

Parallel Multi-Modal Optimal Design and Sensitivity Assessment for Electric Power Systems

By

Ali Yazdanpanah Goharrizi

A thesis submitted to
the Faculty of Graduate Studies
in partial fulfilment of
the requirements for the degree of
Doctor of Philosophy

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba

February 22, 2016

© Copyright

2015, Ali Yazdanpanah Goharrizi

Abstract

This thesis proposes a novel algorithm to optimize multi-modal, nonlinear, black-box objective functions for electric power system design using an electromagnetic transients (EMT) simulator. The algorithm discovers multiple local optimal solutions for a given complex power system, and then generates accurate surrogate models of an objective function around each discovered local optimal solution. These surrogate models represent the local behaviour of the objective function that can be used in the subsequent stages of sensitivity analyses. Using surrogate models instead of intensive transient simulation during sensitivity analysis reduces computational intensity and simulation time. This makes the proposed algorithm particularly suited for optimization of computationally expensive black-box functions. The stages of the algorithm can be implemented independently and hence the computations can be done in parallel. Therefore, the algorithm is implemented in a parallel environment to gain significant speed-up in the design of electric power systems. Comparative studies in terms of objective function evaluation and computation time are provided. Using several multi-modal benchmark objective functions, the superiority of the proposed algorithm compared to other recently developed algorithms is demonstrated. Additionally, the application of the algorithm in the design process of complex electric power system demonstrated through several examples. The case studies show that the parallelized algorithm provides computational savings up to 39 times compared to the conventional sequential approach.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Shaahin Filizadeh, and my co-advisor, Ani Gole, for their encouragements and insightful supervision during each and every stage of this thesis. It would not have been possible to write this dissertation without their continual encouragement, priceless guidance, and constructive feedback.

I would also like to thank R. Singh, J. C. Muller, and R. Jayasinghe for their valuable time, effort, and essential technical support at Manitoba HVDC Research Centre. My special appreciation also goes to D. Muthumuni and F. Mosallat for providing me with great support that helped me to finalize my research.

I would also like to thank Natural Sciences and Engineering Research Council of Canada (NSERC), Manitoba HVDC Research Centre, University of Manitoba and Mitacs program for their financial support during my studies and research.

Last but not least, I am deeply indebted to my beloved parents and siblings for their unconditional dedication and support throughout my both academic and personal life.

Contents

Front Matter

Contents	iv
List of Tables.....	viii
List of Figures.....	x
List of Abbreviations	xiv
1 Introduction	1
1.1 Simulation-Based Design.....	1
1.1.1 Simulation-Based Trial-and-Error Method.....	3
1.1.2 Simulation-Based Multiple-Run Method	4
1.1.3 Principles of Simulation-Based Optimization	6
1.2 Problem Definition	10
1.3 Thesis Objectives	11
1.4 Thesis Organization	13
2 Simulation-Based Optimization Algorithms	15
2.1 Gradient-Based Optimization Methods.....	16
2.2 Direct Search Methods	19
2.2.1 Evolutionary Algorithms	20
2.3 Multi-Modal Optimization Algorithms.....	22
2.3.1 Adaptive Multi-Modal Optimization.....	23

2.3.2	Surrogate Model-Based Multi-Modal Optimization	27
2.4	Conclusions and Discussion	29
3	Description of the Proposed Multi-Modal Optimization Algorithm	32
3.1	Description of the Algorithm.....	33
3.1.1	Initialization.....	37
3.1.2	Localization.....	41
3.1.3	Square Mesh-Grid Generation	43
3.1.4	Objective Function Estimation via Interpolation.....	44
3.1.5	Convergence Check and Final Outcomes.....	46
3.1.6	Density Increment	47
3.1.7	Accuracy Investigation of Local Optimal Solutions and Estimated Objective Functions	48
3.2	Sensitivity Analysis Using Surrogate Models of Objective Function	54
3.2.1	Methodology	54
3.2.2	Implementation of the Proposed Sensitivity Analysis	56
3.3	Numerical Results and Comparative Assessment	62
3.3.1	Comparative Study with the Surrogate-Model Based Algorithm	62
3.3.2	Scalability Assessment	65
3.3.3	Increase in the Number of Minimum Points	65
3.3.4	Increase in the Number of Decision Variables	67
3.4	Conclusions and Discussion	69
3.4.1	Convergence Properties	70
3.4.2	Computational Performance	71
4	Parallel Implementation of the Proposed Multi-Modal Algorithm	73
4.1	Description of the Proposed Parallel Multi-Modal Algorithm.....	74
4.2	Numerical Results and Comparative Assessment	78

4.3	Memory Usage	79
4.4	Conclusions and Discussion	84
5	Power Electronic Design Examples	86
5.1	Design Example 1: Vector-Controlled Induction Motor Drive.....	87
5.1.1	Indirect Vector-Controlled Induction Motor Drive System.....	88
5.1.2	Aggregate Objective Function and Boundaries	90
5.1.3	Sensitivity Assessment.....	98
5.2	Design Example 2: VSC-HVDC Transmission System.....	100
5.2.1	VSC-HVDC Transmission System	101
5.2.2	Aggregate Objective Function and Boundaries	103
5.2.3	Algorithm's Performance	105
5.2.4	Sensitivity Assessment.....	111
5.3	Design Example 3: Three Terminal VSC–HVDC Transmission System.....	112
5.3.1	Three–Terminal VSC-HVDC System.....	113
5.3.2	Aggregate Objective Function and Boundaries	114
5.3.3	Algorithm's Performance	116
5.3.4	Sensitivity Assessment.....	123
5.4	Conclusions and Discussion	125
6	Conclusions, Contributions, and Future Directions	127
6.1	Research Contributions.....	128
6.1.1	Computation Speed-up	128
6.1.2	Memory Usage	129
6.1.3	Sensitivity Assessment.....	130
6.1.4	Applications.....	131
6.2	Suggestions for Future Work	131

6.2.1 Reduction of the Number of Objective Function Evaluations	132
6.2.2 Enhancement of Sensitivity Analysis	132
6.2.3 Enhancement of Parallelism	133
Back Matter	135
Bibliography.....	135

List of Tables

Table 1-1: Comparison of the simulation-based methods.	10
Table 3-1: Summary of the results for the optimization problem in (3-1).	50
Table 3-2: The error in the estimated values for each local area.	51
Table 3-3: One dimensional sensitivity analysis.	59
Table 3-4: 2-dimensional sensitivity analysis.	61
Table 3-5: 2-dimensional sensitivity analysis with continuous variation in detuning.	62
Table 3-6: Comparative study of the surrogate-model based and the proposed algorithm.	63
Table 3-7: Comparative study of the two optimization algorithms.	64
Table 3-8: the parameter values and the associated number of local minima.	66
Table 3-9: Comparative study results as the number of minimum points (n_p) increases. .	66
Table 3-10: Parameter values and the associated number of local minima.	68
Table 3-11: Comparative study results as the number of decision variables (n_v) increases.	69
Table 4-1: Comparative study results.	79
Table 4-2: Required memory as a function of n_v and n_a	80
Table 5-1: Induction machine ratings and parameters.	89
Table 5-2: Coefficient values for the objective functions.	92

Table 5-3: Boundaries of decision variables.....	94
Table 5-4: Local optimal solutions.....	95
Table 5-5: Optimization specifics	97
Table 5-6: HVDC power system parameters.....	102
Table 5-7: Coefficient values for objective functions in (5-5).....	104
Table 5-8: Boundaries of decision variables.....	104
Table 5-9: Optimization specifics	106
Table 5-10: Local optimal solutions for (5-6).....	107
Table 5-11: Coefficient Values for Objective Functions.....	115
Table 5-12: Boundaries of Variables.....	116
Table 5-13: Optimization specifics	119
Table 5-14: Obtained local minima for (5-8).....	120

List of Figures

Figure 1-1: Schematic diagram of simulation-based trial-and-error design.	4
Figure 1-2: Schematic diagram of simulation-based multiple-run design.....	5
Figure 1-3: dc-dc power converter with output voltage controller.	7
Figure 1-4: Output voltage response ($V_{out}(t)$) and reference voltage (V^*) of a dc-dc converter.	7
Figure 1-5: Schematic diagram of simulation-based optimization using a power system simulator.	8
Figure 2-1: Trajectory of movement in a gradient-based optimization method.(a) Starting point X_{in1} and discovered minimum X_{op1} (b) Starting point X_{in2} and discovered minimum X_{op2}	17
Figure 2-2: Trajectory of points in nonlinear Simplex optimization method for the objective function in (2.2); (a) Starting point X_{in1} and discovered minimum X_{op1} (b) Starting point X_{in2} and discovered minimum X_{op2}	20
Figure 2-3: Population scatter plots for a genetic algorithm solution of (2.2); (a) initial population (b) third iteration (c) sixth iteration (d) ninth iteration.....	22
Figure 2-4: Principles of the adaptive multi-modal algorithm: (a) initial mesh generation, (b) objective function evaluation, (c) local areas discovery, (d) addition and evaluation of newly added points in each local area, and (e) accurate local areas discovery.....	26

Figure 2-5: Principle of the surrogate-model based multi-modal algorithm: (a) local area and three sampling points, (b) objective function estimation, (c) new localized area, (d) more accurate objective function estimation.	28
Figure 3-1: Contour plot of Himmelblau’s function and boundary of constraints.....	35
Figure 3-2: Flowchart of the proposed multi-modal optimization algorithm.	36
Figure 3-3: Random sampling of the decision area satisfying non-linear constraints.	39
Figure 3-4: Initial mesh created by the Delaunay triangulation.....	39
Figure 3-5: Equiangular triangle mesh satisfying non-linear constraints.	40
Figure 3-6: Highlighted local areas.	42
Figure 3-7: Square-mesh grids covering local areas.	43
Figure 3-8: Square-mesh covering the 1 st local area.	44
Figure 3-9: Density increment to increase the accuracy of estimated objective function.	48
Figure 3-10: Estimated and actual objective functions for the 1 st -local area. (a) 1 st and (b) 3 rd iteration.	52
Figure 3-11: Estimated and actual functions for the 2 nd -local area. (a) 1 st and (b) 3 rd iteration.	53
Figure 3-12: Sensitivity analysis using second-order derivatives surrogate functions (a) $f_1(x)$ and (b) $f_2(x)$	55
Figure 3-13: OF variations versus detuning for the 1 st - local area. (a) variable x and (b) variable y	57
Figure 3-14: OF variations versus detuning for the 2 nd - local area. (a) variable x and (b) variable y	58

Figure 3-15: OF variation (ΔOF) versus the detuning in the variables. (a) 1 st and (b) 2 nd local area.	60
Figure 3-16: Objective function estimation for each local area.	63
Figure 4-1: Flowchart of the proposed parallel multi-modal optimization algorithm.	75
Figure 4-2: Division of the decision are into three sub-areas in order to reduce the number of sampling points and avoid memory crash.	83
Figure 4-3: Discovered Local areas in each sub-area: (a) I (b) II, (c) III.	84
Figure 5-1: Power electronic drive system including VSC, indirect vector controller and IM.	89
Figure 5-2: Block diagram of the controllers for the drive system.	90
Figure 5-3: Variations in speed and load torque orders.	93
Figure 5-4: Non-optimal dynamic responses. (a) electromagnetic torque (b) shaft speed (c) dc voltage.	96
Figure 5-5: Optimal dynamic responses. (a) electromagnetic torque (b) shaft speed (c) dc voltage.	97
Figure 5-6: Objective function variations (ΔOF) versus detuning in minimum solutions.	99
Figure 5-7: Sensitivity assessment of the minimum solutions.	100
Figure 5-8: VSC-HVDC system.	101
Figure 5-9: Block diagram of VSC1 controllers.	102
Figure 5-10: Block diagram of VSC2 controllers.	102
Figure 5-11 : Step changes in reference parameters.	105
Figure 5-12 : Dynamic response of non-optimized system; (a) VSC1 and (b) VSC2.	109

Figure 5-13 : Dynamic response of optimized controllers; (a) VSC1 and (b) VSC2.....	110
Figure 5-14: Objective function variation (ΔOF) versus detuning in (a) ΔK_{dc} and (b) ΔT_{dc}	112
Figure 5-15: Schematic diagram of the three-terminal VSC-HVDC system.	114
Figure 5-16: Variations in command signals.	117
Figure 5-17 : Optimized and non-optimized dynamic responses of VSC1.	121
Figure 5-18 : Optimized dynamic responses of VSC2.....	122
Figure 5-19 : Optimized dynamic responses of VSC3.....	123
Figure 5-20: Objective function variations (ΔOF) against detuning in minimum solutions.	124
Figure 5-21: Sensitivity assessment of the six local optimal solutions.	125

List of Abbreviations

DC	Direct current
EMT	Electromagnetic transients
EA	Evolutionary algorithm
GA	Genetic algorithm
HVDC	High voltage direct-current
IGBT	Insulated-gate bipolar transistor
IM	Induction machine
ISE	Integral square errors
OF	Objective function
PI	Proportional-integral
PLL	Phase-locked loop
PMR	Parallel multiple run
PWM	Pulse-width modulation
PSO	Particle swarm optimization
RSA	Response surface approximation
SA	Simulated annealing
THD	Total harmonic distortion
VSC	Voltage-source converter

Chapter 1

Introduction

1.1 Simulation-Based Design

Simulation-based design involves the development and use of numerical methods to study and verify the behaviour of complicated systems [1]-[6]. It has now reached a level of maturity that it firmly complements the conventional methods of analytical modeling and experimentation-based observation [7]. There are two main factors that prompt research and development in this area [8]:

- i. development of high-performance computing platforms such as parallel grid computers, which facilitate accurate, comprehensive computer modeling and simulation; and
- ii. increasingly complex engineering systems whose design and analysis demand unprecedented computing resources.

Simulation-based design is used virtually in any situation where a mathematical formulation of acceptable fidelity (i.e., a model that is executable by a computer) can replace the actual system during the design and tuning processes. Simulation-based

design has played an indispensable role in the design process of complex engineering systems such as computational fluid dynamics [9], propulsion system design [10], large-scale wireless sensor networks [11], and hybrid vehicles [12].

Due to its complexity, size, and strategic nature, which prohibit direct field experimentation at the early stages of design, the field of electric power systems has been an early adopter of the simulation-based approach to design and study emerging systems, devices, and technologies [13]-[16][15].

There are many different types of power system simulation tools; however, the main focus of this work is on the electromagnetic transient (EMT) simulation, in which highly detailed component models are used to represent the short-term, fast transient behaviour of a power system [16], [17]. Specifically, this research uses PSCAD[®]/EMTDC[™], which is a widely used EMT simulator. This simulator provides a high degree of accuracy, and simulation results that are in close conformity of the response of a real-world power system [18]. The design and decision-support algorithms developed in this thesis are not specifically designed for EMTDC and can be used along with any other simulator. EMTDC is merely a means for implementation in this research.

For large-scale power systems with nonlinear and time-variant devices, such as power electronic converters and complex control systems, an explicit mathematical formulation of the objective function (OF) is often not attainable [19]; thus the design procedure primarily relies on several rounds of simulation by varying the parameters within their limits [20]. Observing the results of simulations, the most suited set of parameters can then be selected by an expert. Commonly used approaches in power system design using

simulations can be divided into three subcategories namely (i) trial-and-error, (ii) pre-specified multiple-run, and (iii) optimization-based approaches [20].

1.1.1 Simulation–Based Trial–and–Error Method

Figure 1-1 shows a trial-and-error design process for a generic electric power system. The aim of the design is to find the right values for a set of decision variables $\mathbf{x}=\{x_1, x_2, \dots, x_n\}$, so as to meet the design objectives. The design objectives are expressed in terms of properties that are desired, which are then exhibited in the final results. For example, proper tuning of a controller or selection of suitable values for the components of a filter can be considered as objectives. The power system simulator contains and executes a model of the power system wherein the set of decision variables (\mathbf{x}) can be varied by the user. The trial-and-error design process proceeds as follows:

- i. The user (expert) makes an initial guess for the decision variables;
- ii. The decision variables are fed into the simulator where a model of power system is provided;
- iii. Once the simulation is completed, the expert judges the objective values and validates the results; if the objectives are not satisfied for the deployed set of variables, modifications and adjustments are applied to the decision variables in order to improve the results. These modifications are based upon the expert's engineering judgment and experience;
- iv. The simulation is run again using the modified vector of decision variables and the results are validated again.

This process will continue until a set of decision variables are obtained that satisfy the objectives.

The expert's observations and decisions that are made to evaluate the results and adjust the variables are the key elements to ensure that the trial-and-error method is successful; the approach as a whole remains viable because of its reliance on the refined adjustments made by the expert [21], [22]. However, it is possible to replace the human expert's directions during the design process with algorithms that automatically select the subsequent set of decision variables and conduct simulations in order to accelerate the process [19], [21]. These methods are explained next.

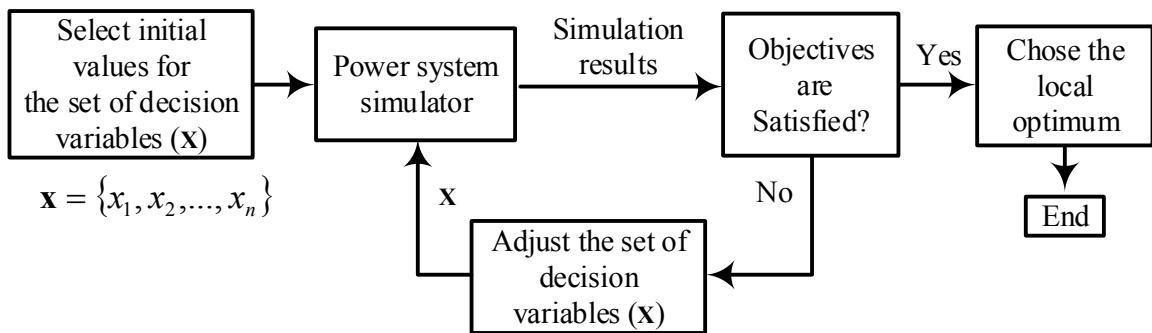


Figure 1-1: Schematic diagram of simulation-based trial-and-error design.

1.1.2 Simulation-Based Multiple-Run Method

To make the simulation-based design process more computationally efficient, multiple-run methods are often utilized [23], [24]. In these methods, several simulations of a case study are conducted for a large number of sequentially or randomly varied values of the decision variables, as shown schematically in Figure 1-2. Once multiple-run simulations

are completed, the results of all the simulations are gathered and compared to the design objectives in order to discern the most suited set of variables. Variable values that provide the highest conformity to the specifications are selected as the final result. The final selection of the variables relies on the judgment of an expert, which may be aided by compute-based analysis of the system's response in the conducted simulations [21]. This method is considered a non-adaptive approach, since a pre-defined set of variables (sequential or random) is used during the process of design [25], [26]. In other words, the sequence of simulations will have a pre-determined path based on the initially selected variable values.

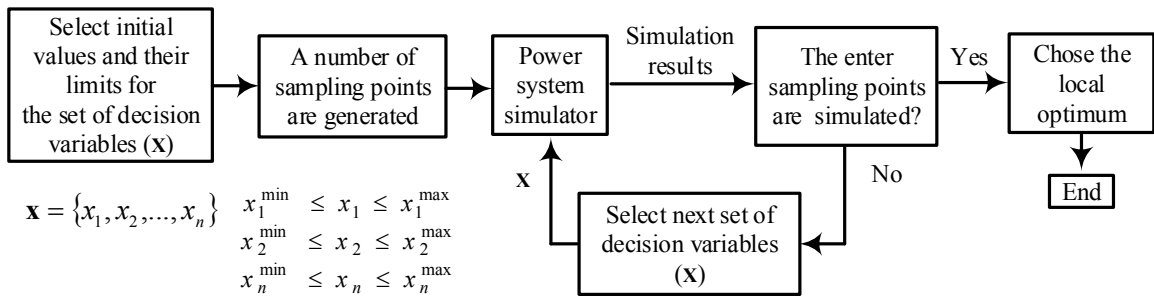


Figure 1-2: Schematic diagram of simulation-based multiple-run design.

As an example, consider a simulation consisting of three variables; with a granularity of 10 steps for each variable (within its boundaries), about 17 hours of computing time will be required to conduct all the 1000 simulations necessary to evaluate each candidate set of variable values, if each run takes one minute to simulate. If the number of design variables is increased to 6 the design will require almost 694 days to be completed. Therefore, with a small increase in the number of decision variables, the number of

required simulations will significantly increase [27]. Such simulations take unreasonably long periods to complete [28].

1.1.3 Principles of Simulation–Based Optimization

The design of a power system can be enhanced by formulating the design objectives as an optimization problem, in which the norm of the difference between the actual performance of the system and the desired one(s) is to be minimized without violating any design constraint(s) [23]. Hence, the optimization problem can be expressed in a generic form as follows:

$$\begin{aligned}
 & \text{minimize } \|\text{actual performance}(\mathbf{x}) - \text{desired performance}\| \\
 & \text{where } \mathbf{x} = \{x_1, x_2, \dots, x_n\} \\
 & \text{subject to constraint(s)}
 \end{aligned} \tag{1-1}$$

where $\|\ \|$ is the norm, \mathbf{x} is a vector of decision variables, and $\|\text{undesired performance}(\mathbf{x}) - \text{desired performance}\|$ is the objective function.

A well-defined objective function increases the chance of finding the desired solution(s) for the optimization problem. For example as it is shown in Figure 1-3 for a power electronic dc-dc converter the output voltage ($V_{out}(t)$) of the converter can be considered as the performance of the system and the proportional–integral (PI) controller gains (i.e., $x_1=K_p$, $x_2=K_i$) as the decision variables. The desired performance requires that the settling time and overshoot of the output voltage in response to a step change in the input be minimized. Figure 1-4 shows the settling time (t_s) and overshoot (V_M) of the

output voltage ($V_{out}(t)$) of the dc-dc converter and the step change in the reference voltage (V^*).

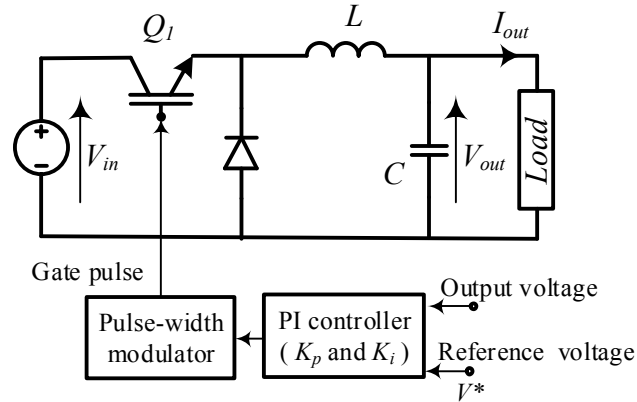


Figure 1-3: dc-dc power converter with output voltage controller.

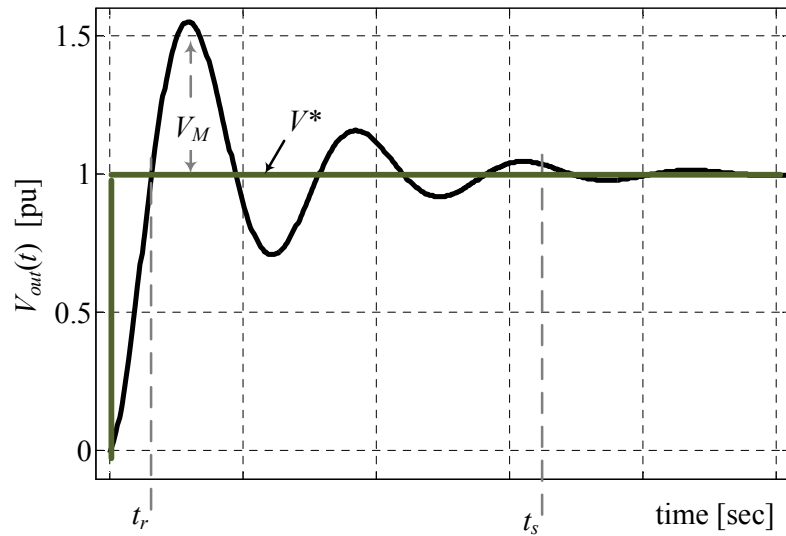


Figure 1-4: Output voltage response ($V_{out}(t)$) and reference voltage (V^*) of a dc-dc converter.

An optimization algorithm can be employed to find a set of decision variables (K_p and K_i) that change the dynamic performance of the dc-dc converter to minimize both the settling time (t_s) and the overshoot (V_M). The simulation-based optimization design setup

is shown schematically in Figure 1-5, where an optimization algorithm provides the power system simulator with judiciously selected values for the decision variables (\mathbf{x}).

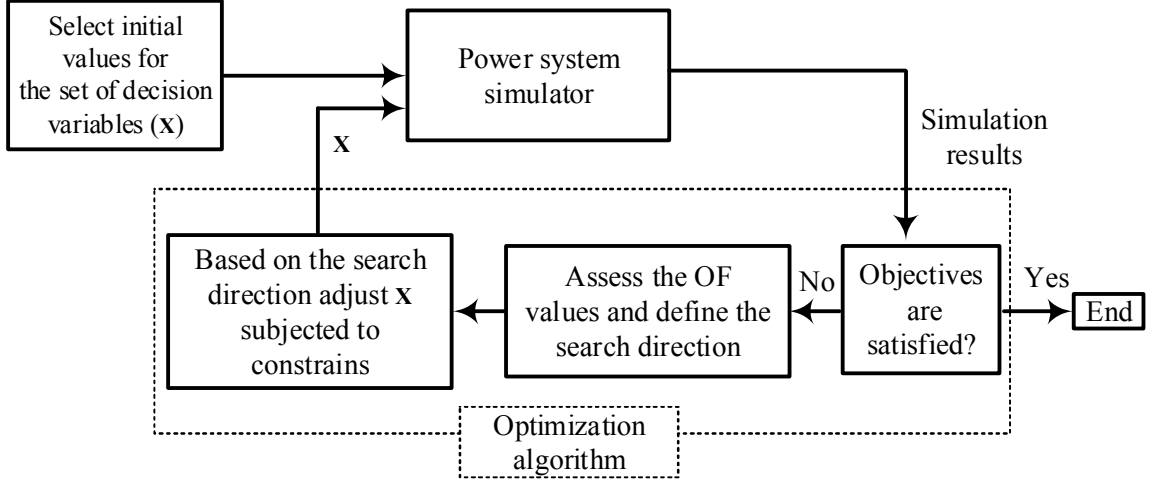


Figure 1-5: Schematic diagram of simulation-based optimization using a power system simulator.

The objective function (OF) for this example can be expressed in terms of an Integral Square Error (ISE) function as shown below.

$$\text{OF}(\mathbf{x}) = w_1 \int_0^{t_r} (V_{out}(t) - V^*)^2 + w_2 \int_{t_r}^{t_s} (V_{out}(t) - V^*)^2 + w_3 \int_{t_s}^{\infty} (V_{out}(t) - V^*)^2 \quad (1-2)$$

The objective function contains three sub-objective functions each associated with a different periods, i.e., $[0, t_r]$, $[t_r, t_s]$ and $[t_s, \infty]$ [22]. Each sub-objective takes a weight coefficient (i.e., w_1 , w_2 and w_3). In a well-defined objective function, all sub-objective functions should have comparable numerical values to avoid domination of one sub-objective over the others [21], [23]. In any of the regions indicated in (1-2), the closer the response $V_{out}(t)$ is to the reference (V^*), the smaller the value of the OF. Therefore, the algorithm aims to obtain optimal decision variables (\mathbf{x}^*) by minimizing the OF without violating the constraint(s). Here in this example the constraints can be defined as follows:

$$\begin{aligned} K_{iMax} &\geq K_i \geq K_{iMin} \\ K_{pMax} &\geq K_p \geq K_{pMin} \end{aligned} \quad (1-3)$$

where K_{iMax} and K_{iMin} are the maximum and minimum values for the K_i , and K_{pMax} and K_{pMin} are the maximum and minimum values for the K_p , respectively.

For a given set of decision variables ($\mathbf{x} = (K_p, K_i)$) a simulation run will be conducted, and at the end of the simulation; the objective function value ($OF(\mathbf{x})$) is calculated using (1-2). The algorithm calculates the direction of its search based on the updated value for the objective function. Obtained from the search direction, a new set of decision variables $\mathbf{x}_1 = (K_{p1}, K_{i1})$ is provided by the optimization algorithm without violating the constraint(s) in (1-3). For the new set of decision variables the simulation runs again and based on the output voltage of the dc-dc converter a new value for objective function can be calculated using (1-2). Then based on the new value of the objective function a new decision for the search direction can be made. Based on this search direction a new set of decision variables ($\mathbf{x}_2 = (K_{p2}, K_{i2})$) is calculated. Doing so, the optimization algorithm evaluates the objective function value and updates the search direction. This procedure continues until successive evaluations of the objective function are essentially unchanged, indicating convergence to at least a local minimum.

As long as the optimization algorithm controls the search direction effectively, fewer simulations are required to find a local optimal solution; at the same time, more accurate results are obtainable in comparison to the previous search approaches [28].

The properties, merits, and disadvantages of the trial-and-error, multiple-run and simulation-based optimal design approaches are summarized in Table 1-1.

Table 1-1: Comparison of the simulation-based methods.

Method	Definition	Advantages	Disadvantages
Trial-and-error	Expert insight and judgement is employed to obtain the local optimal solution	- Simple to implement	- Requires expert knowledge - If the number of parameters increase slightly, then the design process will be impractical - Solutions are not necessarily optimal
Multiple-run	Systematically or randomly varying the design parameters by means of multi-run simulations to find the local optimal solution	- Limits expert's supervision during design process - Simple to implement	- If the number of parameters increase slightly then the simulation time will increase greatly - Solutions are not necessarily optimal
Optimization	Design problem defined as an optimization problem; parameters are varied by the optimization algorithm to find the local optimal solution	- Minimizes expert's supervision during design process - Comparatively requires fewer simulation runs and less simulation time - Robust and reliable - Solutions are optimal	- Requires expert knowledge to set the objectives - Final results highly affected by the starting point - Parallelization may be restricted or impossible

1.2 Problem Definition

Despite their success in the optimal design of power systems, conventional simulation-based optimization and decision-support face difficulties when it comes to real-world implementation of complex power systems. It has been reported that unavoidable tolerances in the physical parameters are expected [29][31]. To overcome this problem multiple local optimal solutions for a design may be obtained during the optimization process. With the knowledge of other potential local optimal solutions, the sensitivity of each local optimum to the inevitable tolerances can be examined [33]. Therefore, among all local optima, the one that is least sensitive to these variations is more acceptable [32].

Most algorithms that are used in the context of simulation-based power system design are merely able to find a local or the global solution [34], [35] and less attention is given to multi-modal optimization algorithms [36], [37]. Recently, an adaptive multi-modal optimization algorithm was proposed to obtain all local optima as well as the global one for complex black-box electrical systems [8], [27]. The algorithm has been improved in several aspects that it can obtain the surrogate models of the objective function around its local optima to perform sensitivity assessments [28]. While it is beneficial to develop local surrogate models for a multi-modal objective function, the method used in [28] still tends to be computationally demanding as it relies on an inner-loop curve fitting technique to adjust the parameters of the explicit surrogate models. Additionally, the number of required objective function evaluations tends to be relatively high, particularly when the number of decision variables is larger than 5. Also sensitivity assessment can be improved and simplified to avoid second-derivative computation of the objective function at the discovered local optimum [7], [28].

Additionally, implementation of multi-modal optimization algorithm has not been conducted in parallel for design of complex black-box power systems with a large number of variables.

1.3 Thesis Objectives

Based on the deficiencies of the conventional simulation-based optimization methods that were described in the previous sections the following potential improvements are identified to be addressed in this thesis.

- i. **Investigation of multi-modality:** It is important to discern whether the optimal solution obtained is either a local optimum or the global optimum of a system at hand. In the conventional design methodology, the globally optimal solution is mainly sought; however, local areas may be preferred to the global optimum, particularly if other objectives such as sensitivity to parameter variations considered.
- ii. **Enhanced computational efficiency via surrogate modeling:** Estimation of the objective function in the local areas brings two-fold benefits. Firstly the surrogate model helps to predict the accurate position of the local optimum and hence speeds up the optimization process, reducing objective function evaluation. Secondly sensitivity assessment can be done with no simulation runs. This simplifies the design process and garners insightful information about the behaviour of the objective function.
- iii. **Improved sensitivity analysis:** An optimization algorithm may be able to satisfy the initial steps of a design by finding the optimal solution for the objectives. Other important assessments such as sensitivity analysis of the local optimal solutions, however, are done as independent studies. Such studies can be fulfilled during the optimization process to avoid extra simulations. This thesis aims to develop an optimization algorithm that is able to obtain local optimal solutions and provides supportive information to conduct sensitivity assessments of them.
- iv. **Investigation of the algorithm's performance in solving complex power system designs:** Application of the proposed multi-modal optimization algorithm

in the design of complex power systems will be investigated in this thesis. The proposed multi-modal optimization algorithm will be integrated with an EMT simulation program to form a general-purpose design platform, which unifies the setting-up, testing, and execution of optimization design for electrical power systems.

v. **Deployment of parallel computing for improved computational performance:**

This thesis aims to develop a parallel version of the proposed multi-modal optimization algorithm. The algorithm increases the efficiency of the design for electric power systems because its computations can be executed independently on a highly parallelized environment. Moreover, comparative studies of computation time are conducted against the sequential version of the proposed algorithm.

1.4 Thesis Organization

Following the introduction presented in this chapter, Chapter 2 describes different optimization algorithms commonly used in the context of simulation-based optimization. A multi-modal optimization algorithm is proposed in Chapter 3 and its stages are described in detail. The superiority of the proposed algorithm compared to widely used multi-modal optimizer [48] in terms of computational intensity is conducted using several well-known multi-modal objective function benchmarks.

In Chapter 4, a parallel version of the algorithm is introduced and it is implemented on a 60-core parallel computing platform; consequently, significant speed-up over a

sequential algorithm is achieved. A technique is proposed to reduce memory usage of the algorithm during optimal design of complex black-box systems where a large number of decision variables are required to be optimized.

Both the sequential and parallel versions of the algorithm will be used in Chapter 5 to illustrate their functionality in solving black-box electric power system design examples. It will show that the use of parallelism and memory reduction techniques described in Chapter 4 enable optimization of a problem with 12 variables within a reasonable timeframe and with acceptable memory usage. The developed surrogate models for sensitivity analysis of the optimized controllers are shown to remove the need for extra simulations and to save significant time.

Chapter 6 will present the contributions that are made by this research. It will also provide recommendations for future research, particularly the immediate extensions of this thesis.

Chapter 2

Simulation–Based Optimization

Algorithms

In this chapter a wide range of optimization algorithms are presented to provide insight into their underlying principles and their role in simulation-based optimization. An optimization algorithm conducts two key tasks during simulation-based optimization design, namely (i) updating the decision variables based on a selected search strategy, and (ii) steering a set of simulations to obtain the objective function (OF) value for the updated variables [16]. The search strategy is a differentiating feature between optimization algorithms and can influence factors such as final solutions (local, global, or both), computational intensity, and memory usage of the algorithm. These are the criteria that will be considered to evaluate each algorithm.

In optimization of power systems it is beneficial to obtain as many local minima (including the global minimum) as possible to provide a wide range of choices for the

designer and insight into system behaviour. As it was stated in Section 1.2 identification of multiple local optimal solutions and evaluating solution sensitivity are important in the design of complex power systems; therefore multi-modality and sensitivity analysis will also be taken into account in assessment of candidate optimization algorithms.

It should be mentioned that development of a multi-modal algorithm for black-box optimization must satisfy two major requirements:

- i. the algorithm must be able to search the decision area to find all (or as many as possible) local optima; and
- ii. the algorithm must have low computational intensity in terms of both algorithmic complexity, and the number of objective function evaluations.

2.1 Gradient-Based Optimization Methods

Gradient-based optimization methods can find a local minimum of a differentiable function by taking steps in the opposite direction of the function's gradient. Starting from an initial set of decision variables and the current function's gradient vector, it defines the step size and the search direction toward a local optimal solution. Mathematically, gradient-based methods can be defined as follows:

$$\mathbf{x}_{m+1} = \mathbf{x}_m + \beta_m \boldsymbol{\psi}(\mathbf{x}_m) \quad (2-1)$$

where \mathbf{x}_m , \mathbf{x}_{m+1} are the current (i.e., m^{th}) and new (i.e., $(m+1)^{th}$) values for the decision variables, respectively. Moreover $\boldsymbol{\psi}(\mathbf{x}_m)$ is the search vector in the decision space and β_m is the step length in that direction.

The search direction $\psi(\mathbf{x}_m)$ is determined using gradient information (i.e., partial derivatives of the objective function with respect to individual variables) at the current point \mathbf{x}_m . The performance of this method is demonstrated in the Figure 2-1 to show the zigzag-like search path that is created by the gradient descent that is applied to the optimization problem in (2-2). The current step length, β_m , should be selected such that the value of the objective function at the next step becomes smaller than the current step, i.e., $f(\mathbf{x}_{m+1}) < f(\mathbf{x}_m)$ [21]. Different methods may be used to find a suitable value for the current step length. The best step length is the one that gives the smallest value for the objective function at the next step $f(\mathbf{x}_{m+1})$ [21].

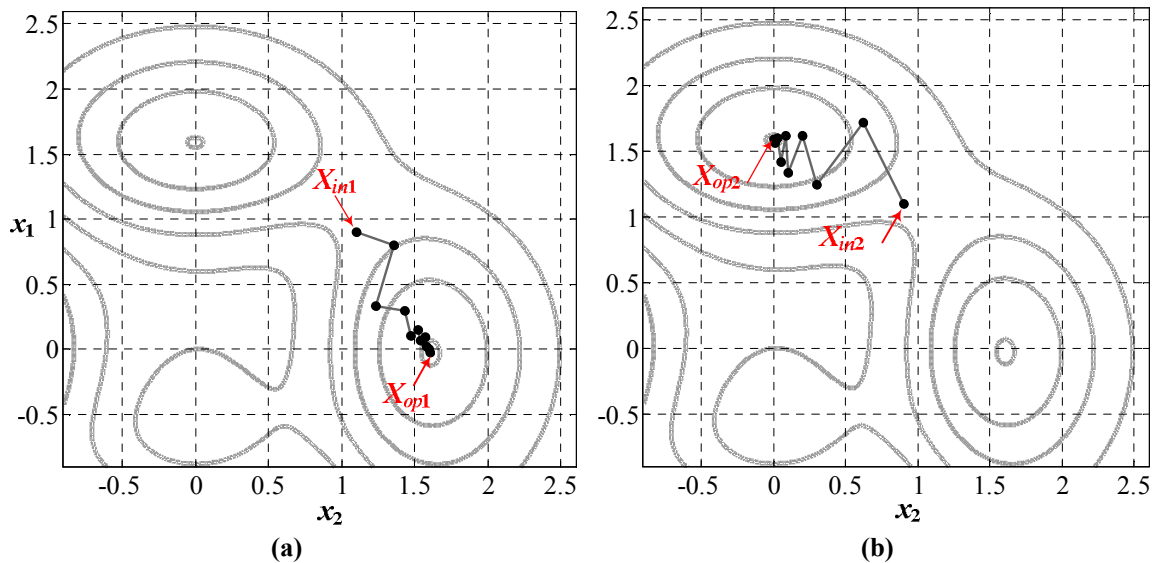


Figure 2-1: Trajectory of movement in a gradient-based optimization method. (a) Starting point X_{in1} and discovered minimum X_{op1} (b) Starting point X_{in2} and discovered minimum X_{op2} .

$$\begin{aligned}
\text{minimize } f(x_1, x_2) &= -3(1 - x_1^2)^2 \exp(-x_1^2 - (x_2 + 1)^2) \\
&+ 10\left(\frac{x_1}{5} - x_1^5 - x_2^5\right) \exp(-x_1^2 - x_2^2) \\
&+ 3 \exp(-(x_1^2 + 1)^2 - x_2^2) \\
\text{subject to } & -0.9 \leq x_i \leq 2.5, \quad i = 1, 2.
\end{aligned} \tag{2-2}$$

Obviously, the solution of an optimization problem using a gradient-based method is highly affected by the starting point. In the above example, if the algorithm starts from $X_{in1}=(1.1, 0.9)$ it possibly converges to $X_{op1}=(1.6, -0.02)$ and if it starts from $X_{in2}=(0.9, 1.1)$ it may converge to $X_{op2}=(-0.02, 1.6)$. The actual step length taken towards the minimum shrinks gradually as the optimization progresses and the derivatives of the objective function become increasingly small in close proximity to a local optimum. This indicates increasingly lower search speed at final stages of optimization [38]. To overcome this deficiency solutions such as use of second order derivatives was proposed [39].

In complex systems, explicit derivatives are hard to obtain for an objective function that is available in a black-box form and its value at each point should be obtained using numerical simulation [7]. Due to the absence of an explicit objective function and the gradient vector, this requires a large number of expensive simulation runs, which is time-consuming and adversely impacts the optimization speed [38].

Gradient-based optimization algorithms are parallelizable because the computation of the gradient vector elements can be done independently for each variable. Examples of gradient-based methods are Cauchy [38], Marquardt [39] and Newton's method [40].

2.2 Direct Search Methods

If an explicit formulation of the objective function is not attainable, it is difficult to calculate the gradient of the objective function during the optimization process. In contrast to gradient-based methods, direct search methods require less information about the objective function, particularly its gradient at each iteration [41]-[45]. Therefore these methods find broad applications in optimization-based design [24], [27]. Methods such as Hooke–Jeeves [41], Powell’s conjugate directions [42], [43] and Nelder-Mead’s nonlinear Simplex [44], [45] are categorized under this class.

The Nelder-Mead’s nonlinear Simplex search movements are shown in the Figure 2-3 for the optimization problem in (2-2). Note that similar to gradient-based approaches this method also relies on a starting point and hence the final convergence can be different depending on the starting point. The Simplex method and other typical direct search methods are local optimizers.

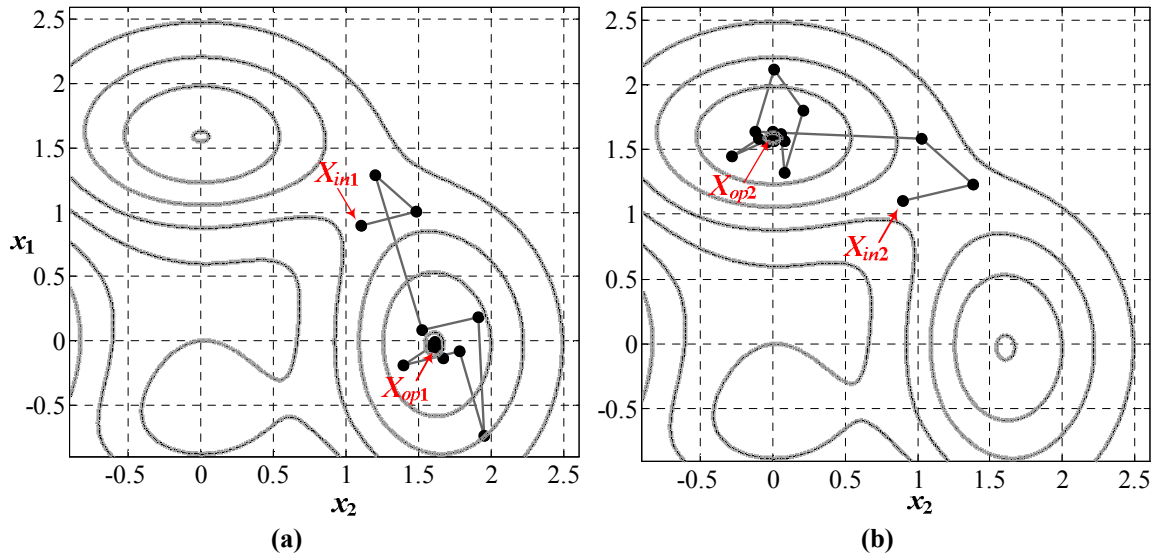


Figure 2-2: Trajectory of points in nonlinear Simplex optimization method for the objective function in (2.2); (a) Starting point X_{in1} and discovered minimum X_{op1} (b) Starting point X_{in2} and discovered minimum X_{op2} .

2.2.1 Evolutionary Algorithms

Evolutionary algorithms (EA) are principally inspired by some form of biological evolution and solve optimization problems based on a natural selection process [46], [47]. Examples of EA methods include genetic algorithms (GA) [48]-[50], particle swarm optimization (PSO) [47], simulated annealing (SA) [51], [52], and ant colony (AC) [53].

Unlike gradient-based and direct search methods these algorithms may converge to the global optimum. As a result of inherent random selections these methods hold a high probability of global convergence [50], [51]. However, these methods, compared to gradient-based and direct search methods, have more complexity, which makes their implementation challenging. They also require a much larger number of objective function evaluations during optimization [50].

One of the most popular EAs is the genetic algorithm (GA) that is able to obtain the global minimum [50]. Starting from an initial population, this algorithm iteratively

modifies a population of individual solutions (chromosomes) and proceeds to exploit the elitist features of them in order to reproduce the next generation of offspring. Doing so, the population evolves toward the global solution [49], [50].

Recently a multi-modal genetic algorithm has been proposed that can obtain multiple local optima by forming a bi-objective function [48]. This approach adds a second objective to the original multi-modal objective function to form a bi-objective formulation [48]. The new bi-objective optimization problem can be solved using a multi-objective algorithm that reveals multiple local optima solutions. Figure 2-3 demonstrates the principles of the multi-modal GA to solve the optimization problem defined in (2-2). A chromosome in the algorithm consists of a pair of genes as the optimization problem presented in (2-2) possesses two decision variables. Figure 2-3 (a) shows how the initial population is scattered randomly over the decision area while in Figure 2-3 (b) it is shown that the algorithm has generated more points (in the third iteration) closer to the vicinity of each minimum. In the sixth iteration in Figure 2-3 (c) and the ninth iteration in Figure 2-3(d) it is shown that the algorithm correctly indicates the vicinity of the two minima and evolves the population to approximate the minimum points.

This multi-modal genetic algorithm requires a large number of objective function evaluations for optimization problems with a larger number of variables and local minima.

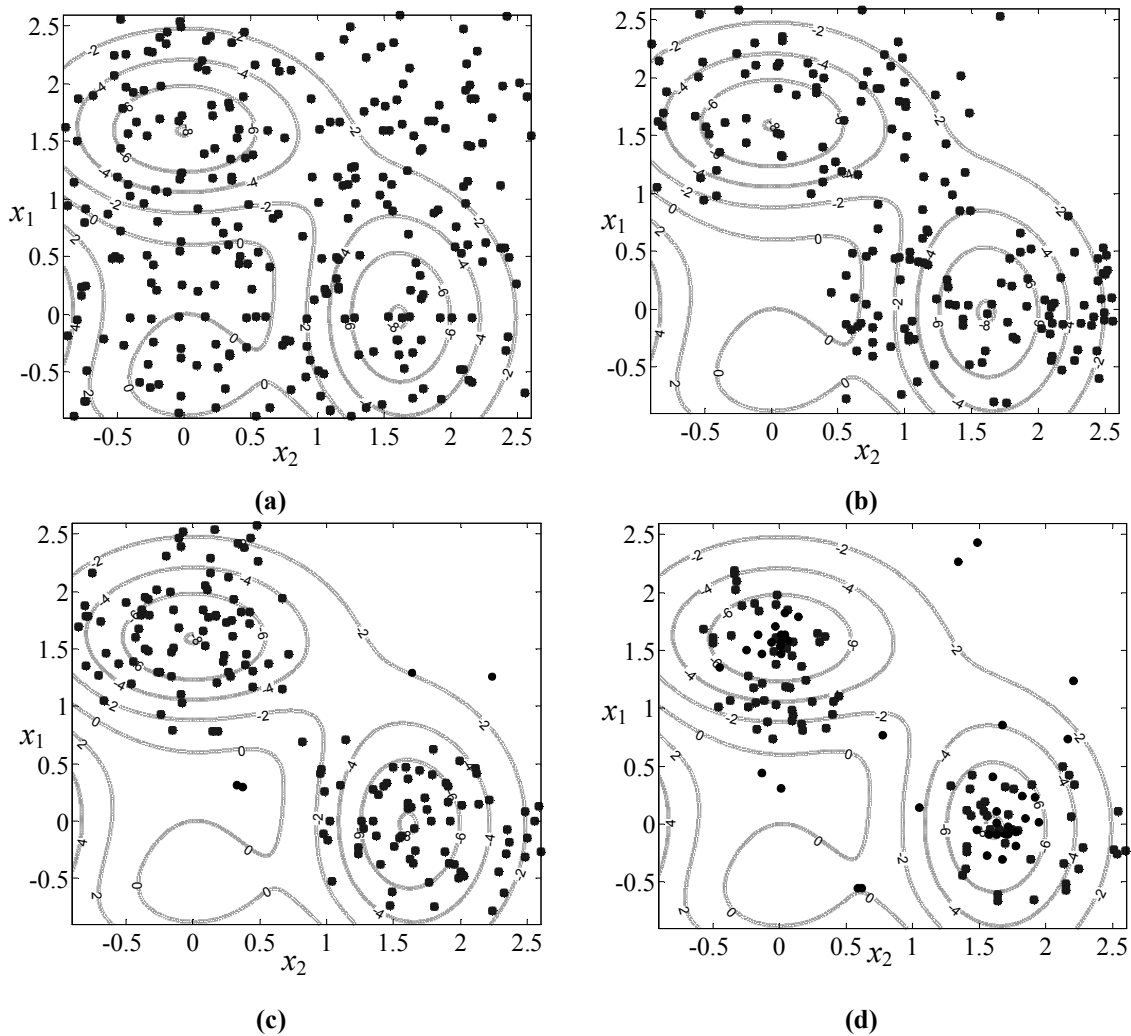


Figure 2-3: Population scatter plots for a genetic algorithm solution of (2.2); (a) initial population (b) third iteration (c) sixth iteration (d) ninth iteration.

2.3 Multi-Modal Optimization Algorithms

Another class of optimization algorithms exists, which are capable of searching for multiple local (and global) optima simultaneously [47]. These so-called multi-modal algorithms are particularly beneficial from a practical point-of-view [47], [48]. There are

several reasons why identification of multiple locally optimal solutions is important in the design of complex systems; these include:

- i. Tolerances may exist in the physical implementation of the optimized system. For example optimal values for circuit components may not be accurately achievable or may deviate from nominal values with temperature or other operating conditions. These tolerances will cause perturbations and some degree of deterioration in the performance of the optimized solutions [47], [31]. Therefore, in selection of a local optimal solution it is important to consider such sensitivities.
- ii. Although the global optimum provides the most desirable solution, its actual implementation maybe sometimes proven to be overly expensive or practically infeasible, due to for example physical limitation of the attainable components or cost implications thereof [29],[30] and [31].

Novel algorithms have been proposed to solve a multi-modal optimization problem that can obtain multiple local solutions with a relatively small number of objective function evaluations [27], [28]. These approaches will be demonstrated in the upcoming sections to provide insight into their principles as they are the backbone of the proposed algorithm presented in Chapter 3.

2.3.1 Adaptive Multi-Modal Optimization

Recently, an adaptive algorithm has been proposed to handle multi-modal optimization problems that reduce the number of objective function evaluations [27]. A brief

description of this algorithm's framework is given in Figure 2-4 , where an arbitrary objective function ($f(x)$) with two local minima is plotted. It is also assumed that the objective function can be obtained by means of costly simulations where a complex system under study is modeled. Therefore the objective function $f(x)$ and hence its local areas are not explicitly known.

Step 1: Generation of the initial points. Figure 2-4(a) shows an initial set of equidistant points placed between x_0 and x_8 (the span of the decision area). The points can be generated randomly or using different methods such as mesh generation techniques [54]. But for simplicity equidistant points are selected in this example. The selection of sampling points is important because it affects the next stages of the algorithm and consequently its final results.

Step 2: Evaluation of the objective function. Figure 2-4(b) shows that the objective function value for each point is obtained by means of a costly simulation that contains a high-fidelity model of a complex system under optimization. As an expensive simulation run is required to obtain the objective function value for each sample point, this stage of the algorithm may take a long time to be completed (depending on the number of sampling points and simulation time). Therefore, decreasing the number of objective function evaluations, or in other words the number of sampling points, leads to reduction of the entire computation time of the algorithm.

Step 3: Localization. The purpose of this stage is to determine areas where local minima are likely to be found. The value of the objective function at each point is compared to its neighboring points. If a point has a smaller objective function value

compared to its nearby points, it will be selected as an intermediate minimum. This point and the nearby points are selected as a local area. Figure 2-4 (c) shows two localized areas.

Step 4: Local search. The algorithm generates new points to increase the density of the sampling points in the localized areas in order to obtain a more accurate estimation of the objective function. In this case, x_{12} and x_{23} for the first local area and x_{56} and x_{67} for the second one are added and shown in Figure 2-4 (c). The algorithm then proceeds to the second step and evaluates the objective function values for the newly added points.

Assuming the algorithm does not converge in Figure 2-4 (c), the algorithm continues to the second iteration and the results are shown in Figure 2-4 (d) to obtain higher resolution sampling points. Figure 2-4 (e) shows the new local areas with better estimations of the local minimum. By repeating these stages, accurate estimates of the multiple local optima are achieved.

This algorithm can be easily implemented in parallel, as the processing of each local area can be done independently on a different processor. This algorithm does not address sensitivity analysis to obtain the least sensitive local optimum which shows the most robustness against perturbations in optimized variables [8].

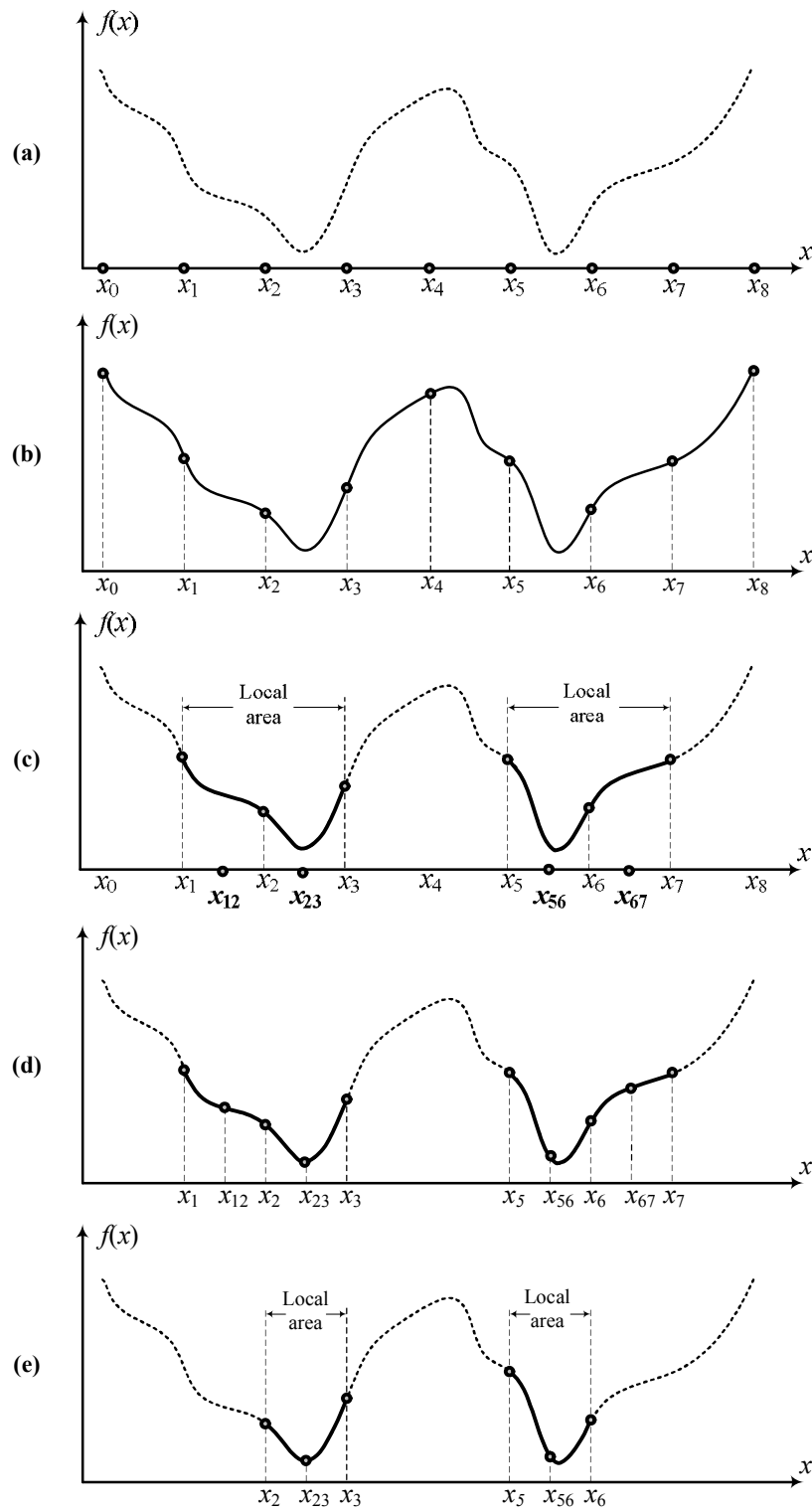


Figure 2-4: Principles of the adaptive multi-modal algorithm: (a) initial mesh generation, (b) objective function evaluation, (c) local areas discovery, (d) addition and evaluation of newly added points in each local area, and (e) accurate local areas discovery.

2.3.2 Surrogate Model-Based Multi-Modal Optimization

This algorithm was developed as a result of major enhancements to the “adaptive multi-resolution optimization algorithm” [28]. The number of objective function evaluations is significantly reduced and an explicit surrogate model of the local behavior of the objective function around each minimum is estimated. The role of surrogate models is twofold:

- i. Firstly, they are used to determine approximate locations of the local minima so that additional trial points can be added to the existing mesh to progressively increase its accuracy of estimating the actual local optima; and
- ii. Secondly, when convergence occurs, the surrogate models can be used for sensitivity assessment of the local optima as they represent the local behavior of the objective function with a high level of accuracy.

A brief description of the algorithm is described in Figure 2-5. The first three steps of the algorithm are identical to the first three steps of the “adaptive multi-modal algorithm” shown in Figure 2-4 (a), (b) and (c). Once the algorithm discovers the local areas (shown in Figure 2-4 (c)), the process of finding an accurate minimum solution can be explained by considering the following stages shown in Figure 2-5.

Figure 2-5 (a) shows the three sample points in the local area. Using these sample points, a surrogate model can be made to represent the objective function. Figure 2-5 (b) shows the surrogate model ($f_s(x)$) that is selected as a second-order polynomial function. The minimum of the surrogate function can be obtained analytically as it is a mathematically explicit function. The position of this minimum point of the surrogate

model (see Figure 2-5) indicates the position of the new point in the local area. Each stage evaluates only one point, which results in a considerable reduction in objective function evaluations when compared to the “adaptive multi-modal algorithm” [27]. The newly added point and the new local area with four sample points are shown in Figure 2-5 (c). Based on these four sample points, a new surrogate model can be developed to obtain a more accurate estimate of the objective function, and therefore, a more accurate estimate of the local optimal solution. By continuously repeating the steps shown in Figure 2-5, the algorithm discovers the local optima and their surrogate models that can be used for sensitivity analysis.

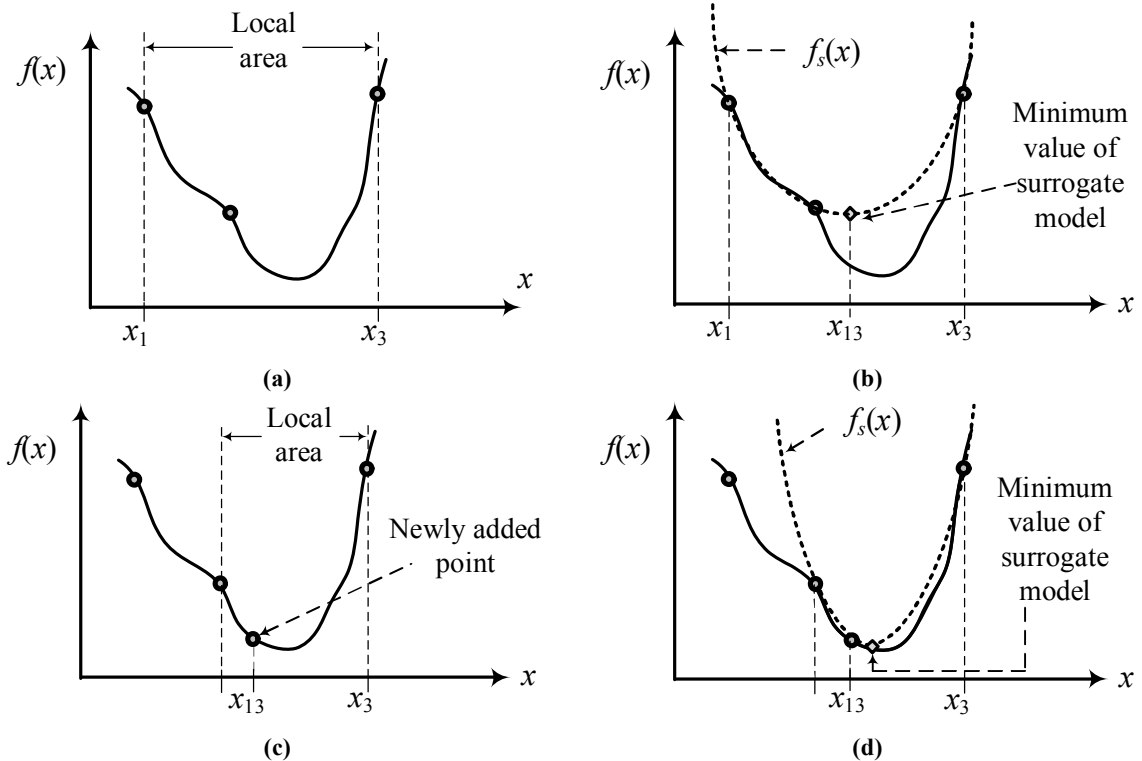


Figure 2-5: Principle of the surrogate-model based multi-modal algorithm: (a) local area and three sampling points, (b) objective function estimation, (c) new localized area, (d) more accurate objective function estimation.

The surrogate models of the objective function around each minimum are made to fulfill such requirements as follows:

- i. to determine approximate locations of the local optima so that the required objective function evaluations are reduced in each iteration compared to the previous method; and
- ii. when convergence occurs, the surrogate models are generated to represent the local behaviour of the objective function with a high level of accuracy.

While it is beneficial to develop local surrogate functions for a multi-modal objective function, this algorithm tends to be computationally demanding as it relies on an inner loop curve fitting technique to adjust the parameters of the explicit surrogate models. During the curve fitting process more emphasis has been given to the points with lower OF values, so that the surrogate model finely matches the actual objective function at the minimum point while it may have some mismatch at the others [28]. This affects the accuracy of sensitivity analysis that is conducted based on these surrogate models. Also a curve fitting computations are required for each iteration [28], which increase the computational intensity of the algorithm.

2.4 Conclusions and Discussion

As discussed earlier local search algorithms such as direct search and gradient-based methods are often trapped in a local optimum. One solution to this problem is to restart the algorithm when no progress is observed [37]. Therefore, the local search algorithm

can be implemented to start multiple instances. This is called a multi-start strategy that requires numerous expensive simulations run for majority of objective functions [38].

Evolutionary algorithms (EAs) are mostly recommended to solve multi-objective optimization while multi-modal problems are less frequently addressed. As described in Section 2.3 in the context of simulation-based optimization for power systems design, multi-modal algorithms offer a better chance for a robust design compared to local and global search algorithms (e.g., direct search, gradient-based and EA methods) [33]. Therefore, multi-modal optimization algorithms have become the center of attention in the field of electric power system design [27], [28].

A bi-objective formulation of a multi-modal optimization problem was described [48] in Section 2.2.1, which functions by adding a second objective to the underlying single-objective function [48]. This multi-modal GA suffers from high number of objective function evaluations [48]. However, it is considered as one of the most outstanding algorithms in the field of optimization and a thorough study of its scalability is reported in [48]. Superiority of the proposed algorithm described in Chapter 3 of this thesis will be verified versus this multi-modal GA through a series of comparative studies.

Recently, an algorithm was proposed for multi-modal optimizations problems [8], [27]; it possesses novel features that can reduce the number of objective function evaluations. This adaptive multi-modal algorithm can be enhanced further if it can provide an estimation of the objective function around each local area. Hence, sensitivity analysis can be conducted with the use of these estimated models instead of experimenting with the computationally expensive black-box simulation.

Therefore, an advanced version of the adaptive multi-modal optimization algorithm [27] was recently provided in [28] where the number of objective function evaluation has been decreased significantly and surrogate models of the objective function around each local minimum are made to speed up sensitivity analysis of the local optimal solutions.

Parallel operation of these multi-modal algorithms to improve their computational power has not been addressed nor are they implemented in a power system simulator. Also approaches for memory management during optimization have not been addressed. Therefore, optimization of complex black-box systems where a large number of variables are to be optimized is infeasible using these multi-modal algorithms as they require unreasonable computation time and memory usage.

In the surrogate model-based algorithm [28], the second derivative of the surrogate model function is necessary for sensitivity analysis. Avoiding extra computations, a new approach can be offered that improves and at the same time simplifies sensitivity analysis.

Chapter 3 will propose a novel multimodal optimization algorithm that overcomes the mentioned difficulties. Moreover its computational complexity will be compared against the surrogate-based multi-modal algorithm [28] in terms of the number of objective function evaluations and sensitivity assessment. The scalability of the proposed algorithm will finally be compared to the multi-modal GA [48].

Chapter 3

Description of the Proposed Multi-Modal Optimization Algorithm

In this chapter a novel algorithm is proposed to (i) discover all local optima of a multi-modal, black-box, nonlinear objective function, and (ii) estimate the objective function around the discovered local minima using analytical interpolation methods. These estimated functions represent the local behaviour of the objective function, and hence can be used in the subsequent stages of post-optimization studies (e.g., sensitivity and statistical analyses) and avoid the need for intensive simulation-based evaluations of the objective function. Therefore, significant improvements in computational intensity are obtained. Comparative studies in terms of objective function evaluation and estimation accuracy are provided, using several multi-modal benchmark objective functions. The superiority of the proposed algorithm compared to other recently-developed algorithms is thereby verified.

3.1 Description of the Algorithm

In this section a new algorithm is proposed for optimization of non-linear black-box multi-modal objective functions. Considering computational and optimality perspectives the proposed algorithm provides the following features:

Computational perspective:

- i. The new algorithm is highly parallelizable. It is implemented in a parallel environment interfaced with an electromagnetic transients (EMT) simulator to significantly speed-up the design of power systems (as will be explained in Chapter 4);
- ii. It is simple to implement. The algorithm is implemented in the PSCAD/EMTDC simulator to provide a convenient means for optimal design of complex electrical power systems.

Optimality perspective:

- i. The algorithm is capable of discovering multiple local solutions for a given complex, nonlinear, black-box objective function. This finds a variety applications in complex power system optimization;
- ii. It is capable of handling both linear and nonlinear constraints;
- iii. It provides surrogate models for the objective function around its local optima. These models can be used in the subsequent stages of post-optimization studies (e.g., sensitivity and statistical analyses) to reduce the need for intensive evaluations of the original objective function using simulations.

In the following sections, the new algorithm is described in the context of a minimization problem, and for a two-dimensional case study, which makes visual inspection of the algorithm's progress possible. Although the proposed algorithm will eventually be used for black-box objective functions, the example selected here is mathematically explicit and hence exactly has known minima, which can be used to check the algorithm's accuracy.

Consider Himmelblau's objective function and its constraints as given below.

$$\begin{aligned} &\text{minimize } f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \\ &\text{subject to: } \begin{cases} -5.5 \leq x \leq 5 \\ -5.5 \leq y \leq 5 \\ \frac{(x+5)^2}{25} + \frac{(y-5)^2}{16} \geq 1 \\ \frac{(x-5)^2}{25} + \frac{(y+5)^2}{25} \geq 1 \\ x^2 - x + y^2 + 2y \leq 41 \end{cases} \end{aligned} \quad (3-1)$$

The contour plot of the objective function and its associated constraints are depicted in Figure 3-1. The 4 local minima, shown by asterisks, are analytically available and are located at (-3.78,-3.28), (+3.58,-1.85), (+3.00,+2.00) and (-2.80,+3.13). The objective function value for all these local minima is equal to zero. The objective of the proposed algorithm is to find the local optima, subject to the constraints indicated in (3-1). Note that the constraints shown in (3-1) eliminate two of the local minima (i.e., (+3.58,-1.85) and (-2.80,+3.13)) as they lie outside the specified boundaries.

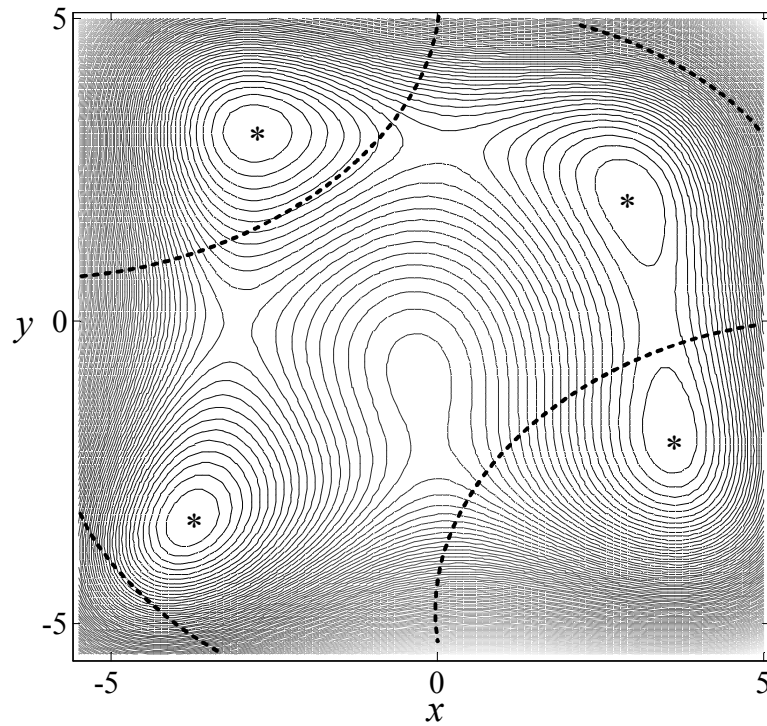


Figure 3-1: Contour plot of Himmelblau's function and boundary of constraints.

Figure 3-2 shows the steps of the proposed multi-modal optimization algorithm, which include (i) initialization, (ii) localization, (iii) square mesh generation, (iv) objective-function estimation, (v) density increment, and (vi) convergence check. These steps will be described in the following sections.

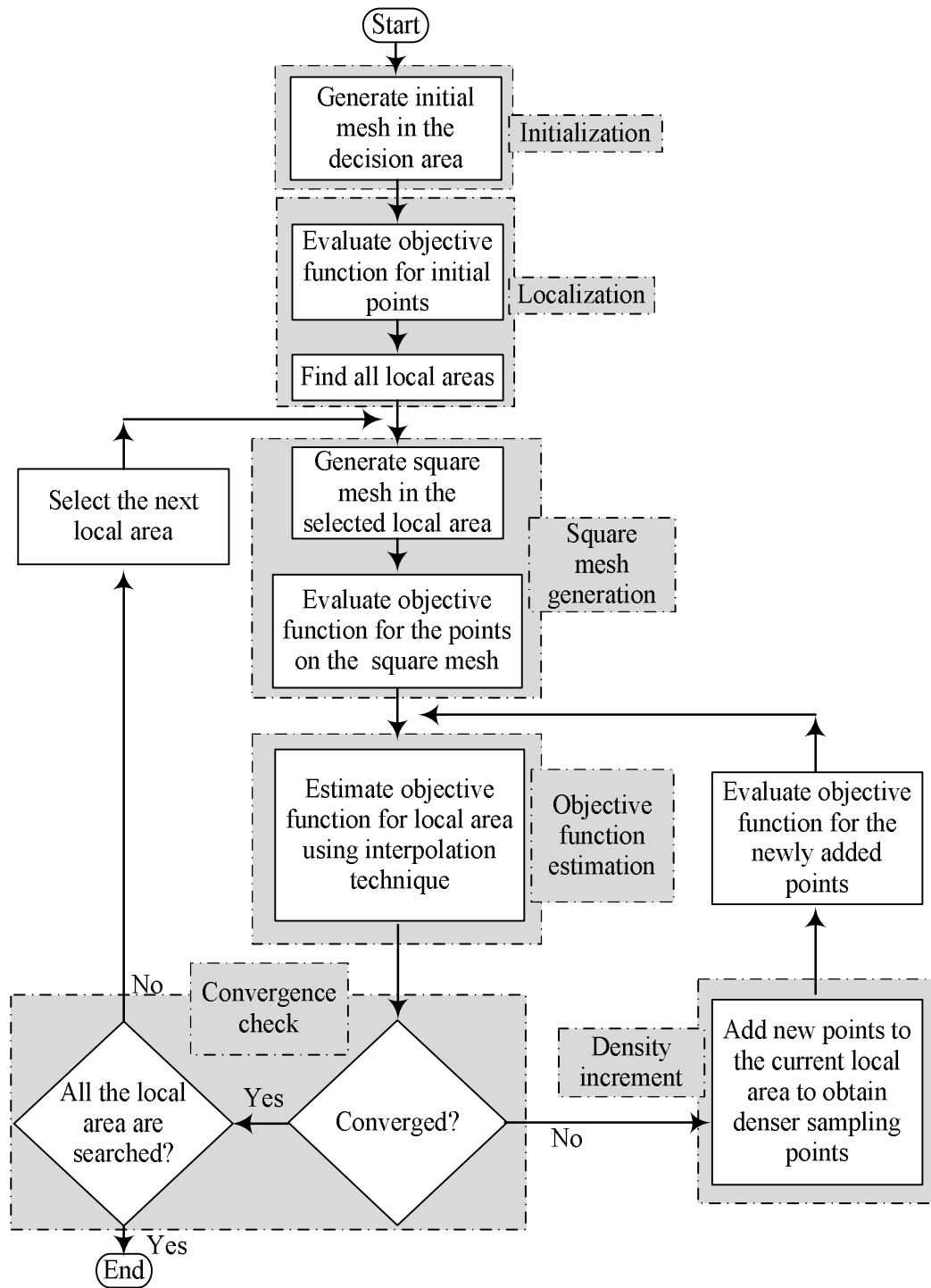


Figure 3-2: Flowchart of the proposed multi-modal optimization algorithm.

3.1.1 Initialization

The first stage of the algorithm (Initialization) aims to place distributed sampling points in the decision area to obtain an appropriate initial estimation of the objective function. To do so a specific mesh generation technique [8], [54] is used that consists of three stages, namely random point generator, Delaunay triangulation, and force relaxation [8].

- i. **Random point generation:** In this stage a number of initial points are randomly scattered in the decision area according to constraints (see for example (3-1)). This process is shown in Figure 3-3, in which the number of points is 110.
- ii. **Delaunay triangulation:** In this stage the points are connected by links using a triangulation algorithm as it is shown in Figure 3-4. In particular Delaunay triangulation is exploited to indicate the direction of a point's movement in order to maximize the smallest angle and minimize the largest angle of all the triangles in the mesh shown in Figure 3-4. As a result the triangles become as equiangular as possible. This will fulfill the requirements for uniformly distributed points that sample the entire decision area.
- iii. **Force relaxation:** This technique aims to fulfill the Delaunay triangulation principle to obtain equiangular triangles in the mesh, which leads to expanding sampling points in the decision area. In this technique links are treated as springs that can impose inward or outward forces to the points at either end. In the mesh shown in the Figure 3-4 the points are connected via imaginary springs whose forces can be calculated by the following formula.

$$repulsive\ force(m, n) = \begin{cases} k(\ell_{m,n} - L_{m,n}) & \ell_{m,n} \leq L_{m,n} \\ 0 & \ell_{m,n} > L_{m,n} \end{cases} \quad (3-2)$$

where k is the spring constant, $L_{m,n}$ is the predefined length of the link between points n and m , and $\ell_{m,n}$ is the actual length of the link connecting the two points.

If the length of a link between two points is smaller than its predefined length ($L_{m,n}$) then the spring element exerts repulsive forces to its end points. So for the squeezed mesh shown in Figure 3-4 the repulsive forces cause the points to expand and fill up the area. Hence, these forces are referred to as internal forces that depend on the position of the points compared to each other. The boundaries are assigned inward forces to ensure movement of the points does not cause their departure from the decision area and hence ensures adherence to the constraints as shown in Figure 3-4. It is possible to assign shorter nominal length to the springs in such areas close to the boundaries, allowing the points to come closer to one another, and hence increase the accuracy of constraint handling [28].

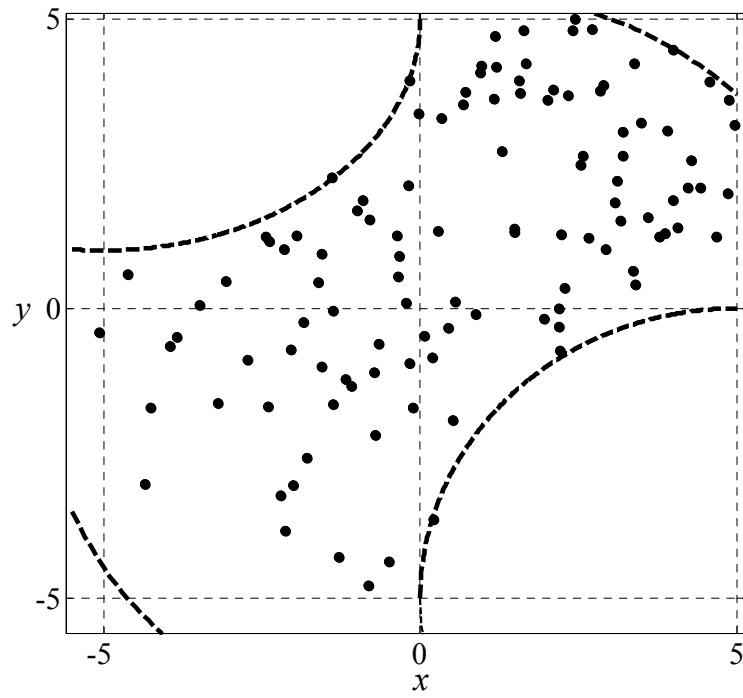


Figure 3-3: Random sampling of the decision area satisfying non-linear constraints.

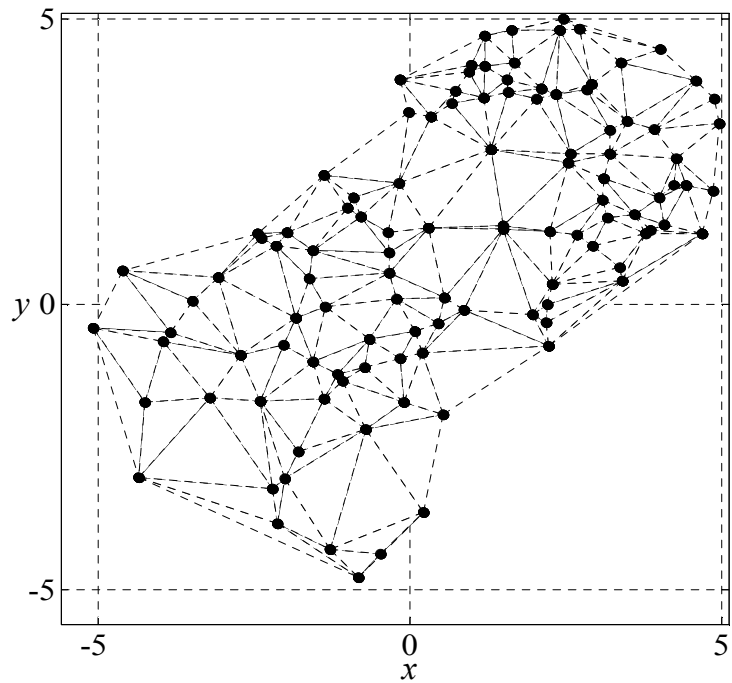


Figure 3-4: Initial mesh created by the Delaunay triangulation.

In the mesh shown in Figure 3-5 a higher density of points is achieved around the upper and lower side boundaries.

The combination of force relaxation and the Delaunay triangulation is a reliable technique to produce a uniform mesh that covers the decision area and at the same time satisfies the constraints. This method was first proposed by Persson in [54], and it is simple to implement and efficient in term of computations.

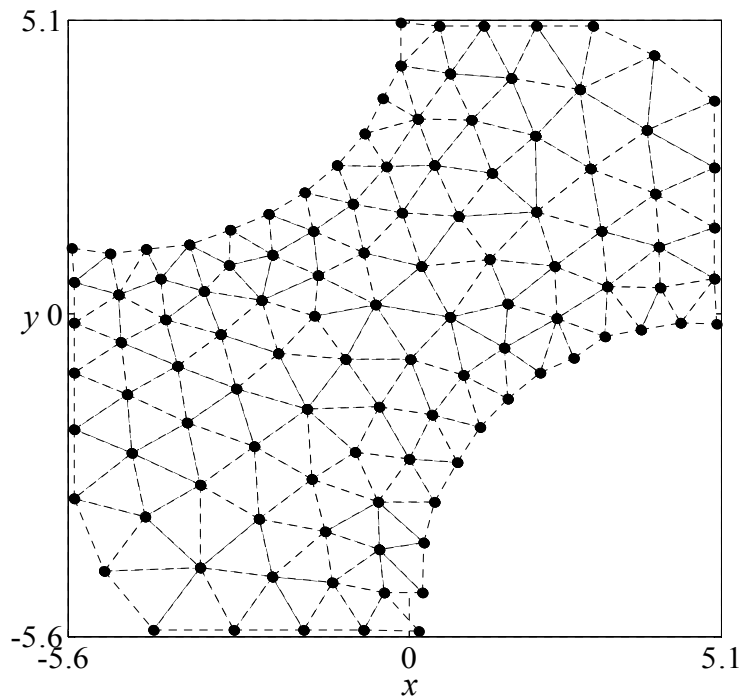


Figure 3-5: Equiangular triangle mesh satisfying non-linear constraints.

Note that the success of the proposed optimization algorithm in discovering all local minima relies on the quality of the mesh generated in this stage of the algorithm [8].

The number of sampling points (n_s) during initialization can be calculated as follows:

$$n_s = \prod_{i=1}^{n_v} (n_a)_i \quad (3-3)$$

where n_v and $(n_a)_i$ are the number of decision variables and samples for the i -th variable, respectively. For this example, $n_v = 2$ and $(n_a)_1 = 11$ and $(n_a)_2 = 10$ then the number of sampling points can be calculated as:

$$n_s = 11 \times 10 = 110$$

Some of these points could be removed during initialization due to mesh refinements and/or applying the constraints. The variable n_a may be defined for each decision variable according to the complexity of optimization problem. However, the number of sampling points must be kept as small as necessary to avoid high memory usage and/or computational problems. In an optimization problem with a large number of decision variables and accordingly a large number of sample points, memory requirements may exceed the available capacity. This issue is addressed in Section 4.3.

3.1.2 Localization

For the distributed points shown in Figure 3-5, the objective function will be evaluated using simulation in the case of a black-box function or direct evaluation if the objective function is explicitly available (as is the case in this example). The points with a lower OF value compared to their adjacent points are identified. These points and their adjacent points are considered as a local area. In a multi-modal optimization problem it is expected that the algorithm will find a number of local areas. The local areas surrounding

these localized points are likely to contain a local minimum and are hence selected for further exploration.

Figure 3-6 illustrates localization of the objective function in (3-1) based on the initial mesh provided in the preceding section (see Figure 3-5). This objective function has two analytically available local minima in the defined area, denoted by the asterisks at $(+3.00, +2.00)$ and $(-3.78, -3.28)$ in the local areas (see Figure 3-6). The aim of the algorithm is to find these points. They are shown in Figure 3-6 to prove that the algorithm has correctly identified the two local areas associated with the local minima. The next steps of the algorithm aim to find the local minima with increasing accuracy by concentrating the search in the discovered local areas. A description of these steps follows next.

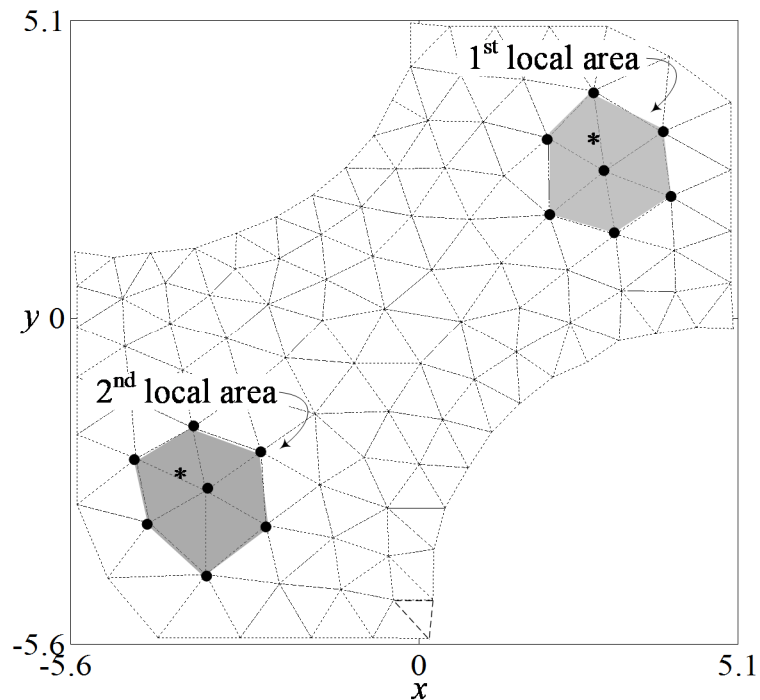


Figure 3-6: Highlighted local areas.

3.1.3 Square Mesh-Grid Generation

Figure 3-7 shows two square-mesh grids, in which the local areas of Figure 3-6 are fully inscribed. As shown the 1st and 2nd local areas are covered by a 3×3 mesh (9 points). The number of points per area is a matter of choice; more points lead to a higher accuracy, but at the expense of additional computational intensity. The reason to choose a uniformly distributed square mesh is that it improves accuracy and reduces complexity of the calculations of interpolation methods that will be used to estimate the objective function. The objective function for these points will be evaluated and used in the interpolation process to estimate a mathematical expression for the objective function.

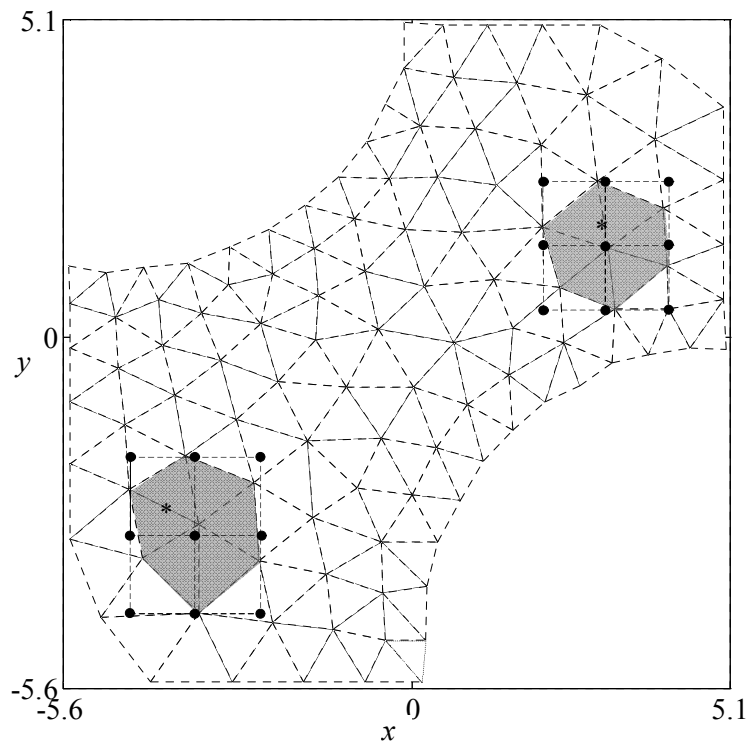


Figure 3-7: Square-mesh grids covering local areas.

3.1.4 Objective Function Estimation via Interpolation

Consider the square-mesh grid formed around the first local area in Figure 3-7. This square mesh has 9 points (p_1 to p_9) as denoted by dark circles in Figure 3-8. The actual local minimum is also indicated by the asterisk for description purposes. Using an interpolation method such as *Cubic-Spline* [55], [56], the objective function can be estimated for the local areas.

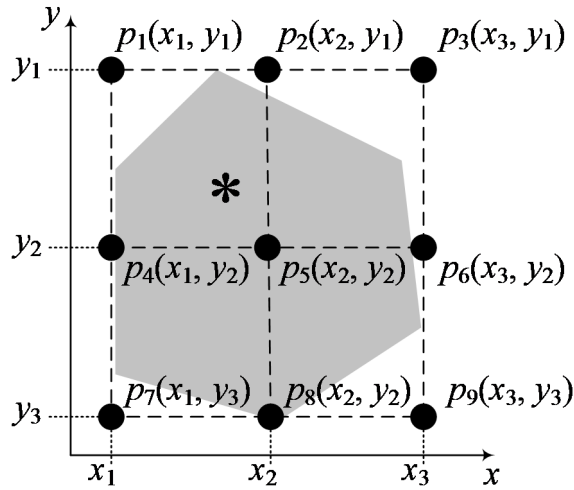


Figure 3-8: Square-mesh covering the 1st local area.

Cubic-Spline is an effective means to interpolate the objective function with a series of cubic polynomials [55]-[57]. The cubic polynomials construct continuous and smooth curves that pass through each of the points to mimic the behaviour of the objective function with no breaks or erratic changes [56]. Therefore the surrogate function ($f_s(x, y)$) obtained by this method provides an appropriate measure to estimate the complex black-box objective function. According to Figure 3-8, a piecewise form of the surrogate function can be shown as follows:

$$f_s(x, y) = \left\{ \begin{array}{l} \left. \begin{array}{l} \text{OF}(p_1) + a_1(x - x_1)^3 + b_1(x - x_1)^2 + c_1(x - x_1) \\ \text{OF}(p_4) + a_2(x - x_1)^3 + b_2(x - x_1)^2 + c_2(x - x_1) \\ \text{OF}(p_7) + a_3(x - x_1)^3 + b_3(x - x_1)^2 + c_3(x - x_1) \end{array} \right\} x_1 \leq x < x_2 \\ \left. \begin{array}{l} \text{OF}(p_2) + a_4(x - x_2)^3 + b_4(x - x_2)^2 + c_4(x - x_2) \\ \text{OF}(p_5) + a_5(x - x_2)^3 + b_5(x - x_2)^2 + c_5(x - x_2) \\ \text{OF}(p_8) + a_6(x - x_2)^3 + b_6(x - x_2)^2 + c_6(x - x_2) \end{array} \right\} x_2 \leq x < x_3 \\ \left. \begin{array}{l} \text{OF}(p_1) + a_7(y - y_1)^3 + b_7(y - y_1)^2 + c_7(y - y_1) \\ \text{OF}(p_2) + a_8(y - y_1)^3 + b_8(y - y_1)^2 + c_8(y - y_1) \\ \text{OF}(p_3) + a_9(y - y_1)^3 + b_9(y - y_1)^2 + c_9(y - y_1) \end{array} \right\} y_1 \leq y < y_2 \\ \left. \begin{array}{l} \text{OF}(p_4) + a_{10}(y - y_2)^3 + b_{10}(y - y_2)^2 + c_{10}(y - y_2) \\ \text{OF}(p_5) + a_{11}(y - y_2)^3 + b_{11}(y - y_2)^2 + c_{11}(y - y_2) \\ \text{OF}(p_6) + a_{12}(y - y_2)^3 + b_{12}(y - y_2)^2 + c_{12}(y - y_2) \end{array} \right\} y_2 \leq y < y_3 \end{array} \right. \quad (3-4)$$

where $\text{OF}(p_i)$ is the value of objective function at the point p_i , and $a_1 - a_{12}$, and $b_1 - b_{12}$, and $c_1 - c_{12}$ are the coefficients of the function that should be calculated.

The following conditions are satisfied by the piecewise surrogate function ($f_s(x, y)$).

- i. it passes through all the available points (i.e. p_1, \dots, p_9);
- ii. it is continuous over the intervals, $[x_1, x_3]$ and $[y_1, y_3]$;
- iii. its first derivatives will be continuous over the intervals, $[x_1, x_3]$ and $[y_1, y_3]$;
- iv. its second derivatives will be continuous over the intervals, $[x_1, x_3]$ and $[y_1, y_3]$.

The Cubic-Spline method used here offers a great deal of flexibility in interpolating functions with complex surface geometries; nonetheless, if the function has excessively complex shapes or discontinuities, the interpolation procedure may require more

computation and be less accurate [57]. Since interpolation in the proposed algorithm takes place in small neighborhoods around each local minimum, it is expected that the accuracy of estimation be high without recourse to a large number of iterations.

3.1.5 Convergence Check and Final Outcomes

Once the OF is estimated for a local area by the algorithm, the point with the lowest value is found analytically from the estimated OF. This point is considered as the final estimation of local minimum in its local area, if it meets the following criteria:

- i. Accuracy is confirmed by re-evaluating the discovered local minimum using the actual OF through simulation. If these two values (i.e., estimated and actual) are close enough with a difference of less than a given threshold, then convergence is assumed. The following expression shows the *error* that is used to test the accuracy.

$$error(i) = \|OF_{ac}(i) - OF_{st}(i)\| \quad ; \quad i \in \{1, 2, 3, \dots, N\} \quad (3-5)$$

where $OF_{ac}(i)$ and $OF_{st}(i)$ are the actual, and estimated values of the objective function at the i^{th} minimum point, respectively, and N is the total number of local minima in the decision area; or

- ii. if a certain level of proximity in the adjacent points in the current searching local area is achieved; or
- iii. if the algorithm reaches a certain number of iterations.

If the algorithm meets the preceding criteria, it has converged to a local optimum in the current local area. Then the algorithm begins to search for the next local optimum in

the next local area until all have been searched. Note that not all the N meshes may indicate convergence at the same time. In that case, those particular meshes must be made denser as described below.

It must be noted that the analytical minimization of the surrogate functions could potentially result in a solution that lies outside the feasible boundary of decision variables. The likelihood of this, however, is remote as the region of interest around a local minimum is generally small, which allows assumption of a locally convex geometry. If required, one solution to this issue could be the modification of the objective function to include this constraint more precisely [21].

3.1.6 Density Increment

This stage of the algorithm aims to further increase the accuracy of the estimation of the local minima. If the convergence criteria are not met, it is necessary to increase the accuracy of the estimated objective function. This is done by adding additional points to the original mesh grid (see Figure 3-7) as shown in Figure 3-9. The newly added points are shown by light colored circles, for which the objective function will be evaluated using simulation. The number of such points is a matter of choice and can be decided by the user. As an example, the new points can be added onto the midpoint of each link in the mesh. Doing so, the total number of newly-added points (n_{na}) can be calculated as follows:

$$n_{na} = (n_{ne})n_v - (n_{ex})^{n_v} = (2n_{ex} - 1)^{n_v} - (n_{ex})^{n_v} \quad (3-6)$$

where n_v , n_{ex} and n_{ne} are the number of decision variables, and the number of points on each axis of mesh grid before and after the new points are added, respectively.

Since the mesh of Figure 3-9 has more evaluated points, the estimated objective function can become more accurate. As a result, it is more likely to obtain a more accurate value for the local optimum.

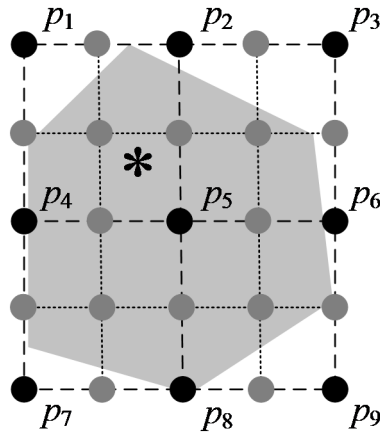


Figure 3-9: Density increment to increase the accuracy of estimated objective function.

3.1.7 Accuracy Investigation of Local Optimal Solutions and Estimated Objective Functions

Table 3-1 shows numerical results of three consecutive iterations of the algorithm for the optimization problem stated in (3-1). Following the steps described earlier, the algorithm detects two local areas, around which square meshes of 9 points are built initially. Based on these 9 points an estimated objective function is obtained for each local area. Based on the estimated objective function a minimum is calculated analytically. Then the algorithm checks the convergence criteria. During the convergence check the objective

function value of the minimum point discovered by the algorithm will be compared with the objective function value obtained from the actual objective function. In Table 3-1, the error shows the closeness of the discovered minimum points to their actual value. For the first iteration this error is not acceptable. Therefore, the algorithm starts the next iteration. In the second iteration the local area is unchanged but new points are added to the exciting points in each local area. The number of new points can be calculated using (3-6). Knowing that $n_v=2$, and $n_{ex}=3$ the number of newly-added points can be calculated as follows:

$$n_{na} = (n_{ne})n_v - (n_{ex})^{n_v} = (2n_{ex} - 1)^{n_v} - (n_{ex})^{n_v} = (2 \times 3 - 1)^2 - (3)^2 = 16$$

These new 16 points are evaluated and then based on the total of 25 (=16+9) evaluated points an estimated objective function will be interpolated for the local area. This time, as expected, a better objective function is estimated, and as a result a better position is projected for the minimum and a smaller error is obtained (see Table 3-1, 2nd iteration). If a more accurate value for the minimum points is desired the algorithm continues to the next iteration (3rd iteration) and adds new point to the 25 exciting points in each local area. From (3-6) the number of new points can be calculated as follows:

$$n_{na} = (2 \times 5 - 1)^2 - (5)^2 = 56$$

In this iteration a very precise estimation of the objective function in the local area is obtained as the density of evaluated points is much larger compared to the 1st and 2nd iterations (see Figure 3-10). Thus, the error in the 3rd iteration becomes almost zero as shown in Table 3-1.

Table 3-1: Summary of the results for the optimization problem in (3-1).

Local area		Iteration		
		1 st	2 nd	3 rd
1 st local area $x \in [4, 2]$ $y \in [0.5, 2.5]$	#Function evaluation	9	9+16=25	25+56=81
	Position of discovered minima	$x=-3.699$ $y=-3.451$	$x=-3.7901$ $y=-3.2797$	$x=-3.8027$ $y=-3.2873$
	RMS_{error}	5.056	1.96×10^{-3}	63.1×10^{-6}
2nd local area $x \in [-2, -4.5]$ $y \in [-2, -4.5]$	#Function evaluation	9	25	81
	Position of discovered minima	$x=2.917$ $y=1.888$	$x=3.0097$ $y=1.9727$	$x=2.9992$ $y=2.0063$
	RMS_{error}	3.873	0.64×10^{-3}	142.7×10^{-6}

Table 3-2 shows the error in the estimated objective function, for each local area. The error is defined by the root mean square error (RMS_{error}) as follows.

$$RMS_{error} = \left(\frac{1}{M} \sum_{j=1}^M \|OF_{ac}(j) - OF_{st}(j)\|^2 \right)^{\frac{1}{2}} ; \quad j \in \{1, 2, 3, \dots, M\} \quad (3-7)$$

where $OF_{ac}(i)$ and $OF_{st}(i)$ are the actual, and estimated values of the objective function at the j^{th} sampling point, respectively. M is the total number of sampled points distributed evenly in each local area.

Figure 3-10 and Figure 3-11 show the estimated and actual objective-function surfaces for the two local minima of the function presented in (3-7). The Cubic-Spline [56] method is used for estimating the objective function values in this example.

In the 1st iteration (see Table 3-2) only 9 points are used to estimate an objective function in the local areas hence the RMS_{error} is relatively high in both local areas. Figure 3-9 (a) shows the estimated and the actual objective functions for 1st local area at the first iteration of the algorithm. There is a noticeable gap, indicating a high RMS_{error} , between the estimated and the actual objective functions. For the second iteration the error is reduced clearly (see Table 3-2, 2nd iteration) as the number of evaluated points is

increased to 25 in both areas. If more accuracy is demanded the number of evaluated points can be increased as is done to 81 in the third iteration.

Figure 3-10(b) shows the estimated and actual objective functions for 1st local area at the third iteration. The surfaces are uniformly united and the RMS_{error} is almost zero. Therefore, the estimated objective function for the 1st local optimum has acceptable accuracy to conduct post-optimization studies such as sensitivity analysis.

Table 3-2: The error in the estimated values for each local area.

Local area		Iteration		
		1 st	2 nd	3 rd
1 st local area	#Function evaluation	9	9+16=25	25+56=81
	RMS_{error}	0.546912	4.136×10^{-3}	194.7×10^{-6}
2 nd local area	#Function evaluation	9	25	81
	RMS_{error}	0.26455	1.88×10^{-3}	288.9×10^{-6}

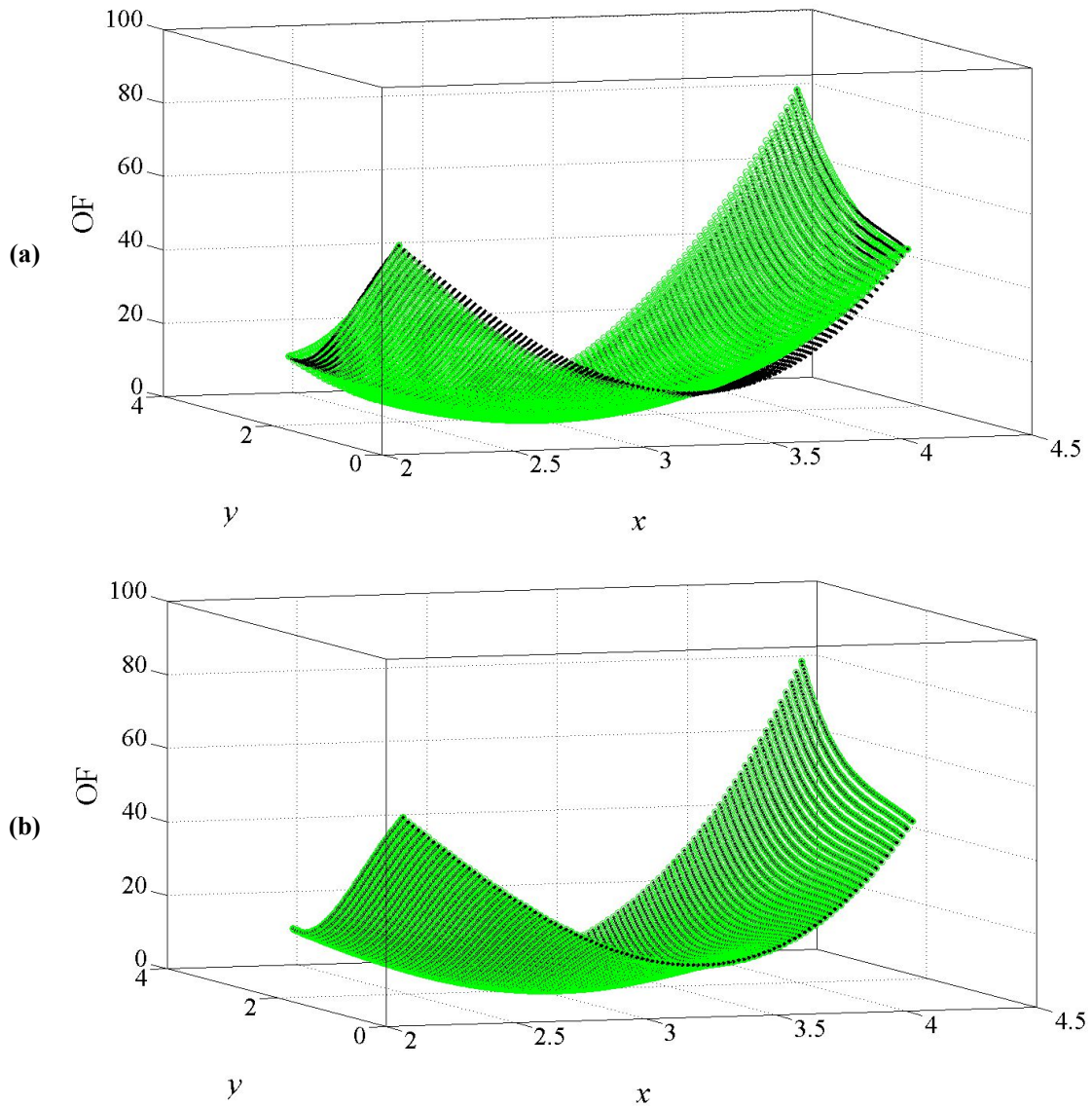


Figure 3-10: Estimated and actual objective functions for the 1st-local area. (a) 1st and (b) 3rd iteration.

The estimated and actual objective functions are shown Figure 3-11(a) and Figure 3-11(b) for the 2nd local area during 1st and 3rd iterations of the optimization algorithm, respectively.

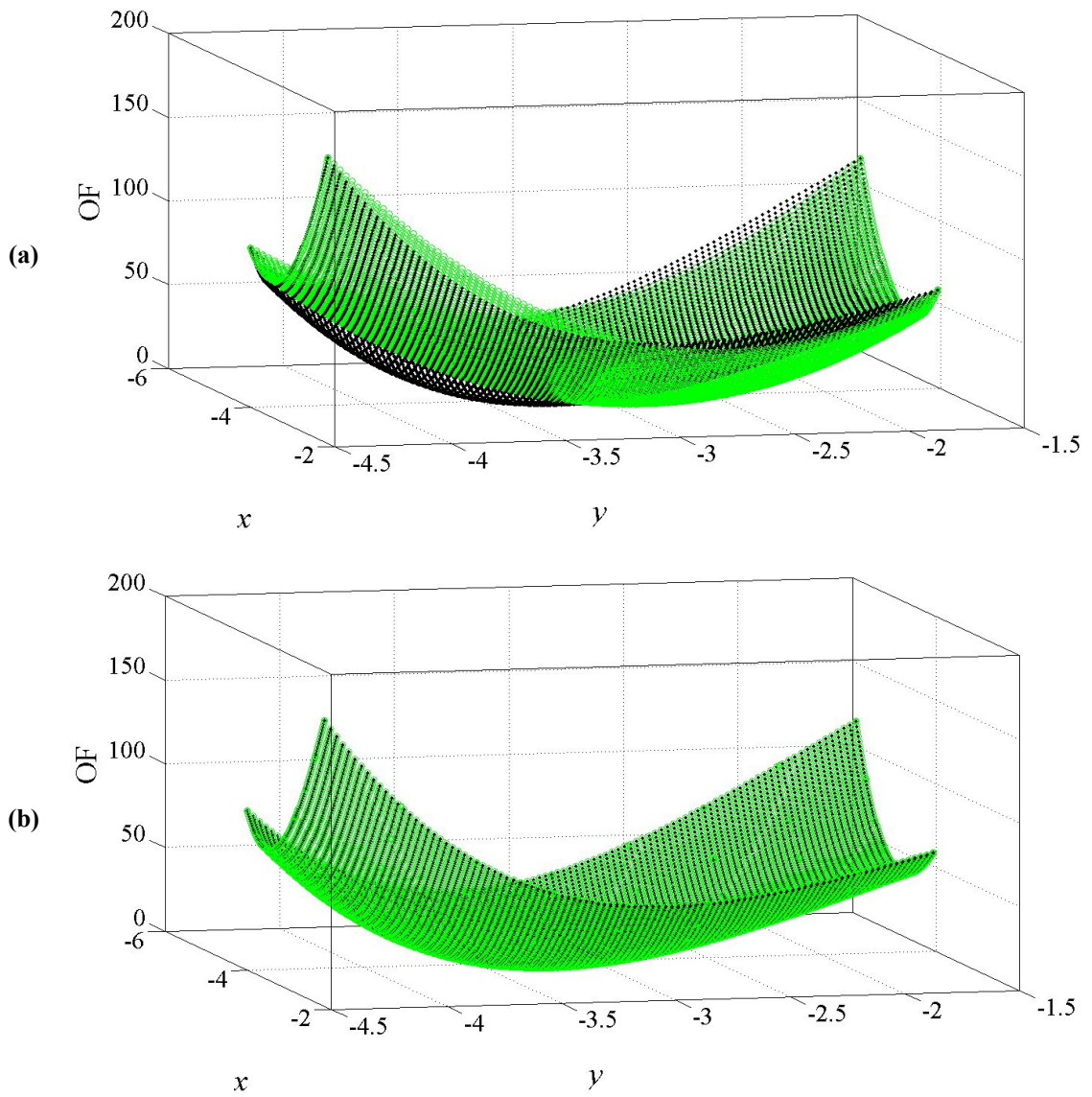


Figure 3-11: Estimated and actual functions for the 2nd-local area. (a) 1st and (b) 3rd iteration.

3.2 Sensitivity Analysis Using Surrogate Models of Objective Function

In any design it is possible that the optimized decision variables may be perturbed during a period of operation due to such reasons as aging and/or temperature variations [28], [29]. Therefore, for a robust design, the objective function value, which is a measure of performance, must be reasonably invariant to perturbations [7], [31]. As long as the algorithm finds a surrogate for the objective function in the vicinity of each local area, it paves the way to study the sensitivity of the objective function at each local optimum without expensive simulation runs [32], [33].

3.2.1 Methodology

There are different methods to conduct sensitivity analysis and recently a method has been developed in [7] where the second derivative of the objective function at each local optimum is derived using its local surrogate model [28]. The most robust local optimum is the one whose second derivative of its local surrogate model has the lowest value compared to those of the others [28].

Figure 3-12 shows two surrogate functions for two different local areas of an objective function whose optima (x_A^* and x_B^*) are indicated. The second derivatives of these surrogate functions are both positive and may be very close or equal at their respective optima (x_A^*, x_B^*) i.e.,

$$\left. \frac{d^2 f_1(x)}{dx^2} \right|_{x_A^*} = \left. \frac{d^2 f_2(x)}{dx^2} \right|_{x_B^*}$$

In this condition the method presented in [7] and [28] may fail to identify the most robust local optimum.

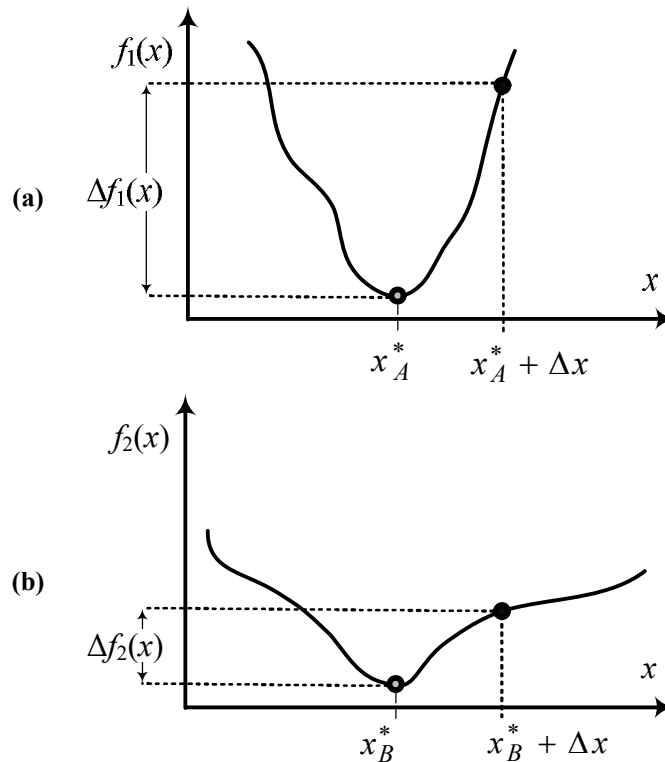


Figure 3-12: Sensitivity analysis using second-order derivatives surrogate functions (a) $f_1(x)$ and (b) $f_2(x)$.

Also if a maximum detuning (Δ) in the decision variable (x) is known then it would be more insightful to study the sensitivity of the objective function applying this detuning to each local area.

A sensitivity assessment approach is developed in this thesis, in which the accrual variations of the objective function can be measured in case of a detuning (Δ) in the local optimal solutions (x_A^* and x_B^*). Therefore, variations in the local surrogate models of the

objective function can be measured by the following formula when a positive detuning (+ Δ) is applied:

$$\begin{aligned}\Delta f_1(x) &= \left| f_1(x_A^* + \Delta x) - f_1(x_A^*) \right| \\ \Delta f_2(x) &= \left| f_2(x_B^* + \Delta x) - f_2(x_B^*) \right|\end{aligned}\tag{3-8}$$

Figure 3-12 shows positive detuning (+ Δ) in the local optimal solutions and the associated variations in the local surrogate functions. Negative detuning ($-\Delta$) is handled by replacing + Δ by $-\Delta$. Therefore the sensitivity study is directional and a vector of detuning can be considered to conduct the analysis. With this new method, prior knowledge of detuning can be applied to the local optimal solutions and also the worst case scenario (maximum detuning) can be investigated. The features of the proposed sensitivity analysis are summarized as follows:

- i. Sensitivity analysis can be conducted focusing on a single variable or several variables;
- ii. A certain range of detuning with a certain direction can be considered to conduct a sensitivity analysis;
- iii. Sensitivity assessment can be conducted with no simulation and less computation - no need to calculate the second derivative of the surrogate model.

3.2.2 Implementation of the Proposed Sensitivity Analysis

In this section sensitivity analysis is conducted for the 1st and 2nd local areas of the objective function shown in (3-1). A vector of variable detuning, which specifies both the

direction and magnitude (i.e., $\pm\Delta$), is considered in this sensitivity study. Two types of analyses will be conducted as introduced in the following:

- i. one parameter is detuned at a time, i.e., 1-dimensional or single-variable sensitivity analysis;
- ii. both parameters are detuned at the same time, i.e., 2-dimensional or in general multi-variable sensitivity analysis.

Figure 3-13 and Figure 3-14 show single-variable sensitivity assessment for the 1st and 2nd local areas, respectively. Variables x and y are detuned one at a time negatively and positively up to $\Delta=20\%$ for each local area.

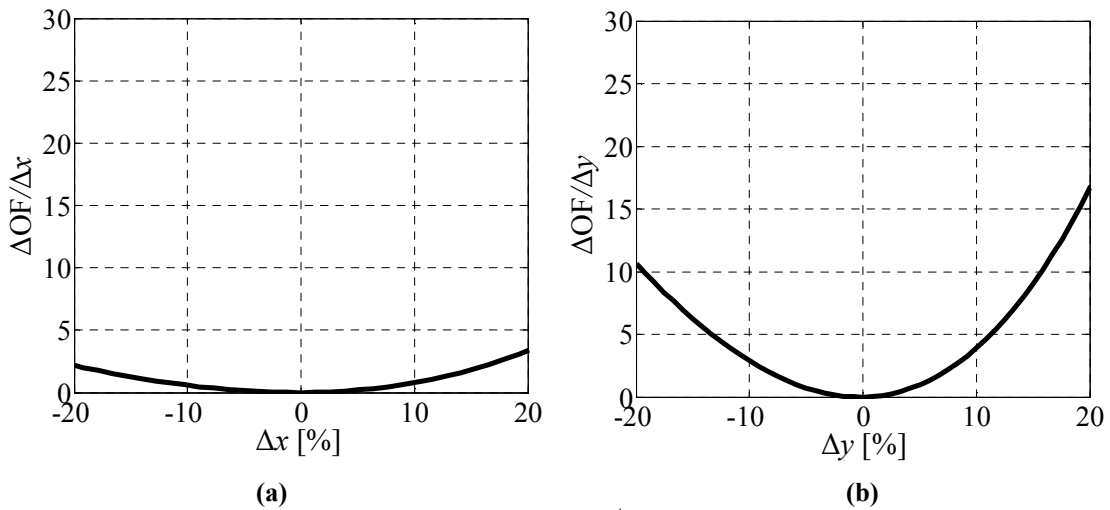


Figure 3-13: OF variations versus detuning for the 1st- local area. (a) variable x and (b) variable y .

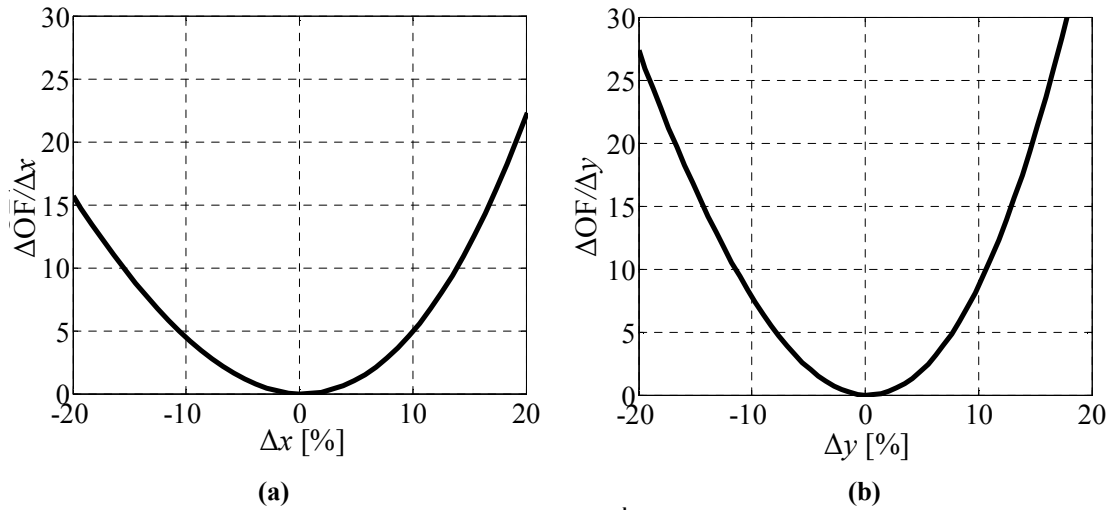


Figure 3-14: OF variations versus detuning for the 2nd- local area. (a) variable x and (b) variable y.

To compare the sensitivity of the objective function at each local area, Table 3-3 shows its variations (ΔOF) for a given detuning value extracted from Figure 3-13 and Figure 3-14. As an example, detuning values at $\Delta = \pm 5\%$ and $\Delta = \pm 10\%$ are selected. Based on the OF variations against the detuning in each variable ($\Delta OF / \Delta x$ or $\Delta OF / \Delta y$) the local area with less sensitivity is highlighted.

Table 3-3: One dimensional sensitivity analysis.

Detuning %	Parameter variations	$\Delta OF/\Delta x$ or $\Delta OF/\Delta y$	
		1 st local area	2 nd local area
$\Delta = \pm 5\%$	$\Delta x = -5\%$ $\Delta y = 0\%$	0.1583	2.1623
	$\Delta x = +5\%$ $\Delta y = 0\%$	0.1485	2.1867
	$\Delta y = -5\%$ $\Delta x = 0\%$	1.6106	3.8727
	$\Delta y = +5\%$ $\Delta x = 0\%$	1.6174	3.0854
$\Delta = \pm 10\%$	$\Delta x = -10\%$ $\Delta y = 0\%$	0.9558	4.8727
	$\Delta x = +10\%$ $\Delta y = 0\%$	1.0852	5.0854
	$\Delta y = -10\%$ $\Delta x = 0\%$	4.2210	7.5727
	$\Delta y = +10\%$ $\Delta x = 0\%$	3.8702	8.1854

Figure 3-15 shows the 2-dimensional sensitivity analysis for the 1st and 2nd local areas, respectively.

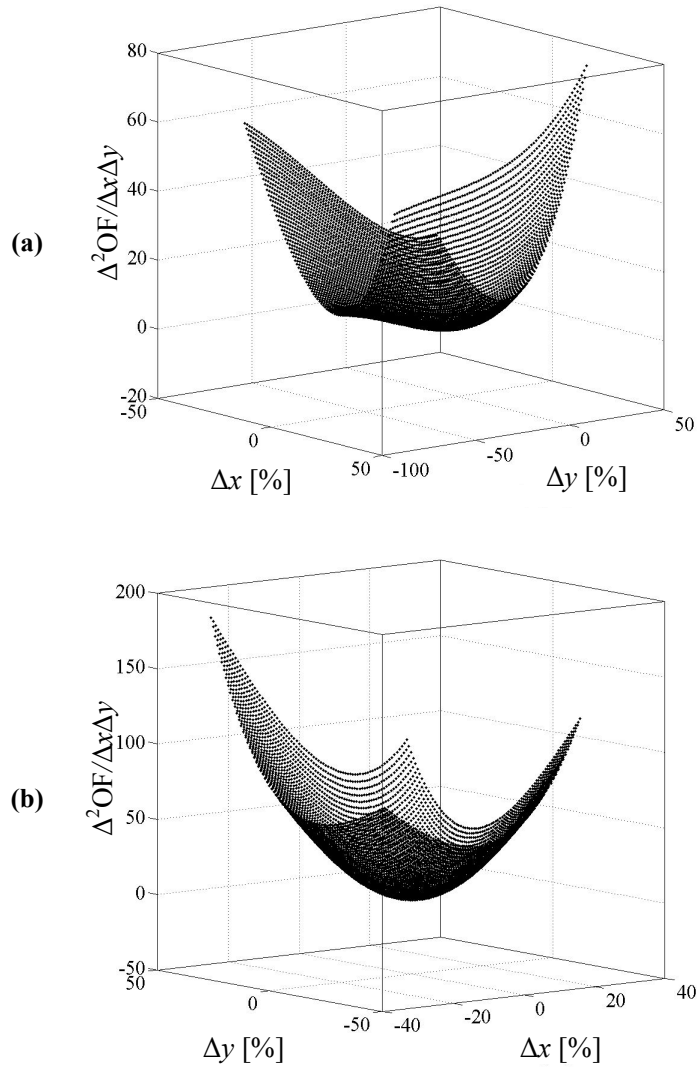


Figure 3-15: OF variation (ΔOF) versus the detuning in the variables. (a) 1st and (b) 2nd local area.

The second-order variations in the OF ($\Delta^2\text{OF}/\Delta x\Delta y$) can be measured from Figure 3-15 for a certain detuning, e.g. 5% and 10%. Table 3-4 shows the 2-dimensional analysis results and for all the combinations of a certain detuning in the 1st and 2nd local areas. The 1st local area has lower sensitivity in all combinations.

Table 3-4: 2-dimensional sensitivity analysis.

Detuning %	Parameter variations	$\Delta^2\text{OF}/\Delta x\Delta y$	
		1 st local area	2 nd local area
$\Delta=\pm 5\%$	$\Delta x = +5\%$ $\Delta y = +5\%$	1.5078	2.0421
	$\Delta x = +5\%$ $\Delta y = -5\%$	0.8251	3.8066
	$\Delta x = -5\%$ $\Delta y = +5\%$	0.6115	3.8526
	$\Delta x = -5\%$ $\Delta y = -5\%$	1.1675	2.3668
$\Delta=\pm 10\%$	$\Delta x = +10\%$ $\Delta y = +10\%$	6.0428	9.2569
	$\Delta x = +10\%$ $\Delta y = -10\%$	3.2833	15.5449
	$\Delta x = -10\%$ $\Delta y = +10\%$	2.4820	15.1331
	$\Delta x = -10\%$ $\Delta y = -10\%$	4.6723	8.4199

Table 3-5 shows the variations of the objective function for a range of detuning values. Maximum detuning in the OF ($\text{Max}(\Delta^2\text{OF}/\Delta x\Delta y)$) is selected to demonstrate the worst case scenario. Note that for $-10\% \leq \Delta x \leq +10\%$ and $-5\% \leq \Delta y \leq +5\%$ the 1st local area is less sensitive to detuning.

Table 3-5: 2-dimensional sensitivity analysis with continuous variation in detuning.

Detuning %	Parameter variations	Max($\Delta^2\text{OF}/\Delta x\Delta y$)	
		1 st local area	2 nd local area
$\Delta \leq \pm 5\%$	$-5\% \leq \Delta x \leq 5\%$ $-5\% \leq \Delta y \leq 5\%$	1.5078	3.8526
$\Delta \leq \pm 10\%$	$-10\% \leq \Delta x \leq 10\%$ $-10\% \leq \Delta y \leq 10\%$	4.9523	15.5449
$ \Delta x \leq \pm 5\%$ $ \Delta y \leq \pm 10\%$	$-5\% \leq \Delta x \leq 5\%$ $-10\% \leq \Delta y \leq 10\%$	1.7838	8.2050
$ \Delta x \leq \pm 10\%$ $ \Delta y \leq \pm 5\%$	$-10\% \leq \Delta x \leq 10\%$ $-5\% \leq \Delta y \leq 5\%$	3.8457	10.9088

3.3 Numerical Results and Comparative Assessment

This section compares the proposed optimization algorithm with its predecessors [8] and [28] as well as a well-known multi-modal evolutionary algorithm [48].

3.3.1 Comparative Study with the Surrogate-Model Based Algorithm

Here a single-variable multi-modal objective function is presented in (3-9) and used to compare the earlier surrogate-based optimization algorithm [28] with the one proposed in this thesis.

$$f(x) = \frac{x}{4} + (1 - \cos(\pi x)) \left(\tanh\left(\frac{x}{4}\right) - 1 \right)^2 \quad (3-9)$$

This function has six local minima over the decision interval $[-7, 6]$, as shown in Figure 3-16. The proposed algorithm was started with 100 points distributed over the given decision interval using mesh generation method described in Section 3.1.1. The algorithm correctly localizes six areas around the six minima of the objective function. The local surrogate models and sample points for each local area are shown in Figure 3-16 by bold lines and asterisks respectively. Table 3-6 shows a numerical comparative study between the surrogate-based optimization algorithm [27][28] and the proposed one. The given minima positions are obtained using the Linear, Cubic polynomial [55] and Spline (Cubic-Spline) interpolations [56] for the proposed algorithm, and the quadratic surface model for the surrogate-based algorithm.

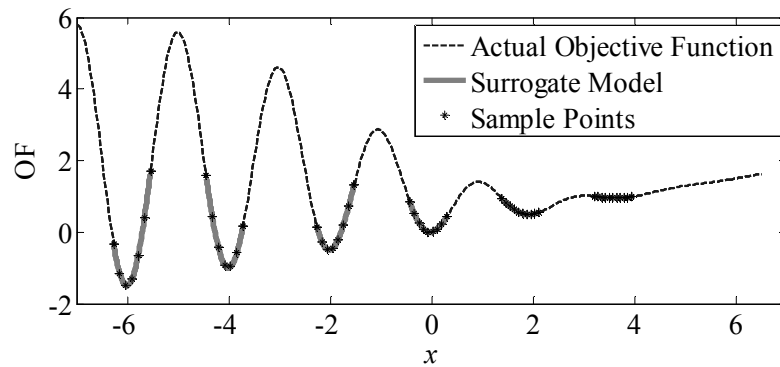


Figure 3-16: Objective function estimation for each local area.

Table 3-6: Comparative study of the surrogate-model based and the proposed algorithm.

Position of minima	Proposed algorithm			Surrogate-based algorithm
	Linear	Cubic	Spline	Quadratic surrogate model
	<i>RMS_{error}</i>			<i>RMS_{error}</i>
-6.0070	14.2×10^{-3}	10.8×10^{-3}	0.74×10^{-3}	0.0573
-4.0082	31.8×10^{-3}	7.56×10^{-3}	0.60×10^{-3}	0.0520
-2.0118	9.85×10^{-3}	3.19×10^{-3}	0.28×10^{-3}	0.0415
-0.0249	9.14×10^{-3}	1.75×10^{-3}	0.17×10^{-3}	0.0180
+1.9189	7.19×10^{-3}	0.41×10^{-3}	0.01×10^{-3}	0.0061
+3.6528	5.27×10^{-3}	0.14×10^{-3}	0.01×10^{-3}	0.0044
# function evaluation	605	235	172	245

The table compares the RMS_{error} between the estimated and the actual values in the vicinity of each local minimum for both algorithms. The RMS_{error} for the proposed algorithm is lower compared to the surrogate-based one [27], if estimation is conducted using the Cubic polynomial or Spline interpolations [57]. It should be mentioned that not only is the number of objective function evaluations smaller for the proposed algorithm using Spline interpolation but also the lower RMS_{error} is especially noticeable.

The number of objective function evaluations that led to the discovery of all local minima for the two optimization algorithms is investigated in Table 3-7, using a number of well-known multi-modal benchmark functions. The proposed algorithm provides optimization using different interpolation methods such as Cubic and Spline. As seen both algorithms find all local minima for all the benchmark functions but with different number of objective function evaluations. For example to discover all the minima of the *six-hump function* the surrogate-based algorithm requires 794 objective function evaluations, which is two times greater than that of the proposed algorithm.

Table 3-7: Comparative study of the two optimization algorithms.

$f(x)$	# variables	# minima	algorithms					
			Proposed			Surrogate-based [28]		
			Maximum $RMS_{error} (\times 10^{-3})$			# of function evaluations		
			Linear	Cubic	Spline			
Branin	2	3	13.53	1.96	0.1	73	100	
Six-hump	2	6	0.78	0.10	0.1	381	794	
Valleys	2	3	7.97	4.11	0.9	156	181	
Shekel	4	6	205.18	6.21	0.7	1104	1450	

The table also shows the maximum value for RMS_{error} among all local minima for each benchmark function resulting from the proposed algorithm. As can be seen the

RMS_{error} values for Spline interpolation technique are the smallest. Therefore it is recommended that the Spline interpolation technique [57] be used in the proposed algorithm.

Therefore, the proposed algorithm not only converges to the local minima but also correctly estimates the actual OF in the vicinity of the local areas.

3.3.2 Scalability Assessment

In this section the ability of the algorithm to handle optimization problems is assessed considering increases in the number of minimum points (n_p) and the number of decision variables (n_v).

3.3.3 Increase in the Number of Minimum Points

In this assessment the following modifiable two-variable Rastrigin's function with as many as 500 local optima is considered [48].

$$\begin{aligned} \text{minimize } f(x_1, x_2) &= (x_1^2 + x_1 + x_2^2 + 2.1x_2) + \sum_{i=1}^2 10(1 - \cos(2\pi x_i)) \\ \text{Subject to : } &\begin{cases} 0.5 \leq x_1 \leq (k_1 + 0.5) \\ 0.5 \leq x_2 \leq (k_2 + 0.5) \end{cases} \end{aligned} \quad (3-10)$$

The parameter values k_1 and k_2 can be adjusted so that the objective function attains a different number of local minima. Table 3-8 shows a few combinations of k_1 and k_2 along with the number of resulting local minima.

Table 3-8: the parameter values and the associated number of local minima.

Parameters	Number of local minima	Decision area
$k_1 = 4, k_2 = 4$	16	$x_1 \in [0.5, 4.5], x_2 \in [0.5, 4.5]$
$k_1 = 5, k_2 = 4$	20	$x_1 \in [0.5, 5.5], x_2 \in [0.5, 4.5]$
$k_1 = 10, k_2 = 5$	50	$x_1 \in [0.5, 10.5], x_2 \in [0.5, 5.5]$
$k_1 = 10, k_2 = 10$	100	$x_1 \in [0.5, 10.5], x_2 \in [0.5, 10.5]$
$k_1 = 20, k_2 = 10$	200	$x_1 \in [0.5, 20.5], x_2 \in [0.5, 10.5]$
$k_1 = 25, k_2 = 20$	500	$x_1 \in [0.5, 25.5], x_2 \in [0.5, 20.5]$

Here a state-of-the-art multi-modal evolutionary algorithm, namely Bi-Objective NSGA-II [48] is considered and the scalability results of the proposed algorithm and the Bi-Objective NSGA-II are demonstrated in Table 3-9. The total number of objective function evaluations (n_{fe}) is divided by the number of local minima, n_p (see the 2nd and 3rd columns from right in Table 3-9) to obtain the average required function evaluation per local area. For the proposed algorithm this metric is noticeably smaller than that of Bi-Objective NSGA-II algorithm. This means that the proposed algorithm requires far fewer function evaluations, which is fundamentally important to reduce the number of expensive black-box simulations in a typical black-box power system design.

Table 3-9: Comparative study results as the number of minimum points (n_p) increases.

n_p	Number of obtained minima		Number of function evaluations		Function evaluations per local minima		Computational complexity ratio (A/B)
	Proposed algorithm	NSGA-II	Proposed algorithm	NSGA-II	Proposed algorithm (A)	NSGA-II (B)	
16	16	16	169	5,600	169/16 \approx 10.5	5,600/16 \approx 350	3.01%
20	20	20	238	\approx 7,000	11.9	\approx 350.0	3.40%
50	50	49	561	\approx 17,500	11.2	\approx 357.1	3.20%
100	100	97	1,089	\approx 35,000	10.8	\approx 360.8	3.11%
200	200	194	6,216	\approx 70,000	31.1	\approx 360.8	8.60%
500	496	483	78,563	\approx 175,000	157.1	\approx 362.3	43.7%

Note that for problems with a larger number of minima (i.e., 100 or more), both algorithms sometimes miss a few. This is because some minima could be in very close proximity to each other and would be missed due to the mesh's granularity.

3.3.4 Increase in the Number of Decision Variables

In the previous section the performance of the proposed algorithm was investigated using optimization problems with few variables but a large number of minima. In this section, the scalability of the algorithm to handle multi-variable multi-modal benchmarks is verified. A multi-variable multi-modal optimization problem is shown as follows [48]:

$$\begin{aligned} \text{minimize } f(\mathbf{x}) &= \sum_{i=1}^{n_v} 10(1 + \cos(2\pi k_i x_i)) + 2k_i x_i^2 \\ \text{Subject to } &: 0 \leq x_i \leq 1, \quad i = 1, 2, 3, \dots, n_v \end{aligned} \quad (3-11)$$

where n_v is the number of decision variables.

The number of minimum points (n_m) for the above objective function can be calculated using the following formula [48].

$$n_m = \prod_{i=1}^{n_v} k_i \quad (3-12)$$

Therefore, the value for k_i can be adjusted to obtain different number of minima for the objective function given in (3-11). As the number of variables (n_v) increases from 2 to 10 it is possible to maintain a certain number of minima (eight in this example) for the objective function by adjusting the value of k_i as shown in the following table.

Table 3-10: Parameter values and the associated number of local minima.

n_v	k_i	$n_m = \prod_{i=1}^{n_v} k_i$
2	$k_1=2, k_2=4$	8
3	$k_1=k_2=k_3=2$	8
4	$k_1=1$ $k_2=k_3=k_4=2$	8
5	$k_1=k_5=1$ $k_2=k_3=k_4=2$	8
6	$k_1=k_2=k_6=1$ $k_3=k_4=k_5=2$	8
7	$k_1=k_2=k_6=k_7=1$ $k_3=k_4=k_5=2$	8
8	$k_1=k_2=k_6=k_7=k_8=1$ $k_3=k_4=k_5=2$	8
9	$k_1=k_2=k_3=k_7=k_8=k_9=1$ $k_4=k_5=k_6=2$	8
10	$k_1=k_2=k_3=k_7=k_8=k_9=k_{10}=1$ $k_4=k_5=k_6=2$	8

Not that all these problems consist of one global minimum and 7 local minima. The eight analytical solutions for the minimum values in the region $x_i \in [0,1]$ can be obtained by solving the following equation.

$$x_i = 5\pi \sin(2\pi k_i x_i) \quad (3-13)$$

Table 3-11 compares the number of objective function evaluation of the Bi-Objective NSGA-II algorithm and the proposed algorithm. Note that the Bi-Objective NSGA-II algorithm uses a population size as follows;

$$P_{size} = 15 \times \max(n_v, n_m) \quad (3-14)$$

where n_v is the number of variables and n_m is the number of minima.

Table 3-11: Comparative study results as the number of decision variables (n_v) increases.

n_v	Number of obtained minima		Number of function evaluations		Function evaluations per local minima		Computational complexity ratio (A/B)
	Proposed algorithm	NSGA-II	Proposed algorithm	NSGA-II	Proposed algorithm (A)	NSGA-II (B)	
2	8	8	154	$\approx 3,840$	19.25	480	4.0%
3	8	8	559	$\approx 4,800$	69.87	600	12.0%
4	8	8	1,393	$\approx 5,760$	175.1	350	24.2%
5	8	8	4,661	$\approx 6,720$	582.62	840	69.3%
6	8	8	18,697	NA	2337.1	NA	NA
7	8	8	84,269	NA	10533.6	NA	NA
8	8	8	402,913	NA	50364.1	NA	NA
9	8	8	1977701	NA	247212.6	NA	NA
10	8	8	9814777	NA	1226847.1	NA	NA

The number of objective function evaluations is shown in Table 3-11 to compare the computational complexity of the algorithms. Note that, the data of function evaluations per local area for the Bi-Objective NSGA-II are extracted from reference [48] and it is not available (NA) for problems with more than six variables. The computational complexity ratio shown in the last column of Table 3-11 is available up to 5 variables for NSGA II. As it is shown for the optimization problem with two variables the proposed algorithm requires just 4% of the computational demand compared to NSGA II.

3.4 Conclusions and Discussion

A multi-modal optimization algorithm was proposed in this chapter; it addressed optimization of several multi-modal benchmarks with multiple local minima. The proposed algorithm has modest computational intensity, due to the relative simplicity of the interpolation techniques used to estimate the objective function, and hence it provides

improvements over the previously developed class of surrogate-model based algorithms [8],[28].

The estimated objective functions produced during its course find use in post-optimization studies such as sensitivity analysis. The method developed for sensitivity analysis in this chapter features the following advantages over the traditional one [7]:

- i. A wider range of detuning can be considered for sensitivity assessment;
- ii. maximum detuning with different directions can be considered to reveal the worst case scenario; and
- iii. less computation is required with no need for the second-derivative calculation.

The scalability studies shown in Table 3-9 and Table 3-11 reveal important properties of the proposed algorithm. It was seen that the number of required objective function evaluations of the proposed algorithm was less compared to those of the considered evolutionary algorithm. This is particularly true when the number of local optima is large. For instance the proposed algorithm requires less than 4 percent of the computation complexity of Bi-Objective NSGA-II to obtain the solutions of a problem with 100 local optima.

3.4.1 Convergence Properties

The optimization algorithm proposed in this thesis is a heuristic algorithm, which is based upon geometric observations and operations. In this regard, it belongs to a large category of heuristic optimization algorithms. A number of heuristic algorithms, e.g. geometrically-based nonlinear Simplex, or biologically-based particle swarm

optimization, are widely used in nonlinear black-box optimization [21], [36] and [47]. Formal proofs of convergence for these algorithms, and the algorithm proposed herein, are unavailable due to their heuristic nature. It is, however, possible to present qualitative arguments to show a converging trend. For example, the first stage of the algorithm proposed in this thesis determines areas of the optimization space wherein local minima may lie by finding points with a lower OF value than their immediate neighbors. This process will discover at least one local area (corresponding to the points with the lowest evaluated OF among the rest). Once such points are discovered and bounded, the remaining stages of the search will only focus on the interior of the localized areas, and as such the search will never grow outwardly. Furthermore, the Cubic-Spline estimation method used is proven [56],[57], to have vanishing error as the number of points used for estimation grows, as is the case in this algorithm.

3.4.2 Computational Performance

The proposed algorithm is more computationally intense than some popular local optimization algorithms, such as the nonlinear Simplex [45]. In other words optimization using a wide range of commonly used algorithms will yield results in a shorter period of time than the proposed algorithm due to their lower computational intensity. It must, however, be noted that this is a direct consequence of the aim of the proposed algorithm in solving a far more general problem of simultaneously finding multiple local optima of a multi-modal function [47]. The majority of commonly used black-box optimization algorithms are local optimizers designed to find one locally optimal solution [29], [34]

and [35]. Global optimization algorithms such as genetic algorithms are also designed to only look for the global solution (again a single solution) [49]. The computational intensity of the proposed algorithm is naturally higher as it looks simultaneously for multiple local solutions. Additionally, the proposed algorithm forms explicit functions around each discovered local minima (via interpolation), such that analyses such as assessment of sensitivity to parameter variations can be performed using computationally inexpensive explicit functions rather than computationally intense EMT simulations. Therefore, this algorithm is far more suited for simulation-based design [10] as it provides multiple locally optimal solutions and a computationally inexpensive method for post-optimization sensitivity assessment. Besides, parallel nature of the proposed algorithm offers additional computational benefits that will arise in Chapter 4.

Chapter 4

Parallel Implementation of the Proposed Multi-Modal Algorithm

One challenge with multi-modal algorithms is that significant computation time may be required in complex optimization problems. The multi-modal algorithm developed in Chapter 3 is no exception in this regard.

This chapter introduces a parallel version of the developed multi-modal optimization algorithm that is proposed as a means to accelerate simulation-based optimal design [13]. Use of parallelism has not been exploited in the multi-modal optimization for power systems design [58], [60]. The proposed parallel algorithm has significantly higher computational speed than its sequential version and hence can greatly shorten the design cycle. Using the same method as was described in Chapter 3, the parallel version of the algorithm can also generate surrogate models of the objective function surface, which can

be readily used for sensitivity assessment of an objective function against possible perturbations in its variables.

4.1 Description of the Proposed Parallel Multi-Modal Algorithm

Figure 4-1 shows the parallel version of the multi-modal optimization algorithm that was proposed in Chapter 3. The parallel algorithm, similar to its sequential version, has the five stages of, (i) initialization, (ii) localization, (iii) objective function estimation, (iv) density increment, and (v) convergence check.

As described in Section 3.1.1, the purpose of the initialization stage of the algorithm is to develop a scatter of n_s initial sampling points in order to provide an estimation of the overall surface of the objective function (OF). In the context of design using an EMT simulator, the OF is only evaluated in “black-box” form and each objective function evaluation requires one EMT simulation run [61]. The EMT simulator used for the implementation of the parallel algorithm is the PSCAD/EMTDC.

In order to run the simulations in parallel a feature of PSCAD/EMTDC namely parallel multiple run (PMR) is used. Using this feature the optimization algorithm is allowed to run a number of EMT simulations in parallel, and hence a number of sample points can be evaluated simultaneously. Obviously, the number of EMT simulations that can be run in parallel is equal to the number of available processors, which should be known prior to optimization.

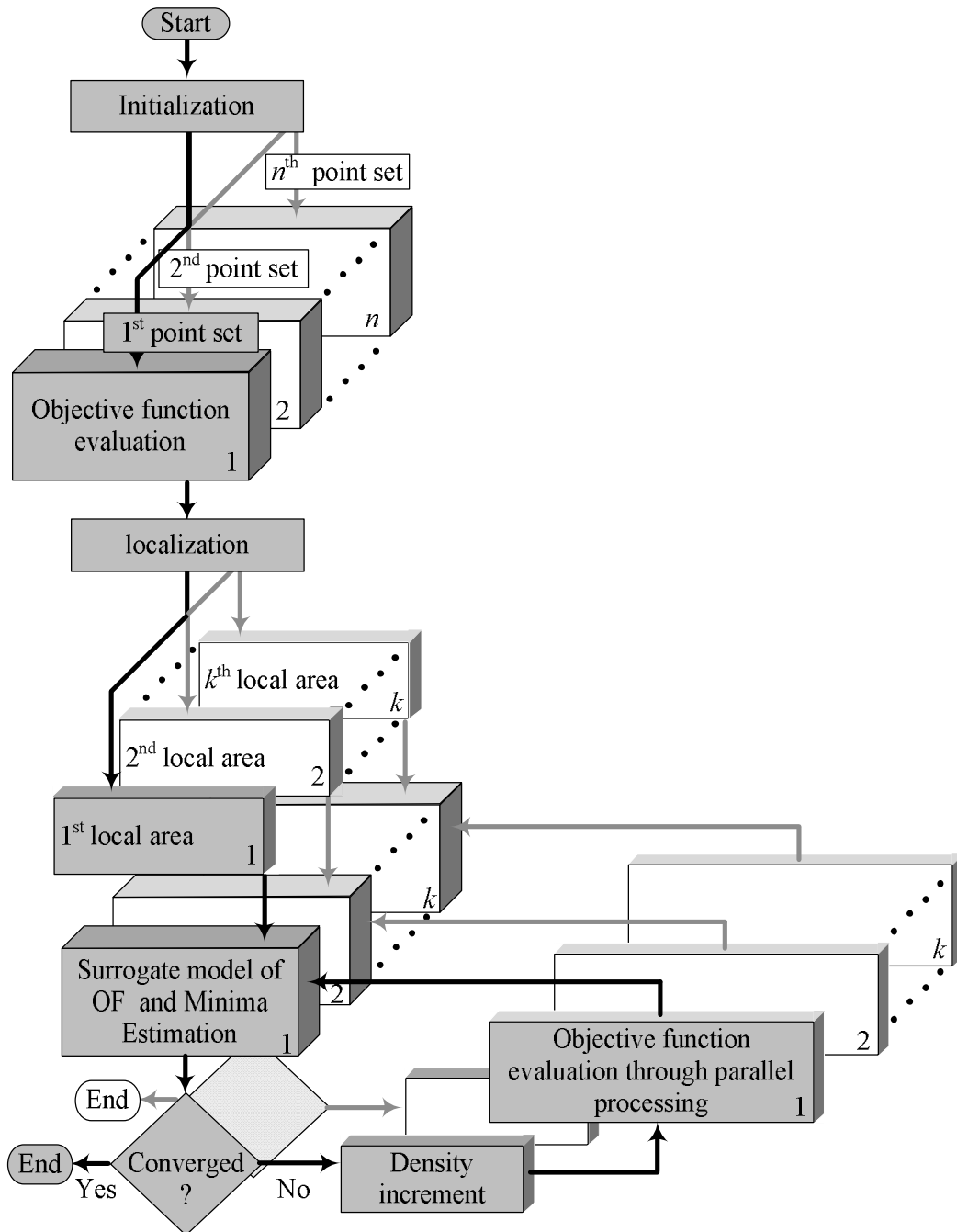


Figure 4-1: Flowchart of the proposed parallel multi-modal optimization algorithm.

For example if 60 processors are available, up to 60 simulations can be carried out simultaneously. Considering the example in Section 3.1, the algorithm must conduct 110 EMT simulations in the initialization stage to obtain the value of the objective function

for each sampling point in the decision area. These simulations can be readily conducted in parallel as it is shown in Figure 4-1. Therefore the number of required iterations (n_i) for the 110 sampling points drops from 110 to 2 according to the following computation.

$$n_i = \left\lfloor \frac{n_s}{n} \right\rfloor + 1 = \left\lfloor \frac{110}{60} \right\rfloor + 1 = 2 \quad (4-1)$$

where, $\lfloor \cdot \rfloor$ is the floor operation, and n_s and n are the number of sampling points and processors, respectively.

Note that the processor utilization is 100 percent (all 60 processors are used) in the first iteration while it is about 83 percent (50 processors are used) in the second iteration. This indicates that the availability of 60 cores in this example does not result in a speed-up factor of 60. However, parallelism reduces the total required time for evaluation of the objective functions potentially by a factor of n [60].

Once the objective function values are obtained for each point in the initial mesh, regions where local optima may exist are identified through the localization stage of algorithm. As described in Section 3.1.2, local optima are discovered by comparing the OF value of every point in the mesh to the OF values of its neighboring points, and recording which points show the smallest OF than all their neighbors.

The proposed algorithm takes advantage of independent computations of the localized areas to further parallelize its ensuing steps. The OpenMP parallelization for FORTRAN program is used for implementation. Further investigation of each localized area is assigned to a number of parallel computing machines as shown in Figure 4-1. The algorithm defines the number of simulations for each local area in order to launch all the

available processors. The maximum number of available processors for each local area (n_{plo}) can be calculated as follows:

$$n_m = \text{mod}\left(\frac{n}{k}\right)$$

$$n_{plo}(j) = \begin{cases} \left\lfloor \frac{n}{k} \right\rfloor & \text{if } j > n_m \\ \left\lfloor \frac{n}{k} \right\rfloor + 1 & \text{if } j \leq n_m \end{cases} \quad j = 1, 2, 3, \dots, k \quad (4-2)$$

where n and k are the total number of available processors and the number of discovered local areas, respectively. Considering the example in Section 3.1 where $k=2$ and $n=60$ the number of available processors for each local area can be calculated as follows:

$$n_m = \text{mod}\left(\frac{n}{k}\right) = \text{mod}\left(\frac{60}{2}\right) = 0$$

$$n_t(j) = \left\lfloor \frac{n}{k} \right\rfloor = \left\lfloor \frac{60}{2} \right\rfloor = 30 \quad j = 1, 2$$

The next stages of the algorithm serve to (i) produce increasingly accurate estimations of the location of local minima, and (ii) provide a surrogate model of the objective function around its local minima to enable sensitivity analysis for the local areas with no requirements of any further EMT simulations. Convergence criteria are similar to what were described for the sequential version of the algorithm in Chapter 3.

4.2 Numerical Results and Comparative Assessment

This section presents two implementations of the proposed algorithm. In the first implementation, the algorithm is executed sequentially on a single processor, as was described in Chapter 3. In the second one, parallelism of the algorithm is exploited on a 60-processor parallel computing platform. The considered objective function is the one introduced in Section 3.11.1. This function is chosen as a controlled case because the number of local minima can be selected via the parameters k_1 and k_2 in the specified range. Table 3-8 shows a few combinations of k_1 and k_2 along with the number of resulting local minima.

Table 4-1 shows the results of the comparative assessment of the sequential and parallel implementations of the algorithm. Note that all the 60 processors are used by the parallel algorithm in the problems shown in the Table 4-1. This is because the number of objective function evaluations for all of them is larger than 60. The parallel algorithm consistently outperforms the sequential algorithm in computation time. As problems with more minima are computationally more complex, the speed-up due to the parallel implementation (10.8 times faster for the problem with 500 minima) is more significant.

Table 4-1: Comparative study results.

No. of actual minima	No. of obtained minima	No. of OF evaluations	Computation elapse (min)		Speed-up (S/P)
			Sequential (S)	Parallel (P)	
16	16	337	00:13	00:11	1.2
20	20	441	00:15	00:12	1.3
50	49	1476	01:41	00:24	4.2
100	99	2276	02:48	00:30	5.6
200	197	8676	13:30	01:33	8.7
500	495	23625	46:08	04:16	10.8

4.3 Memory Usage

As it was discussed in Section 3.1.1 complex optimization problems with a large number of sampling points may require more resources than what is available, and memory-related problems may occur. To prevent such problems a limit (i.e., n_{slim}) should be defined for the number of sampling points (n_s). The memory usage of the initial sampling points can be calculated as a function of the number of decision variables n_v and samples for each variable n_a . Considering that the optimization algorithm is developed in FORTRAN and the data type is selected as *Real* and that each digit requires 64 bits of memory, the memory usage can be calculated as follows:

$$M_u = 64\text{bit} \times (n_v + 1) \times n_s \quad (4-3)$$

Using (4-3), an approximate evaluation of memory usage (M_u) is shown in Table 4-2 as a function of n_v and n_a .

Table 4-2: Required memory as a function of n_v and n_a

M_u		n_v									
		1	2	3	4	5	6	7	8	9	10
n_a	5	0.1563 Kbit	1.1719 Kbit	7.8125 Kbit	0.0477 Mbit	0.2861 Mbit	1.6689 Mbit	0.0093 Gbit	0.0524 Gbit	0.291 Gbit	1.6007 Gbit
	6	0.1875 Kbit	1.6875 Kbit	0.0132 Mbit	0.0989 Mbit	0.7119 Mbit	4.9834 Mbit	0.0334 Gbit	0.2253 Gbit	1.5017 Gbit	0.0097 Tbit
	7	0.2188 Kbit	2.2969 Kbit	0.0209 Mbit	0.1832 Mbit	1.5387 Mbit	0.0123 Gbit	0.0982 Gbit	0.7731 Gbit	6.0132 Gbit	0.0452 Tbit
	8	0.25 Kbit	3 Kbit	0.0313 Mbit	0.3125 Mbit	3 Mbit	0.0273 Gbit	0.25 Gbit	2.25 Gbit	0.0195 Tbit	0.1719 Tbit
	9	0.2813 Kbit	3.7969 Kbit	0.0445 Mbit	0.5006 Mbit	5.4061 Mbit	0.0554 Gbit	0.5702 Gbit	5.773 Gbit	0.0564 Tbit	0.5581 Tbit
	10	0.3125 Kbit	4.6875 Kbit	0.061 Mbit	0.7629 Mbit	0.0089 Gbit	0.1043 Gbit	1.1921 Gbit	0.0131 Tbit	0.1455 Tbit	1.6007 Tbit

According to (3-3) and (4-3) the amount of memory usage (M_u) for $n_v=10$ and $n_a=9$ can be calculated as follows:

$$n_s = \prod_{i=1}^{10} (9)_i = 3.4868 \times 10^9 \quad (4-4)$$

$$M_u = 64\text{bit} \times (n_v + 1) \times n_s = 0.5581 \text{ Tb}$$

This amount of demand can easily run a machine (with $M_{lim} = 64\text{Gb}$ RAM) out of memory and cause memory crash problems. To avoid this problem, the decision area may be divided into smaller regions (sub-areas) and the algorithm may focus on one sub-area at a time. For each sub-area the number of sampling point, n_{slim} , can be calculated as follows:

$$n_{slim} = \frac{M_{lim}}{64\text{bit} \times (1 + n_v)} \quad (4-5)$$

where M_{lim} is the memory limit.

For example the number of sample points in each sub-area for the above example can be calculated as:

$$n_{slim} = \frac{64Gb}{64bit \times (1+10)} = 9.7613 \times 10^7$$

Therefore, the number of sub-areas can be calculated as follows:

$$n_{sub} = \left\lceil \frac{n_s}{n_{slim}} \right\rceil + 1 \quad (4-6)$$

For the current example this value is:

$$n_{sub} = \left\lceil \frac{3.4868 \times 10^9}{9.761 \times 10^7} \right\rceil + 1 = 36$$

Based on the value obtained for the n_{sub} the algorithm splits the decision area into sub-areas and focuses on one sub-area at a time. In this technique, the number of initial sampling points is maintained below the limit to avoid memory crash while the number remains high enough to uphold the resolution of sampling points in each sub-area. However the methodology for finding local areas in the decision space should be modified to preserve accuracy.

The principles of how the algorithm can divide the decision space and how it can find the actual local areas are described considering the example provided in Section 3.1. In this example the number of sampling points is 110 which is obviously too small to cause memory overflow in practice. However, here for the purposes of simplicity and visualization it is assumed that this number of points is large enough to cause such a memory problem. This study will be verified in Section 5.3 where a complex optimization problem will be demonstrated.

Assuming it is required to divide the decision area into 3 sub-areas ($n_{sub}=3$), Figure 4-2 shows such division. The result of searching for the local areas in sub-area I is shown in Figure 4-3 (a). The local minimum and the neighboring points are shown by red and blue circles, respectively, for the discovered local area. The algorithm saves the discovered local area and continues to the next sub-area (i.e. sub-area II). The algorithm finds two local areas in sub-area II as it is shown in Figure 4-3(b). As long as the two current local areas are on the edges of sub-area II the algorithm verifies if the current discovered local areas and the former ones are neighbors. It combines the local areas that have a local minimum in common to correct the vicinity for the local minimum.

The algorithm then continues its search to the third sub-area (i.e. sub-area III) and compares the proximity of the local area in sub-area III with the ones in sub-area II. As long as the points in the local areas of sub-area III and sub-area II are neighbors the algorithm compares the minimum value of the two local areas and removes the one with a higher value of local minimum. The principles of searching for the actual local areas in a divided decision area are described as follows:

- i. Combine the local areas that have a local minimum in common to form the actual local area.
- ii. Compare the minimum point of the local areas with common and/or neighboring points. Select the local area with a smaller minimum and leave out the rest.

This technique will be implemented for a complex power system design presented in Section 5.3.

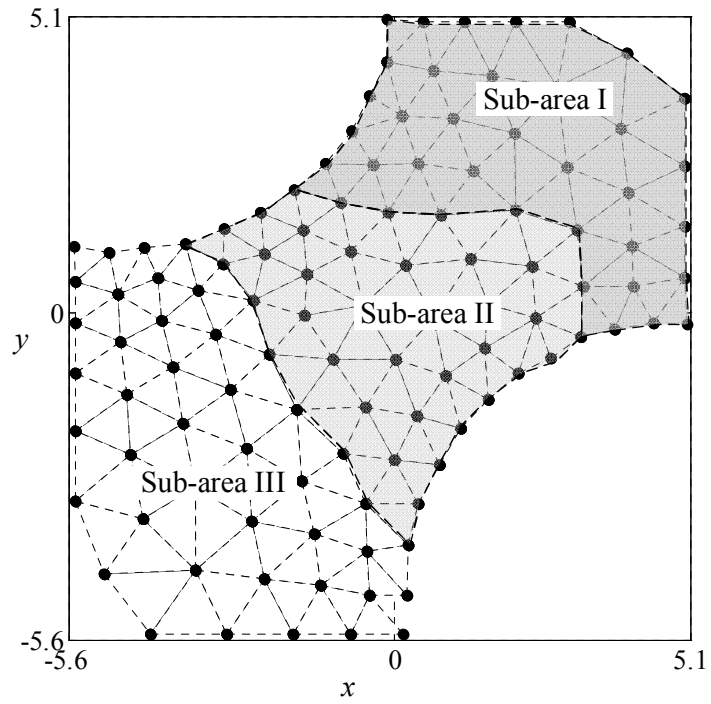


Figure 4-2: Division of the decision area into three sub-areas in order to reduce the number of sampling points and avoid memory crash.

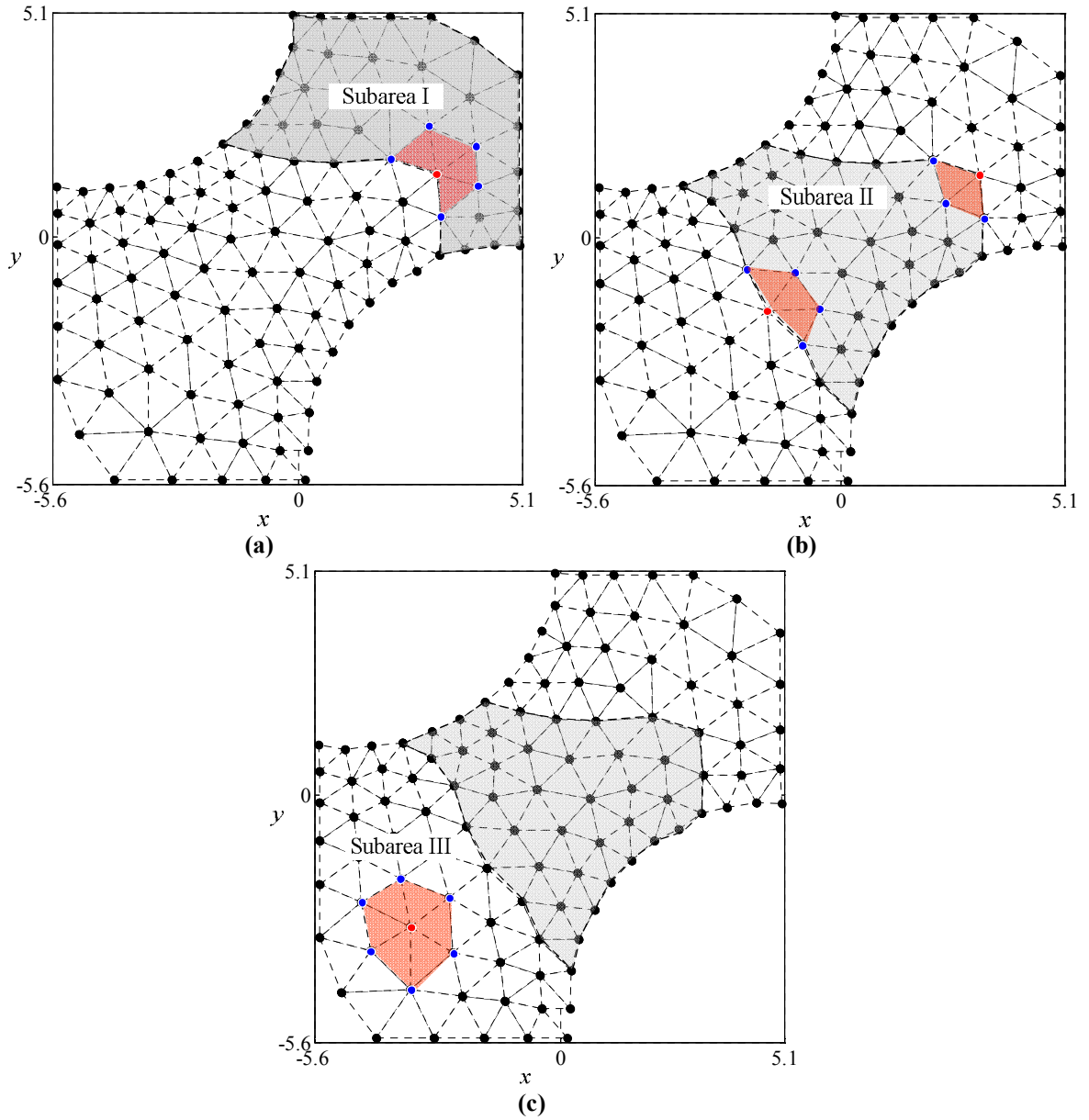


Figure 4-3: Discovered Local areas in each sub-area: (a) I (b) II, (c) III.

4.4 Conclusions and Discussion

A parallel version of the multi-modal optimization algorithm was proposed in this chapter to address simulation-based optimization of computationally expensive black-box

problems. Additionally, in case of a large number of variables a method was proposed to avoid memory overflow, although this method may reduce the speed of the algorithm to some extent.

Comparative assessment of the sequential and parallel implementations of the algorithm was given and it was shown that the parallel version can outperform the sequential one in terms of computation speed (10.8 times faster in the given example with 500 minima). The optimization examples in this chapter were based upon simple algebraic equations for explanatory and analysis purposes. Chapter 5 involves complex black-box power system designs where a number of decision variables - much larger than the 2-variable problem presented earlier in this chapter - are considered for optimization. Additionally, objective function evaluations for the power system designs must be obtained by a detailed EMT simulator, which is much more expensive computationally than the simple mathematical formulae presented in this chapter. Therefore, the specialized version of the proposed optimization algorithm, presented in this chapter, will be used in Chapter 5 where techniques such as parallel computation and memory reduction are implemented to discover multiple local optimal solutions for complex black-box systems within an acceptable time frame and memory usage.

Chapter 5

Power Electronic Design Examples

This chapter provides examples to evaluate the capabilities of the proposed algorithm in solving multi-modal nonlinear black-box objective functions for complex power system designs. These examples are computationally expensive and may require unaffordable time and/or memory usage if conventional design techniques are deployed.

The examples begin with a complex speed-control drive system design for an induction motor (IM) [23]. A step by step implementation of the optimization algorithm will be demonstrated, in which it will be shown how the goals of a design can be formulated into an aggregate objective function and how the algorithm can use this function to tune the controllers. Also it will be shown how sensitivity analysis can be conducted to obtain the most robust optimal design [60]. Local surrogate models of the objective function will be used during sensitivity analysis to avoid expensive simulation runs.

In the second example the advantages of the parallel algorithm over the sequential one are demonstrated. This study confirms that for more complex objective functions the

proposed parallel algorithm can achieve higher speed-up than what was shown in Chapter 4. In this example the controllers of a high-voltage direct-current (HVDC) system based upon voltage-source converters (VSCs) are optimized [60]. This example shows a system that is particularly problematic to tune. In addition, single-variable sensitivity analysis will be carried out for a range of detuning to examine the effects of each variable on the objective function.

In the last example in this chapter, techniques such as parallel computation and memory reduction are used to design the controllers of a three terminal HVDC system to obtain multiple local optimal solutions in an acceptable time frame and with moderate memory usage [61]. It will also be shown that much computation time can be saved during sensitivity analysis by using the local surrogate models of the objective function instead of expensive simulations of the actual power system.

5.1 Design Example 1: Vector-Controlled Induction Motor Drive

This section is devoted to demonstration of the procedure of an optimal design of a highly complex black-box system where a detailed model of a complex induction motor drive system is provided in an EMT simulator. The objective function for optimal design can be evaluated via coupling the optimization algorithm with the EMT simulator. Also principles of sensitivity assessments of the discovered local optimal solutions will be

presented in this section using surrogate models of the objective function provided by the algorithm.

5.1.1 Indirect Vector-Controlled Induction Motor Drive System

An indirect vector-controlled induction motor drive system is considered as a case study in this section. The schematic diagram of the power electronic circuit, indirect vector control system, and induction machine are shown in Figure 5-1. The input stage is a three-phase diode bridge that converts a three-phase ac source voltage to a dc voltage with considerable ripple. Therefore, dc link filtering is required to obtain a smooth dc voltage (V_{dc}) at the terminal of the voltage source converter (VSC). Filtering is obtained by combination of an inductor, L_{dc} and a capacitor, C_{dc} as it is shown in Figure 5-1. The VSC delivers the desired voltages to the terminal of the induction machine based on the gate pulses provided by the indirect vector controller. The inputs of the controller are the reference (ω_{ref}) and measured (ω) speeds of the induction machine rotor. In the indirect vector control scheme the position of the synchronous reference frame is determined based upon the values of the d- and q-axis components of the rotor flux. This orientation leads to independent controllability of the flux linkage and shaft torque and hence simplifies the induction machine speed control.

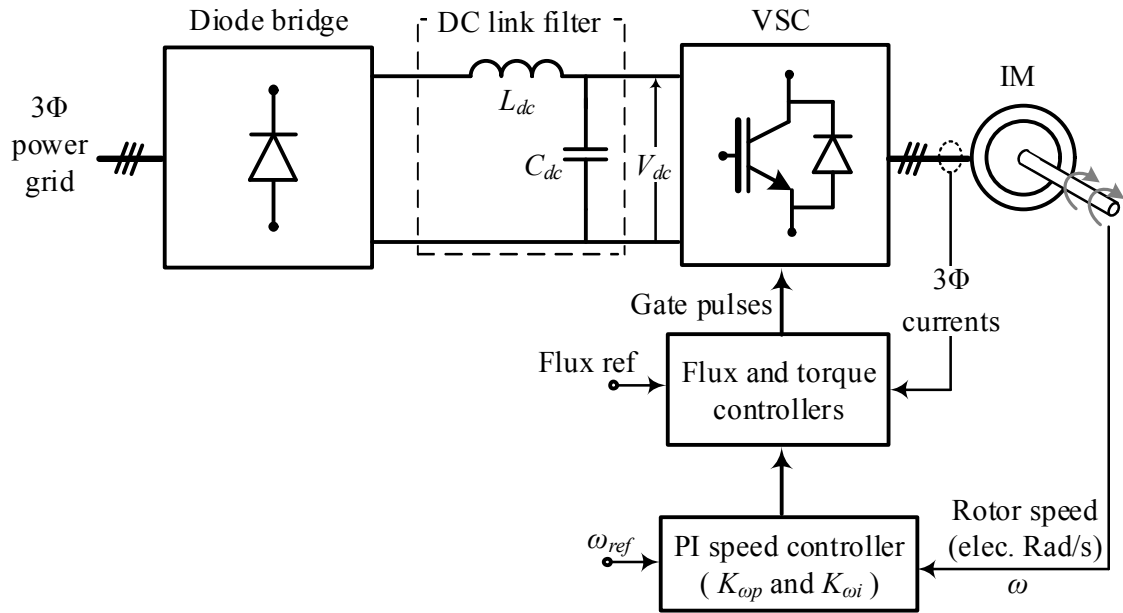


Figure 5-1: Power electronic drive system including VSC, indirect vector controller and IM.

The ratings and parameter values of the 3-phase 4-pole, 60-Hz induction machine are specified in Table 5-1.

Table 5-1: Induction machine ratings and parameters.

Machine Rating			T_B	I_B	r_s	X_{ls}	X_M	X_{lr}	r_r	J
hp	volts	rpm	[N.m]	[amps]	[ohms]	[ohms]	[ohms]	[ohms]	[ohms]	[kg.m ²]
500	2300	1773	1.98×10^3	93.6	0.262	1.206	54.02	1.206	0.187	11.06

The controllers for the indirect field drive system are shown in Figure 5-2.

References [21] and [23] provide more information regarding the controllers.

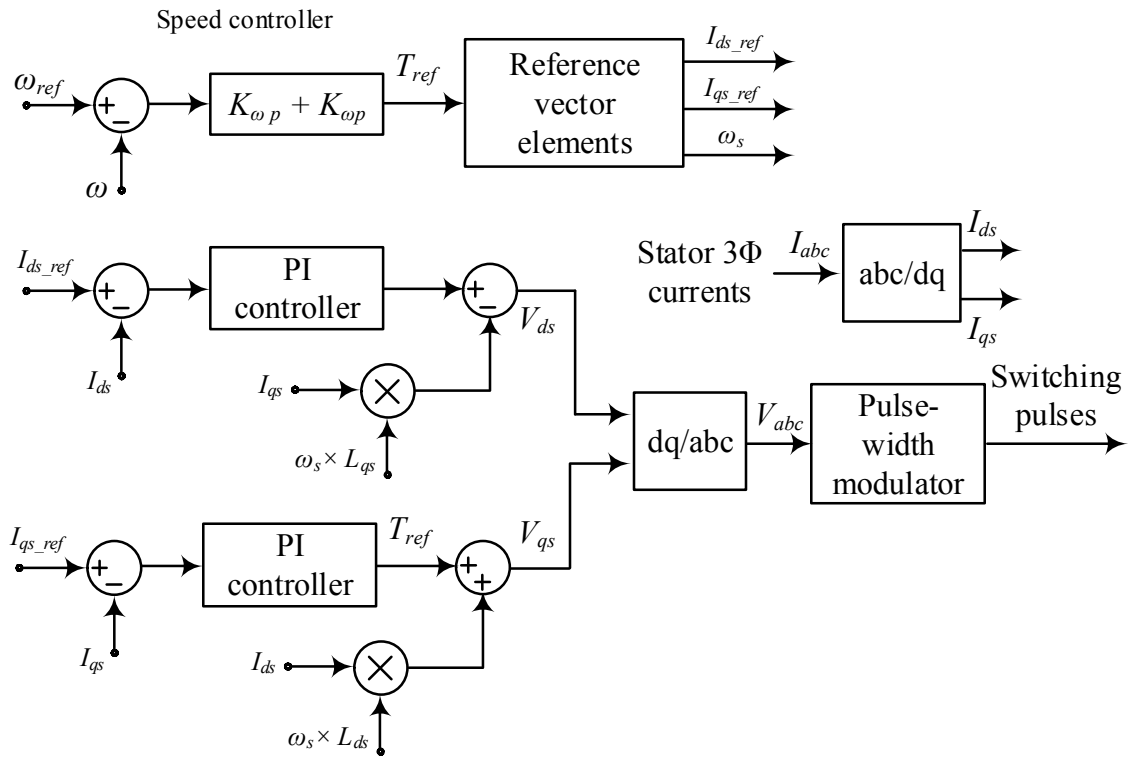


Figure 5-2: Block diagram of the controllers for the drive system.

5.1.2 Aggregate Objective Function and Boundaries

Identifying what the design is to accomplish, four objectives for the induction machine drive system are adapted from [23] and listed as follows:

- i. Minimizing the difference between the rotor speed and its reference: this is the main objective of optimization. Therefore, the controller should be tuned to obtain accurate speed control and reduce the error between the rotor speed and its reference.

- ii. Minimizing the ripple of the electromagnetic torque: the mechanical stress on the shaft of the machine will greatly decrease if the electromagnetic torque ripple can be reduced.
- iii. Minimizing the harmonic content of the dc link voltage: better voltage conversion can be provided by VSC if the harmonic content of the dc link voltage is reduced.
- iv. Minimizing the size of inductor and capacitor of dc link filter: the weight and cost of the drive system can be reduced if smaller sizes for these elements are chosen.

It is also possible to add more design objectives to the above list; however, the stated objectives are adequate for the IM drive system to perform acceptably during transient and steady state conditions. To minimize these objectives, it is necessary to evaluate them numerically during optimization. An approach is proposed in [23] where the stated objectives can be converted into a set of mathematical formulae as follows [23]:

$$\begin{aligned}
 f_1(\mathbf{x}) &= k_1 \int_{\text{SteadyState}} (\omega_{ref}(t) - \omega(t))^2 dt + k_2 \int_{\text{Transient}} (\omega_{ref}(t) - \omega(t))^2 dt \\
 f_2(\mathbf{x}) &= k_3 \int_{\text{SteadyState}} T_{e\text{-ripple}}^2 dt \\
 f_3(\mathbf{x}) &= k_4 \int_{\text{SteadyState}} v_{dc\text{-ripple}}^2 dt \\
 f_4(\mathbf{x}) &= k_5 C_{dc} + k_6 L_{dc}
 \end{aligned} \tag{5-1}$$

where \mathbf{x} is the decision variable and k_1 to k_6 are the weight coefficients that are presented in Table 5-2 .

Table 5-2: Coefficient values for the objective functions.

Coefficient	k_1	k_2	k_3	k_4	k_5	k_6
Value	100	50	1000	2000	10000	5000

Note that, minimizing each objective shown in (5-1) improves a certain property of the drive system. For instance, if f_1 is minimized, the instantaneous rotor speed ($\omega(t)$) of the induction machine will be optimally close to its reference value ($\omega_{ref}(t)$). All the objectives stated in (5-1) are added up to form the aggregate objective function shown as follows:

$$\text{OF}(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) + f_3(\mathbf{x}) + f_4(\mathbf{x}) \quad (5-2)$$

Note that minimization of $\text{OF}(\mathbf{x})$ does not necessarily lead to exact minimization of each individual objective function in (5-2). This is because the $\text{OF}(\mathbf{x})$ is an aggregate objective function and each individual objective function only partly contributes to it. As such selection of the coefficient values in Table 5-2 becomes critical to ensure that each objective is suitably represented in $\text{OF}(\mathbf{x})$. Improper selection of coefficients may give an unfair advantage to one or a few objectives over the rest.

The variables that play significant roles in the performance of the drive system must be selected as the decision variables, i.e., elements of vector \mathbf{x} in the optimization problem defined in (5-1). The speed controller in the drive system adjusts the shaft speed of the induction machine, and its proportional and integral gains, i.e. K_{op} and K_{oi} , are selected as the decision variables. Besides, the dc link inductor (L_{dc}) and capacitor (C_{dc}) are also considered as the other two decision variables because these elements provide

damping to the ripples of the dc voltage (see (5-1)). Therefore, in this particular optimal design the decision variables can be defined in vector form as $\mathbf{x} = [K_{ap}, K_{oi}, L_{dc}, C_{dc}]$.

An EMT model of the induction machine drive system is built in PSCAD/EMTDC simulator. The numerical results of the simulator including the value of the aggregate objective function (see (5-2)) at the end of each simulation are transmitted to the optimization algorithm to select variable values for the next run.

During each simulation a number of dynamic variations are applied to the inputs (i.e., ω_{ref} and T_L) to excite the controller and obtain its dynamic response. Figure 5-3 shows how the speed reference (ω_{ref}) and load torque (T_L) are varied to provide a range of dynamics for the drive system during simulation.

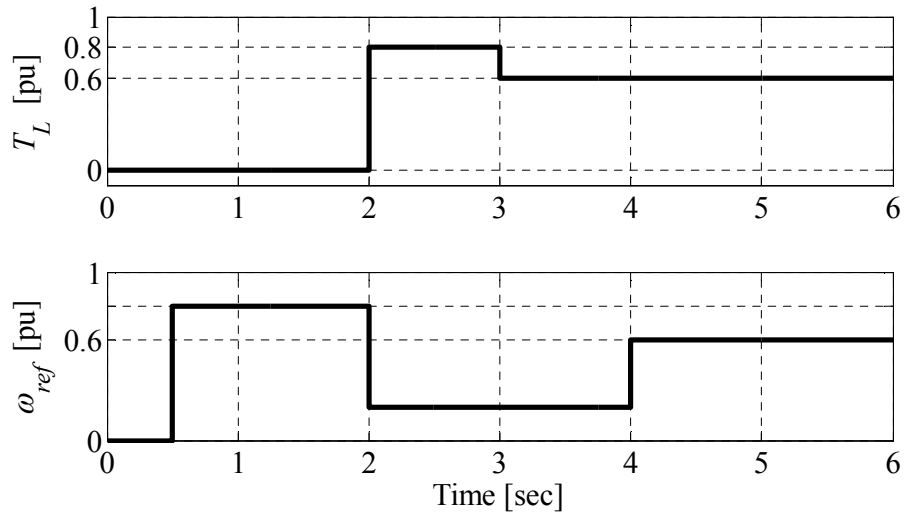


Figure 5-3: Variations in speed and load torque orders.

Also a decision space is required to be defined to limit the search area of the algorithm preceding the optimization process. The objective function in (5-2) is subjected to the

boundaries shown in Table 5-3. The limits for C_{dc} and L_{dc} are defined such that these variables fall into an acceptable range; such a range may come into effect due to such factors as the available component sizes, physical limitations of the circuit, and component cost brackets. For the variables K_{op} and K_{oi} the ranges are selected based on expert knowledge of the PI controller in the drive system.

Table 5-3: Boundaries of decision variables.

	K_{op}	K_{oi}	L_{dc} (mH)	C_{dc} (μ F)
Minimum	100	0.1	0.1	100
Maximum	600	10	1	1000

The optimization procedure starts with initialization in which a number of sample points are generated and scattered according to the decision space defined in Table 5-3 (see Section 3.1.1 for details). Then the algorithm selects one or a few points - depending on the number of available processors (see Section 4.1) from decision area to evaluate their objective function using simulation. At the end of each simulation run, the values of the objective functions f_1 to f_4 shown in (5-1) are gathered and then added up to obtain the aggregated objective function value shown in (5-2). This OF value is saved in the algorithm and the values of the decision variables are renewed according to the new sample points from a new part of the decision space. Doing so the OF values of all the points in the decision area are obtained and then the algorithm searches for local areas as described in Section 3.1.2.

After initial localization, the local areas are searched in greater detail by the algorithm to find more accurate estimation of the local optima \mathbf{x}^* that minimize the aggregated objective function. The objective function is estimated locally for each local optimum

during the course of optimization to obtain a surrogate model for each local area. These stages were addressed in Section 3.1.4, 3.1.5 and 3.1.6 respectively. Once the local optimal solutions \mathbf{x}^* are gathered then the sensitivity analysis may be applied to find the most robust solution.

The dynamic responses of the electromagnetic torque, speed and dc voltage for a non-optimal setting of decision variables $\mathbf{x} = [100, 0.1, 0.1\text{mH}, 100\mu\text{F}]$ are shown in Figure 5-4, where the objective function value is 1481. These variable settings are obtained empirically using simulations and expert tuning of the controller parameters via trial and error to obtain a stable, although not completely desirable dynamic response.

Note that before $t = 2\text{sec}$ the electromagnetic torque undergoes large fluctuations that cause unacceptable ripple on the machine shaft. The dynamics of the torque, speed and dc voltage are poor in terms of settling time and magnitude, and need to be improved.

The multi-modal optimization algorithm converges to six local optimal solutions for this problem as listed in Table 5-4. Compared to the pre-optimization OF value of 1481, there are substantial reductions in the obtained local optimal solutions, among which solution \mathbf{x}_6^* is the global minimum with an objective function value of 100.9.

Table 5-4: Local optimal solutions.

Local minimum	PI controller		DC link filter		OF(\mathbf{x}^*)
	K_{op}^*	K_{oi}^*	L_{dc}^* (mH)	C_{dc}^* (μF)	
\mathbf{x}_1^*	376	0.47	0.48	529	105.9
\mathbf{x}_2^*	220	5.40	0.13	601	101.1
\mathbf{x}_3^*	392	1.20	0.50	550	105.2
\mathbf{x}_4^*	502	0.20	0.23	491	105.5
\mathbf{x}_5^*	201	3.80	0.26	802	108.3
\mathbf{x}_6^*	167	2.5	0.37	590	100.9

The optimal plots of the system response for the global minimum \mathbf{x}_6^* are shown in Figure 5-5. The performance is highly improved using the proposed optimization algorithm. As can be seen the ripple of the electromagnetic torque has been reduced and consequently the mechanical stress on the shaft of the machine will greatly decrease. The speed has faster response with better settling time as its reference changes. The dc voltage ripple in the optimized system has been reduced significantly, which leads to a better voltage conversion performance for VSC.

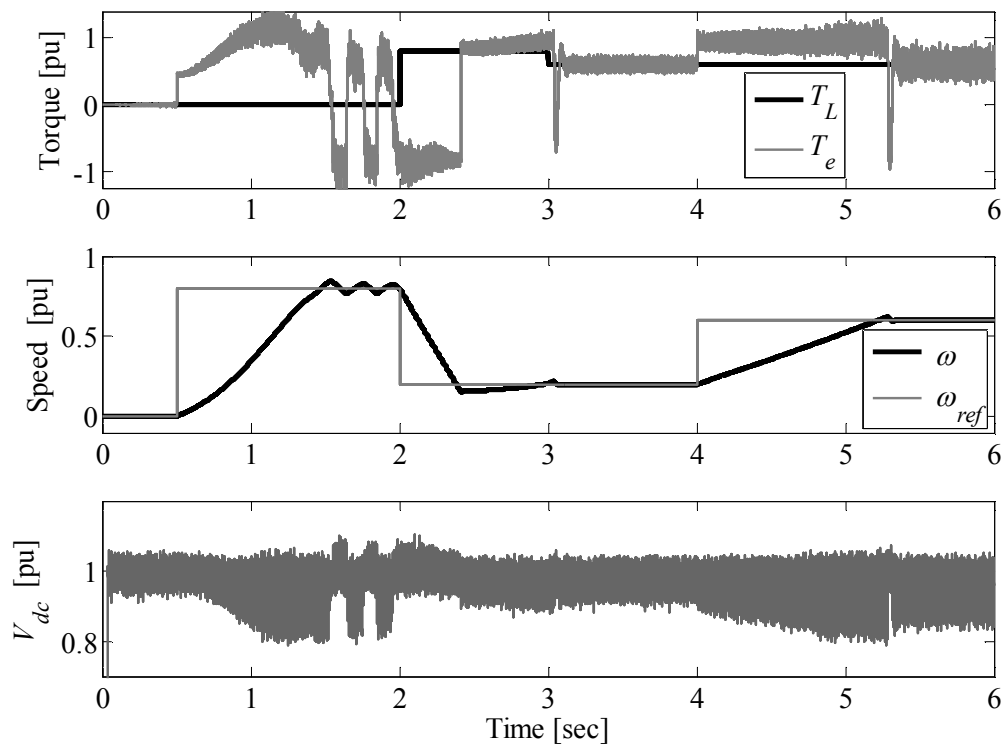


Figure 5-4: Non-optimal dynamic responses. (a) electromagnetic torque (b) shaft speed (c) dc voltage.

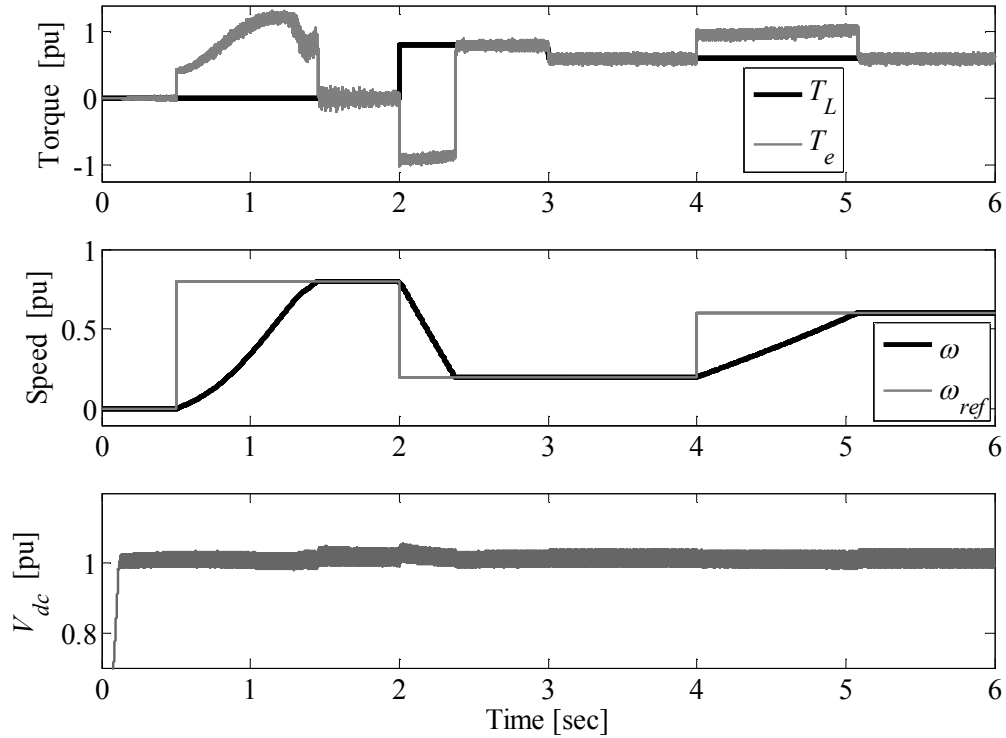


Figure 5-5: Optimal dynamic responses. (a) electromagnetic torque (b) shaft speed (c) dc voltage.

Table 5-5 summarizes the specifics of the algorithm’s performance. The initial grid requires 5661 objective function evaluations, which leads to the discovery of six areas as potential local minima. The total number of objective function evaluation to obtain the six local minima is 29889 for this simulation-based optimization design, which required 83 hours of simulation time.

Table 5-5: Optimization specifics

Number of objective function evaluations to establish initial grid	5661
Number of discovered local areas	6
Number of local objective function evaluations per local area	1296
Total number of objective function evaluations	29889
Total simulation time using sequential version (hours)	83

5.1.3 Sensitivity Assessment

For a robust optimal design, the performance must be reasonably invariant to the perturbations in decision variables. Although local minimum \mathbf{x}_6^* in Table 5-4 is the global minimum, it may not be suitable if it is excessively sensitive to the optimized variable values; another local minimum with lower sensitivity may be preferable. Hence, an analysis is conducted to determine how sensitive the objective function at a particular local minimum is to perturbations in the optimized variables. To analyze the sensitivity of the minimum solutions (\mathbf{x}^*) against deviations a detuning of $\Delta = \pm 10\%$ is applied. The detuned local optimal solutions \mathbf{x}_d^* are shown as follows:

$$\begin{aligned} \Delta \mathbf{x} &= 0.1 \times \mathbf{x} \\ \mathbf{x}_d &= \mathbf{x}^* \pm \Delta \mathbf{x} \\ \mathbf{x}_d &= \{K_{op}^* \pm \Delta K_{op}, K_{oi}^* \pm \Delta K_{oi}, L_{dc}^* \pm \Delta L_{dc}, C_{dc}^* \pm \Delta C_{dc}\} \end{aligned} \quad (5-3)$$

The number of detuned solutions (n_{xd}^*) associated with each local optimum, shown in Table 5-4, can be calculated as follows:

$$n_{xd}^* = \prod_{i=1}^{n_v} (2)_i = \prod_{i=1}^4 (2)_i = 16 \quad (5-4)$$

This formula is obtained based on the fact that each minimum solution consists of four variables ($n_v = 4$) and each variable detuned from its nominal value in two directions ($\pm 10\%$). Therefore, for each local minimum sixteen simulations are required for a total of 96 ($16 \times 6 = 96$) simulations to complete sensitivity analysis for all the local optima. However, sensitivity assessment is conducted with no simulations because the surrogate

models of the objective function, derived around the local minima during the course of optimization, can directly be consulted for this purpose.

Using these surrogate models one can estimate how much the objective function value of each local minimum (i.e. $OF(\mathbf{x}^*)$) would deviate from its value at the detuned solutions (i.e. $OF(\mathbf{x}_d)$). The results of sensitivity analysis are shown in Figure 5-6 where the variations of the objective function for all six local areas versus the sixteen detuning positions (\mathbf{x}_d) are represented. To obtain the overall sensitivity assessment of the minimum solutions to the detuning, the objective function variations (ΔOF) for each local minimum are summed up and averaged as shown in Figure 5-7.

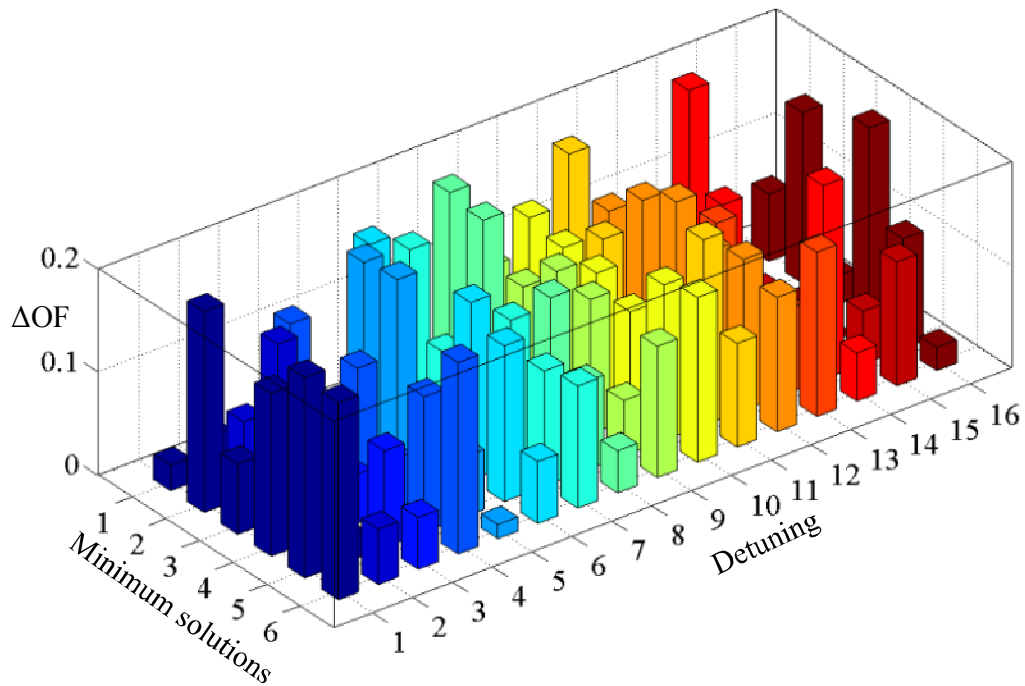


Figure 5-6: Objective function variations (ΔOF) versus detuning in minimum solutions.

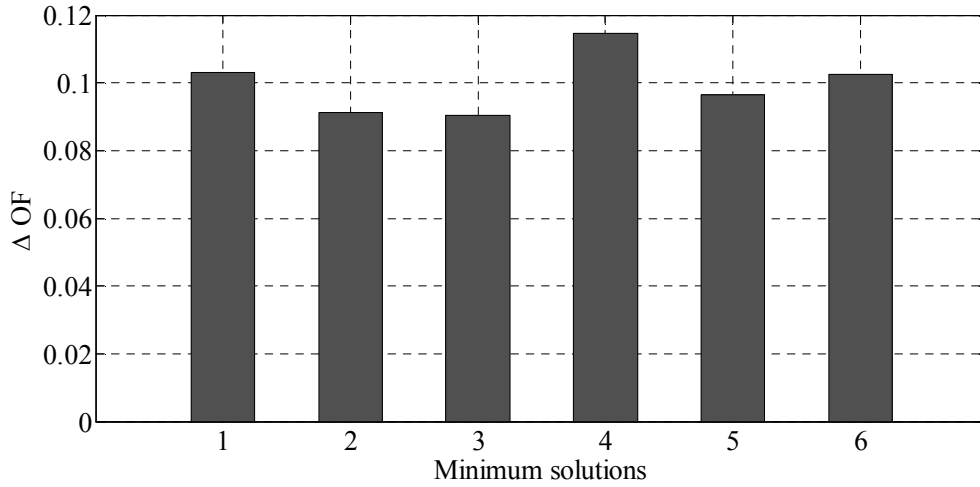


Figure 5-7: Sensitivity assessment of the minimum solutions.

As it is shown in Figure 5-7, the local minimum \mathbf{x}_3^* shows the smallest average variation to the variable's perturbations compared to the others. Although the local minimum \mathbf{x}_6^* is the global minimum, it may be reasonable to select local minimum \mathbf{x}_3^* as the final setting for the induction machine drive system to obtain the most robust design against perturbations.

5.2 Design Example 2: VSC-HVDC Transmission System

This section shows the advantages of the parallel implementation of the proposed algorithm by comparing it with its sequential version. The performance of the parallel algorithm will be investigated in terms of computational speed-up.

The power system example selected is a VSC-HVDC system where parameters of the PI controllers on both sides of the HVDC line are taken as decision variables to be

optimized [60]. The PI controllers are sensitive to each other's operation and difficult to tune. Also sensitivity of the system performance to the variations of the optimal variables is investigated using a specific sensitivity analyser where continuous variations of a certain parameter are considered for a range from 0 to 5 percent.

5.2.1 VSC-HVDC Transmission System

The VSC-HVDC system, as shown in Figure 5-8, consists of two converters (VSC1 and VSC2, both modeled in detail in PSCAD/EMTDC) at the ends of bipolar dc cables. VSC1 controls the ac voltage magnitude (V_{c1}) and dc voltage (V_{dc1}) and VSC2 controls its real power (P_{dc2}) and the ac voltage magnitude (V_{c2}).

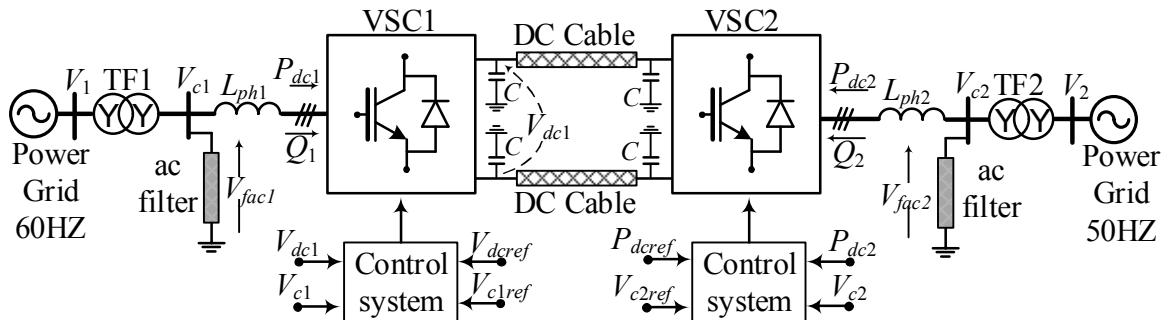


Figure 5-8: VSC-HVDC system.

The block diagrams of the controllers for VSC1 and VSC2 are shown in Figure 5-9 and Figure 5-10, respectively.

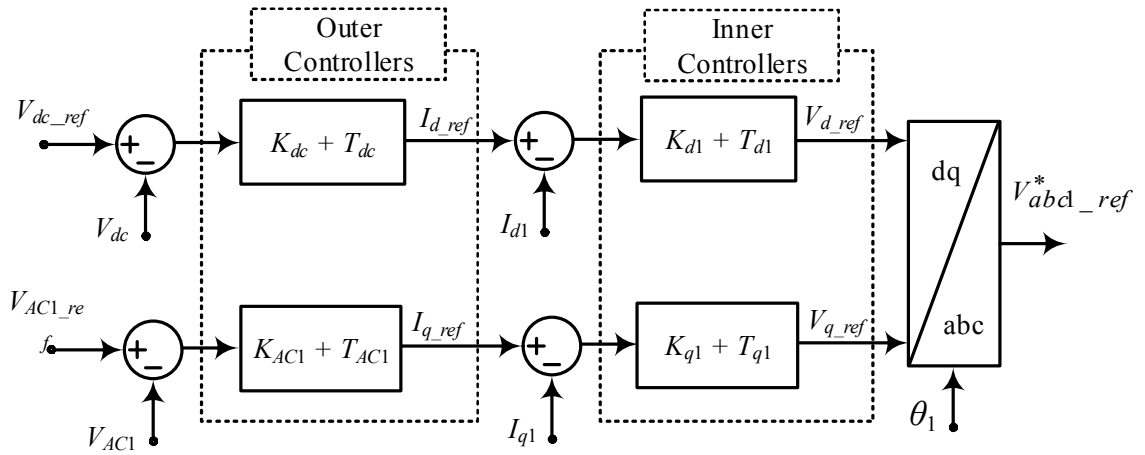


Figure 5-9: Block diagram of VSC1 controllers.

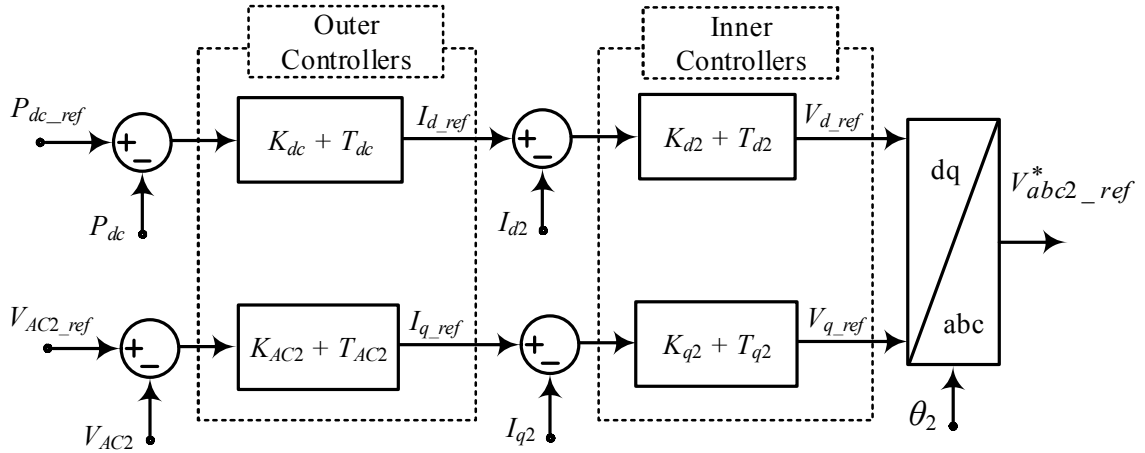


Figure 5-10: Block diagram of VSC2 controllers.

The specifications of the electrical system are given in Table 5-6.

Table 5-6: HVDC power system parameters.

AC grids	$V_1 = V_2 = 420 \text{ kV}$, $L_{ph1} = L_{ph2} = 72.4 \text{ mH}$, $SCR1 = SCR2 = 2.6$, $f_1 = 60 \text{ Hz}$, $f_2 = 50 \text{ Hz}$
Transformers	Ratio1 = Ratio2 = 420/230 kV, $X_{TF1} = X_{TF2} = 10\%$, $S_1 = S_2 = 1.5 \text{ GVA}$
DC filter	$C = 300 \mu\text{F}$
DC rated voltage	$V_{dc} = 500 \text{ kV}$
DC rated power	$P_{dc} = 1000 \text{ MW}$
DC cable	$R_{dc} = 10.71 \text{ m}\Omega/\text{km}$, $L_{dc} = 0.364 \text{ mH}/\text{km}$, $C_{dc} = 0.168 \mu\text{F}/\text{km}$, Length = 200 km

5.2.2 Aggregate Objective Function and Boundaries

The three objectives of the design are stated as follows:

- i. To ensure that the dc voltage, V_{dc1} follows the reference, V_{dcref} as closely as possible during transient and steady state conditions;
- ii. To ensure that the real power, P_{dc2} follows the reference, P_{dcref} as closely as possible during transient and steady state conditions; and
- iii. To ensure that the terminal voltages of VSC1 (V_{c1}) and VSC2 (V_{c2}) follow the reference values as closely as possible during transient and steady state conditions.

The above three objectives can be captured by the following four weighted sub-objective integral square errors (ISE):

$$\begin{aligned}
 f_1(\mathbf{x}) &= C_{VT} \int_{Transient} (V_{dcref}(t) - V_{dc1}(t))^2 dt + C_{VS} \int_{SteadyState} (V_{dcref}(t) - V_{dc1}(t))^2 dt \\
 f_2(\mathbf{x}) &= C_{PT} \int_{Transient} (P_{dcref}(t) - P_{dc1}(t))^2 dt + C_{PS} \int_{SteadyState} (P_{dcref}(t) - P_{dc1}(t))^2 dt \\
 f_3(\mathbf{x}) &= C_{VcT} \int_{Transient} (V_{c1ref}(t) - V_{c1}(t))^2 dt + C_{VcS} \int_{SteadyState} (V_{c1ref}(t) - V_{c1}(t))^2 dt \\
 f_4(\mathbf{x}) &= C_{VcT} \int_{Transient} (V_{c2ref}(t) - V_{c2}(t))^2 dt + C_{VcS} \int_{SteadyState} (V_{c2ref}(t) - V_{c2}(t))^2 dt
 \end{aligned} \tag{5-5}$$

where C_{VT} , C_{VS} , C_{PT} , C_{PS} , C_{VcT} , and C_{VcS} are the weights as given in Table 5-7. Objectives f_1, f_2, f_3 and f_4 measure deviations between the dc link voltage and its reference value and the real power and its reference value and terminal ac voltages and their reference values, respectively. They include deviations during both the transient and steady state periods. Note that the closer a function is to zero, the more closely is the corresponding objective met.

Table 5-7: Coefficient values for objective functions in (5-5).

Coefficient	C_{VT}	C_{VS}	C_{PT}	C_{PS}	C_{VcT}	C_{VcS}
Value	0.05	0.002	0.005	0.002	1.0	0.5

The four objective functions in (5-5) are then combined into a single aggregate objective function as follows.

$$f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) + f_3(\mathbf{x}) + f_4(\mathbf{x}) \quad (5-6)$$

where $\mathbf{x} = [K_{AC1}, T_{AC1}, K_{AC2}, T_{AC2}, K_{dc}, T_{dc}]$ is the set of variables to be optimized; here the K_i denote a proportional gain and the T_i denotes an integral time-constant of the upstream PI controllers used. The three PI controllers to be optimized generate reference d- and q-axis current orders for lower level controllers. The d-axis current order for VSC2 is generated directly from the issued dc reference power and hence does not require a PI controller. Note that the weights used in the individual objectives in (5-6) determine their relative significance in the aggregate. As recommended in an earlier work [19], determination of the weights shown in [15] is done via trial and error to ensure that the sub-objectives in (5-2) have comparable shares in the formation of the final in (5-3). This ensures that the final solution will be satisfactory to all the sub-objectives.

The objective function in (5-5) is subjected to the boundaries shown in Table 5-8.

Table 5-8: Boundaries of decision variables.

	K_{AC1}	T_{AC1}	K_{AC2}	T_{AC2}	K_{dc}	T_{dc}
Minimum	5.0	1.0×10^{-5}	5.0	1×10^{-5}	5.0	1.0×10^{-5}
Maximum	100	1.0×10^{-1}	100	2.0	100	1.0×10^{-1}

5.2.3 Algorithm's Performance

Figure 5-11 shows variation orders that are applied to the dc power (P_{dcref}), the dc voltage (V_{dcref}) and grid voltages (V_1 and V_2) respectively. It is expected that the converter follows the step changes in P_{dcref} and V_{dcref} , but remains unchanged against step changes in the grid voltages i.e. V_1 and V_2 .

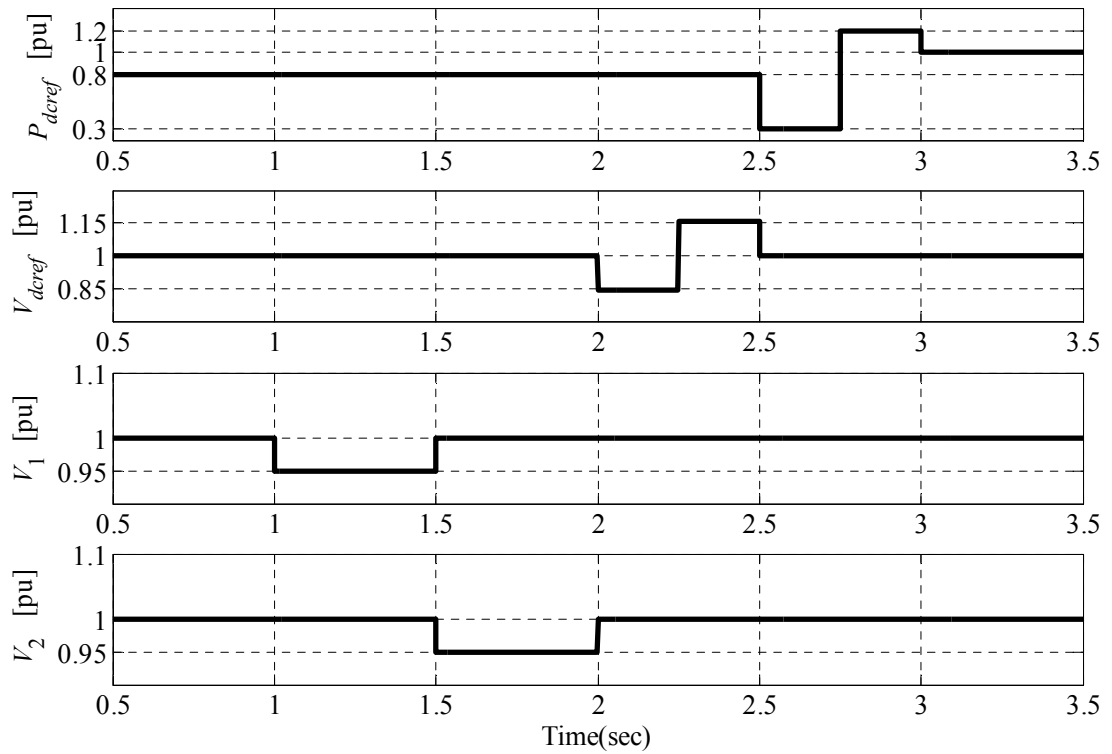


Figure 5-11 : Step changes in reference parameters.

The dynamic responses of the dc voltages and the dc powers of VSC1 and VSC2 for an initial setting of control variables of $\mathbf{x} = [10, 1.0 \times 10^{-3}, 10, 1.0 \times 10^{-3}, 5, 1.0 \times 10^{-4}]$ are shown in Figure 5-12. The dc voltages for VSCs 1 and 2 are labelled as V_{dc1} and V_{dc2} ; the ac voltages are V_{c1} and V_{c2} ; the dc powers are P_{dc1} and P_{dc2} ; and the reactive powers are

Q_1 and Q_2 . Although the controls regulate their assigned quantities, the system performance in terms of rise times, overshoots and ripple is poor.

The performance can be improved via the proposed optimization algorithm as seen in the post-optimization plots of Figure 5-13. Note that the ac voltage control is achieved as the VSCs are quickly able to adjust their reactive power to restore the voltage to its rated value. The objective function value for the initial parameter set was 647. This set of control variables is refined such that to give optimal performance using the proposed algorithm.

A 60-core parallel computing platform is used to run this simulation-based design. Each core has 16 GB of RAM and a clock-speed of 2.4 GHz. As shown in Table 5-9 the initial grid required 15625 objective function evaluations, which discovered 5 areas as the potential local minima locations. Due to parallelism, the total execution time was slightly more than 20 hours. The same algorithm running on a sequential platform (i.e. using only one of the 60 available processors) would have taken 367.2 hours or over 15 days. Due to the computationally intensive nature of the objective function (as it involves detailed EMT simulations of a complex system) speed-up for this is by a factor of over 34; much higher than that for the simple algebraic functions reported in Table 4-1.

Table 5-9: Optimization specifics

Number of objective function evaluations to establish initial grid	15625	
Number of discovered local areas	5	
Number of local objective function evaluations per local area	14897	
Total number of objective function evaluations	93750	
Total simulation time using sequential version (hours)	367.2	34.2 times speed-up
Total simulation time using parallel version (hours)	20	

Note that the availability of 60 cores in this example does not result in a speed-up factor of 60. This is because, as shown in Figure 4-1, parts of the proposed algorithm, such as initialization and localization are not executed in parallel. However, once the algorithm localizes a number of areas wherein local minima exist (5 in this example) it can split into 5 completely parallel paths, thereby harnessing the parallelism efficiently.

The multi-modal optimization algorithm converged to five local optimal solutions for this problem. These solutions are listed in Table 5-10. All OFs are relatively close however solution \mathbf{x}_3^* denotes the global minimum with an objective function value of 66.59. There is a substantial reduction from the pre-optimization OF value of 647.

Table 5-10: Local optimal solutions for (5-6)

Local minimum	K_{AC1}^*	T_{AC1}^*	K_{AC2}^*	T_{AC2}^*	K_{dc}^*	T_{dc}^*	OF(\mathbf{x}^*)
\mathbf{x}_1^*	48.2	13×10^{-4}	28.5	2.1×10^{-4}	15.6	6.5×10^{-3}	68.63
\mathbf{x}_2^*	43.7	3.3×10^{-4}	46.2	1.9×10^{-4}	32.1	5.1×10^{-2}	67.37
\mathbf{x}_3^*	59.8	2.7×10^{-4}	34.0	2.1×10^{-4}	29.2	7.0×10^{-3}	66.59
\mathbf{x}_4^*	71.0	2.0×10^{-4}	37.5	1.0×10^{-3}	12.0	11×10^{-2}	70.13
\mathbf{x}_5^*	61.2	2.1×10^{-4}	27.3	3.1×10^{-4}	41.2	87×10^{-3}	68.55

Figure 5-13 shows the same quantities as in Figure 5-14, when optimized controller parameters of solution \mathbf{x}_3^* (the best solution) are used. It is clearly seen that the optimized variable values yield much improved dynamic performance for both the dc voltage and the real power.

As seen in Figure 5-13 (a), there is a sharp increase in the dc power of VSC1 around 2.3 sec.; the dc power in VSC2 does not show a transient of comparable magnitude around that time. This transient is due to the sharp rise in the reference dc voltage. The dc

cable stores a large amount of energy during steady state operation in its electric field (represented by the equivalent cable capacitance), which is proportional to the square of the dc voltage. When the dc link voltage is commanded to rise, additional energy must be transferred and stored in the cable's electric field. Despite the large magnitude of the real power transient, the amount of the energy that is transferred to the dc link during the short-term transient is only a fraction of the base-line stored energy.

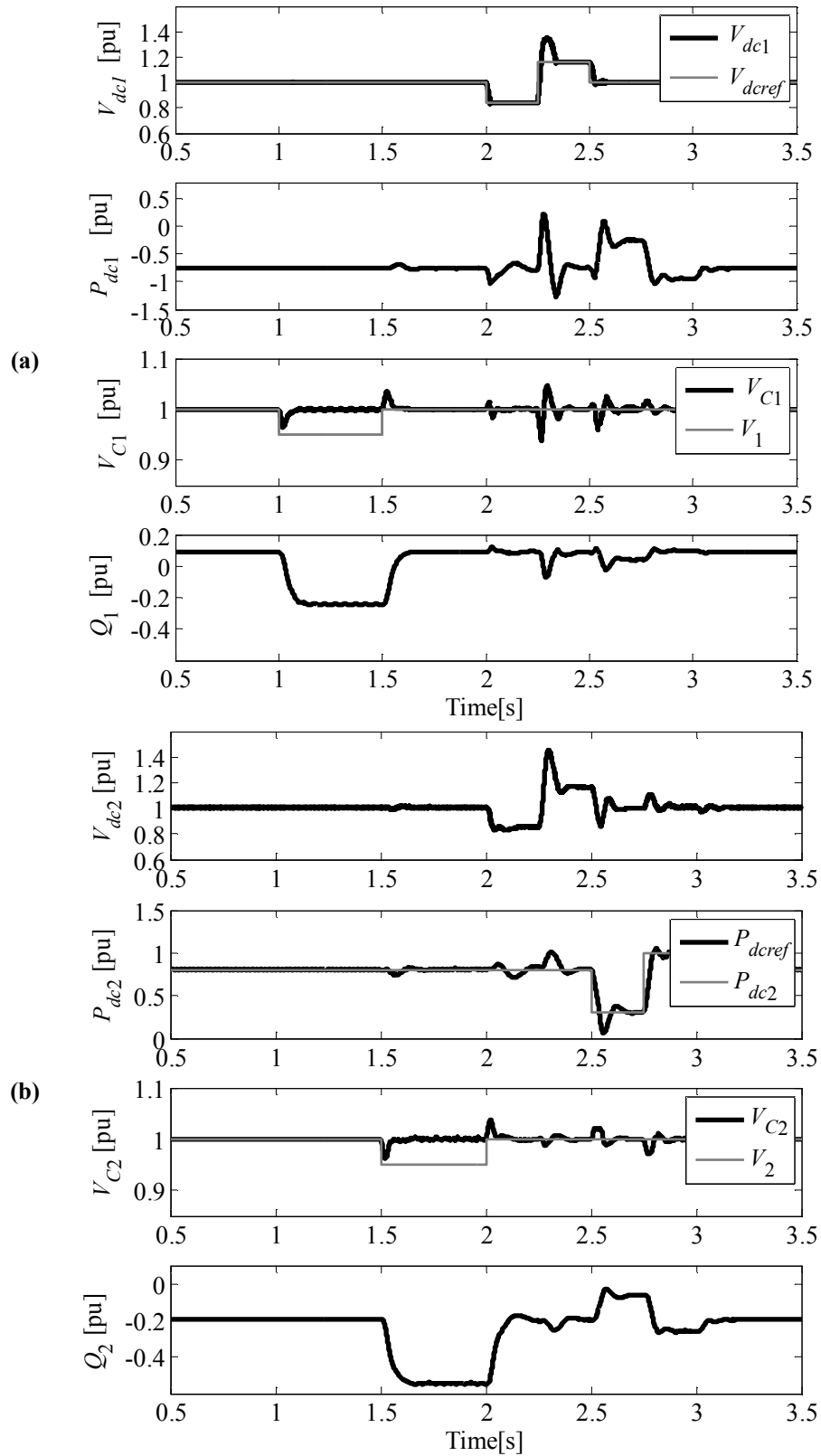


Figure 5-12 : Dynamic response of non-optimized system; (a) VSC1 and (b) VSC2.

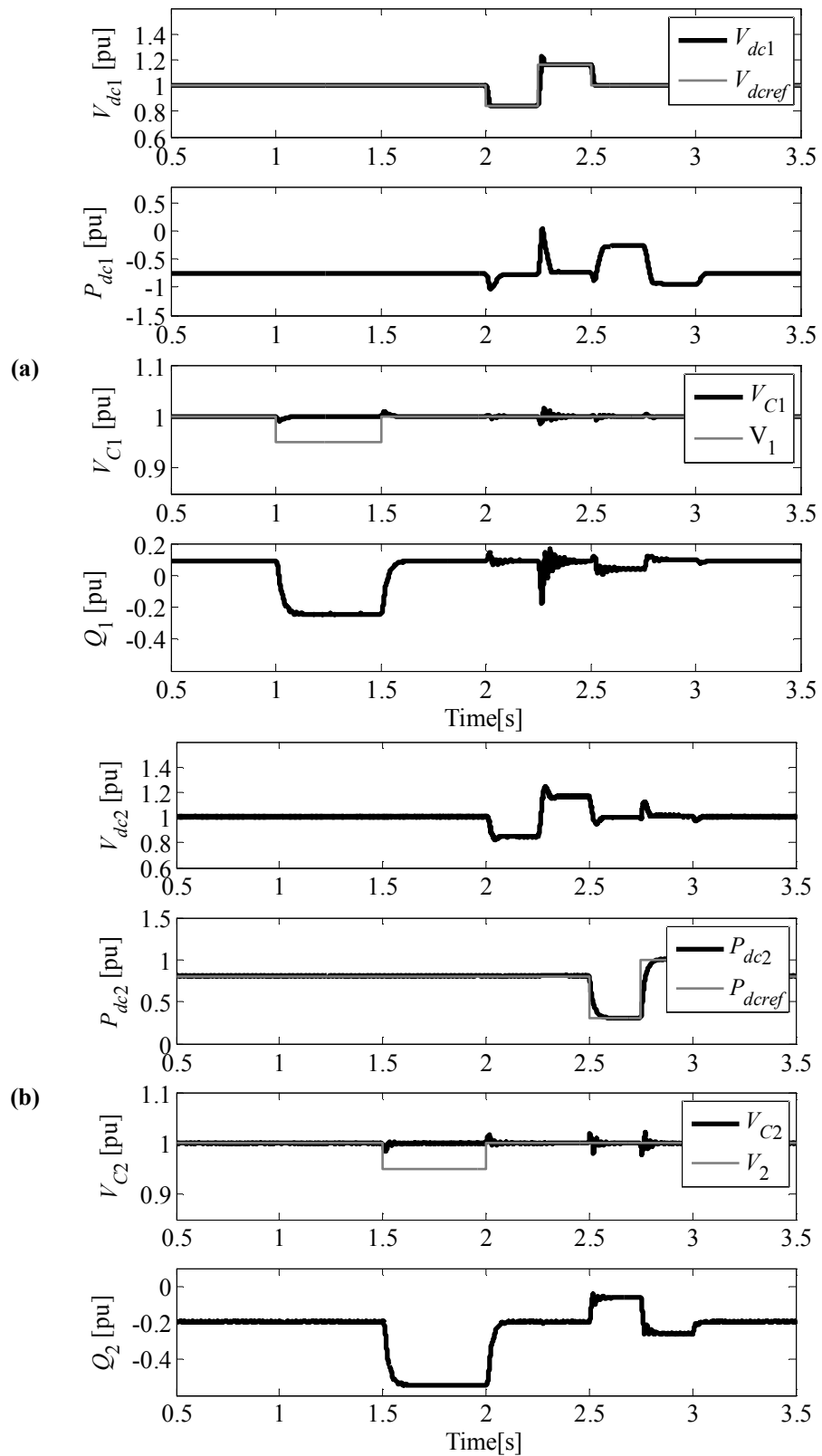


Figure 5-13 : Dynamic response of optimized controllers; (a) VSC1 and (b) VSC2.

5.2.4 Sensitivity Assessment

As it was described in Chapter 3, sensitivity assessment can be conducted for each decision variable individually to observe its effect on the objective function (see, Section 3.2). Also a range of detuning from zero to $\pm 5\%$ is considered in sensitivity assessment for the present case. Figure 5-14 (a) and (b) show the variation of the objective function (i.e. ΔOF) versus ΔK_{dc} and ΔT_{dc} in the proportional and the integral time constant (K_{dc} and T_{dc}), respectively.

As can be seen from Figure 5-14(a) and (b), local minimum \mathbf{x}_1^* shows the least sensitivity to detuning of variables for both positive and the negative parameter variations. Although local minimum \mathbf{x}_3^* has the lowest OF value among the five local solutions, local minimum \mathbf{x}_1^* may be preferred as the system shows higher robustness against detuning of variables. The performance penalty is relatively modest, as the objective functions increases to 68.63 from 66.59.

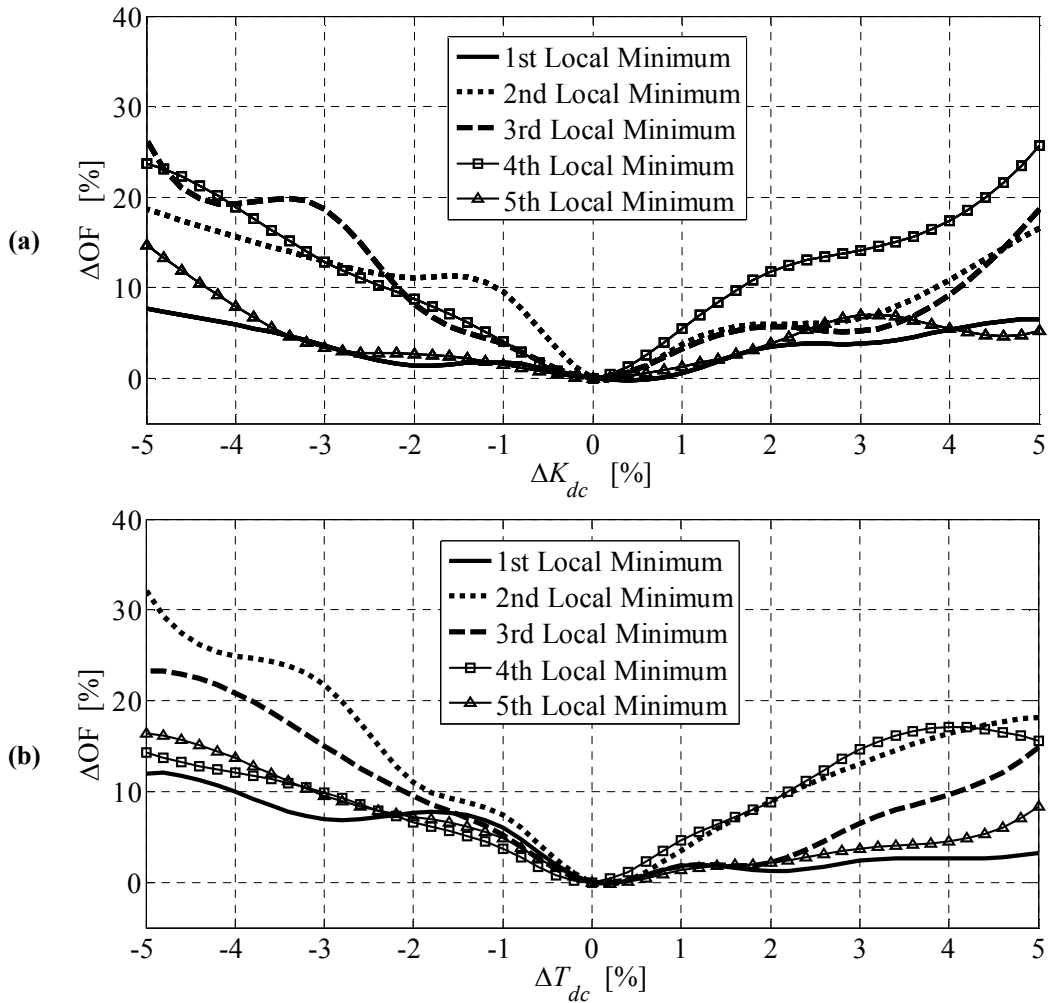


Figure 5-14: Objective function variation (ΔOF) versus detuning in (a) ΔK_{dc} and (b) ΔT_{dc} .

5.3 Design Example 3: Three Terminal VSC–HVDC Transmission System

A three terminal VSC-HVDC transmission system is selected as a case study in this section; in this example the number of variables to be optimized is to twelve which is much larger than the previously shown examples. In such a case memory usage becomes

a challenge. The technique proposed in Chapter 4 will be demonstrated here to show how it can overcome the challenge. As it was discussed the decision space is divided into sub-areas and each of them will be searched individually to avoid memory overflow. The example it will demonstrate that excessive computation time and simulation usage can be avoided during sensitivity assessments by using the surrogate models.

Note that in the previous work [59] an average model of HVDC system is used to reduce the simulation time and consequently the entire optimization process. In this thesis, however, a detailed model of the system is considered because simulations can be conducted in parallel.

5.3.1 Three-Terminal VSC-HVDC System

Figure 5-8 shows the three-terminal VSC-HVDC system, where three converters (VSC1, VSC2, and VSC3) are connected through a network of dc cables. Each bipolar cable has a length of 100km with the rated values of 500kV and 2kA. VSC1 controls ac voltage magnitude (V_{c1}) and the network's dc voltage (V_{dc1}). VSC2 and VSC3 control their ac voltage magnitudes, i.e. V_{c2} and V_{c3} and dc powers i.e., P_{dc2} , and P_{dc3} , respectively. The controllers and power system information are identical to those shown in the second example.

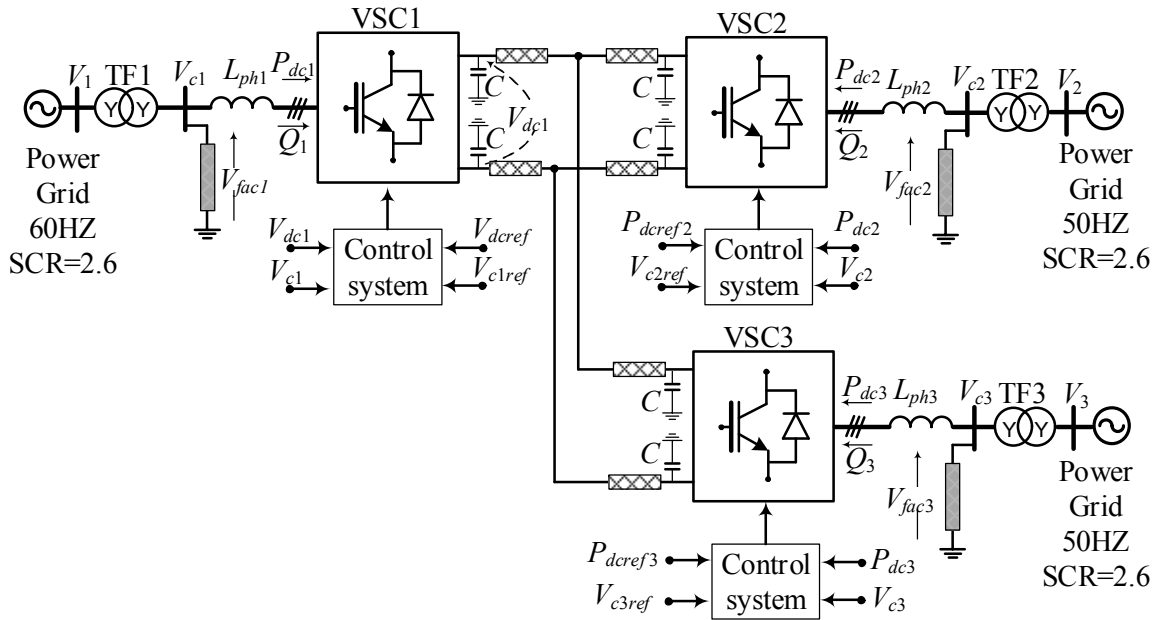


Figure 5-15: Schematic diagram of the three-terminal VSC-HVDC system.

5.3.2 Aggregate Objective Function and Boundaries

The four objectives of the design are stated as follows [61]:

- i. To ensure that the dc voltage, V_{dc1} follows the reference, V_{dc1ref} as closely as possible during transient and steady state conditions;
- ii. To ensure that the real power, P_{dc2} follows the reference, P_{dc2ref} as closely as possible during transient and steady state conditions;
- iii. To ensure that the real power, P_{dc3} follows the reference, P_{dc3ref} as closely as possible during transient and steady state conditions; and
- iv. To ensure that the terminal voltages of VSC1 (V_{c1}), VSC2 (V_{c2}) and VSC3 (V_{c3}) follow their reference values as closely as possible during transient and steady state conditions.

The above four objectives can be captured by the following four weighted sub-objective integral square errors (ISE):

$$\begin{aligned}
f_1(\mathbf{x}) &= C_{VT} \int_{transient} (V_{dcref}(t) - V_{dc1}(t))^2 dt + C_{VS} \int_{steadystate} (V_{dcref}(t) - V_{dc1}(t))^2 dt \\
f_2(\mathbf{x}) &= C_{PT2} \int_{transient} (P_{dcref2}(t) - P_{dc2}(t))^2 dt + C_{PS2} \int_{steadystate} (P_{dcref2}(t) - P_{dc2}(t))^2 dt \\
f_3(\mathbf{x}) &= C_{PT3} \int_{transient} (P_{dcref3}(t) - P_{dc3}(t))^2 dt + C_{PS3} \int_{steadystate} (P_{dcref3}(t) - P_{dc3}(t))^2 dt \\
f_4(\mathbf{x}) &= C_{Vc1T} \int_{transient} (V_{c1ref}(t) - V_{c1}(t))^2 dt + C_{Vc2T} \int_{transient} (V_{c2ref}(t) - V_{c2}(t))^2 dt \\
&\quad + C_{Vc3T} \int_{transient} (V_{c3ref}(t) - V_{c3}(t))^2 dt + C_{Vc1S} \int_{steadystate} (V_{c1ref}(t) - V_{c1}(t))^2 dt \\
&\quad + C_{Vc2S} \int_{steadystate} (V_{c2ref}(t) - V_{c2}(t))^2 dt + C_{Vc3S} \int_{steadystate} (V_{c3ref}(t) - V_{c3}(t))^2 dt
\end{aligned} \tag{5-7}$$

where C_{VT} , C_{VS} , C_{PT2} , C_{PS2} , C_{PT3} , C_{PS3} , C_{Vc1T} , C_{Vc2T} , C_{Vc3T} , C_{Vc1S} , C_{Vc2S} and C_{Vc3S} are the weights as given in Table 5-11.

Table 5-11: Coefficient Values for Objective Functions.

Coefficient	C_{VT}	C_{VS}	C_{PT}	C_{PS}	C_{Vc1T}	C_{Vc1S}	C_{Vc2T}	C_{Vc2S}	C_{Vc3T}	C_{Vc3S}
Value	0.035	4.0	5.0	5.0	10	10	10	10	10	10

The four objective functions in (5-7) are then combined into a single aggregate objective function as follows.

$$\begin{aligned}
f(\mathbf{x}) &= f_1(\mathbf{x}) + f_2(\mathbf{x}) + f_3(\mathbf{x}) + f_4(\mathbf{x}) \\
\mathbf{x} &= \{K_{Aci}, T_{Aci}, K_{Pd1}, T_{Pd1}, K_{Pd2}, T_{Pd2}, K_{Vd}, T_{Vd}\} \quad i = 1, 2, 3
\end{aligned} \tag{5-8}$$

where \mathbf{x} is the set of variables to be optimally selected. Here six PI controllers are to be optimized as follows:

- i. three PI controllers that control the ac voltages of the three converters contains six parameters (i.e., $K_{Aci}, T_{Aci}, i=1,2,3$);
- ii. two PI controllers that control the dc power of VSC2 and VSC3 contains four parameters (i.e., $K_{Pd1}, T_{Pd1}, K_{Pd2}, T_{Pd2}$);
- iii. one PI controller that controls the dc voltage of VSC1 contains two parameters (i.e., K_{Vd}, T_{Vd}).

As it was described in Section 5.2.2 the weights used in the individual objectives in (5-7) determine their relative significance in the aggregate function in (5-8). The objective function in (5-8) is subjected to the boundaries shown in Table 5-12.

Table 5-12: Boundaries of Variables.

	K_{ACi}	T_{ACi}	K_{pd1}	T_{pd1}	K_{pd2}	T_{pd2}	K_{dc}	T_{dc}
Minimum	1.0	1×10^{-5}	1.0	1.0	1×10^{-5}	1×10^{-5}	1.0	1.0×10^{-5}
Maximum	100	1.0	100	100	1.0	1.0	200	1.0×10^{-1}

5.3.3 Algorithm's Performance

To tune the above mentioned PI controllers a number of variations are applied to the reference commands as shown in Figure 5-16. The variations are shaped as follows:

- i. the ac voltage references of the grids are dropped from 1.0pu to 0.95pu to examine the ac voltage controllers in upholding the terminal voltage;
- ii. the dc voltage reference is varied between 0.9pu to 1.1pu to examine if the dc-voltage controller of VSC1 can follow the variations;
- iii. the dc power references of VSC2 and VSC3 are changed drastically (from positive to negative) to dictate different operating points to the system. This is to examine if the dc power controllers of VSC2 and VSC3 can operate appropriately

under markedly different operating conditions and provide acceptable transient performances.

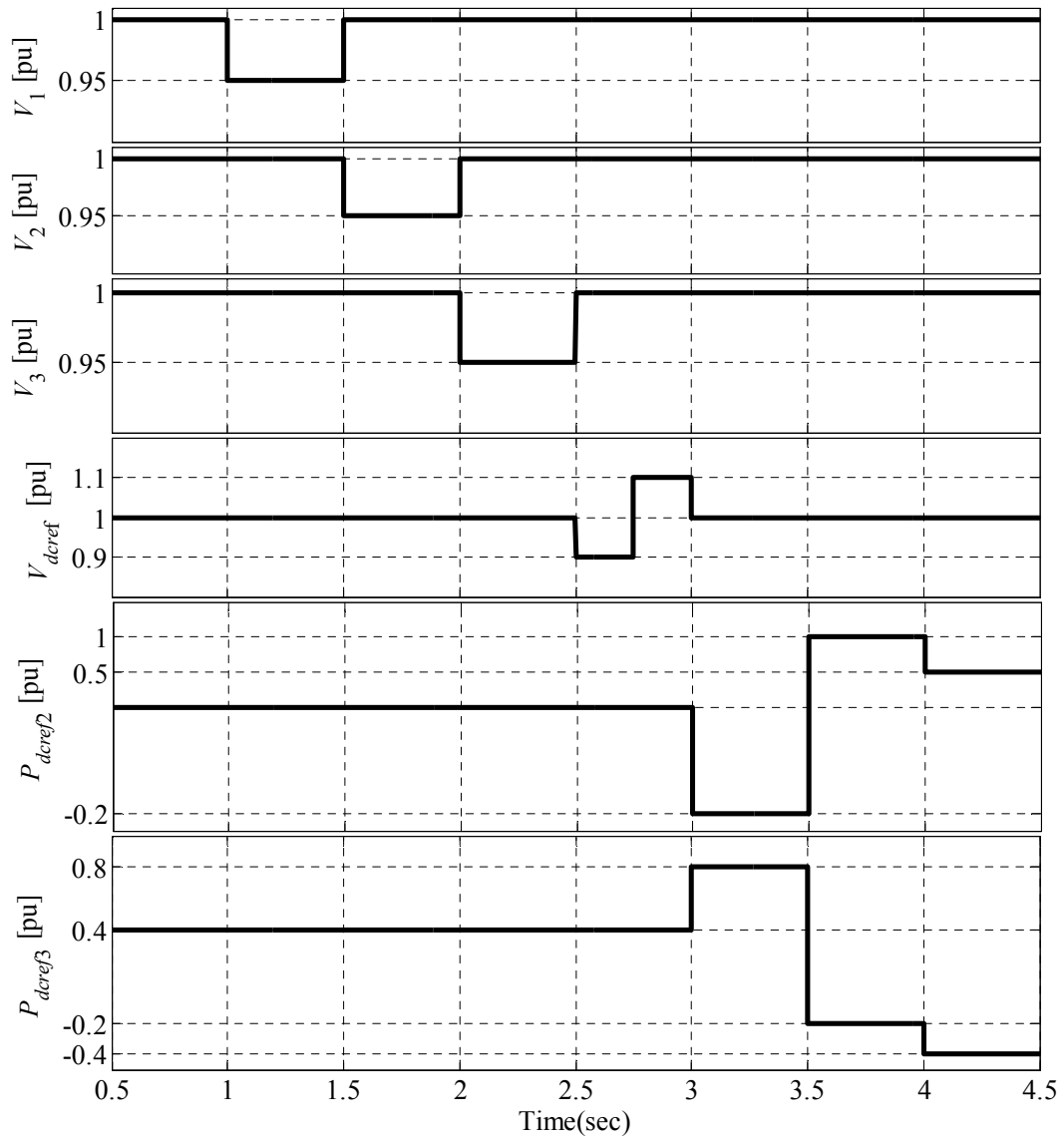


Figure 5-16: Variations in command signals.

The number of initial sampling points (n_s) that are required by the algorithm to approximation the local areas and the associated memory requirement can be calculated using (3-4) as follows.

$$(n_a)_i = \begin{cases} 4 & i = 1, 2, 3 \\ 5 & i = 4, \dots, 12 \end{cases}$$

$$n_s = \prod_{i=1}^{n_v} (n_a)_i = 1.25 \times 10^8$$

$$M_u = 64 \text{ bit} \times (n_v + 1) \times n_s = 96.8575 \text{ Gbit}$$

The amount of memory usage (M_u) is beyond the capacity of the machine (with 64Gbit of RAM) that is used to perform the optimization. This will cause memory crash. To avoid this problems, the decision area is divided into smaller sub-areas. Since the sub-areas are smaller than the original decision area, a smaller number of sampling points is needed for each. In this way the algorithm can search for multiple local areas in sub-areas one at a time and avoid memory crash.

For the current example, the number of sub-areas (n_{sub}) that is required to avoid memory crash can be calculated as follows:

$$n_{slim} = \frac{M_{lim}}{64 \text{ bit} \times (1 + n_v)} = \frac{64 \text{ Gbit}}{64 \text{ bit} \times (1 + 12)} = 8.2596 \times 10^7$$

$$n_{sub} = \left\lceil \frac{n_s}{n_{slim}} \right\rceil + 1 = \left\lceil \frac{1.25 \times 10^8}{8.2596 \times 10^7} \right\rceil + 1 = 2$$

Thus the algorithm splits the decision area in two sub-areas and searches each for multiple local optima following the principles described in Chapter 4.

Table 5-13 shows some of the specifics of the local optimal solution process using the parallelized algorithm. It is important to note that the optimization of this case is completed in less than three weeks; with a sequential solver it would have required more than two years to obtain the local optimal solutions. Therefore, both the parallel

computation and the memory usage reduction technique deployed in the algorithm have contributed to make it possible to solve this complex optimal design.

Table 5-13: Optimization specifics

Number of objective function evaluations per sub-area	1.25×10 ⁸	
Number of discovered local areas	7	
Total number of objective function evaluations	69984	
Total estimated simulation time using sequential algorithm (days)	731.3	39 times
Total simulation time using parallel algorithm (days)	18.7	speed-up

The multi-modal optimization algorithm converged to seven local optimal solutions for this problem. These solutions are listed in Table 5-14. All OFs are relatively close with solution \mathbf{x}_1^* showing the global minimum of 123.02.

The dynamic responses of the VSCs 1, 2 and 3 for the best solution \mathbf{x}_1^* are shown in Figure 5-17, Figure 5-18, and Figure 5-19, respectively. The dynamic responses of the non-optimised system (for the VSC1) are shown in Figure 5-17. As shown the dc voltages V_{dc1} , V_{dc2} , and V_{dc3} and the dc powers P_{dc1} , P_{dc2} , and P_{dc3} follow their assigned commands and the performance in terms of settling time, ripple and overshoot are satisfactory. Also, control of the ac voltages V_{c1} , V_{c2} , and V_{c3} is achieved as the VSCs are quickly able to adjust their reactive powers Q_1 , Q_2 , and Q_3 to restore the voltages to their rated values.

Table 5-14: Obtained local minima for (5-8)

\mathbf{x}^*	\mathbf{x}_1^*	\mathbf{x}_2^*	\mathbf{x}_3^*	\mathbf{x}_4^*	\mathbf{x}_5^*	\mathbf{x}_6^*	\mathbf{x}_7^*
K_{Ac1}^*	43.72	61.2	59.81	53.8	51.0	43.27	71.0
T_{Ac1}^*	3.3×10^{-4}	2.1×10^{-4}	2.7×10^{-4}	2.0×10^{-4}	2.2×10^{-4}	3.4×10^{-4}	2.2×10^{-4}
K_{Ac2}^*	46.23	16.8	27.6	17.2	42.0	27.5	15.2
T_{Ac2}^*	1.9×10^{-4}	2.7×10^{-4}	3.0×10^{-4}	2.0×10^{-3}	1.5×10^{-4}	3.0×10^{-4}	2×10^{-4}
K_{Ac3}^*	44.8	28.3	34.01	37.5	35.2	46.3	27.2
T_{Ac3}^*	2.0×10^{-4}	3.1×10^{-4}	2.1×10^{-4}	1.0×10^{-3}	2.0×10^{-4}	1.8×10^{-4}	1.3×10^{-4}
K_{Pd1}^*	58.2	20.9	41.27	25.4	42.0	60	54.0
T_{Pd1}^*	3.0×10^{-4}	1.1×10^{-4}	2.0×10^{-4}	1.7×10^{-4}	1.5×10^{-4}	2.5×10^{-4}	1.9×10^{-4}
K_{Pd2}^*	56.3	21.4	34.15	75.7	70.5	60.9	65.0
T_{Pd2}^*	1.6×10^{-4}	3.1×10^{-4}	2.0×10^{-4}	3.3×10^{-4}	2.0×10^{-4}	2.2×10^{-4}	2.0×10^{-4}
K_{Vd}^*	32.1	17.5	5.2	12.6	26.0	7.5	15.0
T_{Vd}^*	5.1×10^{-2}	8.7×10^{-2}	7.0×10^{-3}	1.1×10^{-2}	5.0×10^{-3}	7.0×10^{-3}	6.7×10^{-3}
$OF(\mathbf{x}^*)$	123.02	127.76	123.17	130.39	126.97	123.22	129.77

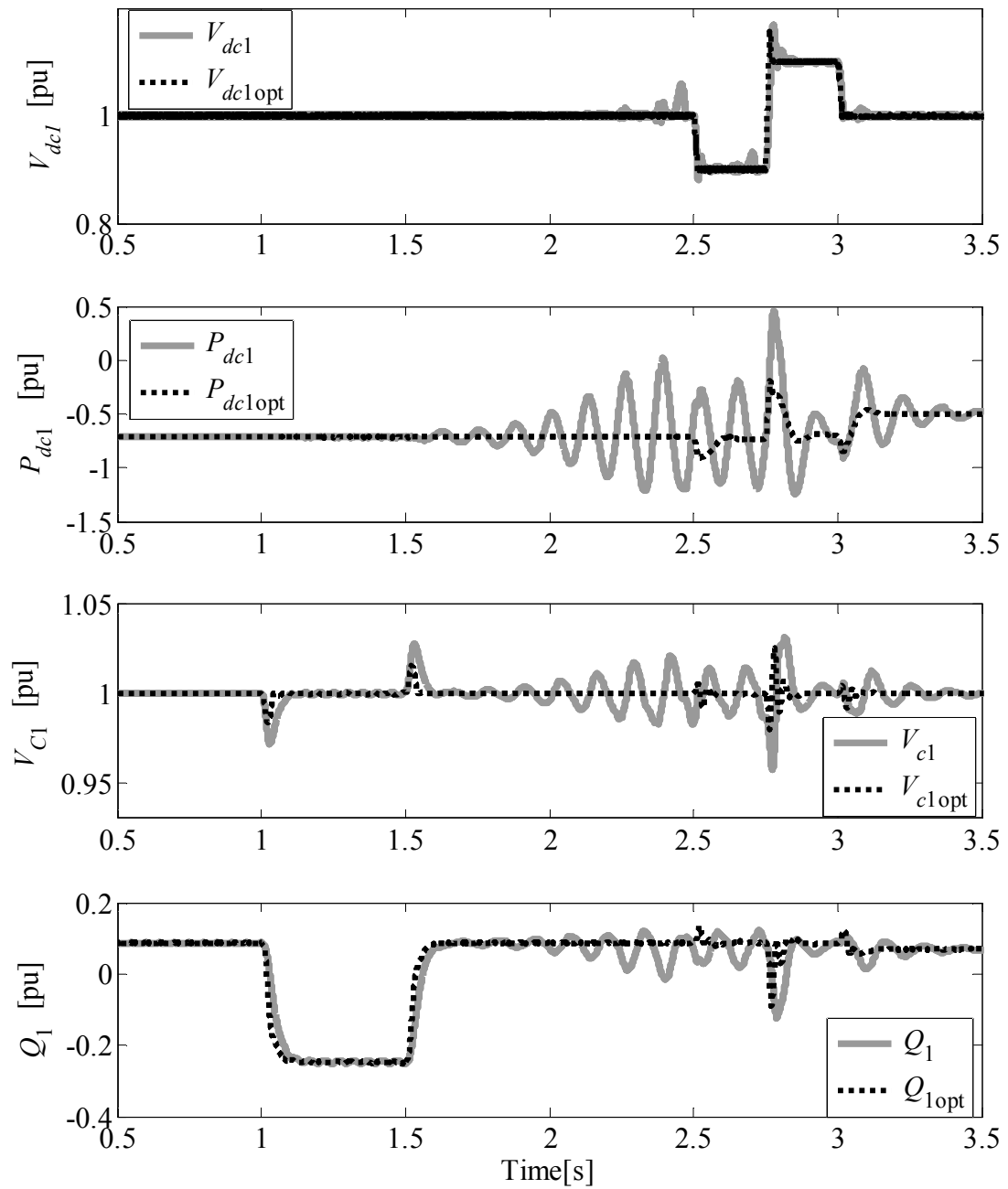


Figure 5-17 : Optimized and non-optimized dynamic responses of VSC1.

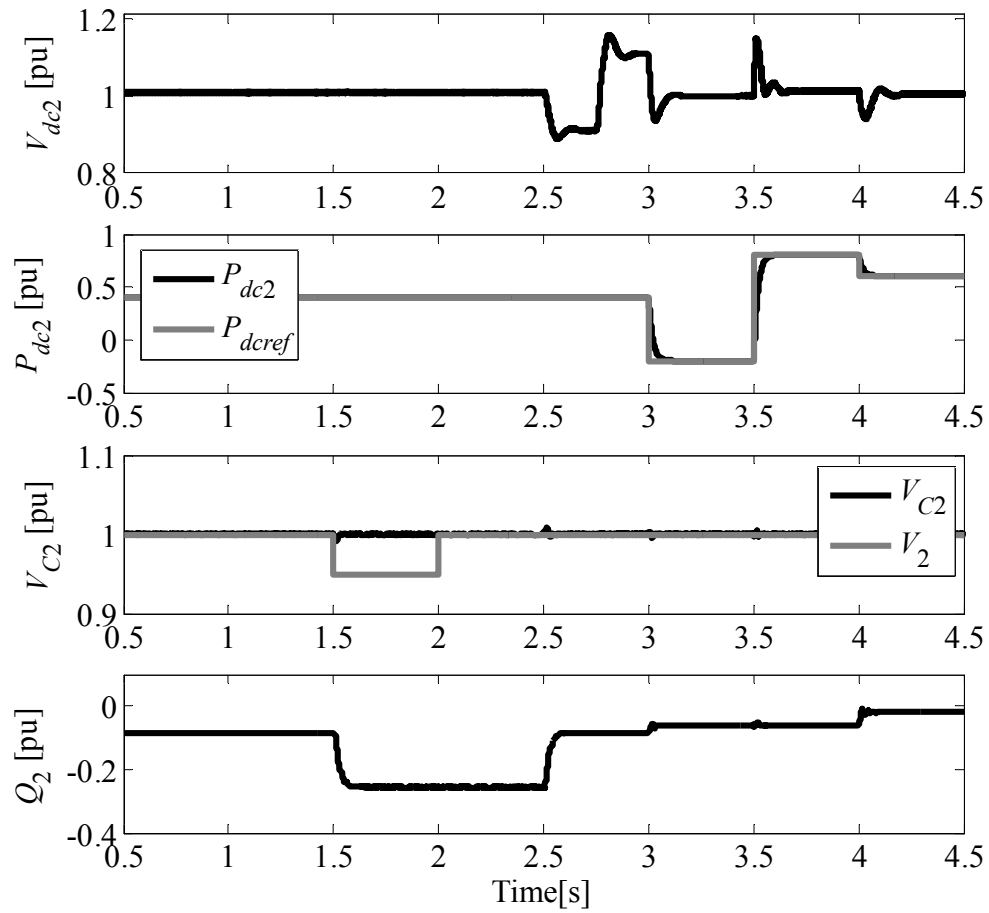


Figure 5-18 : Optimized dynamic responses of VSC2.

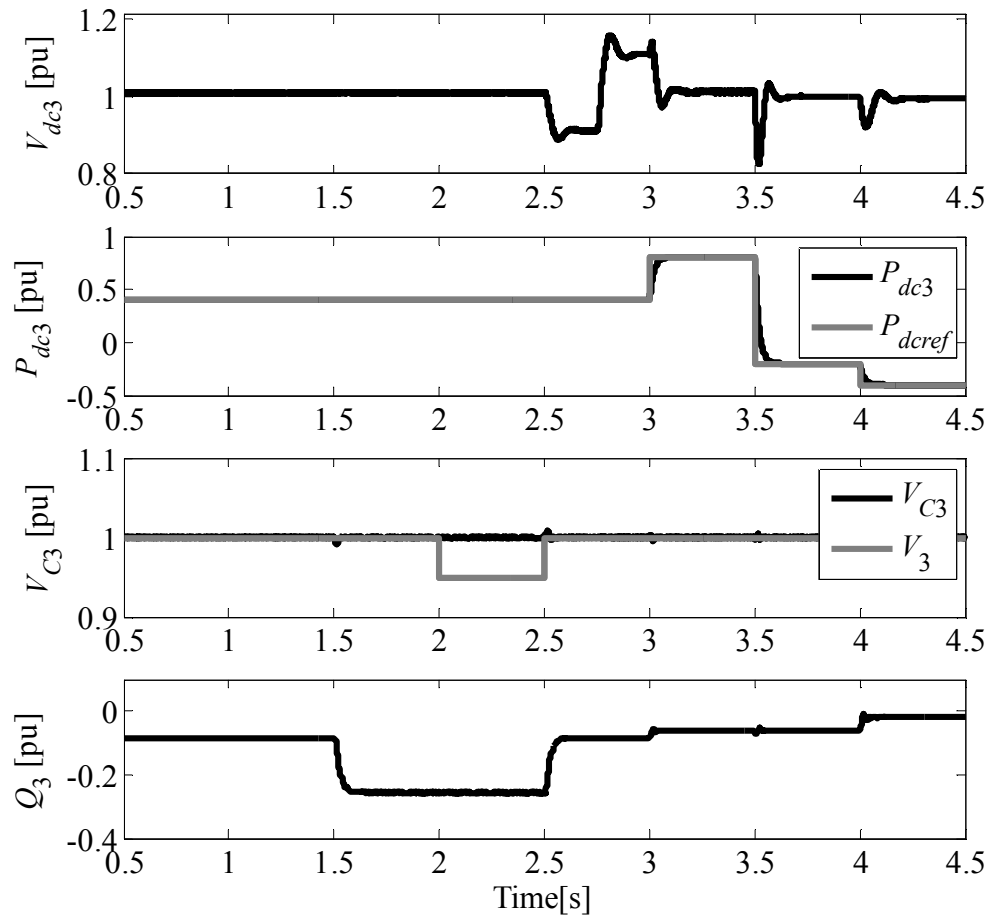


Figure 5-19 : Optimized dynamic responses of VSC3.

5.3.4 Sensitivity Assessment

Sensitivity assessment of the optimal design to the perturbations in the optimized variables is addressed in this section considering a detuning of $\pm 5\%$ in the decision variables. As it was discussed in Section 5.1.3 the number of detuned points associate to each local optimum can be calculated as:

$$n_{xd}^* = \prod_{i=1}^{n_v} (2)_i = \prod_{i=1}^{12} (2)_i = 4096$$

It is required to conduct 4096 simulation for each local optimum leading to $4096 \times 7 = 28672$ simulations to perform the entire sensitivity analysis. Actual EMT simulations are avoided as the surrogate models of the objective function around each local area are available to replace EMT simulations. If it was used to simulate the 28672 cases it would require around 220 hours while using surrogate models requires less than one hour. Therefore, sensitivity analysis using the developed surrogate models saves about 219 hours of computing time.

Figure 5-20 shows the variations of the objective function (ΔOF) against detuning. To obtain the most robust minimum solution these variations for each local area are averaged and shown in Figure 5-21. Note that in this sensitivity study shown in Figure 5-21 all 4096 detuning possibilities are considered.

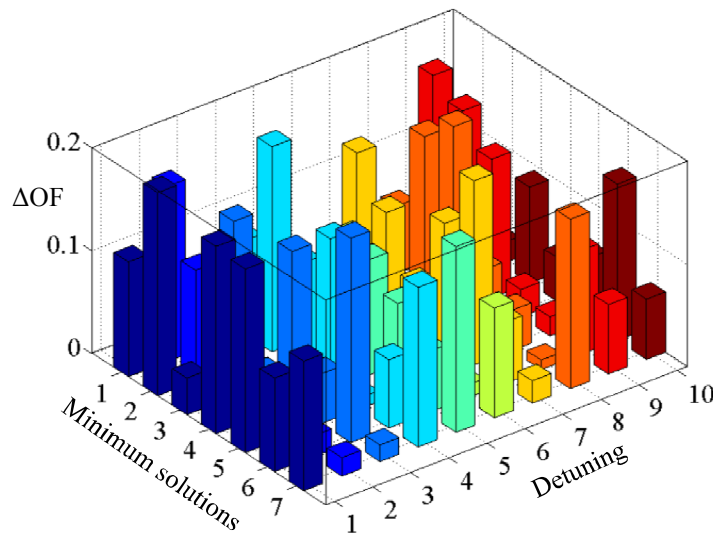


Figure 5-20: Objective function variations (ΔOF) against detuning in minimum solutions.

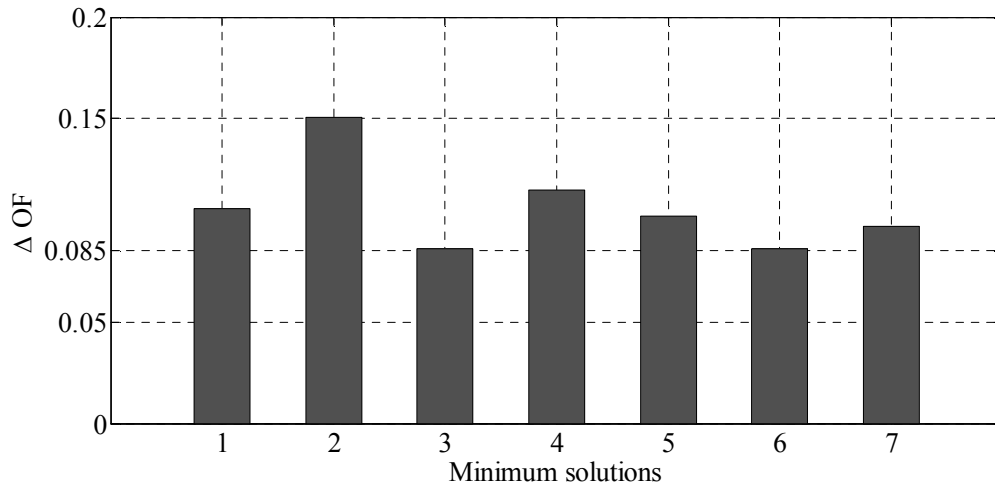


Figure 5-21: Sensitivity assessment of the six local optimal solutions.

As it is shown in Figure 5-21, local minima \mathbf{x}_3^* and \mathbf{x}_6^* show the smallest variations to perturbations compared to the others. From Table 5-14 it is obvious that the objective function values are also very close for these two local optimal solutions. Accordingly one of these two can be selected as the control settings of three-terminal HVDC system.

5.4 Conclusions and Discussion

In this chapter the practical use of the proposed multi-modal optimization algorithms was demonstrated via a number of power system design problems, in which black-box objective functions needed to be evaluated via simulation. The parallel version of the algorithm employed a 60-core parallel computing platform and yielded significant speed-up (up to 39 times) over the conventional single-processor approach.

Example 3 revealed that the algorithm can avoid memory overflow during optimization of a large complex power systems where twelve decision variables were to

be optimized, and consequently a large number of sampling points were required. In addition, it was shown that implementing both the parallel computation and the memory reduction technique in the algorithm paves the way to design complex systems, whose design is not possible by earlier multi-modal optimization algorithms [27], [28].

Local surrogate models of the objective function are generated by the algorithm to estimate the objective functions in the neighborhood of discovered minima. This reduced computation time and saved 219 hours to accomplish the sensitivity assessment for the system in Example 3.

Chapter 6

Conclusions, Contributions, and Future Directions

In this thesis a novel multi-modal optimization algorithm was developed for simulation-based optimization of complex electric power systems. In such optimization problems, the objective function is only available in a “black-box” form and cannot be explicitly expressed as a mathematical equation in terms of the variables to be optimized. Both versions of the proposed algorithm, sequential and parallel, were implemented in PSCAD/EMTDC simulator to unify the setting-up, testing, and execution of optimal designs for power systems.

The main contributions of the thesis in terms of computation speed-up, memory usage, sensitivity analysis, and applications are summarized in Section 6.1 and recommendations for further research directions are pinpointed in section 6.2.

6.1 Research Contributions

The four main contributions of the thesis are summarized in this section.

6.1.1 Computation Speed-up

The proposed optimization algorithm can discover multiple local optimal solutions and conduct sensitivity assessment with higher speed, compared to the other multi-modal and the evolutionary optimization algorithm, due to the following reasons.

- i. Interpolation technique, used to find the surrogate models during optimization, is simpler more accurate and easy to implement. In previous surrogate model-based methods the objective function was estimated using an iterative approach that converged after several rounds of computations. In the proposed method, however, the surrogate models are obtained using fewer computations with more accuracy.
- ii. The algorithm first finds the local areas; therefore, the focus of the search for local optima is limited to those local areas rather than the entire decision area. This causes reduction in the objective function evaluations compared to the multi-modal evolutionary algorithms. Objective function evaluations in the case of power system design require expensive simulations that are time consuming. The scalability studies in the thesis prove that as the number of local optimal solutions increases the number of objective function evaluations for the proposed algorithm is less compared to multi-modal evolutionary algorithm.

iii. Computations of the proposed algorithm are conducted in parallel, and also searching for local optimal solutions. The reason is that the algorithm requires a number of objective function evaluations at an iteration that can be done in parallel. Also, once the algorithm discovers the vicinity of local areas, searching for finer minimum solutions in each local area can be done independently. Therefore, the algorithm is suited for parallel environments as it is provided by PSCAD/EMTDC [60].

6.1.2 Memory Usage

Memory usage reduction during the course of optimization is another feature of the proposed algorithm that is not addressed in the previous researches. The proposed algorithm maintains memory usage under a prescribed value that is defined before optimization. The process is that the algorithm checks if the number of sampling points is less than the predefined limit during initialization; if the number of sampling point is larger than the given limit, it divides the decision area into sub-areas and searches one sub-area at a time with a smaller number of sampling points but with almost the same resolution. The number of sub-areas and the number of sampling points are obtained using principles given in Chapter 4. The actual local areas can be recognized by comparing the geometry of the discovered local areas with their neighbors to maintain the accuracy of the optimization algorithm.

6.1.3 Sensitivity Assessment

In real-world implementations optimized variables may deviate from their optimal values due to reasons such as aging and/or temperature variations. Consequently the behaviour of the system may alter. Sensitivity analysis of an optimal design is the key to find the most robust optimal solution to the possible perturbation in real-world implementation.

During the course of optimization, the proposed algorithm also estimates the objective function by surrogate models around its local optima. Since these surrogate models represent the local behaviour of the objective function, they can be used in the subsequent stages of sensitivity and tolerance analysis of the local minima to substitute intensive simulations of the original black-box system.

In the proposed sensitivity assessment method actual variations in the objective function are considered as a measure of sensitivity. This enhances the sensitivity analysis in several aspects, including:

- i. Sensitivity analysis can be carried out considering a chosen range of detuning in a defined direction. This is important if there is prior knowledge of possible detuning in the system or a certain test needs to be conducted.
- ii. Sensitivity assessment can be conducted with fewer computations; there is no need to calculate the second derivative of objective function, and no simulation is required because accurate surrogate models of the objective function are available.

6.1.4 Applications

The proposed algorithm was implemented in PSCAD/EMTDC to unify optimal design for power systems. A step by step description of the algorithm's usage was given in Chapter 5 to clarify the simplicity of implementation and the potential applications. It was shown how design objectives can be transformed into several mathematical functions, which were then combined to form an aggregate objective function.

The parallel implementation of the algorithm was addressed in example 2 where it was shown how the algorithm solves complex designs to obtain multiple local solutions within a reasonable time frame [60]. Comparative assessment of the sequential and parallel implementation of the algorithm was carried out and presented that the parallel version outperforms the sequential one in terms of computation speed [60].

In the last example of Chapter 5 it was shown that memory management is another feature of the algorithm that limits memory usage through initialization [60]. Also it was shown that the algorithm is able to complete a highly complex optimization task, which exceeds the capabilities of the preceding multi-modal optimization algorithm, without exceeding memory limit and within a reasonable time frame [61].

6.2 Suggestions for Future Work

In this section some of recommendations for future work based on the conducted research in this thesis are made.

6.2.1 Reduction of the Number of Objective Function Evaluations

The proposed optimization algorithm can be modified to obtain multiple local solutions with a smaller number of objective function evaluations. In some cases, such as the third example in Chapter 5, a large number of variables are required to be optimized or a wide decision area must be searched. Therefore, in such cases a large number of sampling points is required. Consequently a large number of objective function evaluations and long computation process are inevitable. A solution would be to estimate the objective function over the entire decision area by distributing a small number of sampling points all over the decision area and evaluate their objective function values. In this way an estimation of the objective function using these values is possible to obtain a general knowledge of the objective function variations versus each decision variable and removing those variables that affect the objective function in a minor ways. The number of variables then can be reduced. Doing so a fewer number of sampling points and consequently less computation time will be required to discover all the local areas while accuracy of the solutions may negligibly degrade.

6.2.2 Enhancement of Sensitivity Analysis

The algorithm provides surrogate models of an objective function around its local minima to estimate its sensitivity to the decision variables at each local optimal solution. In all the studies provided in Chapter 5 sensitivity analysis was carried out considering merely the decision variables and not any other parameter from the power system under study. In other words, in some studies it may be required to estimate the sensitivity of the objective

function to a parameter from the system rather than the decision variables. For example in the second case study in Chapter 5 sensitivity analysis of the local optimal solutions against the transmission line parameters cannot be done using the surrogate models of the objective function. If such a study is required it must be done through direct EMT simulations. This sensitivity analysis could be done without any extra simulations if the algorithm could provide a surrogate model of the objective function against transmission line parameters. However, it is still questionable how this can be accomplished with an acceptable increase in the computational complexity of the algorithm.

6.2.3 Enhancement of Parallelism

The parallel implementation of the algorithm was done using parallel multiple-run (PMR) feature of PSCAD/EMTDC. In this feature multiple simulation runs can be executed automatically and, therefore, the algorithm is allowed to conduct multiple simulations instantaneously. The problem of this feature is that it cannot control the number of processors that are required to do the current task. In other words, if one processor completes its task sooner compared to the others it will not be assigned another task because it is seized until all the processors complete their task. This causes inefficient usage of computational power of the processors in a parallel platform. To overcome this problem an upgraded version of Parallel Multiple Run (PMR) is currently developed in PSCAD/EMTDC that can control each processor independently and paves the way for independent communication between the optimization algorithm and each processor in a parallel platform. In this new feature the computational power of the processors can be

exploited more efficiently and hence the simulation and computation times can be reduced significantly. Therefore, the developed optimization algorithm can be enhanced to adopt this feature and continuously update the number of available processors.

Some of the stages of the proposed algorithm such as initialization, localization and density increment are still sequential in the parallel version of the algorithm. These stages can be parallelized by using OpenMP programming that allows parallelism in FORTRAN programs.

Bibliography

- [1] R. Rutenbar, G. Gielen, B. Antao, *Computer-Aided Design of Analog Integrated Circuits and Systems*, Wiley-IEEE Press, 2002, pp. 3–30.
- [2] R. Rai, “Simulation-Based Design of Aircraft Electrical Power Systems,” *Proceedings of the 8th Modelica Conf.*, 2011, pp. 704–712.
- [3] R. O. Bowden, J. D. Hall, “Simulation optimization research and development,” *Simulation Conference Proceedings*, vol. 2, 1998, pp.1693–1698.
- [4] J. R. Clymer, R. John, “Simulation-based engineering of complex adaptive systems using a classifier block, Simulation Symposium,” *34th Annual IEEE Simulation Symposium*, 2001, pp. 243–250.
- [5] F. Azadivar, “A tutorial on simulation optimization,” *24th Winter Simulation Conf.*, 1992, pp. 198–204.
- [6] Z. Furness, V. Gawron, “Enabling engineering of complex systems through simulation-based experimentation,” *Systems Conference, 4th Annual IEEE*, 2010, pp. 217–222.
- [7] M. Heidari, S. Filizadeh, A. M. Gole, “Electromagnetic Transients Simulation-Based Surrogate Models for Tolerance Analysis of FACTS Apparatus,” *IEEE Trans. Power Delivery*, vol. 28, no. 2, pp. 797–806, Mar. 2013.
- [8] K. Kobravi, “Optimization-enabled transient simulation for design of power circuits with multi-modal objective functions,” M. Sc. Dissertation, Dept. Electrical and Computer Eng., Univ. of Manitoba, Winnipeg, Canada, 2007.
- [9] J. Parry, R. B. Bornoff, P. Stehouwer, L. T. Driessen, E. Stinstra, “Simulation-based design optimization methodologies applied to CFD,” *IEEE Trans. Components and Packaging Technologies*, vol. 27, no. 2, pp. 391–397, June 2004.
- [10] S. Akhtar, H. Linshu, V. Akhtar, “Simulation-based optimization and sizing for propulsion system of liquid rocket using genetic algorithm,” *IEEE Sym. Emerging Techs.*, 2005, pp. 376–381.

- [11] G. Simon, P. Volgyesi, M. Maroti, A. Ledecz, "Simulation-based optimization of communication protocols for large-scale wireless sensor networks," *IEEE Aerospace Conf.* 2003, vol. 3, pp. 1339–1346.
- [12] Y. Seongjin, Ch. Jaewoong, Y. Kyongsu, "Coordinated Control of Hybrid 4WD Vehicles for Enhanced Maneuverability and Lateral Stability," *IEEE Trans. Vehicular Technology*, vol. 61, no. 4, pp. 1946–1950, May 2012.
- [13] R. M. Calazan, N. Nedjah, L. de Macedo Mourelle, "Parallel GPU-based implementation of high dimension Particle Swarm Optimizations," *IEEE Fourth Latin American Symposium on Digital Object Identifier, Circuits and Systems*, 2013, pp. 1–4.
- [14] J. Kumar, L. Singh, S. Paul, "GPU based parallel cooperative Particle Swarm Optimization using C-CUDA: A case study," *IEEE International Conf., Fuzzy Systems*, 2013, pp. 1–8.
- [15] A. Greenhall, R. Christie, J. P. Watson, "Minpower: A power systems optimization toolkit," *IEEE Power and Energy Society General Meeting*, 2012, pp. 1–6.
- [16] M. Heidari, S. Filizadeh, A. M. Gole, "Support tools for simulation-based optimal design of power networks with embedded power electronics," *IEEE Trans. Power Delivery*, vol. 23, no.3, pp. 1561–1570, July 2008.
- [17] F. Shengtao, D. Hui, "Time Domain Transformation Method for Accelerating EMTP Simulation of Power System Dynamics," *IEEE Trans. Power Systems*, vol. 27, no. 4, pp. 1778–1787, Nov. 2012.
- [18] H. W. Dommel, "Digital Computer Solution of Electromagnetic Transients in Single and Multiphase Networks," *IEEE Trans. Power Apparatus and Systems*, vol. 88, no. 4, pp. 388–399, Apr. 1969.
- [19] S. Filizadeh, A. M. Gole, D. A. Woodford, G. D. Irwin, "An optimization-enabled electromagnetic transient simulation-based methodology for HVDC controller design," *IEEE Trans. Power Delivery*, vol. 22, no. 4, pp. 2559–2566, Oct. 2007.
- [20] M. C. Fu, "Optimization via simulation: A review," *Annals of Operations Research*, vol. 53, no. 8, pp. 199–248, Jan. 1994.
- [21] S. Filizadeh, "Optimization-Enabled Electromagnetic Transient Simulation," Ph. D. Dissertation, Dept. Electrical and Computer Eng., Univ. of Manitoba, Winnipeg, Canada, 2004.
- [22] S. Filizadeh, and A. M. Gole, "Optimal design of power electronic systems using electromagnetic transient simulation," *Canadian Conference on Electrical and Computer Engineering*, 2005, pp. 450–453, 2005.
- [23] S. Filizadeh, *Electric Machines and Drives: Principles, Control, Modeling, and Simulation*, CRC Press Taylor & Francis Group, 2013, ch. 9.3, pp. 194-199.

- [24] D. P. Kothari, P. Dwarka “Power system optimization,” *Computational Intelligence and Signal Processing, 2nd National Conf.*, 2012, pp. 18–21.
- [25] A. M. Gole, S. Filizadeh, R. W. Menzies, and P. L. Wilson, “Optimization-enabled electromagnetic transient simulation,” *IEEE Trans. Power Delivery*, vol. 20, no.1, pp. 512–518, Jan. 2005.
- [26] A. M. Gole, S. A. Woodford, J. E. Nordstrom, and G. D. Irwin, “A fully interpolated controls library for electromagnetic transients simulation of power electronic systems,” *International Conference on Power System Transients*, 2001, pp.1–6 .
- [27] K. Kobravi, S. Filizadeh, “An adaptive multi-modal optimization algorithm for simulation-based design of power-electronic circuits,” *Engineering Optimization*, vol. 41, no. 10, pp. 945–969. Mar. 2009.
- [28] F. Yahyaie, S. Filizadeh, “A surrogate-model based multi-modal optimization algorithm,” *Engineering Optimization*, vol. 43, no. 7, pp. 779–799, July 2011.
- [29] J. T. Tsaia, “Robust optimal-parameter design approach for tolerance design problems,” *Engineering Optimization*, vol. 42, no. 12, pp. 1079–1093, Dec. 2010.
- [30] M. Li, et al., “Integrated multi-objective robust optimization and sensitivity analysis with irreducible and reducible interval uncertainty,” *Engineering Optimization*, vol. 41, no. 10, pp. 889–908, Oct. 2009.
- [31] J. T. Tsaia, “Robust optimal-parameter design approach for tolerance design problems,” *Engineering Optimization*, vol. 42, no. 12, pp. 1079–1093, Dec. 2010.
- [32] F. Yuan, and A. Opal, “Sensitivity Analysis of Periodically Switched Linear Circuits Using an Adjoint Network Technique,” *IEEE International Symposium on Circuits and Systems*, 1999, pp. 331–334.
- [33] Y. Wakasa, M. Nishimura, T. Ueno, K. Tanaka, “Application of robust optimization to power systems,” *SICE 2004 Annual Conf.*, 2004, pp. 1311–1314.
- [34] J. Kostrowicki, L. Piela, “Diffusion equation method of global minimization: Performance for standard test functions,” *Journal of Optimization Theory and Applications*, vol. 69, no. 1, pp. 269–284, May 1991.
- [35] A. Younis, Z. Dong, “Trends, features, and tests of common and recently introduced global optimization methods,” *Engineering Optimization*, vol. 42, no. 8, pp. 691–718, Aug. 2010.
- [36] J. Zhu, *Optimal Power Flow*, Wiley-IEEE Press pp. 297- 364, 2009.
- [37] T. Samad, “Intelligent Control: An Overview of Techniques,” *Perspectives in Control Engineering Technologies, Applications, and New Directions*. 2001, pp. 104–133.

- [38] A. Shapiro, Y. Wardi, “Convergence analysis of gradient descent stochastic algorithms,” *J. Optimiz. Theory Appl.*, vol. 91, no. 11, pp. 439–454, Nov. 1996.
- [39] P. W. Glynn, “Likelihood ratio gradient estimation for stochastic systems,” *Commun., ACM*, vol. 33, pp. 75–84, 1990.
- [40] R. Fletcher, C. M. Reeves, “Function minimization by conjugate gradients,” *The Computer Journal*, vol. 7, pp. 149–154, May 1964.
- [41] R. Hooke, T. A. Jeeves, “Direct Search Solution of Numerical and Statistical Problems,” *J.ACM*, vol. 8, pp. 212–229, Mar. 1961.
- [42] M. J. D. Powell, “A direct search optimization method that models the objective and constraint functions by linear interpolation”, *Advances in Optimization and Numerical Analysis*, vol. 275, pp. 51–67, Feb. 1994.
- [43] M. J. D. Powell, “UOBYQA: unconstrained optimization by quadratic approximation”, *Math. Programming*, vol. 92, pp. 555–582, May 2002.
- [44] J. A. Nelder, R. Mead, “A Simplex Method for Function Minimization,” *The Computer Journal*, vol. 7, pp. 308–313, Jan. 1965.
- [45] K. I. M. McKinnon, “Convergence of the Nelder-Mead Simplex Method to a Nonstationary Point,” *SIAM J. Optim.*, vol. 9, pp. 148-158, May 1998.
- [46] S. He, E. Prempan, Q. H. Wu, “An improved particle swarm optimizer for mechanical design optimization problems,” *Engineering Optimization*, vol. 36, no. 5, pp. 585–605, Jan. 2004.
- [47] R. A. Krohling, E. Mendel, M. Campos, “Swarm algorithms with chaotic jumps for optimization of multi-modal functions,” *Engineering Optimization*, vol. 43, no. 11, pp. 1243–1261, Mar. 2011.
- [48] K. Deb, A. Saha, “Multi-modal optimization using a bi-objective evolutionary algorithm,” *Evolutionary Computation*, vol. 20, no. 1, pp. 27–62, Mar. 2012.
- [49] C. A. C. Coello, N. C. Cortés, “Hybridizing a genetic algorithm with an artificial immune system for global optimization,” *Engineering Optimization*, vol. 36, no.5, pp. 607–634, Feb. 2004.
- [50] D. Whitley, “A genetic algorithm tutorial,” *Statistics and Computing*, vol. 4, pp. 65–85, Jan. 1994.
- [51] R. A. Rutenbar, “simulated annealing algorithms: An overview,” *IEEE Circuits and Devices Magazine*, vol. 5, pp. 19–26, Jan. 1989.
- [52] S. Kirkpatrick, “Optimization by simulated annealing: Quantitative studies,” *Journal of Statistical Physics*, vol. 34, pp. 975–986, Nov. 1984.
- [53] J. Fu, L. Lei, G. Zhou, “A parallel Ant Colony Optimization algorithm with GPU-acceleration based on All-In-Roulette selection,” *Advanced Computational Intelligence*, pp. 260–264, 2010.
- [54] P. O. Persson, “Mesh generation for implicit geometries,” Ph.D. dissertation, Dept. Mathematics, Massachusetts Institute of Technology, 2004.

- [55] E. Meijering, M. Unser, "A note on cubic convolution interpolation," *IEEE Trans. Image Processing*, vol. 12, no. 4, pp. 477–479, Apr. 2003.
- [56] S. A. Dyer, J. S. Dyer, "Cubic-spline interpolation," *IEEE Instrumentation & Measurement Magazine*, vol. 4, no. 1, pp. 44–46, Aug. 2001.
- [57] C. De Boor, *a practical guide to spline*, York, berlin, Heidelberg. Springer-verlag, 2001, pp. 171–204.
- [58] R. Singh, A. M. Gole, P. Graham, S. Filizadeh, C. Muller, R. Jayasinghe, "Grid-processing for optimization based design of power electronic equipment using electromagnetic transient simulation," *IEEE Canadian Conf., on Elec., & Comp., Eng.*, May 2012, pp.1-6.
- [59] J. Z. Zhou, A. M. Gole, "Tuning of Converter Controllers in a VSC-HVDC Grid," *Cigre Conf.*, Mar. 2014, pp.1-8.
- [60] A.Y. Goharrizi, R. Singh, A.M. Gole, S. Filizadeh, J.C. Muller, R.P. Jayasinghe, "A Parallel Multimodal Optimization Algorithm for Simulation-Based Design of Power Systems," *IEEE Trans. on Power Delivery*, vol.30, no.5, pp.2128-2137, Oct. 2015.
- [61] A.Y. Goharrizi, S. Filizadeh, A.M. Gole, R. Singh, "Simulation based optimal design and sensitivity assessment of a vector-controlled induction motor drive using a multi-modal optimization algorithm," *IEEE 16th Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2015, pp.1-5.