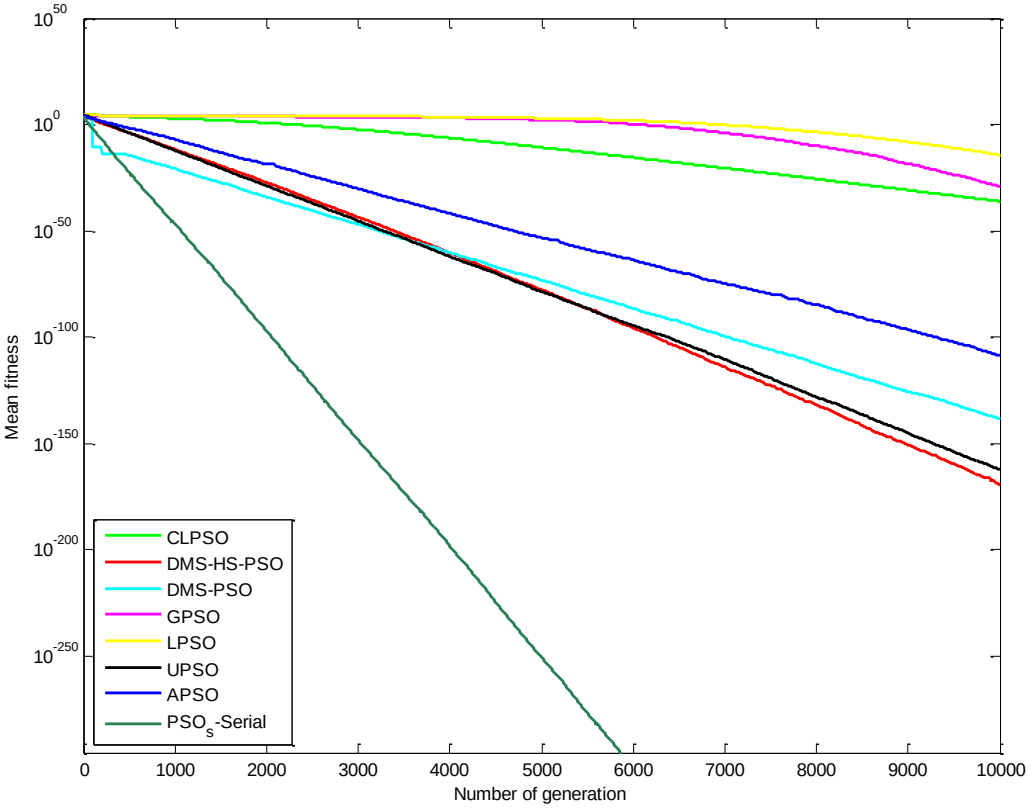

Figure 3.31: Comparison of new method with other algorithms responses on Sphere function.

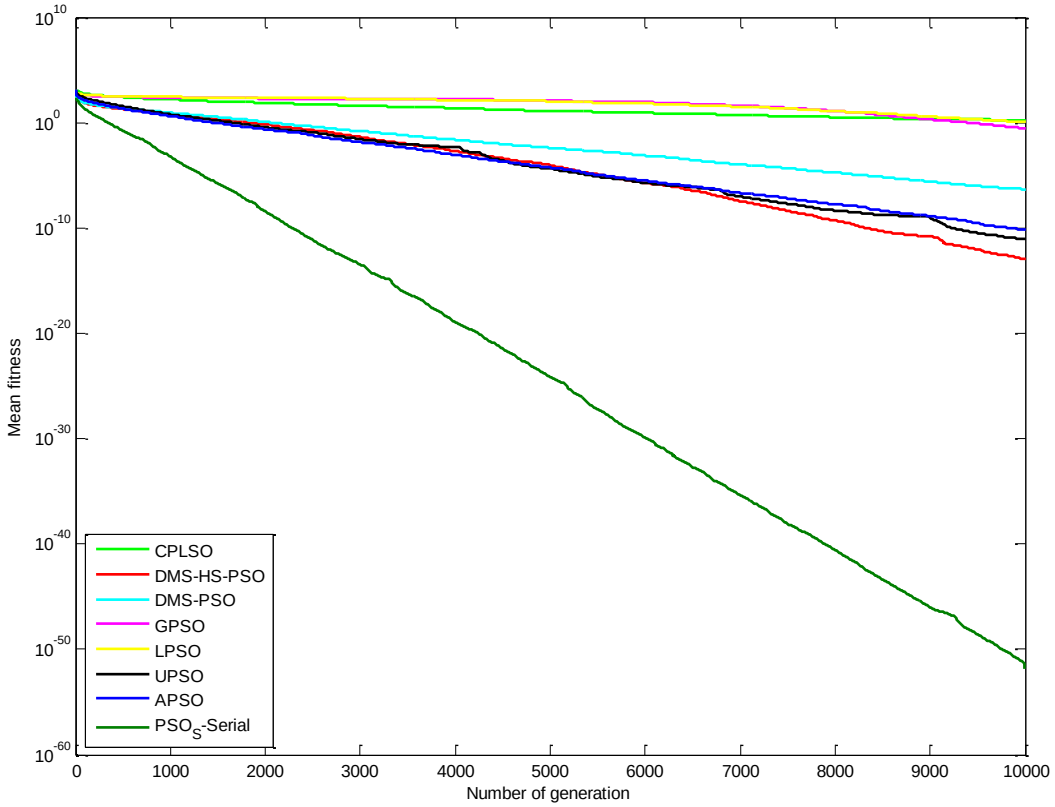

Figure 3.32: Comparison of new method with other algorithms responses on Schwefel's P2.22 function.

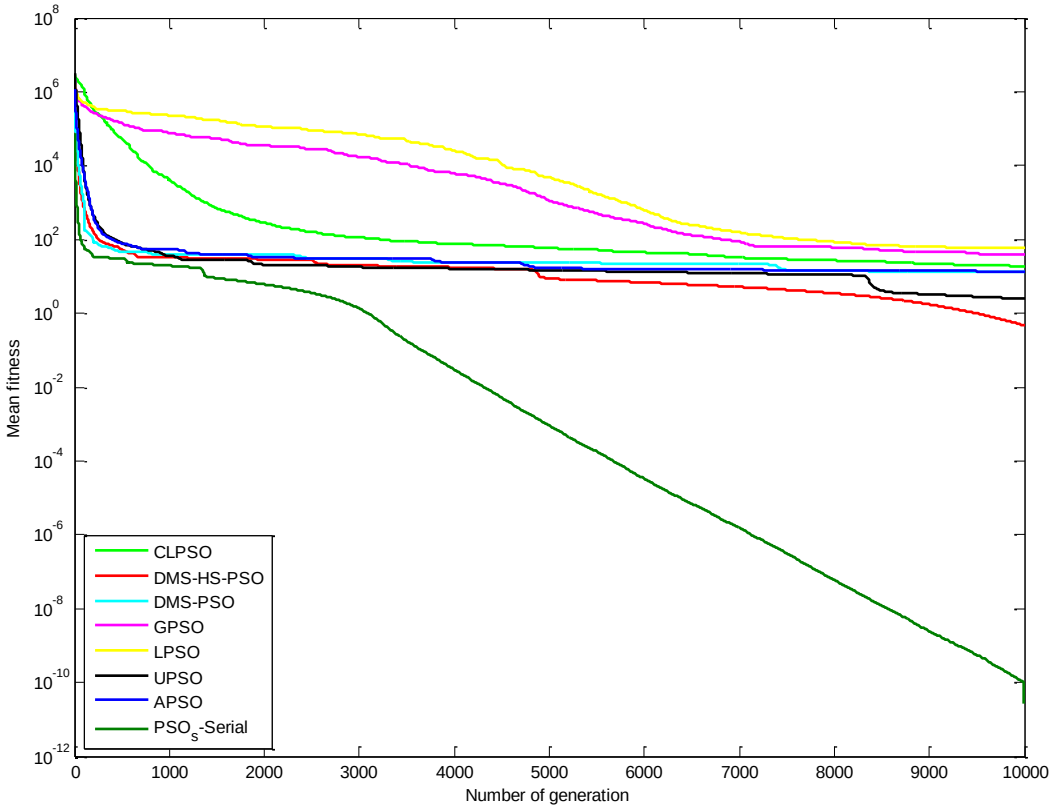

Figure 3.33: Comparison of new method with other algorithms responses on Rosenbrock's function.

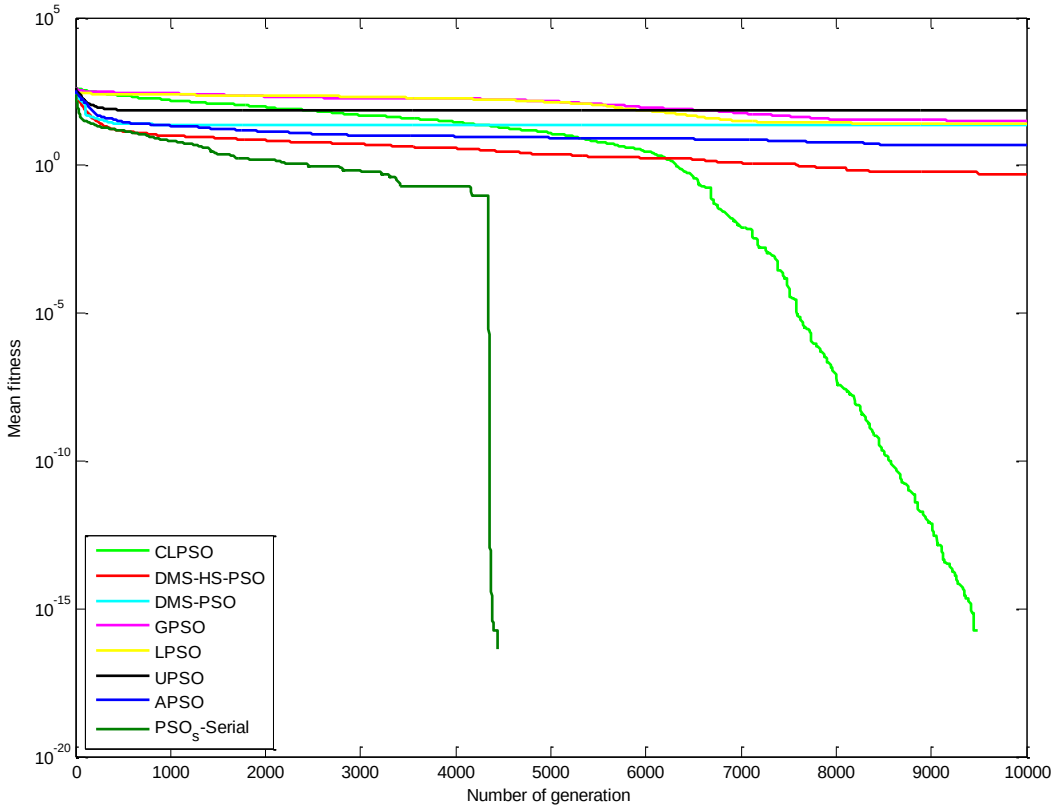

Figure 3.34: Comparison of new method with other algorithms responses on Rastrigin's function.

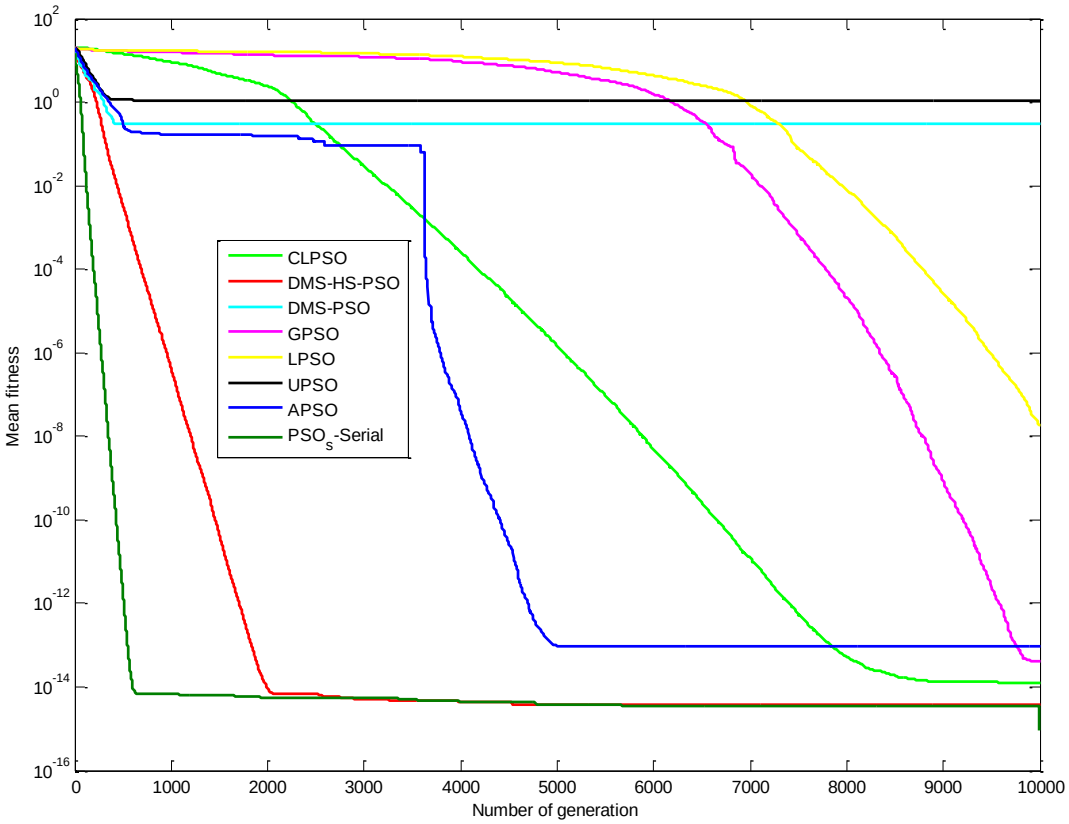

Figure 3.35: Comparison of new method with other algorithms responses on Ackley's function.

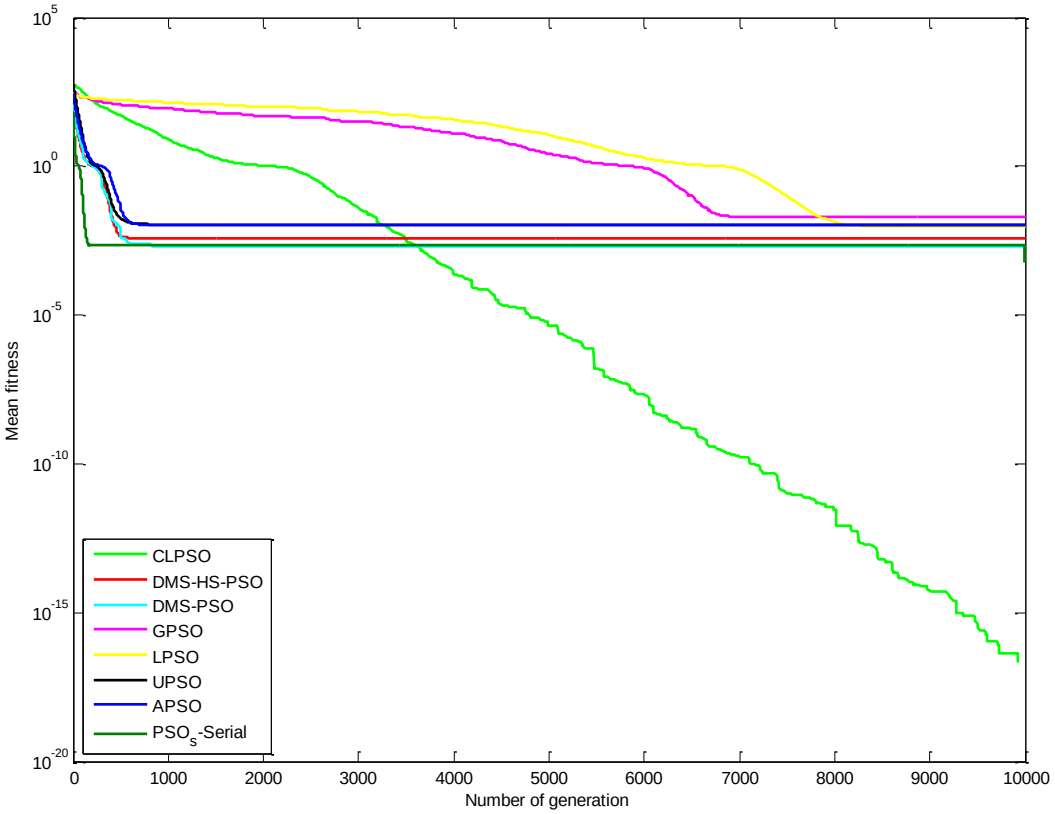

Figure 3.36: Comparison of new method with other algorithms responses on Griewnak's function.

3.7 Summary

The experimental results confirm that SPSO form of the proposed PSO achieves the best results compared to PPSO. All candidate types of PSO were compared separately with a different set of benchmarks with new method of SPSO, and PPSOs were compared by observing the response of each algorithm on all selected benchmark functions in terms of convergence speed and performance to reach the closest correct solution. SPSO and PPSO showed very good results in terms of accuracy and convergence in minimal iteration. However the SPSO exceeded all other PSOs, as is evident from all figures, and the results shown in Table 3.6.

Chapter 4: Power System Stabilisation

4.1 Introduction

Electrical energy is the fundamental form of energy that modern life is based upon, for domestic and industrial applications. There is a constant and increasing need for the generation and transmission of electricity, and it should be more economical, stable and reliable, to face the growing demand for energy upon an increasingly strained infrastructure. Furthermore, the power system generation equipment and customer equipment are usually designed to operate within a range of ±5% of the nominal voltage, therefore many types of equipment perform poorly at low voltage, ultimately expressed in the end forms of less illumination and overheating in electrical bulbs and induction motors (Kirby & Hirst, 1997).

This chapter presents power system stability by discussing the most important four dynamic phenomena affecting power system stability, namely wave, electromagnetic, electromechanical and thermodynamic phenomena. Furthermore the relationship between reactive power voltage and stability will be explored; this variable is one of the most important targets in the search, as actual variable that indicates the state of the electrical power grid in terms of stability. Additionally, this chapter, through mathematical analysis, discusses the most important types of power system stability devices to learn the working methods, as well as tuning methods. The most important faults types which affect the power system stability of the electrical grid are then described, allowing with how these problems can be emulated by software programs for the purpose of analysis. Finally, the latest types of conventional power system stabiliser (PSS4B) devices are explained and tested using different conditions and compared to other devices from prior generations.

4.2 Control System Stability

Power system stability is an important element in the event of any disturbance, such as the loss of part of the generation system, a fault on the transmission line or a change in the demand by losing or changing the load characterises, as shown in Figure 4.1. The power system dynamics can be divided into four different groups (Machowski, Bialek, & Bumby, 2011), based on their physical characteristics, which

are defined as wave, electromagnetic, electromechanical and thermodynamic. However, these broad classifications are affected by multiple dynamics groups.

As shown in Figure 4.1, wave effect is the fastest dynamic phenomenon, by which a surge in high voltage transmission lines occurs with a corresponding spreading of electromagnetic waves initiated during the switching operation or when lightning strikes, which takes a certain time (between microseconds to a millisecond). This disturbance triggers changes in the electromagnetic dynamics in the machine windings, as a result of the interaction between the electrical machines and the network or operation the protection system. This takes between milliseconds to a second. The electromechanical dynamics have slower effect than the following disturbances, namely changing in voltage, main control units of motors and operation of the protection system, which occur due to the oscillation in rotating masses in motors and generators.

The time frame of these dynamics is between a second to several seconds. The last dynamics are the thermodynamic changes, which take between seconds to hours. Such dynamics occur due to changes in temperature when a change in the boiler controls action in steam power plants during the connection of new load. To maintain the electrical system at stationary state in partnership with the generation, transmission and distribution and similar to the stability of any dynamic system in practical life depends on the disturbance type and the initial loading of the system.

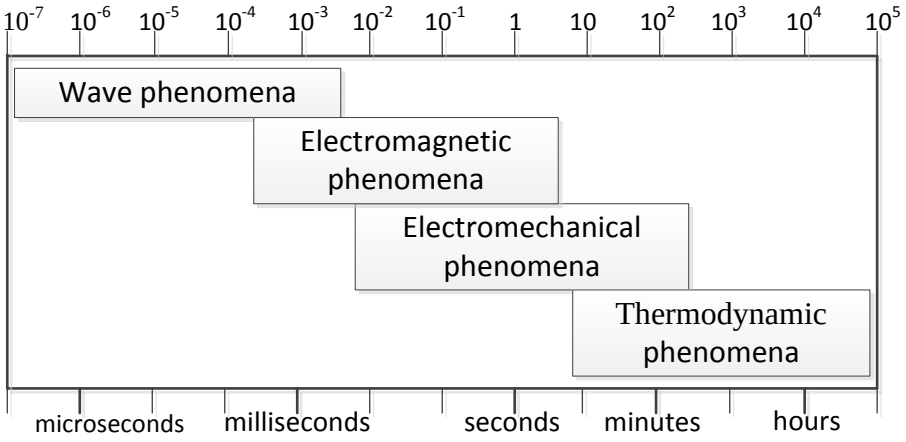

Figure 4.1: Time frame of the basic power system dynamic phenomena (Machowski et al., 2011).

4.3 Relation between Reactive Power Voltage Stability

The concept of the power system stability is defined as the systems' ability to get back to equilibrium state after being subjected to disturbance due to any physical fault. As shown in Figure 4.2, there are three very important quantities for power system stability:

- Voltage angles δ, or load angle.
- Network frequency f.
- Voltage magnitudes V.

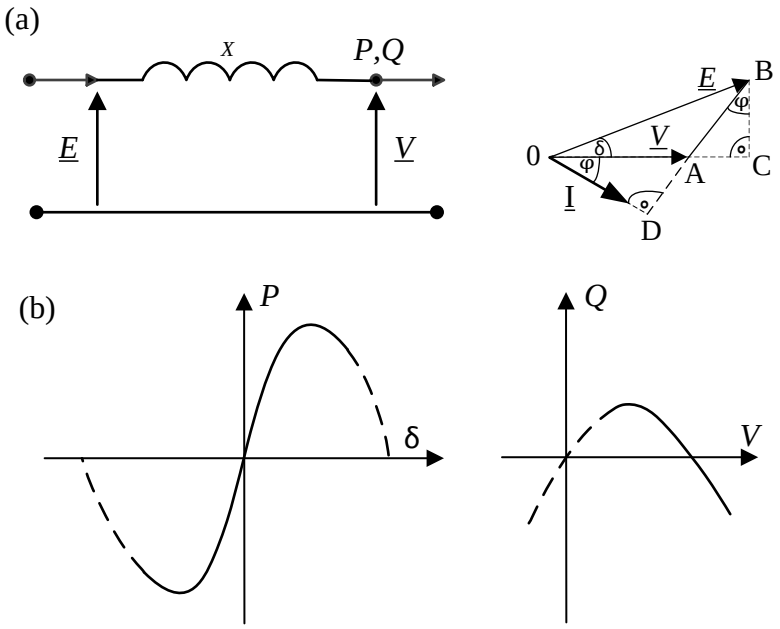

Figure 4.2: A simplified model of a network element. (a) equivalent diagram and phasor diagram; (b) real power and reactive power characteristics (Machowski et al., 2011).

From Figure 4.2, Q and P are for single-phase power while the voltages E and V are the phase voltages. Analysing the triangles OBC and BAC and the similarity of triangles BAC and OAD gives:

$$I\ Sin\ (\varphi) = \frac{E}{X} Cos(\delta) - \frac{V}{X} \tag{4.1}$$

$$I\ Cos\ (\varphi) = \frac{E}{X} Sin(\delta) \tag{4.2}$$

The real power leaving element X is usually expressed as:

$$P = VI\ Cos\ (\varphi) \tag{4.3}$$

Substituting the value of I Cos (φ) of equation 4.3 in equation 4.2 gives:

$$P = \frac{EV}{X} Sin(\delta) \tag{4.4}$$

Also the reactive power leaving X element can be expressed as:

$$Q = VI\ Sin(\delta) \tag{4.5}$$

Substituting the value of I Sin (δ) of question (4.5) into question (4.1) gives:

$$Q = \frac{EV}{X}\ Cos(\delta) - \frac{V^2}{X} \tag{4.6}$$

As the relationship between cosine and sine is $Cos\ (\delta) = \sqrt{1 - Sin(\delta)^2}$, equation 4.6 becomes:

$$Q = \sqrt{\left(\frac{EV}{X}\right)^2 - P^2} - \frac{V^2}{X} \tag{4.7}$$

From the above equations it can be seen that there are strong connections between the power system stability operations and the two pair of Real Power *P*, with Frequency *F* and Reactive Power *Q* with Voltage *V*, which are always being monitored in this thesis as an indicator of the final results of the power system stability (Machowski et al., 2011).

4.4 Power System Stability

From the point view of the definition and classifying power system stability, the voltage angles *δ*, or load angle, network frequency *f* and voltage magnitudes *V* are very important. Hence, the power system stability can be divided into (Figure 4.3):

- Voltage stability.
- Frequency stability.
- Rotor angle stability.

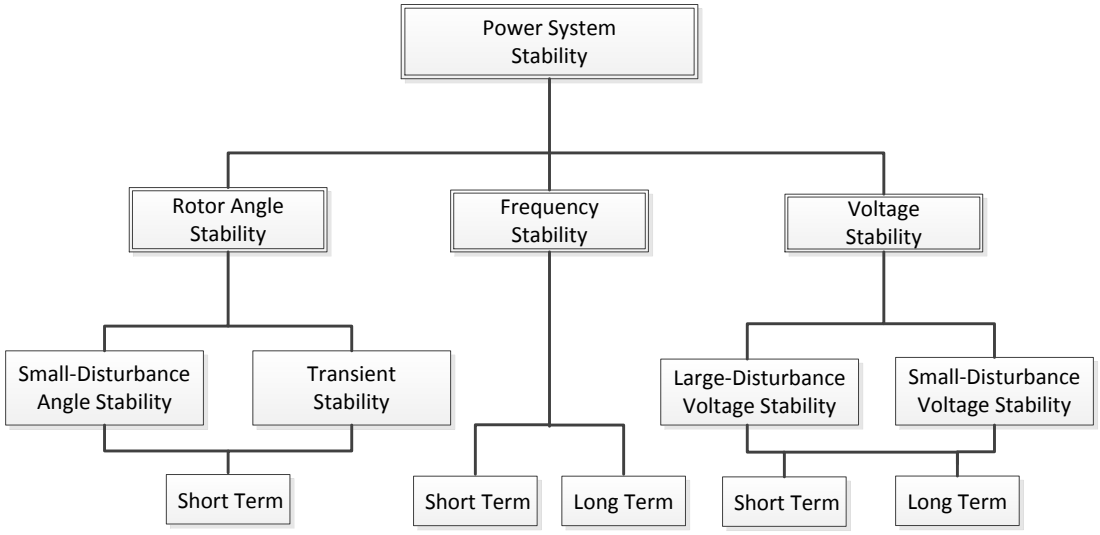

Figure 4.3: Classification of power system stability (Machowski et al., 2011).

The voltage stability is closely linked with other aspects of dynamic performances and power system steady state, for instance the voltage control, rotor angle stability, reactive power compensation and management, control centre and protective relaying all effect on the voltage stability. Therefore stability in power system is commonly referenced as the ability of generating units to maintain synchronous operation (Kundur, Balu, & Lauby, 1994; Saadat, 2002).

4.5 Conventional Control System

There are different forms of control systems, mainly divided into modern and conventional, which can solve the problems associated with signal disturbances in power systems such as oscillations and voltage variation.

Figure 4.4 shows a block diagram of the PSS controller which was used for the compensation. It is usually tuned via pole-placement method. Such controllers are designed to have fixed gain constant. In this situation it can be designed and tested using root locus technique. Then the gains separately designed to select dominant modes only. In a more efficient manner the pole-placement design was proposed in which participation factor were used to determine the size and number of stabilizers in a multi machine system to design of Power System Stabiliser (PSS) (Klein, Rogers, & Kundur, 1991; Pivonka, Veleba, Seda, Osmera, & Matousek, 2009) . All

of that is to enhance the damping of the local noises and inter-area modes of the system under consideration.

$$G_{PSS}(S) = [K]\left[\frac{s+Z}{s+P}\right] \tag{4.8}$$

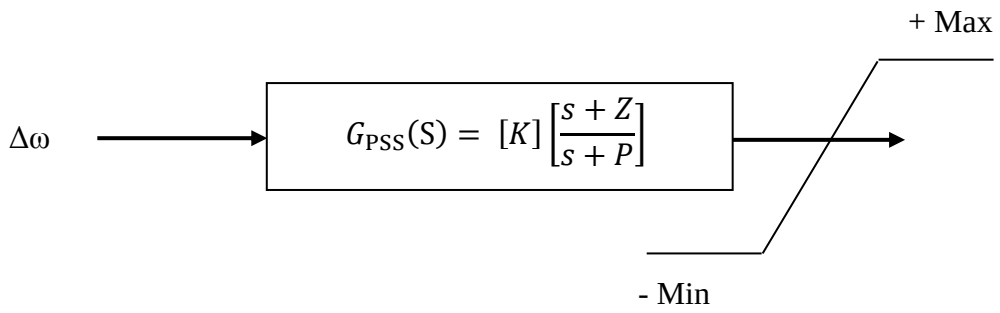

Figure 4.4: CPSS diagram.

As in the close loop system, the poles can be placed in anywhere in the complex plane, using the pole placement technique, therefore the system will always become stable even when the original system was unstable (Li, Zhao, & Yu, 1989). Shifting the system complex eigenvalue (λ) to new location λ_0 in the s-plane to a chosen location must satisfy the characteristic equation of the closed-loop system. It should also meet the specified damping ratio:

$$H(\lambda_0) = \frac{-1}{G(\lambda_0)} \tag{4.9}$$

In terms of phase and magnitude, the equation can be written as follows:

$$|H(\lambda_0)| = \frac{-1}{|G(\lambda_0)|} \tag{4.10}$$

$$arg\big(H(\lambda_0)\big) = 180^0 - arg\big(G(\lambda_0)\big) \tag{4.11}$$

where $\arg(G(\lambda_0))$ denotes phase angle of the residue λ_0.

The phase and magnitude of the compensator at the new pole λ_0 can be calculated using equations 4.9 and 4.10. Whereas $G(\lambda_0)$ is the complex frequency response of the system after the new pole location λ_0. The compensator generally includes a

washout, lead/lag transfer function and constant gain, which can be written as in equation 4.12.

$$G_c(s) = K\frac{sT_w}{1+sT_w}\frac{1+sT_1}{1+sT_2}\ldots\frac{1+sT_{2n-1}}{1+sT_{2n}}, \tag{4.12}$$

The main objective of series compensator is to improve the damping ratio of the selected oscillation mode. Therefore, $\Delta\lambda_0$ must be a real negative value to move the real part of the eigenvalue. Also the phase of each compensation block must be limited to a maximum of 60° for practical reasons. After the selection of the feedback signal, in order to control the direction of the eigenvalue displacement, the lead-lag stabiliser parameters and the maximum angle $\theta_{m,}$ and the i^{th} block can be provided and determined using the following equations:

$$\theta_m = Sin^{-1}\left(\frac{1-\alpha_i}{1+\alpha_i}\right) \tag{4.13}$$

where

$$\alpha_i = \frac{T_{2i}}{T_{2i-1}} \tag{4.14}$$

This maximum angle occurs at a frequency, which is given by [4.8]:

$$\omega_m = \frac{1}{T_{2i-1}\sqrt{\alpha_i}} \tag{4.15}$$

As a result, the conventional control theories depend on the basis of root locus (i.e. phase and gain margins), which advised a control devices such as PID and lead/lag, which were used in turn at the few past decades and which still have the leading hand in most control devices for power system stability control (i.e. multi-band stabilizer).

The gain margin and phase (Demello & Concordia, 1969) are used to design controllers which safeguard the stability in a certain mode with a pre-definite gain margins and phases. They use the frequency response of the open loop system to assist in the relative stability of the closed loop system. Therefore, the compensator function provides phase margins and sufficient gain to a particular mode, which will

provide the required specifications, but it should not have unacceptable opposing effects in other ways.

Characteristics lead/lag is generally used to form the required margin. It should be noted that after the first version of design, the effect of gain and phase margin on other modes should be monitored, and any problems can be solved (e.g. by trial and error). Furthermore, the lead/lag is more suitable when two or more modes are to be governed with same compensator. As explained in (Demello & Concordia, 1969), small disturbance of a synchronous machine connected to an infinite bus can be eliminated by a control system, but this entails a compromise between the damping torque and the synchronization torque. For thyristor-type, a high gain regulator through an excitation control system has a very important effect, depending on the type of thyristor, related to the significant elimination effect on the negative component of synchronizing torque; at the same time it will have an adverse effect on the damping torque.

A long investigation by (Larsen & Swann, 1981) on frequency response to analyse a conformation using a special stabilizing signal was derived from rotor speed and it has a significant amount of damping on the rotor angle. Using a different input signal, such as a combination of accelerating electrical power frequency signal and AC bus frequency signal as the system feedback signal, provides better results. Another approach is to use the power as feedback signal which it requires a lag/lead characteristic integrator as a compensating device which can cause adverse effects in case of mechanical power variations. After comparing the types of feedback signal, it is apparent that a combination of the two signals is more effective that using either the shaft speed of the AC bus frequency. This is more evidence for the efficacy of individual machines but with low sensitivity to oscillations.

In most conventional stability controller system designs, frequency response methods are used such as root locus and Ruth stability, whereas modern control techniques are based on improving the electromechanical oscillation damping using the eigenvalues Evaluation, which leads to the difficulty of quantitative determination of damping from the swing curves supplied by simulation when examining the system response. The contribution of eigenvalue technique is clearly defined on all modes, which can be easily identified in light modes and which is important with various oscillation

modes. Many algorithms were developed to find the critical eigenvalues explicitly for the small signal stability analysis (Oliveira, Ramos, & Bretas, 2008).

Large-scale systems require special treatment solution to save time. Reducing the model order of synchronous single machine connected to an infinite bus has been developed by (Altalib & Krause, 1976) through creating a model that shows the eigenvalues, which predicts the natural vibrations and introduces the rotor oscillations, which were accurately conserved. This is complemented by techniques of modal analysis scale systems to reduce the model size by dividing it into many areas the entire system (Price, Hargrave, Hurysz, Chow, & Hirsch, 1998), whereby each zone is represented by a generator. The results are the same as the complete system. All previous offers of employment with fixed gain PI or lead stabilizers phase shift gave an adequate response to the required nominal operating conditions, but when these points change, a very reasonable response is achieved, which is eliminating the need of modern control.

4.6 Power Generators and Power Grid

Electric power systems are mainly AC, in most cases with uniform frequency over the whole electrical power grid, as a result of using synchronous AC generators. The system voltages are held by generator excitation system control, and speed governors are used to control the generator main movers, to hold the system frequency within miniature acceptable limits in the electrical grids. The torque resulting from the interconnected AC generators depends on the relative angular displacement of each rotor, which acts to maintain the generators in synchronism, which is called synchronizing torques (Bakshi & Bakshi, 2009). Thus, if the angular displacement difference between generators increases, an electrical torque is produced that tries to reduce the angular displacement. The angular displacements should settle to values that sustain the required power flows through the transmission network and supply the system load. If the disturbance is large on the transmission system, the nonlinear nature of the synchronizing torque may not be able to return the generator angles to a steady state. Some or all generators then will lose synchronism and the system exhibits transient instability. On the other hand, if the disturbance is small, the synchronizing torques keeps the generators nominally in synchronism, but the generators' relative angles oscillate. In a correctly designed and operated system,

these oscillations should decay. In an overstressed system, small disturbances may result in oscillations that increase in amplitude exponentially.

Instability and oscillations in the electrical power system have occurred regularly around the world in the past of half century (Rogers, 1996). These appear after the power system is pressed to provide additional power. When the load is rapidly increased on the transmission lines by the consumer, the generators will rely more on their excitation systems to maintain synchronism, as shown in Figure 4.5.

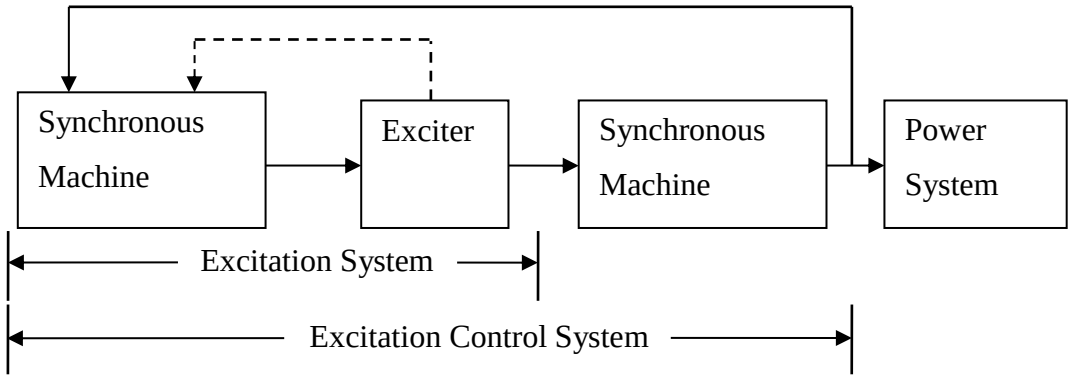

Figure 4.5: Block diagram of excitation control system.

Finally at some point the synchronization of oscillations becomes unstable. As a result, to minimise operation costs, many electric systems become interconnected, so the power can be exchanged between power systems. However, the interrelationships between neighbouring electrical power systems, even though they may not be over loaded, are often weak compared to the connection to other systems. The synchronizing torque is lower, coupled with aggregate inertia of each system being interconnected through these weak ties, which leads to inter-area oscillations at lower frequency regions. Many cases of oscillatory instability occurred at low frequencies, during which interconnections are made (Rogers, 1996).

A critical oscillation mode is called a local mode if it is strongly controllable in only one area and also strongly observable at this same area. Otherwise, it is called an inter-area mode. It should be noted that an inter-area mode can be strongly controllable and weakly observable in one area, but strongly observable and weakly controllable in another area; or strongly controllable in different areas (Feliachi & Yang, 1994).

A common problem of oscillatory instability is that the power flow can be large over the tie to supply remote load without any noticeable problems until the stability limit is achieved. A slight increase in the flow of energy outside the boundary of the results of the oscillation causes the amplitude to increase rapidly without any glitch in the system. The best case is when the system reduces the amplitudes of oscillation; in the worst case, the amplitude of the oscillation reaches the limit, whereby the lines trip and generation secured by the protective relays which causes total or partial system collapse (Feliachi & Yang, 1994).

4.7 Simulations and Implementation

Manual calculation for analysis of the stability of electrical power network is laborious and time-consuming, thus computing software programmes have been developed for this purpose, including but not limited to Matlab, SimPow, ETAP, ASPEN, BCP and IPSA. Matlab is the most widely used software due to its versatility and its strength for simulation. This is main reason behind using this software in this research to analyse and simulate different types of electrical network using the Simulink Power System Toolbox, which provides simulation of a simple electrical power grid, which includes two turbine-driven generators, one with 5000 MW capacity and the other with 1000 MW capacity. These two generators are mutually connected to the grid via high voltage transformers, bas bar and transmission line with a length of 700 km, together with a load of 5000 MW as well as fault circuit breaker, as shown in Figure 4.6. Multi-band (MB) controller stabiliser is simultaneously used as an advanced conventional PSS controller for damping any oscillation in the network (Inamdar & Hasabe, 2009). Subsequently, in this thesis a fuzzy-logic PSS controller (FPSS) is used to replace the conventional PSS (MB) stabiliser to perform the same task. Furthermore comparison for its activity will be made with a PSS MB stabiliser.

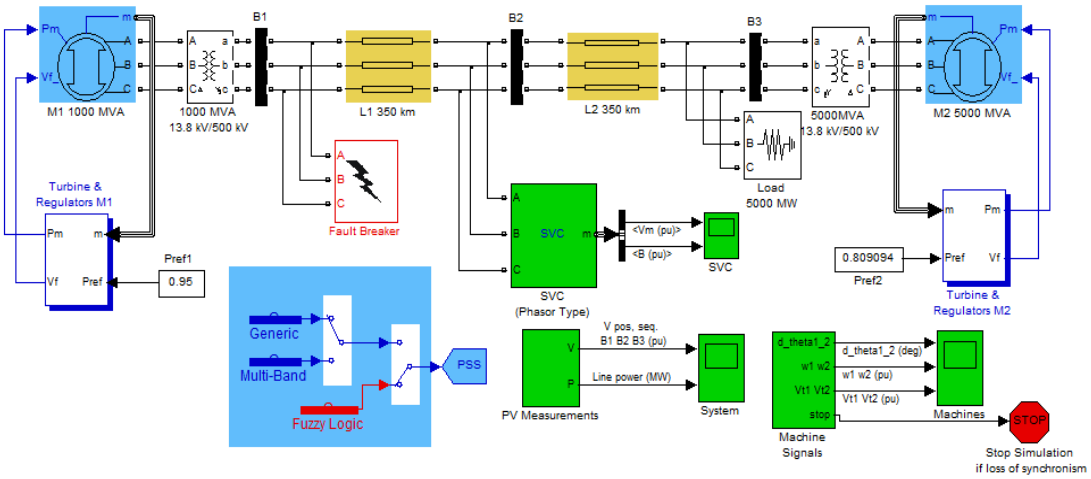

Figure 4.6: Block diagram of the simulated Matlab power system toolbox (Larsen & Swann, 1981).

The system contains two separate controllers working together to govern two synchronized generators driven by two turbines linked at the same electrical network. In order to observe the final outcome effect of both stabilizers controller devices, a single observation point is set after the Static Var Compensator (SVC) on the grid, which will eliminate the need to observe the output of each controller separately.

4.8 Static VAR Compensator

The Static Var Compensator (SVC) is a shunt device of the Flexible AC Transmission Systems (FACTS) family using power electronics to control power flow and improve transient stability on power grids (Song & Johns, 1999) and (Zhang, Rehtanz, & Pal, 2006). The SVC regulates voltage at its terminals by controlling the amount of reactive power injected into or absorbed from the power system. When the system voltage is low, the SVC generates reactive power (SVC capacitive). When the system voltage is high, it absorbs reactive power (SVC inductive). The variation of reactive power is performed by switching three-phase capacitor banks and inductor banks connected on the secondary side of a coupling transformer. Each capacitor bank is switched on and off by three thyristor switches (Thyristor Switched Capacitor, TSC). Reactors are either switched on-off (Thyristor Switched Reactor, TSR) or phase-controlled (Thyristor Controlled Reactor, TCR). Figure 4.7 shows a single-line diagram of a static VAR compensator.

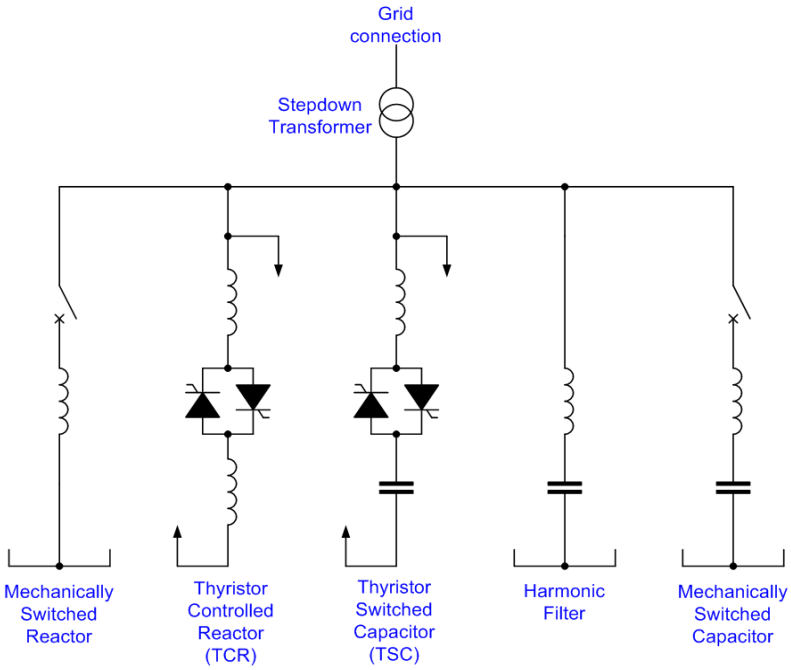

Figure 4.7: Single-line diagram of an SVC.

4.9 Multi Band Stabiliser

In 2005, the Institute of Electrical and Electronic Engineers (IEEE) introduced the newest type of conventional power system stabiliser model called multiband power system stabilizers (M.B.) PSS4B (Kamwa, Grondin, & Trudel, 2005). This gives a better performance than the regular power system stabilizers (PSSs). For this new type there are three levels of parallel control block, each of which aims to reduce the damping in different modes of oscillation and different ranges of frequency bands at the low frequency oscillation in the power system. It requires two input parameters, the high frequency and low with intermediate frequency (ΔωH, (ΔωL-I), similar to the IEEE PSS2B, which was integral to accelerating the power of PSS during the early 1990s as the first practical implementation of a digital PSS (Kamwa et al., 2005). The underlying principle of the new IEEE PSS4B makes it sharply different by choosing tuning method for three different bands, to cover a wider frequency range, designed to handle the high frequency oscillations in the lower band, while the low and intermediate frequency oscillations are handled by the two upper bands, as shown in Figure 4.8.

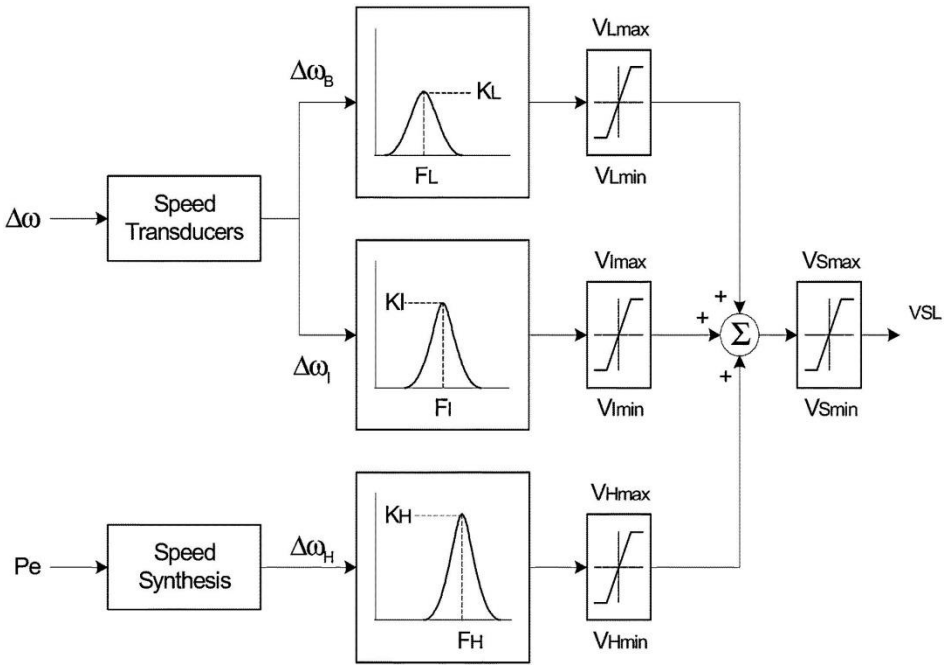

Figure 4.8: Conceptual block diagram of the multiband PSS (IEEE PSS4B).

4.10 Simulation Results

To study and assess the robustness of any new device, it has to be tested under different scenarios covering a wide range of normal work situations, including any disturbance that may occur during the smooth work. This mainly causes actual disturbances to the system, while it simultaneously needs to evaluate and compare the response of the device with different types of similar equipment doing the same thing. In this case, implementing the different PSSs in the test by simulating Matlab/Simulink software showed that the performance of the PSS4B is obviously respectable, resulting in increased damping and reduced oscillation in the electrical network, but this can be improved by the new design presented in this research.

The scenario which was chosen for this test, via closing the fault circuit breaker in three different ways after 0.8 sec of the start of the simulation, for a transition time of 0.1 sec. (i) Via short circuit one phase to the ground. (ii) Via short circuit of two phases to the ground. (iii) Via short circuit of three phases to the ground. Two types of stabilisers (generic stabiliser (PSS2B) and multi-band stabiliser (PSS4B)) were used to control each turbine as advanced conventional MB stabiliser to improve the performance (Feliachi & Yang, 1994), (Inamdar & Hasabe, 2009) and (Ali, Tayeb, & Adam,), as shown in Figures 4.9, 4.10 and 4.11.

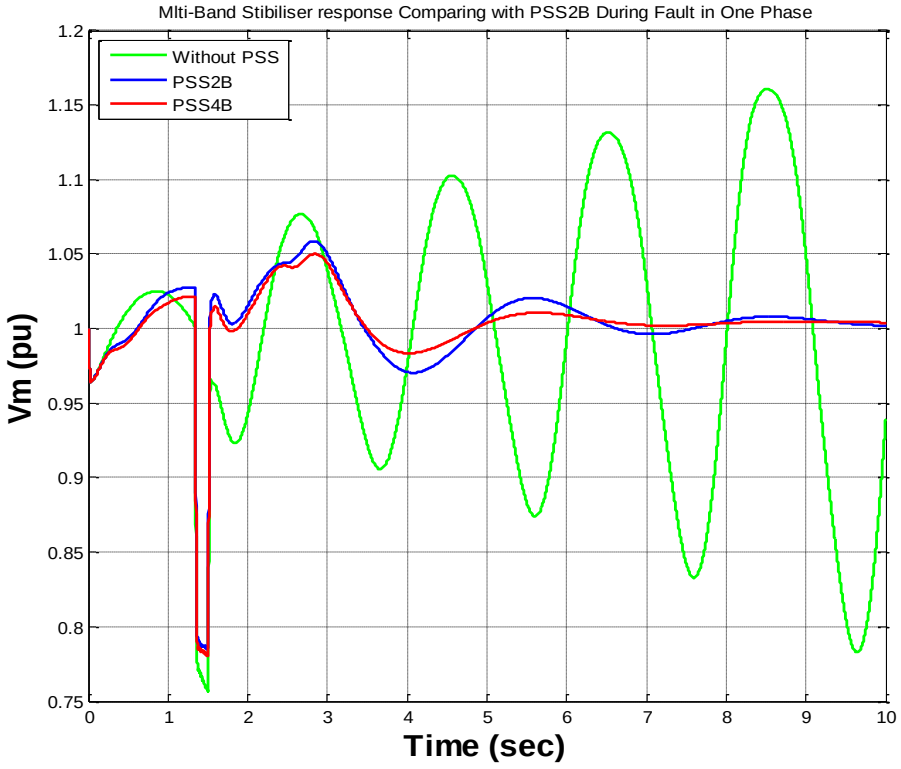

Figure 4.9: Multi-band stabiliser response comparing with PSS2B and without PSSs during fault in one phase.

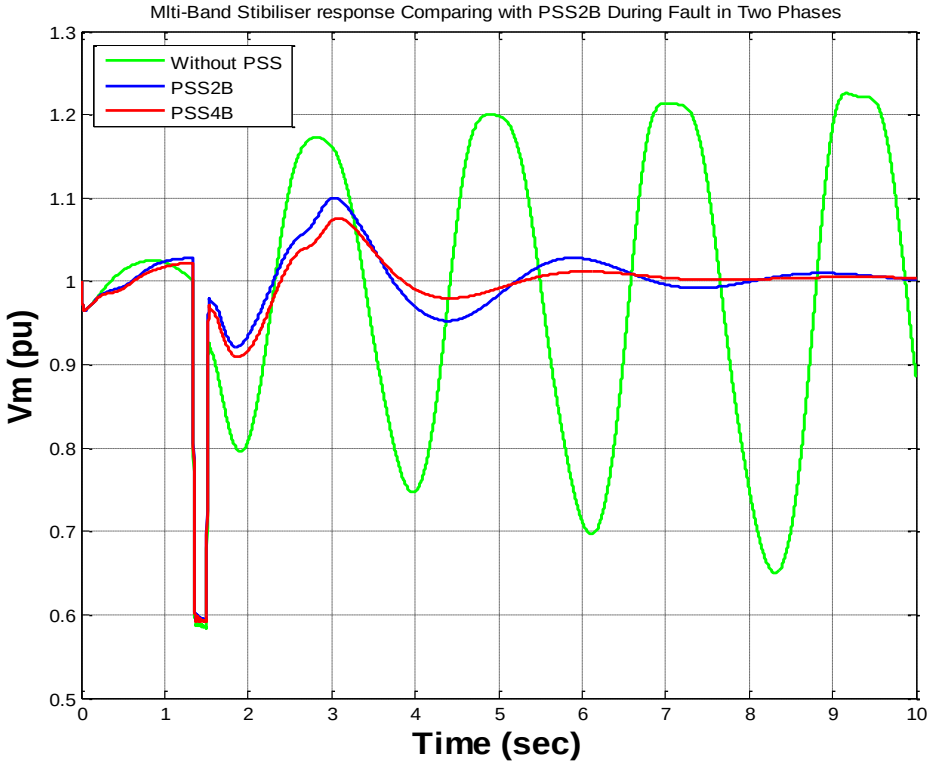

Figure 4.10: Multi-band stabiliser response comparing with PSS2B and without PSSs during fault in two phases.

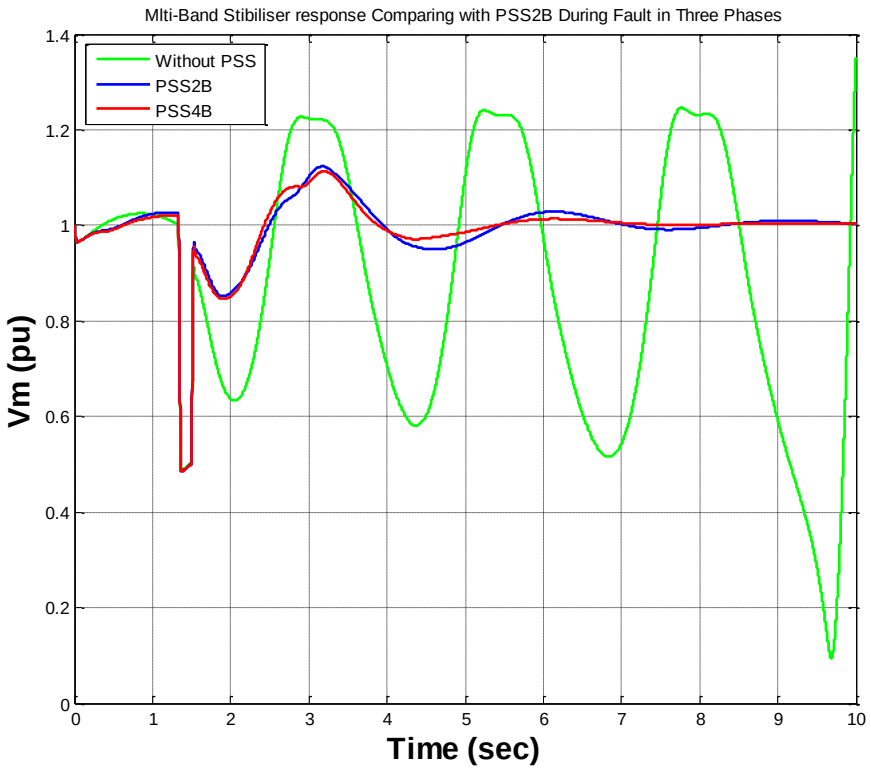

Figure 4.11: Multi-band stabiliser response comparing with PSS2B and without PSSs during fault in three phases.

4.11 Summary

Figures 4.9, 4.10 and 4.11 show the positive impact of the PSS devices on the electrical network systems, whereby the effect of the generic power system stabiliser PSS2B and multiband power system stabiliser PSS4B are convergent in terms of the first overshoot, but the PSS4B still works on damping the electrical network system more effectively than PSS2B, as well as doing its function with higher efficiency compared to the system stability if working without power system stabiliser PSSs at the three case scenarios, after malfunction in the system (i.e. making short circuits between one phase and ground, then between two phases and the ground, then between three phases and the ground, sequentially). As seen in the three figures, PSS4B works with high efficiency on over-shooting and damping of the system oscillation compared with PSS2B, which was working efficiently but slightly less than the PSS4B, and the system becomes completely unstable and oscillation grows after any type of faults when it works without power system stability devices.

Chapter 5: Intelligent Control and Optimisation of Power System Stabilisation

5.1 Introduction

The basis of life and nature in living organisms is to improve their situation; this is known as iterative Optimisation, and is as old as life itself. Many strategies can be conceivable, but only those that are effective are manifest in nature, propagating the continuation of species and offering a wide range of solutions. Mathematics itself arose as a framework with which to understand the world, to 'apprehend the Pythagorean power by which number holds sway above the flux', and many mathematical models to improve decisions explicitly or implicitly use biological behaviours as a starting point, drawing on models originating in genetics, ethics and even ethnology or psychology (Russell, 2009)and (Clerc, 2010).

Particle swarm Optimisation (PSO) adopts the iterative method approach, with an emphasis on cooperation; partly it is random and without selection. This chapter details how this technique is used to design and solve the problem in power system stability. Its steps can be summarized in terms of designing a control device that can be tuned by SPSO, in several different stages, to control the system in any operating conditions (including unstable) and in different scenarios, to coordinate responses to extreme failures in the operating system, where one phase fault can cause the failure of three phase faults simultaneously.

This chapter describes the current state of the intelligent Control and Optimisation of Power System Stabilisation (COPSS). It is organized into nine sections, including this introductory section, which is followed by a brief summary of the fuzzy logic controller then an explanation of the design and tune stable control system using fuzzy logic controller before the specific approach in neuro-fuzzy logic systems. The implementation of ANFIS-PSS controller and training them in different stages is then explained, involving (i) the single phase training, and (ii) the three phase training. The new ANFIS-PSS controller responses to ground fault in tie line in machines A and B is then explained and the power quality in the network is outlined. Section seven explains the auto tuning of scaling factors using intelligent Optimisation, and

section eight presents the simulation results of rotor speed deviation on both machines A and B, in addition to compare the power quality in the network. Section nine presents a summary of the chapter.

5.2 Fuzzy Logic Control

The fuzzy logic controller recently has become a favourable solution receiving a great deal of attention in various applications, most notably in the field of electronic energy (Mohammadpour, Mirhoseini, & Shoulaie, 2009), (Ajami, Taheri, & Younesi, 2009). The trait of control devices that operate on the principle of fuzzy logic are more competitive in terms of the low price, ease of control and tuning and robustness; additionally, the mathematical scheme model of fuzzy logic controller does not need to explain the system under study. However, in fuzzy control, to select the parameters related with the membership functions and the rules depends extensively on the intuition of the engineer, which is the main problem with this solution. Moreover, fuzzy systems are essentially approximate systems to produce a general solution to the adjusting problem. In the case of a control problem including dynamic nonlinear systems, the fuzzy logic control and artificial neural network work together as two methodologies called neuro-fuzzy systems. Both methodologies are powerful design methods with their own strengths and weaknesses, but together they can help to decrease the design time and manage the complexity (Ansarian, Sadeghzadeh, Nasrabadi, & Shakouri, 2005) , (Rojas, Bernier, Rodriguez-Alvarez, & Prieto, 2000) . Therefore, the adaptive neuro fuzzy inference system (ANFIS) controllers are utilized due to the power system stabilizer, which acts as a damping controller to control and improve the power system stability of the electrical power networks.

In this chapter a new power system stabiliser (PSS) is implemented to alleviate the oscillation and improve the power system stability by designing fuzzy logic controller (FLC) strategy, which replaces the conventional power system stabilizer. To reach this design goal necessitates several stages, as described below with the obtained results.

5.3 FPSS Controllers

Fuzzy logic controller (FLC) devices vary in terms of choice of different parameters in the rule base and membership functions. In this research the design was executed via Matlab/Simulink program (J. Jang & Sun, 1995).

5.3.1 Design

Usually, the design and tuning of a stable control system for small and known process transfer functions is done by knowledge of basic characteristics of the transient response of a closed-loop system, which is closely related to the location of the closed-loop poles. A simple graphical method for finding the roots of the characteristic equation was developed by (Evans, 1950), and used extensively in control engineering. This method, called the root-locus method, plots the roots of the characteristic equation for all values of system parameters then locating them on the graph. If the system has a variable loop gain, then the location of the closed-loop poles depends on the value of the loop gain chosen. Selecting the right gain is the main task in this part of search, enabling the designer to know how the closed-loop poles move in the S-plane as the loop gain is varied. From the design viewpoint, in some systems simple gain adjustment may move the closed-loop poles to desired locations, then the design problem may become the selection of an appropriate gain value. If the gain adjustment alone does not yield a desired result, the addition of a compensator to the system becomes necessary. Also in our case the problems are more complicated and laborious in terms of selecting the appropriate gain value, as shown in Figure 5.1 and Figure 5.2, for FLC_1 Ke1, Kc1, Ko1 and Ke2, Kc2, Ko2 for FLC_2 respectively as use one PSS controller on each generator.

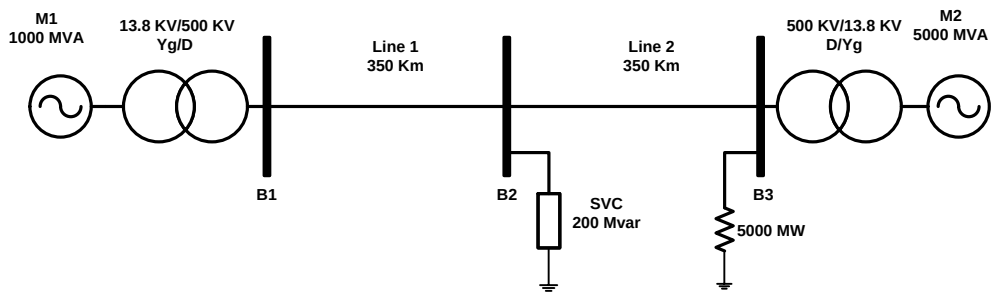

Figure 5.1: Two generators connected together in a small network.

In the design of fuzzy logic controller, the first stage is to choose the correct input signal. In this research the generator speed deviation (Δω), and its derivative (Δώ) are two signals considered as two inputs for the controller, which work as (FPSS) controller, as shown in shown in Figure 5.2.

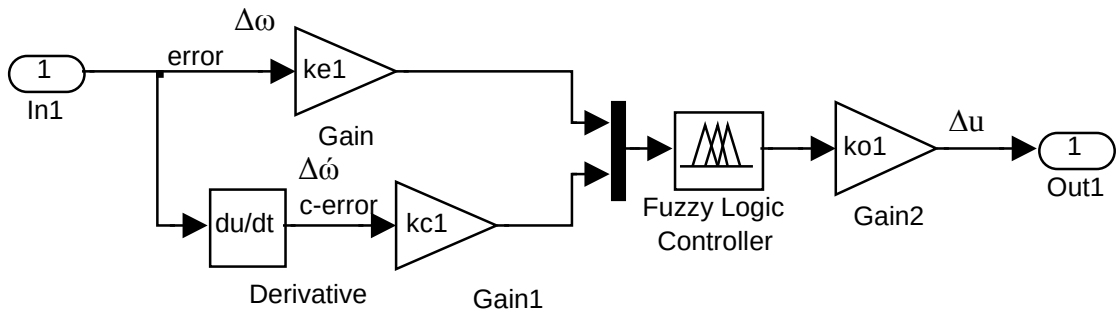

Figure 5.2: Generalized FLC auxiliary fuzzy controller and Structure of FLC.

These two signals are used as rule-antecedent (IF-part) in the formation of rule base, and the output of controller is used to represent the contents of the rule consequent (THEN-part) in performing of rule base (Bevrani & Daneshmand, 2012), which is injected into the input of the excitation circuit controller.

In this part, the inputs and the output singles are normalized for the base values defined for the system. The fuzzy value of controller can be explained by the frame and number of the membership functions (for the inputs and output) described off-line. The S-shaped membership function (Smf) and Z-shaped membership function (Zmf) are employed for the inputs and output fuzzy sets of the FLC. The designed membership functions for: Δω, Δώ as inputs and Δu as output are shown in Figure 5.3. Heuristically selected fuzzy rules obtain the control rules of the fuzzy controller as described below:

The rules base with two suggested input for the fuzzy set membership are determined as: N: Negative, Z: Zero, P: Positive, respectively.

1. If (Δω is N) and (Δώ is N) then (Δu is N)
2. If (Δω is N) and (Δώ is P) then (Δu is Z)
3. If (Δω is P) and (Δώ is N) then (Δu is Z)
4. If (Δω is P) and (Δώ is P) then (Δu is P)

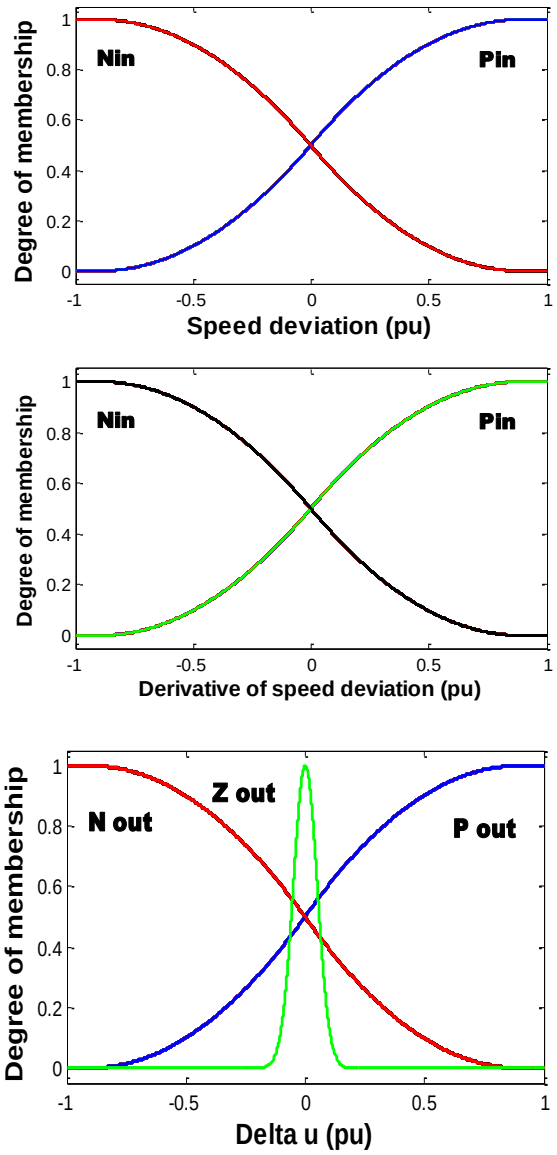

Figure 5.3: Membership functions for inputs and output fuzzy sets of the FLC.

The construction of the FLC control surface is displayed in Figure 5.4. The basic steps followed for designing the FLC controller in Matlab / Simulink are outlined in (SimPowerSystems 4.3 user's guide.):

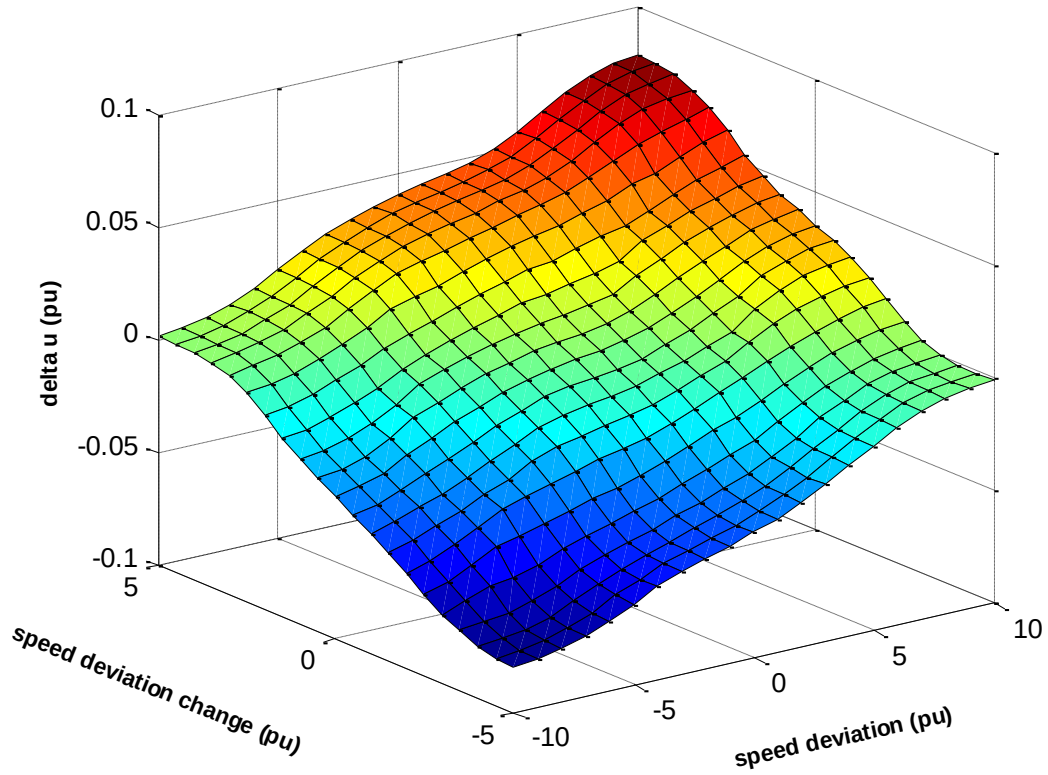

Figure 5.4: Control surface of ANFIS-based FPSS controller.

5.3.2 Determining the Scaling Factors

In the proposed fuzzy power system stabilizer, FPSS, has two inputs and a single output, as shown in Figure 5.5; the two inputs gain Ke, Kc and one output Ko, for the error (e), change of error (ce) and output respectively, for each controller.

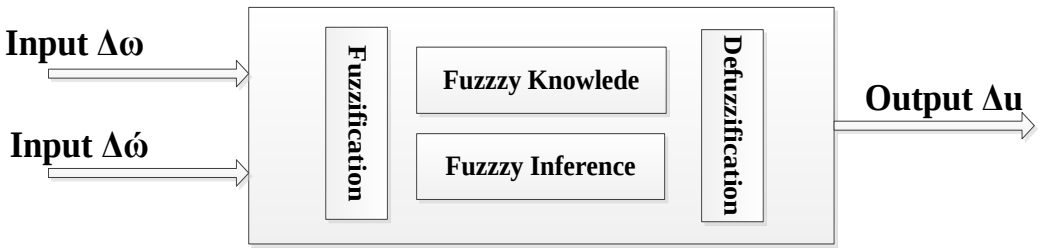

Figure 5.5: Block diagram for FLC controller.

Normalization performs a scale transformation that is also called input normalization. It maps the physical values of the current process state variables into a normalized universe of discourse. It also maps the normalized value of control output variables

into its physical domain (output de-normalization). For this controller, normalization is obtained by dividing each crisp input on the upper boundary value for the associated universe (Subbaraj & Manickavasagam, 2008).

5.3.3 Membership Function Definition

The vagueness of the decision maker in selecting an adequate plan in fuzzy logic programming is reflected as the degree of satisfaction through the shapes of membership functions. Depending on the strictness of objective attainment, it is assumed that the three kinds of membership functions are as shown in Figure 5.6. (Baer *et al.*, 2000). Hence, to convert the measured input variables of the FPSS into suitable linguistic variables, the triangle shapes with three fuzzy subset functions is one of various types of membership functions are given below:

- Triangular Membership Function
- Gaussian Membership Function.
- Trapezoidal Membership Function.
- Sigmoidal Membership Function.
- Generalized bell Membership Function.

The choice of membership functions for each linguistic variable must be performed. This is can be done by classification of control inputs and output into classed fuzzy sets.

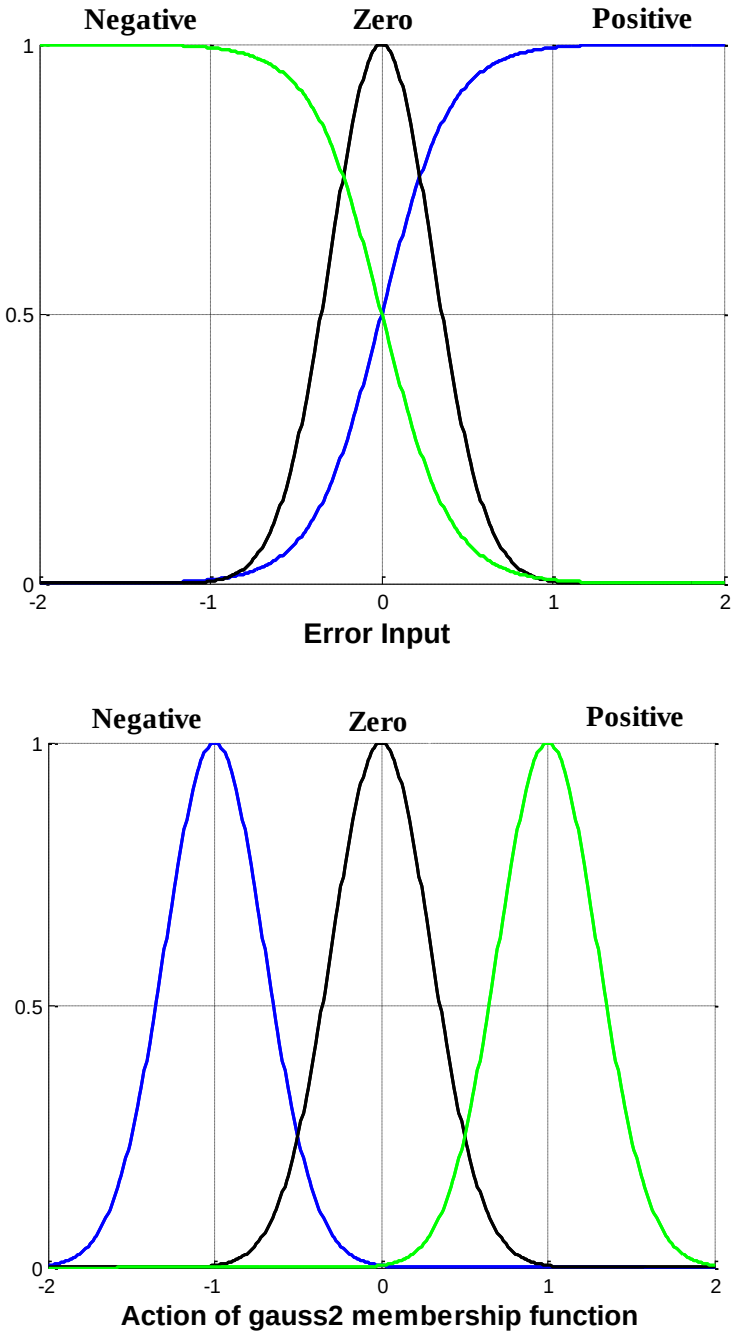

Figure 5.6: Gaussian membership functions.

In this case the generator speed deviation is classified into: [positive (w_p); zero (w_z); negative (w_n)]. The generator speed deviation change is classified into: {positive (dw_p); zero (dw_z); negative (dw_n)}. Also the output of fuzzy controller is classified into: {positive (u_p); zero (u_z); negative (u_n)}.

As shown in Figure 5.7, the linguistic speed deviation to generate the rules for a system consisting of two input variables with three linguistic variables in each range leads to a 3 × 3 decision table with 9 rules, as shown in Table 5.1. Each entity in the

table represents a rule which are as follows: positive (P); big positive (BP); zero (Z); negative (N); big negative (BN).

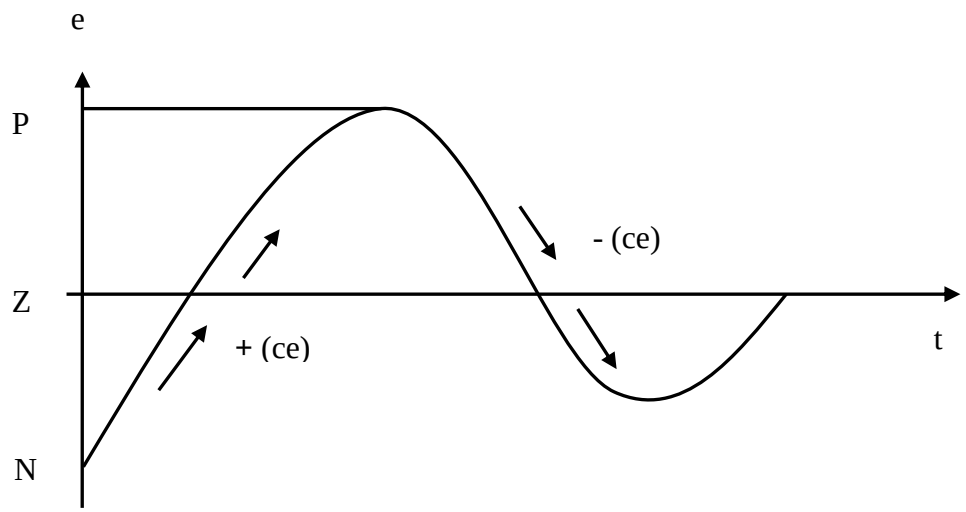

Figure 5.7: The linguistic of speed deviation.

Table 5.1: Rule table for FPSS.

Speed Deviation (e)	Speed Deviation Change (ce)		
	dw_n	dw_z	dw_ p
w_n	BP	P	Z
w_z	P	Z	N
w_p	Z	N	BN

5.3.4 Manual Tuning of the Scaling Factors

In this stage, to improve the FLCs response, the FLCs scaling factors are tuned manually, which is dubbed for the first generator controller as Ke1 for the error gain, Kc1 gain for the change of error and Ko1 is the output gain. For the second generator controller, Ke2 is the error gain, Kc2 is the gain for change of error and Ko2 is the output gain. The best value established for Ke1 = 2, Kc1 = 3.75, Ko1 = 2.25 and Ke2 = 5, Kc2 = 3.75, Ko2 = 10, as shown in Table 5.2.

Table 5.2: Manual tuning of the FLC scaling factors.

	FLC1			FLC2		
	Ke1	Kc1	Ko1	Ke2	Kc2	Ko2
Manual Tuning	2	3.75	2.25	5	3.75	10

5.3.5 Simulation Results

This section presents some results as a sample of tuning FLC controller, and compares them with M B stabilizer. Figure 5.8 to Figure 5.10 illustrate the grid power (Vm) in pu as a response of the whole system after Static Var Compensator (SVC) during fault in one, two and three phase correspondingly without PSS stabilizer, with conventional power system stabiliser (CPSS) called multi band (MB) stabiliser and with FLC stabiliser tuned manually. The response of FLC controllers is to dampen the output of all systems after a few repetitions, thus it is better to run without PSS stabiliser case, but still lower than conventional stabiliser MB PSS, because low range of membership can be selected for three kinds of membership functions, as shown in Figure 5.6. This means a small range of membership functions is available for highly complex systems, resulting in low sensitive action on the final response from this controller. However, the reason for execution this stage to become more familiar with FLC designs. The main target is to focus on the next stage which is dealing with training of FLC using Adaptive Neuro-Fuzzy Inference System (ANFIS), and auto tuning of the scaling factors using Particle Swarm Optimisation (PSO), for both controllers (FLC1 & FLC2) in the process at the same time.

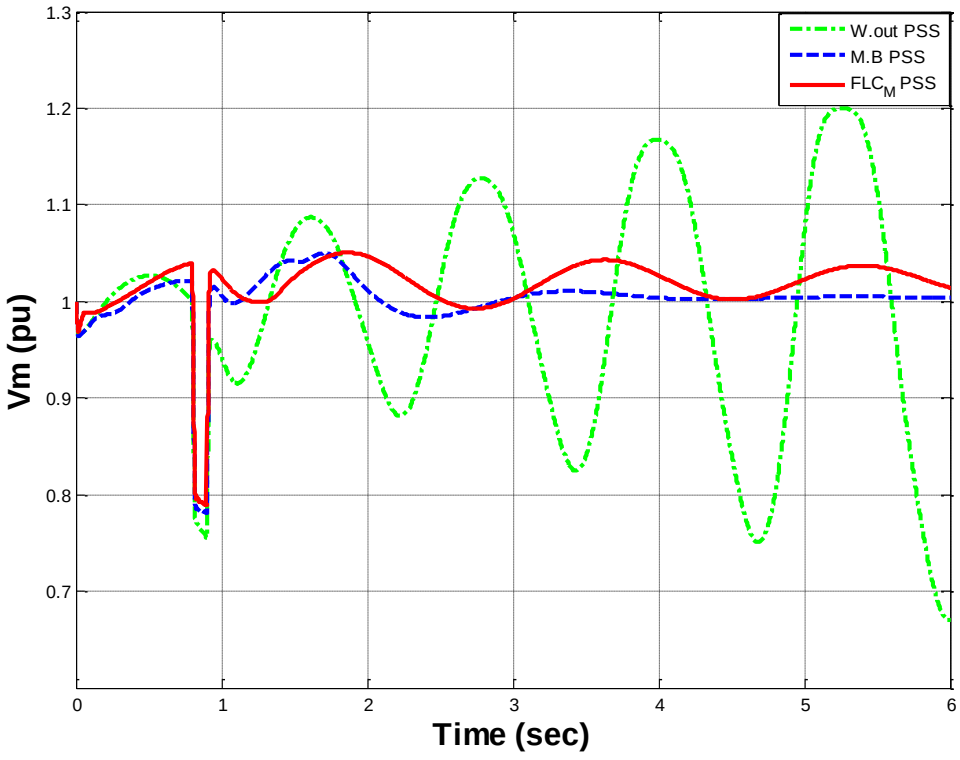

Figure 5.8: System behaviours during one phase fault Simulation results.

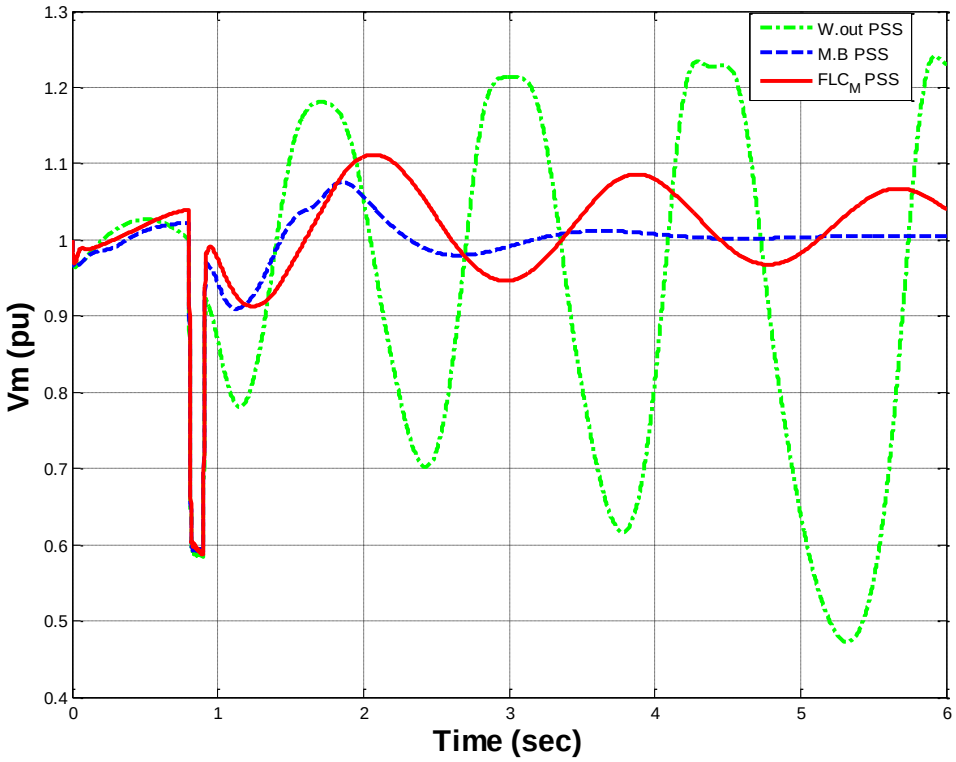

Figure 5.9: System behaviours during two phase fault simulation results.

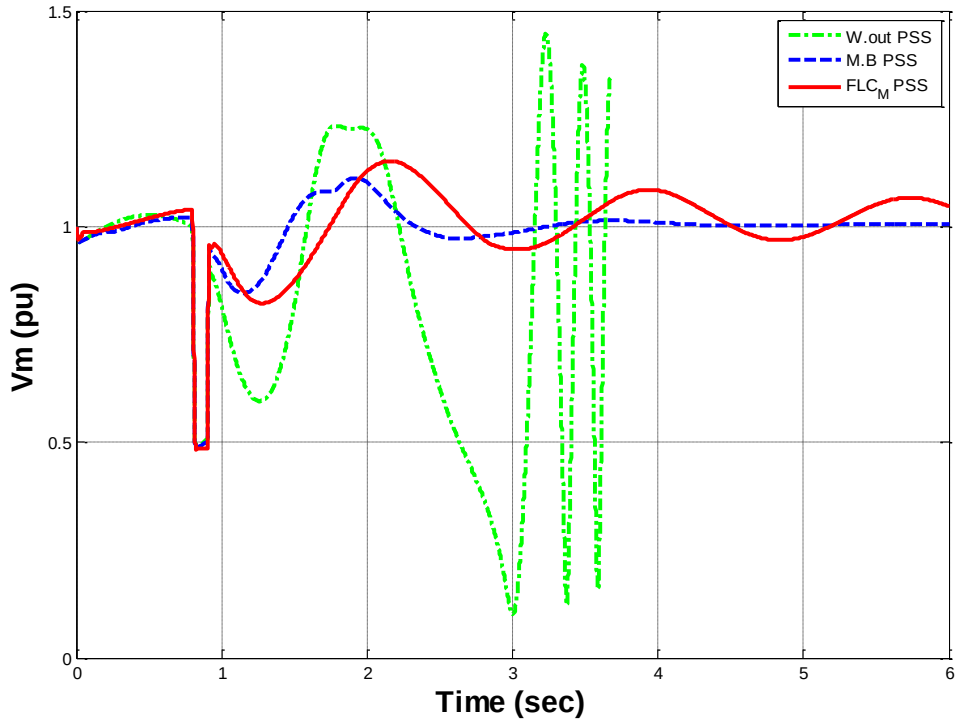

Figure 5.10: System behaviours during three phase fault simulation results.

5.4 Adaptive Neuro Fuzzy Inference System

ANFIS gives significant results in nonlinear function modelling type (J. R. Jang, Sun, & Mizutani, 1997). The ANFIS is new technique for determining the behaviour of imprecisely defined complex dynamical systems, based on improved tool and a data driven modelling approach. On the other hand, it has human-like expertise, to adapt itself and learn from others, to do its functions with high efficiency in different environments (Kurian, George, Bhat, & Aithal, 2006b). From a given input/output data set ANFIS can generate unknown fuzzy rules. A typical ANFIS architecture is shown in Figure 5.11.

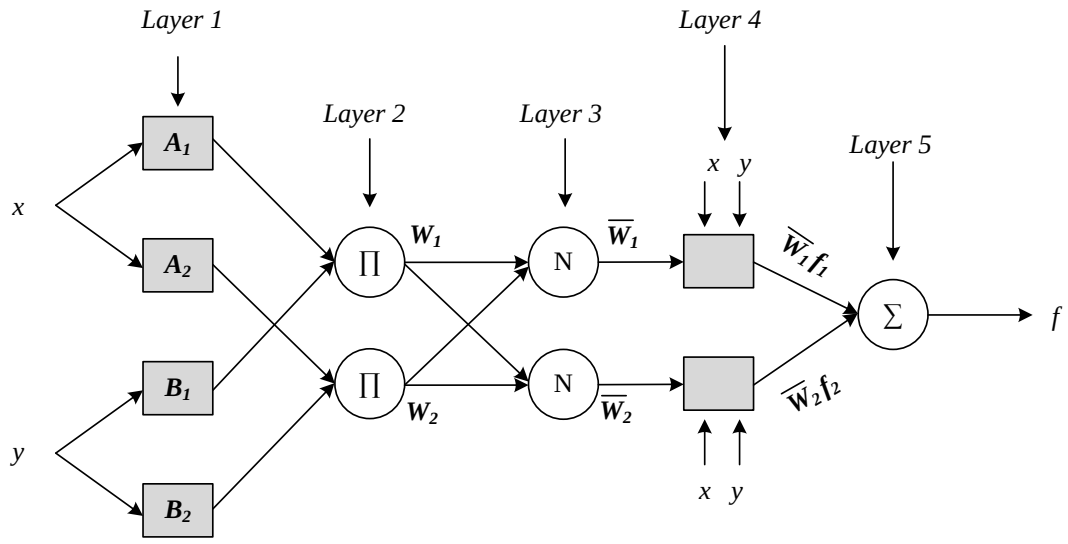

Figure 5.11: A typical ANFIS architecture (Schwefel, 1981).

Figure 5.11, shows two inputs *x* and *y* while *z* is the final output. The linguistic labelling of the system, represented by A_1, A_2, B_1 and B_2, are associated with node function. (*Wi*) is the normalized firing strength to signify the ratio of the i^{th} rule's firing strength (*Wi*) to the sum of the first and second rules' firing strengths ($\overline{W1}$ and $\overline{W2}$). The given concept of ANFIS construction can be described using a simple pattern whose rule base is given below:

Rule 1: If x is A_1 and y is B_1 then

$$f_1 = p_1 x + q_1 y + r_1 \tag{5.1}$$

Rule 2: If x is A2 and y is B2 then

$$f_2 = p_2 x + q_2 y + r_2 \tag{5.2}$$

Each node of the layer 1 is a node with a function adaptation node, which may be a Gaussian membership function or any other membership functions.

$$Q_i^1 = \mu_{Ai}(x), \qquad for\ i = 1,2 \tag{5.3}$$

$$Q_i^2 = \mu_{Bi}(y), \qquad for\ i = 1,2 \tag{5.4}$$

where x and y are the input to node i, and A_i and B_i are the linguistic labels (small, large, etc.) associated with this node function. In other words, Q_i^1 is the membership function of A_i or B_i and it specifies the degree to which the given x and y satisfies the quantifier A_i and B_i. Usually we choose μ_{Ai} and μ_{Bi} to be bell-shaped with maximum equal to 1 and minimum equal to 0, such as the generalized bell function (J. Jang, 1993b), (Takagi & Sugeno, 1983).

$$\mu_{Ai}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i}\right)^2\right]^{b_i}} \tag{5.5}$$

or Gaussian

$$\mu_{Ai}(x) = exp\left[-\left(\frac{x - c_i}{a_i}\right)^2\right] \tag{5.6}$$

where premise parameters (a_i, c_i, b_i) are the parameter set.

Each node of layer 2 is fixed and labelled as Π, which demonstrates the firing strength of each rule:

$$Q_i^3 = W_i = \mu_{Ai}(x) \times \mu_{Bi}(y), for\ i = 1,2 \tag{5.7}$$

Also the nodes of layer 3 are the i-th node, which compute the ratio of each rule's firing strength to the sum of all rules' firing strengths, representing the normalized firing strength of each rule and labelled with N.

$$\overline{W}_i = \frac{W_i}{W_1 + W_2}, i = 1,2 \tag{5.8}$$

Each node of layer 4 is an adaptive node, with the individual nodes.

$$Q_i^4 = \bar{W}_i f_i = \bar{W}_i(p_i x + q_i y + r_i) \tag{5.9}$$

In layer 5 is labelled Σ, where the total output (f) is the sum of all incoming signals as shown in equation 5.10.

$$Q_i^5 = \sum_i \bar{W}_i f_i = \frac{\sum_i W_i f_i}{\sum_i W_i} \tag{5.10}$$

5.5 Implementation of ANFIS-PSS Controller

As explained in the previous paragraph, the stability of a linear closed-loop system can be determined from the location of the closed-loop poles by illustrating graph of root-locus in the left-half of S plane. If any of these poles lie on the right-half of the S plane, then with increasing time they give rise to the dominant mode, and the transient response increases monotonically or oscillate with increasing amplitude, representing an unstable system. In such a power system, as soon as the power is turned on, the output may increase with time. If no saturation takes place in the system and no mechanical stop is provided, then the system may eventually be subjected to damage and fail, since the response of a real physical system cannot increase indefinitely. Therefore, closed-loop poles lie to the left of the jω axis; any transient response eventually reaches equilibrium.

As mentioned earlier manual tuning is one method of tuning enabling the comparison of different stabiliser responses by observing the behaviour of whole system characteristics after disturbance from artificial faults (by short circuit phases with the ground for a short time period).

5.5.1 Training (Single to Three Phases)

In order to increase the controller response quality, the FLC was trained using a learning signal form M B stabiliser using the ANFIS architecture (J. Jang, 1993b). The training is done in two steps: simulation with disruption in the national grid network by the occurrence of short circuit between one phase and the ground; and a short circuit occurrence between three phase and the ground for a period of time of 0.1 milliseconds. For this purpose the outcome of the training is kept in files called ANFIS1D and ANFIS2D for each controller (FLC1 and FLC2 respectively). In terms

of expertise and experience, the training on the three phase fault is significantly better than single phase fault training, as shown in Figure 5.12.

Two types of ANFIS setting are used to produce a new controller, consisting of a combination of two types of fault used in this model, as explained below.

5.5.2 Single Phase Fault Training

In the first case, the system was trained on data generated using MB-PSS with simplified settings: IEEE type PSS4B, during single-phase fault which is dubbed as S-training. The single phase short circuit to the ground starts at time 0.8 sec and terminate at time 0.9 sec. Two controllers were trained, one for each generator, as they differ in their power ratings, as shown in Figure 5.13. ANFIS editor was used with the following parameters:

Error Tolerance: 0.001 (maximum error may be obtained)
Epochs: 50 (number of iterations)
Number of MF: 9 (number of membership functions)
Input MF Type: gauss2mf (input membership function type)
Output MF Type: linear (output membership function type)
ANFIS info:

- Number of nodes: 35
- Number of linear parameters: 27
- Number of nonlinear parameters: 24
- Total number of parameters: 51
- Number of training data pairs: 20001
- Number of checking data pairs: 0
- Number of fuzzy rules: 9

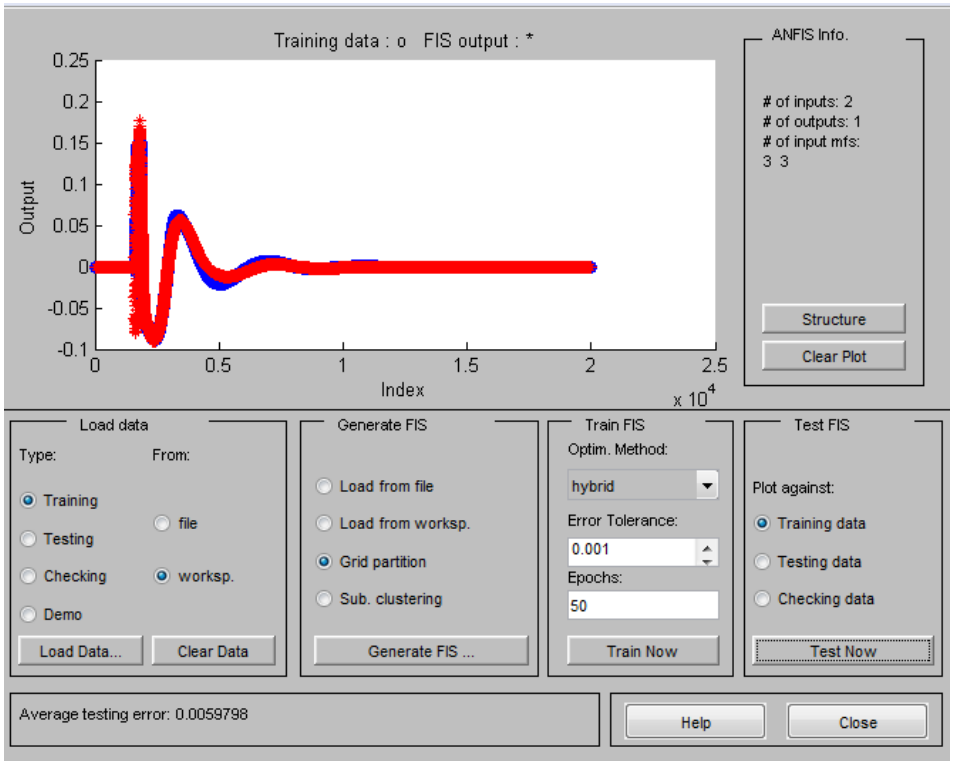

Figure 5.12: FLC training in ANFIS editor.

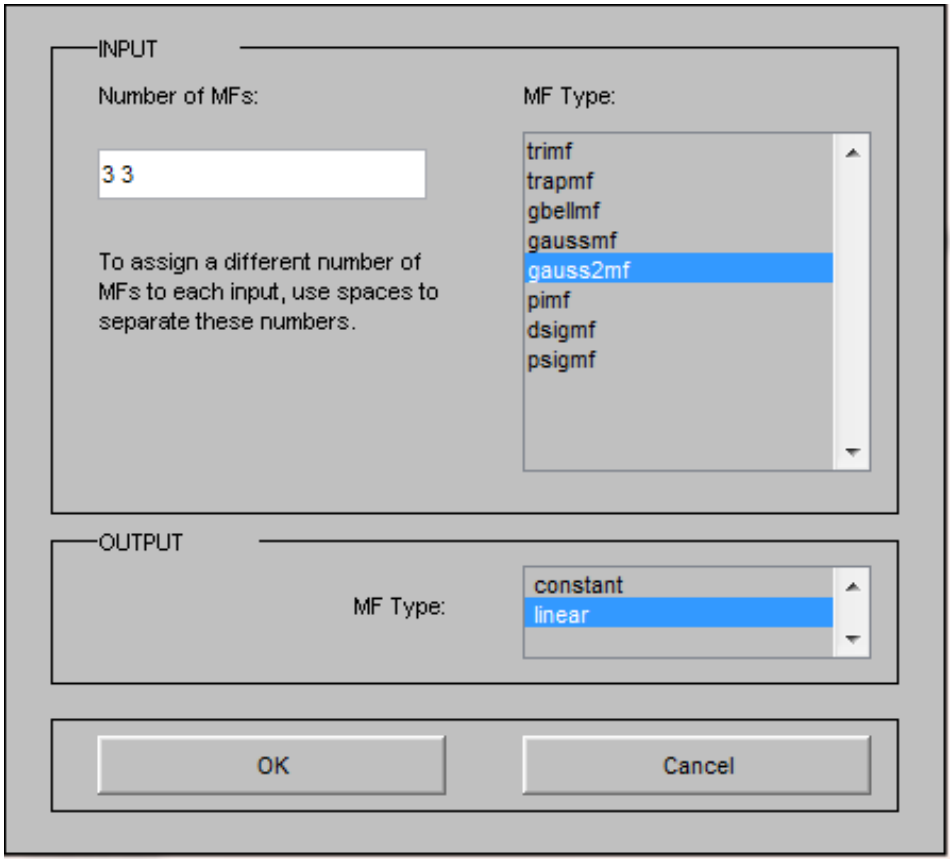

Figure 5.13: ANFIS editor parameters.

The scaling factors were manually tuned. After this training, the controller simulation results show that the system generated acceptable results in the case of a single-phase fault. Figure 5.14 shows the response of the system using the MB and the FLC controllers. It is clearly shown that the FLC achieved faster settling time. However, using the same controller, the system was tested with two and three faults. Figure 5.15 and Figure 5.16 show the system's response for both controllers. Furthermore these two figures show that both controllers have achieved the same settling time but the FLC has higher overshoot because it was trained on one phase fault.

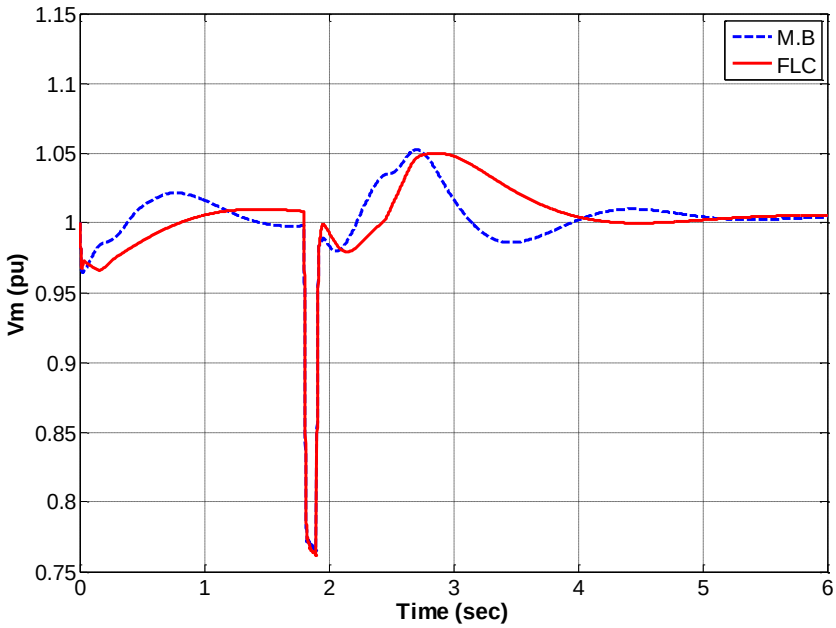

Figure 5.14: System's response to one phase fault with single phase training.

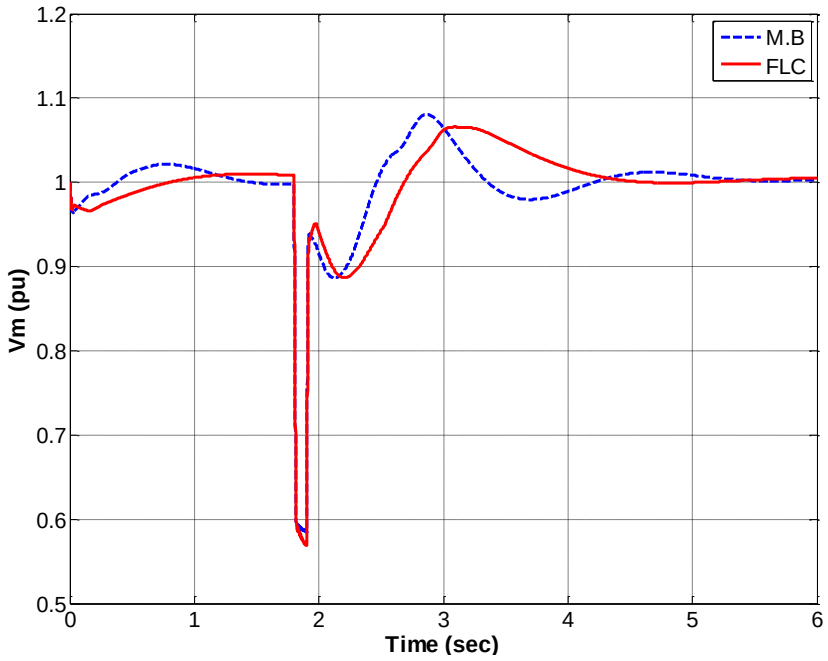

Figure 5.15: System's response to two phase faults with single phase training.

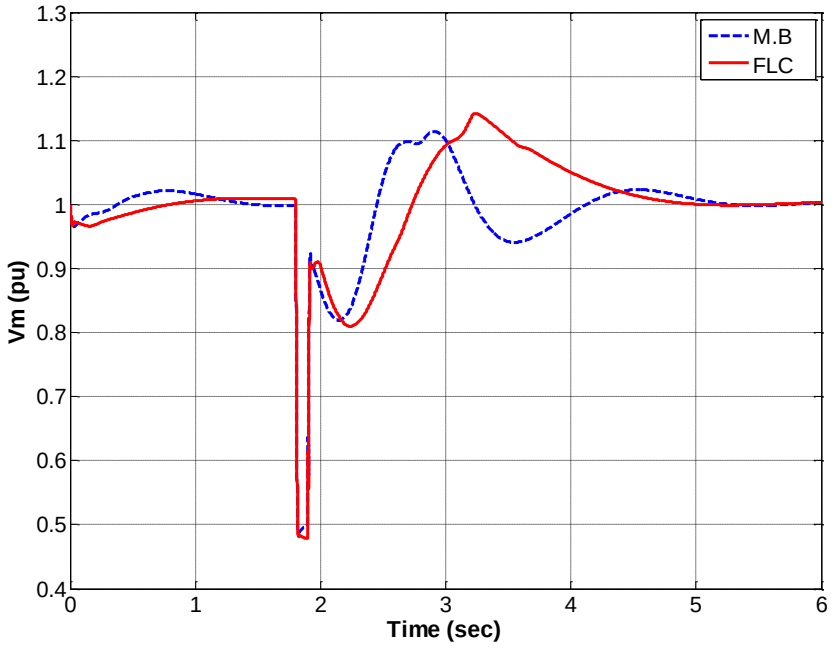

Figure 5.16: System's response to three phase faults with single phase training.

The maximum overshoot of the system is in the range of ±1.2%, as shown in Figure 5.14, denoting that the FLC does the task by reducing the settling time from 2.5 sec to 1.85 sec, moreover fluctuation of two peaks compared to MB controller with four peaks produces an overshoot of 5%. In Figure 5.15, the FLC response to the two phase fault reduces the settling time from 2.85 sec to 2 sec. Furthermore, with fluctuation of two peaks according to the MB controller with four peaks causes

8% overshoot. Figure 5.16 shows the ANFIS response to the three phase fault, which reduced the settling time from 3 sec to 2.7 sec. With fluctuation of two peaks according to the MB controller with four peaks, 11.4% overshoot occurs, lower than the ANFIS controller. The results are tabulated in Table 5.3.

Table 5.3: Simulation results for three phase fault with single phase training.

	Multi Band Controller			ANFIS Controller (manual tuning)		
	Settling time (sec)	Overshoot (%)	Fluctuation	Settling time (sec)	Overshoot (%)	Fluctuation
1 phase fault	2.50	5.24	4 peaks	1.85	5.0	2 peaks
2 phase fault	2.85	8.00	4 peaks	2.00	6.5	2 peaks
3 phase fault	3.00	11.40	4 peaks	2.70	14.2	2 peaks

5.5.3 Three Phase-Training

In this case, the FLC1 and FLC2 were trained on data generated using MB-PSS with simplified settings: IEEE type PSS4B, during three phase fault conditions (dubbed three phase training). The three phase on tie-line short circuit to the ground starts at time 1.8 sec and terminates at time 1.9 sec, as shown in Figure 5.17. ANFIS editor is used with the following parameters:

Error Tolerance: 0.001 (maximum error may be obtained)
Epochs: 30 (number of iterations)
Number of MF: 9 (number of membership function)
Input MF Type: gauss2mf (input membership function type)
Output MF Type: linear (output membership function type)

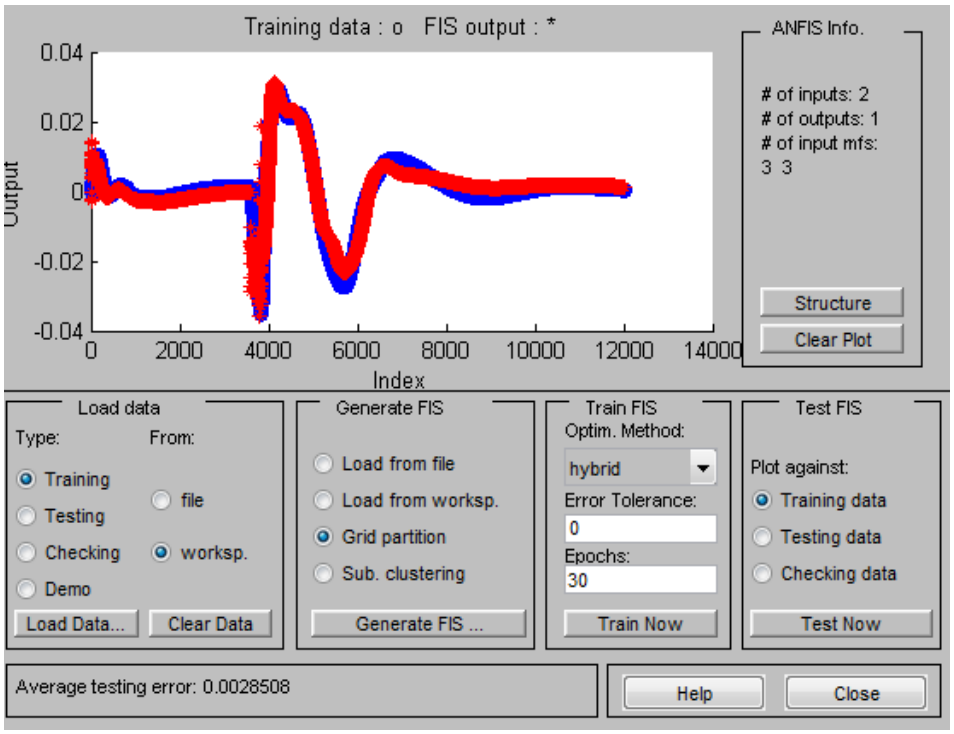

Figure 5.17: FLC training in ANFIS editor.

The two FLCs controllers were trained for each generator and the system was simulated to different fault conditions (single and multi-phase conditions). Simulation results show that the controller performed well for single, two and three phase fault, as shown in Figure 5.18 to Figure 5.20 (respectively).

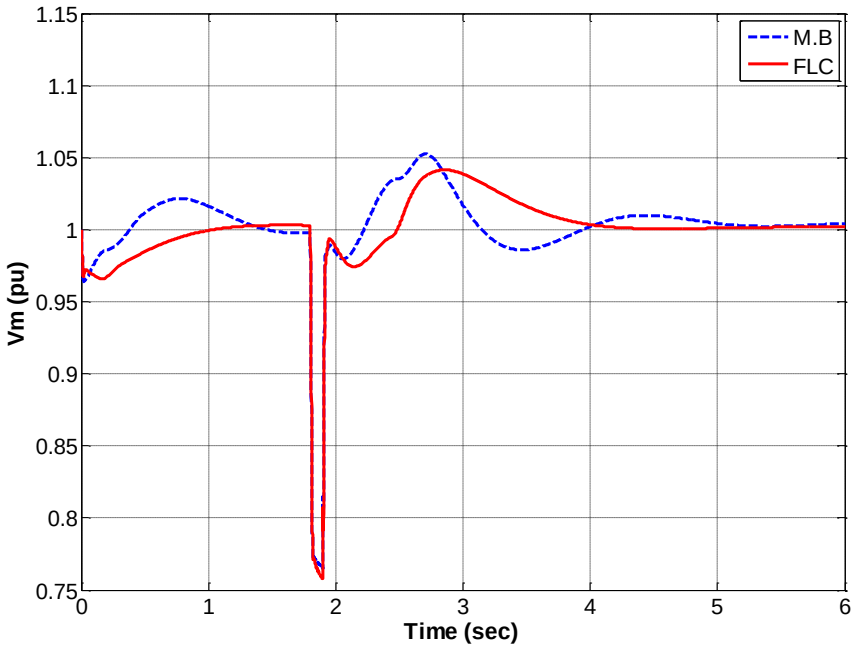

Figure 5.18: System's response to one phase fault with three phase training.

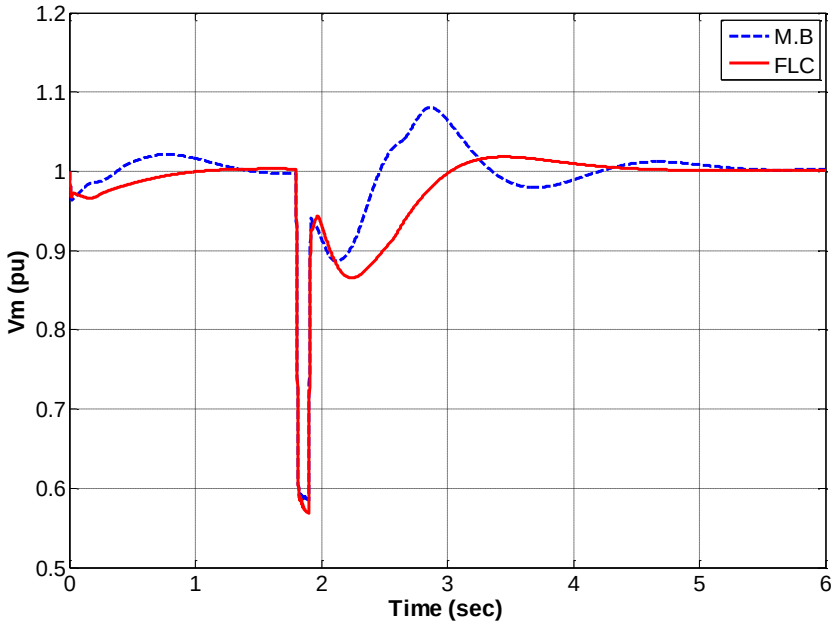

Figure 5.19: System's response to two phase faults with three phase training.

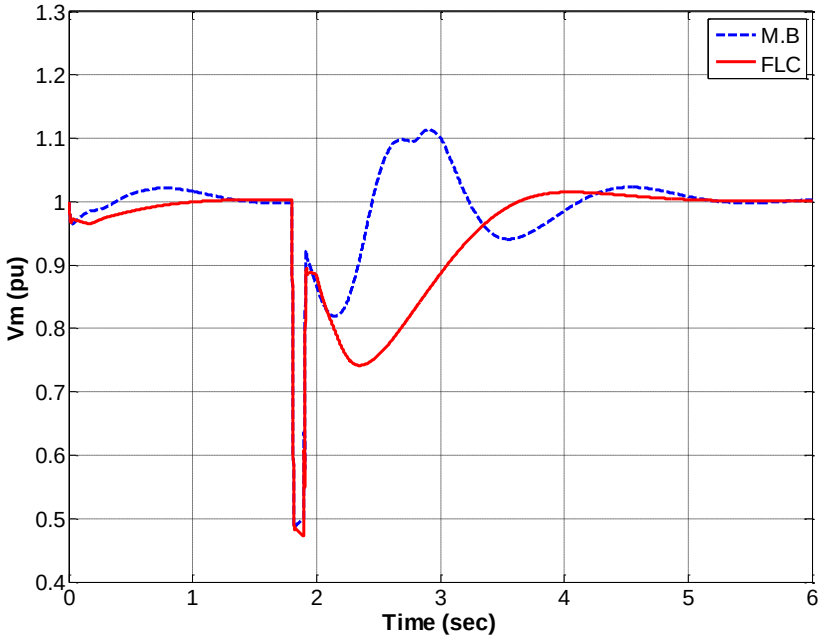

Figure 5.20: System's response to three phase faults with three phase training.

The system maximum overshoot is within ±1.2% of output, as shown in Figure 5.17. The response of the FLC controller with one fault has reduced the stability time from 2.5 sec to 1.75 sec with fluctuation of two peaks compared to the MB controller. In Figure 5.19, the FLC reduced the stability time from 2.9 sec to 1.9 sec, with less fluctuation compared to MB controller, and smaller overshoot for FLC (1.8%) while MB has 8%. More importantly, with the probability of fault happening in three

phase, the FLC reacted with high efficiency, reducing the overshoot from 11.4% to 1.5% and the stability period from 3 sec to 2.5 sec. However, a small delay in final response was noted, as shown in Figure 5.20. The numerical results are shown in Table 5.4

Table 5.4: Simulation results for three phase fault with three phase training.

	Multi Band Controller			ANFIS Controller (manual tuning)		
	Settling time (sec)	Overshoot (%)	Fluctuation	Settling time (sec)	Overshoot (%)	Fluctuation
1 phase fault	2.5	5.2	4 peaks	1.75	4	2 peaks
2 phase fault	2.9	8	4 peaks	1.9	1.8	2 peaks
3 phase fault	3	11.4	4 peaks	2.5	1.5	1 peak

5.6 ANFIS PSS Response to Ground Fault in Tie Line

The complete power system model with FLC is simulated using Matlab/Simulink toolbox, which provides a simulation sample of the electrical power grid, including in this case two generators, one with 5000 MW capacity and the other with 1000 MW capacity, both turbine-driven. These two generators are mutually connected to the grid via high voltage transformers and bas bar line with length of 700 km together with a load of 5000 MW as well as fault breaker, as explained in chapter four. In this case we will test the FLC controller when manually tuned, and compare it with Multi Band (MB) controller, one of the newest and most advanced conventional controllers, to drive the power stability control for both machines, and to compare the response of different controllers used to tune this model when three phase to ground fault in tie-line occurs.

5.6.1 Rotor Speed Deviation on Machine A with (Manual Tuning)

Figure 5.21 to Figure 5.23 show the generator speed deviation responses on Machine A during scenarios, with manually tuning of scale of factor. The rotor speed deviation after fault for Machine A, as a reflection of FLC1 reactions for three types of faults single, double and three phase fault, displays good efficiency (almost identical to the multi band power stability controller).

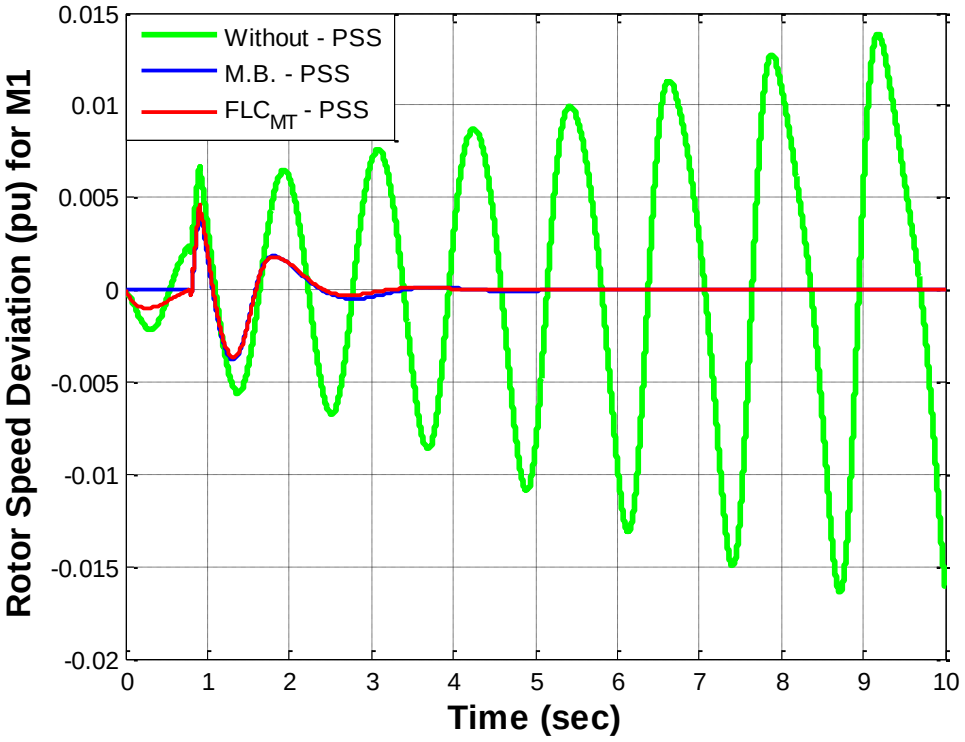

Figure 5.21: Rotor speed deviation at Machine A for one phase fault.

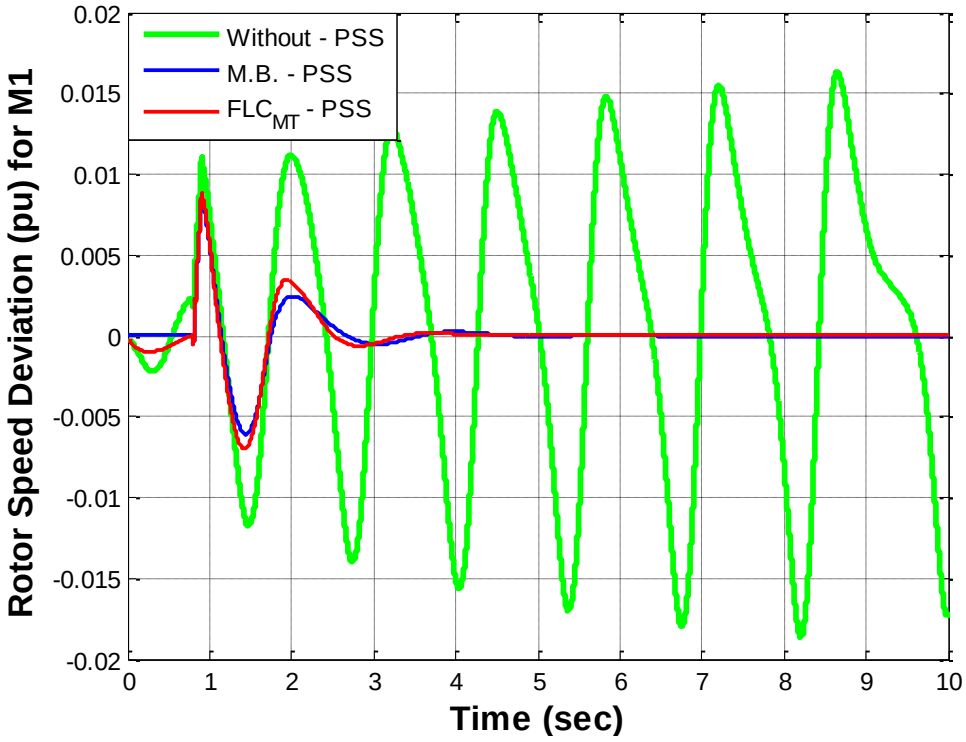

Figure 5.22: Rotor speed deviation at Machine A for two phase fault.

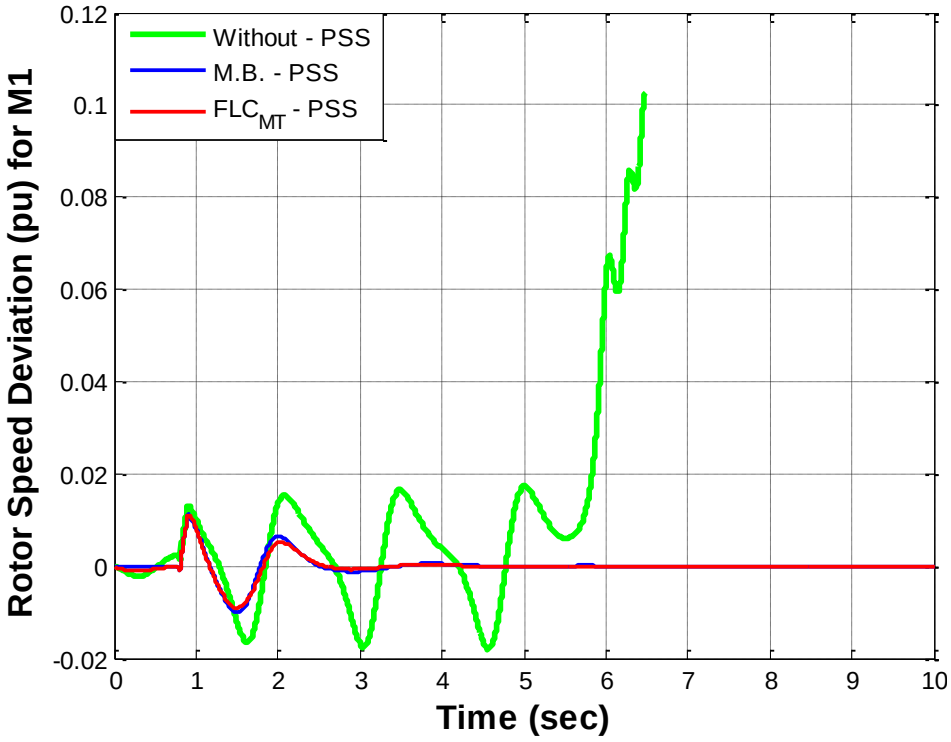

Figure 5.23: Rotor speed deviation at Machine A for three phase fault.

5.6.2 Rotor Speed Deviation on Machine B with (Manual Tuning)

Figure 5.24 to Figure 5.26 show generator speed deviation responses on the Machine B during scenarios, with manually tuning of scale of factor. The rotor speed deviation after fault as, as a reflection of FLC2 reactions for three types of faults single, double and three phase fault, displays high efficiency slightly more stable than the multi band power stability controller). It can be seen that the network becomes unstable if any type of faults occurs when the network is running without a power system stabilizer. Table 5.5 shows the comparison between the FLC2 and MB stabilizer.

Table 5.5: Speed deviation response for three phase fault with three phase training (Machine B).

	Multi Band Controller			ANFIS Controller Manual tuning (FLC_{MT})		
	Settling time (sec)	Overshoot (%)x10^{-3}	Fluctuation	Settling time (sec)	Overshoot (%) x10^{-3}	Fluctuation
1 phase fault	4.2	0.4	3 peaks	2.4	0.08	1 peaks
2 phase fault	3.6	0.8	4 peaks	2.2	0.15	2 peaks
3 phase fault	3.5	1.2	4 peaks	2	0.2	1 peak

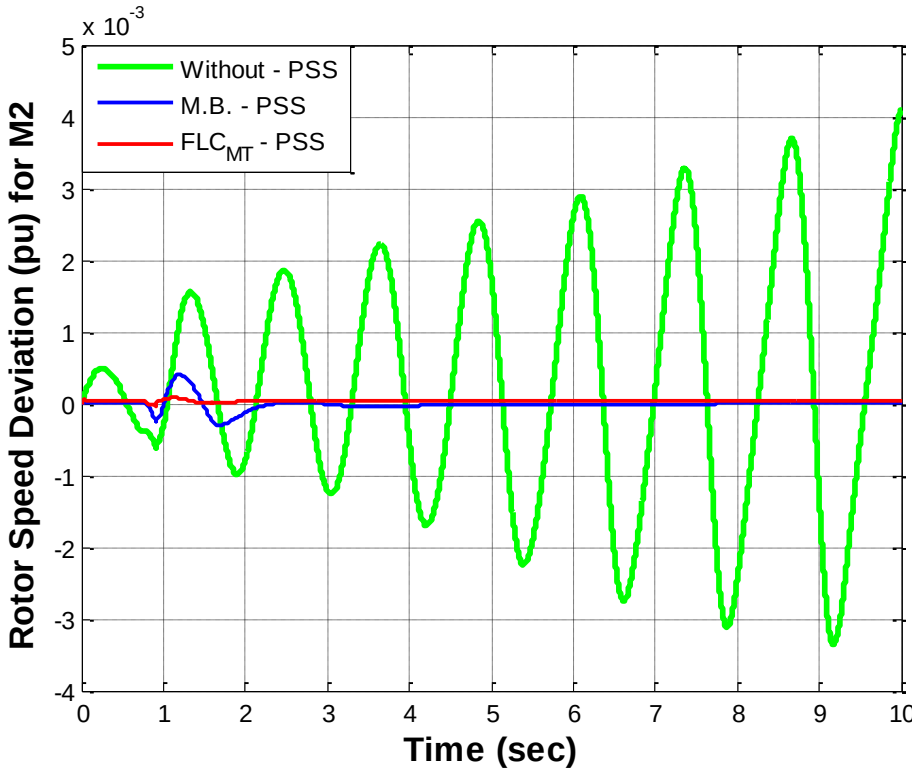

Figure 5.24: Rotor speed deviation at Machine B for one phase fault.

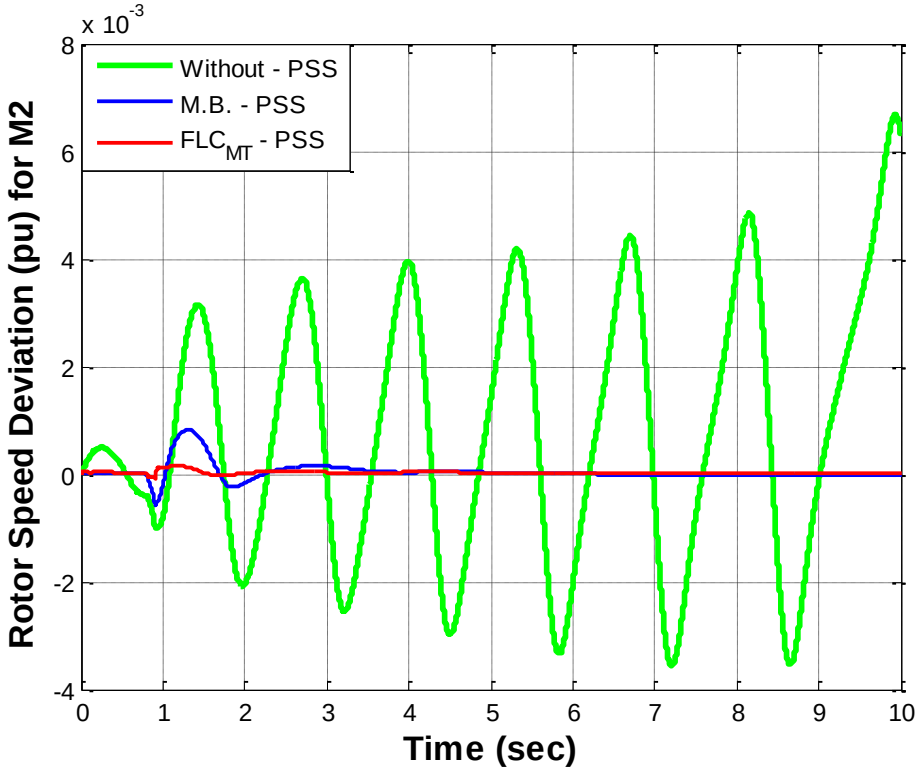

Figure 5.25: Rotor speed deviation at Machine B for two phase fault.

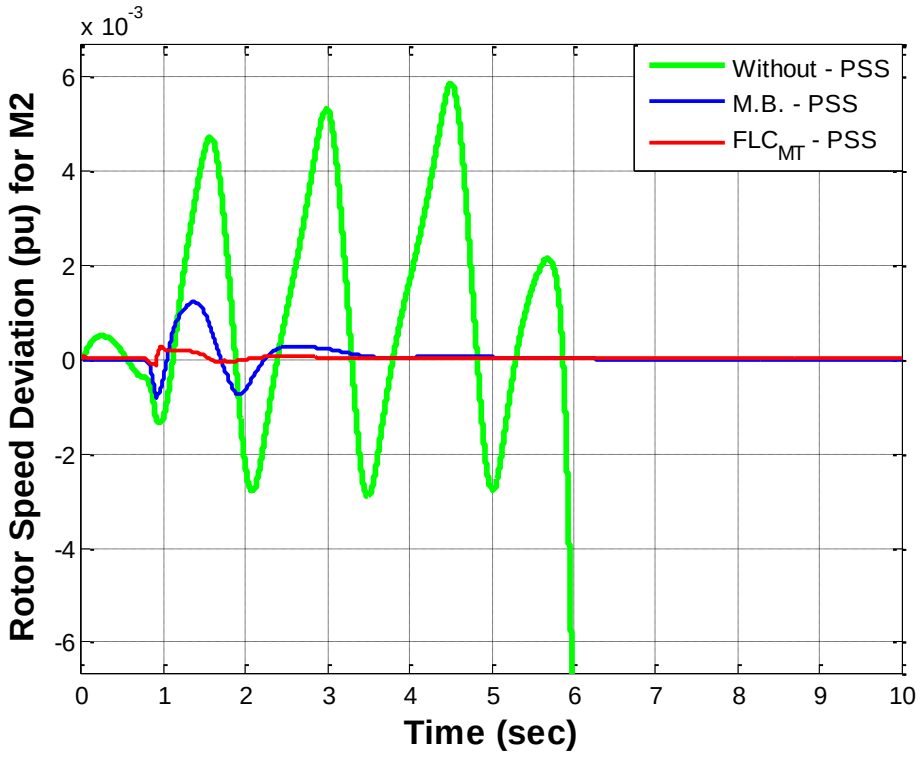

Figure 5.26: Rotor speed deviation at Machine B for three phase fault.

5.6.3 Power Quality Control

In order to test the network as a whole package, as well as to monitor the quality control, the voltage in the network was observed during the all the test conditions. Figure 5.27 to Figure 5.29 show the response of the system to the different faults and illustrate the comparison between MB stabilizer, controller auto-trained and manually tuned FLC_{MT} and without PSS. As mentioned previously, the best value established manually for Ke1 = 2, Kc1 = 3.75, Ko1 = 2.25 and Ke2 = 5, Kc2 = 3.75, Ko2 = 10.

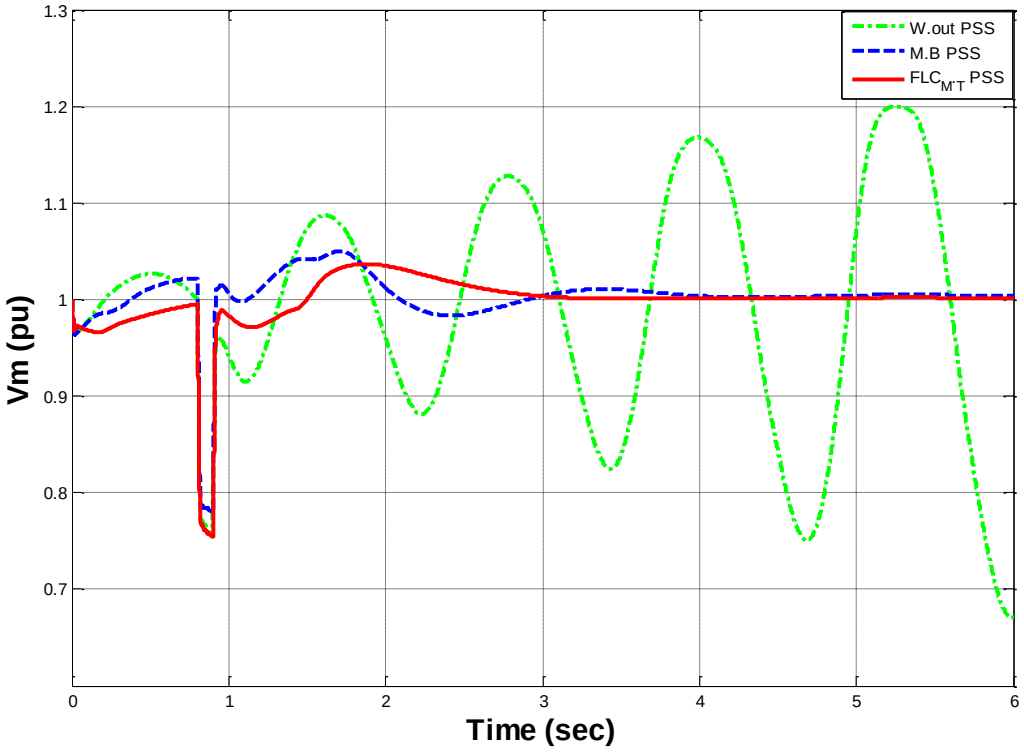

Figure 5.27: System behaviours during one phase fault.

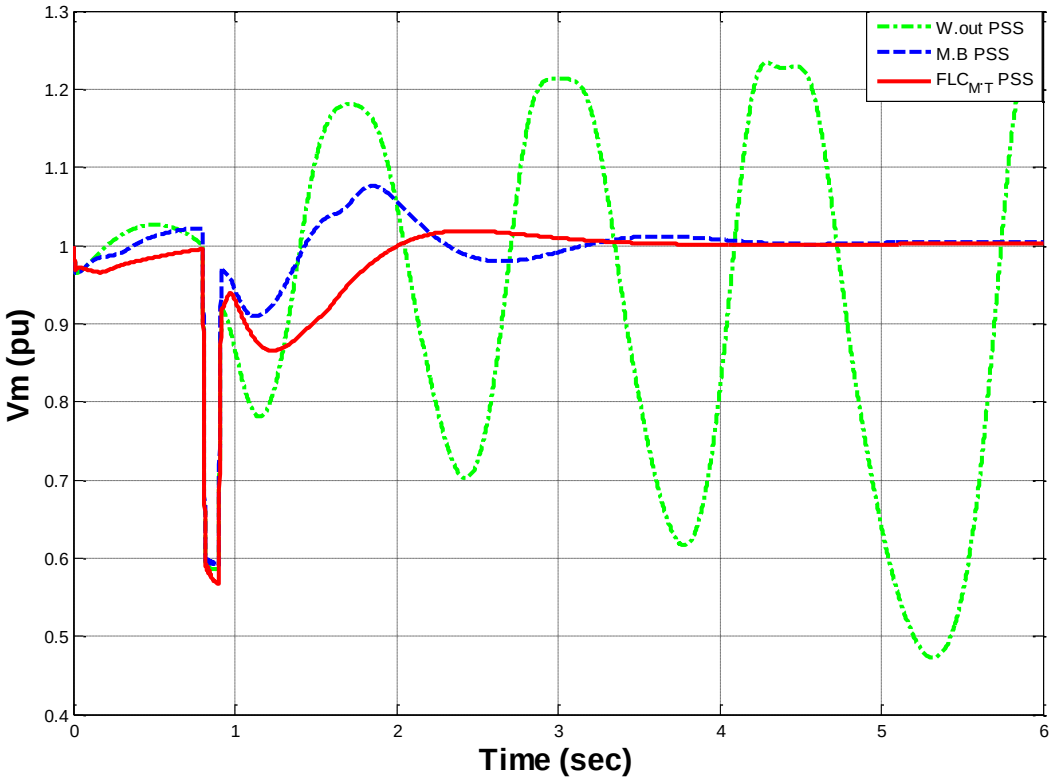

Figure 5.28: System behaviours during two phase fault.

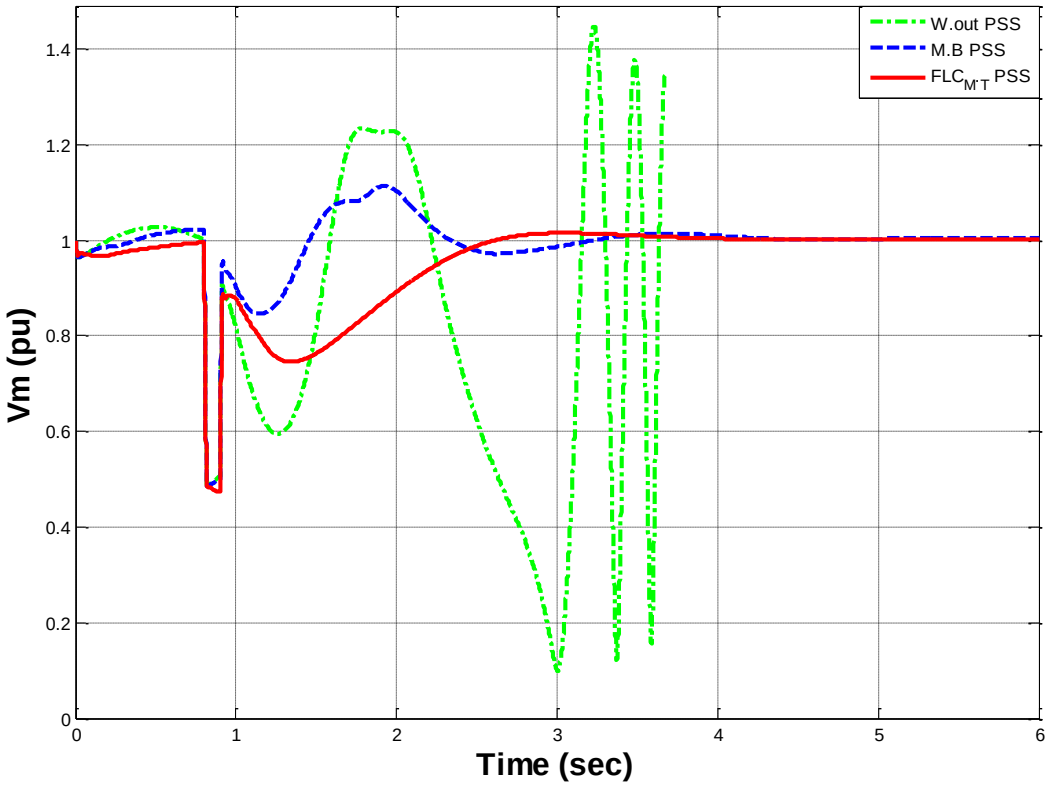

Figure 5.29: System behaviours during three phase fault.

The system stability is within ±1.2% of output, as shown in Figure 5.27; the response of the FLC controller with one fault reduced the stability time from 2.5 sec to 1.75 sec with fluctuation of two peaks compared to the MB controller. In Figure 5.28, the FLC reduced the stability time from 2.9 sec to 1.9 sec, with less fluctuation compared to the MB controller, and smaller overshoot for FLC (1.8% compared to 8% of the MB). In three phase fault the FLC reacted with high efficiency, reducing the overshoot from 11.4% to 1.5% and the stability period from 3 sec to 2.5 sec, with a minor delay in final response, as shown in Figure 5.29. The numerical results are shown in Table 5.6, which compares the behaviour of MB stabiliser with fuzzy logic controller auto-trained and manually tuned FLC_{MT}.

Table 5.6: Simulation results for three phase fault with three phase training.

	Multi Band Controller			FLC Controller (Manual tuning)		
	Settling time (sec)	Overshoot (%)	Fluctuation	Settling time (sec)	Overshoot (%)	Fluctuation
1 phase fault	2.5	5.2	4 peaks	1.75	4	2 peaks
2 phase fault	2.9	8	4 peaks	1.9	1.8	2 peaks
3 phase fault	3	11.4	4 peaks	2.5	1.5	1 peak

5.7 Auto Tuning of Scaling Factors

Particle swarm Optimisation was selected as the Optimisation search method, as explained in chapters two and three. This section presents the results of using the algorithm to find the best values of scale of factor for both power system stability controllers; as explained below, the results were very satisfactory.

5.7.1 Intelligent Optimisation and Cost Function

In this stage the scaling factors of fuzzy logic controllers will be selected automatically using particle swarm Optimisation to improve the response of both controllers. By tuning the power system stability control in the electric grid, the actual rate of speed deviation and speed error can be monitored and analysed for each unit of electric power generation in the network. The different ratios of main objectives must be observed, which can be controlled depending on the type and requirements of the system to be controlled. In this case each of the following four variables affecting the final output (Vm) after Static Var Compensator (SVC) of whole system was monitored:

- Minimising the settling time.
- Minimising steady state error.
- Minimising the overshoot.
- Minimising the first negative off peak.

The SPSO programme was used to tune all controllers in the system with objective function to minimise the weight average for the following objectives: settling time, steady state error, overshoot and first negative peak.

$$J = \int_0^{max} [w_1|e(t)|]dt + w_2 P_v + w_3 OS + w_4 t_{lag} \qquad (5.11)$$

where *e(t)* is system error, P_v is the peak value, *OS* is the overshoot, t_{lag} is the lag time and w_1, w_2, w_3, w_4 are coefficient parameters which reset to the following values 0.45, 0.2, 0.2 and 0.15 respectively.

5.7.2 Particle Swarm Optimisation

Particle swarm Optimisation is a global minimization technique for dealing with problems in which a best solution can be represented as a point and/or a velocity. Each particle assigns a value to its position, based on certain metrics, and remembers the best position seen, which it communicates to the other members of the swarm. The particles adjust their own positions and velocity based on this information. The communication can be common to the whole swarm, or be divided into local neighbourhoods of particles (Kennedy & Eberhart, 1995a).

5.7.3 FLC_{PSS} Auto Tuning

In this stage of the controller design the particle swarm Optimisation theory was used to optimize the selection of the scaling factors. The optimiser's objective function is based on four objectives: steady state error, settling time, overshoot, and the negative peak. The first and second objective has the highest priority, while the third has the next priority, and the fourth objective has the least priority (Jiang, Zheng, & Chen, 2007). The aim is to reduce the settling time first, then reduce the overshoot, and the last objective is to reduce the first negative peak of oscillation. The optimized values and comparison of the scaling factors between the fuzzy logic controller auto-trained and manually tuned FLC_{MT}, and fuzzy logic controller auto-trained and auto-tuned FLC_{AT} were as follows: for FLC1 Ke1 = 0.862, Kc1 = 2.088 and Ko1 = 2.080, while for FLC2 Ke2 = 3.820, Kc2 = 4.811 and Ko2 = 6.620 as shown in Table 5.7.

Table 5.7: Final value for auto-tuning scaling factor.

	FLC1			FLC2		
	Ke1	Kc1	Ko1	Ke2	Kc2	Ko2
Manual Tuning (FLC_{MT})	2	3.75	2.25	5	3.75	10
Auto. Tuning (FLC_{AT})	0.862	2.088	2.080	3.820	4.811	6.620

5.8 Simulation Results

5.8.1 Rotor Speed Deviation on Machine A with Auto Tuning

Figure 5.30 to Figure 5.32 show generator speed deviation responses on the Machine A during scenarios, with auto tuning for the scale of factor. The rotor speed deviation after fault for Machine A, as a reflection of FLC1 reactions for three types of faults single, double and three phase fault, was highly efficient in terms of overshoot at the second and third harmonic, slightly more stable compared with the multi band power stability controller and fuzzy logic manually tuned controller (FLC_{MT}). Table 5.8 shows the comparison between three power system stabiliser (M.B, FLC_{MT} and FLC_{AT}).

Table 5.8: Speed deviation response for three phase fault with three phase training and auto tuning (Machine A).

	Multi Band Controller			Fuzzy Logic Controller (Manual tuning)			Fuzzy Logic Controller (Auto tuning)		
	Settling time (sec)	First -Peak (%)x10-3	Second +Peak (%)$x10^{-3}$	Settling time (sec)	First -Peak (%)$x10^{-3}$	Second +Peak (%)$x10^{-3}$	Settling time (sec)	First -Peak (%)$x10^{-3}$	Second +Peak (%)$x10^{-3}$
1 phase fault	4.7	-3.7	1.8	4.2	-3.6	1.7	3.9	-2.5	0.8
2 phase fault	4.5	-6	2.5	4.2	-7	3.5	3.9	-4.7	1.5
3 phase fault	4.5	-10	6.5	4	-9	5.1	3.8	-4.8	1.4

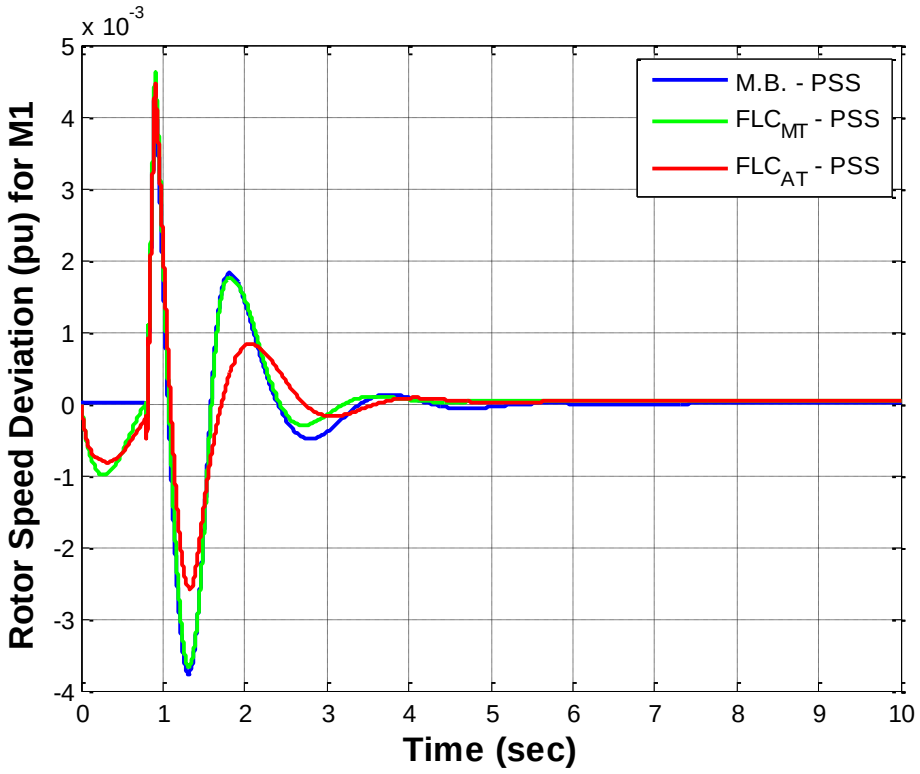

Figure 5.30: System behaviours during one phase fault.

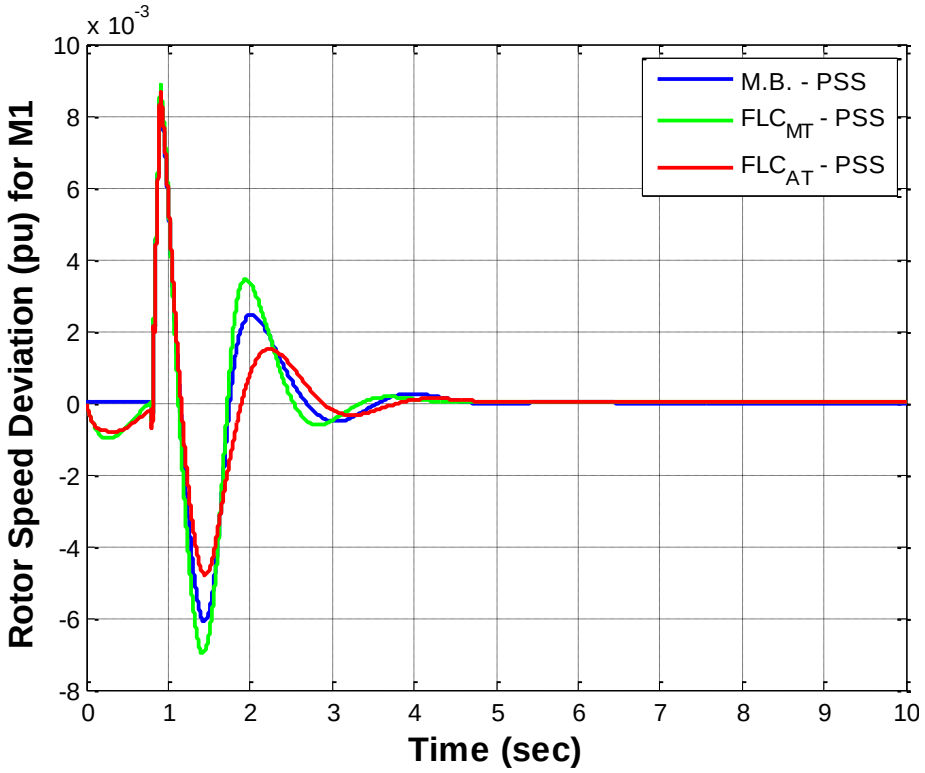

Figure 5.31: System behaviours during two phase fault.

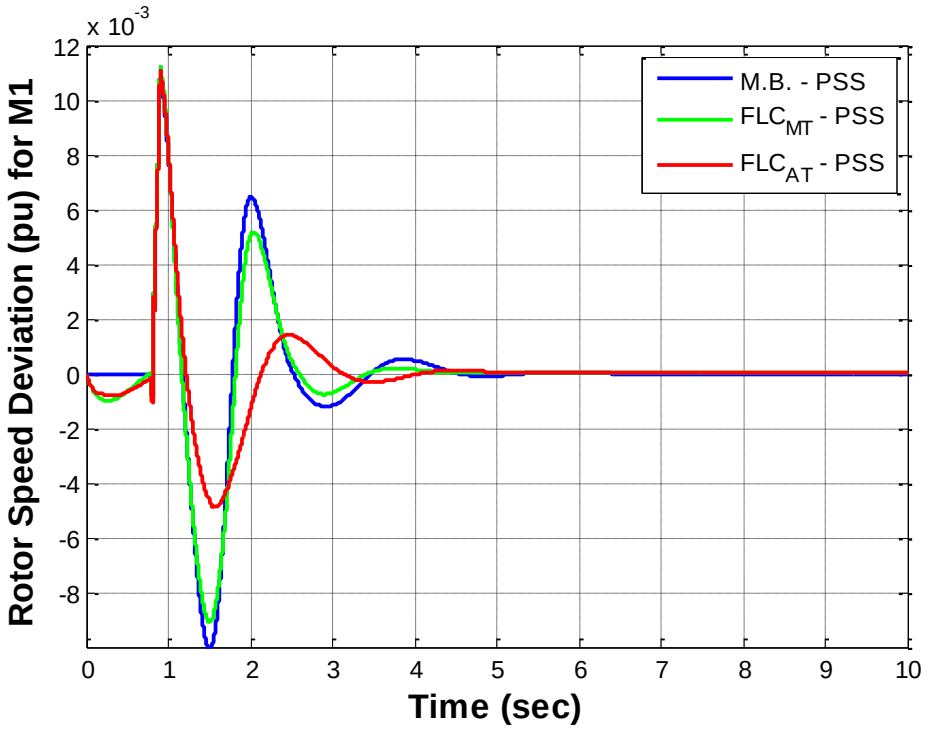

Figure 5.32: System behaviours during three phase fault.

5.8.2 Rotor Speed Deviation on Machine B with Auto Tuning

Figure 5.33 to Figure 5.35 show generator speed deviation responses on the Machine B during scenarios, with auto tuning for the scale of factor. The rotor speed deviation after fault for Machine B, as a reflection of FLC2 reactions for three types of faults single, double and three phase fault, exhibited high efficiency in all behaviours such as first overshoot, second and third harmonic and settling time, displaying more stability than multi band power stability controller and fuzzy logic manually tuned controller (FLC_{MT}), with steady stat error 3.1×10^{-5}. Table 5.9 shows the comparison between three power system stabiliser (M.B, FLC_{MT} and FLC_{AT}).

Table 5.9: Speed deviation response for three phase fault with three phase training. and auto tuning (Machine B).

	Multi Band Controller			FLC Controller (Manual tuning)			FLC Controller (Auto tuning)		
	Settling time (sec)	First -Peak (%)x10^{-4}	Second +Peak (%)x10^{-4}	Settling time (sec)	First -Peak (%)x10^{-4}	Second +Peak (%)x10^{-4}	Settling time (sec)	First -Peak (%)x10^{-3}	Second +Peak (%)x10^{-3}
1 phase fault	4.6	-2.6	4	2.7	-0.34	0.82	1.8	-2.5	0.8
2 phase fault	5	-5.8	8.2	2.8	-1	1.5	2.2	-0.8	1.6
3 phase fault	5.5	-8.2	12.1	2.9	-1.5	2.7	2.3	-1.2	3.3

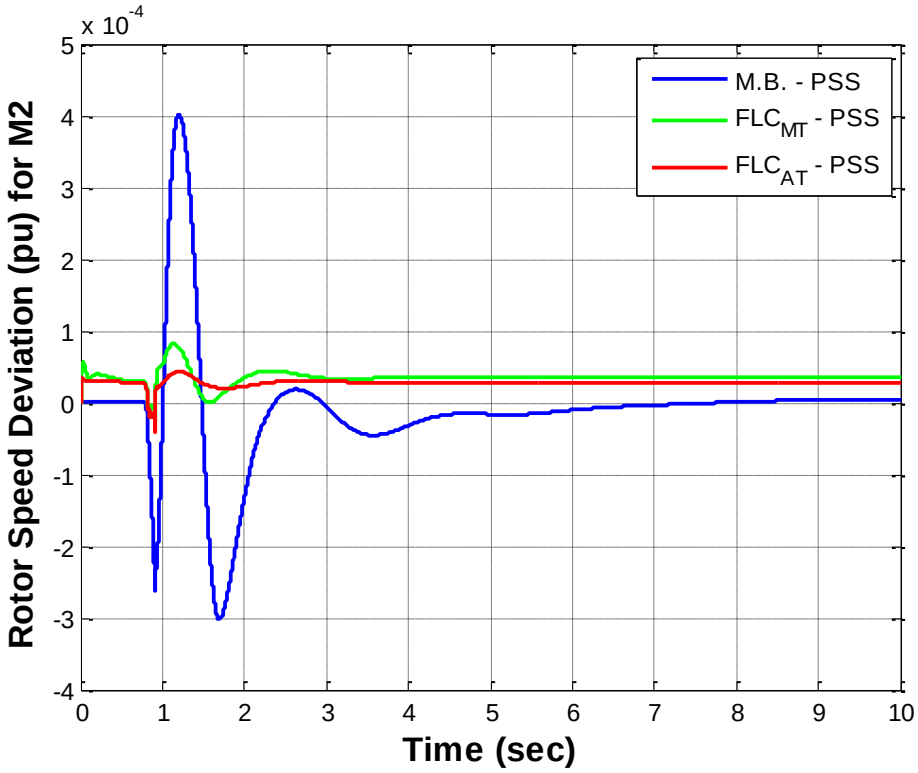

Figure 5.33: System behaviours during one phase fault.

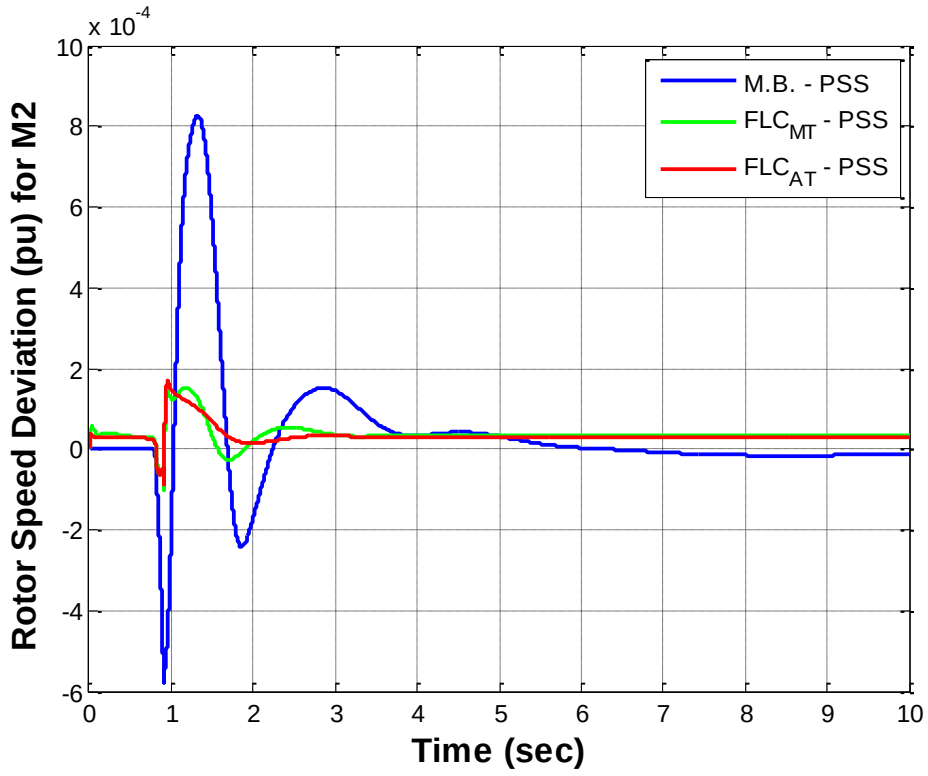

Figure 5.34: System behaviours during two phase fault.

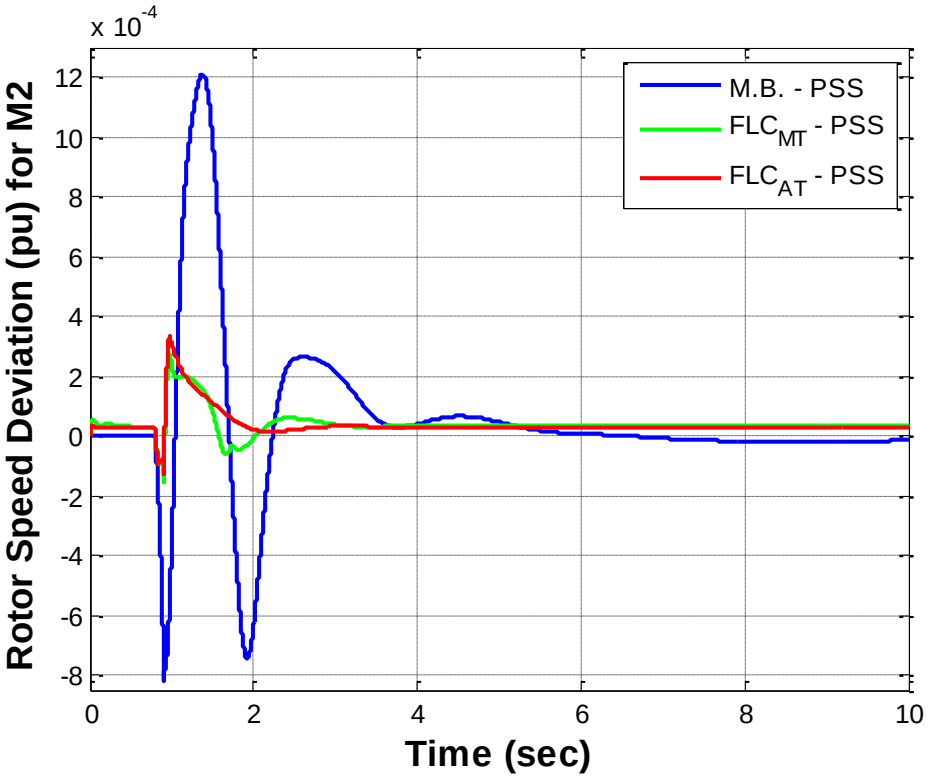

Figure 5.35: System behaviours during three phase fault.

5.8.3 Power Quality Control with Auto Tuning

In order to test the network as a whole package, as well as to monitoring the quality control, the voltages were observed in the network during all test conditions. Figure 5.36 to Figure 5.38 show the response of the system to the faults and illustrate the comparison between MB stabilizer, controller auto-trained and manually tuned FLC_{MT} and auto-trained and auto-tuned FLC_{AT}. As mentioned previously, the best values established automatically by the optimiser are: for Ke1 = 0.862, Kc1 = 2.088, Ko1 = 2.080 and Ke2 = 3.820, Kc2 = 4.811, Ko2 = 6.620.

The simulation results are shown in Figure 5.36 to Figure 5.38 for single, double, and triple phase faults respectively for three type of stabiliser (M.B) PSS, (FLC_{MT}) PSS and (FLC_{AT}) PSS.

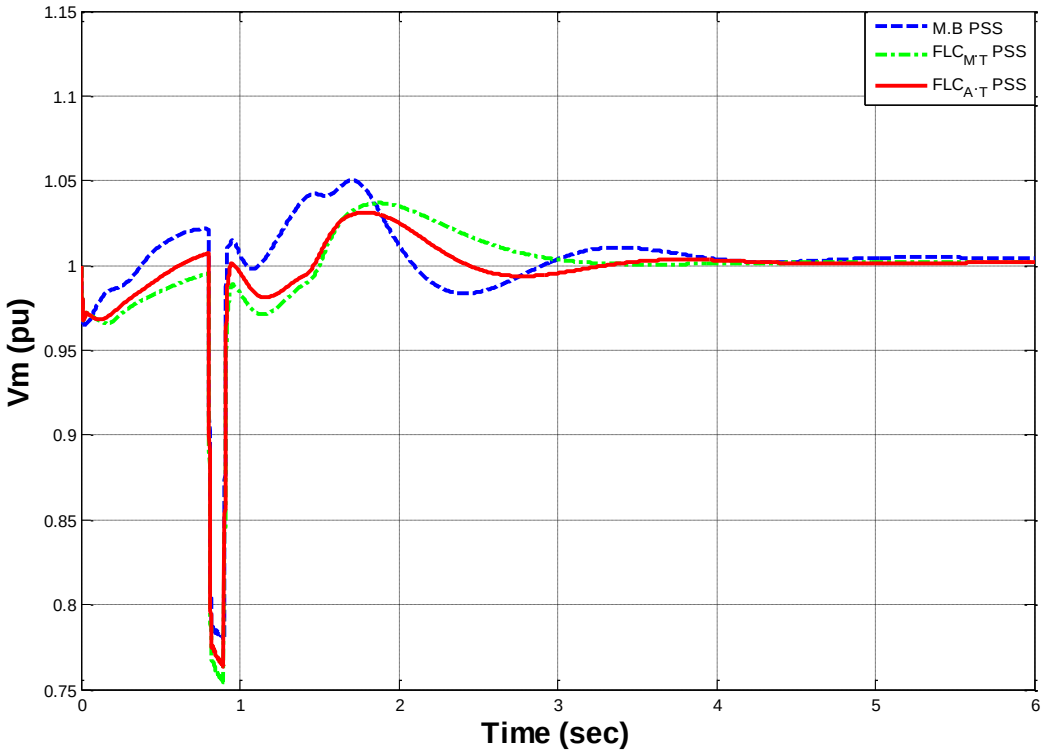

Figure 5.36: System behaviours during one phase fault.

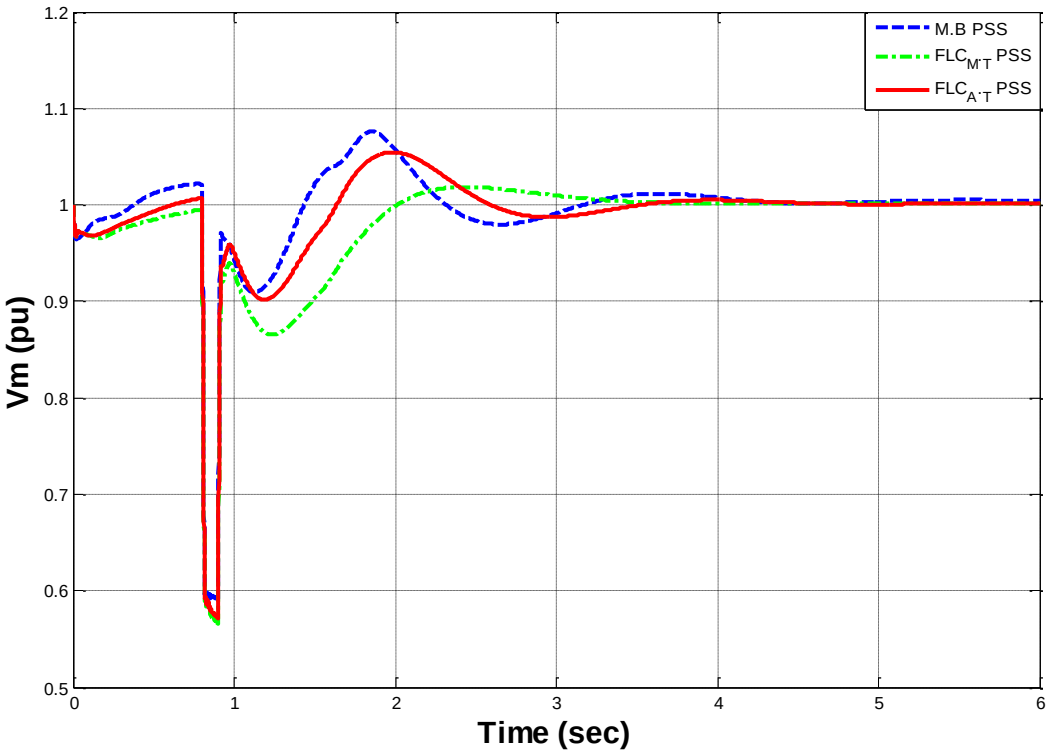

Figure 5.37: System behaviours during two phase fault.

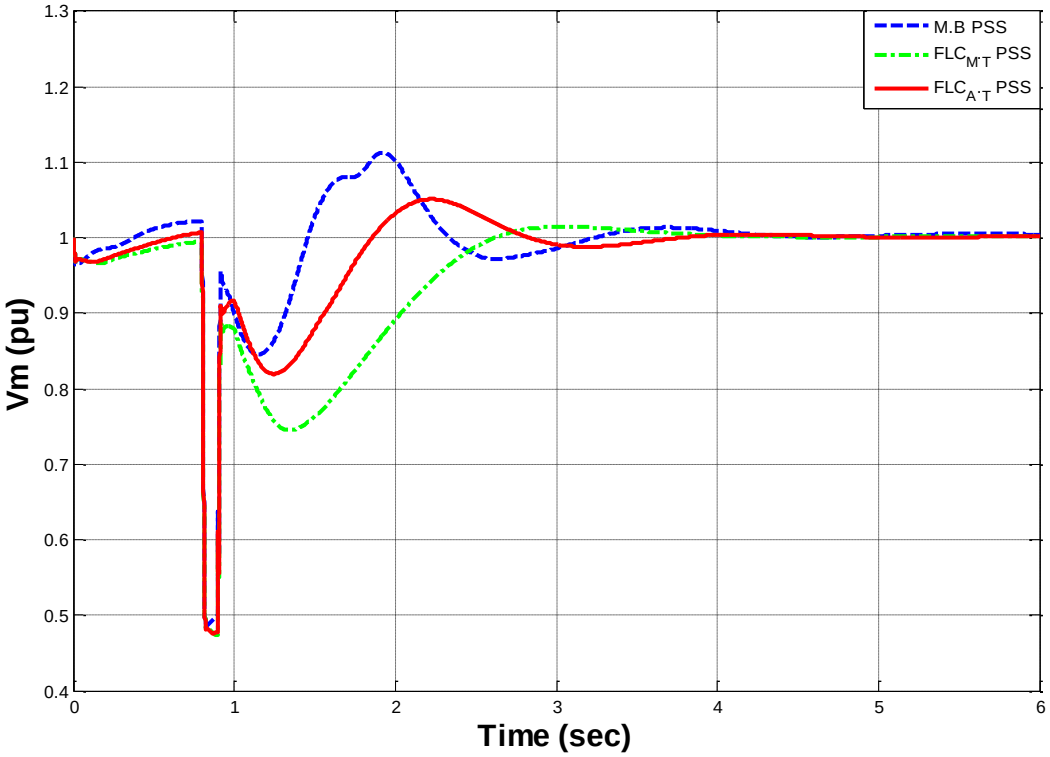

Figure 5.38: System behaviours during three phase fault.

5.9 Summary

From the simulations it is clear that using SPSO as an Optimisation technique gives good results. FLC based on SPSO obtains the advantages of FLC (as fuzzy logic technique) and the advantages of SPSO as an Optimisation technique, selecting the best values to obtain the best results with minimum error in a fast way. Thus the new controller (FLC based on SPSO) is capable of damping oscillations, increasing system stability with minimum energy consumed in fast way, and it has maximum sensitivity to changes in operating conditions. FLC based on SPSO controller has a better response than conventional PSS controller in tuning lead compensator with lagging power factor load (commonly use), and nearly the same response in tuning the lead compensator with leading power factor load.

Chapter 6: Supervisory Control

6.1 Introduction

The general understanding of the supervisory control is one or more human operators continually programming and receiving information from a computer interconnected through artificial effectors and sensors to the controlled process or task environment (Sheridan, 1992). Supervisory expert control concentrates on general information about the process and the controller. The decision-making in the supervisory control system is related to situations involving major disturbances, technical faults, inappropriate human actions, and a combination of such events (Rasmussen, 1985). In such events the established control algorithm does not apply, and the planning for proper actions by the controller depends on knowledge about the functional properties of the system. Planning for a new control strategy will depend on information about the process characteristics in a specific situation. Such a system should be capable of performing the following tasks: monitoring the performance of the controller and the process, detecting possible system component failure or malfunctioning and replacing the control algorithm to maintain stability and selecting the appropriate control algorithm best suited for a particular situation. Such a system can be formed in a closed loop to provide a conceptualized hierarchical system which consists of a supervision level as the highest hierarchical level, and the basic control level as the lowest. In general, more tasks can be handled by the supervisory control algorithm (De Silva & MacFarlane, 2007) (e.g. start up and shut down procedures, process Optimisation, fault diagnosis, response to malfunctioning behaviour, pattern recognition start and stop parameters estimation, and alarm handling procedure).

This chapter describes the design and implementation of advanced Supervisory Power System Stability Controller (SPSSC) using Neuro-fuzzy system, and Matlab S-function tool, whereby the controller is taught from data generated by simulating the system for the optimal control regime. The controller is compared to a multi-band control system which is utilized to stabilize the system for different operating conditions. Simulation results show that the supervisory power system stability controller produced better control action in stabilizing the system for conditions such as: normal, after disturbance in the electrical grid as a result of changing of the plant

capacity like switching renewable energy units, high load reduction or in the worst case of fault in operating the system, e.g. phase short circuit to ground. The new controller decreased the settling time and overshoot after disturbances, which means that the system can reach stability in the shortest time with minimum disruption. Such behaviour improves the quality of the provided power to the power grid.

The chapter is organized into five sections, including this introductory section, which is followed by a brief summary of supervisory control system then an explanation of the design and development of hierarchical supervisory control system using fuzzy logic controller. The Optimisation of the controller and fault detection with diagnosis in different stages is then explained, involving scaled-up the system and auto tuning the controller, and the new supervisory controller response to single, multi-phase fault and scenario of consecutive serious fault. Section five presents a summary of the chapter.

6.2 Supervisory Control

To analyse and study any system, modal is a very important step since it is related to process characterization and design studies. Previously, it was thought that a complicated mathematical approach could model a system more accurately, but this still has problems when non-linear, complicated and undefined systems are encountered. Conversely, the human mind can easily reach a very good result when dealing with very complicated system, such as food preparation, playing football, dealing with machines such as driving a car off-road and so on; all of these tasks can be executed with no consideration of mathematical models that describe the processes. However, there are great potential applications of such models developed during the past few decades, with the emergence of models that can function similar to human thinking, using fuzzy set theory proposed by Zadeh (Zadeh, 1996). Later, several authors conducted research into fuzzy modal, which divided into six different methods (Abbod, von Keyserlingk, Linkens, & Mahfouf, 2001) :

1. Verbalization or linguistics through interaction with the human operator or domain expert (Kickert, 1979), (Kiszka, Kochanska, & Sliwinska, 1985) and (Matsushima & Sugiyama, 1985).

2. Logic analysis of the input and output data, (R. M. Tong, 1978) and (R. Tong, 1980).
3. Fuzzy implication and reasoning algorithms to identify fuzzy models (Sugeno & Tanaka, 1991).
4. Identification and self-learning algorithms fuzzy modelling of (multi-input / single-output) MISO systems (Xu & Lu, 1987).
5. Learning signals to create a rule-base (Van Der Rhee, van Nauta Lemke, Hans R, & Dijkman, 1990).
6. Self-organizing fuzzy modal algorithm to model the system via on-line input and output data (Moore & Harris, 1992).

In order to increase the efficiency of the fuzzy controllers, and cover some of their associated problems such as non-minimum phase processes, knowledge-based fuzzy modelling approach is used to model non-linear systems in general, with particular applications in the electric power network, which faces several problems in connection and loss of large loads from the grid at peak times. A supervisory self-monitors and decision fuzzy logic control (SSMDFLC) structure was developed by (Sallama, Abbod, & Turner, 2012). The system consists of a first level for network monitoring to note any irregular change during normal operation. In the second level, after noting the change, logical analyses and comparison are performed through which a fault can be identified. In the third level, which was trained to perform the supervisory control, the appropriate decision is mode depending on the fault type.

6.2.1 Hierarchical Supervisory Control

Supervisory control involves tracking and focusing on specific information about the process and controller, monitoring the system and detecting the controller performance when disturbances or change occur in the controlled system in order to maintain the basic requirements such as stability and select the appropriate tuning of the controller, as suited to the specific situation. The system consists of a hierarchy construction of three levels, as shown in Figure 6.1. The highest level of supervisory control performs all the decision-making, observing any failures in the system and diagnosing the fault type, then making an appropriate decision. The observer level is an interface between the different levels of the control system and the high level. The lower level works to adjust for level control. For a simple closed circuit using any

control device parameters in any system, certain behaviours are acceptable while others are not.

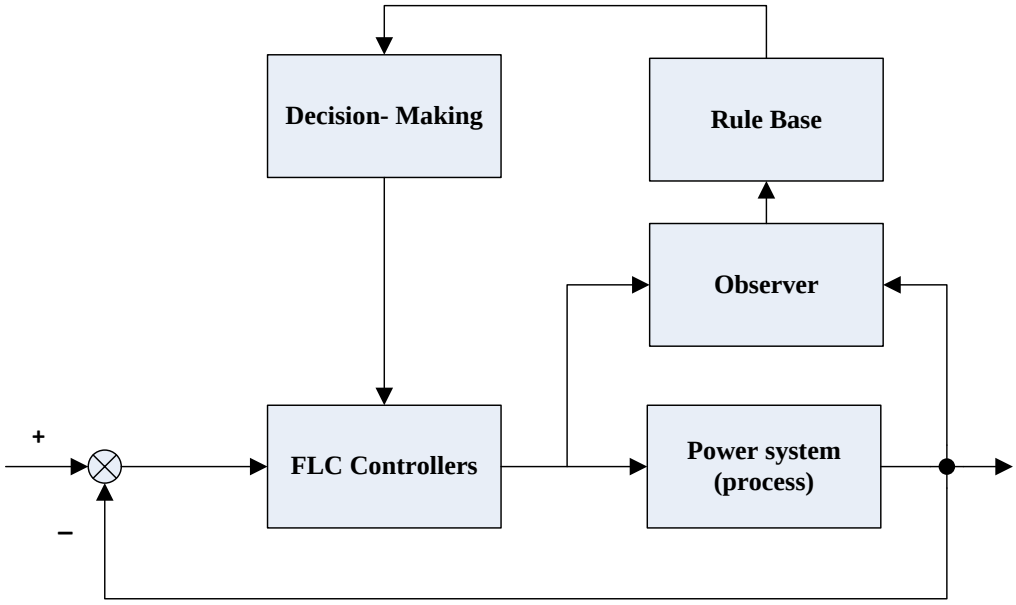

Figure 6.1: Supervisory block diagram.

Acceptable performance in normal operating conditions may include distorted signals due to noise. Unwanted behaviours are caused by changes in the physical structure of the system. Certain types of undesired behaviours can be simple changes in process parameters, which can arise from the interference of actuators, sensors, or the internal structure of the system. It is also possible that behavioural patterns resulting from large load change in the power system network lead to change in the system parameters where are not explicitly identified or observed. Another problem associated with the defective instruments can be detected by comparing the output signal of the system itself with the reference system. It is now possible to define two types of unacceptable behaviours, dysfunction and erratic behaviour. Dysfunction operation is caused by changes in process parameters and is corrected by the controller, which processes change in the control parameter in order to overcome the fault. However, the erratic behaviour must be diagnosed to find the defective part then corrected by the primary or adaptive control, in order to get acceptable behaviour of the system as a result of those actions. The supervisory level must also have an alarm-handing facility, so that when a fault occurs and is detected firstly an alarm issued.

6.2.2 Optimisation of the Controller

There are two main problems to be tackled in the optimiser parameters Optimisation program: one is data coding, namely population expression, the other is how to choose the fitness function. Since the most original Optimisation algorithm is designed for real-value problems, so the controller parameters optimize problem is easily solved. As for choosing the fitness function, in the case of PID controller parameter several fitness functions have been widely used, chosen according to the project demand. The Optimisation criterion for PID controller is often obtained by the following equation 6.1:

$$J = \int_0^\infty [w_1|e(t)|]dt + w_2 t_s \qquad (6.1)$$

where *e(t)* is system error, t_s is the settling time and w_1,w_2 are coefficient parameters, which are very important for the optimiser performance (Y. Wang, Peng, & Wei, 2008).

6.2.3 Fault Detection and Diagnosis

The main problem in process engineering is the abnormal event management (AEM). Fault detection and diagnosis (FDD) is the key element to predict the abnormal event, thus it has attracted a lot of research attention. FDD deals with timely fault detection and correction of abnormal conditions of faults during the running process after diagnosis. The early detection and diagnosis of faults during the process, while the plant is still running in controllable region, can help to prevent the aggravation of abnormal events and significantly reduce losses in productivity. There is considerable interest in this field now from industrial practitioners as well as academic researchers; the petrochemical industries lose an estimated $20 billion every year due to mismanagement of abnormal events, therefore they have rated AEM as their a number one problem that needs to be solved (Venkatasubramanian, Rengaswamy, Yin, & Kavuri, 2003) .

6.3 System Developments

For the purpose of testing the efficiency of the new designed system, it was upgraded to a larger electrical network grid using Matla/Simulink Power System Toolbox.

6.3.1 Scaled-up Network Power System (Four Generators)

Figure 6.2 shows single line diagram of the scaled-up power system, which includes four generators (M1, M2, M3 and M4); machines 1 and 3 with a capacity of 1000 MW, and machines 2 and 4 with a capacity of 5000 MW. All of the generators are turbine driven. The four generators are mutually connected to the network via high voltage the transformer, bas bar and transposed line with a length of 1,500 km together, to serve peak demand load with capacity 11,000 MW.

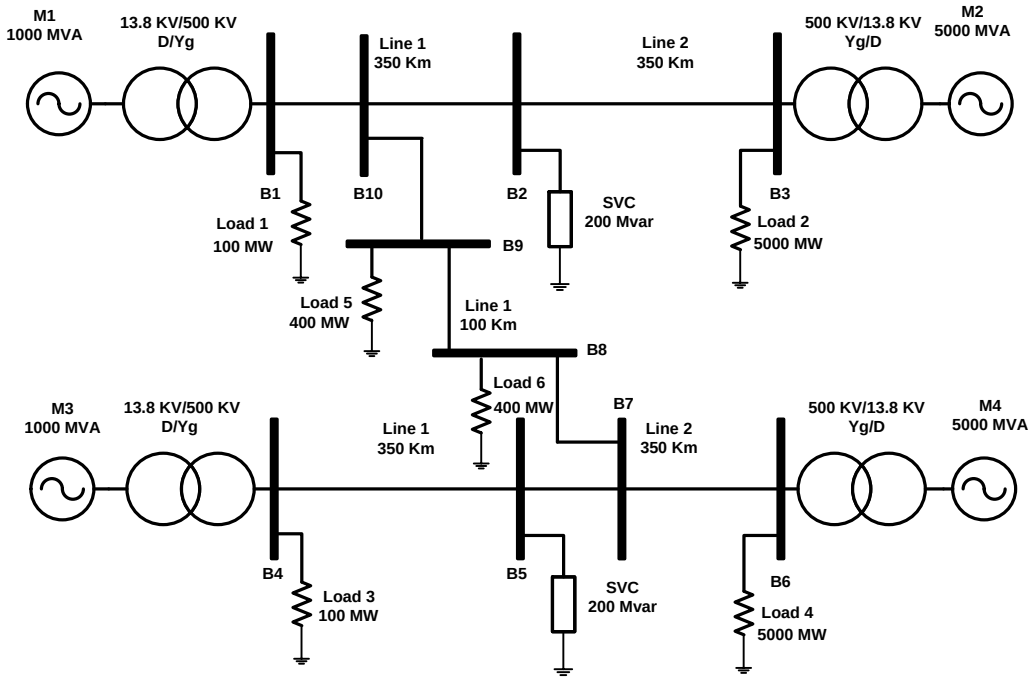

Figure 6.2: Single line diagram of the scaled-up power system.

6.3.2 Controller Auto-Tuning

Details of FLC tuning using the Adaptive Neuro-Fuzzy Inference System (ANFIS) were discussed in the previous chapter. The main target of this phase is to use the SPSO optimiser to tune the scaling factors of all controllers (FLC1, FLC2, FLC3 and FLC4) which are working in the process at the same time. The aim is to improve the

response of all controllers, keeping in mind the main objectives, which are dependent on the type and requirements of the system to be controlled. For the power system, the objective was to minimize the following four variables on the final output (Vm) after the Static Var Compensator (SVC) of whole system:

- Minimize the settling time.
- Minimize the steady state error.
- Minimize the overshoot
- Minimize the first negative peak

The proposed FPSS has two inputs and a single output, which means three scaling factors are considered: the error (Ke), change of error (Kc) and output (Ko) for each controller, where (Ke) is the error gain, (Kc) is the gain for the change of error and (Ko) is the output gain. To improve the system response and for the purposes of comparison, the FLCs scaling factors were tuned in two way: manually and auto. The best values established for the scaling factors are shown in Table 6.1.

Table 6.1: Final auto-tuning scaling factor values.

	FLC1			FLC2		
	Ke1	Kc1	Ko1	Ke2	Kc2	Ko2
Manual Tuning	2	3.75	2.25	5	3.75	10
Auto-Tuning	1.100	1.948	6.559	2.096	27.57	5.666
	FLC3			FLC4		
Manual Tuning	2	3.75	2.25	5	3.75	10
Auto-Tuning	1.139	0.941	9.215	1.556	6.006	5.543

6.3.3 Fault Detection System

As explained earlier, the FDD deals with the timely fault detection and correction of abnormal conditions of faults during the running process. The flowchart shown in Figure 6.3 illustrates the process of fault diagnosis for the power system network.

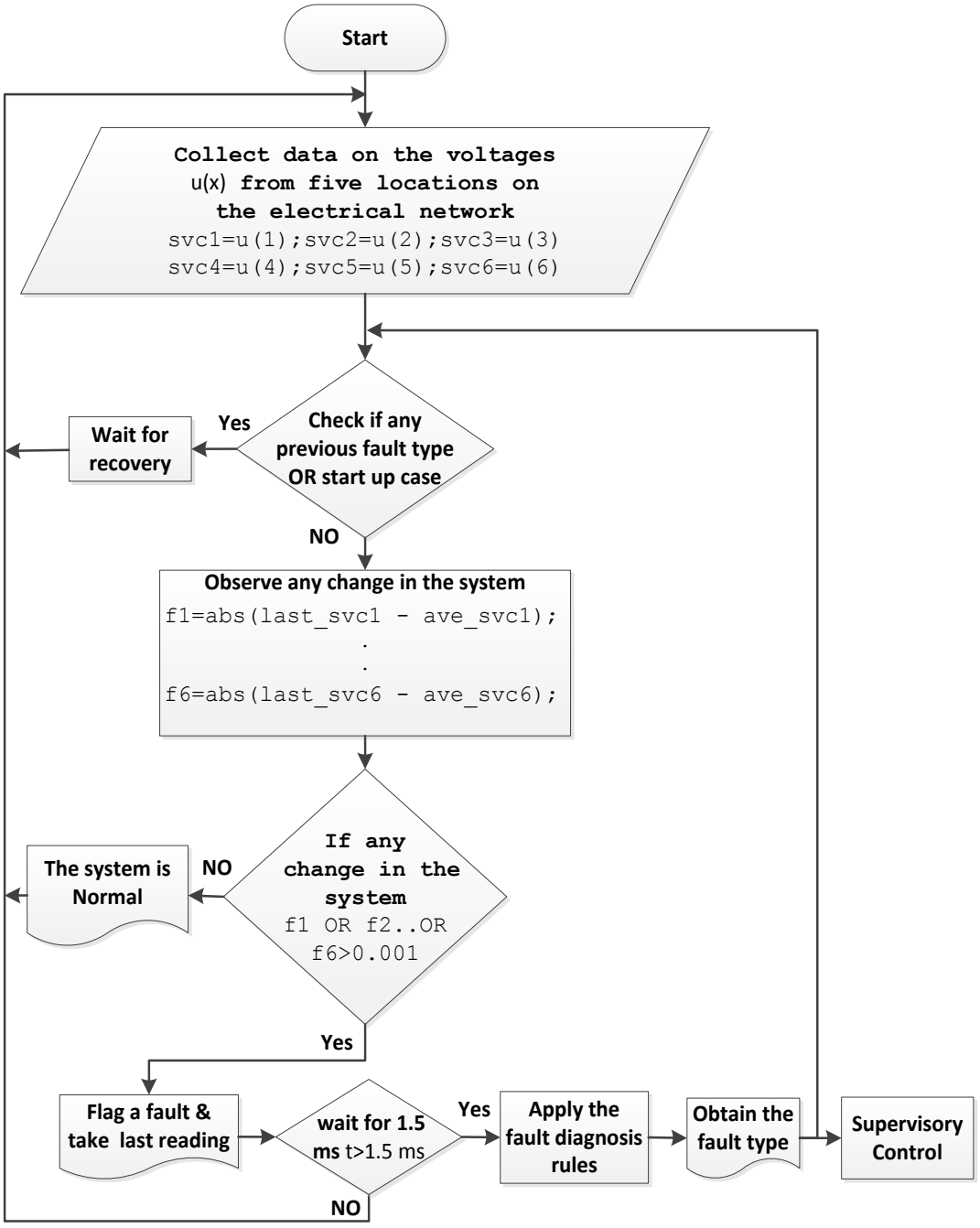

Figure 6.3: Flowchart of fault diagnosis for power system network.

There are four main stages in the fault diagnosis of power system network in the flowchart. The first step is collecting the data about the voltage in different locations in the electrical network in very short time while the process is running. After checking for any previous faults or not in start-up condition, all necessary measurement and comparisons are conducted in the second stage to identify any abrupt changes in the network. If sudden changes are recognised at any part of the electrical network, the programme will trigger the alarm with fault flag and wait for

1.5 ms. Subsequently the diagnosis programme will send all the information to the third stage, which contains the fault diagnosis rules through which the fault type is detected. The last stage is to send the fault type to the first step and all the data from diagnosis programme to the supervisory control.

6.3.4 Fault Diagnosis

As pointed out in section 6.2.3 in the beginning of this chapter, AEM is one of the main problems facing processes engineering. The first FDD step to address this problem in this project was that by monitoring network voltage in several different places. The output voltage (SVC1, SVC2, …, SVC6) as points in different places on the electrical network were monitored throughout. Any very low change $f_x > 0.001$ at any point triggered where f_x at each point is detected using the following equation:

$$f_x = \left| SVCx_n - \frac{SVCx_n + SVCx_{n-1}}{2} \right| \tag{6.2}$$

where n is the number of iteration and the $x = (1, 2, 3\ldots.., 6)$

After a certain time (a few milliseconds) the value of the change at every point from the moment the fault occurred is calculated by:

$$SVCx_{Def} = SVCx_n - SVCx_{fault} \tag{6.3}$$

where n is the number of iteration and the $x = (1, 2, 3\ldots.., 6)$ and the $SVCx_{fault}$ is voltage value at the moment of fault occurring in each point.

The rules listed in Table 6.2 are developed experimentally based on the behaviour of system response to individual failure, such that the type of fault can be diagnosed based on the activated rules.

Table 6.2: Fault diagnosis rule basis values.

If condition	$SVC1_{Def}$	$SVC2_{Def}$	$SVC3_{Def}$	$SVC4_{Def}$	$SVC5_{Def}$	$SVC6_{Def}$
Fault type 1	≤ - 0.004	≥ 0.003	≤ - 0.003	≥ 0.0015	≥ 0.004	≤ - 0.0015
Fault type 2	≥ 0.0004	≤ 0.002	≤ 0.002	≤ - 0.0007	≥ 0.001	≤ 0.00075
Fault type 3	≥ 0.0005	≥ 0.0025	≤ - 0.001	≤ - 0.002	≥ 0.004	≤ - 0.002
Fault type 4	≥ 0.019	≥ 0.04	≤ 0.015	≥ 0.025	≥ 0.04	≥ 0.01
Fault type 5	≤ 0.024	≤ 0.03	≥ 0.010	≥ 0.012	≥ 0.012	≥ 0.012
Fault type 6	≤ 0.002	≥ 0.00049	≥ 0.001	≤ 0.0003	≤ 0.0055	≥ 0.0015
Fault type 7	≤ 0.0135	≤ 0.022	≥ 0.0145	≥ 0.017	≥ 0.022	≥ 0.0175

6.4 Simulation Results

The system was simulated and tested for different fault conditions, namely single and multi-phase conditions, and the FLCs were trained on data generated using three-phase fault conditions, dubbed 3-phase training. Four FLCs controllers were trained for the generator. The scaling factors were tuned automatically for each controller using the SPSO optimiser.

6.4.1 Scaled-up Model

To test the efficiency of the newly designed system, it is upgraded to a higher electrical network grid using Matlab/Simulink Power System Toolbox, as shown in Figure 6.4. The model details are given in section 6.3.1.

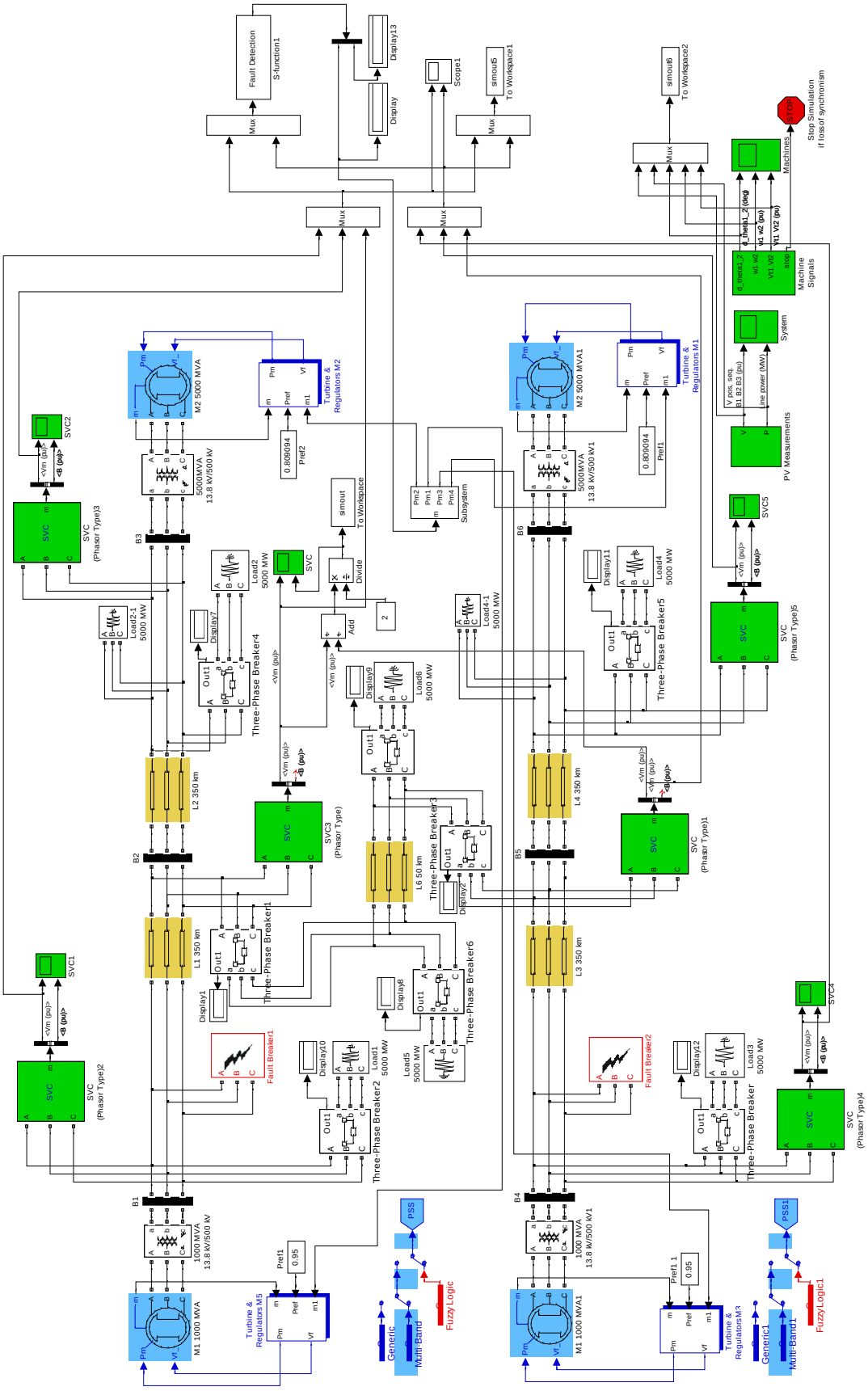

Figure 6.4: Matlab/Simulink Block diagram of the scaled-up power system.

6.4.2 Normal Case Simulation

As noted in the previous chapters, the controller was tuned using SPSO to optimise the selection of the scaling factors for several cases of faults that are expected to occur during normal operation. The supervisory controller can apprehend the largest possible number of faults chosen, tune all control devices for each individual case and keep all obtained results in the database. The optimiser objective function is based on four objectives: steady state error, settling time, overshoot and the negative peak. The first and second objectives have the highest priority, the third has medium priority, whereas the fourth objective has the least priority. The optimised values of the scaling factors for both the auto-tuned fuzzy logic controller and the manually tuned were listed in Table 6.1, and used to compare the response of all electrical network stability driven by supervisory controller, MB stabiliser and without power system stabiliser respectively, during normal operation as shown in Figure 6.5.

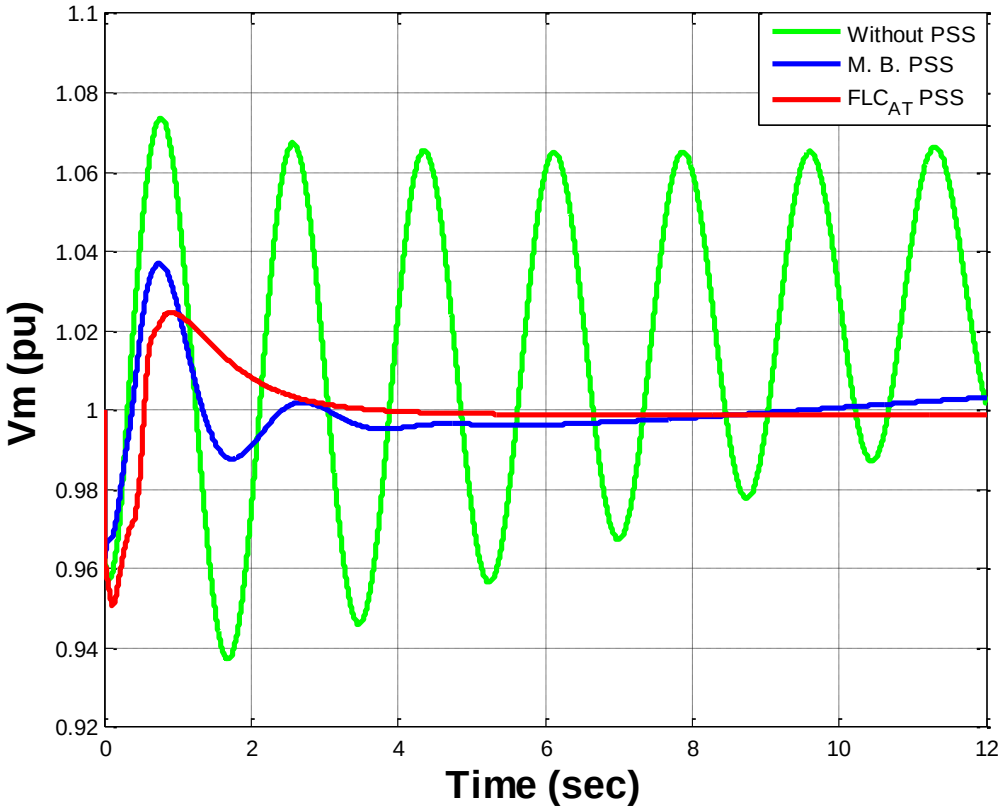

Figure 6.5: System response to normal operation with 3-phase training and auto-tuning.

6.4.3 Multi-Phase Fault

To test the response to major interruptions, such as one, two and three-phase fault, with two fault circuit breakers were connected in the network. One of the circuit breaker closes after 4.8 sec of the start of the simulation for a transition time period of 0.1 sec. Four neuro fuzzy logic controllers were used to stabilise each turbine. The ANFIS replace the conventional MB stabiliser to improve stability, and the system was tested and simulated for different fault conditions, namely single and multi-phase conditions. Simulation results show that supervisory controllers performed well for single-phase fault, two and three-phase fault in comparison to the performance when the system was driven with MB and without PSS. Figure 6.6 shows the response of the system to the faults when was without PSS, driven by MB stabiliser and auto-tuned FLCs respectively, where the response of the FLCs controller under supervisory with 3-phase training and auto-tuning scaling factors to four controllers at the same time. The first test was conducted for one phase fault, and assumes that the settling time of the system at ±1.2 %.

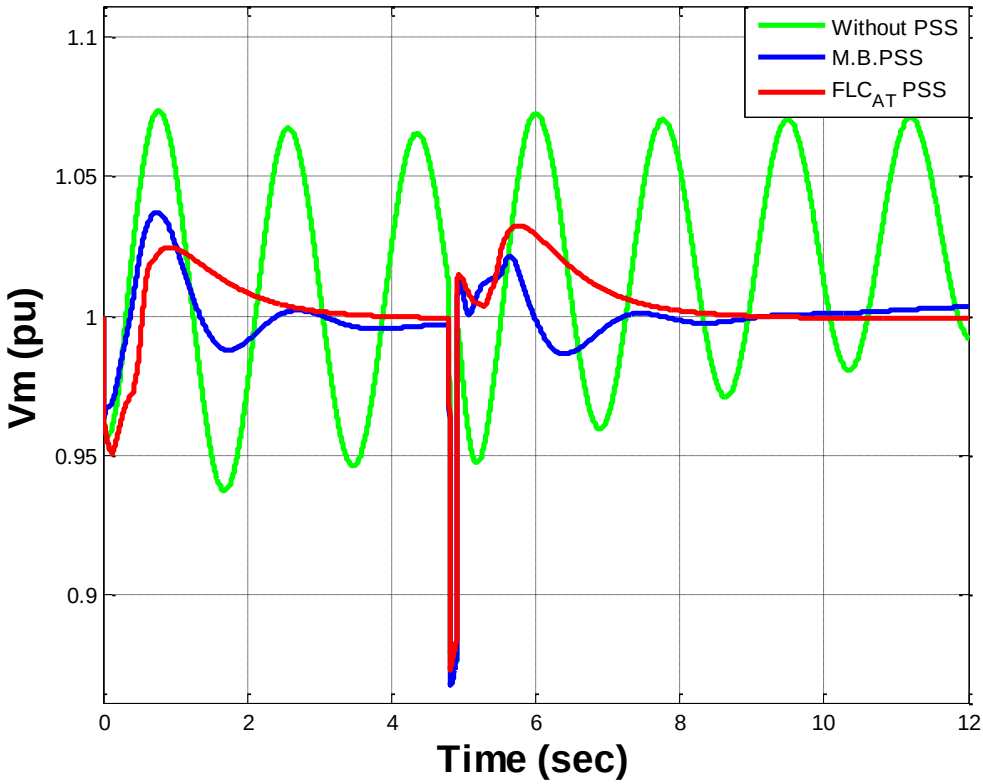

Figure 6.6: System's response to 1-phase fault with 3-phase training and auto-tuning.

The fault starts at 4.8 sec and ends at 4.9 sec; the FLC controllers under supervisory function reduced the settling time from 3.77 sec to 2.5 sec with two peak fluctuations

compared to the MB controllers with six peaks and larger overshoot (with FLCs of 3.2% and 3.7% respectively). Figure 6.7 shows that the FLCs reduced the settling time from 4.03 sec to 2.43 sec, and the oscillation around the set point from six to two compared to MB controllers. More importantly, when a three-phase fault is simulated, as shown in Figure 6.8, the FLCs controllers under supervisory reacted with high efficiency compared to the MB and reduced the overshoot from 4.9% to 4.2% and settling time period from 3.41 sec to 2.5 sec with respect to the MB controller, and over a longer period increasing steady state error appeared. The numerical results are shown in Table 6.3.

Table 6.3: Simulation results for multi fault with auto-tune and M-training.

	Multi Band Controller (MB)			Fuzzy Logic Controller (FLC_{AT})		
	Settling time (sec)	Overshoot (%)	Fluctuation	Settling time (sec)	Overshoot (%)	Fluctuation
1 phase fault	3.77	3.7	6	2.5	3.2	2
2 phase fault	4.03	3.7	6	2.43	4.4	2
3 phase fault	3.41	4.9	6	2.5	4.2	2

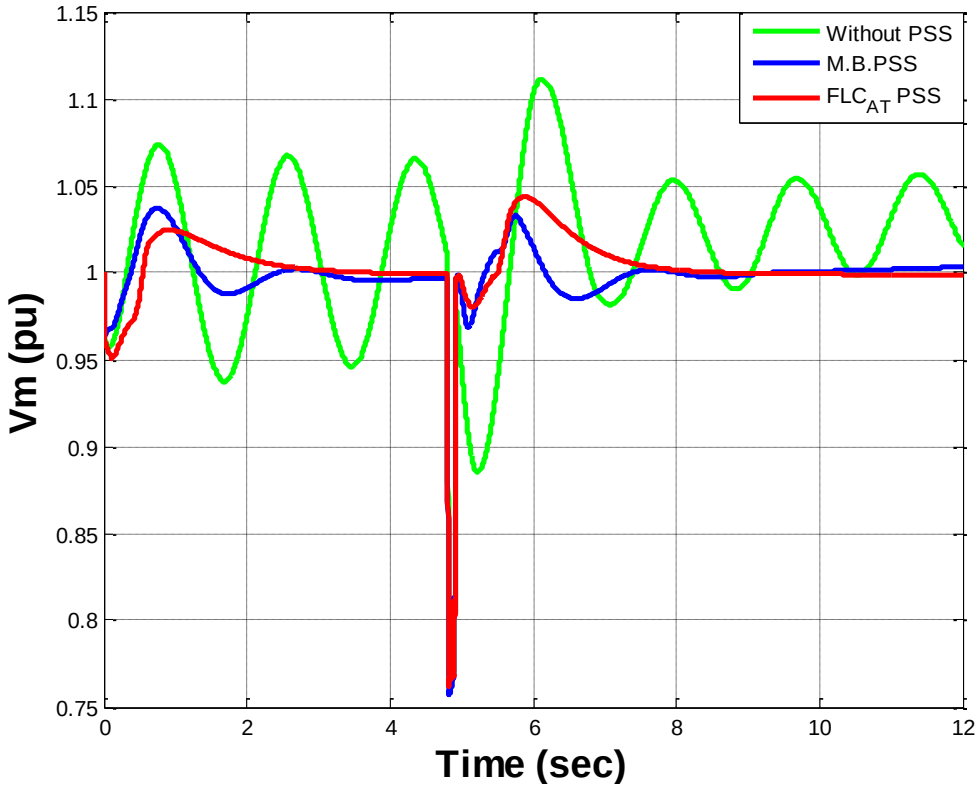

Figure 6.7: System's response to 2-phase fault with 3-phase training and auto-tuning.

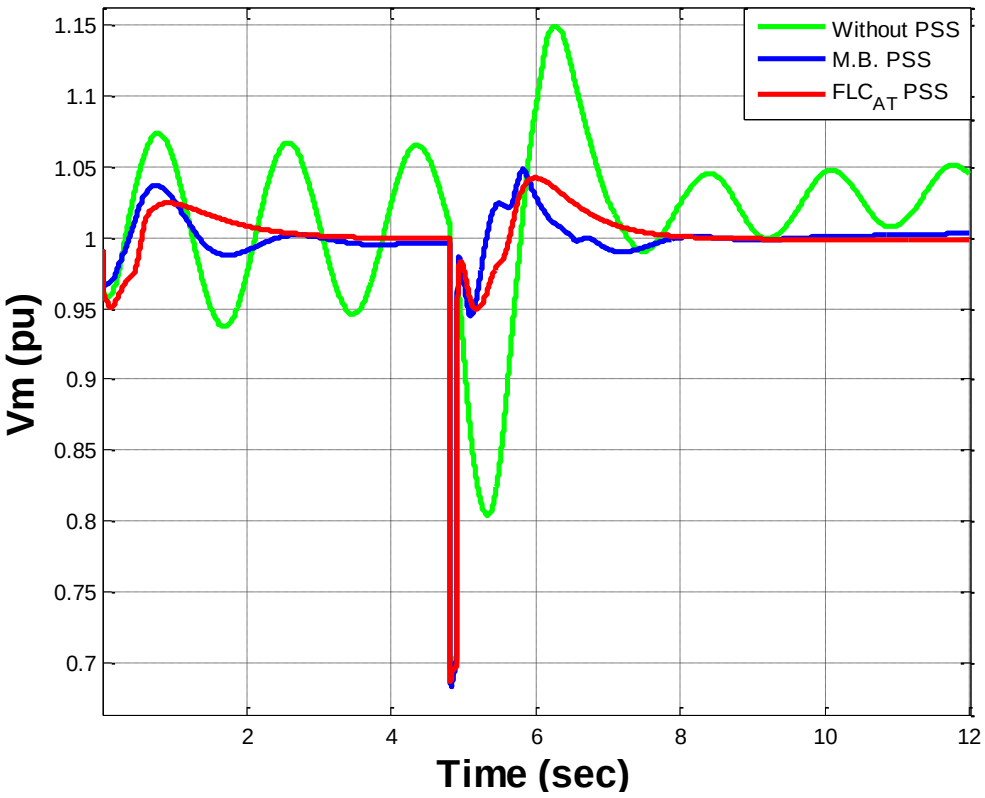

Figure 6.8: System's response to 3-phase fault with 3-phase training and auto-tuning.

6.4.4 Fault Scenarios Simulation

In order to test the supervisory control system more accurately, a scenario was applied on the system process, including connecting and losing a large load at seven different places from the electrical network in sequence. The behaviours of all the systems are compared to its behaviours when governed by the other advanced conventional control systems.

A. Normal Operation Scenario

In this case the system was scaled-up and working with full capacity in normal condition with four machines (M1, M2, M3 and M4), with a generation capacity of 12,000 MW to supply load with demand 11,000 MW, as shown in Figure 6.9.

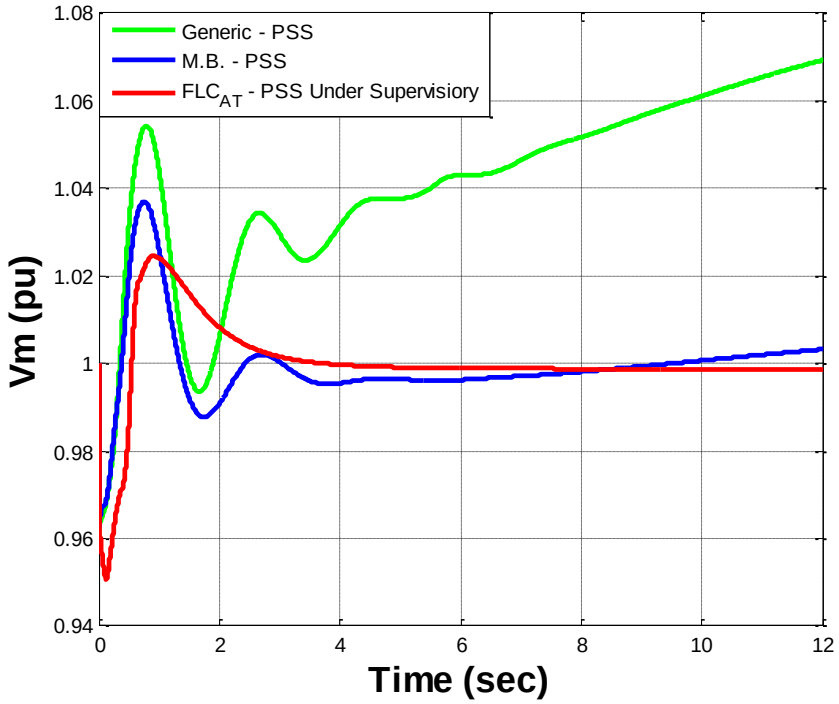

Figure 6.9: System's response to normal operation with comparison supervisory control, MB and Generic PSS.

B. Fault Scenario1

In this scenario, lost Load 1 at B1 is as shown in Figure 6.2 above with capacity of load demand 100 MW at second 4, and the same load back at second 8. The system response is illustrated in Figure 6.10.

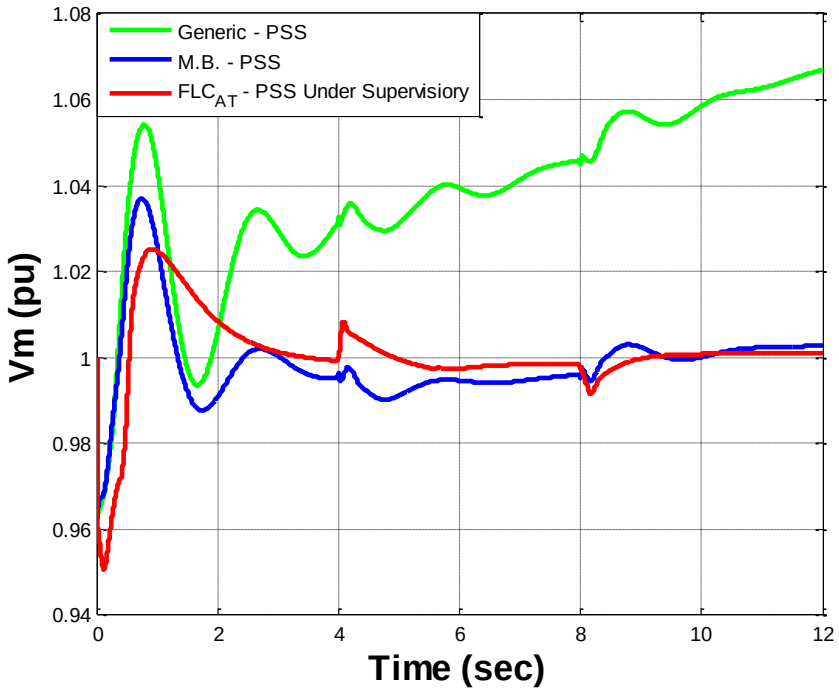

Figure 6.10: System's response to fault 1 with comparison supervisory control, MB and Generic PSS.

C. Fault Scenario 2

In this scenario, lost part of Load 2 at B3 is as shown in Figure 6.2 above with capacity of load demand 75 MW at second 4 and the same load back at second 8. The system response is illustrated in Figure 6.11.

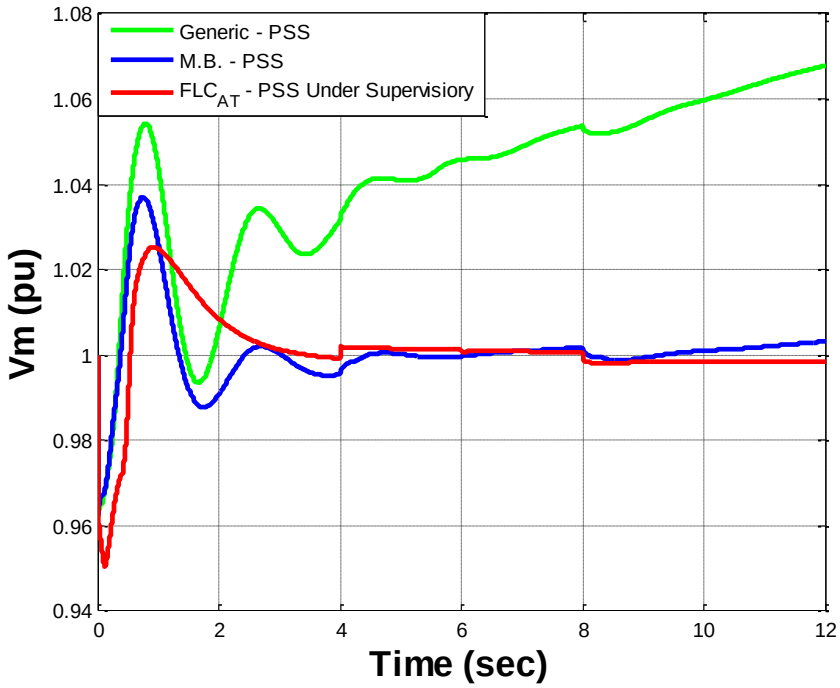

Figure 6.11: System's response to fault 2 with comparison supervisory control, MB and Generic PSS.

D. Fault Scenario 3

In this scenario, lost Load 3 at B4 is shown in Figure 6.2 above with capacity of load demand 100 MW at second 4 and the same load back at second 8. The system response is illustrated in Figure 6.12.

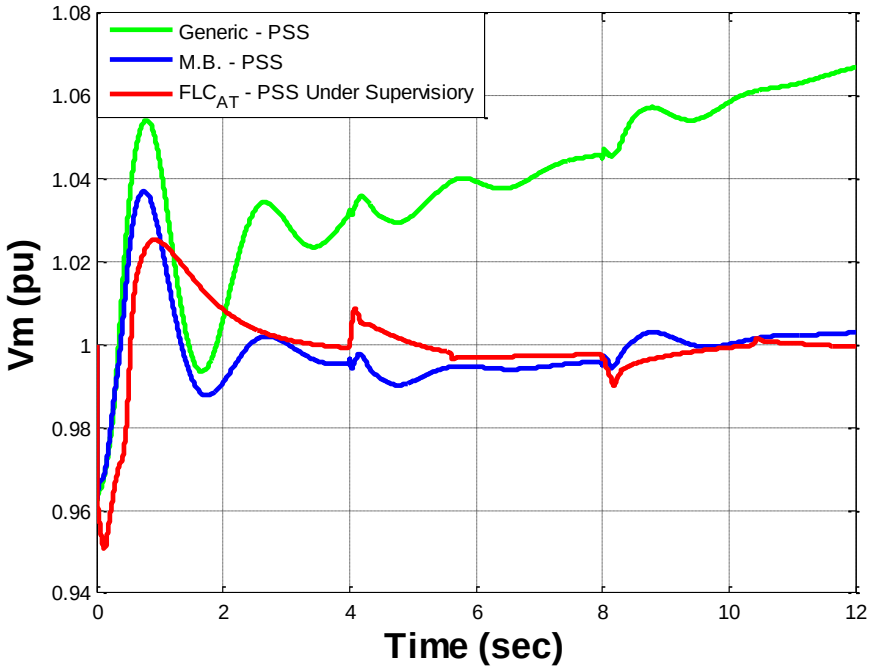

Figure 6.12: System's response to fault 3 with comparison supervisory control, MB and Generic PSS.

E. Fault Scenario 4

In this scenario, the system lost part of load 4 at B6 as shown in Figure 6.2 above with capacity of load demand 75 MW at second 4 and the same load back at second 8. The system response is illustrated in Figure 6.13.

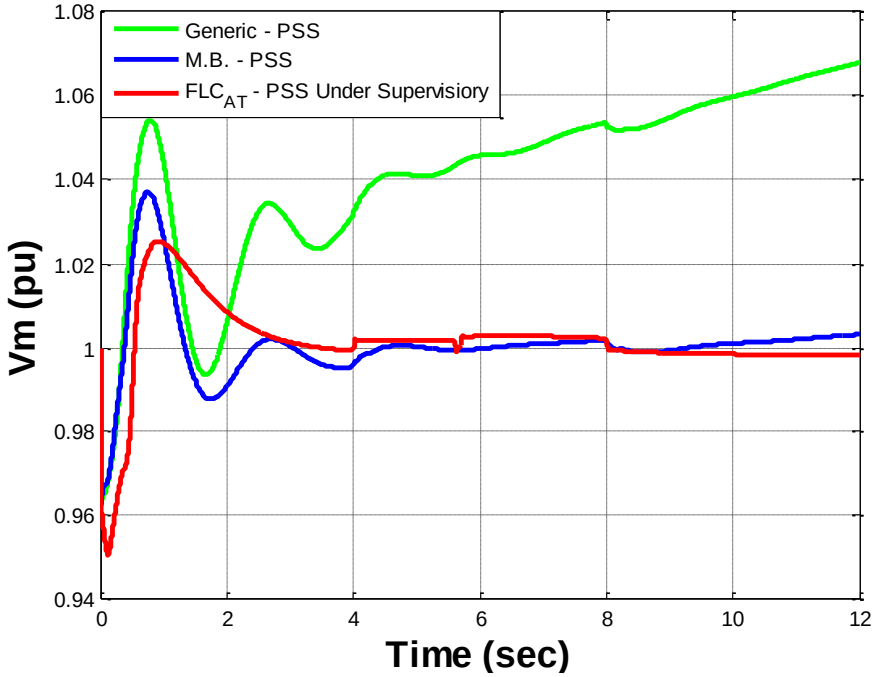

Figure 6.13: System's response to fault 4 with comparison supervisory control, MB and Generic PSS.

F. Fault Scenario 5

In this scenario, lost load 5 at B2 as shown previously in Figure 6.2 above with capacity of very high load demand 400 MW at second 4 and the same load back at second 8. The system response is illustrated in Figure 6.14.

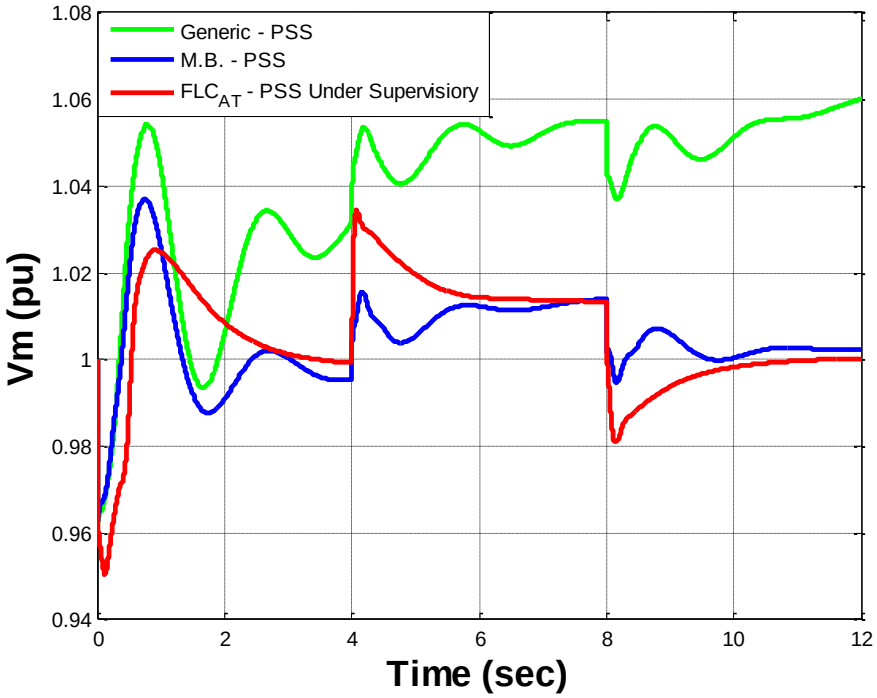

Figure 6.14: System's response to fault 5 with comparison supervisory control, MB and Generic PSS.

G. Fault Scenario 6

In this scenario, lost load 6 at B5 as shown in Figure 6.2 above with capacity of very high load demand 400 MW at second 4 and the same load back at second 8. The system response is illustrated in Figure 6.15.

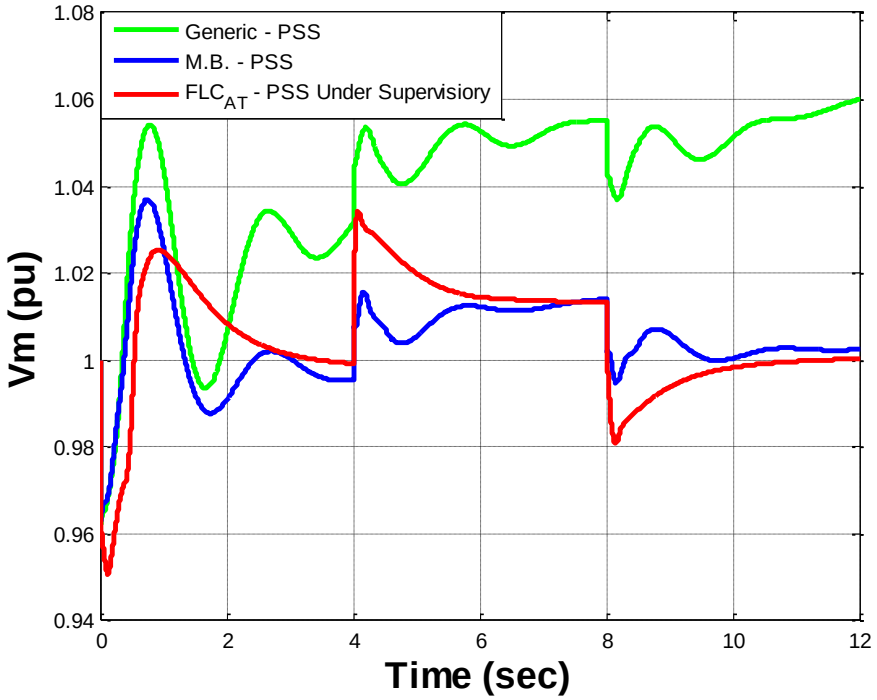

Figure 6.15: System's response to fault 6 with comparison supervisory control, MB and Generic PSS.

H. Fault Scenario 7

In this scenario, lost load 6 and load 5 at B5 and B6 as shown previously in Figure 6.2 above with capacity of very high load demand 800 MW. At second 4 a cross connection link was disconnected, and then went back to normal at second 8. The system response is illustrated in Figure 6.16.

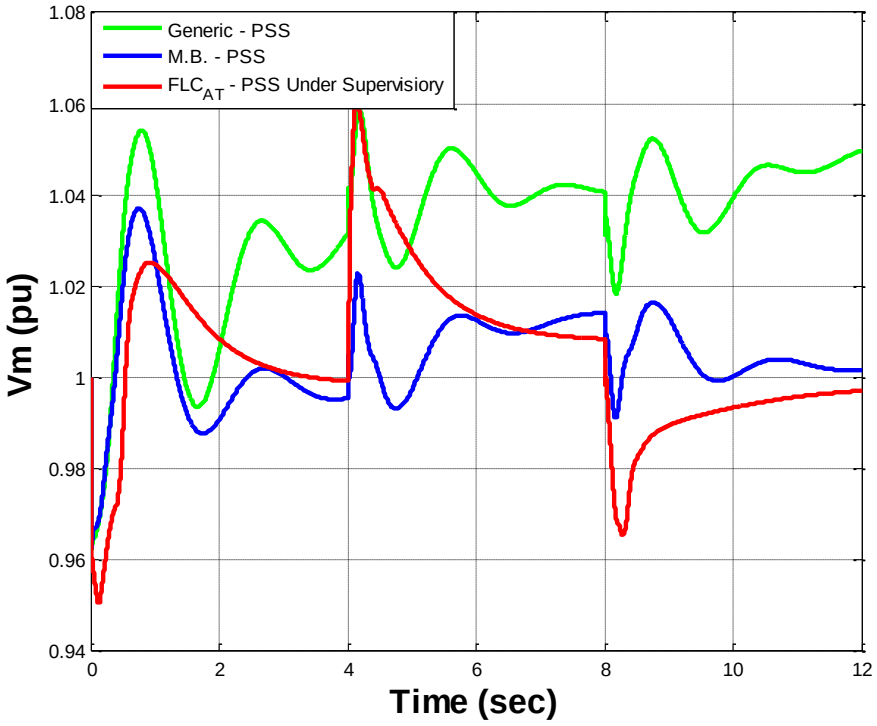

Figure 6.16: System's response to fault 7 with comparison supervisory control, MB and Generic PSS.

6.4.5 Simulation of Consecutive Serious Fault

Figure 6.17 illustrates the comparison between the latest conventional stabiliser controller and the supervisory power stability control system for different faults. From the seven different types of faults during 30 second duration, it can be seen that directly after the start-up notes the highest speed of response was observable, with less overshoot, and very clear stability in the whole system. The second fault accrued after start up in the scenario simulated a loss of 75 MW load at second 5 then the load returned back to normal at second 8; at second 10, a cross connection link was disconnected, and then went back to normal at second 15; at the 20th second a high load loss occurred with 400 MW capacity then returned to normal at second 25. The figure shows that the supervisory controller reaction is much smoother, stable, faster in response and less steady-state error-prone than other competitive controllers. Table 6.4 shows the auto-tuned value of scale of factors for all low level controllers, which are reset it by the supervisory controller automatically concerning to the fault type.

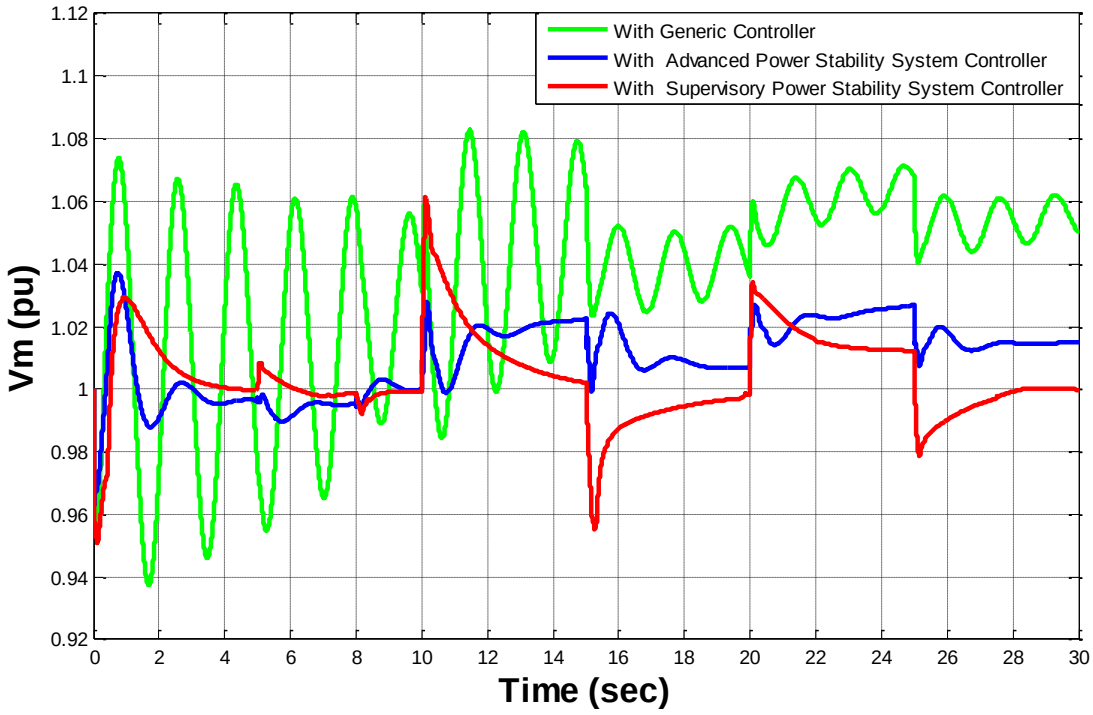

Figure 6.17: Different fault types.

6.5 Summary

To conclude, it was shown that the SPSSC achieved better performance during the occurrence of sudden faults, and the possibilities to return to the stability status of the network safely and with smooth running within the allowable time limits, whereby the osculation did not exceed the high limit of ±0.6 % range. This is clearly shown in Figure 6.17 when simulating consecutive serious faults scenarios; the system responded with high efficiency, compared to the other two types of conventional power stabiliser controllers, in particular when the SPSSC was tuned with the aid of SPSO to select the scaling factors for multi controller automatically and simultaneously for each fault individually. In addition, the SPSSC system demonstrated superior performance compared to the other two controller devices, even in the case of normal operation, as it eliminated steady state error almost completely, which means that the SPSSC system has better functionality than other standard stabilisers during normal and fault conditions.

Table 6.4: Final auto-tuning scaling factor values for all controller in deferent scenario.

	FLC1			FLC2		
	Ke1	Kc1	Ko1	Ke2	Kc2	Ko2
Normal Operation	1.100	1.950	7.100	1.990	27.200	5.360
Load 1 with 100 MW	1.150	1.948	4.959	2.096	23.577	5.666
Load 2 with 75 MW	1.782	1.946	7.808	1.899	5.514	6.552
Load 5 with 400 MW	1.095	1.940	7.911	2.052	32.137	5.630
Network separation	1.346	2.274	11.00	8.722	5.857	5.797
Load 3 with 100 MW	1.139	0.941	9.215	1.556	2.006	5.543
Load 4 with 75 MW	1.7805	1.9948	7.8161	1.8986	5.5144	6.5832
Load 6 with 400 MW	1.0906	0.8479	8.5644	1.5054	11.9482	5.5334
	FLC3			FLC4		
	Ke3	Kc3	Ko3	Ke4	Kc4	Ko4
Normal Operation	1.150	1.100	8.810	1.700	5.500	5.550
Load 1 with 100 MW	1.139	0.941	9.215	1.556	2.006	5.543
Load 2 with 75 MW	1.780	1.994	7.816	1.898	5.514	6.583
Load 5 with 400 MW	1.090	0.847	8.564	1.505	11.948	5.534
Network separation	1.104	1.317	18.156	15.682	6.596	21.442
Load 3 with 100 MW	1.150	1.948	4.959	2.096	23.577	5.666
Load 4 with 75 MW	1.782	1.946	7.808	1.899	5.514	6.552
Load 6 with 400 MW	1.095	1.940	7.911	2.052	32.137	5.630

Chapter 7: Conclusions and Future Work

7.1 Conclusions

This chapter presents the main conclusions of the thesis and identifies areas for future work. The research mechanism that has been presented in the earlier chapters is summed up and concluded along with the overall accomplishment to achieve the aims and objectives of the research. After that, all the probable modifications which could give suitable or knowledgeable benefit to the performance of the presented methodologies that are used for this work are discussed in detail in terms of future work.

This research focused on the integration problem of the intelligent search technique for optimum and the stability control in power system using artificial intelligence techniques. The literature review suggested that electrical power network system model is classified mostly as non-linear Optimisation problems that have at least one optimum solution. Though the analysis may be influenced by slow convergence and the curse of multi dimension level, heuristics-based swarm intelligence can be a very effective substitute PSO, which is part of the family of swarm intelligence is known widely to be effective to solve nonlinear Optimisation problems as an artificial intelligence search technique. The principles of PSO and its variants are very useful for analysis, and the application of PSO as an Optimisation technique on the power system has been powerfully effective. The majority of studies found that major improvement is achieved using artificial intelligence techniques. Furthermore, the Optimisation methodology leading the electrical network stability problems has played a main role in improving the performance of power quality by using multi particle swarm Optimisation as artificial intelligent to solve this problem.

Several types of PSOs algorithms are developed, such as LPSO, GPSO, DMS-PSO-SHS, APSO and others. Moreover, an effective paradigm has been developed using all previews algorithms based on multi swarm methodology in two patterns, parallel and serial to solve stability problems that are classified as complex problems. In fact, this complexity is basically caused by the extent growth of electrical grid with diversity components and some other phenomena such as wave, electromagnetic, electromechanical and thermodynamic phenomena.

Different types of benchmark function are presented in detail in terms of implementing and evaluating Optimisation using existing algorithms and comparing their performance with that of the new algorithm. According to the results (chapter three), the proposed SPSO achieved superior performance to PPSO. The comparisons were made by observing the response of each algorithm on all used benchmark functions in terms of convergence speed and performance to reach the closest correct solution.

Electricity network stability has been explained, with detailed data about the latest conventional PSS control devices currently used to improve the stability of the electrical network as well as the relationship between reactive power and voltage stability, thereby providing a model with full components to simulate electrical power grid using Matlab/Simulink. Tests were conducted and comparisons were made between the currently used PSS devices in terms of efficiency and the ability to bring back the network in equilibrium state after disturbance as a result of fault occurrence or sudden change in operating conditions. Figures 4.9-4.11 in chapter four illustrate the positive impact of the PSS devices on the electrical network systems, whereby the effect of the generic power system stabiliser PSS2B and multiband power system stabiliser PSS4B are convergent in terms of the first overshoot, but the PSS4B still works on damping the electrical network system more effectively than PSS2B, as well as performing its function with higher efficiency than system stability working without power system stabiliser PSSs in the three case scenarios, after a malfunction in the system (i.e. making short circuits between one phase and ground, then between two phases and the ground, then between three phases and the ground, sequentially). As seen in the three figures, PSS4B works with high efficiency on over-shooting and damping of the system oscillation compared with PSS2B, which was working efficiently but slightly less than the PSS4B and the system becomes completely unstable and oscillation grows after any type of faults when it works without power system stability devices.

The proposed fuzzy logic control system FLC is used to control the power system stability (FPSS), thus the control scale of factor was selected and different parameters in the rule base and membership functions. Where the generator speed deviation is classified into [positive (w_p); zero (w_z); negative (w_n)], the generator speed deviation change is classified into: {positive (dw_p); zero (dw_z);

negative (dw_n)}. Also the output of fuzzy controller is classified into: {positive (u_p); zero (u_z); negative (u_n)}.

To improve the FLCs response, ANFIS methodology for membership function is used to train them on the behaviour of conventional MB-PSS when facing the three phase fault as worst case of failure. The FLCs scaling factors are tuned manually, than automatically using the SPSO as new optimiser algorithm. Several comparisons in power quality voltages control and rotor speed deviation were conducted on the electrical network, including two generators governed by the new stabiliser, and the results in chapter five indicated the superiority of new proposed PSS, particularly on high capacity generator M2, and it is clear that using SPSO as an Optimisation technique gives good results. FLC based on SPSO obtains the advantages of FLC (as fuzzy logic technique) and the advantages of SPSO as an Optimisation technique, selecting the best values to obtain the best results with minimum error in a fast way. Thus the new controller (FLC based on SPSO) is capable of damping oscillations, increasing system stability with minimum energy consumed in fast way, and it has maximum sensitivity to changes in operating situations.

The new optimiser, as an important tool, is designed to use an advanced supervisory control system in electrical network stability, and for this purpose was scaled up to the electric grid to simulate the natural operating conditions and to increase the complexity of the simulation. The new control system involves tracking and focusing on specific information about the process and controller, monitoring the system and detecting the controller performance when disturbances or change occur in the controlled system in order to maintain the basic requirements (such as stability), and selecting the appropriate tuning of the controller, as suited to the specific situation.

In order to test the capacity and efficiency of the innovative system it is tested on a complex scenario of sequence failures and compared to other PSS systems. The results in chapter six illustrate the efficiency of supervisory controller, which achieved better performance during the occurrence of sudden faults, and the possibilities to return to the stability status of the network safely and with smooth running within the allowable time limits, whereby the osculation did not exceed the high limit

It can be concluded, based on the experimental results and predicted performance models, that using FLC for PSS in electrical grids considerably reduced the effect of variations failures in the electrical grid. This was clearly shown when simulating consecutive different faults scenarios; the proposed system responded with high efficiency compared to the other types of conventional power stabiliser controllers, in particular when the FLC was tuned with the aid of SPSO to select the scaling factors for multi controller automatically and simultaneously for each fault individually.

In part of Optimisation research methodology, multi PSO algorithms and the testing results of this paradigm were applied with all algorithms on the proposed benchmark. The performance of each algorithm was measured individually. This work introduced a new algorithm called SPSO with advantages of fast convergence speed and good Optimisation to find the global optimum solution for the most complex benchmark functions; it clearly performed better than the other proposed algorithm (PPSO). Therefore, from these results it is recommended that Optimisation methodology and power system stability control be deployed in real-life scenarios and that supervisory control be utilized in high throughput computing systems and other manufacturing processes. Much work remains to be done, but the potential benefits are considerable.

7.2 Future Work

The simulations conducted in these experiments identified a number of areas for future research, including improving and expanding some parts of this work, such as scaling up works on the electrical network model and improving the Optimisation algorithm function, neuro fuzzy logic controller model and the supervisory control system model.

7.2.1 Optimisers

Improved Optimisation can be attained by increasing the performance of the optimiser itself. Several techniques can be used in order to develop the optimiser performance, such as using more different and effective Optimisation methodology algorithm. In this thesis, the optimiser introduced rules suggested to tackle twelve parameters at the same time for the purpose of tuning controller scaling factors. It

was also explored whether it is possible to optimise the membership rule functions simultaneously. Some optimising experiments produced some promising results, but this line of enquiry was stopped due to the inability of available computers to fulfil this purpose.

Furthermore, although the each algorithm could find the optimal solutions for most cases, in some cases some algorithms were not successful enough to find the optimal solution, as shown in chapter three. Future work could improve the performance of these algorithms by changing some parameters or including them in other complicated optimisers. Algorithms should be tested individually to identify their particular impacts.

7.2.2 Neuro Fuzzy Logic System Model

This project focused on the design of FLC instead of PSS, measuring the error in speed deviation and change of error signal as input signals. To develop the controller function another variable could be processed from the electrical network as an input to the fuzzy controller, such as voltages, frequency and power factor, by adding them as new membership function rules, to improve the functioning and controlling.

7.2.3 Benchmarks

In this thesis seven benchmarks were selected, described and clearly defined before being used to test the efficiency of the newly developed algorithm and to compare it with others. However, additional algorithms could be used for comparison to broaden the test. Although selecting the 30 dimensions for each benchmark function illustrates the differences in the results between algorithms, this could be increased to 60 to make the system more complex and to increase the difficulty of finding solutions, so the differences in the results would be more obvious.

References

[1] Abbod, M. F., von Keyserlingk, D. G., Linkens, D. A., & Mahfouf, M. (2001). Survey of utilisation of fuzzy technology in medicine and healthcare. *Fuzzy Sets and Systems, 120*(2), 331-349.

[2] Abe, S., Fukunaga, Y., Isono, A., & Kondo, B. (1982). Power system voltage stability. *Power Apparatus and Systems, IEEE Transactions On*, (10), 3830-3840.

[3] Abido, M. (2002). Optimal power flow using particle swarm optimization. *International Journal of Electrical Power & Energy Systems, 24*(7), 563-571.

[4] Ajami, A., Taheri, N., & Younesi, M. (2009). A novel hybrid fuzzy/LQR damping oscillations controller using STATCOM. *Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference On, , 1* 348-352.

[5] Ali, A. T., Tayeb, E. B. M., & Adam, M. K. A.A multi-machine power system stabilizer using fuzzy logic controller. *Editorial Committees, ,* 28.

[6] Altalib, H., & Krause, P. (1976). Dynamic equivalents by combination of reduced order models of system components. *Power Apparatus and Systems, IEEE Transactions On, 95*(5), 1535-1544.

[7] Ansarian, M., Sadeghzadeh, S., Nasrabadi, A., & Shakouri, H. (2005). Optimized power system stabilizer based on artificial neural network structure. *PSC,*

[8] Baer, B., Butler, K., Hiyama, T., Kubokawa, J., Lee, K., Luh, P., . . . Mori, H. (2000). Tutorial on fuzzy logic applications in power systems.

[9] Bakshi, U., & Bakshi, M. (2009). *Generation, transmission and distribution* Technical Publications.

[10] Bevrani, H., & Daneshmand, P. R. (2012). Fuzzy logic-based load-frequency control concerning high penetration of wind turbines. *Systems Journal, IEEE, 6*(1), 173-180.

[11] Bóna, M. (2011). *A walk through combinatorics: An introduction to enumeration and graph theory* World Scientific.

[12] Carlisle, A., & Dozier, G. (2001). An off-the-shelf PSO. *Proceedings of the Workshop on Particle Swarm Optimization, , 1* 1-6.

[13] Clerc, M. (2010). *Particle swarm optimization* John Wiley & Sons.

[14] Coath, G., Al-Dabbagh, M., & Halgamuge, S. (2004). Particle swarm optimisation for reactive power and voltage control with grid-integrated wind farms. *Power Engineering Society General Meeting, 2004. IEEE,* 303-308.

[15] Cutsem, T., & Vournas, C. (2008). Voltage stability of electric power systems.

[16] De Silva, C. W., & MacFarlane, A. G. J. (2007). Knowledge based structure for robotics manipulators. *1988 Proc.IFAC.Workshop on Artificial Intelligence in Real Time Control, ,* 143-148.

[17] Demello, F. P., & Concordia, C. (1969). Concepts of synchronous machine stability as affected by excitation control. *IEEE Transactions on Power Apparatus and Systems, 88*(4), 316-329.

[18] Eberhart, R. C., & Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization. *Evolutionary Programming VII,* 611-616.

[19] Eberhart, R. C., & Shi, Y. (2001a). Particle swarm optimization: Developments, applications and resources. *Evolutionary Computation, 2001. Proceedings of the 2001 Congress On, , 1* 81-86.

[20] Eberhart, R. C., & Shi, Y. (2001b). Particle swarm optimization: Developments, applications and resources. *Evolutionary Computation, 2001. Proceedings of the 2001 Congress On, , 1* 81-86.

[21] Eberhart, R. C., & Shi, Y. (2001c). Particle swarm optimization: Developments, applications and resources. *Evolutionary Computation, 2001. Proceedings of the 2001 Congress On, , 1* 81-86.

[22] El-Dib, A. A., Youssef, H. K., El-Metwally, M., & Osman, Z. (2006). Maximum loadability of power systems using hybrid particle swarm optimization. *Electric Power Systems Research, 76*(6), 485-492.

[23] El-Gallas, A., El-Hawary, M., Sallam, A., & Kalas, A. (2001). Swarm-intelligently trained neural network for power transformer protection. *Electrical and Computer Engineering, 2001. Canadian Conference On, , 1* 265-269.

[24] Emara, H. M., Ammar, M., Bahgat, A., & Dorrah, H. (2003). Stator fault estimation in induction motors using particle swarm optimization. *Electric Machines and Drives Conference, 2003. IEMDC'03. IEEE International, , 3* 1469-1475.

[25] Esquivel, S. C., & Coello Coello, C. A. (2003). On the use of particle swarm optimization with multimodal functions. *Evolutionary Computation, 2003. CEC'03. the 2003 Congress On, , 2* 1130-1136.

[26] Evans, W. R. (1950). Control system synthesis by root locus method. *American Institute of Electrical Engineers, Transactions of The, 69*(1), 66-69.

[27] Feliachi, A., & Yang, X. (1994). Identification and control of inter-area modes. *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference On, , 4* 4061-4066.

[28] Geem, Z. W. (2009). Particle-swarm harmony search for water network design. *Engineering Optimization, 41*(4), 297-311.

[29] Goldberg, D. E. (1990). E.(1989). genetic algorithms in search, optimization and machine learning. *Reading: Addison-Wesley,*

[30] Gomes, C. P., & Selman, B. (2001). Algorithm portfolios. *Artificial Intelligence, 126*(1), 43-62.

[31] Guo, Y., Hill, D. J., & Wang, Y. (2001). Global transient stability and voltage regulation for power systems. *Power Systems, IEEE Transactions On, 16*(4), 678-688.

[32] He, S., Wen, J., Prempain, E., Wu, Q., Fitch, J., & Mann, S. (2004). An improved particle swarm optimization for optimal power flow. *Power System Technology, 2004. PowerCon 2004. 2004 International Conference On, , 2* 1633-1637.

[33] Hirata, N., Ishigame, A., & Nishigaito, H. (2002). Neuro stabilizing control based on lyapunov method for power system. *SICE 2002. Proceedings of the 41st SICE Annual Conference, , 5* 3169-3171.

[34] Huang, C., Huang, C., & Wang, M. (2005). A particle swarm optimization to identifying the ARMAX model for short-term load forecasting. *Power Systems, IEEE Transactions On, 20*(2), 1126-1133.

[35] Inamdar, H., & Hasabe, R. (2009). It based energy management through demand side in the industrial sector. *Control, Automation, Communication and Energy Conservation, 2009. INCACEC 2009. 2009 International Conference On,* 1-7.

[36] Jang, J. (1993a). ANFIS: Adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions On, 23*(3), 665-685.

[37] Jang, J. (1993b). ANFIS: Adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions On, 23*(3), 665-685.

[38] Jang, J., & Sun, C. (1995). Neuro-fuzzy modeling and control. *Proceedings of the IEEE, 83*(3), 378-406.

[39] Jang, J. R., Sun, C., & Mizutani, E. (1997). Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [book review]. *Automatic Control, IEEE Transactions On, 42*(10), 1482-1484.

[40] Jiang, H., Zheng, J., & Chen, L. (2007). A particle swarm algorithm for multi-objective optimization problem. *Chinese Journal of Pattern Recognition and Artificial Intelligence, 20*(5), 606-611.

[41] Kamwa, I., Grondin, R., & Trudel, G. (2005). IEEE PSS2B versus PSS4B: The limits of performance of modern power system stabilizers. *Power Systems, IEEE Transactions On, 20*(2), 903-915.

[42] Kannan, S., Slochanal, S. M. R., & Padhy, N. P. (2005). Application and comparison of metaheuristic techniques to generation expansion planning problem. *Power Systems, IEEE Transactions On, 20*(1), 466-475.

[43] Kassabalidis, I. N., El-Sharkawi, M. A., Marks, R. J., Moulin, L. S., & Alves da Silva, Alexander P. (2002). Dynamic security border identification using enhanced particle swarm optimization. *Power Systems, IEEE Transactions On, 17*(3), 723-729.

[44] Kennedy, J., & Eberhart, R. (1995a). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks, , 4*(2) 1942-1948.

[45] Kennedy, J., Kennedy, J. F., & Eberhart, R. C. (2001). *Swarm intelligence* Morgan Kaufmann.

[46] Kennedy, J., & Eberhart, R. (1995b). Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference On, , 4* 1942-1948 vol.4. doi:10.1109/ICNN.1995.488968

[47] Kickert, W. J. (1979). Towards an analysis of linguistic modelling. *Fuzzy Sets and Systems, 2*(4), 293-307.

[48] Kirby, B., & Hirst, E. (1997). *Ancillary service details: Voltage control* Oak Ridge National Laboratory Oak Ridge.

[49] Kiszka, J., Kochanska, M., & Sliwinska, D. (1985). The influence of some parameters on the accuracy of a fuzzy model. *Industrial Applications of Fuzzy Control, ,* 187-230.

[50] Klein, M., Le, L., Rogers, G., Farrokhpay, S., Balu, N., KAMWA, I., & GRONDIN, R. (1995). H∞ damping controller design in large power systems. discussion. author's reply. *IEEE Transactions on Power Systems, 10*(1), 158-166.

[51] Klein, M., Rogers, G., & Kundur, P. (1991). A fundamental study of inter-area oscillations in power systems. *IEEE Transactions on Power Systems, 6*(3), 914-921.

[52] Koay, C. A., & Srinivasan, D. (2003). Particle swarm optimization-based approach for generator maintenance scheduling. *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE,* 167-173.

[53] Koper, K. D., Wysession, M. E., & Wiens, D. A. (1999). Multimodal function optimization with a niching genetic algorithm: A seismological example. *Bulletin of the Seismological Society of America, 89*(4), 978-988.

[54] Kundur, P., Balu, N. J., & Lauby, M. G. (1994). *Power system stability and control* McGraw-hill New York.

[55] Kundur, P., Paserba, J., Ajjarapu, V., Andersson, G., Bose, A., Canizares, C., . . . Taylor, C. (2004). Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions. *Power Systems, IEEE Transactions On, 19*(3), 1387-1401.

[56] Kurian, C. P., George, V., Bhat, J., & Aithal, R. S. (2006a). ANFIS model for the time series prediction of interior daylight illuminance. *International Journal on Artificial Intelligence and Machine Learning, 6*(3), 35-40.

[57] Kurian, C. P., George, V., Bhat, J., & Aithal, R. S. (2006b). ANFIS model for the time series prediction of interior daylight illuminance. *International Journal on Artificial Intelligence and Machine Learning, 6*(3), 35-40.

[58] Larsen, E., & Swann, D. (1981). Applying power system stabilizers part III: Practical considerations. *Power Apparatus and Systems, IEEE Transactions On*, (6), 3034-3046.

[59] Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering, 194*(36), 3902-3933.

[60] Li, Q., Zhao, D., & Yu, Y. (1989). A new pole-placement method for excitation control design to damp SSR of a nonidentical two-machine system. *Power Systems, IEEE Transactions On, 4*(3), 1176-1181.

[61] Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *Evolutionary Computation, IEEE Transactions On, 10*(3), 281-295.

[62] Liang, J., & Suganthan, P. N. (2005). Dynamic multi-swarm particle swarm optimizer. *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, 124-129.

[63] Liu, Y., & Passino, K. M. (2000). Swarm intelligence: Literature overview. *Department of Electrical Engineering, the Ohio State University,*

[64] Liu, Y., & He, X. (2005). Modeling identification of power plant thermal process based on PSO algorithm. *American Control Conference, 2005. Proceedings of the 2005,* 4484-4489.

[65] Machowski, J., Bialek, J., & Bumby, J. (2011). *Power system dynamics: Stability and control* John Wiley & Sons.

[66] Mahfoud, S. W. (1995). Niching methods for genetic algorithms. *Urbana, 51*(95001)

[67] Matsushima, K., & Sugiyama, H. (1985). Human operator's fuzzy model in man-machine system with a nonlinear controlled object. *Digest Industrial Applications in Fuzzy Control, ,* 175-185.

[68] Matyas, J. (1965). Random optimization. *Automation and Remote Control, 26*(2), 246-253.

[69] Mendes, R., Kennedy, J., & Neves, J. (2004). The fully informed particle swarm: Simpler, maybe better. *Evolutionary Computation, IEEE Transactions On, 8*(3), 204-210.

[70] Miranda, M., & Win-Oo, N. (2006). New experiments with EPSO—Evolutionary particle swarm optimization. *Proc. IEEE Swarm Intell. Symp,* 162-169.

[71] Mohammadpour, H. A., Mirhoseini, S. M. H., & Shoulaie, A. (2009). Comparative study of proportional and TS fuzzy controlled GCSC for SSR

mitigation. *Power Engineering, Energy and Electrical Drives, 2009. POWERENG'09. International Conference On*, 564-569.

[72] Moore, C. G., & Harris, C. (1992). Indirect adaptive fuzzy control. *International Journal of Control, 56*(2), 441-468.

[73] Naka, S., Genji, T., Yura, T., & Fukuyama, Y. (2003). A hybrid particle swarm optimization for distribution state estimation. *Power Systems, IEEE Transactions On, 18*(1), 60-68.

[74] Naka, S., Genji, T., Yura, T., Fukuyama, Y., & Hayashi, N. (2000). Distribution state estimation considering nonlinear characteristics of practical equipment using hybrid particle swarm optimization. *Power System Technology, 2000. Proceedings. PowerCon 2000. International Conference On, , 2* 1083-1088.

[75] Oliveira, R., Ramos, R., & Bretas, N. (2008). A mixed procedure based on classical and modern control to design robust damping controllers. *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, 1-1.

[76] Omran, M. G., & Mahdavi, M. (2008). Global-best harmony search. *Applied Mathematics and Computation, 198*(2), 643-656.

[77] Panduro, M. A., Brizuela, C. A., Balderas, L. I., & Acosta, D. A. (2009). A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays. *Progress in Electromagnetics Research B, 13*, 171-186.

[78] Parsopoulos, K., & Vrahatis, M. (2004). UPSO: A unified particle swarm optimization scheme. *Lecture Series on Computer and Computational Sciences, 1*, 868-873.

[79] Parsopoulos, K. E., & Vrahatis, M. N. (2005). Unified particle swarm optimization for solving constrained engineering optimization problems. *Advances in natural computation* (pp. 582-591) Springer.

[80] Pivonka, P., Veleba, V., Seda, M., Osmera, P., & Matousek, R. (2009). The short sampling period in adaptive control. *Proc. of the IAENG Int. Conference WCECS*, 724-729.

[81] Pohlheim, H. (2007). Examples of objective functions. *Retrieved, 4*(10), 2012.

[82] Price, W. W., Hargrave, A. W., Hurysz, B. J., Chow, J. H., & Hirsch, P. M. (1998). Large-scale system testing of a power system dynamic equivalencing program. *IEEE Transactions on Power Systems, 13*(3), 768-774.

[83] Rasmussen, J. (1985). The role of hierarchical knowledge representation in decisionmaking and system management. *IEEE Transactions on Systems, Man and Cybernetics, 15*(2), 234-243.

[84] Ratnaweera, A., Halgamuge, S., & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *Evolutionary Computation, IEEE Transactions On, 8*(3), 240-255.

[85] Ribeiro Filho, J. L., Treleaven, P. C., & Alippi, C. (1994). Genetic-algorithm programming environments. *Computer, 27*(6), 28-43.

[86] Robinson, D. G. (2005). Reliability analysis of bulk power systems using swarm intelligence. *Reliability and Maintainability Symposium, 2005. Proceedings. Annual*, 96-102.

[87] Rogers, G. (1996). Demystifying power system oscillations. *Computer Applications in Power, IEEE, 9*(3), 30-35.

[88] Rojas, I., Bernier, J. L., Rodriguez-Alvarez, R., & Prieto, A. (2000). What are the main functional blocks involved in the design of adaptive neuro-fuzzy

inference systems? *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference On, , 6* 551-556.

[89] Russell, B. (2009). *Autobiography* Routledge.

[90] Saadat, H. (2002). *Power system analysis* McGraw-Hill Primis Custom.

[91] Sallama, A., Abbod, M., & Turner, P. (2012). Neuro-fuzzy system for power generation quality improvements. *Universities Power Engineering Conference (UPEC), 2012 47th International,* 1-6.

[92] Schwefel, H. (1981). *Numerical optimization of computer models* John Wiley & Sons, Inc.

[93] Sensarma, P. S., Rahmani, M., & Carvalho, A. (2002). A comprehensive method for optimal expansion planning using particle swarm optimization. *Power Engineering Society Winter Meeting, 2002. IEEE, , 2* 1317-1322.

[94] Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control* MIT press.

[95] Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., the 1998 IEEE International Conference On,* 69-73.

[96] Shi, Y., & Eberhart, R. C. (2001). Fuzzy adaptive particle swarm optimization. *Evolutionary Computation, 2001. Proceedings of the 2001 Congress On, , 1* 101-106.

[97] SimPowerSystems 4.3 user's guide. Retrieved from http://www.mathworks.com/products/simpower

[98] Slochanal, S. M. R., Kannan, S., & Rengaraj, R. (2004). Generation expansion planning in the competitive environment. *Power System Technology, 2004. PowerCon 2004. 2004 International Conference On, , 2* 1546-1549.

[99] Snyder, A., Mohammed, A., Georges, D., Margotin, T., Hadjsdid, N., & Mili, L. (1999). A robust damping controller for power systems using linear matrix inequalities. *Power Engineering Society 1999 Winter Meeting, IEEE, , 1* 519-524.

[100] Song, Y., & Johns, A. (1999). *Flexible ac transmission systems (FACTS)* IET.

[101] Srinivasan, D., & Malik, I. M. (2002). Flexible generator maintenance scheduling in a practical system using fuzzy logic and genetic algorithm. *Hybrid information systems* (pp. 395-413) Springer.

[102] Subbaraj, P., & Manickavasagam, K. (2008). Automatic generation control of multi-area power system using fuzzy logic controller. *European Transactions on Electrical Power, 18*(3), 266-280.

[103] Sugeno, M., & Tanaka, K. (1991). Successive identification of a fuzzy model and its applications to prediction of a complex system. *Fuzzy Sets and Systems, 42*(3), 315-334.

[104] Takagi, T., & Sugeno, M. (1983). Derivation of fuzzy control rules from human operator's control actions. *Proceedings of the IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, , 6* 55-60.

[105] Tandon, V., El-Mounayri, H., & Kishawy, H. (2002). NC end milling optimization using evolutionary computation. *International Journal of Machine Tools and Manufacture, 42*(5), 595-605.

[106] Tong, R. M. (1978). Synthesis of fuzzy models for industrial processes-some recent results. *International Journal of General System, 4*(3), 143-162.

[107] Tong, R. (1980). The evaluation of fuzzy models derived from experimental data. *Fuzzy Sets and Systems, 4*(1), 1-12.

[108] Van Der Rhee, F., van Nauta Lemke, Hans R, & Dijkman, J. G. (1990). Knowledge based fuzzy control of systems. *Automatic Control, IEEE Transactions On, 35*(2), 148-155.

[109] Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N. (2003). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & Chemical Engineering, 27*(3), 293-311.

[110] Victoire, T., & Jeyakumar, A. E. (2004). Hybrid PSO–SQP for economic dispatch with valve-point effect. *Electric Power Systems Research, 71*(1), 51-59.

[111] Victoire, T., & Jeyakumar, A. (2005). Unit commitment by a tabu-search-based hybrid-optimisation technique. *Generation, Transmission and Distribution, IEE Proceedings-, , 152*(4) 563-574.

[112] Wang, B., Tai, N., Zhai, H., Ye, J., Zhu, J., & Qi, L. (2008). A new ARMAX model based on evolutionary algorithm and particle swarm optimization for short-term load forecasting. *Electric Power Systems Research, 78*(10), 1679-1685.

[113] Wang, Y., Peng, X., & Wei, B. (2008). A new particle swarm optimization based auto-tuning of PID controller. *Machine Learning and Cybernetics, 2008 International Conference On, , 4* 1818-1823.

[114] Wu, F., Yen, Z., Hou, Y., & Ni, Y. (2004). Applications of AI techniques to generation planning and investment. *Power Engineering Society General Meeting, 2004. IEEE,* 936-940.

[115] Xu, C., & Lu, Y. (1987). Fuzzy model identification and self-learning for dynamic systems. *Systems, Man and Cybernetics, IEEE Transactions On, 17*(4), 683-689.

[116] Yang, S., & Li, C. (2010). A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. *Evolutionary Computation, IEEE Transactions On, 14*(6), 959-974.

[117] Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *Evolutionary Computation, IEEE Transactions On, 3*(2), 82-102.

[118] Yuhui Shi, & Eberhart, R. (1998). A modified particle swarm optimizer. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., the 1998 IEEE International Conference On,* 69-73. doi:10.1109/ICEC.1998.699146

[119] Zadeh, L. A. (1996). Fuzzy sets. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A.Zadeh, ,* 19-34.

[120] Zhan, Z., Xiao, J., Zhang, J., & Chen, W. (2007). Adaptive control of acceleration coefficients for particle swarm optimization based on clustering analysis. *Evolutionary Computation, 2007. CEC 2007. IEEE Congress On,* 3276-3282.

[121] Zhan, Z., & Zhang, J. (2008). Adaptive particle swarm optimization. *Ant colony optimization and swarm intelligence* (pp. 227-234) Springer.

[122] Zhan, Z., Zhang, J., Li, Y., & Chung, H. (2009). Adaptive particle swarm optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions On, 39*(6), 1362-1381.

[123] Zhang, X., Rehtanz, C., & Pal, B. (2006). *Flexible AC transmission systems: Modelling and control* Springer.

[124] Zhao, B., Guo, C., & Cao, Y. (2004). Improved particle swam optimization algorithm for OPF problems. *Power Systems Conference and Exposition, 2004. IEEE PES,* 233-238.

[125] Zhao, S., Liang, J. J., Suganthan, P. N., & Tasgetiren, M. F. (2008). Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress On,* 3845-3852.