Spring 2009

Applications of swarm, evolutionary and quantum algorithms in system identification and digital filter design

Bipul Luitel

APPLICATIONS OF SWARM, EVOLUTIONARY AND QUANTUM ALGORITHMS IN SYSTEM IDENTIFICATION AND DIGITAL FILTER DESIGN

by

BIPUL LUITEL

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER ENGINEERING

2009

Approved by

Ganesh K. Venayagamoorthy, Advisor
Steve E. Watkins
Steven L. Grant

ABSTRACT

The thesis focuses on the application of computational intelligence (CI) techniques for two problems- system identification and digital filter design. In system identification, different case studies have been carried out with equal or reduced number of orders as the original system and also in identifying a blackbox model. Lowpass, Highpass, Bandpass and Bandstop FIR and Lowpass IIR filters have been designed using three algorithms using two different fitness functions. Particle Swarm Optimization (PSO), Differential Evolution based PSO (DEPSO) and PSO with Quantum Infusion (PSO-QI) algorithms have been applied in this work. PSO-QI is a new hybrid algorithm where global best particle (*gbest*) obtained from PSO goes into a tournament with an offspring produced by mutating the *gbest* of PSO using the quantum principle in Quantum behaved PSO (QPSO) and the winner is selected as the new *gbest* of the swarm. In QPSO, unlike traditional PSO, exact values of particle's position and velocity cannot be determined. However, its position in the solution space is determined by mapping the probability of its appearance in the quantized search space. The results obtained from PSO-QI have been compared with the DEPSO hybrid algorithm and the classical PSO. In all of the cases, PSO-QI has outperformed the other two algorithms in its ability to converge to the lowest error value and its consistency in finding the solution every time and thus proven to be the best. However, the computational complexity of PSO-QI is higher than that of the other two algorithms.

ACKNOWLEDGMENTS

I would like to acknowledge my advisor Dr. Ganesh Kumar Venayagamoorthy for his continued support, suggestion and guidance throughout the course of my research and also in my coursework. I would also like to acknowledge NSF CAREER grant #0348221 for the financial support for my research. The knowledge I got is a valuable addition to my education and couldn't have been possible without this support.

I would like to acknowledge my committee members, Dr. Steven L. Grant and Dr. Steve E. Watkins, for their suggestions and support throughout the course of my studies and in preparing this thesis.

I would like to acknowledge my colleagues at Real-Time Power and Intelligent Systems Laboratory for their suggestions and support during my course of studies and research. Finally, I would also like to acknowledge this success to my friends and my family, especially to Sirjana Dahal for her continued moral support throughout.

TABLE OF CONTENTS

Page

ABSTRACT iii

ACKNOWLEDGMENTS iv

LIST OF ILLUSTRATIONS viii

LIST OF TABLES xi

1. INTRODUCTION 1
   - 1.1. INTRODUCTION 1
   - 1.2. OBJECTIVES 2
   - 1.3. THESIS LAYOUT 2
   - 1.4. NEW CONTRIBUTIONS 3
   - 1.5. RESEARCH PUBLICATIONS 3
   - 1.6. SUMMARY 4
2. SYSTEM IDENTIFICATION 5
   - 2.1. INTRODUCTION 5
   - 2.2. SYSTEM IDENTIFICATION PROBLEM 5
   - 2.3. SYSTEM IDENTIFICATION TECHNIQUES 7
   - 2.4. SUMMARY 8
3. DIGITAL FILTER DESIGN 9
   - 3.1. INTRODUCTION 9
   - 3.2. DIGITAL FILTER 9
     - 3.2.1. Finite Impulse Response. 10
     - 3.2.2. Infinite Impulse Response. 11
     - 3.2.3. Lowpass Filter. 12
     - 3.2.4. Highpass Filter. 13
     - 3.2.5. Bandpass Filter. 14
     - 3.2.6. Bandstop Filter. 14
   - 3.3. FILTER DESIGN TECHNIQUES 15
     - 3.3.1. Traditional Techniques. 15
     - 3.3.2. Computational Intelligence Techniques. 16

3.4. SUMMARY 16
4. PARTICLE SWARM OPTIMIZATION 17
4.1. INTRODUCTION 17
4.2. PSO ALGORITHM 17
4.2.1. Parameters. 18
4.2.2. Topologies. 20
4.3. MODIFIED PARTICLE SWARM OPTIMIZATION 23
4.4. SUMMARY 24
5. DIFFERENTIAL EVOLUTION PARTICLE SWARM OPTIMIZATION 25
5.1. INTRODUCTION 25
5.2. DEPSO ALGORITHM 25
5.2.1. Differential Evolution 25
5.2.2. DEPSO.. 26
5.3. SUMMARY 28
6. PARTICLE SWARM OPTIMIZATION WITH QUANTUM INFUSION 29
6.1. INTRODUCTION 29
6.2. PSO-QI ALGORITHM 29
6.2.1. QDPSO. 29
6.2.2. QPSO. 32
6.2.3. PSO-QI.. 33
6.3. SUMMARY 35
7. SYSTEM IDENTIFICATION USING PSO, PSO-EA, DEPSO AND PSO-QI 36
7.1. INTRODUCTION 36
7.2. APPLICATION IN SYSTEM IDENTIFICATION 40
7.3. RESULTS WITH PSO 44
7.4. RESULTS WITH PSO-EA 44
7.5. RESULTS WITH DEPSO 44
7.6. RESULTS WITH PSO-QI 44
7.7. DISCUSSION 81
7.8. SUMMARY 82
8. DIGITAL FITLER DESIGN USING PSO, DEPSO AND PSO-QI 83

8.1. INTRODUCTION 83
8.2. APPLICATION IN DIGITAL FILTER DESIGN 86
8.3. RESULTS WITH PSO AND DEPSO 89
8.4. RESULTS WITH PSO-QI 101
8.5. DISCUSSION 129
8.6. SUMMARY 130
9. CONCLUSION 131
9.1. INTRODUCTION 131
9.2. SECTION SUMMARY 131
9.3. MAIN CONCLUSION 132
9.4. FUTURE RESEARCH 133
9.5. SUMMARY 134
BIBLIOGRAPHY 135
VITA 138

LIST OF ILLUSTRATIONS

Page

Figure 2.1. Schematic showing system identification 6
Figure 3.1. Illustration of filter parameters. 10
Figure 3.2. Lowpass filter 12
Figure 3.3. Highpass filter 13
Figure 3.4. Bandpass filter 14
Figure 3.5. Bandstop filter 15
Figure 4.1. Flowchart for PSO 19
Figure 4.2. Star Topology 20
Figure 4.3. Ring Topology 21
Figure 4.4. Wheel Topology 22
Figure 4.5. Flowchart for PSO-EA 24
Figure 5.1. Flowchart for DEPSO 27
Figure 6.1. Flowchart for PSO-QI 34
Figure 7.1. System identification block diagram 40
Figure 7.2. Flowchart for the system identification 43
Figure 7.3. Error graph for full order model of Case I 45
Figure 7.4. Error graph for reduced order model of Case I 46
Figure 7.5. Pole zero plot for the full order model of Case I 47
Figure 7.6. Error graph for the full order model of Case II 48
Figure 7.7. Error graph for the reduced order model of Case II 49
Figure 7.8. Pole zero plot for the full order model of Case II 50
Figure 7.9. Error graph for the full order model of Case III 51
Figure 7.10. Error graph for the reduced order model of Case III 52
Figure 7.11. Pole zero plot for the full order model of Case III 53
Figure 7.12. Error graph for the full order model of Case IV 54
Figure 7.13. Error graph for the reduced order model of Case IV 55
Figure 7.14. Pole zero plot for the full order model of Case IV 56
Figure 7.15. Error graph for Case V 57

Figure 7.16. Pole zero plot for the full order model of Case V 58
Figure 7.17. Error graph for Case VI... 59
Figure 7.18. Pole zero plot for the full order model of Case VI 60
Figure 7.19. Error graph for full order model of Case I. 67
Figure 7.20. Error graph for reduced order model of Case I. 68
Figure 7.21. Error graph for full order model of Case II. 69
Figure 7.22. Error graph for the reduced order model of Case II. 70
Figure 7.23. Error graph for the full order model of Case III. 71
Figure 7.24. Error graph for the reduced order model of Case III. 72
Figure 7.25. Error graph for the full order model of Case IV. 73
Figure 7.26. Error graph for the reduced order model of Case IV. 74
Figure 7.27. Error graph for the full order model of Case V. 75
Figure 7.28. Error graph for the full order model of Case VI. 76
Figure 7.29. Error graph for full order model of Case III in 50 iterations. 77
Figure 7.30. Error graph for the full order model of Case IV in 50 iterations. 78
Figure 8.1. Flowchart for the design of digital filters using PSO-QI. 88
Figure 8.2. Error graph for Case I in 200 iterations. 90
Figure 8.3. Magnitude response of filters for Case I in 200 iterations. 91
Figure 8.4. Gain response for the filters designed in Case I in 200 iterations. 92
Figure 8.5. Error graph for Case II in 200 iterations. 93
Figure 8.6. Magnitude plot for the filters designed in Case II in 200 iterations. 94
Figure 8.7. Comparison of magnitude response for the two cases. 95
Figure 8.8. Comparison of gain response for the two cases. 96
Figure 8.9. Error graph for Case I in 40 iterations. 97
Figure 8.10. Magnitude response of the filters designed in Case I in 40 iterations. 98
Figure 8.11. Gain plot for the filters designed in Case I in 40 iterations. 99
Figure 8.12. Error graph showing the comparison of PSO, QPSO and DEPSO. 102
Figure 8.13. Error graph for LP FIR filter designed in Case III 103
Figure 8.14. Magnitude response of the LP FIR filter designed in Case III 104
Figure 8.15. Gain response of the LP FIR filter designed in Case III. 105
Figure 8.16. Error graph for the LP IIR filter designed in Case III 106

Figure 8.17. Magnitude response for the LP IIR filter designed in Case III 107
Figure 8.18. Gain response for the LP IIR filter designed in Case III 108
Figure 8.19. Magnitude response of HP FIR filter designed in Case III 109
Figure 8.20. Gain response of HP FIR filter designed in Case III 110
Figure 8.21. Error graph for the BP FIR filter designed in Case III 111
Figure 8.22. Magnitude plot of the BP FIR filter designed in Case III 112
Figure 8.23. Gain plot of the BP FIR filter designed in Case III 113
Figure 8.24. Magnitude response of the BS FIR filter designed in Case III.................. 114
Figure 8.25. Gain response of the BS FIR filter designed in Case III 115
Figure 8.26. Error graph for the LP FIR filter designed in Case IV 116
Figure 8.27. Magnitude plot for the LP FIR filter designed in Case IV 117
Figure 8.28. Gain plot for the LP FIR filter designed in Case IV................................... 118
Figure 8.29. Magnitude plot of the HP FIR filter designed in Case IV 119
Figure 8.30. Gain plot of the HP FIR filter designed in Case IV 120
Figure 8.31. Error graph for the BP FIR filter designed in Case IV 121
Figure 8.32. Magnitude response of the BP FIR filter designed in Case IV 122
Figure 8.33. Gain response of the BP FIR filter designed in Case IV 123
Figure 8.34. Magnitude response of the BS FIR filter designed in Case IV 124
Figure 8.35. Gain response of the BS FIR filter designed in Case IV 125
Figure 8.36. Comparison of PSO and PSO-QI in terms of time..................................... 126

LIST OF TABLES

Page

Table 7.1. Parameters used in the study 36
Table 7.1. (cont.) Parameters used in the study 37
Table 7.2. Study of Cases I and II 37
Table 7.3. Study of Case III 38
Table 7.4. Study of Case IV 39
Table 7.5. Study of Cases V and VI 39
Table 7.6. Data for the Full Order Model 61
Table 7.6. (cont.) Data for the Full Order Model 62
Table 7.7. Data for the Reduced Order Model 62
Table 7.8. Coefficients for Full Order Model 63
Table 7.8. (cont.) Coefficients for Full Order Model 64
Table 7.9. Coefficients for Reduced Order Model 65
Table 7.9. (cont.) Coefficients for Reduced Order Model 66
Table 7.10. Results of full order model for 500 iterations. 79
Table 7.11. Results of reduced order model for 500 iterations. 80
Table 7.12. Results of full order model for 50 iterations. 81
Table 8.1. Parameters used in the study of Cases I and II 84
Table 8.2. Parameters used in the study of Cases III and IV 85
Table 8.3. Passband and stopband ripples with 200 iterations 100
Table 8.4. Passband and stopband ripples with 40 iterations 100
Table 8.5. Passband and stopband ripples for FIR filter in Case III 127
Table 8.6. Passband and stopband ripples for the IIR filter in Case III 128
Table 8.7. Passband and stopband ripples for the FIR filter in Case IV 129

1. INTRODUCTION

1.1. INTRODUCTION

System identification is a challenging and complex optimization problem due to nonlinearity of the systems and even more in a dynamic environment. Adaptive infinite impulse response (IIR) systems are preferably used in modeling real world systems because of their reduced number of coefficients and better response over the finite impulse response (FIR) filters. In this work, system identification has been viewed as a problem of adaptive IIR filtering so that it becomes a parameter estimation problem. Digital filter design is also a complex optimization problem due to the number of filter parameters that can be optimized. Hence different computational intelligence (CI) techniques can be used to estimate the filter coefficients so as to optimize these parameters and design the desired filter response.

Particle Swarm Optimization (PSO) and its other variants have been a topic of research over the past decade. Inspired by social behavior of bird flocking and fish schooling, PSO has proven to be an effective stochastic search technique. Hence it has been applied to a wide variety of problems related to search optimization, clustering, routing, scheduling. PSO has gone through various changes and different variants have been introduced in order to solve the problems more effectively. It has also been combined with other different algorithms to create hybrid optimization algorithms. These algorithms have been reported in different literatures and applied to different practical applications. In this thesis, two problems have been studied- system identification and digital filter design. These applications have been implemented using the standard PSO and two hybrid algorithms- Differential Evolution Particle Swarm Optimization (DEPSO) and PSO with Quantum Infusion (PSO-QI). The results of system identification have also been compared with another hybrid algorithm PSO with Evolutionary Algorithm (PSO-EA). The thesis covers the details of these algorithms, the research work carried out towards the implementation of the above mentioned problems and their results.

1.2. OBJECTIVES

The main objective of this research is to apply swarm, evolutionary and quantum based algorithms to solve two practical problems viz. system identification and digital filter design. PSO, DEPSO and PSO-QI are the major algorithms involved in this work for system identification and in the design of digital filters. The results of the case studies are also presented.

1.3. THESIS LAYOUT

The thesis has been divided into 9 chapters. Chapter 1 introduces to the topic and outlines the objectives of the research work carried out. The next two chapters explain the major areas of this research work. In Chapter 2, system identification has been explained. This chapter introduces to the problem of system identification and traditional and modern techniques used to solve it. In Chapter 3, digital filter design is explained. This chapter introduces to the problem and traditional and modern techniques used in digital filter design.

In next three chapters, the three algorithms have been explained in detail. In Chapter 4, PSO has been covered. This chapter explains the basics of the algorithm and how it has been applied to the above mentioned problems. In Chapter 5, DEPSO has been explained. Similarly, PSO-QI has been explained in Chapter 6.

In the next two chapters, case studies carried out during the research and the results obtained from them have been presented. In Chapter 7, studies and results of system identification have been presented. This chapter shows the comparison of results obtained from system identification, and is presented as figures and tabulated data. In Chapter 8, similar results obtained for digital filter design are presented. These results are also presented as figures and tabulated data and show a comparison of different algorithms as applied to the problem.

Conclusion of the thesis and future work is presented in Chapter 9.

1.4. NEW CONTRIBUTIONS

The research work leading towards this thesis makes the following contributions:

System identification:

- Application of hybrid algorithms DEPSO and PSO-QI (new algorithm).
- Comparison of the results for different case studies and in both full and reduced order system cases show that hybrid algorithms introduced in this work perform better than the standard PSO or PSO-EA and PSO-QI performs better than DEPSO.

Digital filter design:

- Use of new hybrid algorithms DEPSO and PSO-QI for the design of Lowpass, Highpass, Bandpass and Bandstop FIR and Lowpass IIR filters.
- Use of two different fitness functions for the filter design in order to illustrate the robustness of the CI algorithms.
- Comparison of results obtained from different algorithms in terms of execution time, fitness obtained and consistency of convergence, and their analysis to understand the efficiency of the algorithms in the design of digital filters.
- All of the results show that the new hybrid algorithms introduced in this work have a much better and consistent performance over the standard PSO.

1.5. RESEARCH PUBLICATIONS

As a result of the research work carried out over the course of studies, two refereed conference papers were published [Luitel and Venayagamoorthy, 2008(a); 2008(b)] and two journal papers are to be submitted [Luitel and Venayagamoorthy, 2008(c); Luitel and Venayagamoorthy, 2008(d)].

1.6. SUMMARY

This chapter briefly introduced to the topic of the research and the content layout of the thesis. The objective of the study and the major contributions of it are also presented in this chapter. The chapter also listed the publications that came out as a result of the research work leading towards this thesis.

2. SYSTEM IDENTIFICATION

2.1. INTRODUCTION

System identification is a challenging and complex optimization problem due to nonlinearity of the systems and even more in a dynamic environment. Adaptive infinite impulse response systems are preferably used in modeling real world systems because of their reduced number of coefficients and better performance over the finite impulse response filters. Particle Swarm Optimization (PSO) and its other variants has been a subject of research for the past few decades for solving complex optimization problems. In this thesis, the concept of Differential Evolution based Particle Swarm Optimization (DEPSO) is implemented for system identification. A hybrid of Particle Swarm Optimization and Evolutionary Algorithm (PSO-EA) has been considered for comparison with PSO and DEPSO algorithms.

2.2. SYSTEM IDENTIFICATION PROBLEM

System identification is the mathematical modeling of an unknown system by monitoring its input output data. This is achieved by varying the parameters of the developed model so that for a set of given inputs, its output match that of the system under consideration. For a plant whose behavior is not known, an adaptive system can be modeled and its parameters can be continuously adjusted using any adaptive algorithms. By the use of such adaptive algorithms, the required parameters can be obtained such that the output of the plant and the model are same for the same set of inputs, which is the goal of system identification (Panda et al., 2007). Traditionally, Least Mean Square (LMS) and other algorithms have been studied for the identification of linear and static systems (Windrow et. al., 1976). But, almost all physical systems are nonlinear to certain extent and recursive in nature and hence it is more convincing to model such systems by using nonlinear models (Panda et. al., 2007; Krusienski and Jenkins, 2005). Thus nonlinear system identification has attracted attention in the field of science and engineering. Hence these are better modeled as Infinite Impulse Response (IIR) models as they can provide better performance than a Finite Impulse Response (FIR) filter with the same number of coefficients (Shynk, 1989(a)). Thus the problem of nonlinear system identification can also be viewed as a problem of adaptive IIR filtering. Also, IIR models are more efficient than the FIR models for implementation as they require less parameter and hence fewer computations for the same level of performance. However, there are few problems associated with the use of IIR models in identification of a system, such as instability of the algorithms, slow convergence and convergence to the local minimum (Netto et al., 1995). In order to overcome these, different techniques have been developed over the years. Fig. 2.1 shows a block diagram describing the problem of system identification.

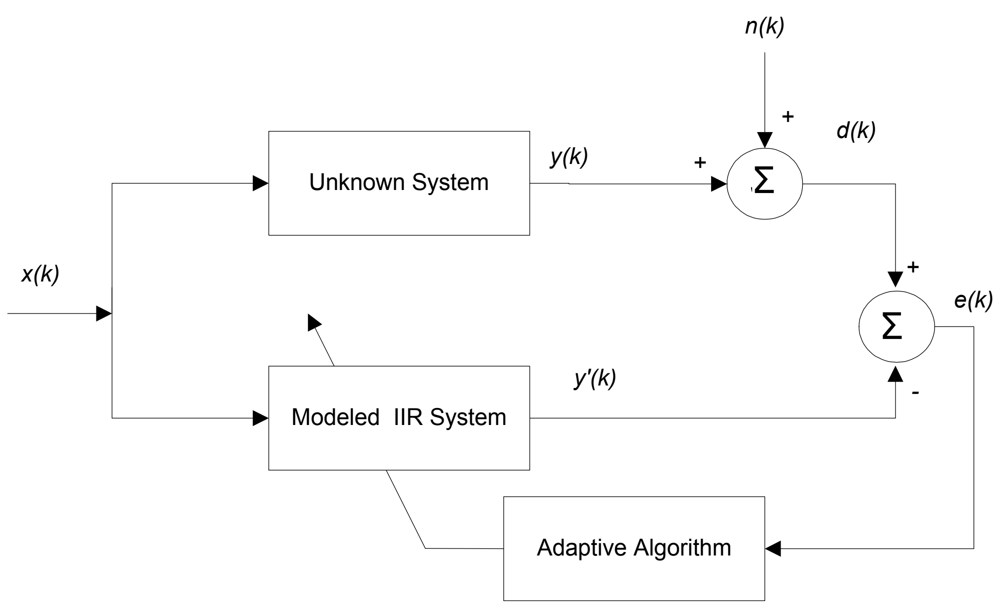

Figure 2.1. Schematic showing system identification

2.3. SYSTEM IDENTIFICATION TECHNIQUES

Different learning algorithms have been used in the past for nonlinear system identification. These techniques include use of neural network (Hongwei and Yanchun, 2005) and gradient based search techniques such as least mean square algorithm (Shynk, 1989(a)). Unfortunately, the error surface of such recursive systems such as a multi-machine power system (Kundur, 1993) tends to be multi-modal and hence traditional techniques of parameter approximation fail as they get trapped into local minimum and cannot attain the global minimum (Krusienski and Jenkins, 2005). Various algorithms that are implemented in the adaptive IIR filtering for system identification are described in (Netto et al., 1995).

Population based search algorithm such as Genetic Algorithm (GA) has also been used for the system identification. It uses a population of potential solutions encoded as chromosomes which go through genetic operations such as crossover and mutation to find the best solution (Kristinsson and Dumont, 1992). But its effectiveness is affected by the convergence time (the time it takes to find the global minimum). So to eliminate such deficiencies, population based stochastic optimization techniques have been discussed in various literatures. Particle Swarm Optimization (PSO) is one of the most known techniques (delValle et al., 2007). Application of PSO in the system identification has been discussed in (Panda et al., 2007). In (Lee et al., 2006), a method for the identification of nonlinear system and parameter optimization of the obtained input-output model has been described. The proposed method uses least squares support vector machines regression based on PSO. In another work, PSO has been used for optimizing the parameters of Elman neural network which is used for speed identification of ultrasonic motors (Hongwei and Yanchun, 2005). A modified form of PSO called as the self-organizing particle swarm optimization and its application in the system identification has been discussed in (Shen and Zeng, 2007). Radial Basis Function Neural Network (RBFNN) has been used for system identification in (Chen et al., 2007), where a hybrid gradient-based PSO algorithm has been used to adjust the parameters of the RBFNN. In (Liu et al., 2006), particle swarm optimization and quantum-behaved particle swarm optimization have been used for the system identification. Use of different types of stochastic optimization techniques in adaptive IIR filters and nonlinear systems has been explained in (Krusienski and Jenkins, 2005). Use of Differential Evolution (DE) and Ant Colony Optimization (ACO) in IIR filter design has been presented in (Karaboga, 2005) and (Karaboga et al., 2004) respectively. They also talk about the possible use of these approaches in system identification and other applications. But these algorithms have the tendency to get stuck in the local minimum when the complexity of the problem increases and in dynamic systems where time allowed for convergence is constrained. Hybrid algorithms are used to improve the performance by combining the best feature of both algorithms. In (Cai et. al., 2007), one such hybrid algorithm has been shown. In the paper, PSO and Evolutionary Algorithm (PSO-EA) hybrid has been implemented to combine the best features of PSO (co-operation) and EA (competition).

2.4. SUMMARY

Identification of complex systems is an optimization problem and is viewed as IIR system identification in this chapter. By the use of swarm and evolutionary algorithms, the coefficients of the filter are determined. The results of the study are shown in Chapter 7.

3. DIGITAL FILTER DESIGN

3.1. INTRODUCTION

This chapter introduces digital filter design as an optimization problem and discusses various methods applied in the design of digital filters traditionally and currently using the computational intelligence techniques.

3.2. DIGITAL FILTER

A filter is a frequency selective circuit that allows a certain frequency to pass while attenuating the others. Filters could be analog or digital. Analog filters use electronic components such as resistor, capacitor, transistor etc. to perform the filtering operations. These are mostly used in communication for noise reduction, video/audio signal enhancement etc. In contrast, digital filters use digital processors which perform mathematical calculations on the sampled values of the signal in order to perform the filter operation. A computer or a dedicated digital signal processor may be used for implementing digital filters. Filters mostly find their use in communication for noise reduction, audio/video signal enhancement etc.

Any time varying signal $C=x(t)$ sampled at a sampling interval of h has input signals $x_0, x_1, x_2, x_3, \ldots\ldots\ldots\ldots, x_n$ in intervals $0, h, 2h, 3h, \ldots\ldots\ldots\ldots, nh$. These inputs have corresponding outputs $y_0, y_1, y_2, y_3, \ldots\ldots\ldots\ldots, y_n$ depending upon the kind of operation performed. Thus, the order of the filter is determined by the number of the previous input terms used to calculate the current output. The a_0, a_1, a_2 terms appearing in the following equations are called the filter coefficients and determine the operation of the filter. These determine the characteristics of the filter. Various filter parameters which come into picture are the stopband and passband normalized frequencies (ω_s, ω_p), the passband and stopband ripple (δ_p) and (δ_s), the stopband attenuation and the transition width. This has been shown in Fig. 3.1.

$$ZeroOrder: y_n = a_0 x_n \tag{1}$$

$$FirstOrder: y_n = a_0 x_n + a_1 x_{n-1} \tag{2}$$

$$SecondOrder: y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} \tag{3}$$

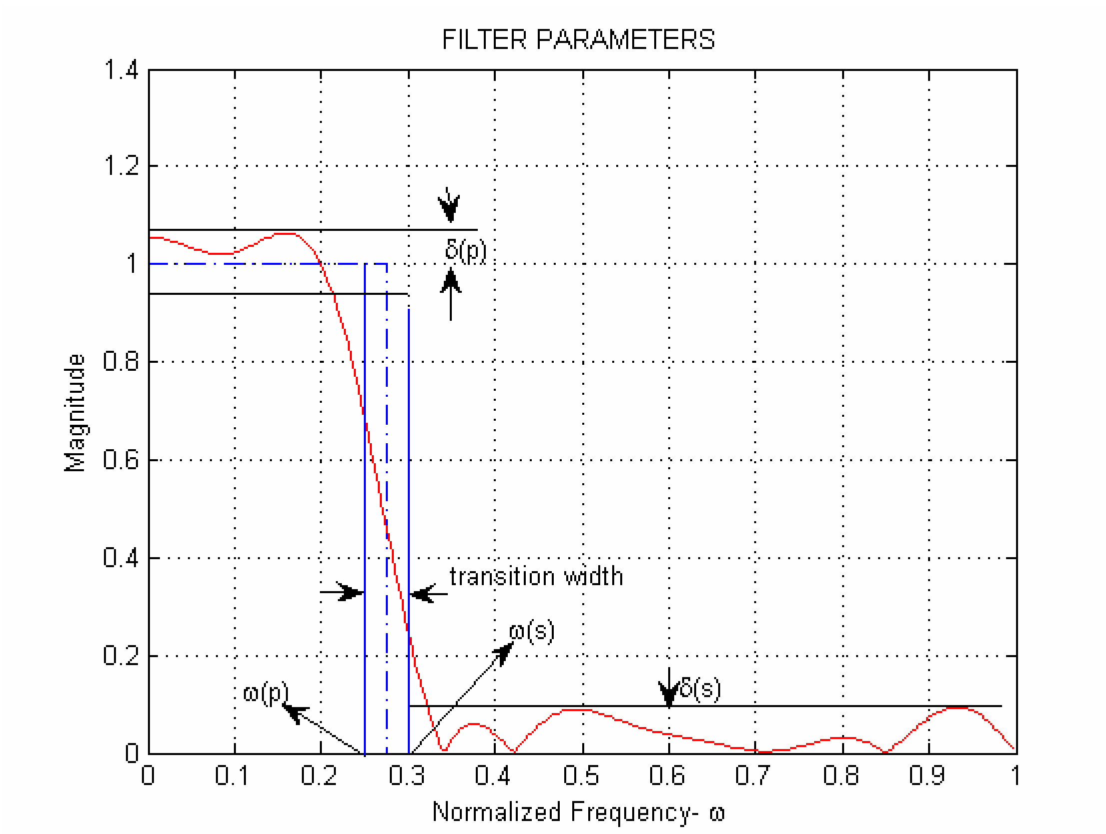

Figure 3.1. Illustration of filter parameters.

3.2.1. Finite Impulse Response. Finite Impulse Response filters are those for which the output of the filter depends only on the present inputs. FIR filter, or also called the non-recursive filter can be represented by the following difference equation:

$$y_n = b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} + .. + b_N x_{n-M} \tag{4}$$

By introducing a unit delay element z^{-1}, such that $z^{-1}x_n=x_{(n-1)}$, the transfer function of FIR filter can be represented as:

$$\frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2} + .. + b_n z^{-M} \tag{5}$$

$$H(z) = \sum_{i=0}^{M} b_i z^{-i} \tag{6}$$

The FIR filter has following advantages:

- Since FIR filter has its poles located at the origin, it is inherently stable since the poles lie within the unit circle.
- FIR filters can be designed as linear phase filters, making them a better choice in phase sensitive applications.

3.2.2. Infinite Impulse Response. Infinite impulse response filters are those for which the output of the filter at any given time depends upon the present inputs and past outputs. The difference equation for a Linear Time Invariant IIR filter can be written as:

$$y_n = b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} + .. + b_N x_{n-M} - a_1 y_{n-1} - ... - a_N y_{n-N} \tag{7}$$

Similar to FIR, introducing a unit delay element z^{-1}, such that $z^{-1}x_n=x_{(n-1)}$ and $z^{-1}y_n=y_{(n-1)}$, the transfer function of IIR filter can be represented as:

$$Y(z)(1 + a_1 z^{-1} + ... + a_N z^{-N}) = X(z)(b_0 + b_1 z^{-1} + b_2 z^{-1} + .. + b_N z^{-M}) \tag{8}$$

$$H(z) = \frac{\sum_{i=0}^{M} b_i z^{-i}}{1 + \sum_{i=0}^{N} a_i z^{-i}} \tag{9}$$

The IIR filters have the following advantages over FIR:

- They can achieve much sharper transition region than FIR filters of the same order.
- They require less memory and are computationally less complex for the same length of the filter.

However, due to the feedback element present in the IIR filters, chances of accumulating the rounded errors over summed iterations are higher.

3.2.3. Lowpass Filter. Lowpass filters are those that allow the frequencies below a threshold to pass while attenuating the frequencies beyond the threshold. The threshold frequency is called the cut-off frequency. Fig. 3.2 shows the Lowpass filter.

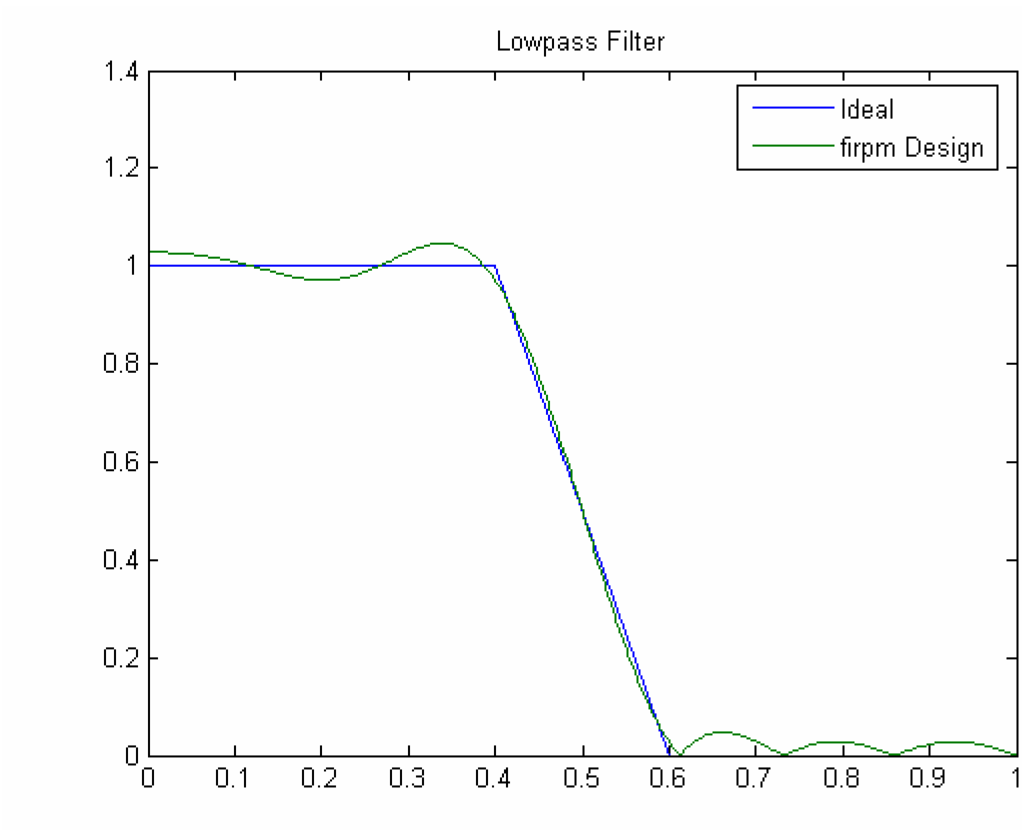

Figure 3.2. Lowpass filter

3.2.4. Highpass Filter. Highpass filters allow the frequencies beyond a threshold frequency to pass while attenuating others. Fig. 3.3 shows the highpass filter.

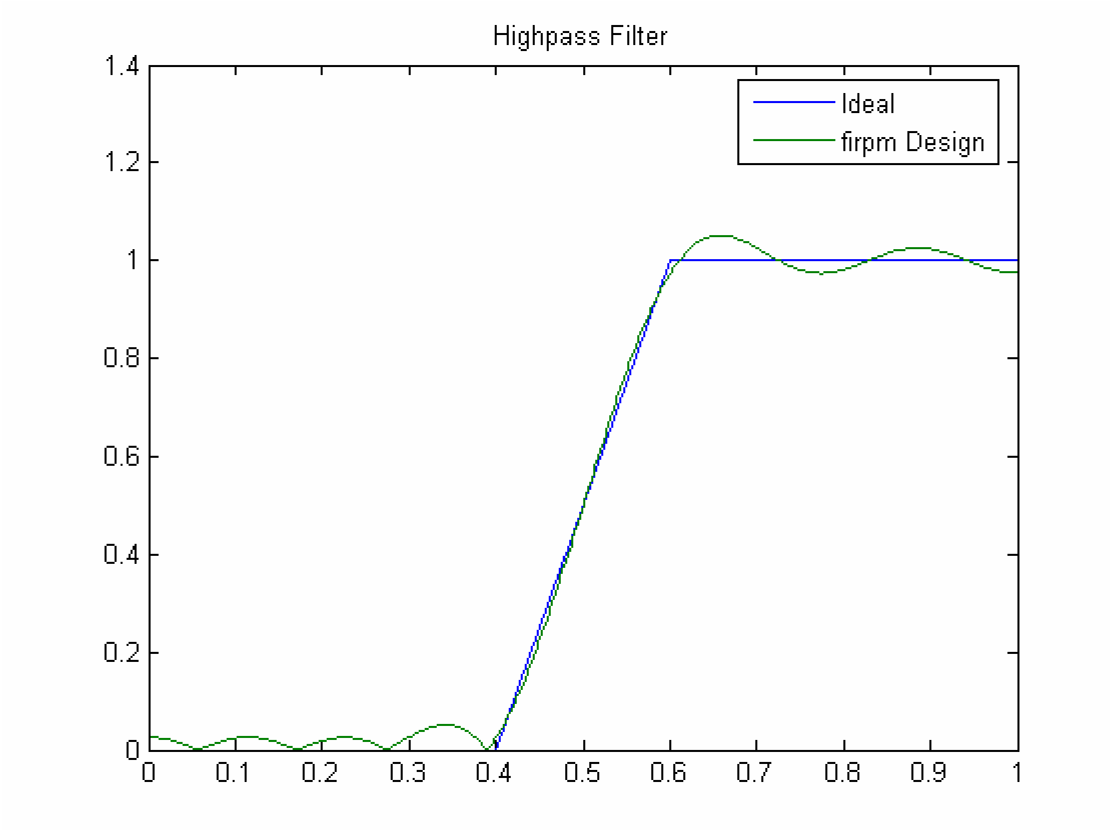

Figure 3.3. Highpass filter

3.2.5. Bandpass Filter. In a bandpass filter, frequencies which lie between a lower cutoff frequency and an upper cutoff frequency are allowed to pass while others are attenuated. The frequency band for which the filter allows to pass is called the pass band and the bands of frequencies which are attenuated are called the stopband frequencies. Fig. 3.4 shows the bandpass filter.

3.2.6. Bandstop Filter. In a bandstop filter, the frequencies between two cutoff frequencies are attenuated while the others are allowed to pass. Fig. 3.5 shows the bandstop filter.

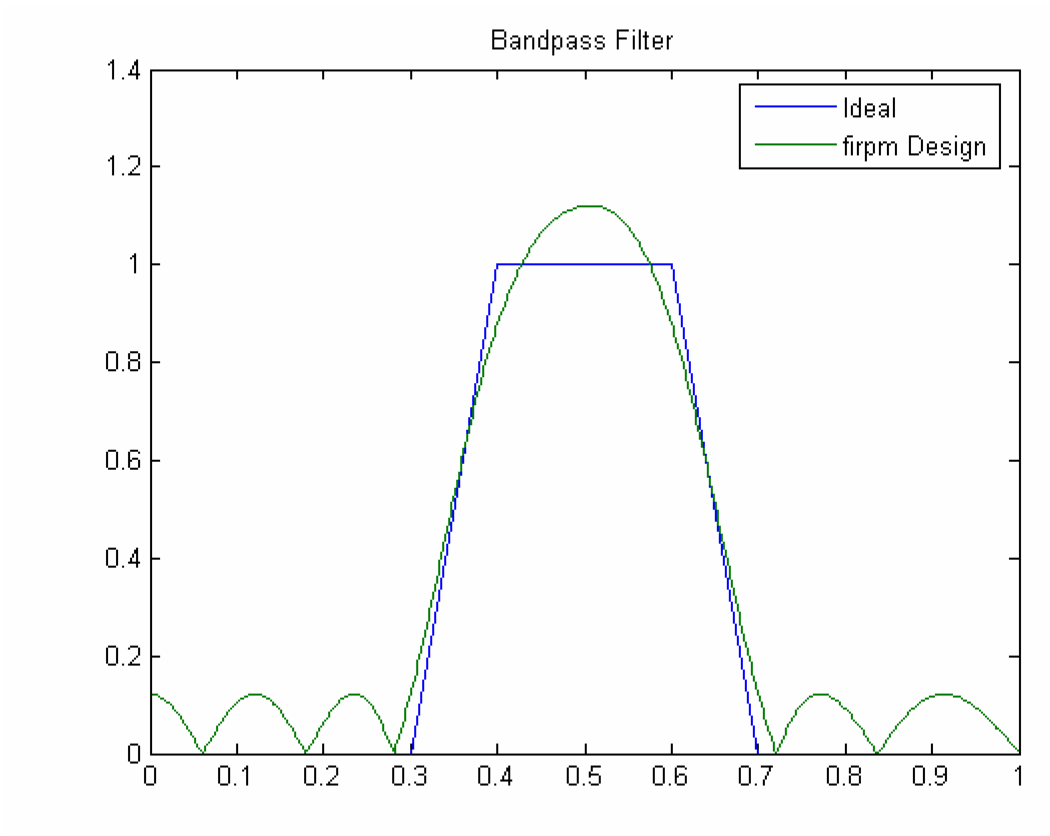

Figure 3.4. Bandpass filter

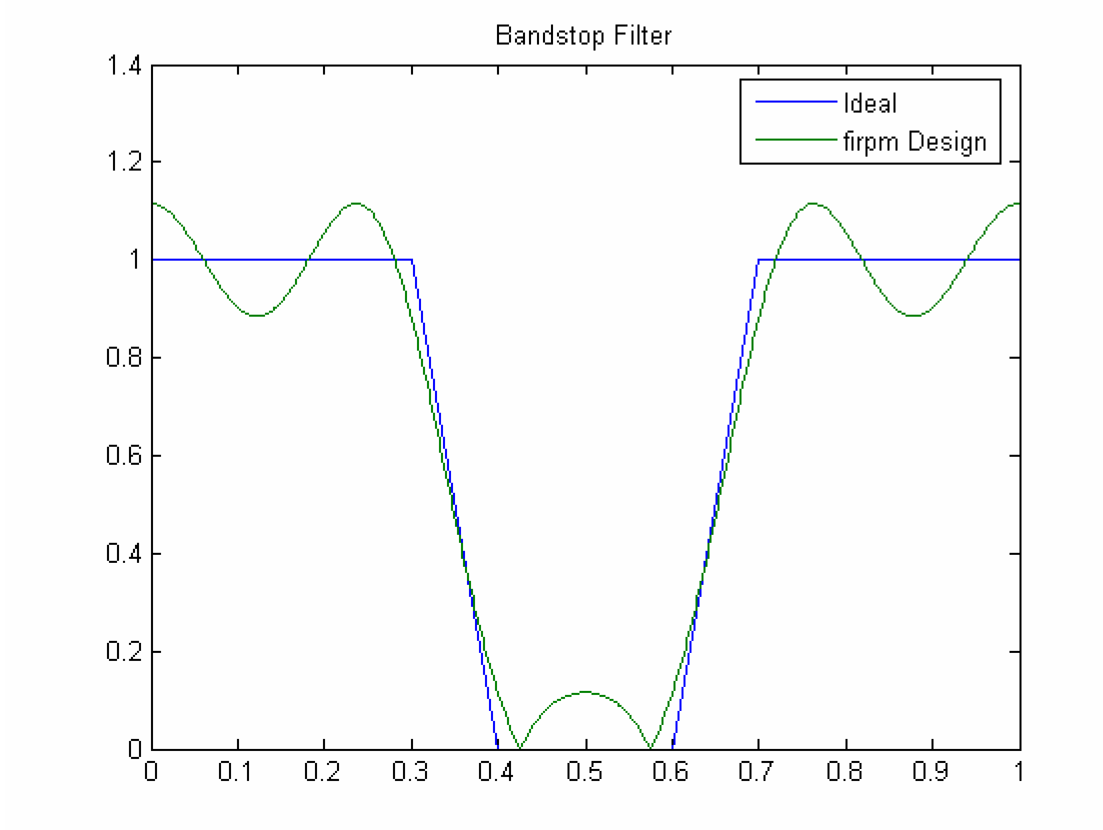

Figure 3.5. Bandstop filter

3.3. FILTER DESIGN TECHNIQUES

3.3.1. Traditional Techniques. Traditionally, different techniques exist for the design of digital filters. Of these, windowing method is the most popular. In this method, ideal impulse response is multiplied with a window function. There are various kinds of window functions (Butterworth, Chebyshev, Kaiser etc.), depending on the requirements of ripples on the passband and stopband, stopband attenuation and the transition width. These various windows limit the infinite length impulse response of ideal filter into a finite window to design an actual response. But windowing methods do not allow sufficient control of the frequency response in the various frequency bands and other filter parameters such as transition width. Designer always has to compromise on one or the other of the design specifications. So, computational intelligence techniques have been implemented in the design of digital filters to design with better parameter control and to better approximate the ideal filter. Since population based stochastic search methods have proven to be effective in multidimensional nonlinear environment, all of the constraints of filter design can be effectively taken care of by the use of these algorithms.

3.3.2. Computational Intelligence Techniques. Computational intelligence based techniques such as particle swarm optimization (PSO) and genetic algorithms (GA) have been implemented in the design of digital filters. Use of PSO and GA in the design of digital filters is described in (Ababneh and Bataineh, 2007). Use of differential evolution in the design of digital filters has been implemented in Storn's work (Storn, 1996; Storn, 2005; Karaboga, 2005). Design of infinite impulse response (IIR) filters using PSO is described in (Krusienski and Jenkins, 2004). Quantum behaved PSO (QPSO) and its application in filter design has been described in (Fang et al., 2006(a); Fang et al. 2006(b)).

3.4. SUMMARY

Digital filter design is an important aspect of digital signal processing. Although various traditional techniques have been used in the past, digital filter design as an optimization problem can be solved by using computational intelligence based techniques. The use of these intelligent stochastic search approaches tend to produce better results in a short period of time, thus opening grounds for adaptive filter to be designed to use in an online environment.

4. PARTICLE SWARM OPTIMIZATION

4.1. INTRODUCTION

Introduced by Eberheart and Kennedy in 1995 (del Valle et al., 2007), PSO is a search technique based on social behavior of bird flocking and fish schooling. There are different kinds of bio and social behavior inspired algorithms. PSO is one of the different swarm based algorithms. In PSO, each particle of the swarm is a possible solution in the multi-dimensional search space. The particles change their positions with a certain velocity in each iteration, according to the standard PSO equations, thus moving towards the global best (*gbest*) solution. Being easy to implement and yet so effective, PSO has been utilized in a wide variety of optimization applications. In this thesis, PSO has been used in system identification and to design digital filters.

4.2. PSO ALGORITHM

Particle swarm optimization is a population based search algorithm and is inspired by the observation of natural habits of bird flocking and fish schooling. In PSO, a swarm of particles moves through a D dimensional search space. The particles in the search process are the potential solutions, which move around the defined search space with some velocity until the error is minimized or the solution is reached, as decided by the fitness function. Fitness function is the measure of particles fitness which is the deviation of the particle from the required solution. The particles reach to the desired solution by updating their position and velocity according to the PSO equations. In PSO model, each individual is treated as a volume-less particle in the D-dimensional search space with initial random velocity. Each particle has memory which keeps track of its previous best position and fitness, with the position and velocity of i^{th} particle represented as:

$$X_i = (x_{i1}, x_{i2}, \ldots\ldots.., x_{iD}) \tag{10}$$

$$V_i = (v_{i1}, v_{i2}, \ldots\ldots.., v_{iD}) \tag{11}$$

These particles are randomly distributed over the search space with initial position and velocity. They change their positions and velocity according to (12) and (13) where c_1 and c_2 are cognitive and social acceleration constants, $rand_1()$ and $rand_2()$ are two random functions uniformly distributed in the range of [0,1] and w is the inertia weight introduced to accelerate the convergence speed of PSO (del Valle et al., 2007). Vector $P_i = (P_{i1}, P_{i2}, \ldots, P_{iD})$ is the best previous position (the position giving the best fitness value) of particle i called the *pbest*, and vector $P_g = (P_{g1}, P_{g2}, \ldots, P_{gD})$ is the position of the best particle among all the particles in the population and is called the *gbest*. X_{id}, V_{id}, P_{id} are the d^{th} dimension of vector of X_i, V_i, P_i. The *gbest* is changed to *lbest* in local PSO where *lbest* is the best value in the neighborhood.

$$V_{id}(k+1) = w * V_{id}(k) + c_1 * rand_1() * (P_{id} - X_{id}) + c_2 * rand_2() * (P_g - X_{id}) \quad (12)$$

$$X_{id}(k+1) = X_{id}(k) + V_{id}(k+1) \quad (13)$$

Other variations of PSO equations also exist for discrete and binary PSO. The flowchart in Fig. 4.1 shows the PSO algorithm.

4.2.1. Parameters. PSO equation consists of three parameters. These c_1 and c_2 are called cognitive and social acceleration constants and help to guide the particles towards the *gbest*. These constant are equal and have the values from 0 to 2 but studies have shown their values set to 2 gives the best results. Another parameter of PSO is w called the inertia weight. Value of w ranges from 0.2 to 1.2 but is usually considered to be 0.8 for best results in case of Constrained PSO (CPSO). For unconstrained PSO, w is linearly decreasing from 0.9 to 0.4 over iterations.

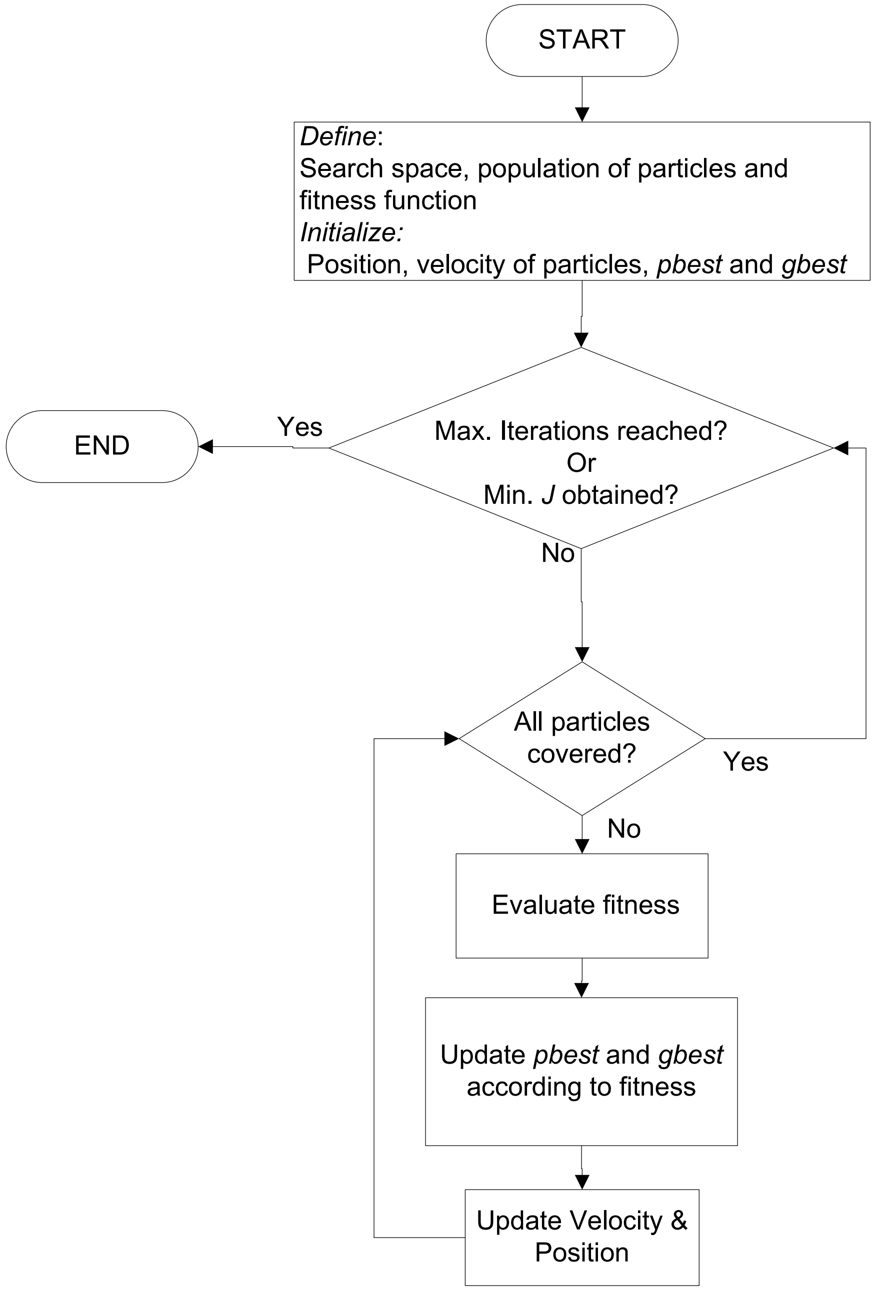

Figure 4.1. Flowchart for PSO

4.2.2. Topologies. Social Interaction is the key factor in the success of PSO. Individual particles in the population learn from their own and their neighbors past experiences to get to a better fitness value. There are three different topologies for PSO. These control the flow of information within the network. Particles in PSO are capable of communicating within the network but information outside the network is inaccessible. The type of network is determined by the following three topologies:

Star Topology: In this topology, each particle is connected with every other particle in the population and shares information among all. It is also called as the global version of PSO and hence the particle which performs the best in the swarm has an influence over the entire population. This is shown in the Fig. 4.2.

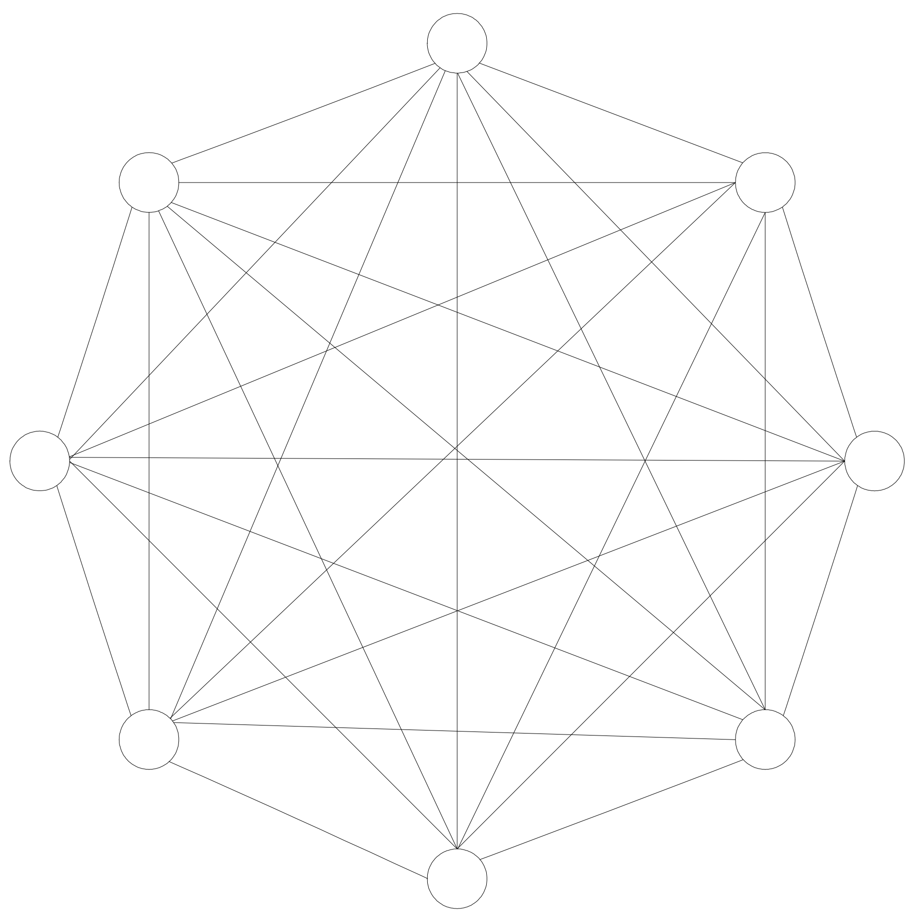

Figure 4.2. Star Topology

Ring Topology: In this topology, each particle is connected with its immediate neighbors and information sharing is only between the neighbors. This topology is used in the local PSO where *lbest* is considered instead of *gbest*, which is the best position among the neighbors. This is shown in Fig. 4.3.

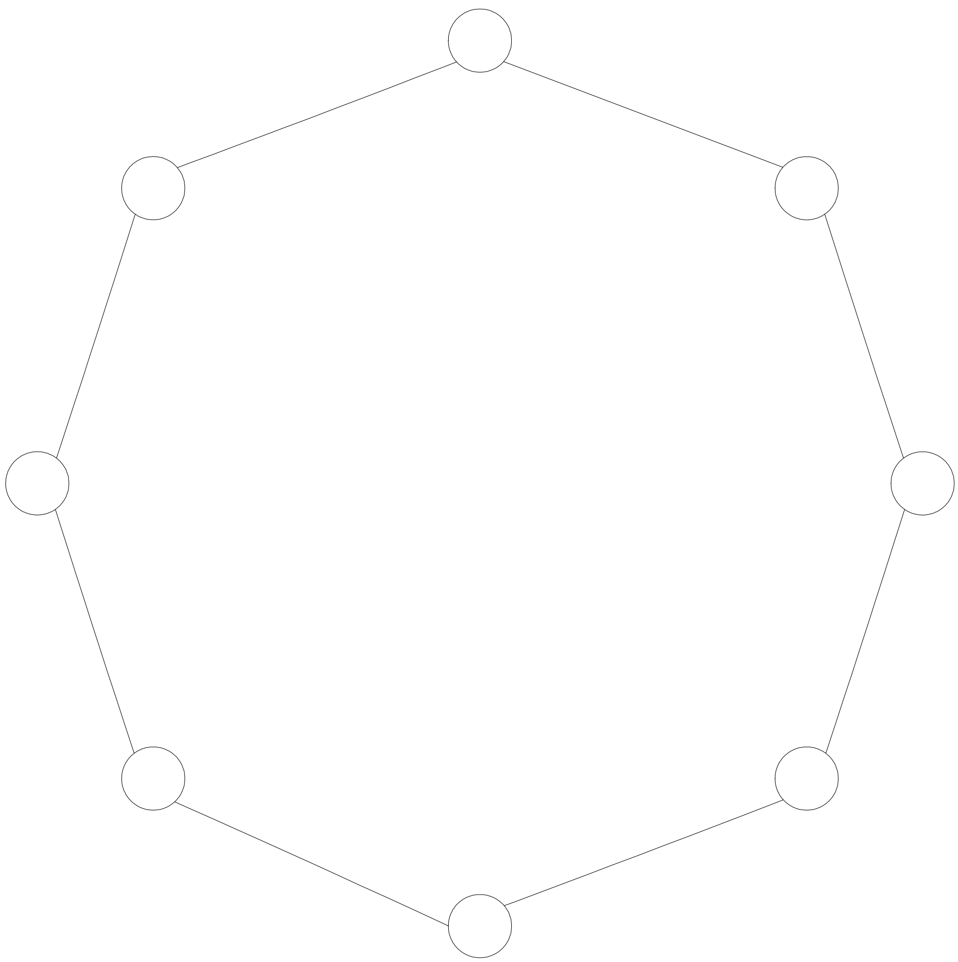

Figure 4.3. Ring Topology

Wheel Topology: In this topology, one particle is connected with all of the other particles while the rest of the particles are not connected to each other directly. The information sharing takes place through the node, which is the center of the wheel. This is shown in Fig. 4.4.

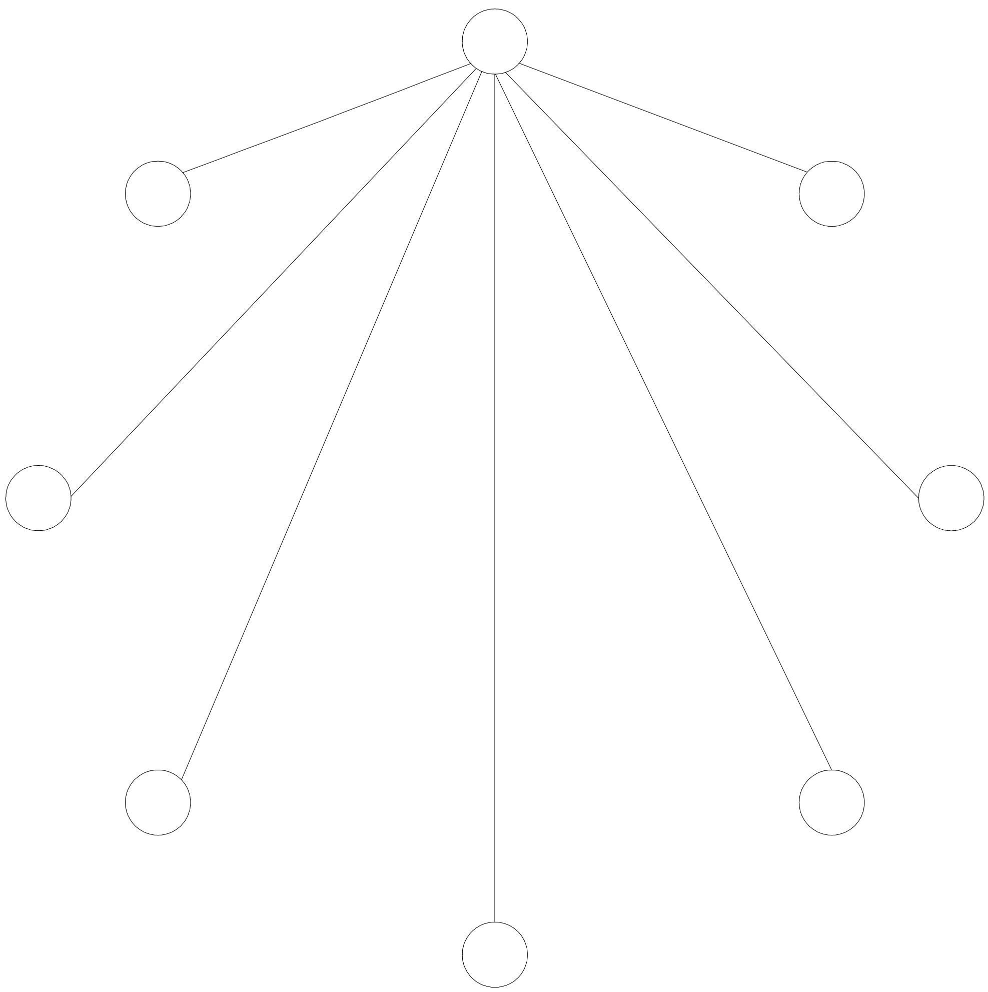

Figure 4.4. Wheel Topology

4.3. MODIFIED PARTICLE SWARM OPTIMIZATION

In (Cai et. al., 2007), hybrid PSO-EA has been explained. It uses selection and mutation operations on the PSO and thus combines the co-operative and competitive characteristics of PSO and EA. In this algorithm, half of the population selected as winners based on their fitness are copied and mutated where as the losers are discarded in each generation. EA loses the valuable search information from its population by discarding the particles, which PSO maintains. Thus PSO-EA combines the advantages of information sharing from PSO with enhanced elites of EA. This algorithm has been used in the thesis to show a comparison among the hybrid algorithms. The flowchart for PSO-EA is shown in Fig. 4.5.

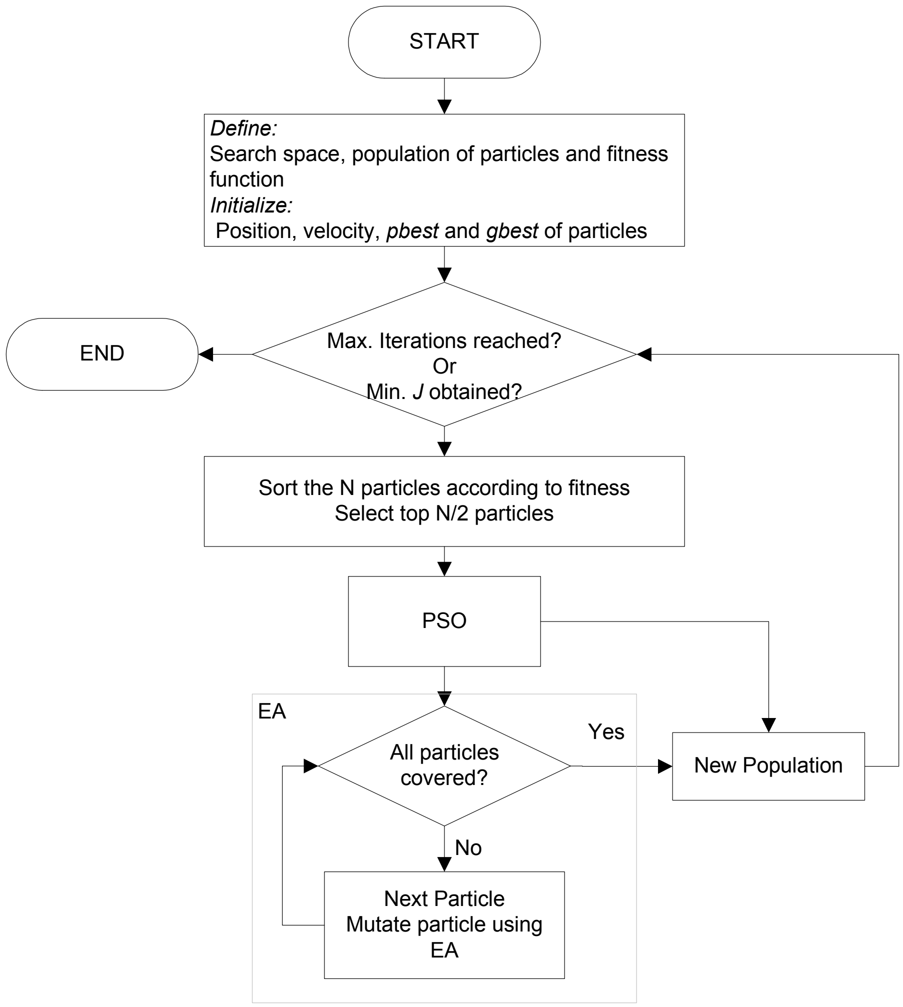

Figure 4.5. Flowchart for PSO-EA

4.4. SUMMARY

In this chapter, PSO and its applications in system identification and digital filter design was presented. Although PSO is effective in solving a lot of optimization problems, it still suffers from premature convergence and thus getting stuck in local minima before reaching the global minimum. To overcome these shortcomings of PSO, different enhancements are brought and hence PSO has deviated a lot from its initial form of classical PSO. A form of modified PSO was also discussed in this chapter. Different modifications of PSO and its hybrid with other algorithms will be discussed in the rest of the chapters.

5. DIFFERENTIAL EVOLUTION PARTICLE SWARM OPTIMIZATION

5.1. INTRODUCTION

Due to the limitations of PSO in finding the best solution, different other approaches were also considered. Over the past few years, research in the field of computational intelligence gave birth to a number of different approaches. All of these algorithms had some special features in finding the best solutions, either their convergence speed or their ability to find the better solution. However, they suffered from one or the other problems. In order to overcome these shortcomings and utilize their effective best properties, hybrid algorithms were introduced. Hybrid algorithms take the best features of the individual algorithms and thus tend to be more effective than the individual algorithms. DEPSO is one of such hybrid algorithms. In this chapter, DEPSO and its applications in system identification and digital filter design is discussed.

5.2. DEPSO ALGORITHM

DEPSO is the hybrid of DE and PSO.

5.2.1. Differential Evolution. Differential Evolution was introduced by Storn and Price in 1995 (Storn, 1996). It is also a population based stochastic search technique for function minimization. In (Storn, 1996), DE has been applied in the field of filter design. In DE, the weighted difference between the two population vectors is added to a third vector and optimized using selection, crossover and mutation operators as in GA. Each individual is first mutated according to the difference operation. This mutated individual, called the offspring, is then recombined with the parent under certain criteria such as crossover rate. Fitness of both the parent and the offspring is then calculated and the offspring is selected for the next generation only if it has a better fitness than the parent (Karaboga, 2005). The mutation takes place according to (14).

$$If\,(rand() < CR \quad OR \quad d == k \;\; then \quad T_{id}(i) = P_{pd}(i) + \delta_{2,d} \tag{14}$$

$$\delta_{2,d} = \frac{(P_{1,d} - P_{2,d}) + (P_{3,d} - P_{4,d})}{2} \tag{15}$$

where $\delta_{2,d}$ is the weighted error in different dimensions, $T_{id}(i)$ is the i^{th} offspring and $P_{pd}(i)$ is the *pbest* position of the i^{th} parent.

5.2.2. DEPSO. Differential evolution particle swarm optimization is a stochastic search technique utilizing the hybrid of the particle swarm optimization and the differential evolution (Zhang and Xie, 2003). In DEPSO, new offspring is created by the mutation of the parent. In this work, both *gbest* and *pbest* have been taken as the parent for different applications and a Gaussian distribution is considered (Moore and Venayagamoorthy, 2006). For mutation, 4 particles are randomly chosen from the population. The weighted error between these particles' *pbest* positions is used to mutate the parent and create an offspring. The mutation takes place under the condition when a random number between [0,1] is less than the crossover rate CR or the particle's position in any one randomly chosen dimension, k, is mutated. This ensures that offspring is never the same as the parent. Then the fitness of the offspring is evaluated and the offspring replaces the parent only if it has a better fitness than the parent, otherwise the parent is retained for the next iteration (Zhang and Xie, 2003). In (Hao et al., 2007), different scheme for mutation in DEPSO is also proposed where position update in PSO is carried out either in canonical PSO way or in DE way depending upon the crossover rate. The flowchart in Fig. 5.1 shows the DEPSO algorithm.

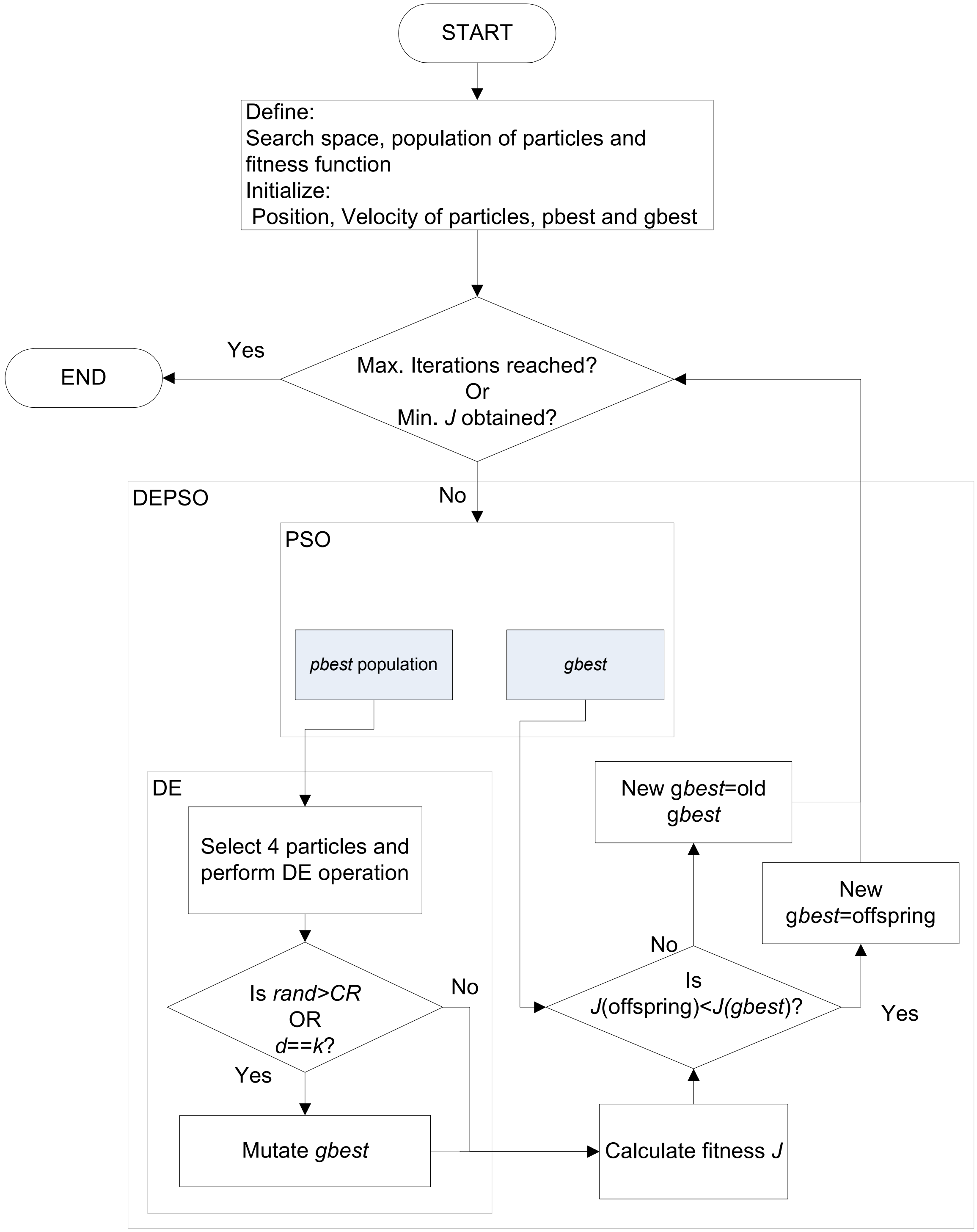

Figure 5.1. Flowchart for DEPSO

5.3. SUMMARY

In this chapter, a hybrid algorithm was described and implemented in system identification and in the design of FIR and IIR filters. Two different approaches are explained for DEPSO where either only the *gbest* is taken as a parent and mutated or all of the *pbest* particles are taken as parents and each creates its own offspring, and goes through a tournament with the offspring. Thus it is shown that DEPSO can only be either equal to or better than PSO in its performance. The results to support the claim are presented in Chapters 7 and 8.

6. PARTICLE SWARM OPTIMIZATION WITH QUANTUM INFUSION

6.1. INTRODUCTION

In this chapter, another hybrid algorithm called PSO-QI is introduced which combines QPSO and PSO. It utilizes the principal of quantum mechanics to improvise the PSO algorithm and thus gain better solution. It has been applied in the design of FIR and IIR filters, and has outperformed other two algorithms in the results.

6.2. PSO-QI ALGORITHM

PSO-QI is a new algorithm developed by combining QPSO (Sun et. al., 2004(a)) with PSO. QPSO is improved from QDPSO where particles position in the search space is updated using the quantum mechanics.

6.2.1. QDPSO. According to the uncertainty principle, position and velocity of a particle in quantum world cannot be determined simultaneously. Thus quantum behaved PSO differs from traditional PSO mainly in the fact that exact values of *x* and *v* cannot be determined. In quantum mechanics, a particle, instead of having position and velocity, has a wavefunction given by:

$$\psi(r,t) \tag{16}$$

which has no physical meaning but its amplitude squared gives the probability measure of its position in any one dimension *r* at time *t*. The governing equation of quantum mechanics is the Schrodinger's equation given by:

$$j\hbar\frac{\partial}{\partial t}\psi(r,t) = \hat{H}(r)\psi(r,t) \tag{17}$$

where *H* is a time-independent Hamiltonian operator given by:

$$\hat{H}(r) = -\frac{\hbar^2}{2m}\nabla^2 + V(r) \tag{18}$$

where $\hbar$ is Planck's constant, *m* is the mass of the particle and *V(r)* is the potential energy distribution (Mikki and Kishk, 2006). Based on the probability density function, a particle's probability of appearing in position *x* can be determined. Therefore in QDPSO, a Delta-potential-well based probability density function has been used with center at point $P = (p_1, p_2, .., p_d)$ in order to avoid explosion and help the particles in PSO to converge (Sun et al., 2004(b)). Assuming a particle in one-dimensional space having its center of potential at *P*, normalized probability density function *Q* and distribution function D_f can be obtained (Sun et al, 2005). Let $y=x-p$, then the form of this probability density function is given as follows and depends on the potential field the particle lies in

$$Q(y) = \frac{1}{L} e^{-2|y|/L} \tag{19}$$

$$D_f(y) = \int_{-\infty}^{y} Q(y)dy = e^{-2|y|/L} \tag{20}$$

where the parameter L is the length of the potential field which depends on the energy intensity and is called the creativity or imagination of the particle that determines the search scope of each particle (Sun et al., 2004(b)).

In QDPSO, the search space and the solution space are two different spaces of different qualities. So a mapping mechanism is necessary to interpret the position of a particle in solution space by looking at its position in quantized search space. This is called the collapse and is achieved by applying the Monte Carlo simulation. This is explained in (Sun et al., 2004(a)) as follows.

Let *s* be any random number uniformly distributed between *0* and $1/L$. For a random number $u=rand(0,1)$, *s* is defined as:

$$s = \frac{1}{L} u \tag{21}$$

Now, equating (19) and (21), we get:

$$u = e^{-2|y|/L} \tag{22}$$

$$y = \pm \frac{L}{2} \ln(1/u) \tag{23}$$

As y=x-p, it results in the position equation as shown (Zhang and Xie, 2003):

$$x = P \pm \frac{L}{2} \ln(1/u) \tag{24}$$

where the particle's local attractor point $P = (p_1, p_2, .., p_d)$ has coordinates given by the following equation:

$$P_d = \alpha_1 p_{gd} + \alpha_2 p_{id} \tag{25}$$

where $\alpha_1 = a/(a + b)$ and $\alpha_2 = b/\ (a + b)$, and *a* and *b* are two uniformly distributed random numbers and *L* can be evaluated as the distance between the particles' current position and point *P* as follows:

$$L = 2\beta .| P - x | \tag{26}$$

From (24) and (26), the new position of the particle is calculated as:

$$x(k+1) = P \pm \beta .| P - x(k) | .\ln(1/u) \tag{27}$$

The parameter β is the only parameter of the algorithm. It is called the creativity coefficient and is responsible for the convergence speed of the particle. The term *u* is a uniformly distributed random number. This Delta-Potential-well based quantum PSO is called the QDPSO.

6.2.2. QPSO. An improvement to it is brought by defining a mainstream thought or the Mean Best Position, *mbest*, defined in (Sun et al., 2005) as:

$$mbest = \frac{1}{M}\sum_{i=1}^{M} p_i = \left(\frac{1}{M}\sum_{i=1}^{M} p_{i1},\ldots,\frac{1}{M}\sum_{i=1}^{M} p_{iD}\right) \tag{28}$$

where *M* is the size of the population, *D* is the number of dimensions and pi is the *pbest* position of each particle. Now the positions update equation in (29) can be written as:

$$x = P \pm \beta.|\,mbest - x\,|\,.\ln(1/u) \tag{29}$$

By using (25), this can also be written as follows to show the mutation on *gbest*:

$$x(k+1) = \alpha_1 p_{gd} + [\alpha_2 p_{id} \pm \beta.|\,mbest - x(k)\,|\,.\ln(1/u)] \tag{30}$$

The pseudocode for the QPSO algorithm is written as follows:

Initialize x, pbest and gbest of the particles.

Do

For i from 1 to population size

evaluate fitness

If fitness (x)<fitness (pbest)

pbest=x

gbest=min(pbest)

Calculate mbest

For d from 1 to dimension size

r_1*=rand(0,1)*

r_2*=rand(0,1)*

$P=(r_1*p_{id}+r_2*p_{gd})/(r_1+r_2)$

r_3*=rand(0,1)*

$L=\beta*abs(mbest-x_{id})$

If rand(0,1)>0.5

$x_{id}=P-L*ln(1/r_3)$

else

$x_{id}=P+L*ln(1/r_3)$

end

While termination criteria not met

6.2.3. PSO-QI. Particle swarm optimization with quantum infusion is a novel approach to hybridization of PSO and QPSO. Here, the quantum theory based on QPSO has been used to create a new offspring. After the position and velocity of the particles are updated using PSO, a randomly chosen particle from the *pbest* population is utilized to do the quantum operation as described in QPSO algorithm and thus create an offspring by mutating the *gbest*. The fitness of the offspring is evaluated and the offspring replaces the *gbest* particle of PSO only if it has a better fitness. This ensures that the fitness of the *gbest* particle is equal to or better than its fitness in the previous iteration. Thus, it gets improved and pulled towards the best solution over iterations. By infusing the quantum theory to the traditional PSO, a new hybrid algorithm is obtained which incorporates the best features of both participating algorithms and thus achieve better performance. In PSO-QI, fast convergence obtained by PSO which is the rate of convergence for first few iterations, and the lower value of average error obtained by QPSO, have been utilized and hence the performance has significantly improved, as is seen in the results and figures. PSO-QI is shown in the flowchart in Fig. 6.1.

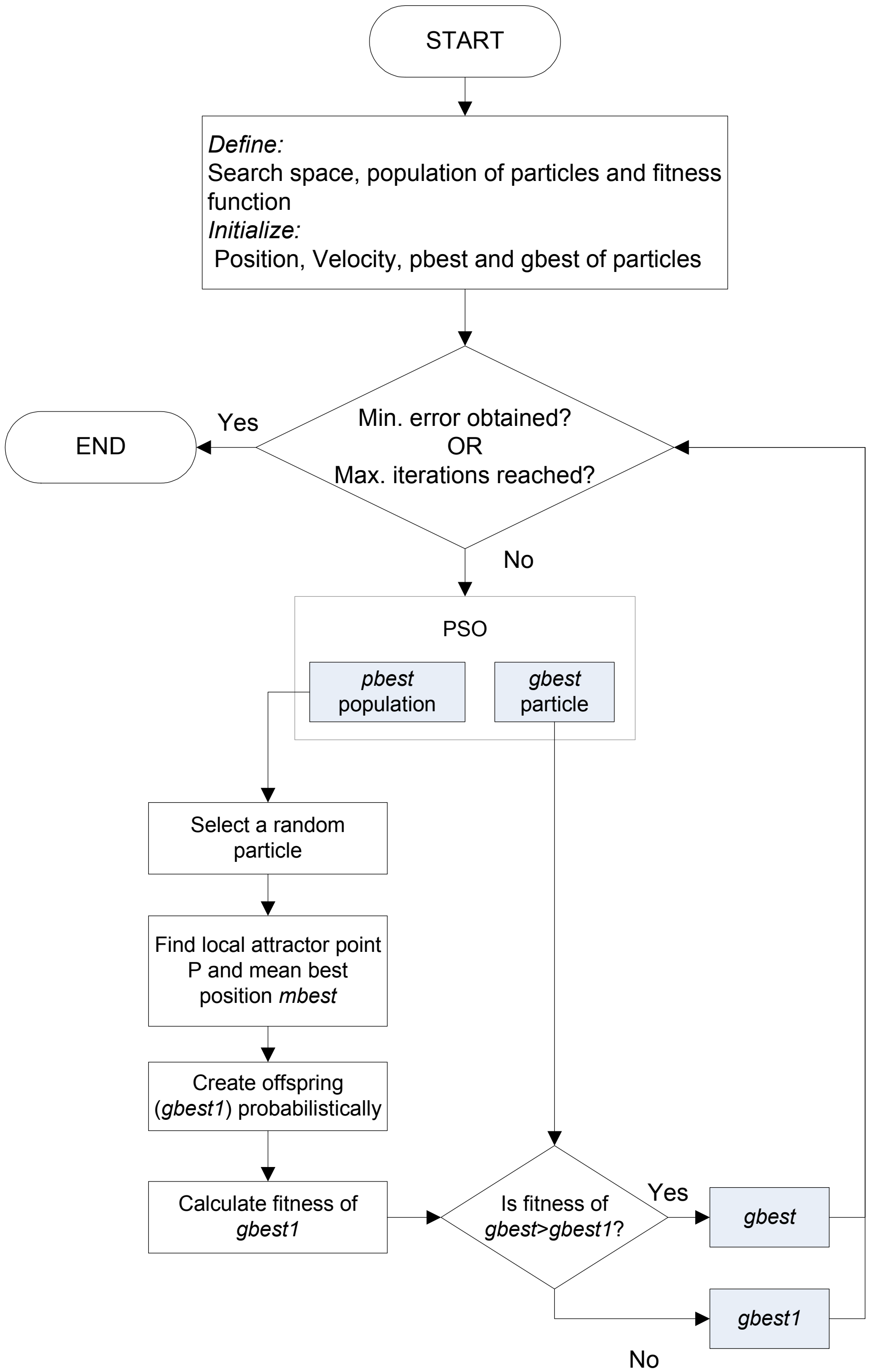

Figure 6.1. Flowchart for PSO-QI

6.3. SUMMARY

In this chapter, PSO-QI and its applications were discussed. It is observed that application of quantum mechanics in the swarm behavior could also produce useful results and provide yet another method of stochastic search in a multi-dimensional space. Further, by combining the quantum based swarm optimization technique to traditional PSO, even better results were obtained in real world application. The results of the case studies using PSO-QI are presented in Chapter 8.

7. SYSTEM IDENTIFICATION USING PSO, PSO-EA, DEPSO AND PSO-QI

7.1. INTRODUCTION

PSO, PSO-EA, DEPSO and PSO-QI are the four algorithms used in the case studies of system identification. PSO forms the basis of comparison for other algorithms, as it gets stuck in local minima and cannot converge to the best solution all of the time. So, this research work introduces two hybrid algorithms- DEPSO and PSO-QI which perform better than PSO and can better approximate the coefficients of the given IIR system. A modified PSO using the hybrid of PSO and EA is also compared against PSO and DEPSO. Two different models are studied in implementing these algorithms for the identification of an IIR system. The study is carried out in two different scenarios. In the first scenario, PSO, PSO-EA and DEPSO are used and DEPSO is shown as the best performer among the three. In the second scenario, PSO, DEPSO and PSO-QI have been used and it is shown that the new hybrid algorithm PSO-QI performs the best in terms of both consistency and minimum error achieved.

The study is carried out for 6 different benchmark systems. The Table 7.1 below shows the parameters used in the study. Table 7.2 trough 7.5 show the six different cases studied for the full and reduced order models.

Table 7.1. Parameters used in the study

Symbol	Parameter Description	Value
P	Population Size	25
c_1	Cognitive constant	2
c_2	Social constant	2
w	Inertia weight	Linearly decreasing from 1.4 to 0
V_{max}	Maximum Velocity	1.3
X_{max}	Maximum Position	1.3
β	Creativity Coefficient	Linearly increasing from 0.5 to 1

Table 7.1. (cont.) Parameters used in the study

CR	crossover rate	0.8
N	Number of Inputs	50
	Number of Iterations	500, 50
	Number of Trials	50
	Noise Introduced	-30dB

Table 7.2. Study of Cases I and II

		Case I (Krusienski and Jenkins, 2004)	Case II (Ng et al., 1996)
Transfer Function		$\frac{1.25z^{-1}-0.25z^{-2}}{1-0.3z^{-1}+0.4z^{-2}}$	$\frac{-0.2z^{-1}-0.4z^{-2}+0.5z^{-3}}{1+0.6z^{-1}-0.25z^{-2}+0.2z^{-3}}$
Full Order	L	2	3
	M	1	2
	Model	$\frac{b_0+b_1z^{-1}}{1+a_1z^{-1}+a_2z^{-2}}$	$\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}}$
Reduced Order	L	1	2
	M	0	1
	Model	$\frac{b_0}{1+a_1z^{-1}}$	$\frac{b_0+b_1z^{-1}}{1+a_1z^{-1}+a_2z^{-2}}$

Table 7.3. Study of Case III

		Case III (Shynk, 1989(b))
Transfer Function		$\dfrac{z^{-1} - 0.9z^{-2} + 0.81z^{-3} - 0.729z^{-4}}{1 - 0.04z^{-1} - 0.2775z^{-2} + 0.2101z^{-3} - 0.14z^{-4}}$
Full Order	L	4
	M	3
	Model	$\dfrac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}$
Reduced Order	L	3
	M	2
	Model	$\dfrac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}$

Table 7.4. Study of Case IV

		Case IV (Karaboga, 2005)
Transfer Function		$\frac{1-0.4z^{-2}-0.65z^{-4}+0.26z^{-6}}{1-0.77z^{-2}-0.8498z^{-4}+0.6486z^{-6}}$
Full Order	L	6
	M	6
	Model	$\frac{b_0+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+b_4z^{-4}+b_5z^{-5}+b_6z^{-6}}{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_4z^{-4}+a_5z^{-5}+a_6z^{-6}}$
Reduced Order	L	5
	M	5
	Model	$\frac{b_0+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+b_4z^{-4}+b_5z^{-5}}{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_4z^{-4}+a_5z^{-5}}$

Table 7.5. Study of Cases V and VI

		Case V (Karaboga et al., 2004)	Case VI (Karaboga et al., 2004)
Transfer Function		$\frac{1}{1-1.2z^{-1}+0.6z^{-2}}$	$\frac{0.05z^{-1}-0.4z^{-2}}{1-1.1314z^{-1}+0.25z^{-2}}$
Full Order	L	2	2
	M	0	1
	Model	$\frac{b_0}{1+a_1z^{-1}+a_2z^{-2}}$	$\frac{b_0+b_1z^{-1}}{1+a_1z^{-1}+a_2z^{-2}}$

7.2. APPLICATION IN SYSTEM IDENTIFICATION

In this thesis, system identification is carried out using different algorithms. As already discussed, system identification is a complex optimization process and hence CI techniques can be effectively and efficiently used for identifying complex dynamic systems. Most nonlinear systems are also recursive in nature. Hence, models for real world systems are better represented as IIR systems. By doing so, the problem of system identification now becomes the problem of adaptive IIR filtering, for which different adaptive algorithms can be applied for adjusting the feed forward and feedback parameters of the recursive system. PSO, PSO-EA, DEPSO and PSO-QI have been used in the identification of various IIR systems. Fig. 7.1 shows the block diagram for system identification.

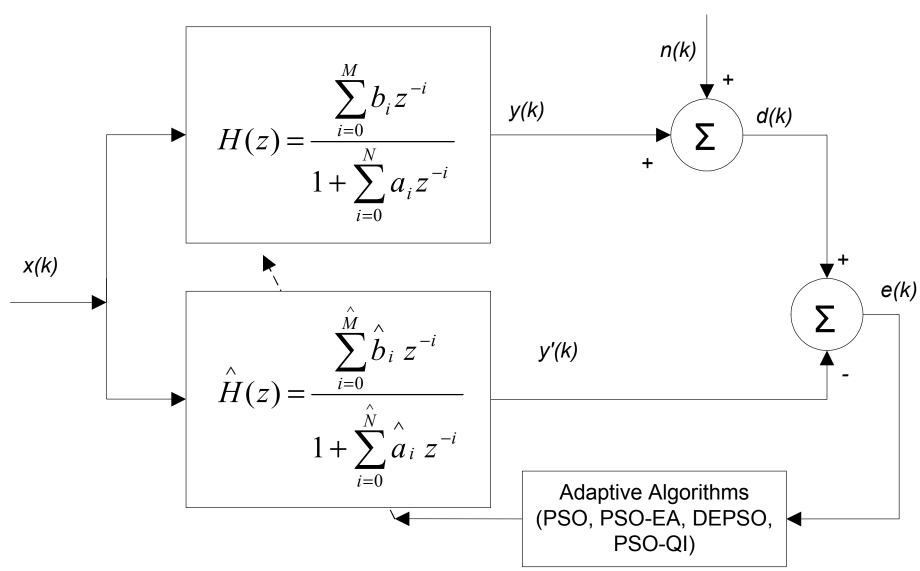

Figure 7.1. System identification block diagram

An IIR system can be represented by the transfer function:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}} \tag{31}$$

where *m* and *n* are the number of numerator and denominator coefficients of the transfer function and a_n and b_m are the pole and zero parameters of the IIR filter. This can be written as difference equation of the form (Krusienski and Jenkins, 2005; Karaboga, 2005):

$$y(k) = \sum_{n=1}^{L} a_n(k) y(k-n) + \sum_{n=0}^{M} b_n(k) x(k-n) \tag{32}$$

where *x(k)* and *y(k)* represent the k^{th} input and output of the system. Also, *n = 1, 2, 3, ….., L* and *n = 0, 1, 2, ……, M* represent the coefficients of the IIR filter. Considering the block diagram of Fig. 1, the output *y(k)* for input *x(k)* to the system is mixed with a noise signal *n(k)*. The output of the plant added with the noise gives the final system output *d(k)*. On the other hand, the output of the IIR filter in the modeled system for the same input *x(k)* has an output of *y'(n)*. The difference of the output from the actual system with that of the modeled system gives the error *e(k)*. This error is used by the adaptive algorithm to adjust the parameters of the IIR filter, and thus reduce the error in a number of iterations so as to exactly identify the actual system. This has been shown in the following equations:

$$d(k) = y(k) + n(k) \tag{33}$$

$$e(k) = d(k) - y'(k) \tag{34}$$

For the identification of the system, the adaptive algorithm tries to minimize the error *e(k)* by adjusting the parameters of the modeled system, which are the pole-zero coefficients in case of an IIR system. The different kinds of algorithms that can be used for error minimization in adaptive systems are explained in (Netto et al., 1995). In this paper, Mean Squared Error (MSE) between the output of the actual system and the designed system as given by (35) has been considered as the feedback to the adaptive algorithm.

$$J = \frac{1}{N}\sum_{k=1}^{N}(d(k) - y'(k))^2 \tag{35}$$

The fitness function used by the different algorithms that are illustrated in the paper is given by:

$$Fitness = \frac{1}{1+J} \tag{36}$$

The numerator and denominator coefficients of the IIR filter are represented by *D* dimensions (*D* = *L*+*M*). In (Karaboga, 2005), DE has been used for adjusting the parameters of the IIR system to reduce the MSE or to increase the fitness of the system. The application of system identification is illustrated in the flowchart in Fig. 7.2.

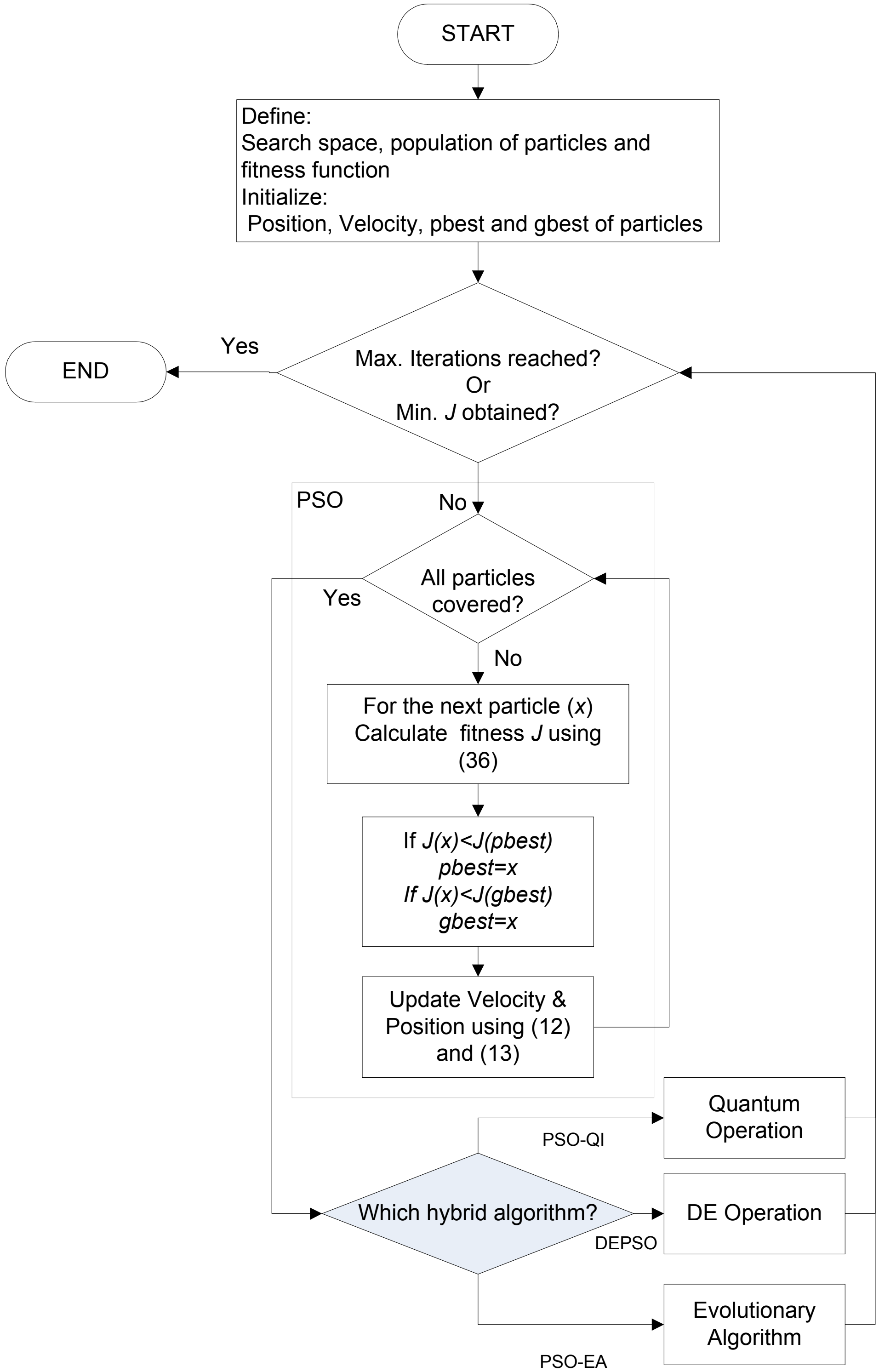

Figure 7.2. Flowchart for the system identification

7.3. RESULTS WITH PSO

Classical PSO with linearly decreasing inertia weight is used in the study. In all of the cases PSO has the poorest performance of the three algorithms. This is because of the tendency of the particles in PSO to get stuck in the local minimum quickly. Since all of the particles are attracted towards the *gbest* particle, which when gets stuck, the particles thus lose diversity. Thus PSO fails to explore the search space extensively.

7.4. RESULTS WITH PSO-EA

The modified form of PSO used in the study is the hybrid of PSO and EA called as PSO-EA. Due to the fact that only elites are selected after mutation of the particles in EA, the best particles are involved in the search in PSO-EA. This leads to faster convergence time of PSO-EA. However, the losing population is then discarded and hence diversity is lost in this algorithm, which leads to the particle getting stuck in local minima. This justifies the ability of PSO-EA to converge quickly and perform better than PSO.

7.5. RESULTS WITH DEPSO

DEPSO is the winning algorithm of the three when compared with PSO and PSO-EA, performing either as good as or better than the other two algorithms. DEPSO shows the ability to converge to the minimum average error in all of the cases while the other two deviate over different trials. Unlike PSO-EA, DEPSO is able to come out of the local minima and reach the global solution every time. The following figures show the results of the study in different cases under the two models.

7.6. RESULTS WITH PSO-QI

PSO-QI is the best performing algorithm in terms of minimum error achieved and consistency of performance when compared with PSO and DEPSO. Although time taken by PSO-QI is longer than that of the other two algorithms, performance of PSO-QI is remarkable.

Figs. 7.3 and 7.4 show the average error graph for the full and reduced order model in Case I. Case I is a second order IIR system. In full order model, DEPSO has quickly overcome the other two algorithms although all of them converged to nearly equal values after a certain number of iterations.

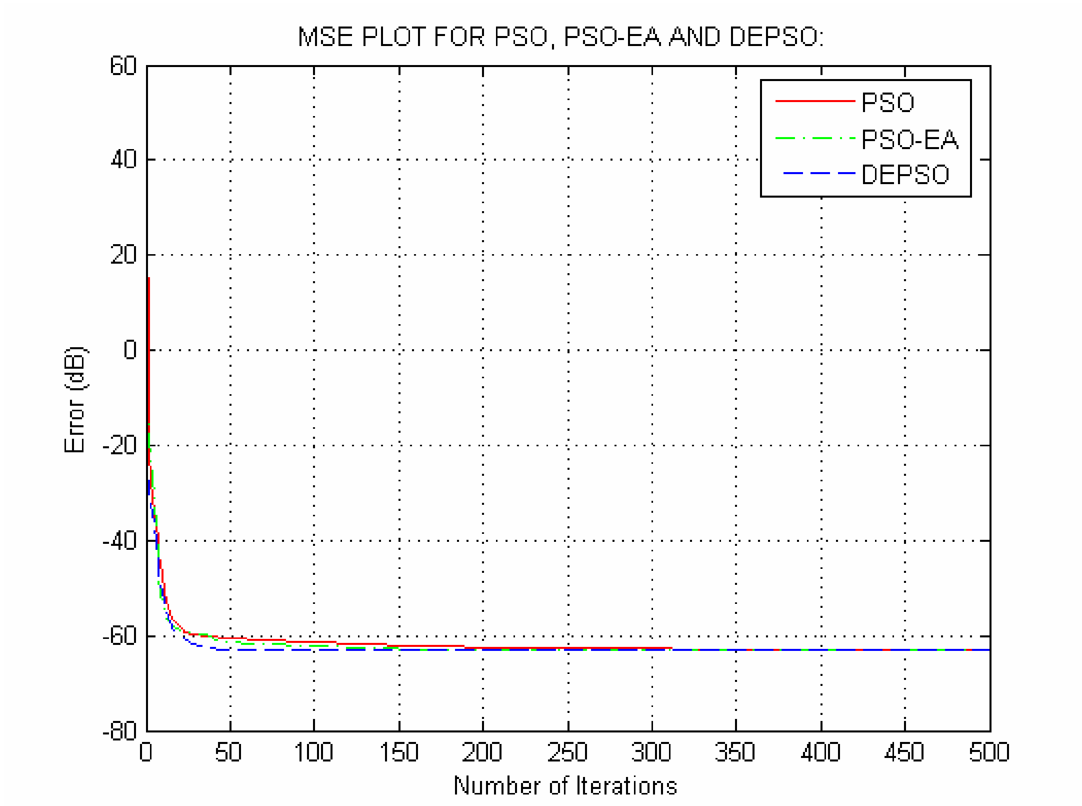

Figure 7.3. Error graph for full order model of Case I

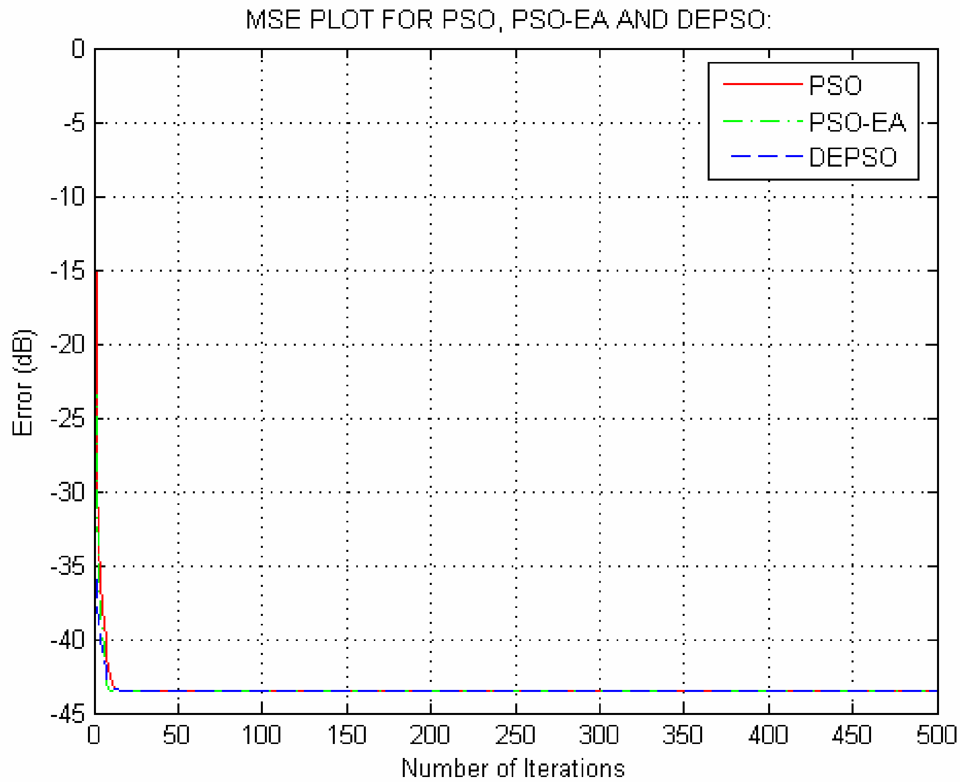

Figure 7.4. Error graph for reduced order model of Case I

The pole-zero plot of the coefficients obtained from PSO, PSO-EA and DEPSO is shown in Fig. 7.5.

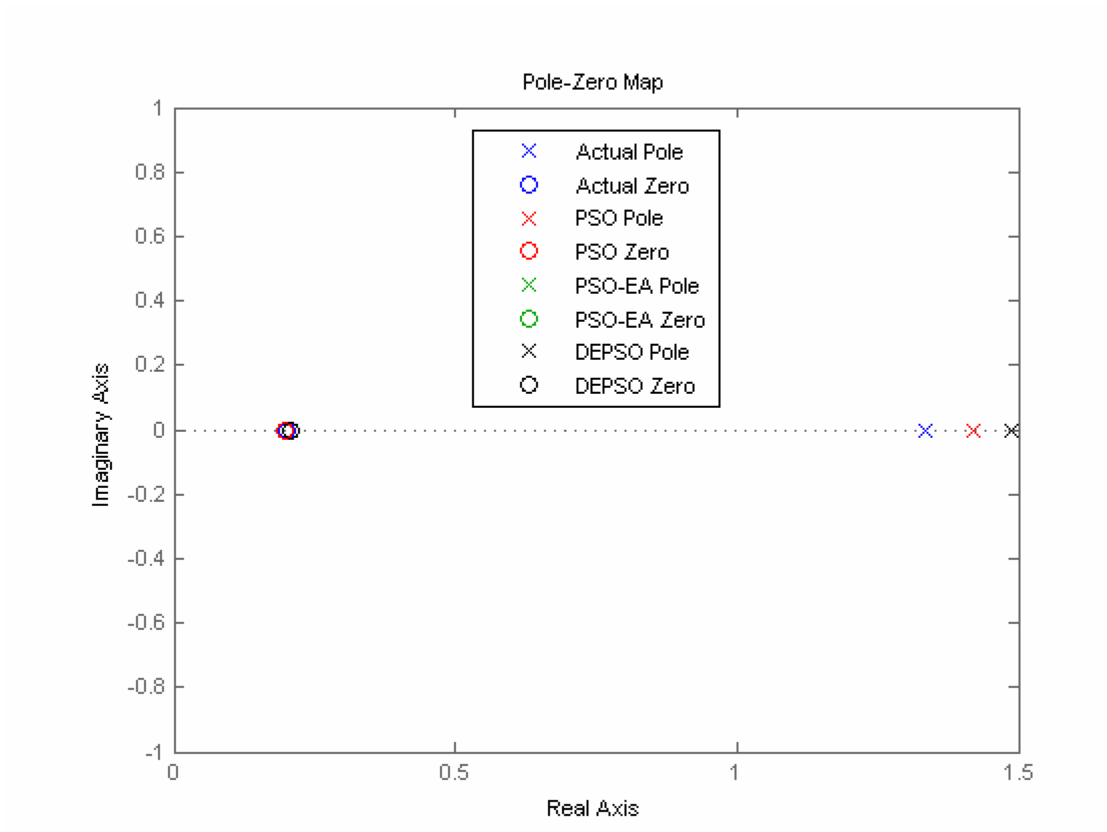

Figure 7.5. Pole zero plot for the full order model of Case I

Figures 7.6 and 7.7 show the error graph for the full and reduced order models of Case II Case II is a third order IIR system and its modeled as a second order system in reduced order. In both models, DEPSO has quickly converged to a much lower value than the other two algorithms.

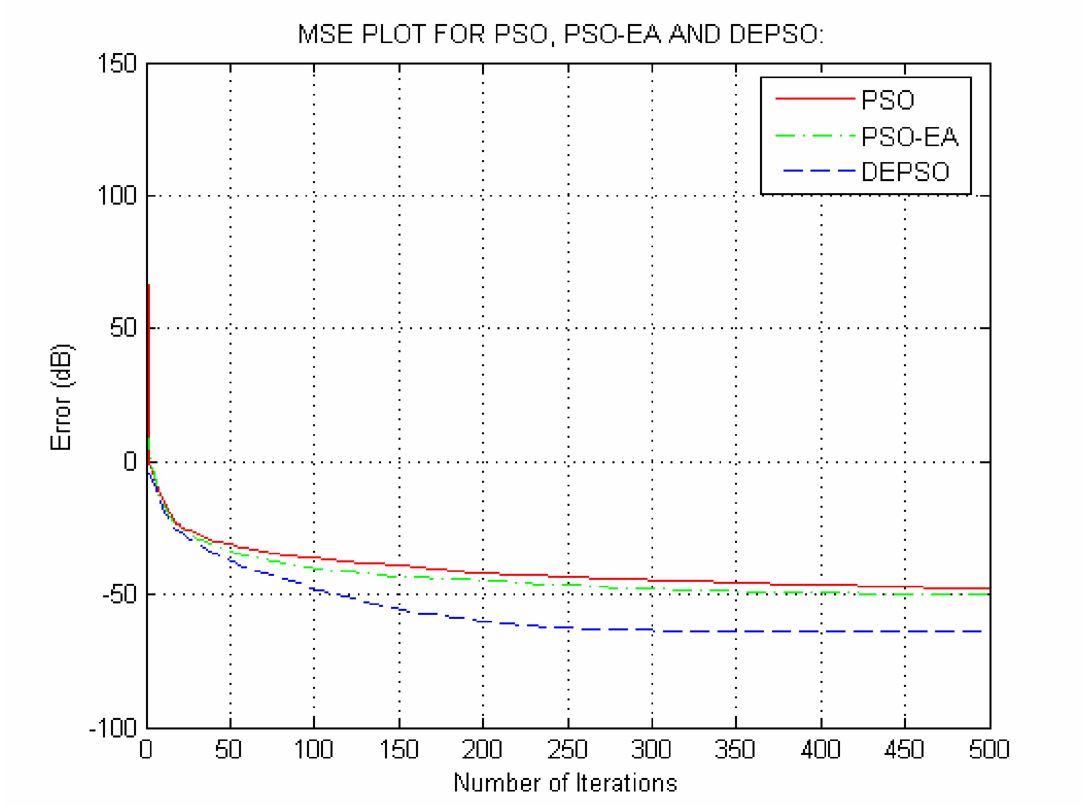

Figure 7.6. Error graph for the full order model of Case II

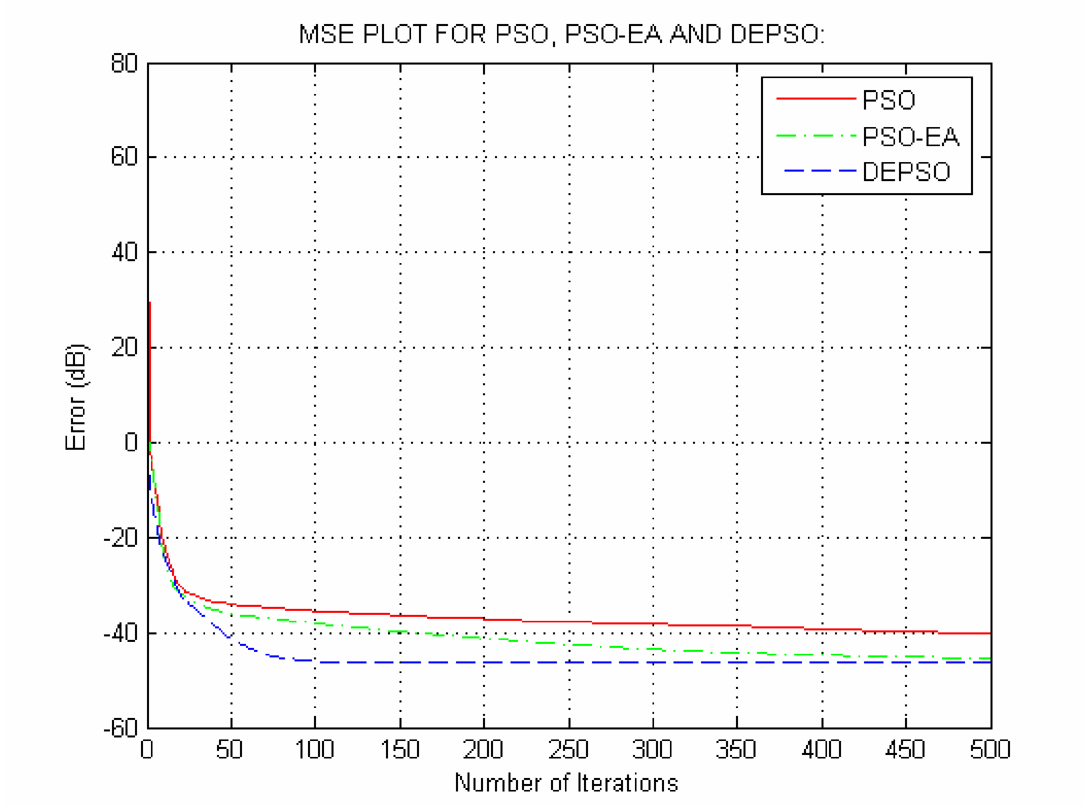

Figure 7.7. Error graph for the reduced order model of Case II

The pole-zero plot for the coefficients obtained from PSO, PSO-EA and DEPSO in Case II is shown in Fig. 7.8.

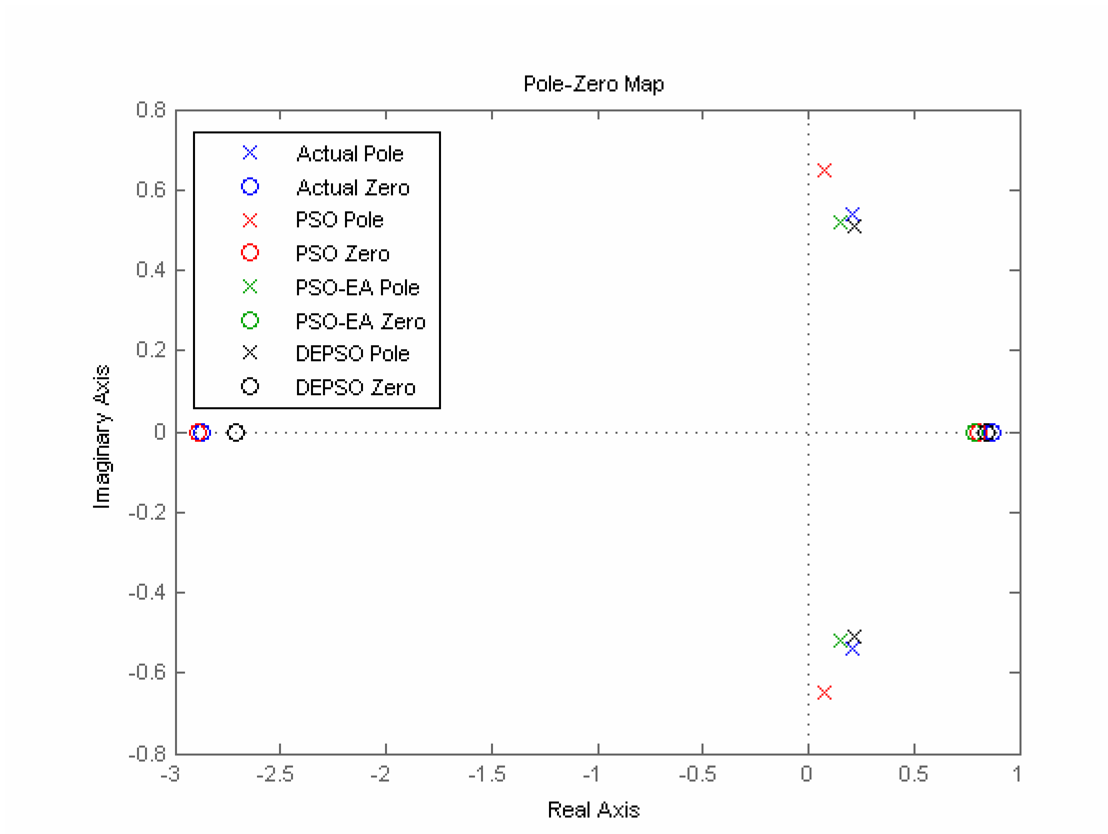

Figure 7.8. Pole zero plot for the full order model of Case II

The full order model of Case III is a fourth order IIR system. It is modeled as a third order system in reduced order. Figures 7.9 and 7.10 show the error graph for the three algorithms. In both the figures, PSO-EA has performed better than PSO where as DEPSO has performed the best. Although PSO-EA has converged to a lower value of error in less iteration, it is then stuck.

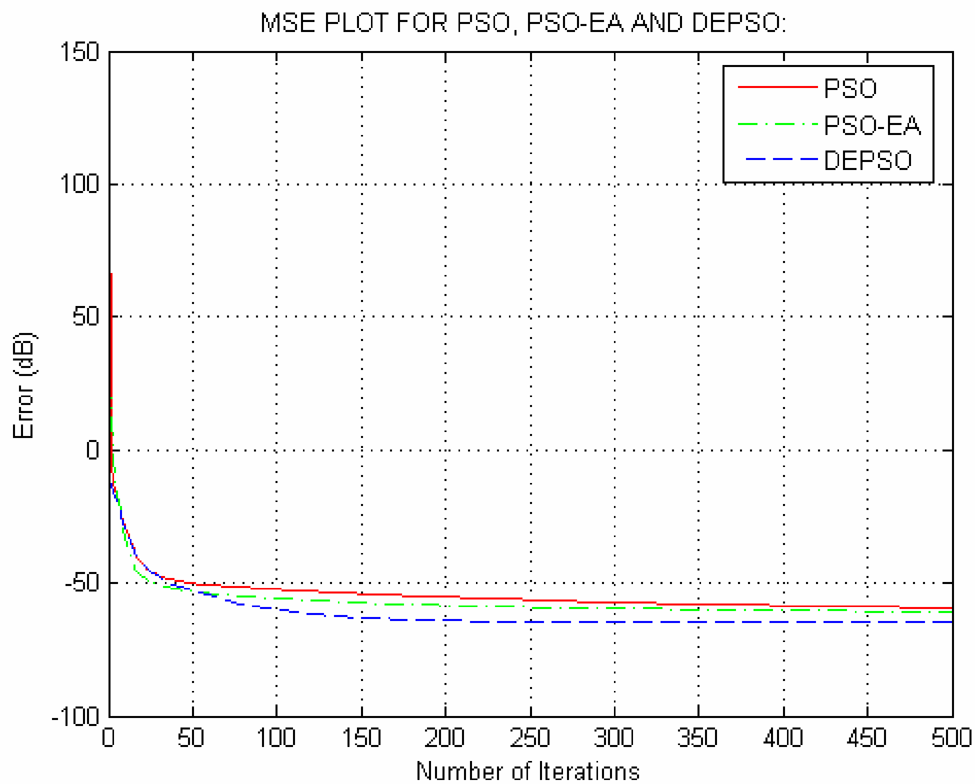

Figure 7.9. Error graph for the full order model of Case III

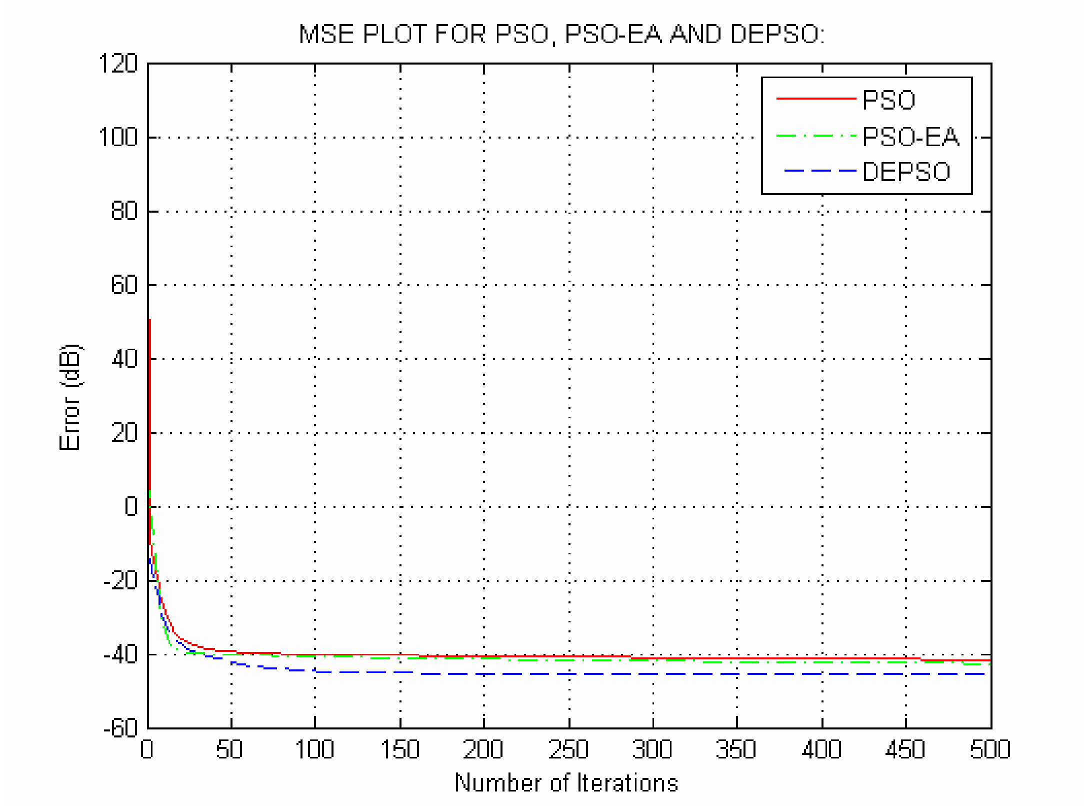

Figure 7.10. Error graph for the reduced order model of Case III

The pole-zero plot of the coefficients obtained from PSO, PSO-EA and DEPSO for the full order model of Case III is shown in Fig. 7.11.

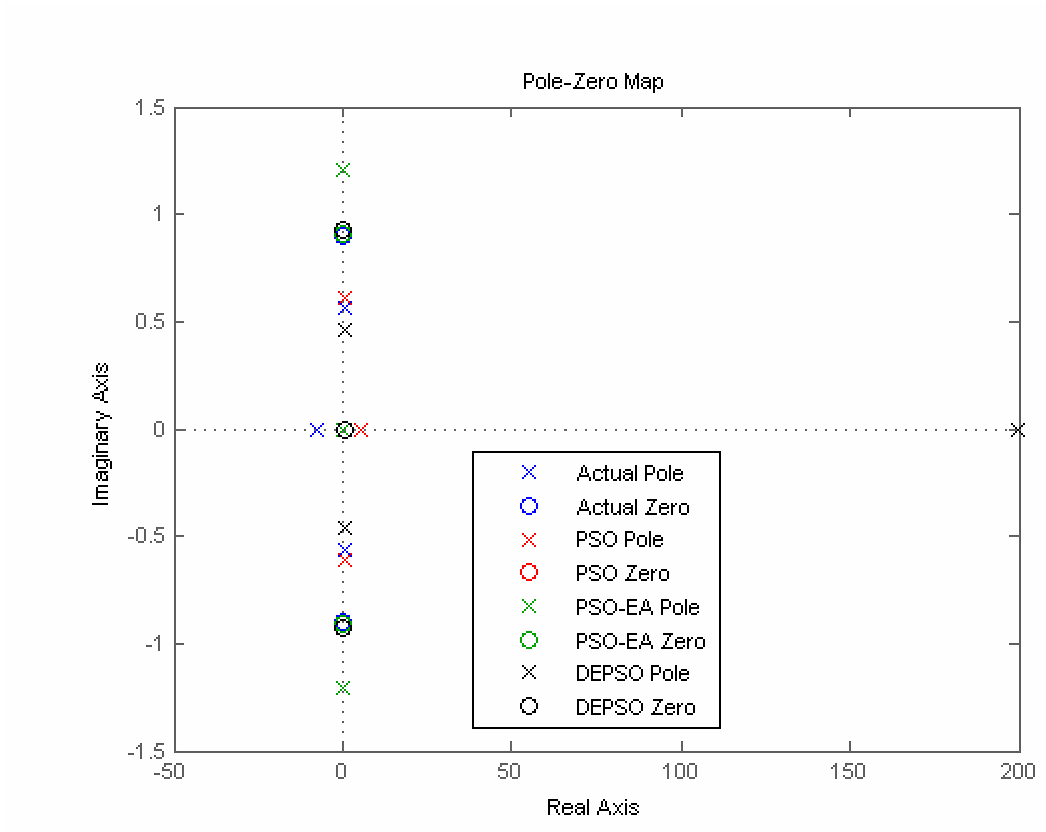

Figure 7.11. Pole zero plot for the full order model of Case III

Error graphs for the full and reduced order models of Case IV are shown in Figs. 7.12 and 7.13. Case IV is a sixth order system modeled as a fifth order system in its reduced order. DEPSO has shown the best result in both the cases. Also observable in the figures is the speed of convergence of PSO-EA.

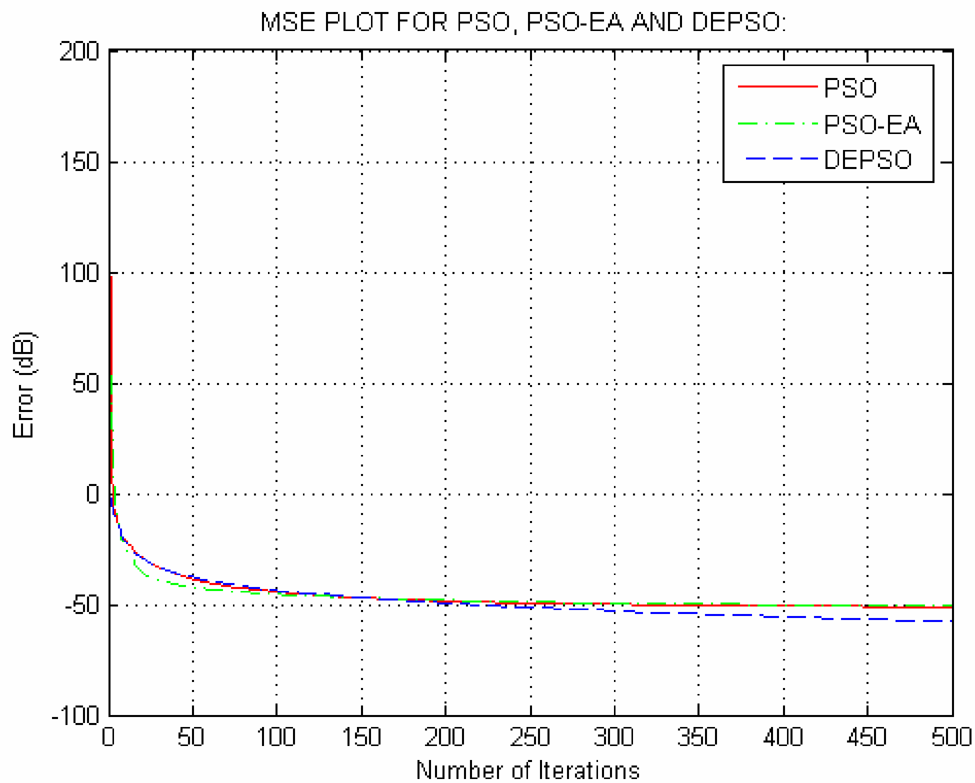

Figure 7.12. Error graph for the full order model of Case IV

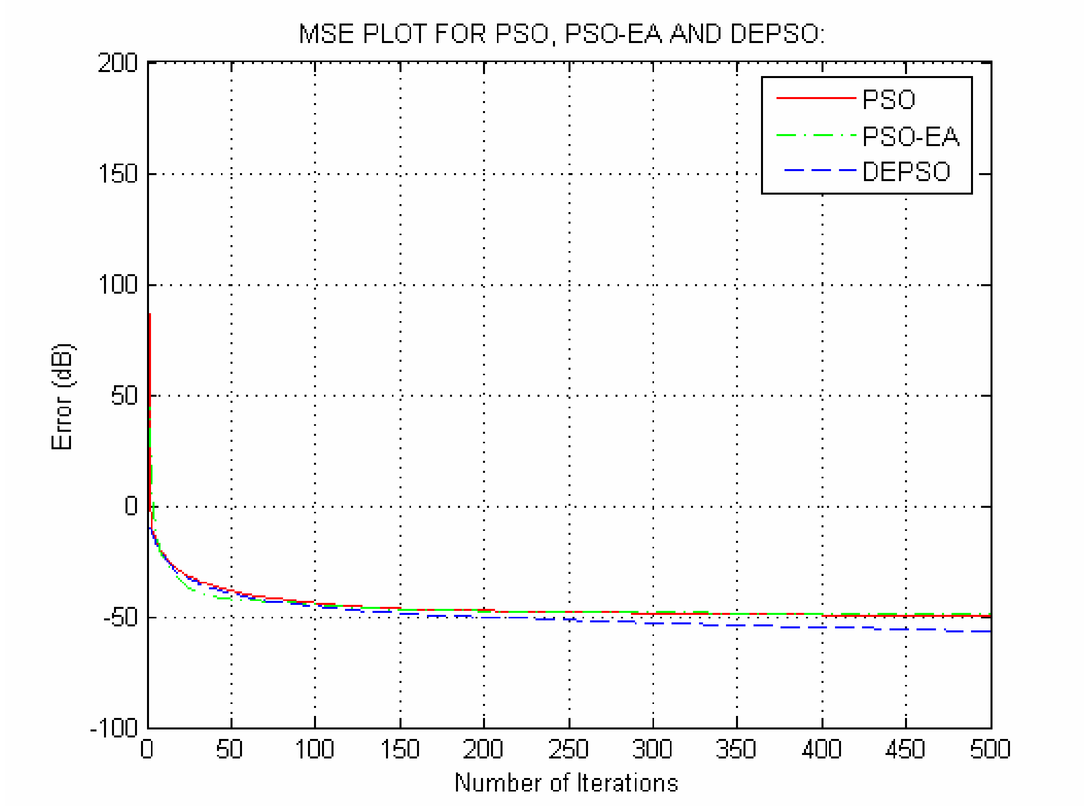

Figure 7.13. Error graph for the reduced order model of Case IV

The pole-zero plot of the coefficients obtained by PSO, PSO-EA and DEPSO from the full order model of Case IV is shown in Fig. 7.14.

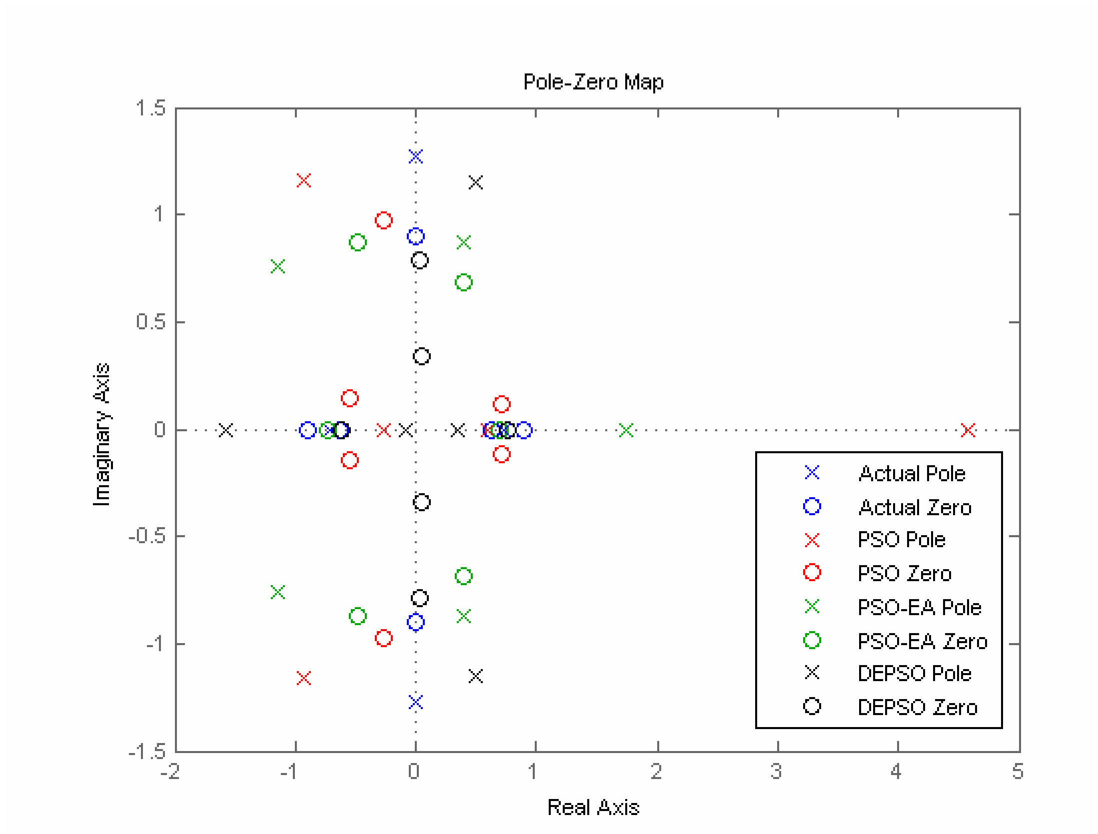

Figure 7.14. Pole zero plot for the full order model of Case IV

Cases V and VI are both second order IIR systems and are only studied in their full order. Due to the fewer number of coefficients in the transfer function, search space is limited and hence each of the algorithms finds almost the same set of solution. Hence, it is difficult to see any significant improvement among the three algorithms in the figures. However, PSO-EA seems to have taken lead over the first few iterations of the run. The result for Cases V is shown in Fig. 7.15.

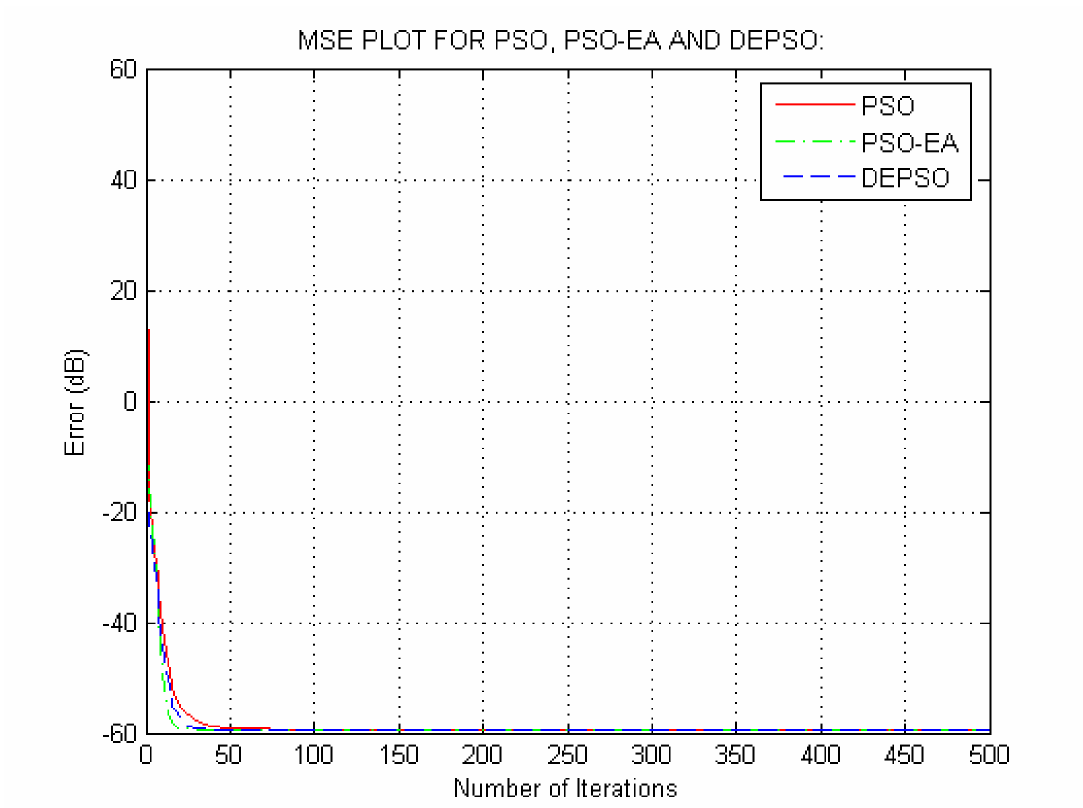

Figure 7.15. Error graph for Case V

The pole-zero plot of the coefficients obtained by PSO, PSO-EA and DEPSO for the full order model of Case V is shown in Fig. 7.16.

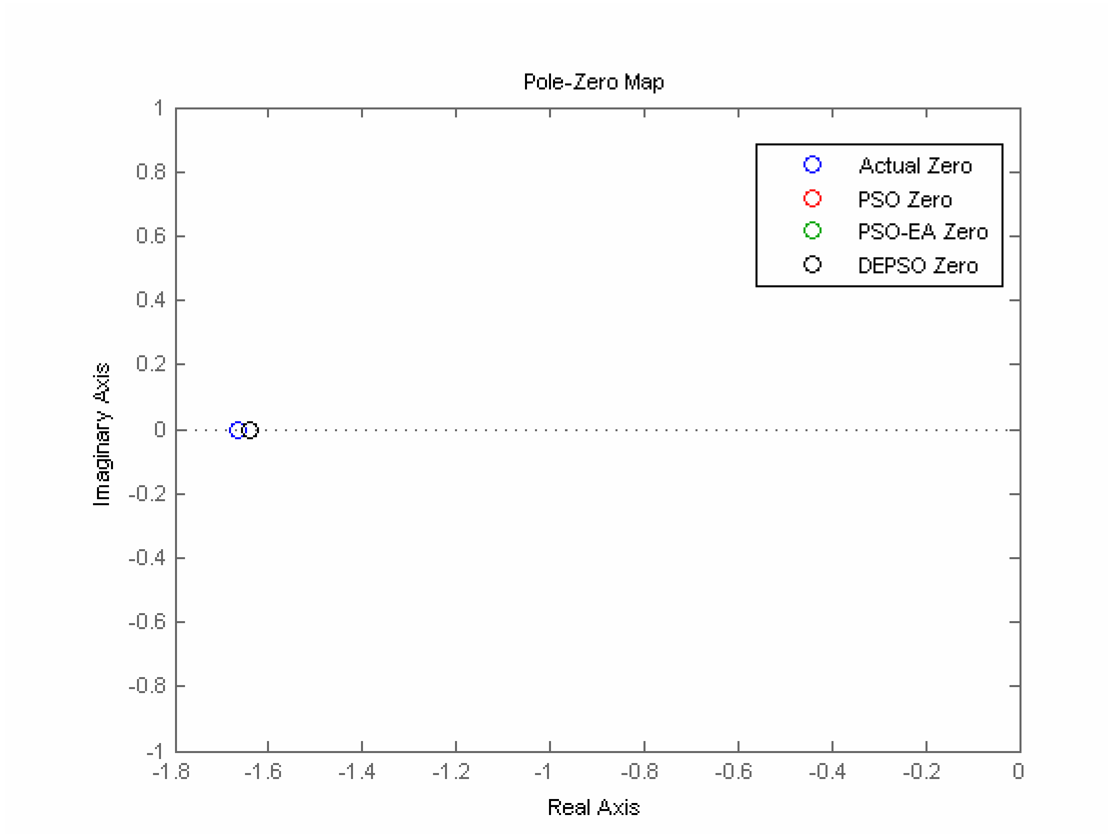

Figure 7.16. Pole zero plot for the full order model of Case V

The error graph for the results obtained in Case VI is shown in Fig. 7.17.

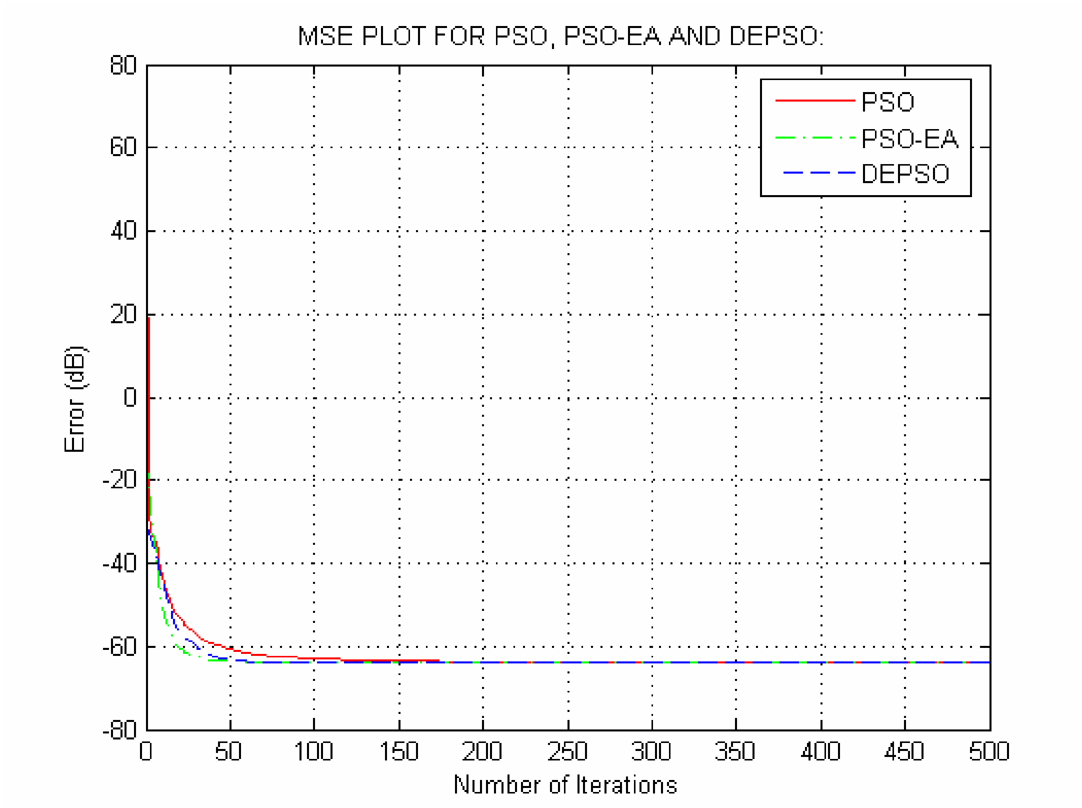

Figure 7.17. Error graph for Case VI

The pole-zero plot of the coefficients obtained by PSO, PSO-EA and DEPSO for the full order model of Case VI is shown in Fig. 7.18.

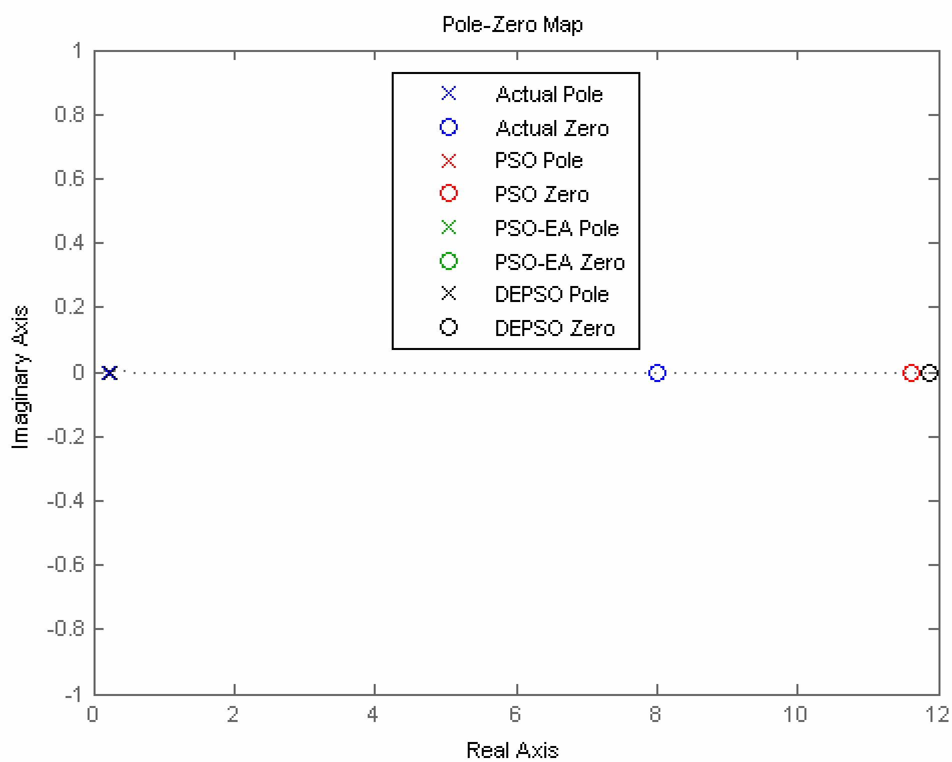

Figure 7.18. Pole zero plot for the full order model of Case VI

The data obtained from the studies are tabulated below. Table 7.6 shows the data obtained for the full order model of all six cases for the first scenario. The table presents the average, minimum and standard deviation of the mean squared error for the three algorithms in each case, along with time taken by each. These results show that DEPSO was the most consistent and also converged to better fitness in most of the cases where as PSO-EA performed the best in terms of time. Table 7.7 shows the similar data for the reduced order model of the four cases. The coefficients obtained by each algorithm along with the actual parameters in full and reduced order model are presented in tables 7.8 and 7.9 below.

Table 7.6. Data for the Full Order Model

Case		MSE (dB)			Time(seconds)*	
		Min.	Avg.	Std.	Min.	Avg.
Case I	PSO	-62.564	-62.449	0.364	4.313	5.163
	PSO-EA	-62.564	-62.563	0.003	3.656	4.160
	DEPSO	-62.564	-62.564	5.024e-14	5.391	6.222
Case II	PSO	-61.854	-47.728	11.952	3.734	4.248
	PSO-EA	-63.732	-50.238	16.312	2.703	3.354
	DEPSO	-63.817	-63.815	0.002	4.141	5.446
Case III	PSO	-64.553	-59.322	5.401	2.094	2.558
	PSO-EA	-64.558	-60.806	5.107	2.515	2.941
	DEPSO	-64.559	-64.559	1.318e-4	3.375	3.869
Case IV	PSO	-58.323	-51.134	4.389	3.406	3.694
	PSO-EA	-62.512	-50.689	7.757	3.062	3.306
	DEPSO	-62.641	-57.855	2.923	4.234	4.506
Case V	PSO	-63.699	-63.699	1.621e-11	3.125	3.834
	PSO-EA	-63.699	-63.699	1.463e-14	3.172	3.518
	DEPSO	-63.699	-63.699	7.177e-15	4.625	5.405

Table 7.6. (cont.) Data for the Full Order Model

Case VI	PSO	-62.753	-62.752	0.001	3.641	4.098
	PSO-EA	-62.753	-62.753	1.510e-9	3.391	3.984
	DEPSO	-62.753	-62.753	5.994e-14	2.609	6.643

*Performed on the same computer for 500 iterations.

Table 7.7. Data for the Reduced Order Model

Case		MSE (dB)			Time (seconds)*	
		Min.	Avg.	Std.	Min.	Avg.
Case I	PSO	-44.8085	-44.8085	7.177e-15	4.468	5.242
	PSO-EA	-44.8085	-44.8085	7.177 e-15	3.782	4.322
	DEPSO	-44.8085	-44.8085	7.177 e-15	5.015	6.496
Case II	PSO	-46.393	-40.235	4.553	3.594	4.116
	PSO-EA	-46.433	-45.412	1.875	2.875	3.652
	DEPSO	-46.440	-46.440	3.343e-11	4.156	5.896
Case III	PSO	-45.367	-41.697	2.782	3.156	3.737
	PSO-EA	-45.391	-42.666	2.881	1.781	3.177
	DEPSO	-45.392	-45.392	8.099e-6	2.468	4.997
Case IV	PSO	-57.445	-49.754	3.545	3.375	4.755
	PSO-EA	-60.427	-48.755	6.181	3.203	3.971
	DEPSO	-60.415	-56.530	2.221	5.328	6.333

*Performed on the same computer for 500 iterations.

Table 7.8. Coefficients for Full Order Model

Actual Parameters	Achieved Parameters		
	PSO	PSO-EA	DEPSO
	Case I		
0.3	0.2892	0.2803	0.2803
-0.4	-0.4108	-0.4173	-0.4173
1.25	1.2537	1.2536	1.2536
-0.25	-0.2469	-0.2586	-0.2586
	Case II		
0.6	0.6666	0.6573	0.6001
-0.25	-0.1012	-0.1980	-0.2653
0.2	0.2853	0.1936	0.1849
-0.2	-0.2005	-0.2184	-0.2184
-0.4	-0.4186	-0.4205	-0.4072
0.5	0.4642	0.4625	0.4976
	Case III		
-0.04	0.0250	0.3075	0.0009
-0.2775	-0.1730	0.0651	-0.1809
0.2101	0.2469	0.4515	0.2643
-0.14	-0.1415	0.0375	-0.1355
1	0.9706	0.9518	0.9731
-0.9	-0.8370	-0.5618	-0.8576
0.81	0.8373	0.8331	0.8508
-0.729	-0.7040	-0.3833	-0.7071

Table 7.8. (cont.) Coefficients for Full Order Model

	Case IV		
0	0.0977	0.1878	-0.3127
-0.77	-0.2965	-0.0489	-0.1042
0	-0.5443	-0.3007	0.0495
-0.8498	-0.7497	-0.2044	-0.7689
0	0.4205	0.1387	0.1998
0.6486	0.1619	-0.5669	0.0238
1	1.0059	0.9828	0.9665
0	0.1977	0.2045	-0.3036
-0.4	0.1024	0.3571	0.3012
0	-0.6116	-0.2301	-0.0919
-0.65	-0.5135	0.1924	-0.2569
0	0.2111	0.1158	0.0202
0.26	0.1789	-0.3122	-0.0332
	Case V		
-1.2	-1.2058	-1.2058	-1.2058
0.6	0.6037	0.6037	0.6037
1	0.9903	0.9903	0.9903
	Case VI		
-1.1314	-1.1721	-1.1731	-1.1731
0.25	0.2905	0.2913	0.2913
0.05	0.0335	0.0327	0.0327
-0.4	-0.3893	-0.3880	-0.3880

Table 7.9. Coefficients for Reduced Order Model

Actual Parameters	Achieved Parameters		
	PSO	PSO-EA	DEPSO
	Case I		
a_1	0.769	0.769	0.769
b_0	1.097	1.097	1.097
	Case II		
a_1	0.674	1.172	1.189
a_2	-0.370	0.148	0.166
b_0	0.006	-0.238	-0.191
b_1	-0.536	-0.627	-0.569
	Case III		
a_1	0.559	0.042	1.017
a_2	-0.603	-1.043	0.558
a_3	-0.318	-0.313	0.344
b_0	0.996	0.975	1.038
b_1	-0.475	-1.039	0.217
b_2	-0.184	-0.068	0.858

Table 7.9. (cont.) Coefficients for Reduced Order Model

	Case IV		
a_1	-0.254	0.470	-0.024
a_2	-0.436	-0.204	0.124
a_3	0.268	-0.107	-0.136
a_4	-0.293	-0.627	-0.776
a_5	-0.187	-0.335	-0.043
b_0	0.956	0.941	0.987
b_1	-0.234	0.413	0.031
b_2	-0.046	0.117	0.499
b_3	0.212	0.081	-0.150
b_4	-0.0274	-0.142	-0.283
b_5	-0.1944	-0.108	-0.117

For the second scenario, PSO-QI is the undoubted winner. The results are shown in the following figures. Figs. 7.19 and 7.20 show the results of the full order and reduced order model in Case I. The figures have been magnified in scale to compare the fitness of the algorithms.

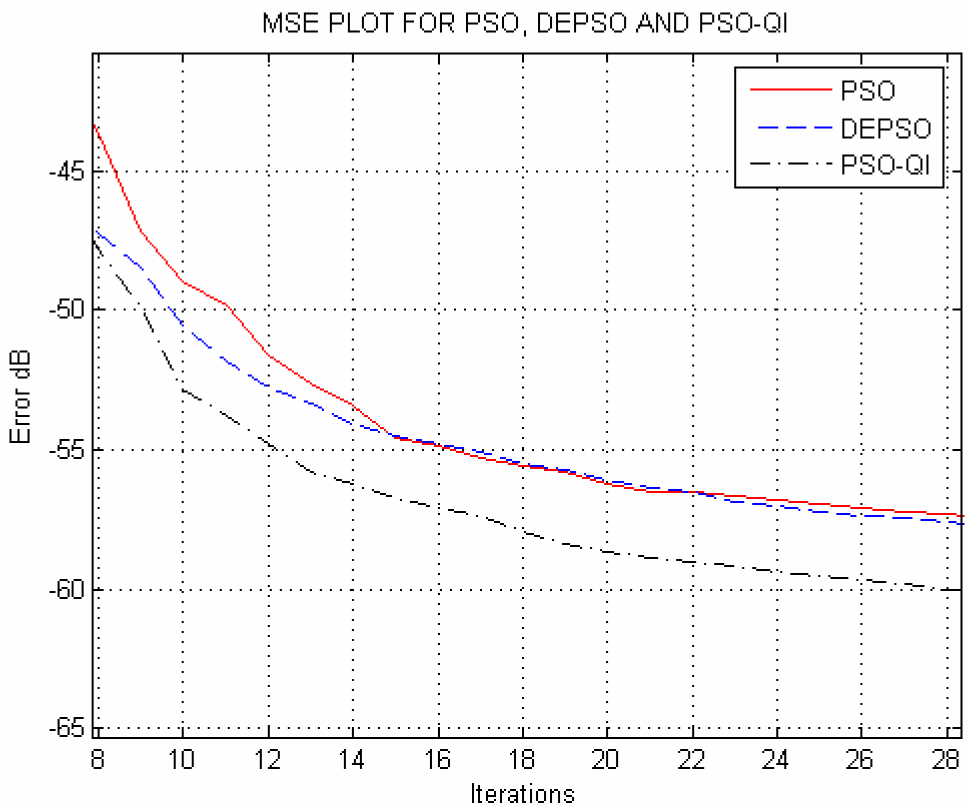

Figure 7.19. Error graph for full order model of Case I.

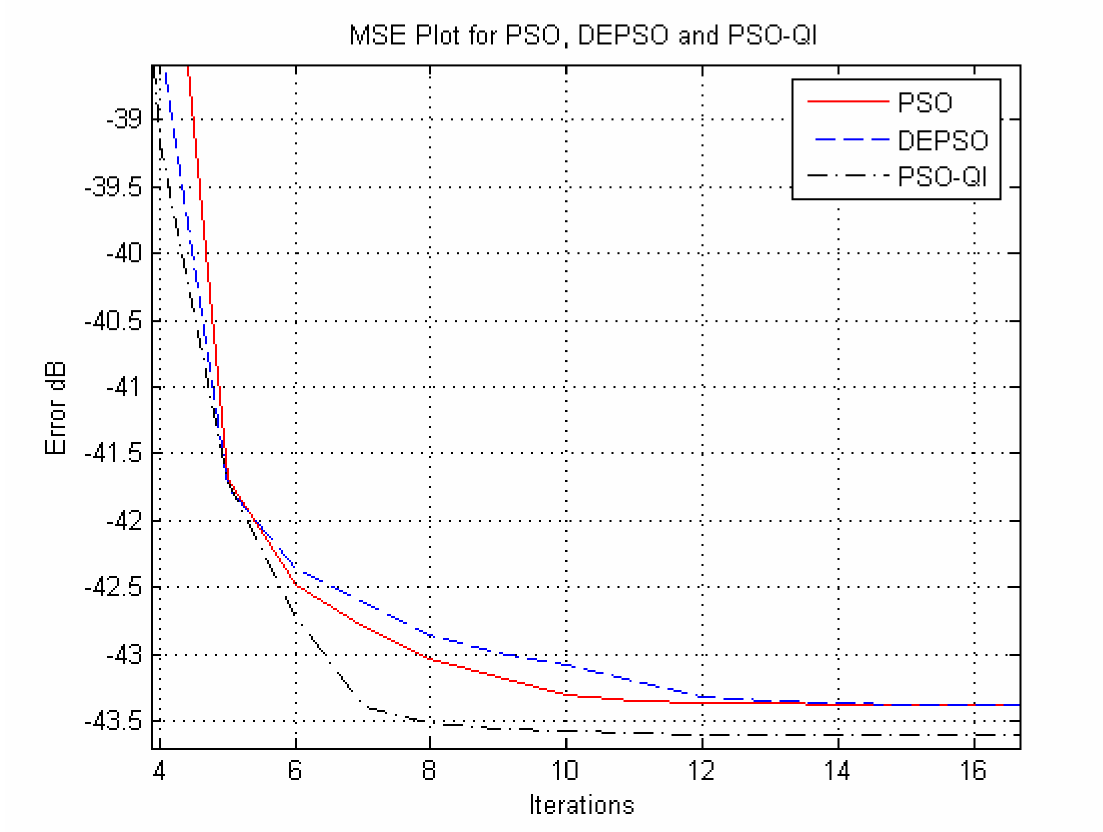

Figure 7.20. Error graph for reduced order model of Case I.

The full order model and the reduced order model implementation results for Case II are also shown in Figs. 7.21 and 7.22.

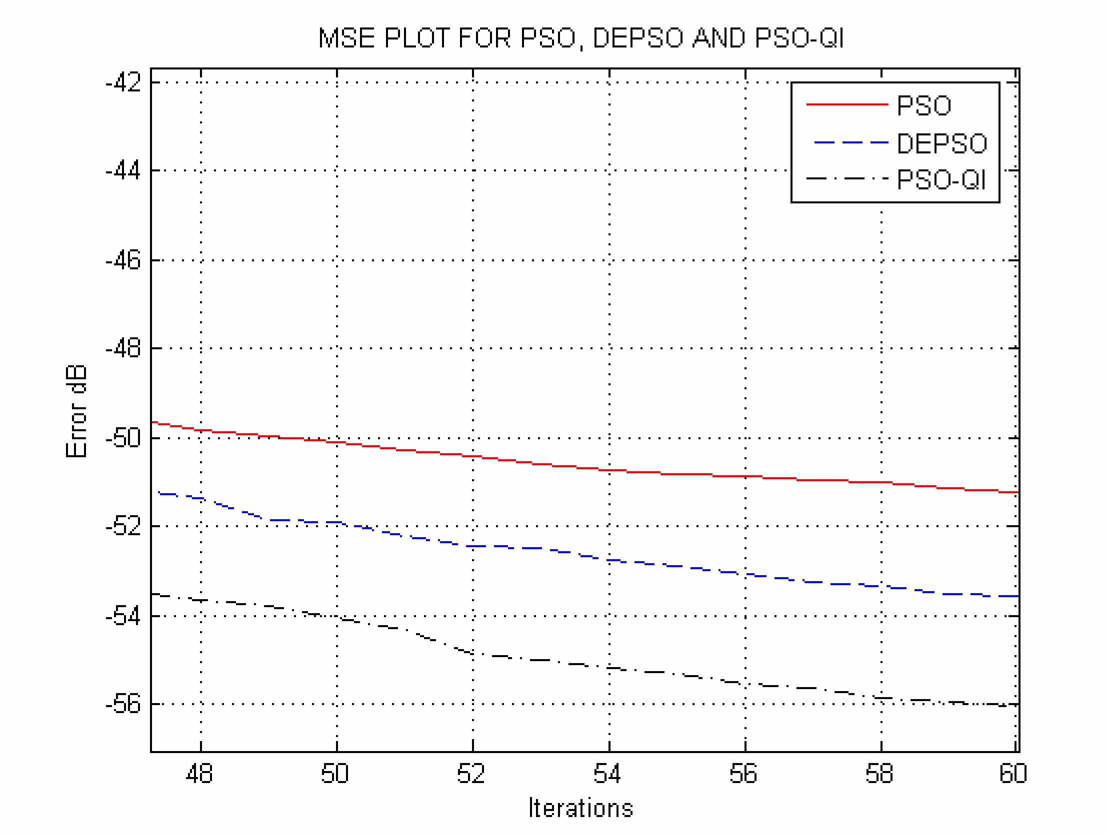

Figure 7.21. Error graph for full order model of Case II.

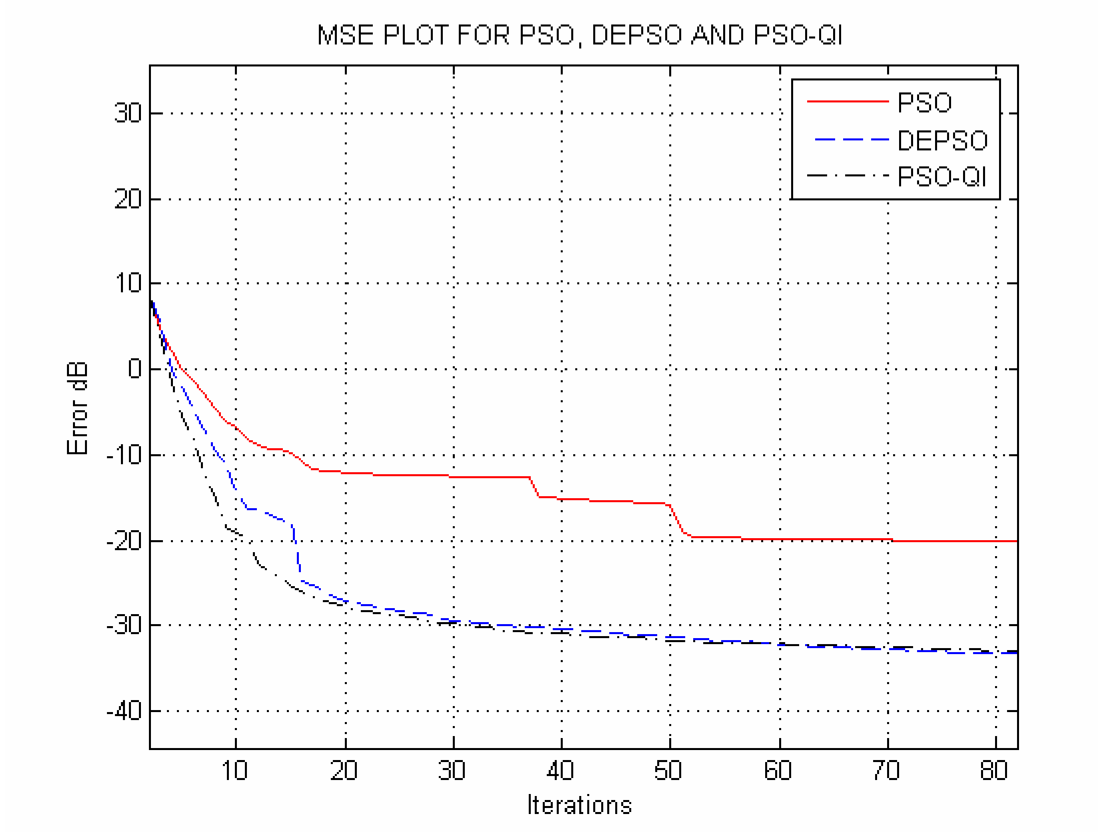

Figure 7.22. Error graph for the reduced order model of Case II.

The full order and reduced order model of Case III are shown in Figs. 7.23 and 7.24. These figures also show the better performance of PSO-QI over PSO and DEPSO.

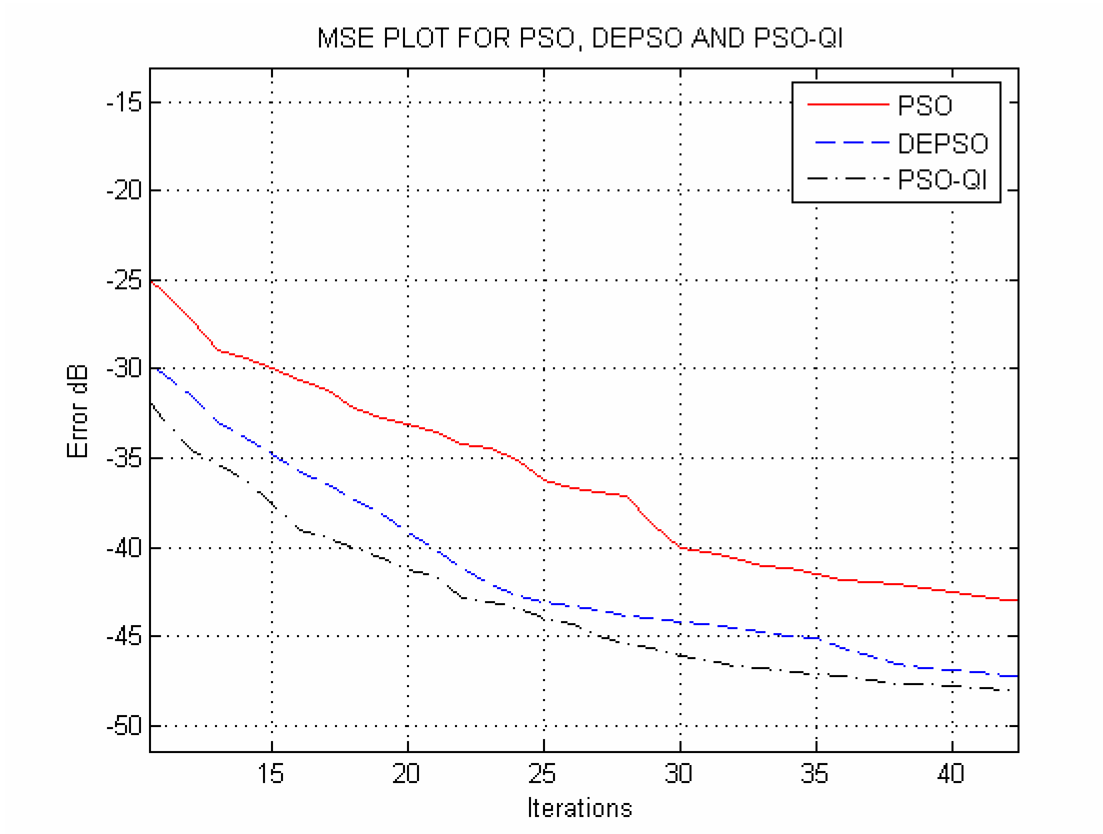

Figure 7.23. Error graph for the full order model of Case III.

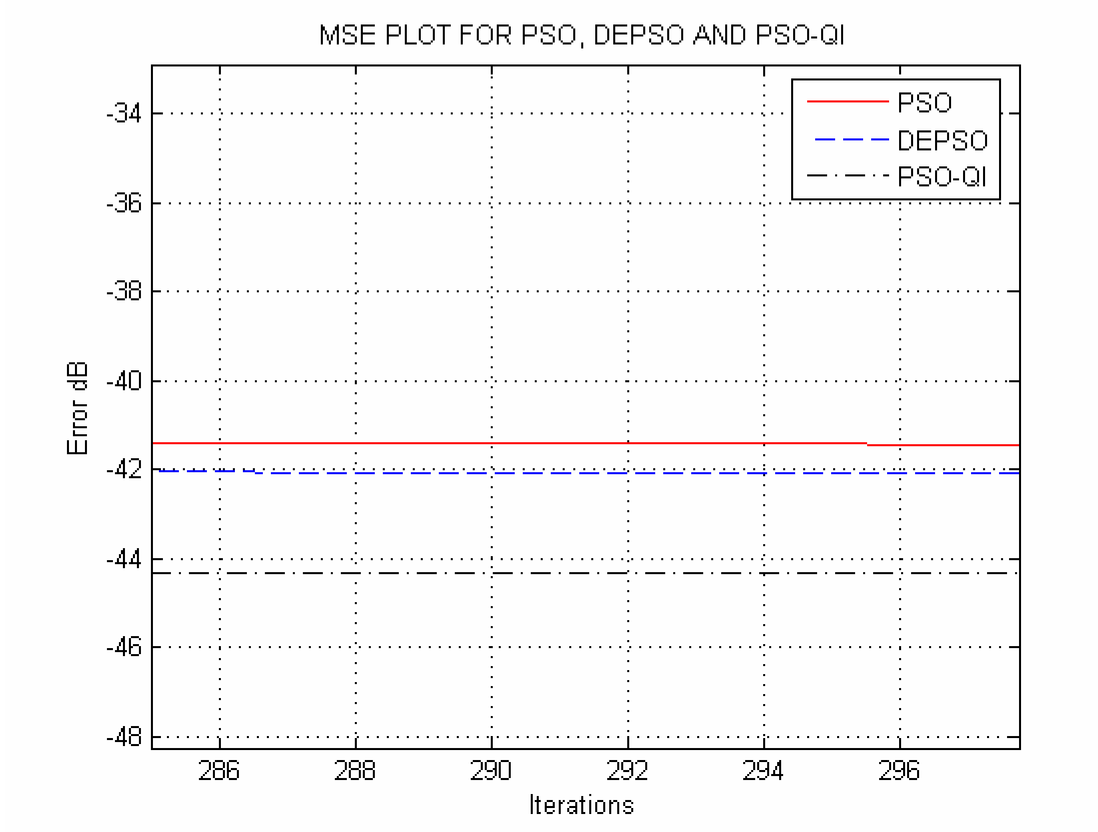

Figure 7.24. Error graph for the reduced order model of Case III.

Figs. 7.25 and 7.26 show the full order and reduced order model results for Case IV.

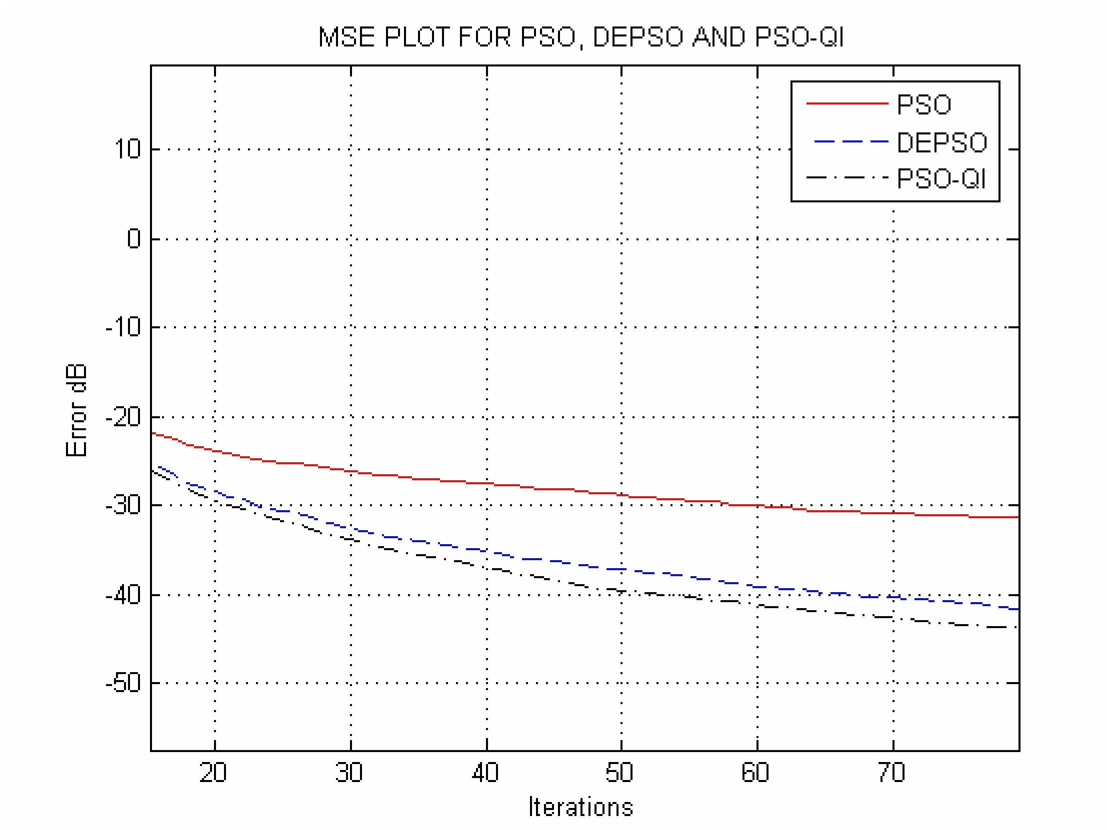

Figure 7.25. Error graph for the full order model of Case IV.

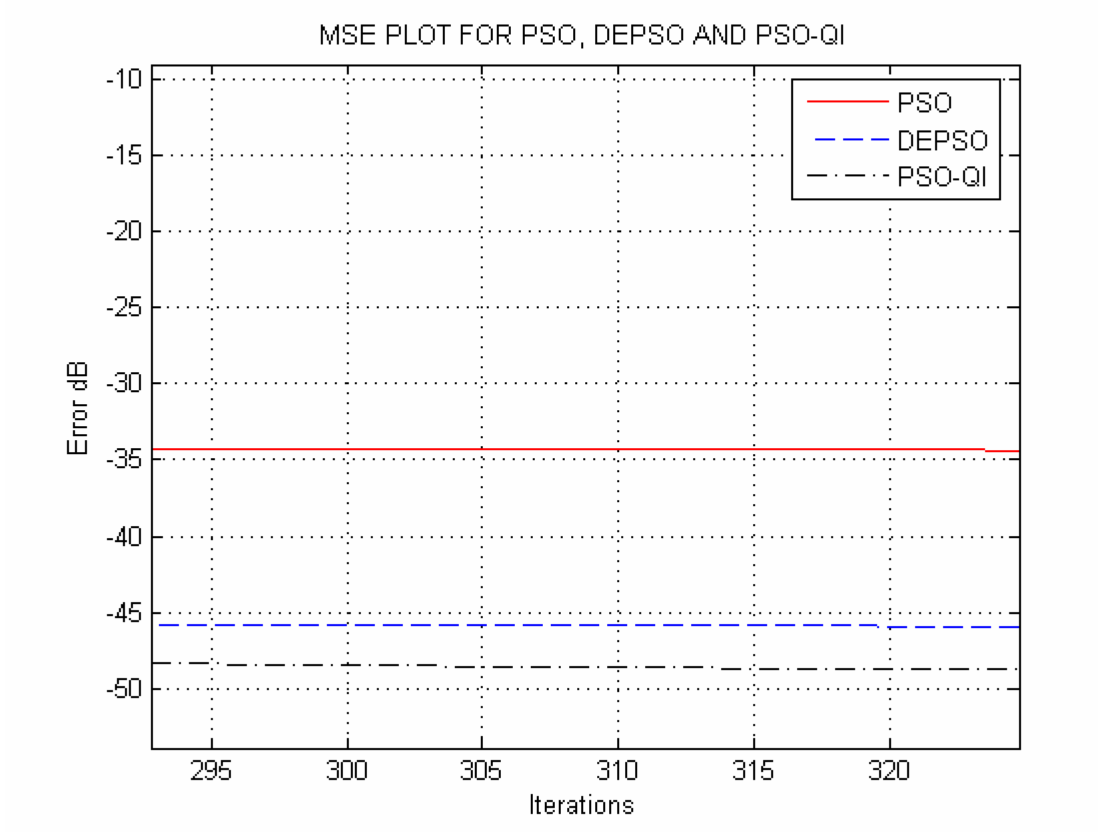

Figure 7.26. Error graph for the reduced order model of Case IV.

Cases V and VI are implemented only for full order and the error curves obtained for these cases are shown in Figs. 7.27 and 7.28, respectively.

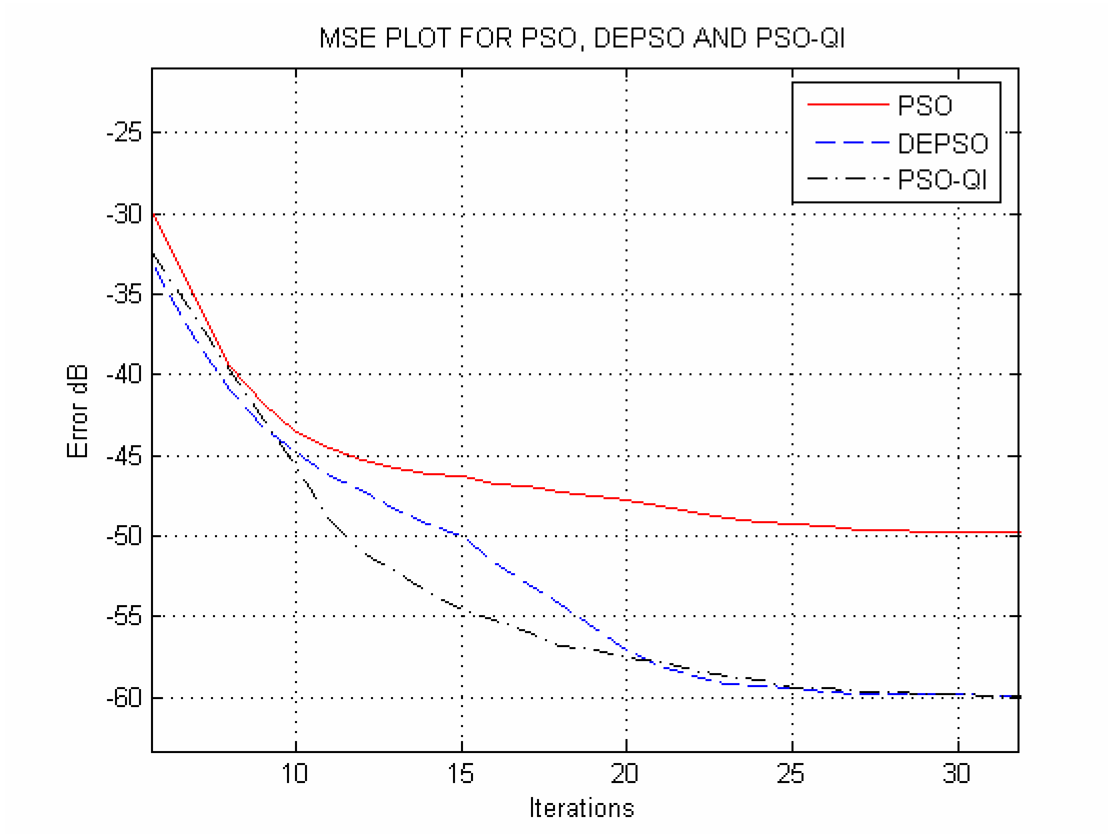

Figure 7.27. Error graph for the full order model of Case V.

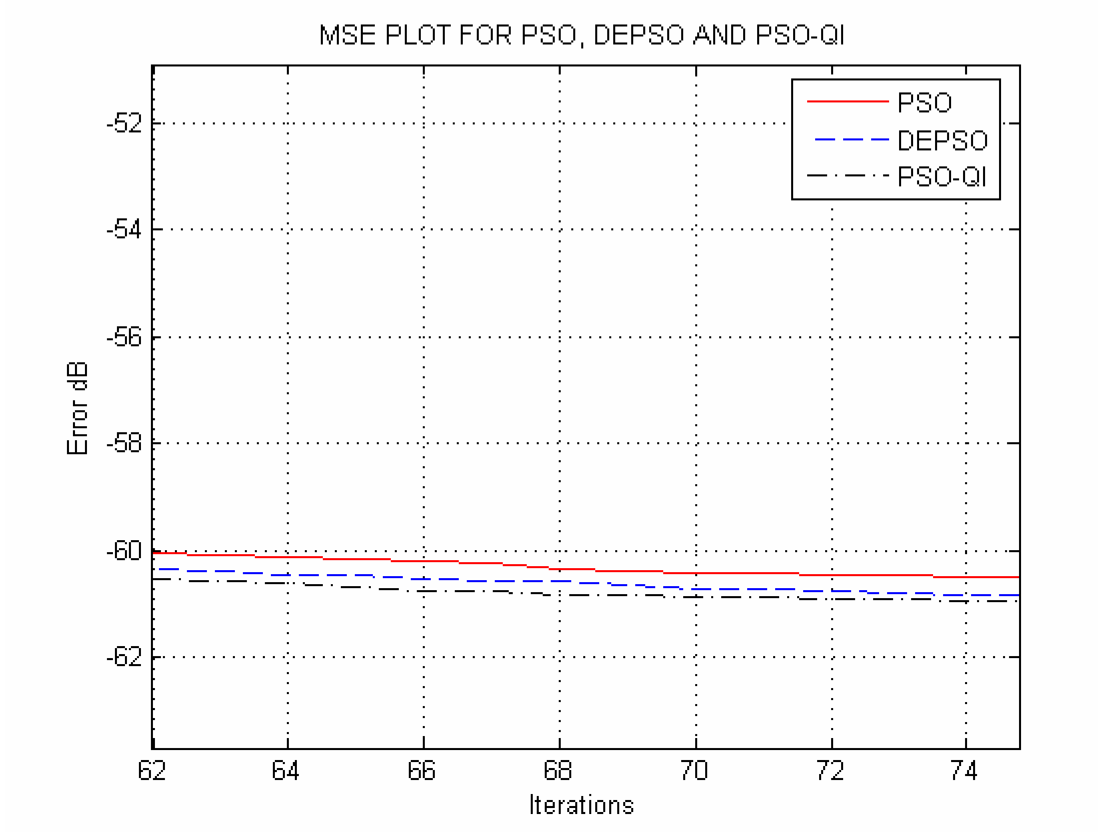

Figure 7.28. Error graph for the full order model of Case VI.

The study was also carried out to show that PSO-QI starts to converge in lesser number of iterations than the other algorithms and that this could be useful in online adaptation. This is shown by implementing the system identification for 50 iterations. It is seen that PSO-QI converges to a reasonable error level in lesser number of iterations. Figs. 7.29 and 7.30 show these results for full order model of Case III and IV implemented in 50 iterations.

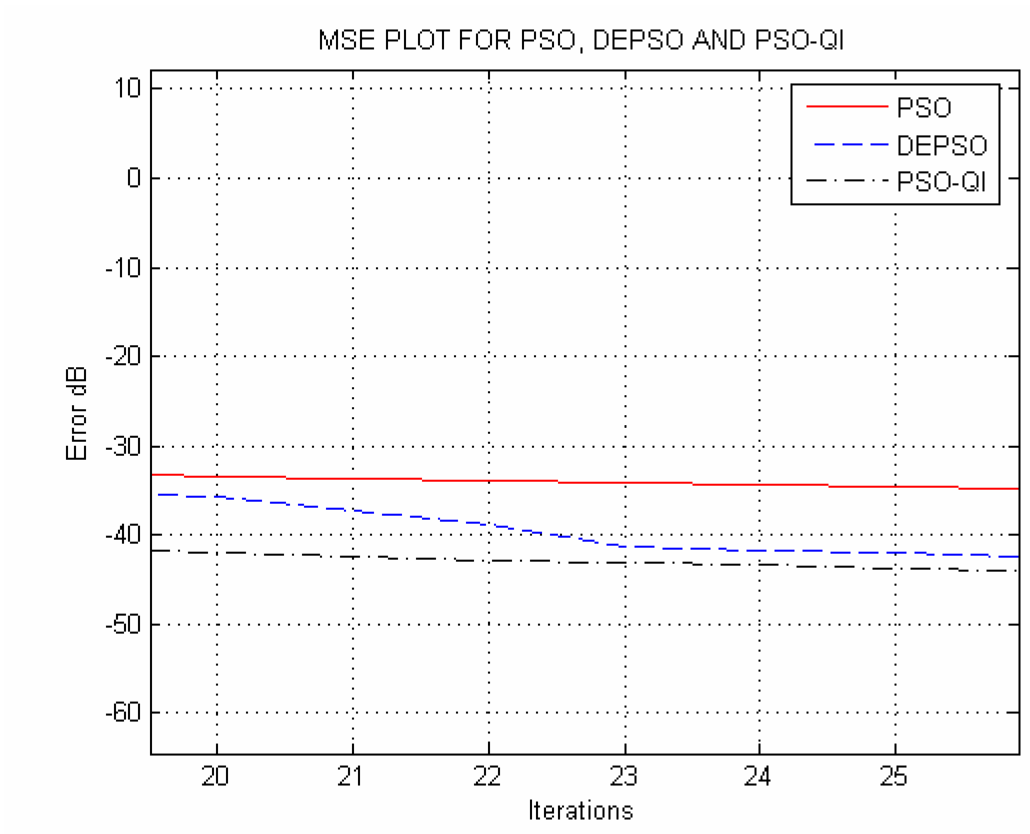

Figure 7.29. Error graph for full order model of Case III in 50 iterations.

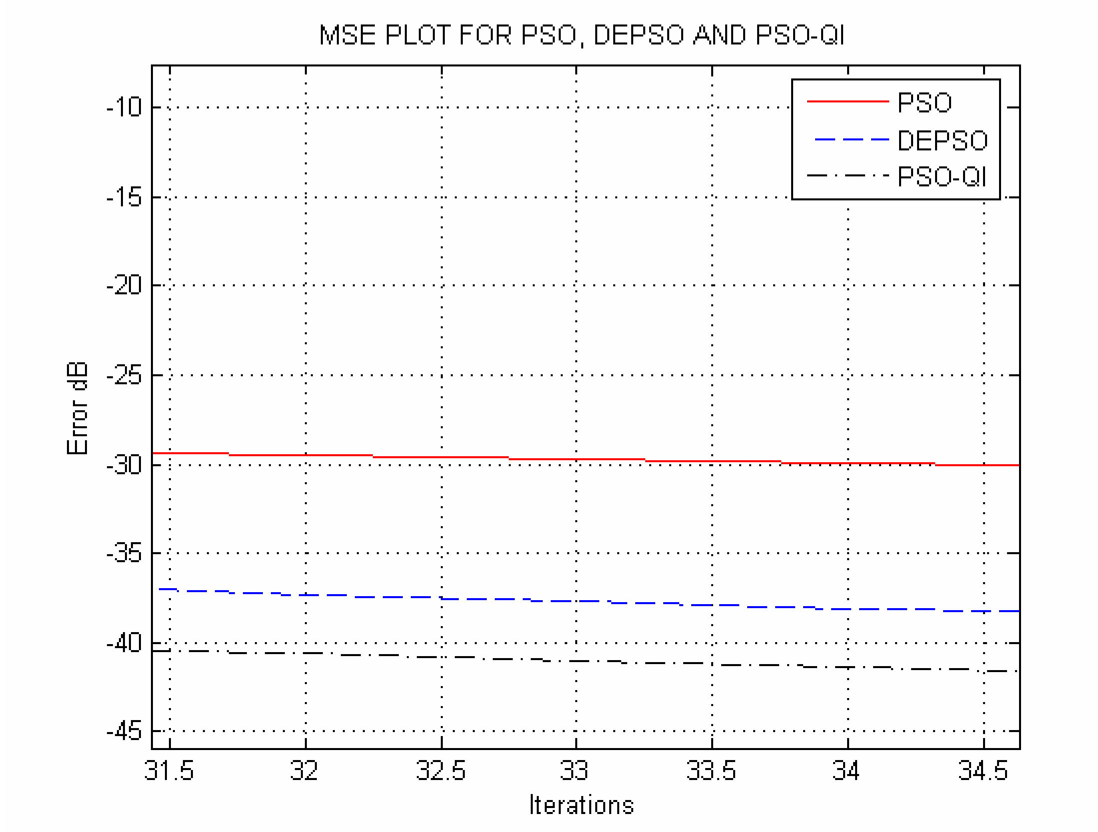

Figure 7.30. Error graph for the full order model of Case IV in 50 iterations.

The results of system identification carried out in the second scenario are tabulated in the following tables. In Table 7.10, the results obtained for full order model in 500 iterations are presented. Similar results for reduced order model are presented in Table 7.11. The results obtained for full order model using 50 iterations are also presented in Table 7.12.

Table 7.10. Results of full order model for 500 iterations.

Case		MSE (dB)			Time(seconds)*	
		Min.	Avg.	Std.	Min.	Avg.
Case I	PSO	7.102e-4	8.612e-4	5.074e-4	3.422	3.769
	DEPSO	7.102e-4	7.278e-4	4.391e-5	2.547	3.166
	PSO-QI	7.102e-4	7.102e-4	1.148e-7	2.984	3.227
Case II	PSO	7.791e-4	0.001	5.222e-4	3.563	3.778
	DEPSO	7.791e-4	9.480e-4	4.011e-4	2.703	2.826
	PSO-QI	7.791e-4	9.215e-4	3.627e-4	3.281	3.432
Case III	PSO	7.245e-4	0.003	0.003	2.609	3.404
	DEPSO	7.245e-4	0.001	0.001	2.672	3.056
	PSO-QI	7.245e-4	0.001	0.001	3.421	3.734
Case IV	PSO	7.821e-4	0.011	0.014	0.938	2.240
	DEPSO	7.623e-4	0.002	0.003	1.046	2.329
	PSO-QI	7.984e-4	0.002	0.004	3.063	4.008
Case V	PSO	7.542e-4	0.002	0.004	3.468	3.791
	DEPSO	7.542e-4	0.001	0.001	2.594	2.888
	PSO-QI	7.542e-4	7.542e-4	8.241e-19	2.953	3.237
Case VI	PSO	8.567e-4	9.138e-4	1.729e-4	2.250	2.286
	DEPSO	8.567e-4	8.681e-4	8.071e-5	2.421	2.455
	PSO-QI	8.567e-4	8.567e-4	2.750e-11	2.703	2.733

Table 7.11. Results of reduced order model for 500 iterations.

Case		MSE (dB)			Time (seconds)*	
		Min.	Avg.	Std.	Min.	Avg.
Case I	PSO	0.006	0.006	7.149e-4	2.234	2.356
	DEPSO	0.006	0.006	4.214e-4	2.125	2.326
	PSO-QI	0.006	0.006	4.085e-18	2.500	2.601
Case II	PSO	0.004	0.089	0.443	3.625	3.799
	DEPSO	0.004	0.010	0.005	3.609	3.700
	PSO-QI	0.004	0.011	0.006	3.079	3.130
Case III	PSO	0.005	0.008	0.003	0.766	1.269
	DEPSO	0.005	0.007	0.003	0.859	1.392
	PSO-QI	0.005	0.005	0.001	2.312	2.700
Case IV	PSO	0.001	0.018	0.042	2.281	2.766
	DEPSO	0.001	0.004	0.004	2.515	2.678
	PSO-QI	0.001	0.003	0.001	3.375	3.627

Table 7.12. Results of full order model for 50 iterations.

Case		MSE (dB)			Time(seconds)*	
		Min.	Avg.	Std.	Min.	Avg.
Case I	PSO	9.448e-4	0.001	5.011e-4	0.218	0.261
	DEPSO	9.448e-4	0.001	5.806e-4	0.234	0.302
	PSO-QI	9.447e-4	9.988e-4	1.222e-4	0.265	0.275
Case II	PSO	0.001	0.002	0.003	0.234	0.264
	DEPSO	0.001	0.002	0.003	0.235	0.263
	PSO-QI	0.001	0.001	5.674e-4	0.297	0.343
Case III	PSO	0.001	0.013	0.045	0.233	0.269
	DEPSO	0.001	0.004	0.002	0.250	0.275
	PSO-QI	8.353e-4	0.003	0.002	0.343	0.371
Case IV	PSO	0.001	0.024	0.032	0.234	0.257
	DEPSO	8.688e-4	0.007	0.010	0.250	0.267
	PSO-QI	9.994e-4	0.004	0.006	0.375	0.399
Case V	PSO	8.675e-4	0.003	0.005	0.203	0.240
	DEPSO	8.675e-4	8.677e-4	4.662-7	0.233	0.321
	PSO-QI	8.675e-4	8.677e-4	4.060-7	0.250	0.277
Case VI	PSO	9.562e-4	0.001	2.669e-4	0.203	0.238
	DEPSO	9.561e-4	0.001	9.524e-5	0.234	0.259
	PSO-QI	9.560e-4	9.858e-4	3.322e-5	0.265	0.287

7.7. DISCUSSION

From these studies it is observed that PSO, PSO-EA, DEPSO and PSO-QI all could be used in the identification of complex systems. However, hybrid algorithms have an edge over the classical PSO due to the fact that they combine the best features of both the algorithms. From the figures and results, it is observed that PSO-EA could reach a lower value of error in a short period of time because of the use of elites in its search process. However, this caused it to lose diversity and hence it could not explore the search space. This is where DEPSO overcame this limitation and continued to explore the search space, thus settling to a much lower value of error. Moreover, the benefit of using DEPSO is in the fact that it converges to a lower value of error every time, while the other two deviate from the final error value over a number of runs, as is evident from the standard deviation values. In the second scenario when PSO-QI is introduced, PSO-QI outperformed the convergence capacity of DEPSO and also in its performance. Although time taken by PSO-QI is higher than the other two, it is a trade-off against its performance. From the results, it is also observed that coefficients approximated by DEPSO are more close to the coefficients of the actual system. The lower average error and lower values of standard deviation even in the reduced order case prove the ability of DEPSO and PSO-QI to find better results more consistently even in multimodal error surfaces.

7.8. SUMMARY

In this chapter, result of application of PSO, PSO-EA and DEPSO in system identification was presented. The results showed DEPSO to be the best of the three algorithms in terms of its performance and consistency. PSO-EA outperformed the others in terms of execution time. The results of successful implementation of six benchmark IIR systems proved the abilities of swarm and evolutionary algorithms, and opened ground for research into more novel algorithms that are more efficient in terms of both time and performance.

8. DIGITAL FITLER DESIGN USING PSO, DEPSO AND PSO-QI

8.1. INTRODUCTION

PSO, DEPSO and PSO-QI are used for the design of Lowpass (LP), Higpass (HP), Bandpass (BP) and Bandstop (BS) FIR and IIR digital filters. These algorithms have been applied in different case studies where linear phase FIR, non-linear phase FIR and IIR filters have been designed. Two different kinds of fitness function have been used in one of the cases. In all of the cases, the goal is to effectively approximate the filter coefficients using the algorithms, so that the magnitude response of the filter is as close as possible to that of ideal filter. One of the fitness functions considers the mean squared error between the magnitude of the ideal and the designed filter where as the other fitness function considers the ripples on the passband and stopband. In the former case, the fitness function tries to match the magnitudes of the designed and ideal filter, where as in the later case, the fitness function tries to keep the passband and stopband ripples of the designed filter within a given range.

In Case I, linear phase FIR filter has been designed using PSO and DEPSO using the first fitness function given by (40). In Case II, the same case is repeated using the second fitness function given by (41). In order to show the effectiveness of DEPSO in time critical design environment where fast convergence is required, the study is also carried out for a less number of iterations. This study has shown DEPSO to be able to design the filter in a very short period of time, making it suitable for use in online environment for adaptive systems. The parameters for the filter and the parameters used by the algorithms in Cases I and II are summarized in Table 8.1.

Table 8.1. Parameters used in the study of Cases I and II

Symbol	Parameter Description	Value
P	Population Size	25
c_1	Cognitive constant	2
c_2	Social constant	2
w	Inertia weight	Linearly decreasing from 0.95 to 0.4
V_{max}	Maximum Velocity	1
X_{max}	Maximum Position	1
CR	crossover rate	0.5
N	Number of Inputs	256
D	Dimension	10
δ_p	Passband ripple	0.1
δ_s	Stopband ripple	0.01
ω_p	Passband cutoff frequency	0.25
ω_s	Stopband cutoff frequency	0.3
	Number of Coefficients	20
	Number of Iterations	40, 200
	Number of Trials	10

In Case III, LP, HP, BP and BS FIR filters are designed using PSO, DEPSO and PSO-QI. The same algorithms are also used in designing a LP IIR filter. Both of these cases use the fitness function given by (40). The parameters used in the design are summarized in Table 8.2. In Case IV, these algorithms have been used in the design of LP, HP, BP and BS FIR filters using the fitness function given by (41). The specifications of the filter and number of coefficients are taken from (Ababneh and Bataineh, 2005; Fang et al., 2006). The algorithm's parameters used in the study are obtained from the best parameters as reported in (Eberhart and Shi, 2000).

Table 8.2. Parameters used in the study of Cases III and IV

Symbol	Parameter Description	Value
P	Population Size	25
c_1	Cognitive constant	2
c_2	Social constant	2
w	Inertia weight	Linearly decreasing from 0.9 to 0.4
β	Creativity coefficient	Linearly increasing from 0.5 to 1
V_{max}	Maximum Velocity	1
X_{max}	Maximum Position	1
CR	crossover rate	0.5
N	Number of Inputs	256
D	Dimension	20
δ_p	Passband ripple	0.1
δ_s	Stopband ripple	0.01
ω_p	Passband cutoff frequency	0.45 (LP, HP) , 0.3 (BP, BS)
ω_s	Stopband cutoff frequency	0.55 (LP, HP) , 0.7 (BP, BS)
	Number of Coefficients	20
	Number of Iterations	500,1500
	Number of Trials	50

8.2. APPLICATION IN DIGITAL FILTER DESIGN

Transfer function of the FIR filter can also be represented as:

$$\frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_n z^{-n} \tag{37}$$

Similarly, an IIR filter can have the following transfer function:

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_n z^{-n}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_m z^{-m}} \tag{38}$$

Now for (37), the numerator coefficient vector $\{ b_0, b_1, b_2, \ldots\ldots\ldots\ldots\ldots, b_N\}$ is represented in N dimensions where as for (38), the numerator as well as denominator coefficient vector is $\{ b_0, b_1, b_2, \ldots\ldots\ldots\ldots, b_M, a_0, a_1, a_2, \ldots\ldots\ldots\ldots\ldots, a_N \}$ which is represented in (N+M) dimensions. The particles are distributed in a D dimensional search space, where $D = N$ for FIR and $D = (N+M)$ for IIR filter. The position of the particles in this D dimensional search space represents the coefficients of the transfer function. In each iteration, these particles find a new position, which is the new set of coefficients. Fitness of particles is calculated using the new coefficients. This fitness is used to improve the search in each iteration, and result obtained after a certain number of iterations or after the error is below a certain limit is considered to be the final result. Different kinds of fitness functions have been used in different literature. An error function given by (39) is the approximate error used in Parks-McClellan algorithm for filter design.

$$E(\omega) = G(\omega)[H_d(e^{j\omega}) - H(e^{j\omega})] \tag{39}$$

where $G(\omega)$ is the weighting function used to provide different weights for the approximate errors in different frequency bands, $H_d(e^{j\omega})$ is the frequency response of the desired filter and $H(e^{jw})$ is the frequency response of the approximate filter (Ababneh and Bataineh, 2007).

Now the error to be minimized is defined as:

$$J_1 = \underset{w \le w_p}{Max}(|E(\omega)| - \delta_p) + \underset{w \ge w_s}{Max}(|E(\omega)| - \delta_s) \qquad (40)$$

where δ_p and δ_s are the ripples in the passband and stopband, and ω_p and ω_s are passband and stopband normalized cut off frequencies respectively. The algorithms try to minimize this error and thus increase the fitness.

The second fitness function takes the mean squared error between the frequency response of the ideal and the actual filter. An ideal filter has a magnitude of 1 on the passband and a magnitude of 0 on the stopband. So the error for this fitness function is the squared difference between the magnitudes of this filter and the filter designed using the evolutionary algorithms, summed over desired frequency range and divided by the total number of input samples for which the frequency response is evaluated. This is called the mean squared error and is given by (41)

$$J_2 = \frac{1}{N}\sum_{k=1}^{N}(Ideal(k) - Actual(k))^2 \qquad (41)$$

where *ideal(k)* and *actual(k)* are the magnitude response of the ideal and the actual filter, and *N* is the number of samples used to calculate the error.

In one of the works, a linear phase FIR filter is designed. Since the coefficients of the linear phase filter are matched, meaning the first and the last coefficients are the same; the dimension of the problem could be reduced by one-half. By only determining one half of the coefficients, the filter could be designed. This greatly reduced the computational complexity of the algorithms. Application of digital filter design using PSO-QI is shown in the flowchart in Fig. 8.1.

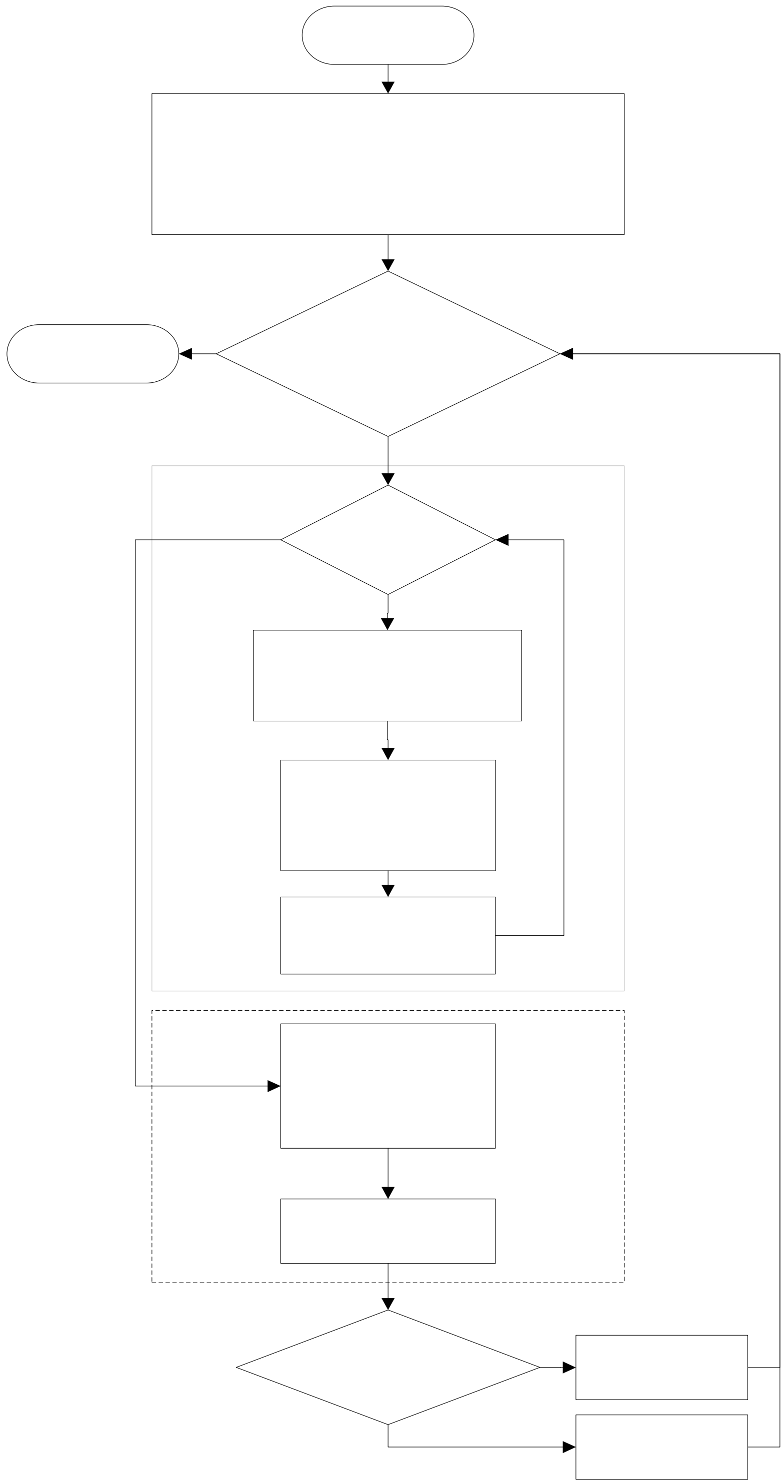

Figure 8.1. Flowchart for the design of digital filters using PSO-QI.

8.3. RESULTS WITH PSO AND DEPSO

Performance of PSO in digital filter design gives a basis of comparison for other algorithms. The results show that DEPSO performs better than PSO in Cases I and II. FIR and IIR filters are designed in Case III and FIR filters are designed in Case IV. In both of these cases performance of PSO is the baseline for the other algorithms' performance. In Case II run for 200 iterations, PSO has converged as well as DEPSO. For Case I, it is observed that PSO performed better on the stopband. Execution time for PSO is the least in all of the cases. Although PSO could sometimes converge to a much lower minimum value of error, its final value of convergence varied greatly over a number of iterations, thus making it highly inconsistent algorithm. DEPSO has performed better than PSO in terms of its consistency as well as ability to converge to a lower value of average error for both fitness functions. This is more convincing when it is run for only 40 iterations. DEPSO could converge to the same value of average error in less than 40 iterations as is done by PSO in 200 iterations, thus making it a better choice of algorithm for online adaptation. In Cases III and IV, DEPSO is better with respect to execution time for its performance. However its overall performance is midway between PSO and PSO-QI.

The error graph with PSO and DEPSO for Case I run for 200 iterations is shown in Fig. 8.2.

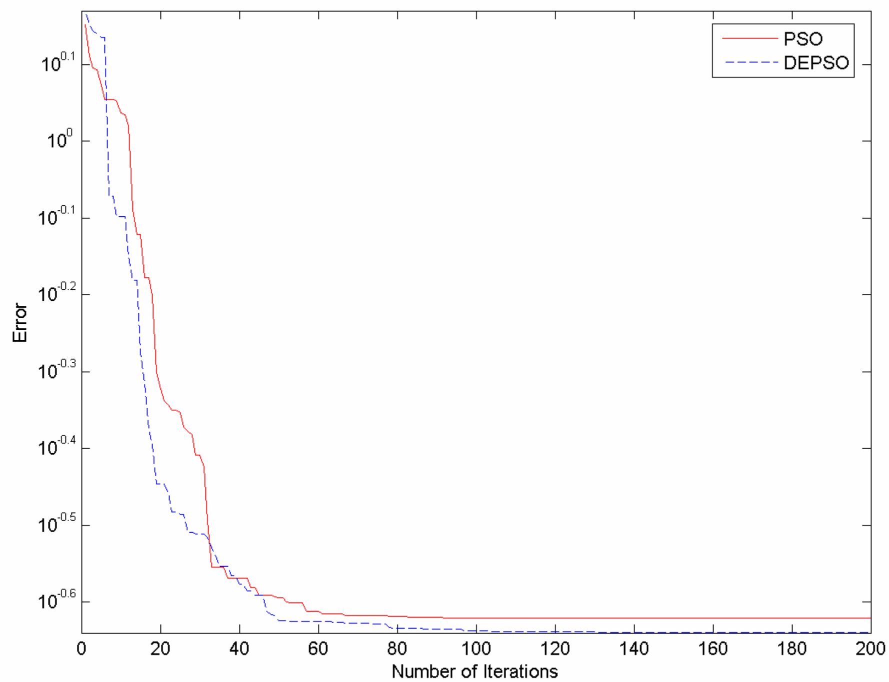

Figure 8.2. Error graph for Case I in 200 iterations.

The magnitude and gain plots for the filters designed in Case I are shown in Figs. 8.3 and 8.4 respectively.

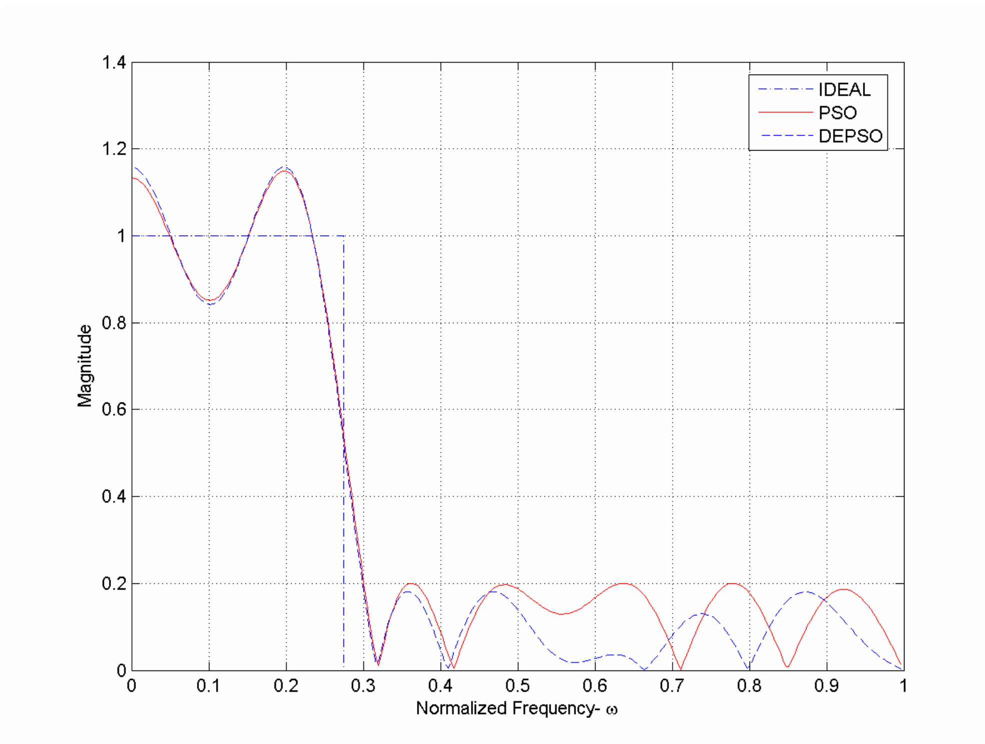

Figure 8.3. Magnitude response of filters for Case I in 200 iterations.

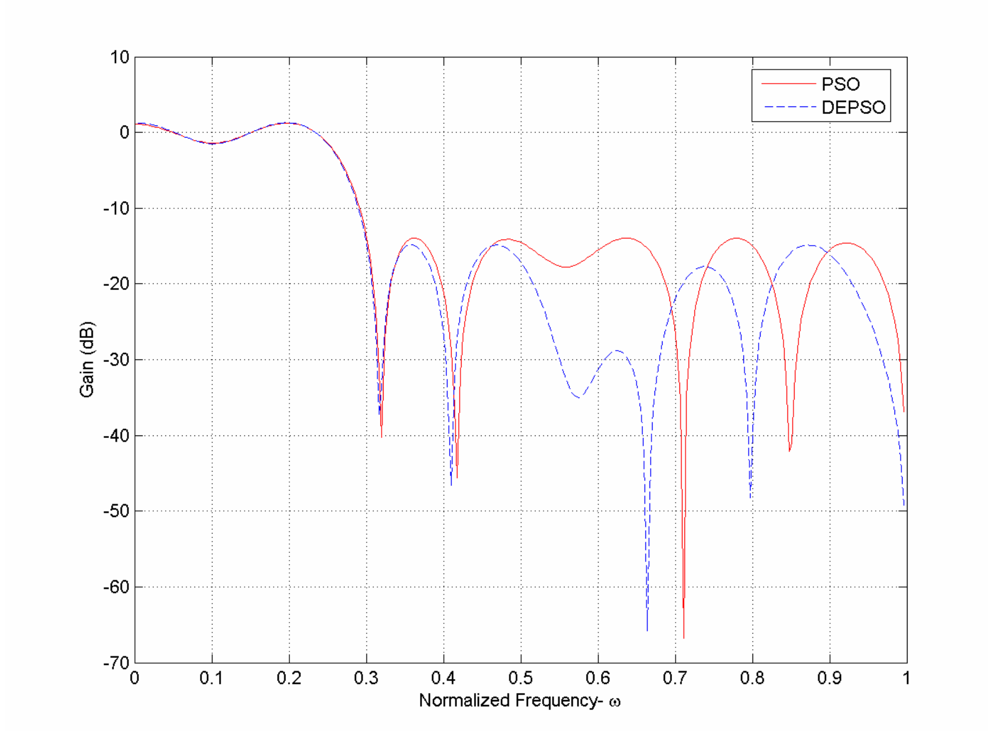

Figure 8.4. Gain response for the filters designed in Case I in 200 iterations.

The error graph and gain plot for the filters designed in Case II are presented in the Figs. 8.5 and 8.6 respectively.

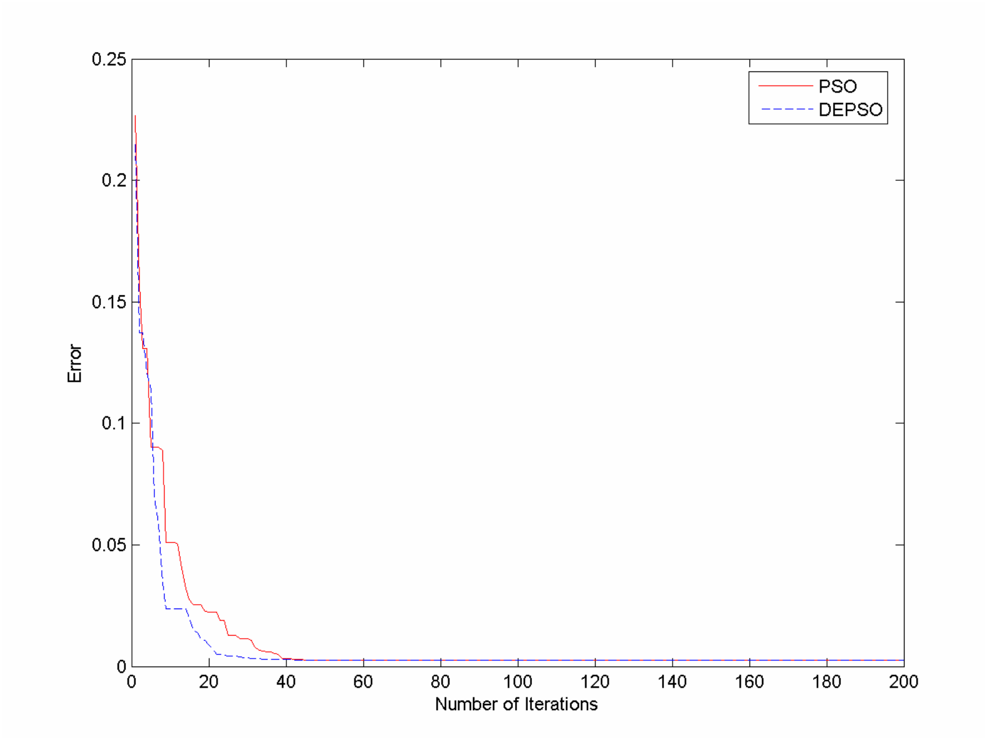

Figure 8.5. Error graph for Case II in 200 iterations.

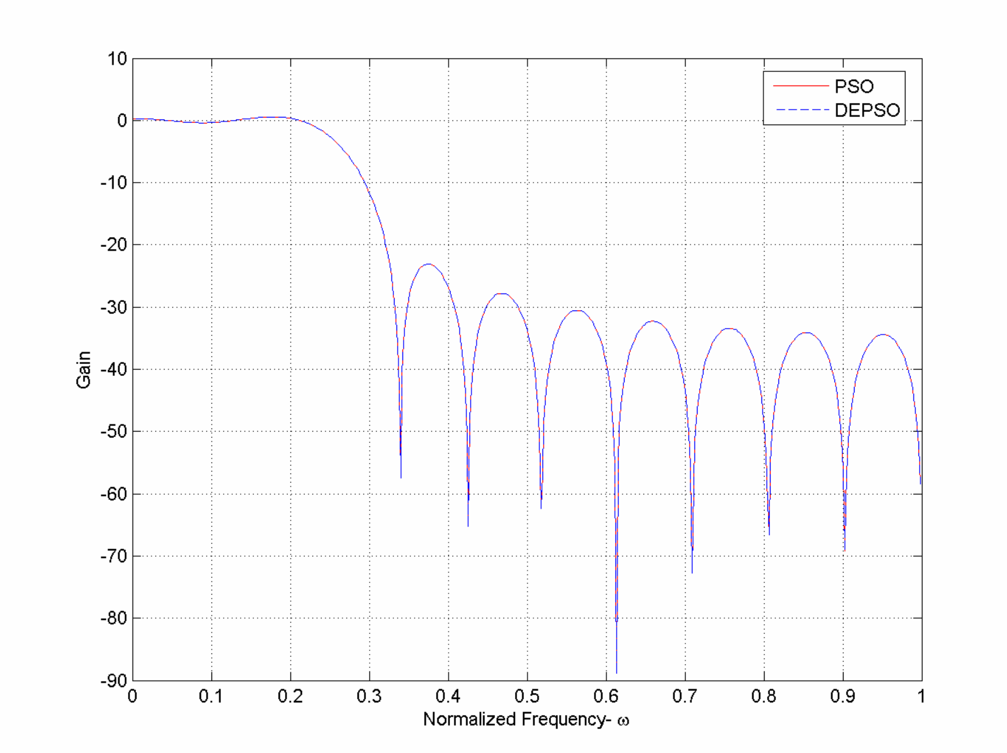

Figure 8.6. Magnitude plot for the filters designed in Case II in 200 iterations.

The two cases represent the two kinds of fitness function used. This effect of fitness function in the magnitude and gain response of the designed filters is shown in the following figures. The comparison of magnitude response for DEPSO is shown in Fig. 8.7. The comparison of gain response for PSO is shown in Fig. 8.8.

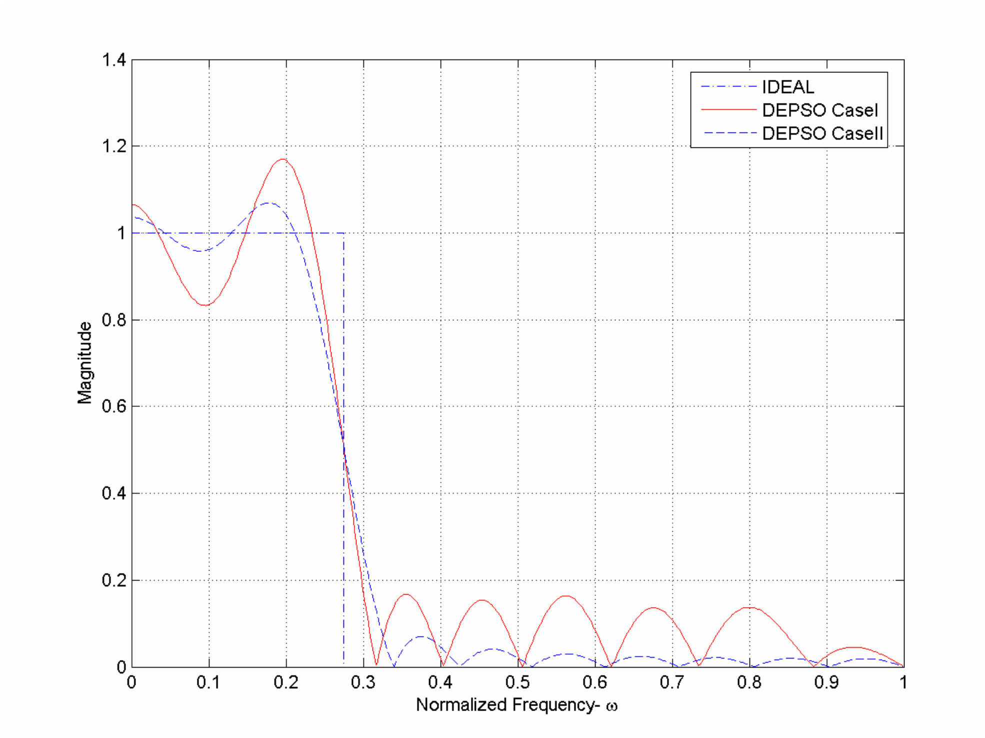

Figure 8.7. Comparison of magnitude response for the two cases.

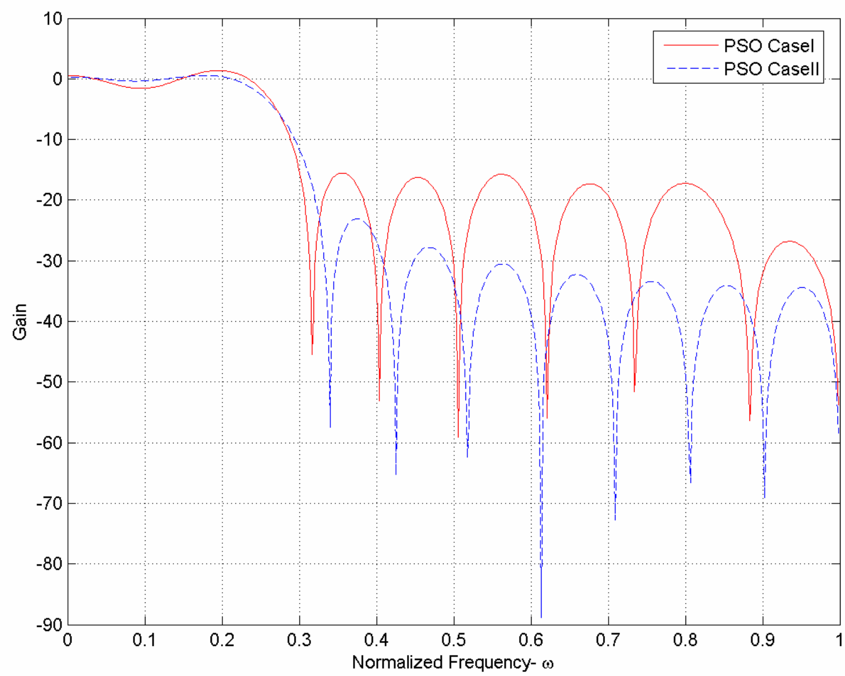

Figure 8.8. Comparison of gain response for the two cases.

Both the cases were also subjected to 40 iterations, in which case better performance of DEPSO is evident. The Figs. 8.9, 8.10 and 8.11 show the error graph, magnitude plot and gain plot for Case I run for 40 iterations.

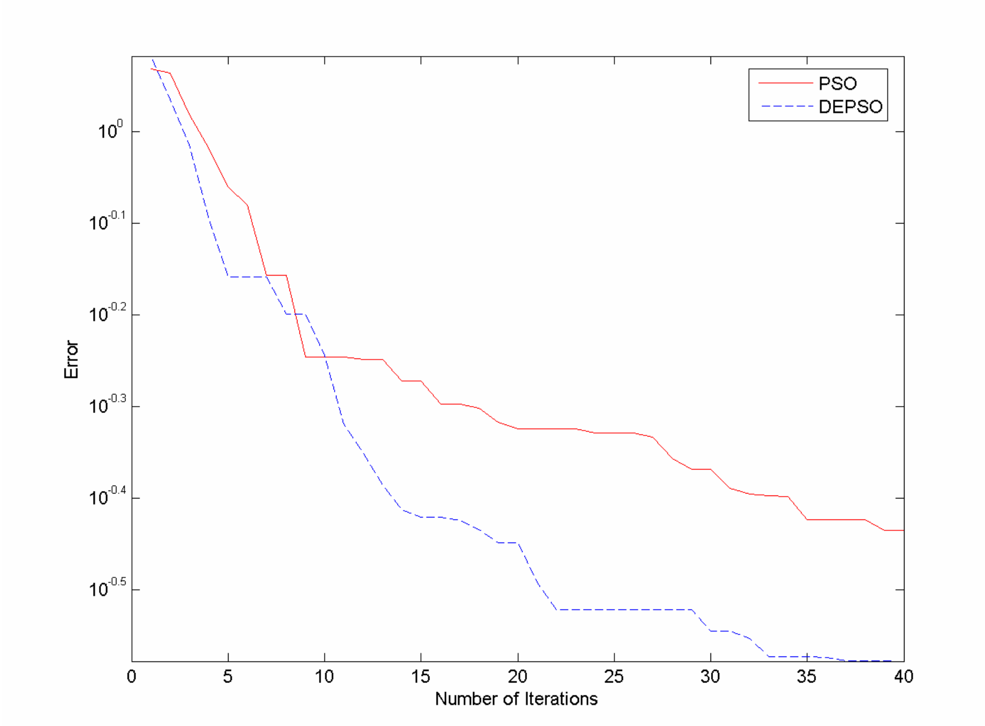

Figure 8.9. Error graph for Case I in 40 iterations.

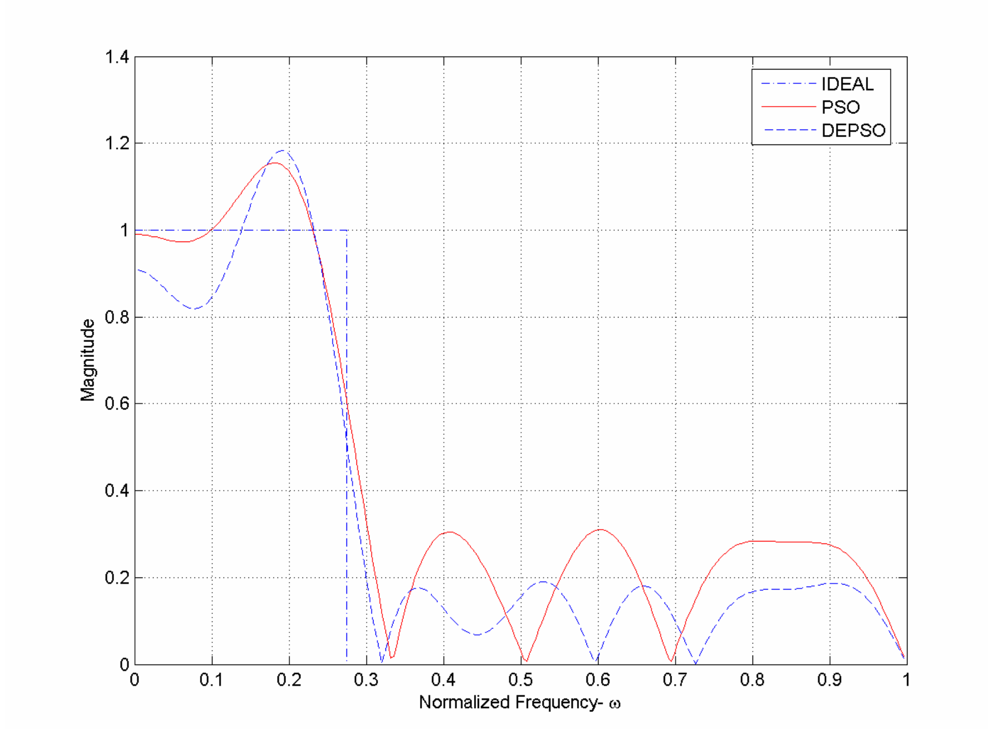

Figure 8.10. Magnitude response of the filters designed in Case I in 40 iterations.

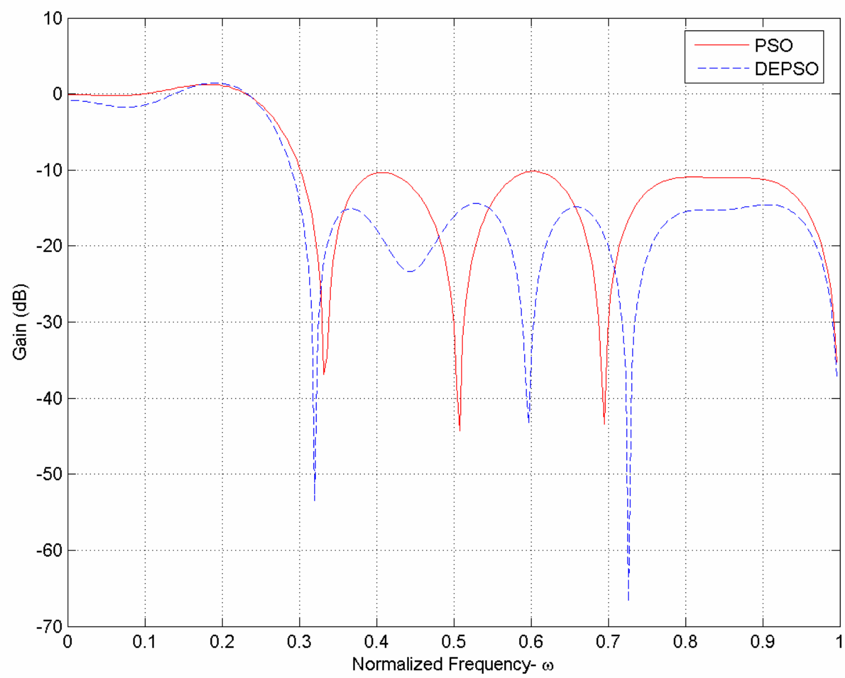

Figure 8.11. Gain plot for the filters designed in Case I in 40 iterations.

The data obtained from the case studies are tabulated below. Table 8.3 presents the data for the two cases in 200 iterations where as Table 8.4 presents the same set of data for 40 iterations. These data have been compared with the results obtained in [2].

Table 8.3. Passband and stopband ripples with 200 iterations

		PSO		DEPSO		Ref [2]
		Case I	Case II	Case I	Case II	
Time (s)	Avg.	9.054	8.899	9.396	9.382	<60
	Min.	8.828	8.796	9.032	9.267	<60
Passband (δ_p)	Avg.	0.174	0.257	0.195	0.257	0.073
	Min.	0.166	0.257	0.169	0.257	0.071
	Max.	0.200	0.257	0.218	0.257	0.075
	Std.	0.009	0.000	0.014	0.000	0.0013
Stopband (δ_s)	Avg.	0.160	0.259	0.182	0.259	0.073
	Min.	0.141	0.259	0.158	0.259	0.071
	Max.	0.185	0.259	0.235	0.259	0.075
	Std.	0.012	0.000	0.025	0.000	0.0013

Table 8.4. Passband and stopband ripples with 40 iterations

		PSO		DEPSO		Ref [2]
		Case I	Case II	Case I	Case II	
Time (s)	Avg.	3.108	3.077	4.573	3.015	<60
	Min.	3.021	2.954	3.200	2.875	<60
Passband (δ_p)	Avg.	0.169	0.275	0.172	0.269	0.073
	Min.	0.124	0.256	0.152	0.253	0.071
	Max.	0.266	0.290	0.194	0.291	0.075
	Std.	0.041	0.016	0.018	0.016	0.0013
Stopband (δ_s)	Avg.	0.124	0.263	0.203	0.245	0.073
	Min.	0.190	0.246	0.169	0.207	0.071
	Max.	0.262	0.275	0.257	0.270	0.075
	Std.	0.063	0.012	0.041	0.027	0.0013

8.4. RESULTS WITH PSO-QI

PSO-QI performed the best among all of the algorithms used in different cases. In Case III, Lowpass, Highpass, Bandpass and Bandstop FIR filters were designed. PSO-QI performed the best in terms of its consistency as well as ability to converge to a lower value of average error. However, its execution time is almost two times that of either PSO or DEPSO. This is the trade-off over performance in case of PSO-QI. Similar results are obtained for the design of LP IIR filter using the same fitness function. In Case IV, second fitness function was used to design LP, HP, BP and BS FIR filters. In this case also, PSO-QI performed better than PSO and DEPSO. All FIR filters have been plotted against the filter designed using standard Parks McClellan method. IIR filter has been plotted against standard elliptical window based filter design. The performance of filters designed using CI algorithms is better than the filters designed using standard techniques in all cases. In Case III, these techniques are either comparable to or better than the standard techniques and in Case IV, the CI techniques perform much better than the standard techniques, as is observed in the magnitude and gain plots and the tabulated results shown below. However, IIR filter designed using CI techniques is unable to gain as sharp transition as an IIR filter designed using elliptical window technique.

Although cases have not been studies exclusively for QPSO but a new algorithm PSO-QI derived from it, it was used to show a comparison in order to prove that PSO-QI is better than QPSO. QPSO could perform better than PSO but only when allowed to run for a large number of iterations. It possessed the ability to escape the local minima, which made it a better choice over PSO but it converged very slowly. Therefore it is allowed to run for 1500 iterations to see any effect. In 1000 iterations, QPSO outperformed PSO. In more than 1500 iterations it is comparable to PSO-QI. However, it continues to converge until 4000 iterations and more. This suggests that QPSO could converge to a better fitness if given enough time. But PSO-QI which combines the same convergence characteristics of QPSO with PSO, converges much faster. Hence, depending upon the requirements of time and amount of convergence according to applications and/or design environment, the trade-off can be maintained. The error graph in Fig. 8.12 shows the performance of QPSO compared with the other two algorithms.

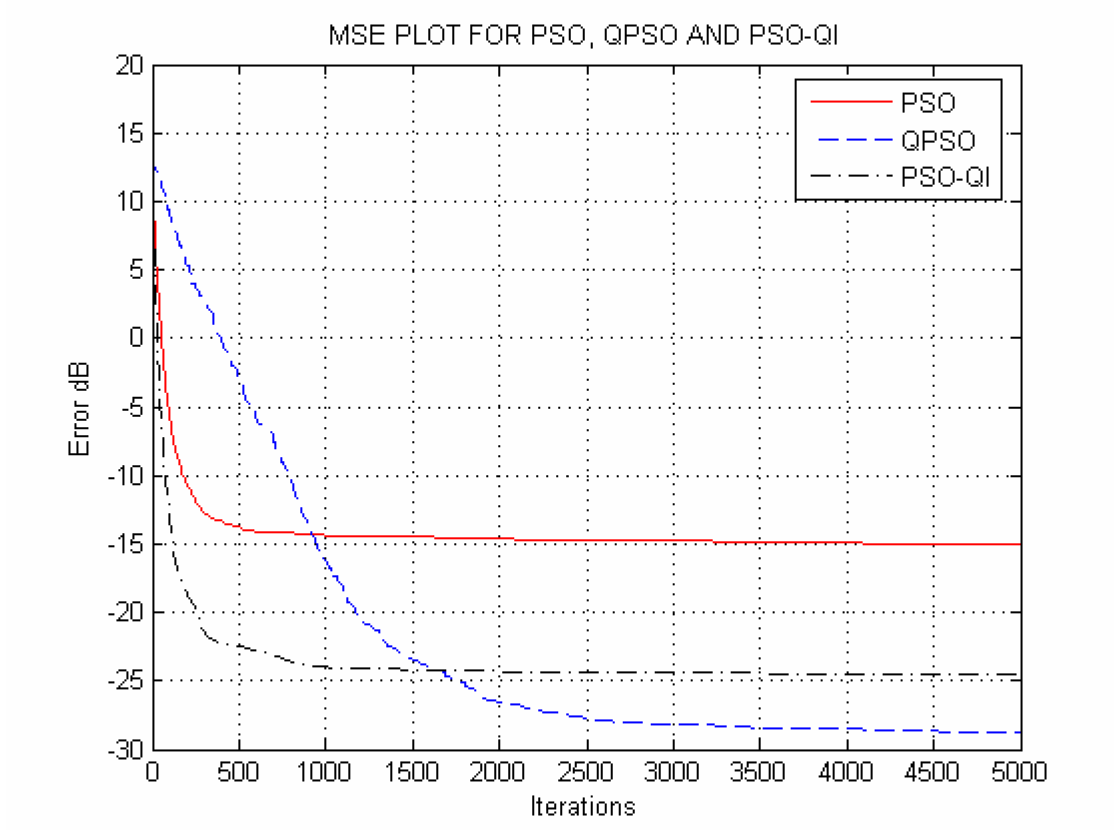

Figure 8.12. Error graph showing the comparison of PSO, QPSO and DEPSO.

The error graph, magnitude response and gain response for the LP FIR filter designed in Case III are shown in Figs. 8.13, 8.14 and 8.15 respectively.

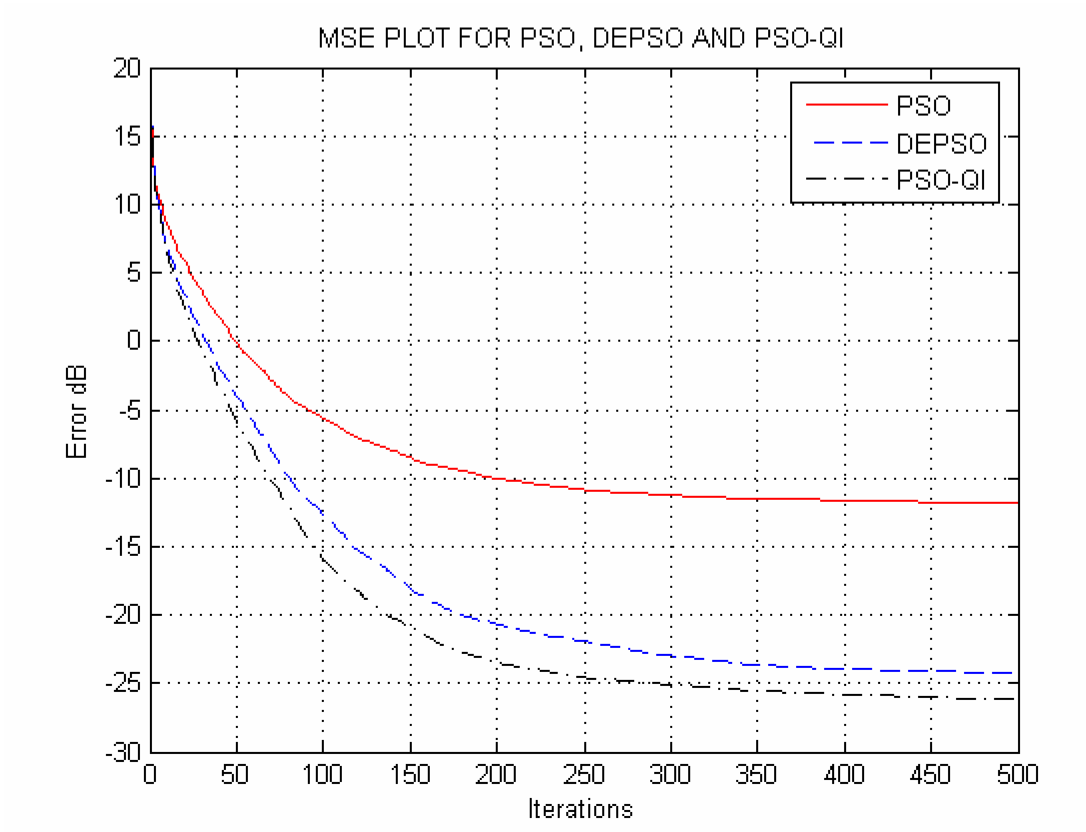

Figure 8.13. Error graph for LP FIR filter designed in Case III

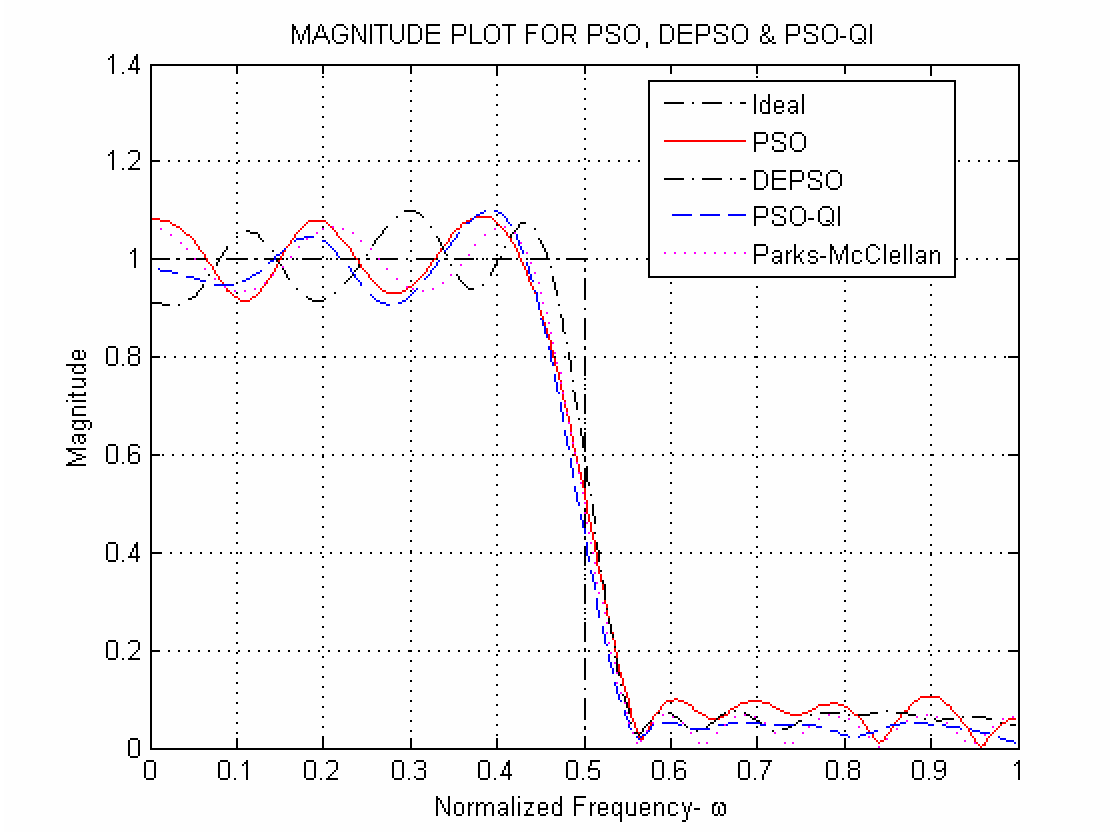

Figure 8.14. Magnitude response of the LP FIR filter designed in Case III

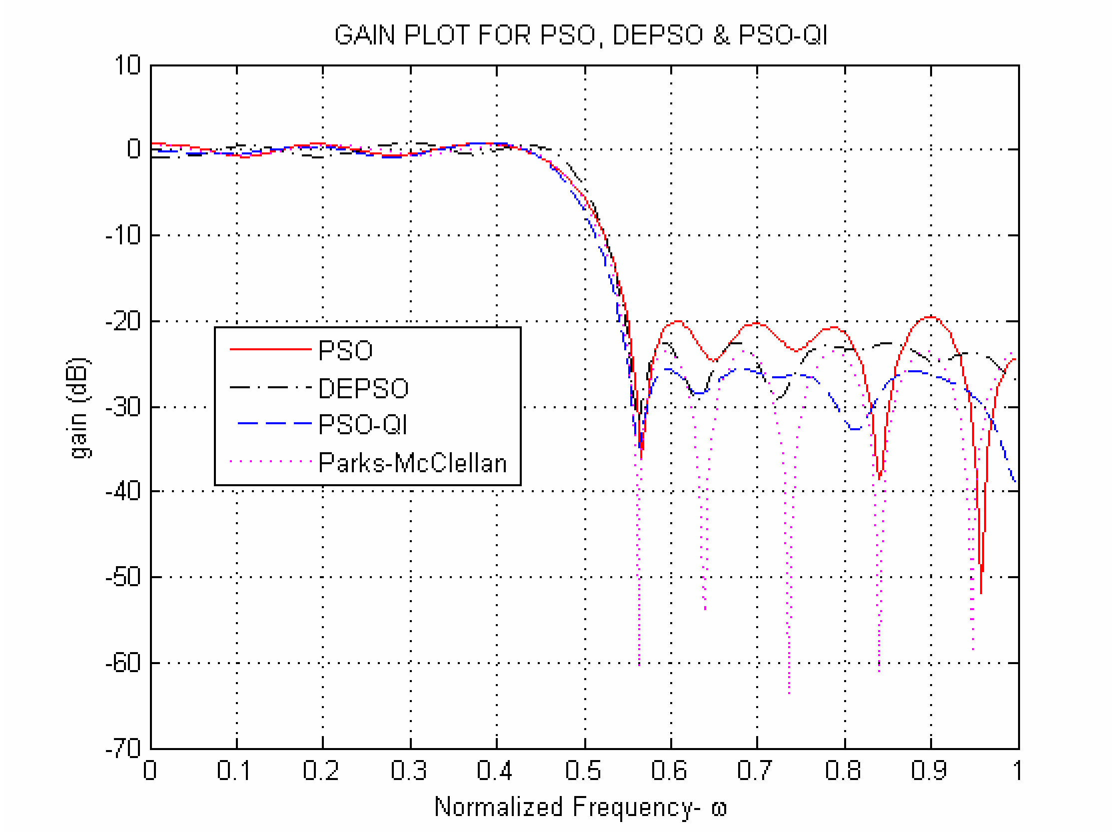

Figure 8.15. Gain response of the LP FIR filter designed in Case III.

The error graph, magnitude response and gain response for the LP IIR filter designed in Case III are shown in Figs. 8.16, 8.17 and 8.18 respectively.

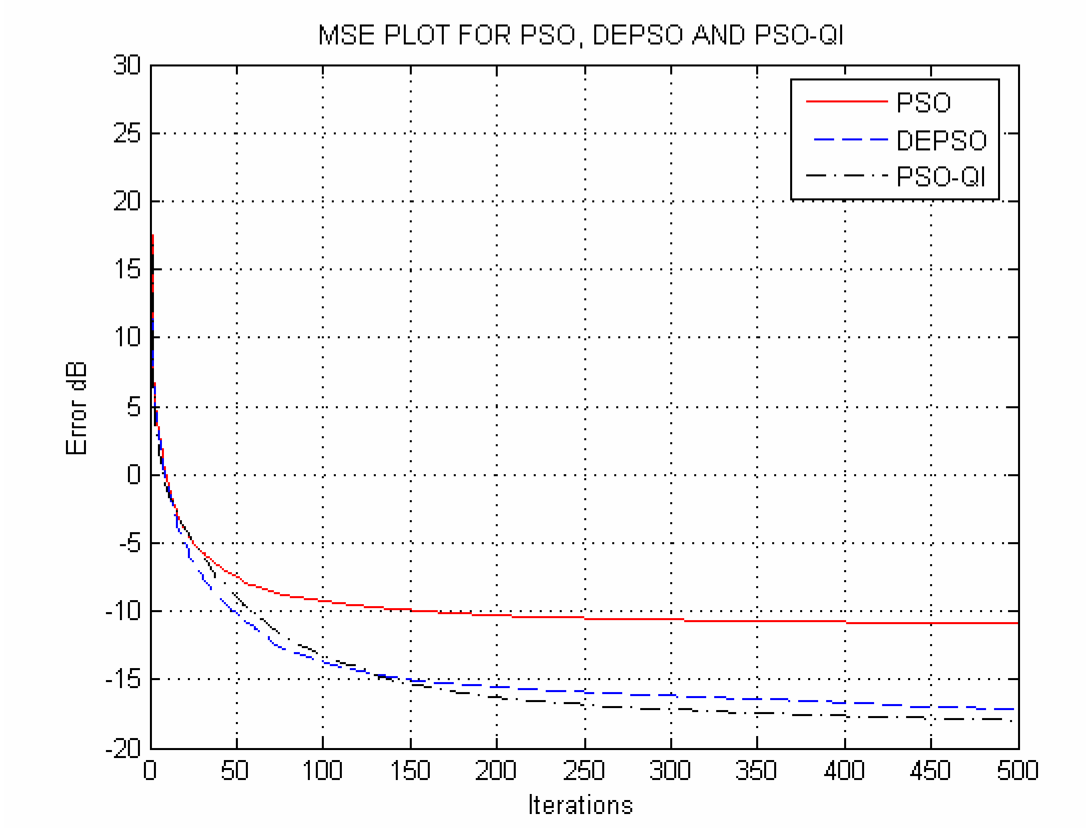

Figure 8.16. Error graph for the LP IIR filter designed in Case III

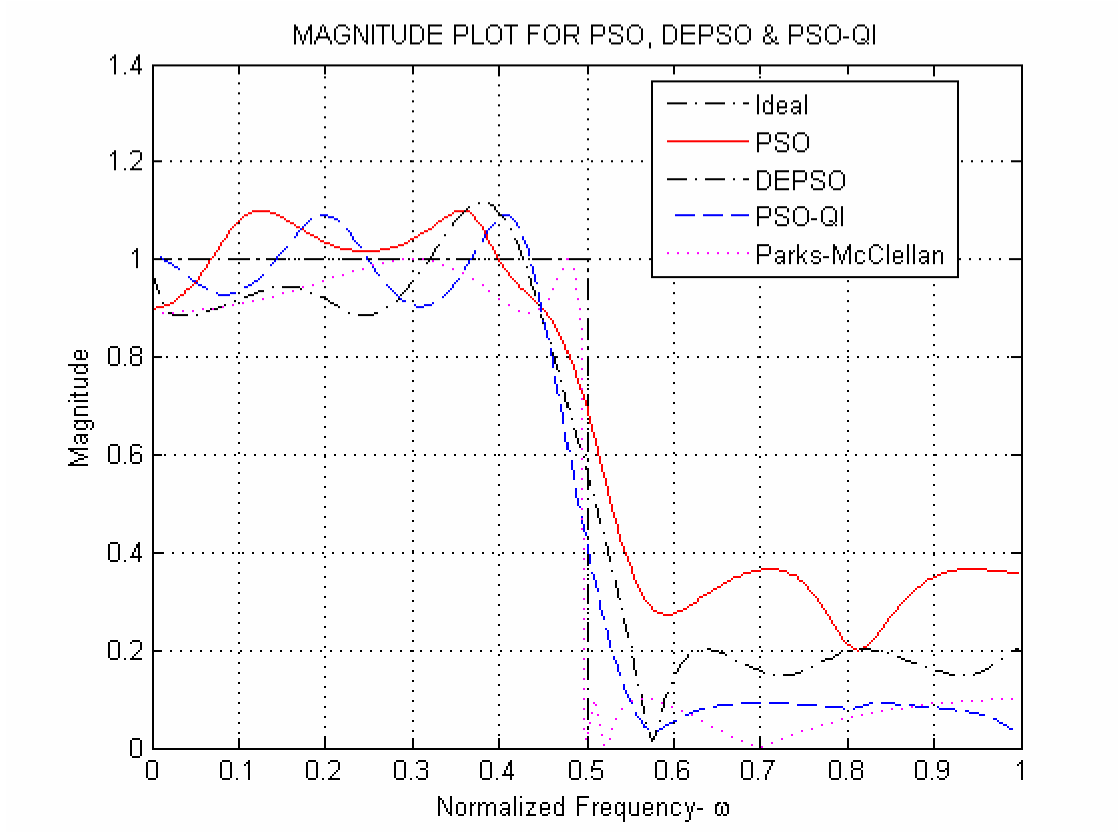

Figure 8.17. Magnitude response for the LP IIR filter designed in Case III

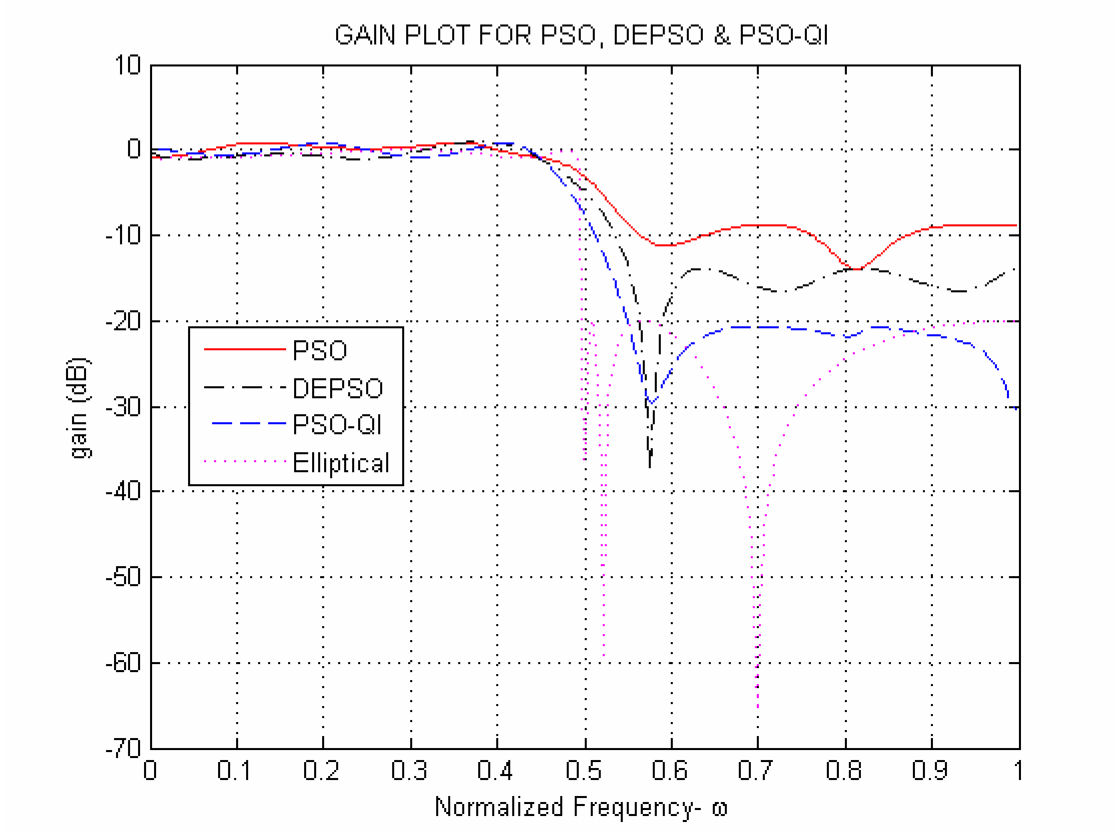

Figure 8.18. Gain response for the LP IIR filter designed in Case III

The magnitude and gain response of the HP FIR filter designed in Case III are shown in Figs. 8.19 and 8.20, respectively.

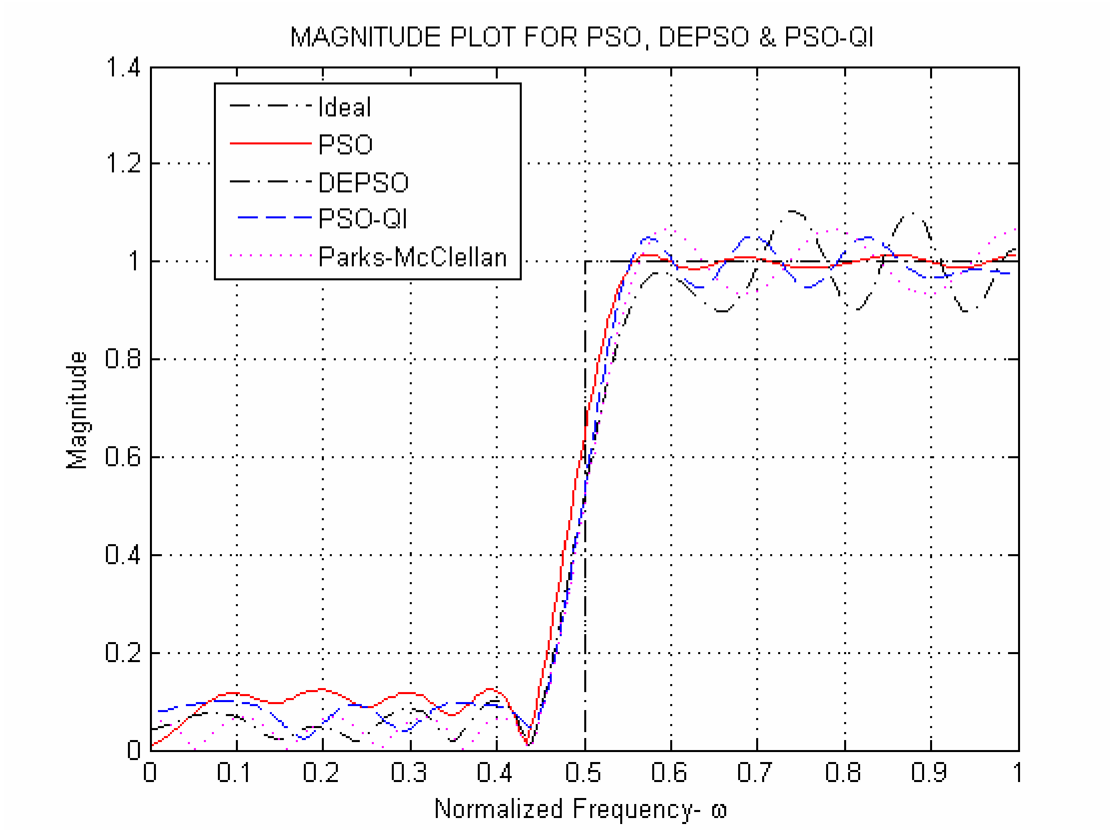

Figure 8.19. Magnitude response of HP FIR filter designed in Case III

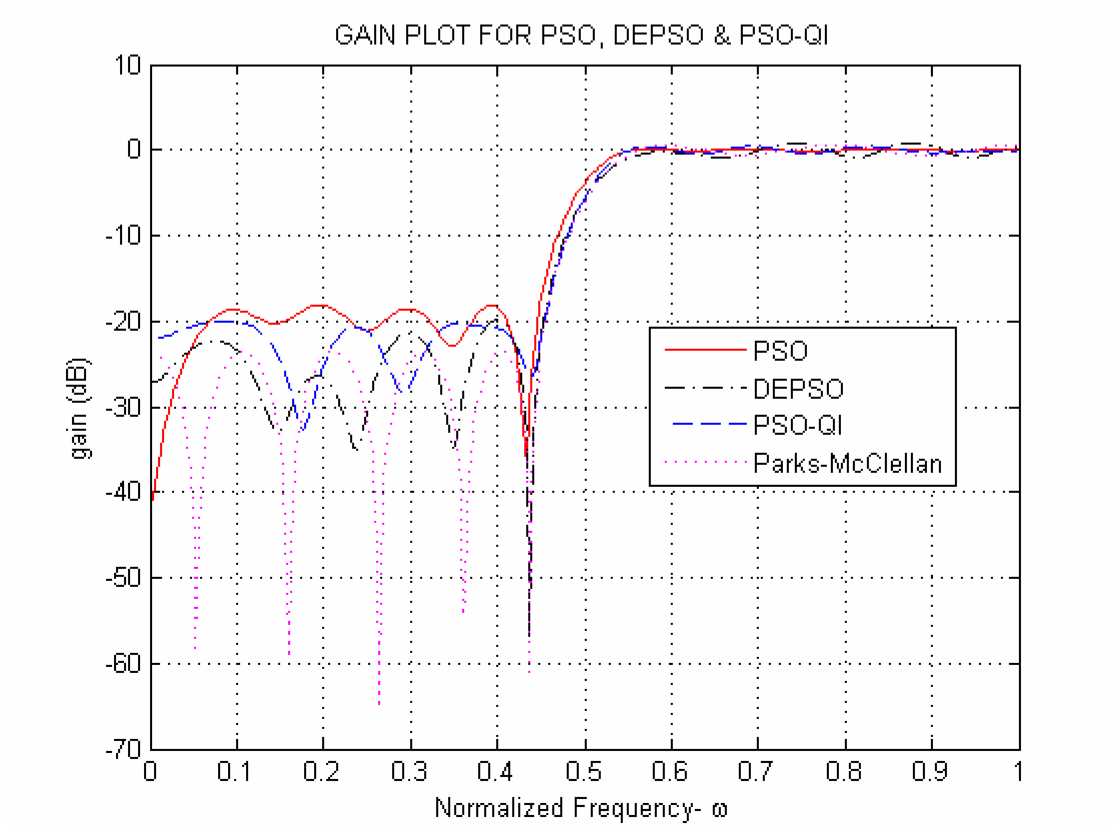

Figure 8.20. Gain response of HP FIR filter designed in Case III

The error graph, magnitude plot and gain plot for the BP FIR filter designed in Case III are shown in the Figs. 8.21, 8.22 and 8.23 respectively.

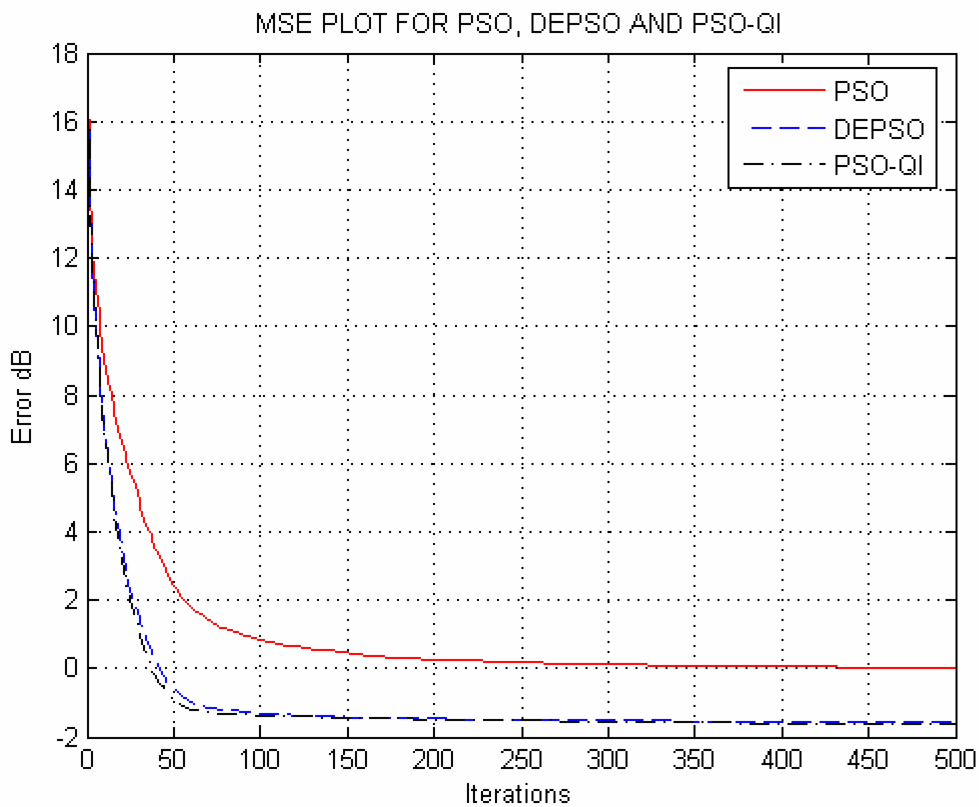

Figure 8.21. Error graph for the BP FIR filter designed in Case III

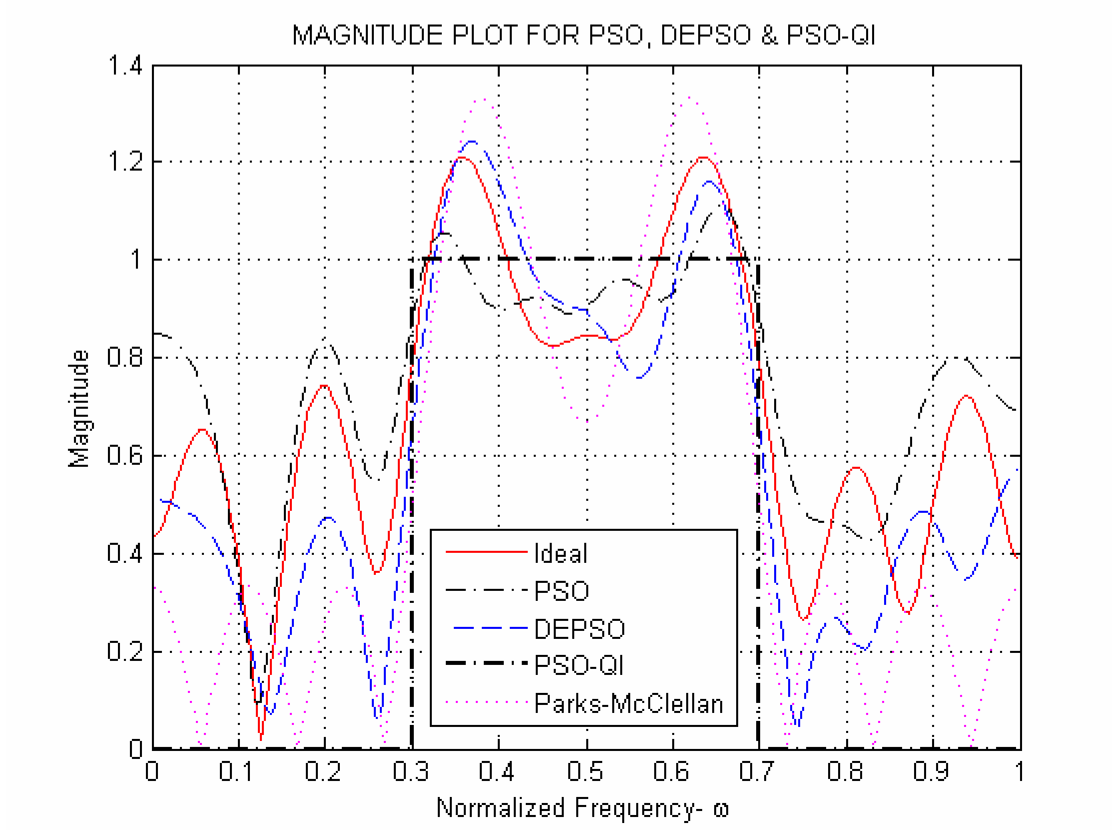

Figure 8.22. Magnitude plot of the BP FIR filter designed in Case III

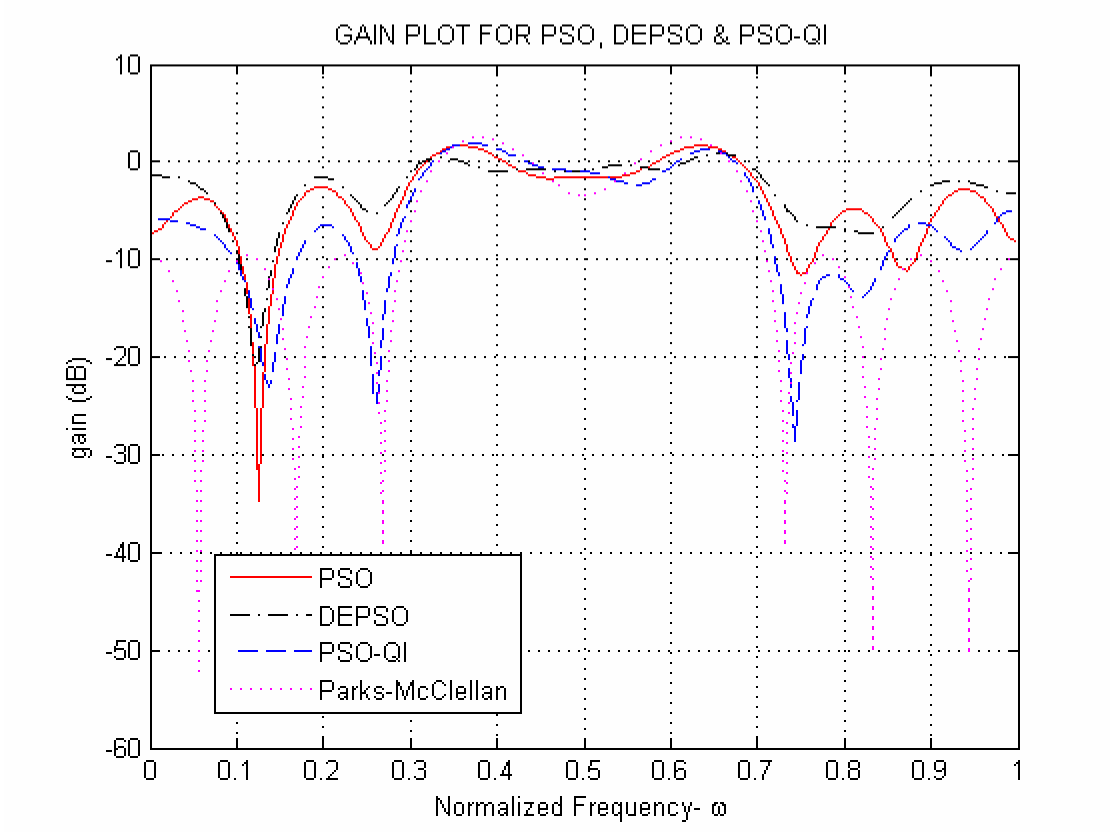

Figure 8.23. Gain plot of the BP FIR filter designed in Case III

In the following Figs. 8.24 and 8.25, magnitude and gain response of the BS FIR filter designed in Case III have been shown.

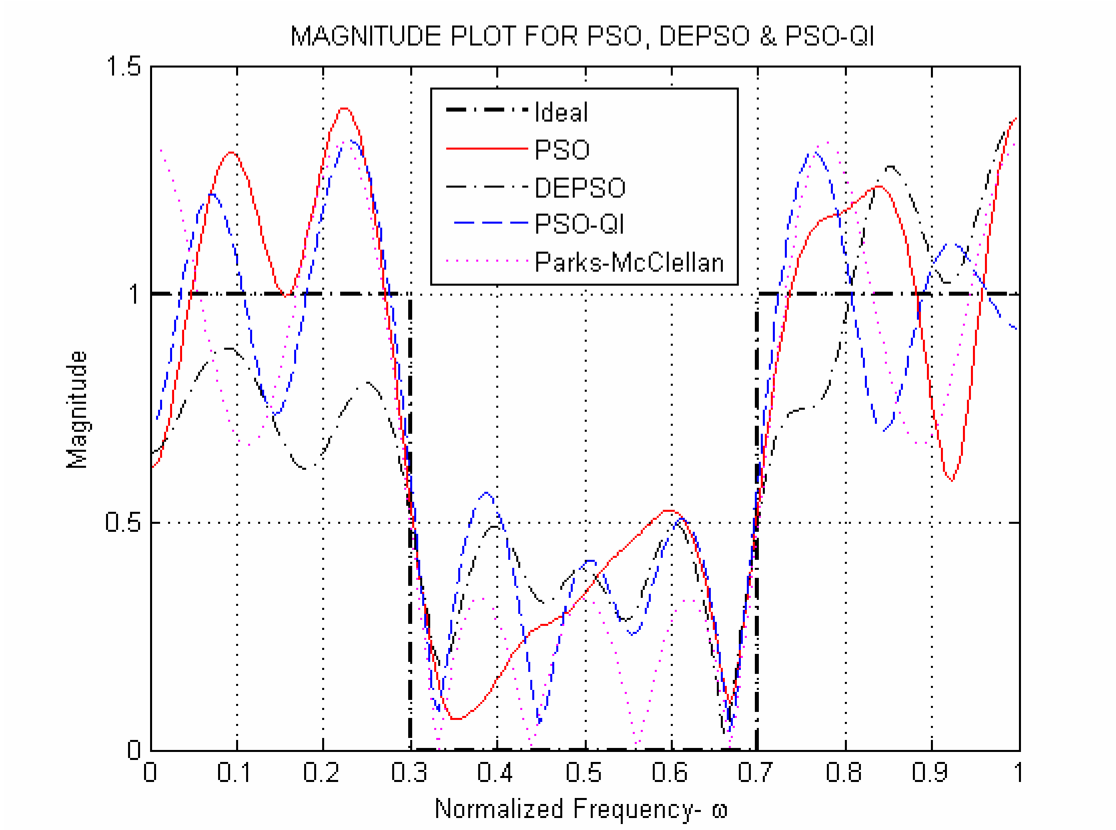

Figure 8.24. Magnitude response of the BS FIR filter designed in Case III

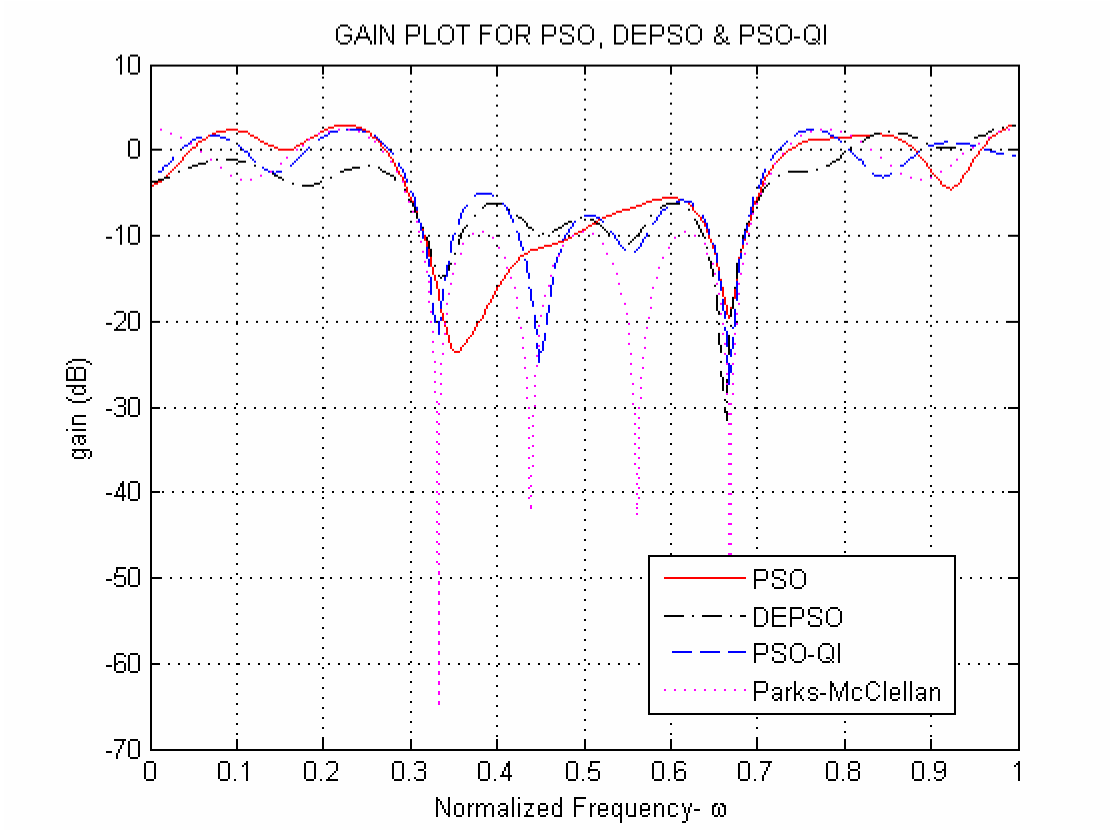

Figure 8.25. Gain response of the BS FIR filter designed in Case III

In Case IV, a different fitness function is used to design the digital FIR filters. The error graph, magnitude response and gain response of the LP FIR filter designed in this case are shown in Figs. 8.26, 8.27 and 8.28 respectively.

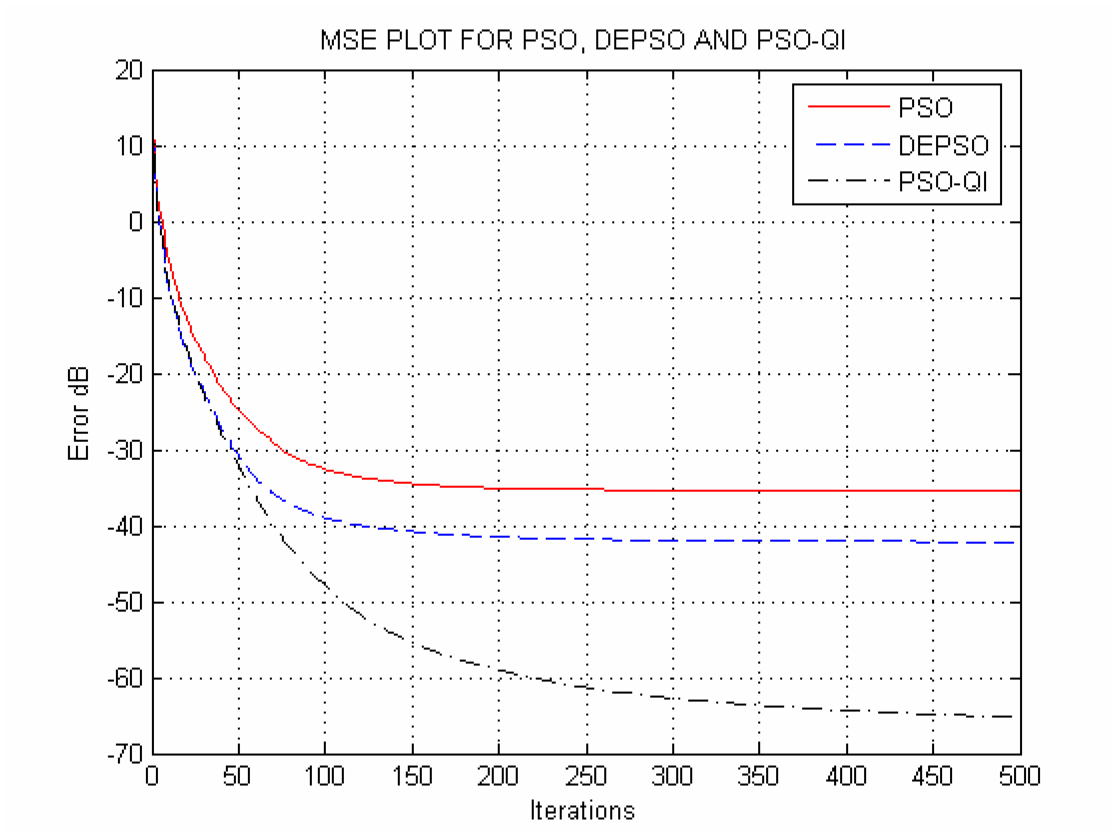

Figure 8.26. Error graph for the LP FIR filter designed in Case IV

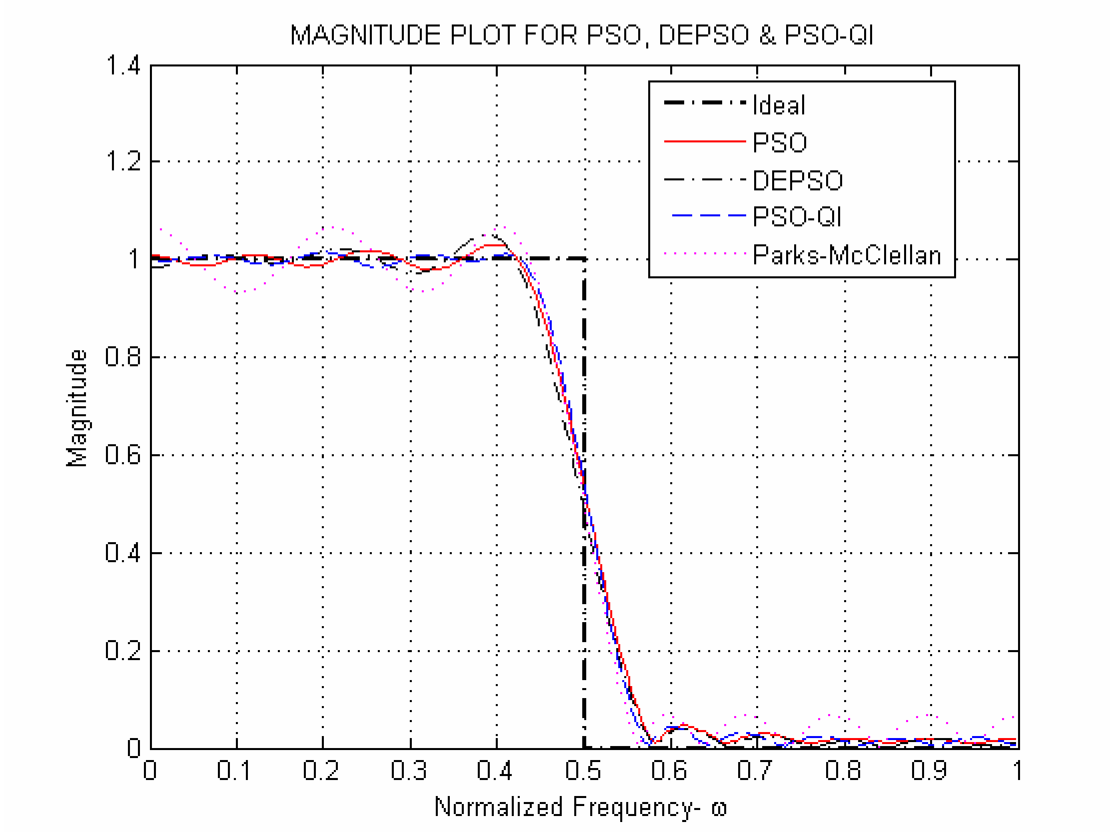

Figure 8.27. Magnitude plot for the LP FIR filter designed in Case IV

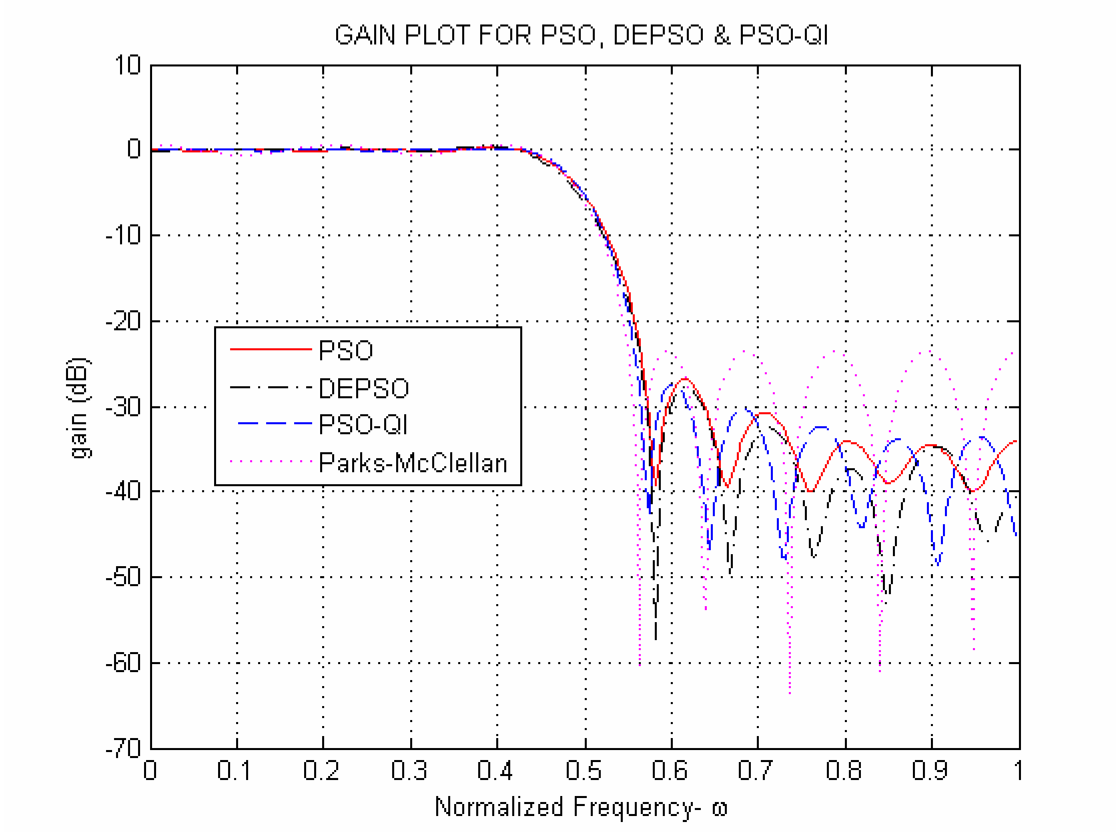

Figure 8.28. Gain plot for the LP FIR filter designed in Case IV

Figs. 8.29 and 8.30 represent the magnitude and gain plots respectively, of the HP FIR filter designed in Case IV.

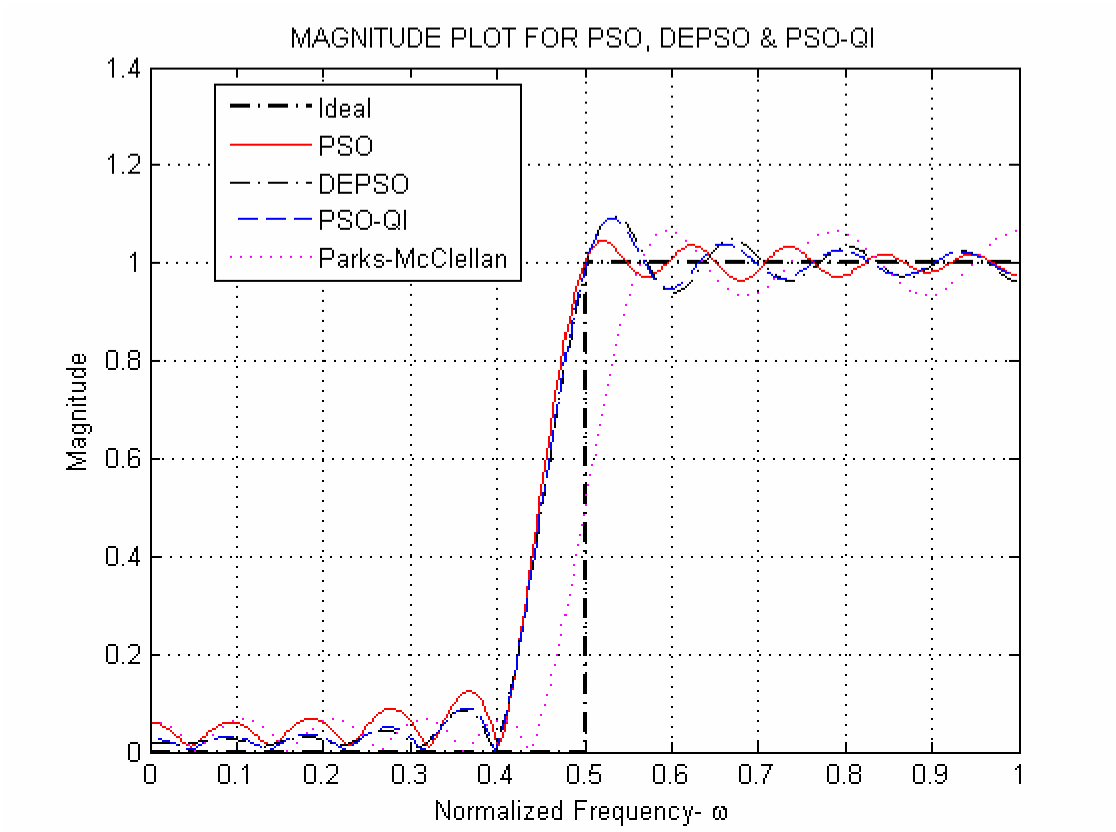

Figure 8.29. Magnitude plot of the HP FIR filter designed in Case IV

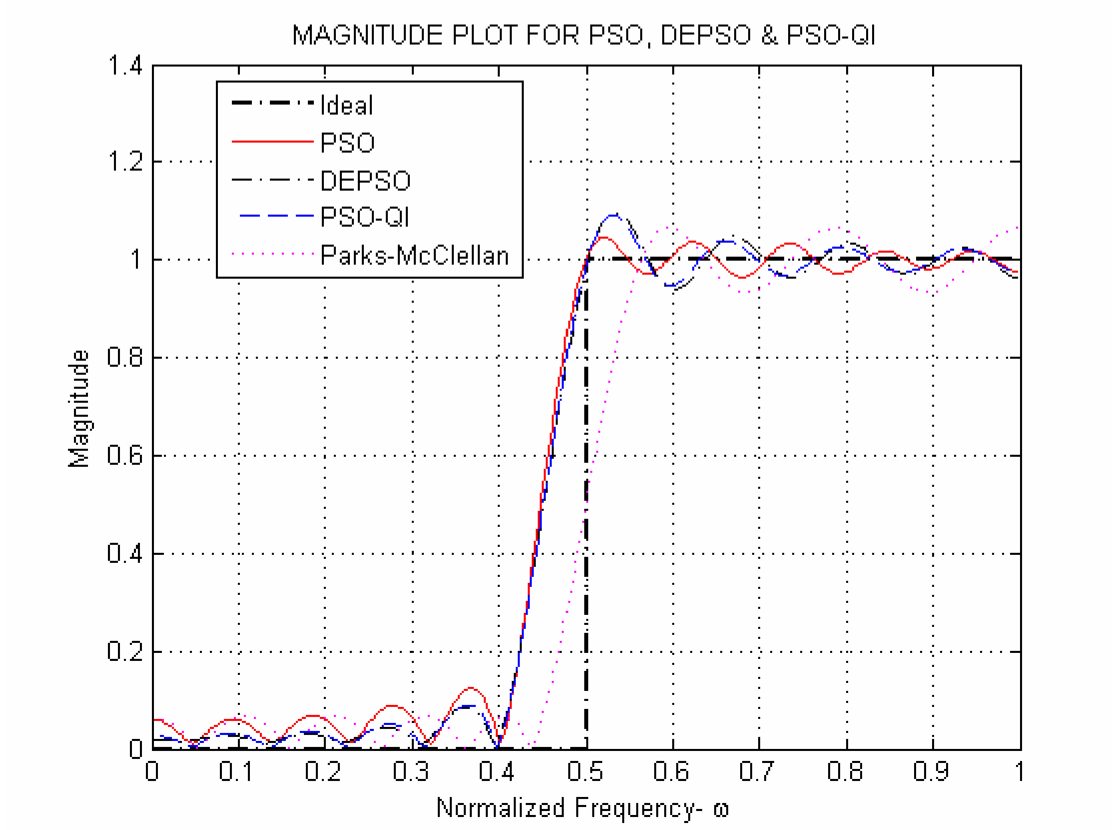

Figure 8.30. Gain plot of the HP FIR filter designed in Case IV

The error graph, magnitude plot and gain plot of the BP FIR filter designed in Case IV are shown in Figs. 8.31, 8.32 and 8.33, respectively.

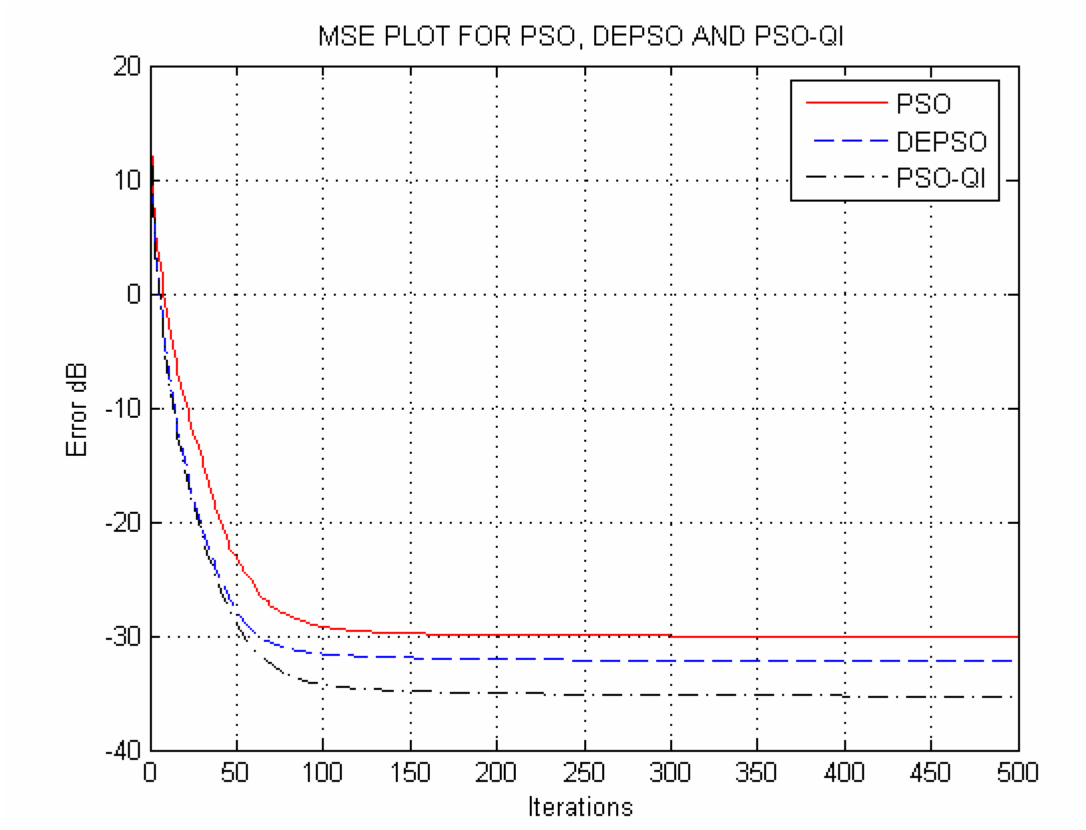

Figure 8.31. Error graph for the BP FIR filter designed in Case IV

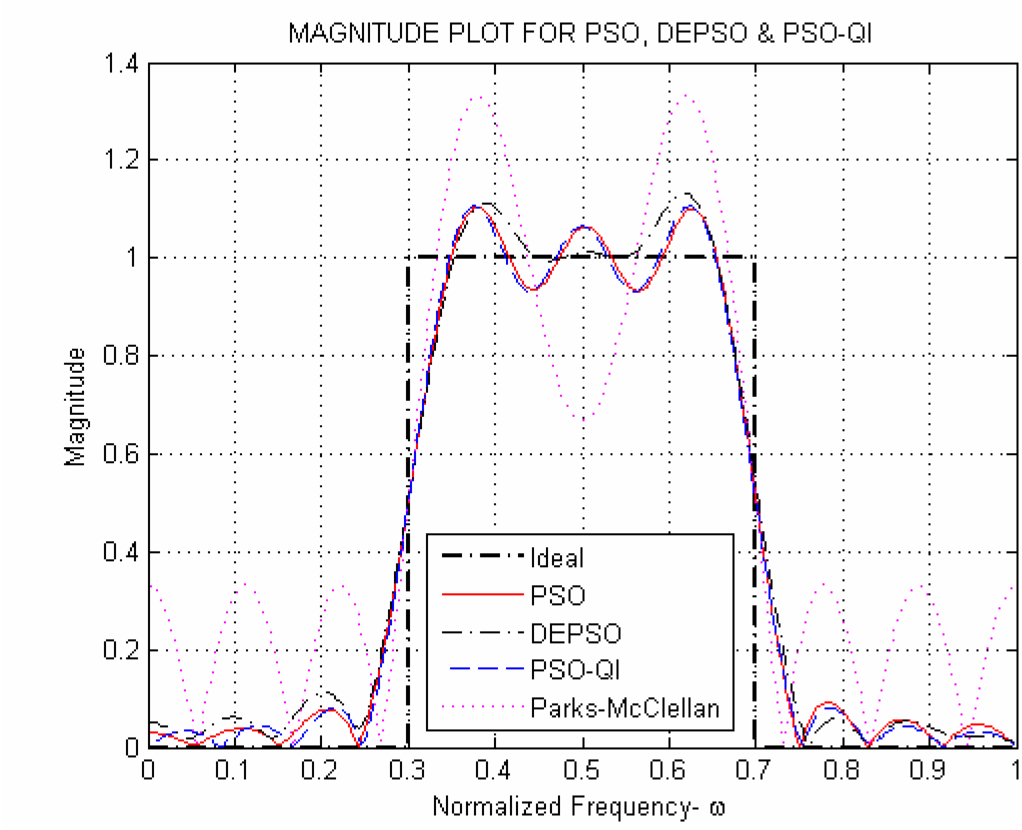

Figure 8.32. Magnitude response of the BP FIR filter designed in Case IV

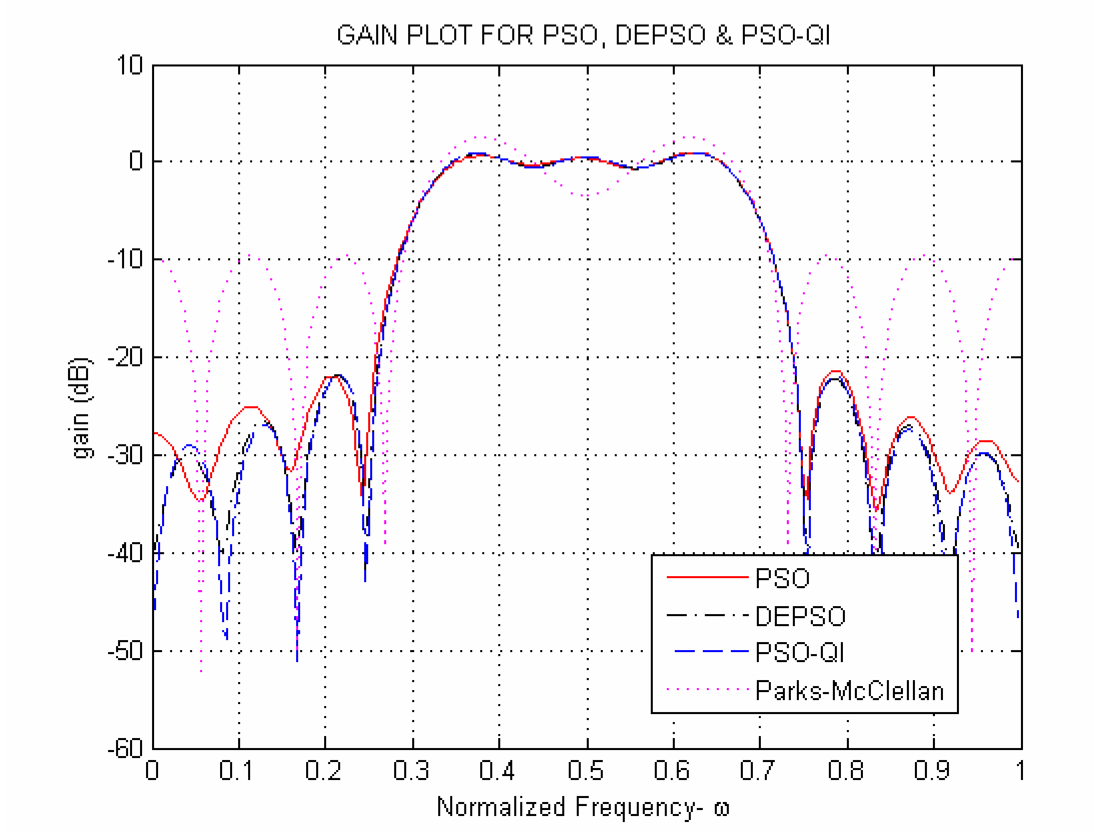

Figure 8.33: Gain response of the BP FIR filter designed in Case IV

The magnitude and gain response of the BS FIR filter designed in Case IV are shown in the following Figs. 8.34 and 8.35 respectively.

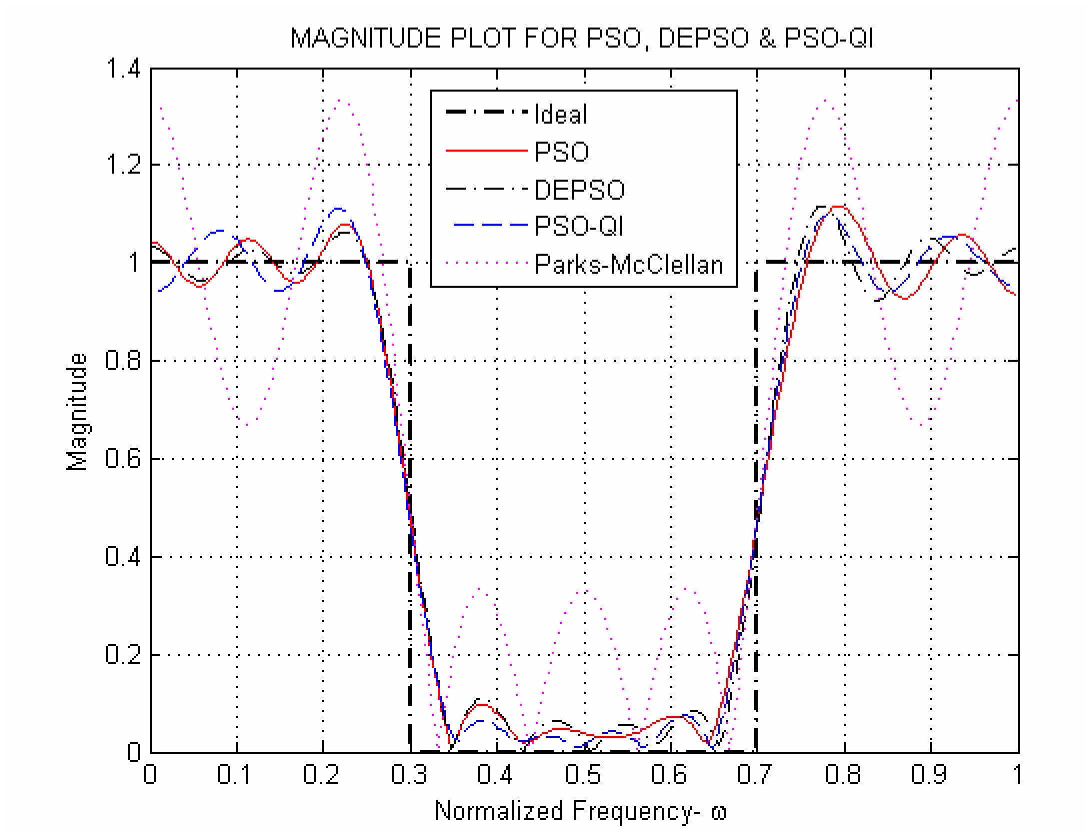

Figure 8.34. Magnitude response of the BS FIR filter designed in Case IV

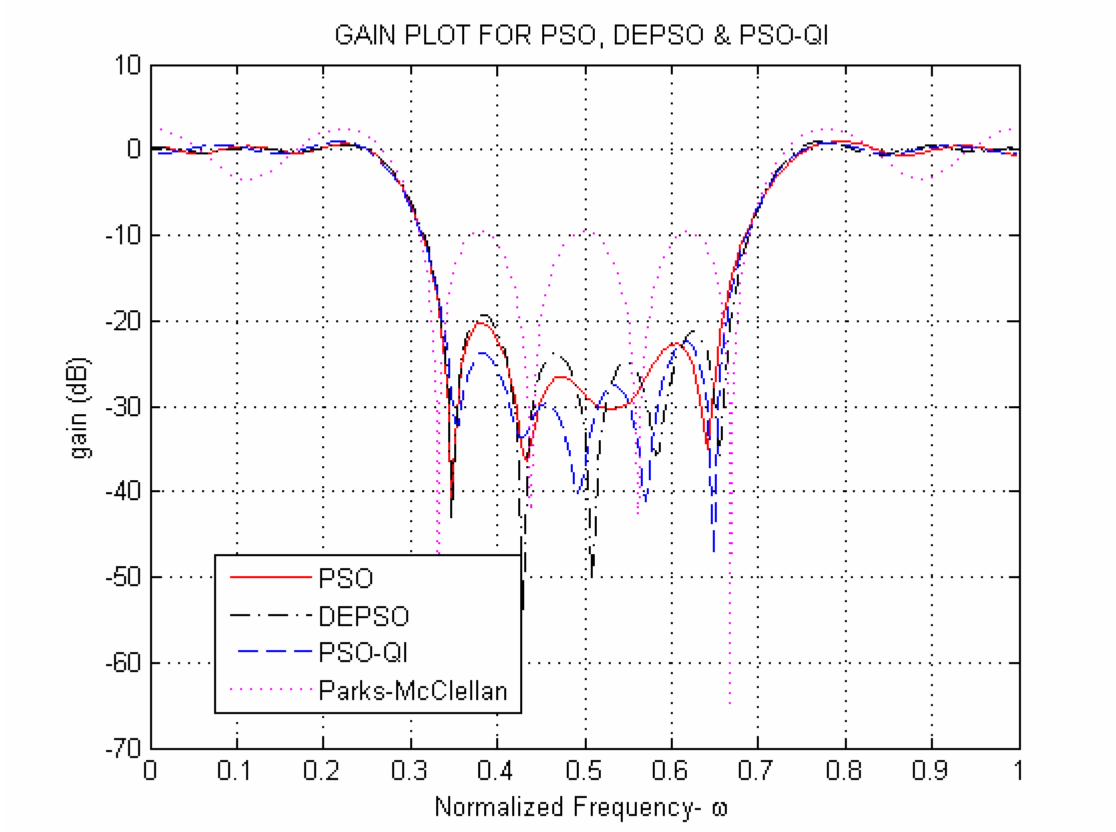

Figure 8.35. Gain response of the BS FIR filter designed in Case IV

PSO-QI took twice as much time as PSO in the finding the solution. Hence a test was done to allow PSO to run for as much time as is taken by PSO-QI and see if it performs as good as PSO-QI. However, PSO could not converge to a lower average error. This has been shown in Fig. 8.36.

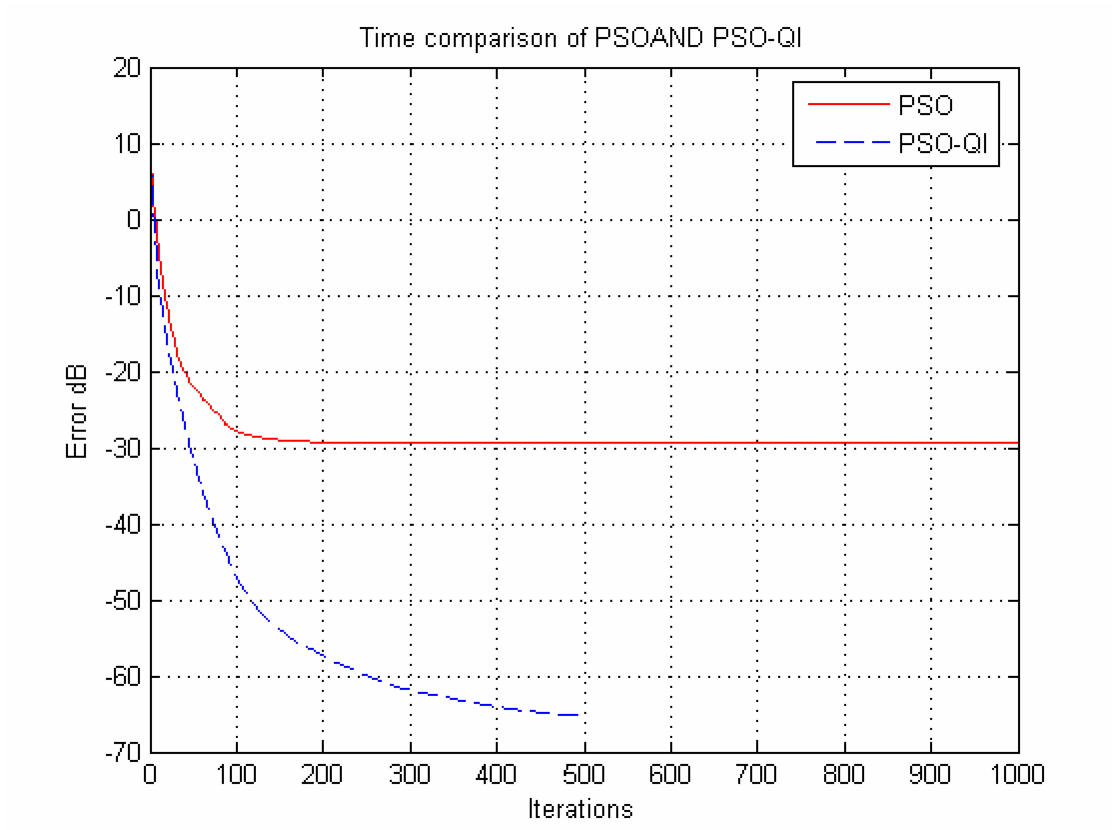

Figure 8.36. Comparison of PSO and PSO-QI in terms of time

The data obtained from the case studies are presented in Table 8.5. This shows the minimum, maximum and standard deviation values of passband and stopband ripples for each of the algorithms. It also shows the time taken by each algorithm for 500 iterations. The results are clearly in favor of PSO-QI except for the time.

Table 8.5. Passband and stopband ripples for FIR filter in Case III*

		PSO		DEPSO		PSO-QI		Parks McClellan	
		LP	BP	LP	BP	LP	BP	LP	BP
Time (s)	Avg.	16.780	17.010	17.440	17.006	32.398	32.528	0.1560	0.1570
	Min.	16.203	16.781	16.938	16.110	31.656	31.593		
Error	Avg.	0.256	1.003	0.061	0.833	0.049	0.828	1.3692	2.2196
	Min.	0.019	0.819	0.019	0.812	0.014	0.808		
	Std.	0.430	0.394	0.032	0.013	0.026	0.007		
Passband (δ_p)	Avg.	0.175	0.240	0.102	0.304	0.102	0.335	0.9988	0.9969
	Min.	0.099	0.099	0.099	0.104	0.099	0.226		
	Max.	1.297	0.647	0.133	0.517	0.157	0.476		
	Std.	0.222	0.112	0.006	0.084	0.010	0.063		
Stopband (δ_s)	Avg.	0.190	0.873	0.069	0.641	0.056	0.602	0.4804	1.3327
	Min.	0.029	0.414	0.029	0.431	0.024	0.452		
	Max.	1.173	2.324	0.189	0.883	0.133	0.717		
	Std.	0.281	0.408	0.030	0.092	0.025	0.066		

*Carried out on the same computer for 500 iterations

Table 8.6. Passband and stopband ripples for the IIR filter in Case III*

		PSO	DEPSO	PSO-QI	Elliptical
		LP	LP	LP	LP
Time (s)	Avg.	18.222	18.989	34.976	0.6560
	Min.	17.453	18.140	33.672	
Error	Avg.	0.285	0.138	0.126	0.9987
	Min.	0.036	0.011	0.014	
	Std.	0.208	0.093	0.093	
Passband (δ_p)	Avg.	0.173	0.130	0.131	0.1087
	Min.	0.099	0.093	0.099	
	Max.	0.378	0.255	0.391	
	Std.	0.085	0.048	0.070	
Stopband (δ_s)	Avg.	0.221	0.119	0.104	1.0000
	Min.	0.027	0.021	0.024	
	Max.	0.894	0.391	0.279	
	Std.	0.175	0.085	0.054	

*Carried out in the same computer for 500 iterations

Table 8.7. Passband and stopband ripples for the FIR filter in Case IV*

		PSO		DEPSO		PSO-QI		Parks McClellan	
		LP	BP	LP	BP	LP	BP	LP	BP
Time (s)	Avg.	14.542	14.852	15.516	15.260	28.891	28.288	0.1560	0.1570
	Min.	14.297	14.781	15.403	15.234	28.251	28.187		
Error	Avg.	0.016	0.031	0.007	0.024	5.458e-4	0.017	0.2912	0.5172
	Min.	3.173e-4	0.016	2.677e-4	0.016	3.189e-4	0.016		
	Std.	0.052	0.071	0.036	0.051	3.299e-4	0.001		
Passband (δ_p)	Avg.	0.177	0.498	0.118	0.492	0.081	0.490	0.9988	0.9969
	Min.	0.033	0.466	0.018	0.464	0.027	0.460		
	Max.	1.154	0.705	0.929	0.625	0.138	0.503		
	Std.	0.289	0.036	0.168	0.021	0.026	0.007		
Stopband (δ_s)	Avg.	0.235	0.500	0.158	0.494	0.118	0.478	0.4804	1.3327
	Min.	0.080	0.450	0.081	0.463	0.078	0.463		
	Max.	1.399	1.105	1.280	1.105	0.210	0.524		
	Std.	0.347	0.122	0.232	0.091	0.026	0.017		

*Carried out on the same computer for 500 iterations

8.5. DISCUSSION

The results shown in the tables and figures show that DEPSO is better than PSO while PSO-QI is the best of the three algorithms under consideration. PSO-QI has obtained the best features of PSO and QPSO and thus presented itself as a powerful algorithm. Its ability to escape the local minima and thus better explore the search space is highlighted by the lower values of average error. Also the lower values of standard deviation confirm its consistency in finding the best result every time. The amount of time taken is justified by the fact that QPSO is carried out on the *gbest* particle after PSO and fitness is calculated. Execution of QPSO mutation turns out to be more time consuming than DE mutation in DEPSO. As a result, DEPSO outperforms the other algorithms when its performance is compared with the amount of time it takes. This still supports its suitability in online adaptations. At the expense of time, better filter response could be obtained by the new hybrid algorithm.

The study still leaves room for more research into the area. PSO-QI has not been subjected to the kind of approach taken in DEPSO-by creating the offspring of the whole population and then carrying out their tournament with the parents. Also, Case III and IV have not been implemented using the fitness function studied in Cases I and II. With further research into the topic, PSO-QI could be used in a wide variety of filter design applications. Trade-off between various parameters of the filter can lead to designing different kinds of filter according to different requirements in various kinds of applications. This also remains to be explored.

8.6. SUMMARY

The result for the various cases of digital filter design was presented in this chapter. The results showed that DEPSO performed better than PSO and PSO-QI performed better than both of the two algorithms. Although the response of the filter in stopband in one of the cases was better in case of PSO, it failed to find the best solution most of the time and deviated highly from the standard value of minimum error. From the studies and their results, it is concluded that combining quantum based approach with classical swarm based search technique can help the swarm communicate with each other more effectively and thus come out of local minima and avoid premature convergence. However, also mentionable is the fact that this lower value of error comes with higher execution time and hence a proper trade-off is required depending upon the kind of application.

9. CONCLUSION

9.1. INTRODUCTION

In this work, swarm, evolutionary and quantum based intelligent optimization algorithms are used in system identification and to design digital filters. It was shown that the swarm based algorithms has many variants and has been hybridized with other algorithms to increase its effectiveness. It was also seen that by hybridization of the algorithms, best features of both the algorithms are retained and thus new algorithm so developed is more robust. In this chapter, a conclusion of all the chapters is provided.

9.2. SECTION SUMMARY

The first three chapters of the thesis cover the introduction to the problem. In Chapter 1, introduction to the thesis is provided. Chapter 2 covers the description of system identification. Introduction to the problem and traditional and modern methods applied to solve it are explained in this chapter. In Chapter 3, digital filter design is explained. This chapter also introduces to the problem of digital filter design and various traditional and new methods applied in the design.

The next three chapters of the thesis describe the involved algorithms and their operation in detail. In Chapter 4, particle swarm optimization has been explained. As one of the pioneer stochastic search optimization technique based on the social behavior of bird flocking and fish schooling, algorithm of PSO has be described in this chapter. In Chapter 5, a hybrid optimization algorithm DEPSO has been explained. A combination of DE and PSO, it uses the differential evolution operation on the *pbest* of *gbest* particle of the PSO to mutate the particle and create an offspring. The chapter covers the detail of its operation. In Chapter 6, another hybrid algorithm, PSO-QI has been introduced. PSO-QI emerges from the infusion of quantum operation obtained from QPSO on the *gbest* particle of the PSO. Concepts of quantum particle swarm optimization and its application on the PSO have been explained in this chapter.

In the next two chapters, the results obtained from the case studies have been presented. In Chapter 7, the results obtained from the application of different algorithms in the system identification have been presented. These results show the effectiveness of the new hybrid algorithms in comparison to the traditional PSO. In Chapter 8, the results for the digital filter design are shown. This chapter shows the results of designing different kinds of digital filters using various algorithms described in the previous chapters. These results also suggest that the new hybrid algorithms are more effective than the traditional PSO. These two chapters present their comparison in terms of figures and tabulated data from different case studies.

9.3. MAIN CONCLUSION

The main focus of the thesis is in system identification and in the design of digital filters. The research work leading to the thesis is related to identification of an IIR system. This is achieved by modeling the unknown system with IIR systems of same or reduced number of orders. In digital filter design, Lowpass, Highpass, Bandpass and Bandstop FIR and Lowpass IIR filters are designed using different optimization algorithms. The results for system identification as well as digital filter design have been shown. In this work, particle swarm optimization is used as the baseline algorithm. Two other algorithms are considered to improve the results obtained from PSO. These are hybrid algorithms based on differential evolution and quantum particle. The DEPSO algorithm performed better than PSO in system identification as well as in digital filter design. Results obtained from PSO-QI are better than both PSO and DEPSO and hence it has outperformed the other two algorithms in all the case studies of system identification and digital filter design.

Fitness function based on passband and stopband ripples of the filter response is used to design both FIR and IIR filters where as the fitness function based on MSE is used to design FIR filters only. It is observed that all three of the algorithms are able to approximate the filter coefficients in a number of iterations but PSO-QI always performed the best among them. Figures and tabulated results all show that PSO-QI is more consistent in its performance and it can achieve a lower value of average error in either of the cases using two different fitness functions. Although it took longer for the algorithm to converge because of its computational complexity, it found much better solution than PSO, DEPSO and QPSO. The results are not tabulated for QPSO because of the higher number of iterations and the results are clear from the figures. However, comparison has been made to confirm that PSO can not achieve the same amount of convergence even when allowed to run for the amount of time taken by PSO-QI. Hence, it can be concluded that swarm, evolutionary and quantum algorithms can be effectively used in digital filter design, and PSO-QI is a better choice. It is evident from the figures and results how the best features of two algorithms can be extracted and performance can be improved by the hybridization of these algorithms.

9.4. FUTURE RESEARCH

This thesis covered application of different optimization algorithms in system identification and digital filter design problems. However, there is more room for research. The most open ground for research is the improvement of the algorithms themselves. The parameters tuning is a big issue in the use of these algorithms and efforts are being made to reduce the number of parameters that determine the effectiveness of the algorithm. Apart from that, the hybrid algorithms leave a lot of room for research in how the hybridization should be carried out. In some cases, the *gbest* particle obtained from PSO is used; where as the whole population is mutated in other cases. The mutation operation is sometime applied to a random member of the *pbest* population where as sometimes on the *gbest* particle itself. These different choices affect the effectiveness of the algorithms differently and no fixed convention has been defined. It is up to the researcher to decide and apply his intuition and experience based on trail and error over a number of trials. Thus exploration of these areas in improving the effectiveness of the algorithms based on the best parameters and best approach to hybridization remains to be a work for future research.

In this thesis, a quantum behaved particle swarm optimization was introduced whose concepts are radical to the classical concept of swarm optimization. However, it was shown that these algorithms are more effective than the classical PSO. So, it is also a ground for future research how new algorithms can be developed by borrowing concepts from different fields of science and applied to improve the existing algorithms. Apart from that, the application of these and other various algorithms in other different kinds of real world applications also remains to be the work for future research.

This research mainly focused on carrying out simulations on the computer using MATLAB. So, its implementation on a dedicated digital signal processor (DSP) on real data can also be looked at in the future. By implementing the digital filters on a DSP with actual data from various sources such as power systems, the ability of the algorithms to actually identify the filter coefficients and design adaptive filters could be tested. On a hardware environment, various other constraints such as memory, storage size, speed of the processor etc. will also come into the effect and hence design of algorithms according to these requirements will pose more challenge to the research.

9.5. SUMMARY

In this chapter, summary of all the chapters was covered. The chapter covered the main motivation of the thesis and briefly summarized how different algorithms are used in two different kinds of optimization problems in the research work. The chapter also concluded that the hybrid algorithms have given better results and also explained the remaining work that can be taken forward for the future research.

BIBLIOGRAPHY

Ababneh, J. I., Bataineh, M. H., Linear phase FIR filter design using particle swarm optimization and genetic algorithms, Digital Signal Processing, vol. 18, July 2008, pp. 657-668.

Chen, S., Mei, T., Luo, M., Yang, X., Identification of Nonlinear System Based on a New Hybrid Gradient-Based PSO Algorithm, International Conference on Information Acquisition, July 2007, pp. 265-268.

Cai, X., Zhang, N., Venayagamoorthy, G. K., Wunsch, D. C., Time series prediction with recurrent neural networks trained by a hybrid PSO-EA algorithm, Neurocomputing, Vol. 70, Issues 13-17, Aug. 2007, pp. 2342-2353.

del Valle Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., Harley, R. G., Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems, in IEEE Transactions on Evolutionary Computation, vol. 12, Apr 2008, pp. 171-195.

Fang, W., Sun, J., Xu, W., Analysis of Adaptive IIR Filter Design Based on Quantum-behaved Particle Swarm Optimization, Sixth World Congress on Intelligent Control and Automation, vol. 1, 2006(a), pp. 3396-3400.

Fang, W., Sun, J., Xu, W., Liu, J., FIR Digital Filters Design Based on Quantum-behaved Particle Swarm Optimization, in Proc. Of the First Intl. Conf. on Innovative Computing, Information and Control, vol. 1, Aug 2006(b), pp. 615-619.

Hao, Z. F., Guo, G. H., Huang, H., A Particle Swarm Optimization Algorithm with Differential Evolution, International Conference on Machine Learning and Cybernetics, vol. 2, Aug. 2007, pp. 1031-1035.

Hongwei, G., Yanchun L., Identification for Nonlinear Systems Based on Particle Swarm Optimization and Recurrent Neural Network [ultrasonic motor control applications], on Proc. International Conference on Communications, Circuits and Systems, vol.2, May 2005.

Karaboga, N., Digital IIR Filter Design Using Differential Evolution Algorithm, EURASIP Journal of Applied Signal Processing 2005:8, 1269-1276.

Karaboga, N., Kalinli, A., Karaboga, D., Designing digital IIR filters using ant colony optimization algorithm, Engineering Applications of Artificial Intelligence 17 (2004), 301-309.

Kristinsson, K., Dumont, G. A., System identification and control using genetic algorithms, IEEE Transactions on Systems, Man and Cybernetics, vol. 22, Sept.-Oct. 1992, pp. 1033-1046.

Krusienski, D. J., Jenkins, W.K., Design and Performance of Adaptive Systems Based on Structured Stochastic Optimization Strategies, IEEE Circuits and Systems Magazine, vol. 5, 2005, pp. 8-20.

Krusienski, D. J., Jenkins, W. K., Particle Swarm Optimization for Adaptive IIR Filter Structures, Congress on Evolutionary Computation, vol. 1, June 2004, pp. 965-970.

Kundur, P., Power System Stability and Control, New York McGraw-Hill, 1993.

Lee, B. H., Kim, S., Seok, J., Won, S., Nonlinear System Identification based on Support Vector Machine using Particle Swarm Optimization, International Joint Conference, SICE-ICASE, Oct. 2006, pp. 5614-5618.

Liu J., Wenbo, X., Sun, J., Nonlinear System Identification of Hammerstien and Wiener Model Using Swarm Intelligence, IEEE International Conference on Information Acquisition, Aug. 2006, pp. 1219-1223.

Luitel B., Venayagamoorthy, G. K., Differential Evolution Particle Swarm Optimization for Digital Filter Design, Proceedings of World Congress on Computational Intelligence, June 2008 (a), pp. 3954-3961.

Luitel B., Venayagamoorthy, G. K., Particle Swarm Optimization with Quantum Infusion for the Design of Digital Filters, Proceedings of Swarm Intelligence Symposium (SIS), Aug. 2008 (b).

Luitel B., Venayagamoorthy, G. K., Particle Swarm Optimization with Quantum Infusion for the Design of Digital Filters, Engineering Applications of Artificial Intelligence (to be submitted for publication), Oct. 2008(c).

Luitel B., Venayagamoorthy, G. K., Particle Swarm Optimization with Quantum Infusion for System Identification, (to be submitted for publication), Oct. 2008(d).

Mikki, S. M., Kishk, A., Quantum Particle Swarm Optimization for Electromagnetics, IEEE Trans. On Antennas and Propagation, vol. 54, Oct. 2006.

Moore, P. W., Venayagamoorthy, G. K., Evolving Digital Circuits using Hybrid Particle Swarm Optimization and Differential Evolution, International Journal of Neural Systems, vol. 16, 2006, pp. 1-15.

Netto, S. L. Diniz, P. S. R., Agathoklis, P., Adaptive IIR Filtering Algorithms for System Identification: A General Framework, IEEE Transactions on Education, vol. 38, Feb 1995, pp. 54-66.

Ng, S. C., Leung, S. H., Chung, C. Y., Luk, A., Lau, W. H., The genetic search approach. A new learning algorithm for adaptive IIR Filtering, IEEE Signal Processing Magazine, Nov. 1996, pp. 38-46.

Panda, G. Mohanty, D., Majhi, B., Sahoo, G., Identification of Nonlinear Systems using Particle Swarm Optimization Technique, IEEE conference on Evolutionary Computation, Sept. 2007, pp. 3253-3257.

Shen, Y., Zeng C., A Self-organizing Particle Swarm Optimization Algorithm and Application, Third International Conference on Natural Computation, vol. 4, Aug 2007, pp. 668-672.

Shynk, J. J., Adaptive IIR Filtering, IEEE ASSP Magazine, April 1989 (a), pp. 4-21.

Shynk, J. J., Adaptive IIR filtering using parallel form realization, IEEE Trans. Acoustic, speech, signal processing, vol. 37, no. 4, April 1989 (b), pp 519-533.

Storn, R., Differential Evolution Design of an IIR Filter, on Proc. IEEE International Conference on Evolutionary Computation, May 1996, pp. 268-273.

Storn, R., Designing Nonstandard Filters with Differential Evolution, Signal Processing Magazine, IEEE, vol. 22, Jan. 2005, pp. 103-106.

Sun, J., Feng, B., Xu, W., Particle Swarm Optimization with Particles having Quantum Behavior, Congress on Evolutionary Computation, vol. 1, June 2004(a), pp. 325-331.

Sun, J., Xu, W., Feng, B., Global search strategy of quantum-behaved particle swarm optimization, IEEE Conf. on Cybernetics and Intelligent Systems, vol. 1, Dec 2004(b), pp. 111-116.

Widrow, B., McCool, J. M., Larimore, M. G., Johnson, C. R. Jr., Stationary and nonstationary learning characteristics of the LMS adaptive filter, on Proc. IEEE, vol. 64, Aug. 1976, pp. 1151-1162.

Zhang, W. J., Xie, X. F., DEPSO: Hybrid Particle Swarm with Differential Evolution Operator, IEEE International Conference on Systems, Man and Cybernetics, vol. 4, Oct. 2003, pp. 3816-3821.

VITA

Bipul Luitel was born on July 10, 1981. He completed his Bachelor's degree in Electronics and Communication from Maulana Azad National Instituted of Technology, Bhopal, India in 2006. He earned his Masters degree in Computer Engineering from Missouri University of Science and Technology, Rolla, MO, USA in May 2009.