Simulation-Based Design of Multi-Modal Systems

By

Farhad Yahyaie

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

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

July 2010

Abstract

This thesis introduces a new optimization algorithm for simulation-based design of systems with multi-modal, nonlinear, black box objective functions. The algorithm extends the recently introduced adaptive multi-modal optimization by incorporating surrogate modeling features similar to response surface methods (RSM). The resulting optimization algorithm has reduced computational intensity and is therefore well-suited for optimization of expensive black box objective functions. The algorithm relies on an adaptive and multi-resolution mesh to obtain an initial estimation of the objective function surface. Local surrogate models are then constructed to represent the objective function and to generate additional trial points in the vicinity of local minima discovered. The steps of mesh refinement and surrogate modeling continue until convergence criteria are met. An important property of this algorithm is that it produces progressively accurate surrogate models around the local minima; these models can be used for post-optimization studies such as sensitivity and tolerance analyses with minimal computational effort. This algorithm is suitable for optimal design of complex engineering systems and enhances the design cycle by enabling computationally affordable uncertainty analysis. The mathematical basis of the algorithm is explained in detail. The thesis also demonstrates the effectiveness of the algorithm using comparative optimization of several multi-modal objective functions. It also shows several practical applications of the algorithm in the design of complex power and power-electronic systems.

Acknowledgments

I would like to express my most sincere gratitude to my advisor, Prof. Shaahin Filizadeh, who was most helpful to me in a variety of ways. He has been decisive, highly supportive and more importantly, a good friend. His perceptive comments and unfailing support have helped me a lot in my academic growth. I am so honoured and privileged that have had this opportunity to work with him. This thesis truly reflects his inspired ideas and wisdom which he shared so generously with me in our many collegial discussions.

I would also like to thank my committee members who kindly agreed to take part in this process. I learnt a great deal from Prof. Ani Gole, the internal examiner, who shared his interesting experience through stimulating conversations in coffee breaks. I also want to thank Prof. Nariman Sepehri, the external examiner, for taking the time and effort to read my thesis and to take part in the examination process.

The Manitoba HVDC Research Centre and the Natural Sciences and Engineering Research Council of Canada (NSERC) both provided financial support, for which I am very grateful. Staff of the Manitoba HVDC Research Centre, especially Mr. Rohitha Jayasinghe and Mr. Farid Mosallat, shared generously their knowledge and I enjoyed my time with them.

I would also like to thank all the staff, students and my friends at the University of Manitoba and Winterpeg who made the life more joyful and rewarding. I appreciate suggestions from my friend Ehsan with whom we had so much great time. I also thank Maziar who enthusiastically answered my questions and Udana for his FORTRAN help.

I want to thank my cousin Iman, who is now as close as a brother, for without his support I would not have been able to come to this university.

Last but not least, my wonderful parents and family deserve my sincere gratitude for supporting me in my academic goals and in every other way.

To my wonderful

Parents & Family

for a debt I can never repay

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Design of complex engineering systems is often done through simulation-based optimization, where a nonlinear optimization algorithm steers a sequence of simulations of the underlying system towards an optimized set of parameters. The typically high cost of real world experiments, the proliferation of fast computing instruments (which in turn expedite the computer implementation and simulation of systems), and the inclination towards detailed modeling of systems to predict their transient and steady state behaviour are the main reasons for the increasingly popular trend of computer modeling and simulation-based design.

This approach has been used extensively in the design stage of costly systems, such as finite element-based design of automotive systems [1], computational fluid dynamics applications in aircraft design [2], or design of combustion engines [3]. Particularly in the field of power systems, where a large number of components have intricate dynamical interactions, this approach has attracted much attention [4, 5]. A common feature among all these engineering systems is that formulation of an explicit objective function is prohibitively difficult. This necessitates that the design objective function be evaluated through simulation. This class of optimization problems is often referred to as black box optimization, implying that a *simulation model* is used in lieu of an unattainable *explicit formulation* of the objective function. Depending on the complexity of the engineering

system at hand, simulations can be computationally intensive, thus leading to lengthy and resource-demanding design cycles.

Designers face two major challenges during the simulation-based optimization of expensive black box systems. Firstly, the intensity of the simulation-based design can be exacerbated by the complexity of the nonlinear optimization algorithm. In other words, an optimization algorithm that requires a considerable number of (time-consuming) simulations does not serve as a strong candidate for simulation-based design. Secondly, once an optimal solution is obtained, it must be determined whether there are other local optima that may have superior performance. The latter is a property of the design objective function and is referred to as multi-modality.

The bulk of the research on the optimization of multi-modal functions has tended to focus on finding the global optimum [6-8]. This is a natural tendency as the global optimum, by definition, has the most superior performance. From a practical point-of-view, however, it is desirable to have access to not only the global optimum but also as many local optima as possible (ideally all of them). A number of considerations contribute to this desire, including:

- in the physical implementation of the optimized parameters, inevitable tolerances may cause small deviations in the actual values, thus slightly perturbing the optimum. For example electronic components are manufactured with specific tolerance levels, and they are also affected by external conditions such as aging and temperature. In the case of a highly sensitive local (or global) optimum, such deviations may lead to significant degradation of the objective function.
- despite its globally best performance, a global optimum may indeed be comprised of parameter values that are deemed impractical or prohibitively expensive, thus limiting their adoption in practice. For example, a local optimum with acceptable performance and modest cost may be favoured over a costly global solution with marginally better performance.

It is therefore desirable to develop optimization algorithms capable of determining all local optima of a given black box objective function. In the optimization of expensive

objective functions, it is further desirable to maintain a low computational intensity to expedite the design process. Once a simultaneous search for the local optima (including the global) is completed, the designer can examine and scrutinize the solutions with consideration of factors such as sensitivity and cost to select the most suitable solution.

1.1 Motivation

Modern electric power networks are strategic infrastructure upon which today's technologically advanced society relies for meeting its growing energy demand. It is therefore essential to ensure that electric power systems are designed and operated in a fashion that allows them to continue to supply energy securely, steadily and with the highest quality.

At the design stage of a modern power network, particularly when power-electronic equipment are embedded in the system, important decisions need to be made that include selection of suitable network configurations, control system architecture and parameter tuning, to name a few. The purpose is to ensure that the equipment will work as intended and their interactions will not adversely affect the entire system. Conventional design and analysis methods, which rely heavily on simplified analytical methods and often treat subsystems in isolation, find stringently limited applications in the design of today's power systems. Alternatively designers tend to use highly accurate computer simulation tools to assess a new design and study interactions among network elements. There is therefore a growing demand for simulation tools that offer advanced design-support facilities for tasks such as parameter tuning, sensitivity analysis and statistical design assessment. In this simulation-based approach the network is treated as a black box whose dynamic behaviour is described through time-domain simulations rather than simplified closed-form analytical equations.

One design-support tool that has shown great promise is the optimization-enabled transient simulation, in which a nonlinear optimization algorithm is interfaced with an electromagnetic transient simulation tool. Search for the optimal system parameters is conducted by the optimization algorithm, which feeds the simulator with trial parameters

selected intelligently according to the cumulative history of the previous simulations. It has been shown that this approach is a rapid and reliable way for optimal design of highly complex systems.

Moreover it is important to allow the designer of advanced power electronic equipment to have access to all possible optimal solutions of a given design so that the solution that offers the most desired overall performance is selected with scrutiny.

1.2 Problem Definition

Electromagnetic transient simulation programs, also referred to as emtp-type programs, are extensively used to mimic the behaviour of power systems components in both transient and steady state periods. They can precisely describe the system performance if a sufficiently small simulation time step is chosen. A model of a large power system can then be developed to study and predict its real world response to certain contingencies; due to the high level of details considered and also fine simulation time-steps emtp-type simulations tend to be computationally intensive. This implies that such simulations take a long time to complete.

The designer, on the other hand, normally requires several simulation runs to judiciously select the control parameters of a system which satisfy the design objectives. For example, design of control systems can be tremendously cumbersome and become an arduous task if numerous control parameters and objectives need to be optimized for a large power system with nonlinear components interacting with each other. *Multiple-run* feature, which has been offered through some of simulation programs [9], can rectify this problem by automatically varying the design parameters (e.g. linearly or logarithmic) and storing the user's pre-defined objectives. The designer thereby is able to decide on the best suitable solution. Another possible solution is the employment of random search algorithms such as Monte Carlo method which are capable of randomly searching for an optimum solution [10].

None of the mentioned methods is computationally efficient since a large number of simulations are conducted with no formal and adaptive search approach. Alternatively,

optimization techniques can be deployed to strategically select the design parameters. The evaluation of the objectives would specify the conformity of the results. A design problem therefore can be stated as an optimization problem; solutions with closer conformity to design objectives generate smaller objective function evaluations.

A design problem is often associated with restrictions (constraints) on the parameters; e.g. positivity of a control's gains and circuit elements' sizes or some strict conditions on the feasible operating region. The following general formulation is used for the minimization of a function $f(\mathbf{x})$, subject to equality and inequality constraints $h_i(\mathbf{x})$ and $g_j(\mathbf{x})$:

$$\begin{aligned} &\text{minimize } f(\mathbf{x}) \\ &\text{subject to:} \\ &\quad h_i(\mathbf{x}) = 0 \qquad i = 1, \dots, m \\ &\quad g_j(\mathbf{x}) \leq 0 \qquad j = 1, \dots, p \end{aligned} \tag{1.1}$$

where $\mathbf{x} \in \mathcal{R}^n$; $f: \mathcal{R}^n \to \mathcal{R}$; $h_i: \mathcal{R}^n \to \mathcal{R}$; $g_j: \mathcal{R}^n \to \mathcal{R}$; and $m \leq n$.

Several local optimization techniques have been offered by emtp-type programs [11-13]. However, as stated earlier, it is always preferable to access all local minima of a given design. The research presented here aims to devise an optimization algorithm that enables the designer to (i) detect and estimate all local minima, (ii) adhere to constraints, and (iii) incorporate a-priori knowledge into the search for local minima.

1.3 Research Objectives

It is often desired to determine whether the optimal solution obtained using the optimization-enabled transient simulation is a local or the global optimum of the system under consideration, as designers often prefer the global optimum for its superior performance, e.g. a power converter control scheme with the lowest cost. Conventional nonlinear optimization algorithms, such as nonlinear simplex and gradient-based methods are essentially local optimizers and do not necessarily provide the global solution.

Specialized global optimization algorithms exist; however, their search tends to be overly tedious and resource-demanding. Moreover, it is often desired to have access to both the local minima and the global minimum solutions. The reason is that for practical implementation of optimally-designed equipment for example, a local optimum with less sensitivity to varying operating conditions is preferable to a highly sensitive global minimum, although the global optimum has a nominally superior performance.

The proposed research therefore aims to develop an optimization method for use with an electromagnetic transient simulator, which is capable of finding both local and global minima. The designer can then assess all possible optimal solutions with a view to select the one(s) that not only satisfy the design objectives but also provide low sensitivity, resulting in optimal performance even under varying operating conditions. Another prime consideration in the development of such an algorithm is to ensure that its computational intensity remains low to allow expedited design cycles.

In order to enable detection of multiple minima, trial parameter settings will be generated carefully for use in the simulation model of the system under consideration. Once an initial estimation of the shape of the design objective function is obtained, areas likely to contain local minima will be detected and localized. In order to lower the computational intensity of the algorithm, regression functions will be used to form analytical approximations of the objective function in the localized areas. These approximating functions will be improved in the subsequent iterations of the algorithm leading to the discovery of all local minima (including the global). The algorithm is a hybrid combination of the existing response surface methods and newly developed mesh-based multi-modal optimization algorithms. An added benefit of the proposed method is that upon convergence the approximating functions will readily provide a direct means to evaluate the sensitivity of the optimal solutions.

The proposed research will develop a notably-improved design-support tool for the simulation-based design of complex power and power-electronic systems. This hitherto unavailable ability allows designers to optimize highly complex systems with ease and confidence without recourse to compromised analytical methods. Aside from several mathematical evaluations, the algorithm will be interfaced with the PSCAD®/EMTDC™

transient simulator (developed by the Manitoba HVDC Research Centre) for the appraisal of the efficiency and applicability of the algorithm in real world power system design cases.

1.4 Thesis Organization

As highlighted earlier, the thesis aims to develop a new optimization algorithm to be deployed for discovering all local minima of a black box objective function, particularly for simulation-based design purposes. A comprehensive description of the proposed algorithm is presented in the thesis and also to demonstrate its effectiveness several mathematical and real world design problems are tested.

Following the introduction presented in this chapter, Chapter 2 reviews the research that has been conducted in the area of simulation-based design. Description of each method and drawbacks related to each of them are presented. The response surface methodology, as the backbone of the proposed method, is presented in the second section of Chapter 2. An overview of the current multi-modal optimization algorithms together with the newly introduced surrogate modeling concept are described to provide the foundation of the proposed algorithm, which is described in the next chapter. Chapter 2 will be concluded by the challenges associated with the simulation based design of power systems.

Chapter 3 illustrates each step of the proposed algorithm, namely mesh generation, localization and mesh refinement and surrogate modeling. It provides in-depth information regarding each stage of the algorithm. The step-by-step pictorial examples used in this chapter would help the reader to more clearly understand the procedure and facilitate the user to suitably adjust the control parameters of the program such as weightings and penalty factors (described later in this chapter) to achieve its desired performance. Previously introduced mesh-based multi-modal optimization algorithm is partially described in this chapter.

Chapter 4 is devoted to the properties of the proposed algorithm. Use of weighting factors to control the fitting of the surrogate models in different regions is a distinct

feature of the proposed algorithm and is depicted in the first section. Furthermore applying other constraint is also possible by introducing penalty parameters in the fitting stage of the algorithm. To expedite the procedure, parallel processing is proposed and some other characteristics of the algorithm to work more efficiently and even faster are introduced in this chapter.

Although finding only the global minimum is not our ultimate goal, it is shown that in case of interest in the global optimum the algorithm is capable of locating it rapidly.

Four different studies are done in Chapter 5 to examine the features of the proposed algorithm. In the first section, the computational burden of the proposed algorithm is compared with the recently introduced mesh-based multi-modal optimization algorithm. It is shown that the new algorithm is much faster in the subsequent iterations in the course of optimization of a nonlinear objective function. Surrogate models developed by the algorithm are then tested to check their accuracy in representing the function. A mathematical example is used in the third section of this chapter to verify the effectiveness of the force displacement technique, which is employed in the mesh generation and refinement stages of the algorithm. The last section of this chapter employs several well-known benchmark multi-modal functions to study the effectiveness of the algorithm in detecting all the local minima.

In Chapter 6, three different power electronic design examples are presented. It starts with the design of filters and tuning the controller gains for a simple dc motor drive. A highly complex induction machine drive system is exercised following that section. The system operates under an indirect vector control strategy and uses a hysteresis current control loop to maintain the desired current for the motor. Similarly, the filter parameters and the gains of the controller are defined as the design parameters of this system. The last section involves tuning the gains of a controller in both rectifier and inverter sides of the Cigre HVDC benchmark model.

Chapter 7 reviews contributions made in the research reported in the thesis; it also provides recommendations for future research and possible extensions of the thesis.

Chapter 2

Background

A design procedure could turn out to be highly challenging where a sophisticated system with intricate nonlinear sub-systems is considered. The design can be prohibitively costly when it involves real world and laboratory prototype experiments. The simulation-based design approach, on the other hand, uses a low-cost computer model of the system; this model closely imitates the system's behaviour and provides an adequately in-depth representation for the system. An explicit mathematical formulation of such systems is generally unattainable and as such, computer models of complex systems are treated as black box functions.

The design can become especially difficult when multiple design objectives are tackled. It is also possible that multiple sets of design parameters satisfy the design objectives (multi-modal systems); the designer therefore has to choose the one which not only satisfies the pre-defined objectives, but also provides an acceptable performance in real world conditions (i.e. an insensitive design). As a result, a design process can easily turn out to be a tedious and resource-demanding procedure involving search for the most suitable set of parameters among multiple sets of qualifying ones.

In this chapter, an overview of the currently available simulation-based design algorithms and their properties are presented; it is followed by a brief explanation of the response surface methodology and the concept of surrogate modeling together with the simulation-based design of power systems.

2.1 Simulation-Based Design

For a given design finding one (or more) set of design parameters that most closely satisfies the design objectives, i.e. minimizes the design objective functions, is desirable. Finding these optimal solutions, however, is not always trivial and the preliminary solution is to study all different combinations of parameters, if at all possible, and decide on the best suited one. High cost and potential hazard of real world experiments encourages the designer to develop a computer simulation model of the system in a simulation engine that can precisely emulate the behaviour of the real system. Although there might be some inherent inaccuracy in the simulation model due to either lack of knowledge in modeling all the aspects or deliberately ignoring some, computer modeling is viewed as a reliable step in design.

Another reason that leads to adoption of a simulation-based approach to design a complex system [14] is inaccessibility of an analytical formulation of the objective function, which thwarts direct mathematical evaluation of the design objective function.

Stating all the above reasons, simulation is still the only cost-effective approach for design of complex systems. The *simulation experiment*, i.e. a run of the *simulation model* for a specific combination of design parameters, is used in lieu of an actual experiment on the real system. It is often required to conduct several simulation experiments before an optimal solution is obtained.

Two general categories of the simulation-based design approaches are available to select design experiments, namely (i) adaptive and (ii) non-adaptive approaches. An adaptive approach deploys results of the past simulations to judiciously determine the next set of design parameters; the benefit is a profound impact on the length of the design cycle as well as the accuracy of the final results. A non-adaptive approach, on the other hand, only uses a pre-defined set of parameters. Multiple run and random search methods, which are widely used in simulation programs, are examples of non-adaptive algorithms [9]. The simulation engine is run several times, each corresponding to a set of parameters, and a numerical index is assigned to each run signifying the conformity of the results. The optimal solution then can be identified by analysing those indices.

Fig. 2.1 depicts the two different approaches used in the simulation based design. As shown, in an adaptive search approach the simulation results encapsulated in a single value (OF($\mathbf{x}_i$)) are fed back to the algorithm in order to effectively choose the next set of design parameters $\mathbf{x}_i$. In the following subsections, different methods under the general category of adaptive design approaches are described.

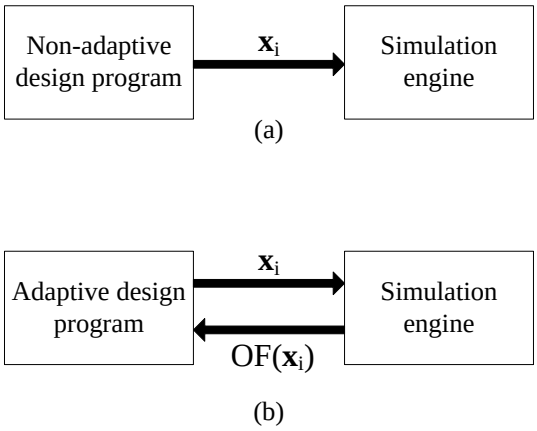


Fig. 2.1. Simulation-based design approaches

2.1.1 Gradient-based search methods

The gradient of the objective function (the *simulation experiment* outcome) at a specific point together with the objective function evaluation at that point are needed in a category of *simulation optimization* called "gradient-based optimization algorithms" in order to determine the next design parameters [15]. The general formula of a gradient-based optimization is as follows:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{s}\left(\mathbf{x}^{(k)}\right) \tag{2.1}$$

Equation (2.1) shows how the next input parameter vector ($\mathbf{x}^{(k+1)}$) is defined conducting a line search in the direction of $\mathbf{s}(\mathbf{x}^k)$ from the current point ($\mathbf{x}^{(k)}$) using steps of α^k. It should be noted that $\mathbf{x}$ and $\mathbf{s}$ are *N*-dimensional vectors, whereas α is a scalar.

This category of optimization algorithms requires an explicit definition of the objective function to allow calculation of derivatives; a luxury that is rarely available in

practical designs. Although estimation of derivatives using numerical techniques is also possible, it is often accompanied by a large number of intermediate simulations and is also prone to numerical inaccuracies. Despite these observations, there have been simulation-based design methods employing gradient-based algorithms [4].

There are different methods to estimate the gradient of an objective function, the crudest of which is the *method of finite differences* [16]. Two approaches of the *forward differences* and the *central differences* need at least N+1 and 2×N simulation runs (for an N-variable case) respectively to estimate the gradient [17]. Despite the ease of implementation, this technique requires high computational costs and has slower convergence rate when used in a stochastic approximation algorithm (see Appendix A) [18].

Other methods such as *likelihood ratios* [19], *perturbation analysis* [20-22] and the *frequency domain method* [23] also exist; even though they require fewer number of simulation runs, the complexity of their implementation is a major hindrance to a wide range of applications.

2.1.2 Heuristic methods

This category comprises techniques that only use the objective function values to pursue the course of optimization (also known as direct search methods) and therefore are particularly suitable for the purpose of simulation-based optimization, where an explicit formulation of the objective function in terms of design parameters is inaccessible.

Several optimization algorithms are available in this category including Nelder and Mead's simplex search [24-27], Genetic Algorithm (GA) [28, 29], Evolutionary Strategies (ES) [30, 31], Simulated Annealing (SA) [32, 33] and Tabu Search (TS) [34-36]. GA, ES and SA are suited for global optimization; however, they require a large number of function evaluations. TS is used for solving combinatorial problems and, like the other algorithms, is notorious for high computational intensity. On the other hand, Nelder and Mead's Simplex method is used as a local optimizer, but its main drawback is the assumption of a convex feasible region.

2.1.3 Statistical methods

In cases with limited number of feasible input parameters' sets, some statistical approaches can be employed to explore the minimum point.

Two of the frequently used techniques are: (i) ranking and selection; and (ii) multiple comparisons [37, 38]. The ranking-and-selection technique determines some specific criterion, such as choosing the best with some pre-specified confidence level, and then derives a statistical procedure, usually sequential, to meet that criterion. On the other hand, multiple-comparison procedures specify the use of certain pair-wise comparisons to make interferences in the form of confidence intervals. This technique is not necessarily a sequential one.

These techniques are easy to implement; however, they cannot be used in a wide range of application. As stated earlier, the number of parameters' set should be limited for such a purpose.

2.2 Response Surface Methodology

Response surface methodology (RSM) uses a collection of statistical and mathematical techniques to fit one or several functions of simple and explicit forms to a complex and explicitly unknown function to describe its global or local behaviour. In this section, RSM and its application in the engineering design problems are discussed.

2.2.1 Introduction

The RSM seeks to find a closed-form formulation for the relationship between several design variables, also referred to as *independent variables*, and one or more *response variables*. These response variables can be either the performance measure or the quality characteristic of a product or a process [39].

RSM can be used to study, improve, and modify a product design and has been used for the simulation-based design purposes. Its aim is to fit a series of regression models to

a response variable by running the simulation case at different input parameters sets and optimizing their resulting regression model. This procedure is often done using simple functions such as first order or second order polynomials. Higher order functions also can be used; however, as it will be shown later, they need larger number of simulation runs in their fitting process.

RSM also has generated considerable interest when tackling robust design targets [40]; that is, a design process with the least amount of sensitivity to the environmental changes; e.g. humidity, aging or temperature.

The purpose of the RSM, in general, is to develop an appropriate approximation model for a system, and optimization methods to find values for the independent variables, which result in desired values for the corresponding response variables.

2.2.2 Mathematical review

In general, a system with k controllable input parameters (ξ_i) can be expressed as the following equation:

$$y = f(\xi_1, \xi_2, \dots, \xi_k) + \epsilon \tag{2.2}$$

where ϵ represents the unknown sources of variation in f, called error; including measurement errors, noises or the effect of other variables. Usually the error is considered as a statistical error with a normal distribution around zero. In this study for the sake of simplicity, it has been assumed that simulation results of a fixed set of parameters to be consistent (devoid of any stochastic impacts); this implies ignoring of the ϵ term. Moreover, note that the response function f is not known in advance for most practical cases.

The controllable parameters (ξ_i) in Eq. (2.2) are called *natural variables* and are expressed in terms of natural units. They can be converted to coded variables (x_i) which are dimensionless. Eq. (2.3) represents the amalgamation of the general form of the Eq. (2.2) with the above two assumptions:

$$\eta = f(x_1, x_2, \dots, x_k) \tag{2.3}$$

First order polynomials can be used to approximate the response function in a small region. This *main effect model* for a two-dimensional case is shown in the following equation:

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \tag{2.4}$$

To incorporate the curvature of the response function in the model an *interaction* term between the two variables can also be employed, as follows:

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 \tag{2.5}$$

However, this model is still not precise enough to properly describe some cases; for more complex curvatures embedded in a true response surface, the second-order model with the following format has been proposed:

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 \tag{2.6}$$

There are several reasons that justify the use of the above second-order model for RSM, including:

1. its flexibility to generate twisted surfaces by applying different coefficients to $\beta_i's$ in the corresponding formula.
2. the ease of implementation. Polynomials are linear equations in terms of $\beta_i's$; using the *linear regression analysis* it is easy to obtain those coefficients; i.e. least square method can be employed as a straightforward approach.
3. it has been shown that its performance is acceptable in most of the practical cases [40-42].

The general forms of the first-order and second-order models are presented in the next equations:

$$\eta = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j \tag{2.7}$$

$$\eta = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k} \beta_{jj} x_j^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j \tag{2.8}$$

Rarely are higher order models used. Third or higher order models are employed when second-order models are found to be inadequate [43]. This, however, may involve a large number of parameter sets.

First and second order models resemble the Taylor series expansion of the response variable around a point. As an illustration, the following equation shows the first-order derivatives which correspond to the first-order polynomial model proposed for the RSM:

$$\begin{aligned} f(x_1, x_2, \dots, x_k) & \\ & \cong f(x_{10}, x_{20}, \dots, x_{k0}) + \frac{\partial f}{\partial x_1}\Big|_{\mathbf{x}=\mathbf{x}_0} (x_1 - x_{10}) \\ & + \frac{\partial f}{\partial x_2}\Big|_{\mathbf{x}=\mathbf{x}_0} (x_2 - x_{20}) + \cdots \\ & + \frac{\partial f}{\partial x_k}\Big|_{\mathbf{x}=\mathbf{x}_0} (x_k - x_{k0}) \end{aligned} \tag{2.9}$$

where $\mathbf{x}$ is the vector of independent variables and $\mathbf{x}_0$ is the vector of variables at a specific point $(x_{10}, x_{20}, \dots, x_{k0})$.

2.2.3 Design procedure

The RSM comprises of several stages in a design procedure. At the first step, which is called *screening experiment*, the number of necessary input parameters is defined. A set

of experiments is conducted to determine all the factors that have considerable impacts on the response variable.

The objective of the first step is to reduce the number of independent variables prior to the investigation process and retain only those variables that have a significant impact on the outcome.

Following screening experiment, the experimenter should study how close the response of the current set of independent variables to the optimum is; that is, whether the current point satisfies the conditions for optimality or it is remote from the desired optimum. The first-order model then will be used together with an optimization technique called the *steepest descent* method to define a new set of control parameters that is toward the optimum solution.

The last step in RSM begins when the results of the experiment are marginally close to the optimum. Second-order or higher-order models can be used in this stage to represent the exact behaviour of the function around its optimum.

Since the true response of a system usually has curvatures around its optima, second-order models suitably can be fitted in a small vicinity of the optima. Once the second-order models are obtained, they can be used to determine the optimal conditions for the process.

The RSM in general requires fewer number of function evaluations compared with many gradient-based methods [19]. The sequential nature of the RSM in model fitting and its subsequent iterations provides the potential exploitation of this technique for the simulation based design purposes.

Adaptive response surface methodology (ARSM), which has been introduced recently, is capable of finding the global minimum of a system within a reasonable number of function evaluations [7]. Nevertheless, this method has failed to discover the global minimum of generally concave functions.

2.3 Multi-Modal Optimization

While there are several local optimizer algorithms available [24, 44-46], only a few studies have reported multi-modal optimization [47-49]. These techniques aim to find all or the maximum possible number of optima.

Multi-modal optimization techniques can be crucially helpful in engineering design problems in which some solutions either may not be practically possible or incur high cost. In such cases, having access to both local and global optima of a function, the designer can proficiently decide on the best suited one.

Although it is possible to employ some of the classical local optimizer techniques with different starting points hoping for new solutions upon convergence, it does not guarantee convergence to the global minimum nor is it computationally efficient.

2.3.1 A review of current methods for multi-modal optimization

Dealing with multi-modal functions is always a challenging task since the global minimum is unlikely to be found in case of a high number of attractors. Evolutionary algorithms (EA's) however can be successful due to their population based and stochastic nature. As a result of the second, they are not strongly influenced by the initial population.

Goldberg and Richardson first employed EA to handle multi-modal functions [50]. *Niche-preserving technique* was introduced in their work and the new genetic algorithm was capable of maintaining multiple optimal solutions. Moreover, a modification of simulated annealing was introduced in [51] which also made it possible to access all local and global minima. However, it is declared that it requires unreasonably large number of function evaluations.

Niching methods are added to the conventional GA's to maintain the diversity of the population and reduce the effect of genetic drifting, which may be caused by the selection operator [52]. A niching method should be able to find and maintain multiple diverse solutions whether they are from the same or different fitness functions.

Several niching methods are introduced in [52-56]; each method has different capability for finding, locating and maintaining multiple local minima in terms of the computational time and the number of peaks maintained. A comparative study on this topic together with a new modified approach is presented in [47].

A hybrid algorithm combining simulated annealing, TS and a descent method has been proposed recently. It is shown that the algorithm is fast, flexible and easy to use [57]; it is capable of finding most of the equal global minima of a function, albeit this technique hardly discovered any of local minima.

2.4 Surrogate Model-Based Optimization

Although accurate, high-fidelity simulation models can be used for design purposes, they are still time-consuming and may easily hinder the optimization procedure. As stated earlier, most of the conventional optimization techniques require a large number of function evaluations to discover all local minima. This motivates the designer to develop approximation models, also known as *surrogate models*, in order to imitate the behaviour of simulation models as closely as possible, while keeping the computational burden of the process at the lowest level [58].

Surrogate-based modeling and optimization has attracted much interest in recent years [59-61] and a large number of research and studies have been conducted in this area.

The main steps of the surrogate model-based design procedure are illustrated in Fig. 2.2.

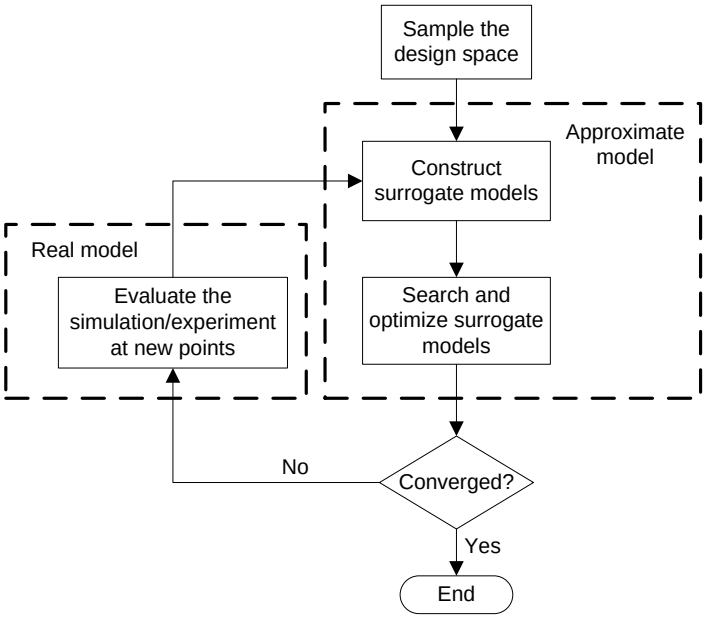


Fig. 2.2. A surrogate model-based design procedure

As shown, the first step is to define a number of control input parameters' sets in the design space and evaluate the computationally intensive model (real model). *Design of experiment* (DOE) has been used in this step to define such set of points.

There are several DOE techniques available; e.g. Latin hypercube design; full factorial design; fractional design; and orthogonal main effect design, to name a few. A comprehensive study on each technique and their computational advantages and shortcomings are presented in [62].

Selection of a surrogate model that can accurately characterize a system under consideration and the determination of its corresponding parameters are the main two controversial issues which are addressed in [63]. There are several options available for surrogate models; polynomial regression models, Kringing modeling, and radial basis functions are commonly used as surrogate models.

Once a surrogate model has been constructed, optimization technique such as Genetic Algorithm can be employed to determine new experimental sets. Note that surrogate models are used to estimate the expensive objective function and hence it is computationally expedient to use techniques that require large number of function evaluations in this stage.

The resultant points will be evaluated using the real simulation model and then will be added to the current set of samples. The surrogate models are modified accordingly and will be used in the next iteration. This procedure will be continued until certain termination criteria being met.

2.5 Optimization-Based Simulation Design for Power Systems

Electricity is a convenient, fast supplying, and the most common source of energy. Modern electric power networks are used to generate, transmit and distribute electricity in today's technologically advanced society. It is therefore necessary to ensure that all components of the electric power network are operating properly providing a secure and stable source of energy.

To achieve this, different control strategies have to be employed. However, presence of nonlinear complex components in a power network (e.g. power-electronic equipment) makes the design of such systems exorbitantly difficult and challenging. The purpose of the design is to ensure that the equipment will work as intended and their interactions will not adversely affect the entire system.

2.5.1 Challenges

As stated earlier, conventional design and analysis methods, which rely heavily on simplified analytical methods and often treat subsystems in isolation, find stringently limited applications in the design of today's power systems. Alternatively designers tend to use highly accurate computer simulation tools to assess a new design and study interactions among network elements. There is therefore a growing demand for simulation tools that offer advanced design-support facilities for tasks such as parameter tuning, sensitivity analysis and statistical design assessment.

Electromagnetic transient simulation programs (EMTP-type programs) can accurately model electric power components and represent an in detail modeling of the power network. However, these simulation models are drastically slow and searching for the optimum solution using the available multiple-run feature is a highly time-consuming assignment. Therefore efficient optimization techniques need to be deployed to expedite the design procedure.

The inherent nonlinear switching structure of the embedded power electronic devices in power systems prevents the accessibility of an explicit mathematical formulation for the system and consequently any analytical optimization algorithm such as gradient based methods are not applicable. As a result, a simulation-based approach, which treats the network as a black box whose dynamic behaviour is described through time-domain simulations rather than closed-form analytical equations, has to be employed.

2.5.2 Recent studies and developments

Nonlinear optimization techniques have been linked to the electromagnetic transient simulators and extensively used in the design stage of power systems [4, 64]. The power system simulation program in this case has been considered as a black box function whose mathematical input-output relationship is not accessible; the nonlinear optimization technique is used to supervise this simulation model [65].

Conventional nonlinear optimization algorithms, such as nonlinear simplex and gradient-based methods are essentially local optimizers and do not necessarily provide the global solution. However, designers often prefer the global optimum for its superior performance. As stated earlier in this chapter, dedicated global optimization algorithms exist; but their search tends to be overly tedious and resource-demanding.

In general, it is desired to have access to both local and the global minima of a system, since for the practical implementation of optimally-designed equipment, a local optimum with less sensitivity to varying operating conditions is preferable to a highly sensitive global minimum, regardless of its nominally superior performance.

An adaptive multi-modal optimization algorithm has been proposed recently [66]; it is shown that it has been capable of finding both local and global minima of a black box nonlinear system. The algorithm uses a multi-resolution mesh to locate multiple minima of a system and iteratively enhances the accuracy of the mesh in regions that are likely to have a local minimum. This technique, however, requires a large number of function evaluations and therefore it is not computationally efficient. The optimization algorithm presented in the thesis (Chapter 3) addresses some of these issues and offers further suitability for simulation-based design.

Chapter 3

Description of the Proposed Multi-Modal Optimization Algorithm

A novel multi-modal nonlinear optimization algorithm is proposed in this chapter. The algorithm steers a simultaneous search of the underlying system for multiple local minima and does not require functions that are differentiable or explicit. Upon convergence of the algorithm, the designer can assess all possible optimal solutions with a view to select the one(s) that not only satisfy the design objectives but also provide low sensitivity, resulting in optimal performance even under varying operating conditions. Another prime consideration in the development of such an algorithm is to ensure that its computational intensity remains low to allow expedited design cycles.

This chapter presents a comprehensive step-by-step description of the proposed algorithm. As shown schematically in Fig. 3.1, the proposed algorithm consists of three major steps of (i) initial mesh generation, (ii) localization, and (iii) mesh refinement through surrogate modelling. The algorithm shares some backbone with the recently-developed adaptive multi-modal optimization algorithm [66].

The descriptions to follow will include explicit statements about the improvements introduced in the new algorithm. Without loss of generality, the descriptions that follow are given for a minimization problem.

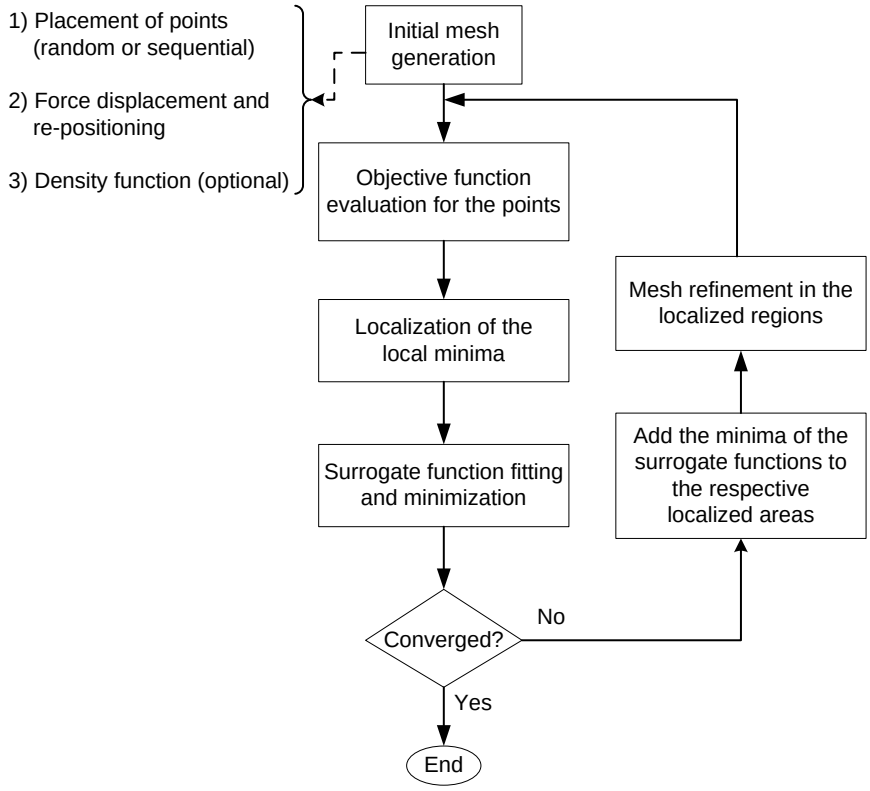


Fig. 3.1. A schematic flow diagram of the proposed multi-modal optimization algorithm

3.1 Mesh Generation

The first stage of the algorithm is to generate a set of initial experiments in the optimization space, through random, sequential or user-specified placement of trial points. Each point represents a combination of the optimization variables. A triangulation method [67] is then employed to create connections between the trial points, thereby creating a raw mesh.

Note that the success of the proposed optimization algorithm in discovering all local minima in the subsequent iterations, to some extent, depends on the density of the initial mesh; a well-populated mesh may require more simulation runs whereas it has a higher likelihood to capture most of the local minima. The designer has to define an affordable

number of initial points that gratifies both the simulation run-time and the required reliability.

A well-defined mesh should offer the coverage of the entire design space. This can be done by a uniform distribution of points. However, the designer may wish to selectively concentrate on some regions of the design space. This necessitates a mesh generation algorithm that has the ability to generate non-uniform meshes.

3.1.1 Description of the mesh generation

Several mesh generation techniques have been proposed recently. Some of the most popular ones are described in [69, 70]. Most of these techniques have complex inaccessible structures; therefore, they are treated as black box functions. A new mesh generation technique has been proposed by Persson [68]; it is simple, effective and easy to implement. It uses the mechanical analogy between a triangular mesh and a two-dimensional truss structure, akin to an interconnected structure of springs.

The algorithm commences with a set of randomly distributed points inside the design space. A triangulation algorithm (such as the Delaunay algorithm) is then used to connect the points (vertex nodes). The Delaunay algorithm prevents overly narrow triangles by maximizing their minimum angles; such triangles fill the convex hull of the initial points while they are not overlapping. Each edge of the triangles (the connection between points) resembles a bar in a truss structure and is shared with at most two triangles in a two-dimensional case. Dictated by the Delaunay algorithm, the circum-circle of each triangle does not contain any other vertex nodes.

For example consider Fig. 3.2, which shows a randomly-spread set of points in a two-dimensional *x*-*y* space subject to the following constraints:

$$-10 < x < 10, -10 < y < 10$$
$$(x-10)^2 + (y-10)^2 > 100 \tag{3.1}$$

Fig. 3.3 shows the initial mesh created by the Delaunay triangulation, in which links are established between points shown in Fig. 3.2.

Once a raw mesh is created, a procedure known as the *force displacement* is employed to further adjust the location of the points with the aim of creating a well-shaped mesh. There are two structurally different forces that will be applied to each point, namely internal and the boundary forces. The former is due to the points directly connected to a specific point, and the latter is used to keep the points inside the boundary.

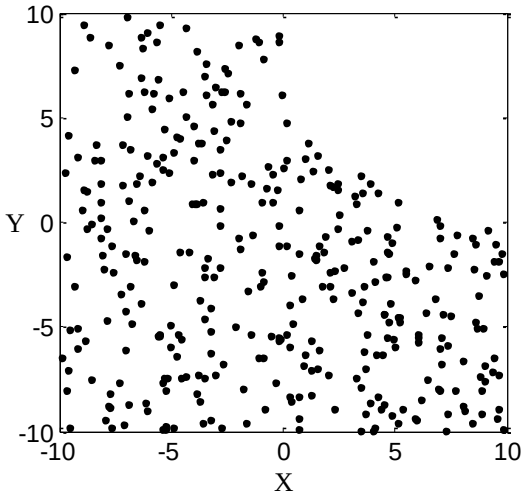


Fig. 3.2. Randomly placed initial points

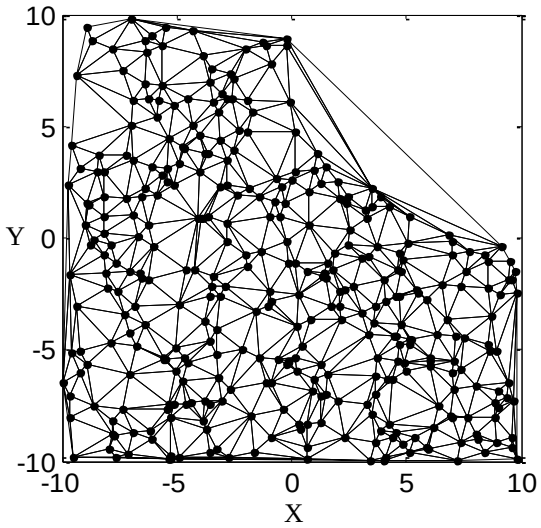


Fig. 3.3. Triangulation and raw mesh

The internal forces totally depend on the position of the points and their relative distances from each other (bars length). In the proposed mesh generator algorithm, points encounter forces when they are too close to each other, similar to the behaviour of like electric charges. These forces are therefore called repulsive and the links between the points are treated as nonlinear spring elements. The following equation formulates the nature of such nonlinear repulsive forces:

$$f(l_{ij}) = \begin{cases} k(L_{ij} - l_{ij}) & l_{ij} \leq L_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where k is the spring constant, L_{ij} is the expected length of the link between points i and j, and l_{ij} is the actual length of the link connecting the two points.

As seen from (3.2) a spring element exerts repulsive forces to its end points only if the two ends are closer than the expected length (L_{ij}). This causes the points to move outwards, tending to *fill up* the space.

In order to ensure that most of the points would experience repulsive forces, the expected lengths of the bars (L_{ij}) have been slightly exaggerated. This will also facilitate to cover the whole design space.

Given the only internal forces, the points do not necessarily stay inside the design space. As a result, inward forces are applied at the boundaries defined by the constraints imposed on the design to ensure that the movement of points does not cause them to leave the specified boundaries. A non-flexible structure illustrated in Fig. 3.4, can be assumed for the boundaries.

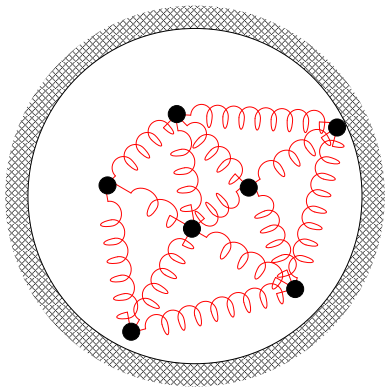


Fig. 3.4. The rigid structure of the boundary

As a result of force displacement, a mesh with a relatively uniform distance between the points and with adherence to constraints is generated. Fig. 3.5 shows the mesh modified by force displacement, in which the points (dots omitted for clarity) are distributed within the boundaries given in (3.2).

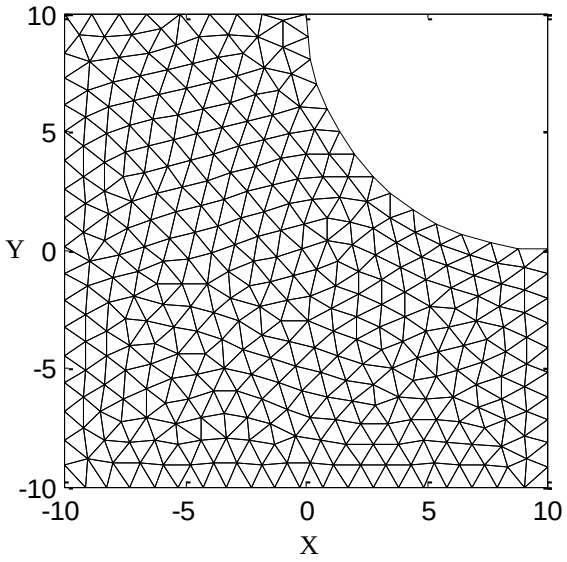


Fig. 3.5. Force displacement

Note that the expected length parameter in (3.2) is a local variable; i.e. it can be assigned different values for different regions of the optimization space. If the designer has a-priori knowledge about the potential neighbourhood of the local minima, or desires

to focus the search in one or more specific areas of the optimization space, it is possible to increase the density of the initial trial points through the use of a *density function.*

The role of a density function is to specify the expected length of links (L_{ij} in (3.2)) in various areas of the search space. By prescribing a smaller expected length, it allows the points in the respective area to come closer to each other before they face repulsive forces. Such areas will, therefore, have a higher density of trial points, thus increasing the resolution of the search for local optima.

It is difficult to determine the exact number of points where non-uniform meshes are tackled; this is mainly due to the variable length of bars at each location. To rectify this, the density function $d(x, y)$ is defined relatively to determine the length of bars in conjunction with an overall value which defines the average of the bars' length; the density function in this case elaborates the distribution of points. As an example, if $d(x, y) = x^2 + y^2 + 1$, the average length of bars around the boundary of the unit circle would be almost twice as the ones close to the origin. The following scaling factor has been used to switch between the relative and actual length of bars in the program:

$$\text{Scaling factor} = \sqrt{\frac{\sum l_{ij}^2}{\sum d(x_i, y_i)^2}} \tag{3.3}$$

where l_{ij} is the actual length of the links as before; and $d(x_i, y_i)$ is the relative expected length obtained at the middle of each link.

Fig. 3.6 shows a non-uniform mesh created by force displacement with a density function that prescribes smaller lengths around the points (-5,+5) and (+5,-5). As seen the initial mesh for such a density function has a higher concentration of points around (-5,+5) and (+5,-5).

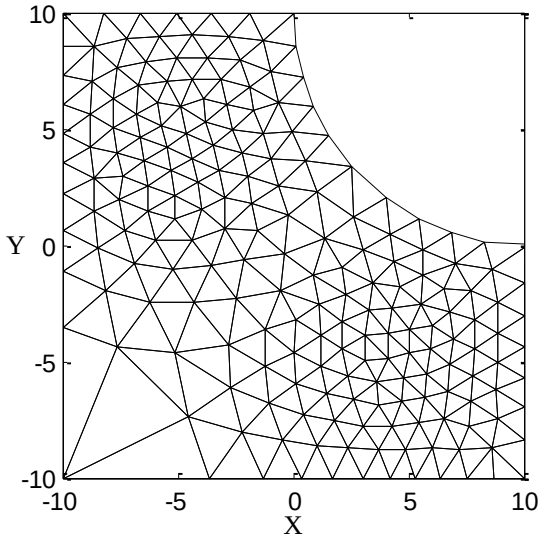


Fig. 3.6. Non-uniform density function

Note that the success of the proposed optimization algorithm in discovering local minima in its subsequent iterations depends, to an extent, on the initial mesh in providing an adequately detailed representation of the objective function; i.e. the location and density of the sampling points. This is more elaborated in the next chapter of numerical assessments when several benchmark functions are used to justify the performance of the algorithm.

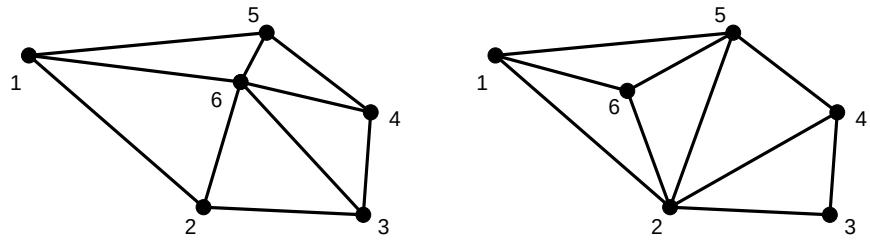
3.1.2 Force displacement

As stated in the previous section, applying internal and boundary forces to the nodes of a mesh-grid is the first step in mesh generation. A well-shaped mesh will be obtained when an equilibrium point is reached; that is, all points are in stationary positions, or:

$$\vec{\boldsymbol{F}}_{total} = 0 \Rightarrow \left\{\forall P_i | \vec{F}_{internal}(P_i) + \vec{F}_{boundary}(P_i) = 0\right\} \qquad (3.4)$$

There are two challenges in solving the above equation: (i) it is a set of nonlinear equations, which is difficult to solve explicitly; (ii) forces are discrete functions, this is because of the changes in the topology of the mesh.

As an example for the second condition, consider Fig. 3.7. In both Figures, points 1 to 5 are kept constant while point 6 can freely move. After a few iterations, the new position of point 6 requires the new configuration in which some joints are added (2-4 and 2-5) whereas some others may be ignored (6-3, 6-4).



(a) (b)

Fig. 3.7. Changing the topology of a mesh

To solve equation (3.4), the following discrete approach based on the forward Euler method is used iteratively:

$$P_i^{n+1} = P_i^n + \Delta t.\vec{F}(P_i) \tag{3.5}$$

A conforming distribution will be achieved when $P_i^{n+1} = P_i^n$, or equivalently:

$$\frac{dP_i}{dt} = \vec{F}(P_i) = 0 \tag{3.6}$$

In order to keep the points inside the boundary, a restriction strategy projects them back as soon as they violate the boundary conditions. This projection is in the same direction as the gradient on the *boundary function*. Moreover, the sign of the boundary function is used to determine points outside the region.

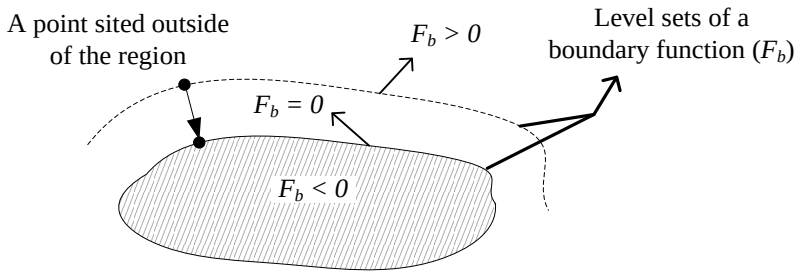


Fig. 3.8. Boundary projection

The mathematical representation of the projection procedure is shown in (3.7).

$$P_i^{new} = P_i^{old} - F_b\left(P_i^{old}\right).\overrightarrow{\nabla F_b} \tag{3.7}$$

where P_i^{old} and P_i^{new} are the previous and current position of a point (P_i) and F_b represents the boundary function. The following figure summarizes the whole procedure of the initial mesh generation:

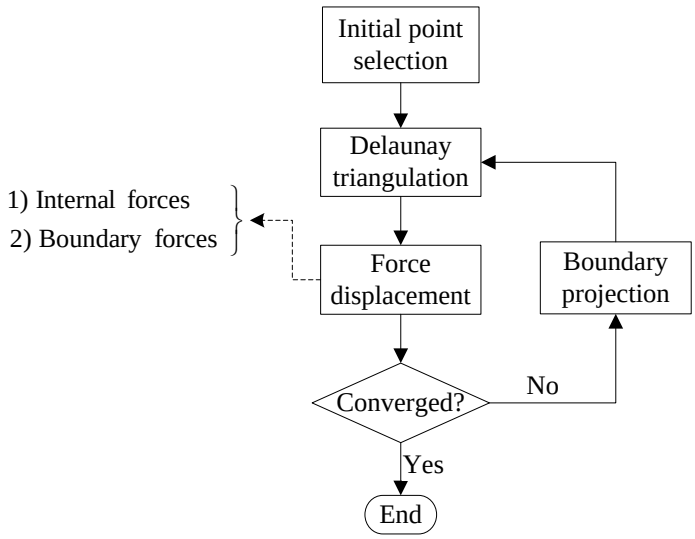


Fig. 3.9. A schematic diagram of mesh generation procedure

3.1.3 Examples of initial mesh generation

In this section, several examples for the initial mesh generation are presented. The purpose is to show the flexibility of the algorithm to create complex meshes. It is shown that any implicit function can be used to determine the boundary region. Also, it is possible to create meshes with different density of points.

Apart from defining one implicit function for the boundary, it is also possible to use combinations of mathematical functions to create sophisticated shapes; i.e. the union; intersection; or relative complement of two or more regions.

Fig. 3.10 shows a number of examples using different boundary and density functions. The following table describes the boundary and density functions related to each case together with the corresponding average length of bars.

Table 3.1. Defining density and boundary functions

Case	L_0	*Density Function*	*Boundary Function(s)*
(a) Circle	1	$d(x,y)=1$	$\sqrt{(x-1)^2+(y-2)^2}-8$
(b) Ellipse	0.3	$d(x,y)=1$	$\sqrt{\left(\frac{x}{2}\right)^2+y^2}-1$
(c) Donut	1	$d(x,y)=1$	$\left(\sqrt{x^2+y^2}-8\right)\cap(3-\sqrt{x^2+y^2})$
(d) Rectangle	1	$d(x,y)=1$	$rec^*\ \cap\ (3-\sqrt{x^2+y^2})$
(e) Circle	0.5	$d(x,y)=e^{\frac{(x-6)^2+(y-4)^2}{100}}$	$\sqrt{(x-1)^2+(y-2)^2}-8$
(f) Sphere	0.5	$d(x,y)=e^{\frac{x^2+y^2+z^2}{20}}$	$f_1=\sqrt{x^2+y^2+z^2}-5$ $f_2=5-\sqrt{x^2+y^2+(z-5)^2}$ $f_{tot}=f_1\cap f_2$

*where *rec* is:

$$f_1=\min\big((x-6),(-6-x)\big)$$
$$f_2=\min\big((y-6),(-2-y)\big)$$
$$rec=-\min(f_1,f_2)$$

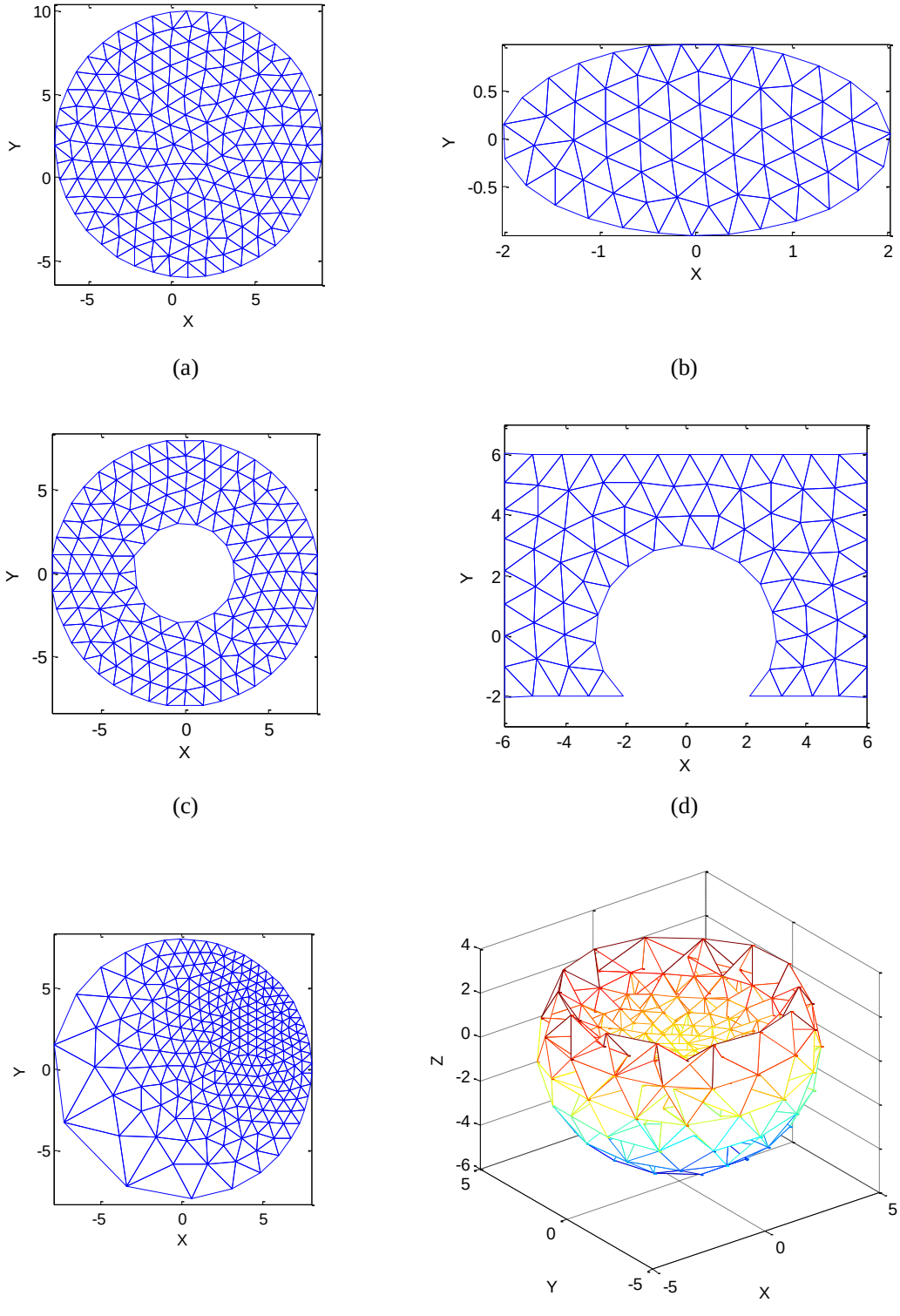


Fig. 3.10. Mesh examples

3.2 Localization

Once an initial mesh is generated, the objective function (OF) for each of its points must be evaluated. This is done either through direct evaluation of an explicit function or through simulation of the underlying system in optimization of black box systems.

As shown in Fig. 3.11, the simulation engine (electromagnetic transient simulation program) is used to evaluate the value of the objective function in this case.

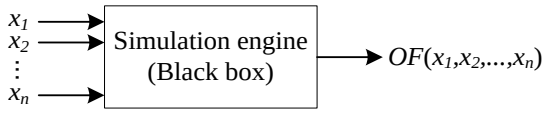


Fig. 3.11. Objective function evaluation

Localization is the stage of the proposed algorithm following evaluation of the OF. The aim of localization is to find areas of the optimization space where local minima are likely to lie. During localization, points that have lower OF evaluations than their neighbouring points (i.e. the ones directly linked to them) are identified. The areas surrounding these localized points are likely to contain local minima and are hence selected for further and finer exploration.

As an example of localization procedure, consider an objective function, which has three local minima at (-4,+5), (+5, -4) and (-6, -6). Using the initial mesh shown in Fig. 3.5, the localization has identified the areas highlighted in Fig. 3.12 as potential vicinities of local minima. The points designated by a circle are the centres of the identified areas and their neighbouring points are shown in squares. The actual local minima of the function are also shown using asterisks.

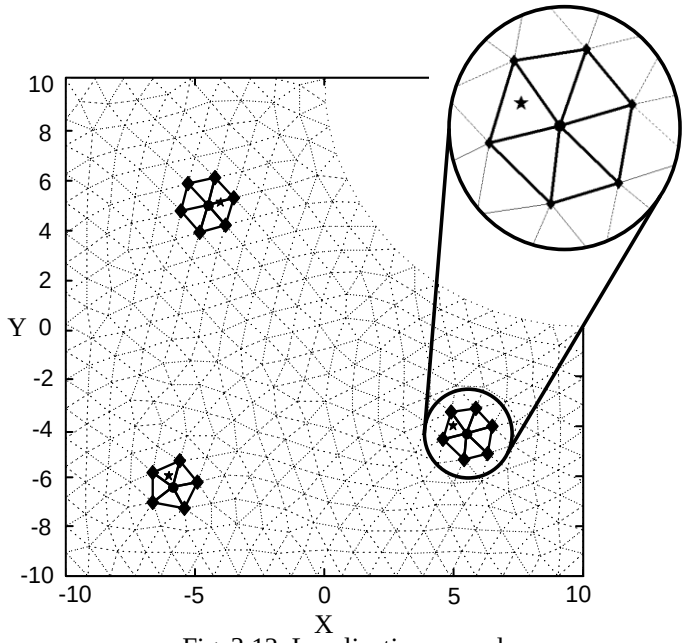


Fig. 3.12. Localization procedure

3.2.1 Implementation considerations

In order to improve the localization procedure, several preliminary considerations are applied. During testing of the algorithm, it was discovered that the following factors can have beneficial impacts on the final results.

a. Treatment of flat areas

If the surface of an objective function contains a flat region, where all points have equal OF evaluations, only one of the points in the region is selected and the search for neighbouring points extends to the ones that have indirect links to the selected point as well. With the inclusion of a small threshold band, this technique can also be used for situations where a region is not *completely* flat but has points with marginally different OF evaluations.

This is illustrated in Fig. 3.13 where three points with OF evaluations residing with a narrow threshold (horizontal lines) are shown. One of the points within the threshold (point A in the figure) is considered and localization extends to the points B and C, which are not directly connected to it.

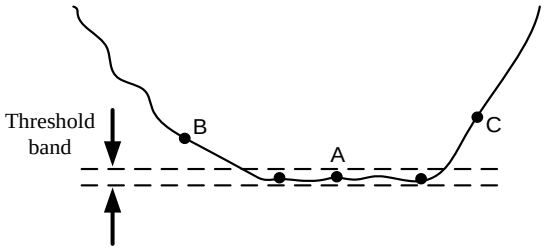


Fig. 3.13. A function with a minimum located in a relatively flat area

b. Speeding up the procedure

The procedure of localization goes through all the points and checks their direct/indirect connected vertices to identify local minima. To expedite this procedure, points located inside the region of uncertainty can bypass the checking cycle. However, their connection is still available and they would be considered for evaluation of points outside this region. The following diagram summarizes the steps of OF evaluations and localization procedure.

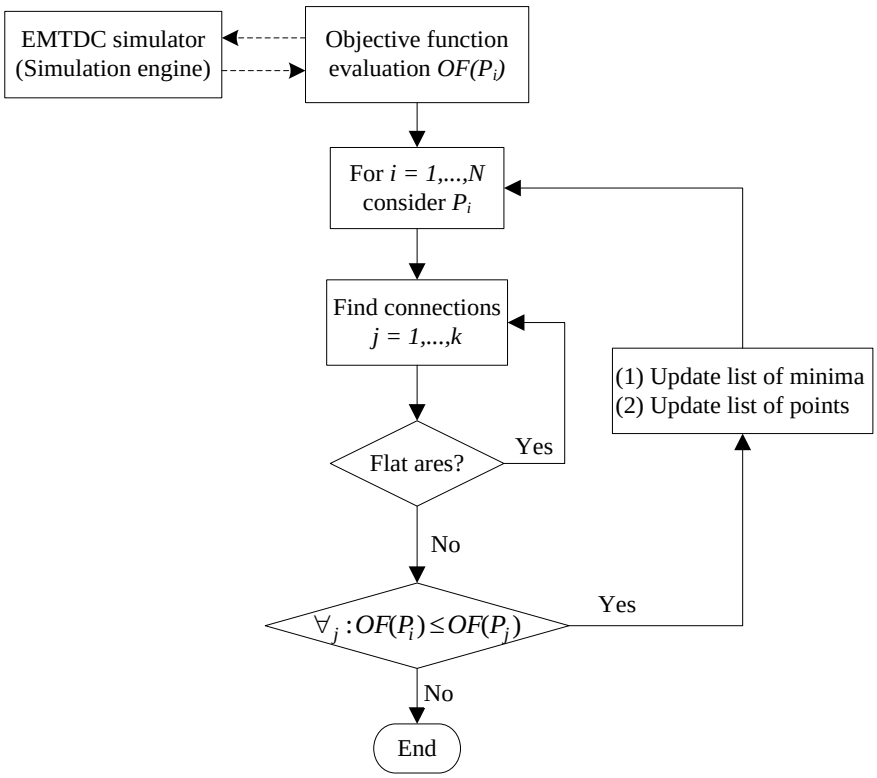


Fig. 3.14. Flow diagram for the localization procedure

3.3 Mesh Refinement and Surrogate Modelling

3.3.1 Purpose of mesh refinement

Mesh refinement is the next step of the proposed optimization algorithm following localization of the minima. The aim of mesh refinement is to increase the accuracy of the estimation of local minima. It consists of addition of a number of suitably selected points to the localized areas of the mesh, followed by triangulation to establish connections (links) between the newly-added and existing points.

Similar to the points in the initial mesh, calculation of the OF for the new points requires a corresponding number of simulations of the black box system. Therefore a compromise should be made between the number of points to be added in each stage of mesh refinement and the resulting intensity of the simulations required for their OF evaluation.

Several strategies for adding points to improve the accuracy of the final results are suggested in the following section. Some of the upcoming methods had been used in the previous adaptive multi-modal optimization algorithm [66]; these methods specify the number of required points based on the dimension of a problem at hand and hence prohibitively increase the number of expensive objective function evaluations in the subsequent iterations of a high-dimension design case. The algorithm proposed here uses the concept of surrogate modelling (borrowed from the general class of response surface methods) to add only *one* strategically selected point to each localized area. As a result, the computational intensity of the subsequent simulations drops significantly.

3.3.2 Addition of new points and mesh refinement

Once the potential regions likely to have local minima have been detected, new sample points will be required to further improve the accuracy of estimation of local minima. Strategically selected points can accomplish the design optimization at reduced

computational cost. Several intuitive strategies have been described in the next sub-section; however, they are tediously slow and impose a large number of samples, especially when the number of design parameters increases. Alternatively, a thoughtful approach has been proposed in the subsequent sub-section. This approach uses the surrogate models to wisely select new trial points.

A. Intuitive approaches

In this section three different intuitive approaches for addition of new points around a local minimum are presented. These approaches are able to progressively locate minima.

i. Center of mass simplex-based approach [71]

In this approach, the center of mass concept is employed to add trial points around a local minimum. Only the simplices comprising a localized region are considered for this purpose; these simplices are available through the Delaunay algorithm. The geometric center, *centroid*, of each simplex is analytically obtained and corresponding points are considered for the next iteration. Fig. 3.15 depicts a two-dimensional representation of this approach, where the added points are denoted by squares.

The main drawback of this approach is the indeterminate and unfixed number of required sampling points; this number completely depends on the number of simplices surrounding a local minimum. More importantly the computation intensity of this approach in calculating all centroids retards the design procedure; the number of trial points also can rapidly increase in higher dimensions.

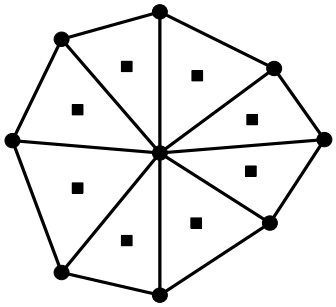


Fig. 3.15. Center of mass approach

ii. Hyper-cube based approach

A hypothetical hyper-cube is considered around a local minimum; the points located on its corners are treated as new trial points. As a result, 2^n points are added each time; this number drastically will increase as the number of variables increases. Fig. 3.16 portrays an example of a three-dimensional case, where L_{av} defines the distance of the new points (squares) from the discovered local minimum, denoted by the dot; this distance can be either (i) a fraction of the average distances of neighbouring points from the local minimum or (ii) a portion of the initial length of the links.

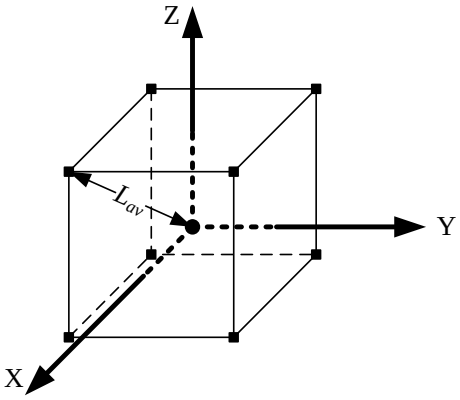


Fig. 3.16. Hyper-cube based approach

iii. 2×n approach [66]

As the name implies, this approach considers $2 \times n$ points, where n is the number of variables, around a local minimum; that is, two points on each axis. Fig. 3.17 shows a three-dimensional example of this approach where the added points denoted by squares have the fixed distance of L_m from the local minimum; this distance is equivalent to L_{av} in the hyper-cube case. This approach necessitates smaller number of points comparable to the previous approaches and does not have the computational concerns of the first one. However, it still requires large number of points especially when the number of variables grows.

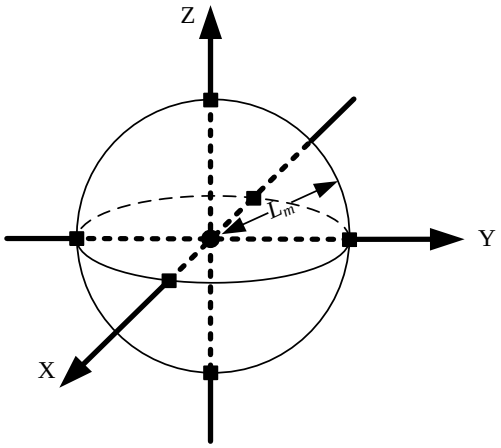


Fig. 3.17. 2×n approach

B. A surrogate-based approach

The approaches discussed in the previous sub-section have considerably slow convergence rate and require a relatively large number of sampling points. On the other hand, the unparalleled surrogate-based approach presented here is fast, accurate, and only adds one judiciously selected point for each local minimum.

i. Surrogate modelling and fitting

Surrogate models are simple, closed-form functions that are described by a relatively small number of parameters. For example, the following shows quadratic and Gaussian functions.

$$\text{Quadratic: } S(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{B}^\mathbf{T}\mathbf{x} + c \quad \text{(a)}$$

$$\text{Gaussian: } S(\mathbf{x}) = ke^{-\left(\frac{(x-x_1)^2}{2\sigma_1^2} + \frac{(x-x_2)^2}{2\sigma_2^2} + \cdots + \frac{(x-x_n)^2}{2\sigma_n^2}\right)} + c \quad \text{(b)} \tag{3.8}$$

In the optimization algorithm presented here surrogate models are developed and fitted to the localized areas of the objective function. Development of these surrogate models follows two distinct purposes as stated below:

1. Since a surrogate function describes the local behaviour of an OF, its minimization provides a good estimation of the actual location of the local minimum of the OF; and
2. A suitably fitted surrogate function provides a closed-form (and hence with minimal computational complexity) representation of the actual OF. It can therefore substitute actual, intensive system simulations in the subsequent sensitivity analysis of a design; thereby offering significant savings in computer resources.

In the proposed algorithm, surrogate models of a given form (e.g. quadratic) are created around each localized minimum using a user-defined number of points in its neighbourhood. The following expression is used for determination of the parameters of local surrogate models, i.e. fitting of the model.

$$\min \sum_{i=1}^{M} \| W(\mathbf{x}_i) OF(\mathbf{x}_i) - S^k(\mathbf{x}_i) \|^2 \tag{3.9}$$

where:

M: the number of points around a localized minimum selected for fitting;

$\mathbf{x}_i$: *N*-dimensional vector of optimization variables, i.e. a point in an *N*-variable optimization;

OF: the actual objective function (evaluated through simulation);

S^k: the local surrogate model around the *k*-th minimum;

and *W*: the weighting factor corresponding to each point.

Solution of this minimization problem yields the parameters of the surrogate model. Note that use of a weighting function (*W*) allows the user to judiciously assign weights to the points during the fitting process. For example, points with a lower OF evaluation are considered to be closer to the actual local minimum and hence may be assigned higher weights so that the surrogate model matches them more closely.

ii. Addition of new points and mesh refinement

Note that the surrogate model for each minimum is a close representation of the actual behaviour of the OF, in at least a sufficiently small neighbourhood around the minimum. Following fitting its parameters, the global minimum of each local surrogate model is determined and is then added to the existing mesh. Note that minimization of a surrogate model does not require simulations as the model has a known, simple and explicit analytical form. Moreover, due to the concave shape of the selected surrogate functions, they will only have one (global) minimum. Therefore, only *one* new point will be added to each localized area of the mesh.

Once added, the new points should be linked to the other points already present in the mesh. This is done using a triangulation algorithm (Delaunay triangulation in this case). It will be followed by force displacement to slightly adjust the position of the new points in accordance with the density function used. Two additional considerations are included in the mesh refinement stage:

1. Force displacement in the mesh refinement stage only moves the newly added points and will maintain the position of already existing ones. This will ensure that the expensive simulation-based OF evaluations already done for existing points do not need to be repeated, thus saving significant computational time;
2. The local expected length of the links for the points to be added needs to be adjusted to accommodate the fact that a higher density of points is being created by mesh refinement. This is done by modifying the density function so that it attains smaller values in the localized areas, as follows.

$$h_{\text{new}}(\mathbf{x}) = h_{\text{old}}(\mathbf{x}) \prod_{i=1}^{Nc} \left(1 - \alpha e^{-\left(\mathbf{x}-\mathbf{p}_{c_i}\right)^2 / 2\bar{l}_{p_{c_i}}^2}\right) \tag{3.10}$$

where $h_{\text{new}}(\mathbf{x})$ and $h_{\text{old}}(\mathbf{x})$ are the new and old density functions, respectively; N_c is the current total number of minima detected, $\mathbf{p}_{c_i}$ is the i-th local minimum and $\bar{l}_{p_{c_i}}$ is the average length of the bars connected to the i-th local minimum. The parameter α is used

to determine how much the expected length of bars is reduced in the refined mesh. In the implementation of the proposed algorithm in this thesis, α is set to 0.9.

3.3.3 Convergence check and subsequent iterations

The steps of objective function evaluation for the newly-added points, localization and surrogate function fitting, and mesh refinement are continued until a user-specified termination criterion is met. Examples of commonly used termination criteria are (i) achieving a desired value for the objective function, (ii) achieving a certain level of accuracy in the independent search variables, or simply (iii) exceeding a certain number of algorithm iterations.

Chapter 4

Properties of the Proposed Algorithm

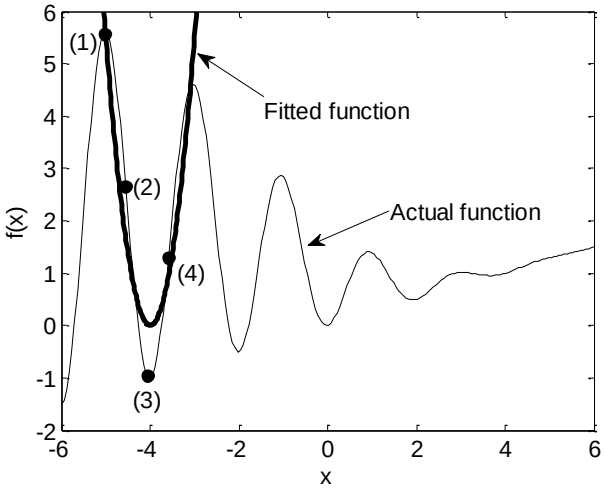
The proposed optimization algorithm provides several degrees of freedom to adapt to specific requirements of a problem at hand. These would assist the algorithm to find the local minima of a function easier, faster, with fewer number of intensive function evaluations. Besides, the surrogate models developed during the course of the search represent the function around its local minima and hence can be used for further assessments of the local minima after their discovery.

Some of the properties of the proposed optimization algorithm are highlighted in this chapter. Numerical examples, each highlighting a specific aspect of the algorithm, are presented in Chapter 5.

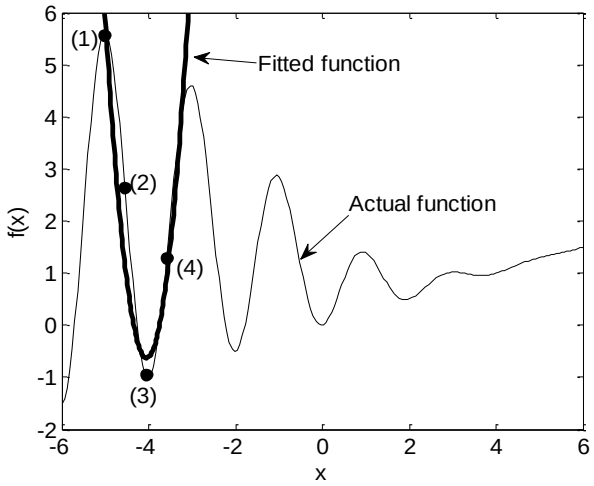
4.1 Weighting Factors

Weighting factors (in Eq. (3.9)) can control the accuracy of the fitted surrogate models to ensure that important features of the actual objective function around its local minima are captured, and represented in the fitted functions. Figs. 4.1(a) and 4.1(b) show two examples of quadratic surrogate functions fitted to a multi-modal objective function around its local minimum located at x = -4 using four sampling points (shown in circles and labelled).

The weighting factors in Fig. 4.1(a) are equal for the four points, whereas in Fig. 4.1(b) they are given different weights depending on how far their OF evaluations are from that of the current lowest point (point 3). As shown, use of unequal weights has resulted in a superior fit.



(a) Equal weightings

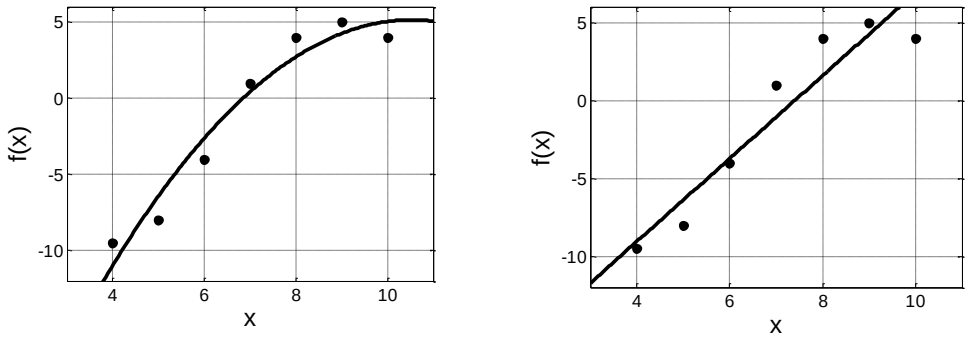


(b) Unequal weightings

Fig. 4.1. Effect of weighting factor

4.2 Additional Constraints on Surrogate Functions

Enforcement of other mandatory conditions also becomes possible through the use of suitable constraints in (3.9). For example in a minimization problem the local surrogate models around discovered local minima must be convex downwards [72]. For a quadratic function, for example, this implies that the matrix **A** in (3.8) must be positive definite, otherwise fitting using the same points may result in a surrogate function with a maximum. Fig. 4.2(a) and 4.2(b) show two surrogate functions fitted to the same set of points in a minimization problem, without and with the requirement of positive-definiteness, respectively. As seen the surrogate model fitted with no requirement for proper convexity (Fig. 4.2(a)) has a convex upwards shape, whereas enforcing the condition results in a convex downwards function as in Fig. 4.2(b).



(a) Without penalty parameter (b) Penalty parameter applied

Fig. 4.2. Effect of constraints in the fitting process

4.3 Parallelism

In the implementation stage, access to parallel processing platforms can significantly expedite the design process using the proposed optimization algorithm. Since evaluation of the OF for the trial points in the initial mesh is essentially a parallel task, it can be

assigned to a number of parallel processors. In the subsequent stages of the algorithm where the mesh is refined around the localized areas, parallel processors can be assigned to each localized area to perform the required OF evaluations.

The parallel processing is only available through simulation-based design. This feature can be used either in the objective function evaluation of the initial sampling points or in the subsequent iterations of the algorithm when localized areas and their corresponding points are determined.

4.4 Convergence Appraisal

The intuitive approaches for addition of points around a local minimum described in Chapter 3 can progressively locate minima provided that:

(i) Marginally close estimations for the locations of local minima have been obtained (output of *localization procedure*); the convergence may be disrupted or delayed, if the first or subsequent estimations of the potential locations of local minima are far from the actual minimum, whereas some closer points exist. This is due to the fact that closer points to a local minimum do not necessarily have smaller *OF* values in different directions; i.e. one point may be located at a relatively farther distance from a local minimum but has smaller *OF* value than a closer one; these two points obviously cannot be in a same direction toward the actual minimum.

(ii) A fixed distance value for the placement of new trial points obviously cannot be practical as a progressive search towards a minimum necessitates smaller distance values for succeeding iterations. Therefore, an adaptive distance value which can suitably change as the program proceeds has to be used. A suitably selected distance from a *discovered local minimum* for adding new points can re-direct the algorithm toward the *actual minimum* in cases where some inaccuracies are presented in the initial estimation of the location of the local minimum. A relatively short-distance from a local minimum of a well-behaved function can result in a rapid convergence. A long-distance point

placement strategy, on the other hand, increases the number of the subsequent iterations and consequently the number of *OF* evaluations to reach a specific accuracy, whereas it guarantees the convergence to some extent.

In general, the intuitive approaches can have the beneficial convergence property when a reasonable distance value for point placement is selected with the embedded cost of more function evaluations.

The proposed surrogate based approach, on the other hand, is exceptionally fast in locating the exact location of minima in most cases; but, it may fail to converge for functions with steep changes (rises and falls). Since the surrogate models initially developed for such functions may not accurately describe such ill-behaved functions, their resultant minima are not entirely useful as estimators of the locations of the actual minima. By checking the general trend of the optimization results, such conditions can be detected and proper treatment be applied. One possible solution, which has been implemented in the proposed algorithm, is to take the advantage of intuitive approaches (convergence property) in such situations. As a result, an intuitive approach, $2 \times n$ approach in this case, is in control for only addition of points and the rest of the procedure is kept intact. It has been tested that, modified surrogate models after this step can be successful in providing accurate representations of such functions and their minimization provides good estimations of the location of minima in subsequent iterations.

4.5 Sampling Strategies

One important question at the first step of the proposed optimization algorithm (mesh generation) is to define the exact number of sampling points based on the desired average length of the bars. This together with a proposed strategy to further improve the procedure of adding new points where an ill-behaved function is considered (akin to cases described in the previous section) are addressed in the following subsections.

4.5.1 Defining the number of initial points

The mesh generation algorithm proposed by Persson [73] calculates the initial number of points using the following equation.

$$\prod_{k=1}^{N}\left(\left\lfloor\frac{X_{max}^{k}-X_{min}^{k}}{h_0}\right\rfloor+1\right) \tag{4.1}$$

where h_0 is the desired average length of the bars; *N* is the number of variables; and $X_{min/max}^{k}$ is the lower/upper band of each variable.

Eq. (4.1) determines the total number of points based on the boundary intervals considered for each variables and the average desired length of the bars. However, due to the presence of the *floor function* in the above equation, the number of initial points is a discrete function of bars' length; i.e. for two different average length of bars (e.g. h_0 and h_0') a same resultant number of initial points may be obtained. The numbers tabulated here are calculated for a six-variable case in which each variable supposed to cover the entire interval of zero to 10 considered as the design space. As an example, any average length of bars within (2 – 2.5] results in the same number of initial points of 15,625 and then the number of points suddenly jumps to either 46,656 or 4,096 for smaller or larger average bars' length respectively. This can be further aggravated where more variables are involved.

Table 4.1. The number of initial points obtained applying Eq. (4.1)

L_0	*Number of points*
$(1.66-2]$	46,656
$(2-2.5]$	15,625
$(2.5-3.34)$	4,096
$(3.34-5]$	729

To rectify the above problem, the following formula has been proposed and implemented in the program. Even though it still embeds the discreteness due to the use

of the *ceiling function*; but, as it will be shown later, the discontinuity for the number of points is insignificant compared to the previous one. Since real numbers are not practical as for the number of points, the ceiling function is applied at the closing stage of Eq. (4.2). Also note that the *maximum function* ensures the selection of at least two points on each axis.

$$\left\lceil \prod_{k=1}^{N} \left(\max \left(\frac{X_{max}^{k} - X_{min}^{k}}{h_0} + 1 \, , 2 \right) \right) \right\rceil \tag{4.2}$$

As can be seen in Table 4.2, the number of points is progressively changing as the initial length of the bars L_0 is changing within the same interval of $(2 - 2.5]$ where the previous method only suggests a fixed number of 15,625 points for the entire interval. Note that the numbers presented in Table 4.2 are not discontinuous as a result of the proposed equation; on the contrary, the discontinuity is because of the step changes in L_0. Notice that the initial length of bars has been defined as intervals in Table 4.1, whereas a single real number is used in the following table. As an example for the discontinuity of the proposed formula, any initial length of the bars selected within the interval of $[2 - 2.000008)$ results in the same number of 46,656 points and any lengths equal to or larger than 2.000009 will result in a smaller number of points; on the other hand, any lengths smaller than 2 will generate a different larger number of initial samplings.

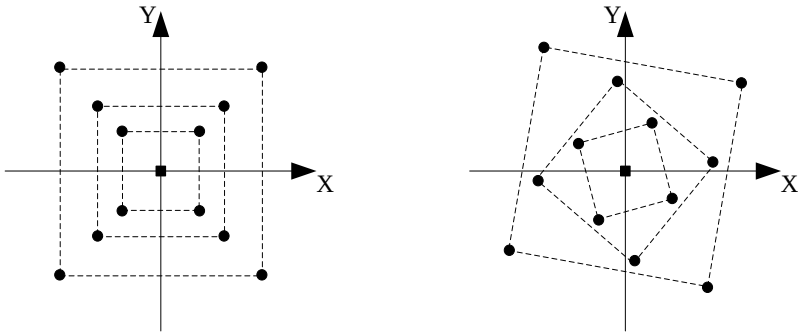
Table 4.2. The number of initial points obtained applying Eq. (4.2)

L_0	*Number of points*
2	46,656
2.1	36,593
2.2	29,082
2.3	23,392
2.4	19,023
2.5	15,625

4.5.2 Rotation procedure

To improve the convergence property of the proposed multi-modal optimization algorithm where ill-behaved functions are tackled, the added points by an intuitive based approach (refer to Sections 3.3.2 and 4.4) can be further modified to discursively explore the design space. The rotation procedure proposed here can result in new search directions; this increases the chance of finding a smaller *OF* value in subsequent iterations and the results are likely to be more reliable.

Fig. 4.3 shows two examples of adding points in which the circles are the points added by the hyper-cube based approach around the local minima denoted by squares. Fig. 4.3(b) has a higher chance of capturing a local minimum as it comprises more search directions compared to Fig. 4.3(a) where only a fix set of search directions are considered.



a. Conventional hyper-cube approach b. Rotated hyper-cube approach

Fig. 4.3. The effect of rotation procedure

The transformation matrix used in the rotation procedure is shown in the following equation:

$$\hat{\mathbf{p}} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} . \left(\mathbf{p} - \mathbf{p}_{c_i} \right) + \mathbf{p}_{c_i} \tag{4.3}$$

where φ is chosen randomly (and uniformly) within $[0, \frac{\pi}{2}]$; $\mathbf{p}$ represents the set of added points by a conventional intuitive approach and $\mathbf{p}_{c_i}$ represents the *i*-th local minimum; $\widehat{\mathbf{p}}$ corresponds to the resulting rotated points.

This rotation procedure becomes more complex in higher dimensions.

4.6 Surrogate Modeling

Two imperative questions that should be addressed prior to the model fitting are: (i) the number of necessary points to be considered for the least square problem; in other words, the required levels of accuracy for the surrogate models, and (ii) an effective approach in the development of surrogate models. These two are discussed in the following sub-sections.

4.6.1 Levels of accuracy for surrogate models

The least square technique is used to fit a desired surrogate model to a black box objective function. The least square error of fitting quadratic functions to the objective function results in a linear set of equations. However, for a general surrogate model, the solution of the resultant nonlinear equations, from least square errors, is only available through a nonlinear optimization algorithm; e.g. Gaussian functions defined previously as potential surrogate models necessitate a nonlinear optimization algorithm to determine parameters corresponding to a suitably fitted function. As a result, a least number of points is required to distinctively determine parameters of a surrogate model; i.e. the number of equations and the number of unknowns should be at least equal; or an over-determined system can be used for the course of optimization in the fitting process.

The following equations present the minimum number of points required for the suggested surrogate models in Eq. (3.8).

$$\text{Quadratic: } \frac{(n+1)\times(n+2)}{2} \quad \text{(a)}$$
$$\text{Gaussian: } 2 \times (n+1) \quad \text{(b)} \qquad (4.4)$$

where n is the number of optimization parameters.

Although any larger number can also be used, an excessively large number may result in surrogate models that properly represent the overall shape of the objective function but they are not locally accurate enough to be successful in the subsequent stage of adding points. On the other hand, a surrogate model that only uses the minimum number of points has a limited trust region. Hence a compromise should be made to define an appropriate number of points, or levels of accuracy, for the resultant surrogate models of the fitting process.

4.6.2 Adaptive surrogate modeling

A direct least-square matrix solution is available to fit the quadratic surrogate models to the sampling data points [74]. Nevertheless, in order to incorporate the weighting factors and the constraints in the resultant surrogate models as suggested earlier in this chapter, engagement of a nonlinear optimization technique, Nelder and Mead's nonlinear simplex algorithm in this thesis, is inevitable.

As a result of taking advantage of a nonlinear optimization technique, an iterative approach can be used. This would even further expedite the procedure as the previously fitted functions (surrogate models) can be used in the current iteration.

Fig. 4.4 shows a single variable multi-modal function together with its first three surrogate models for each local minimum developed by applying the above concept of adaptive surrogate modeling.

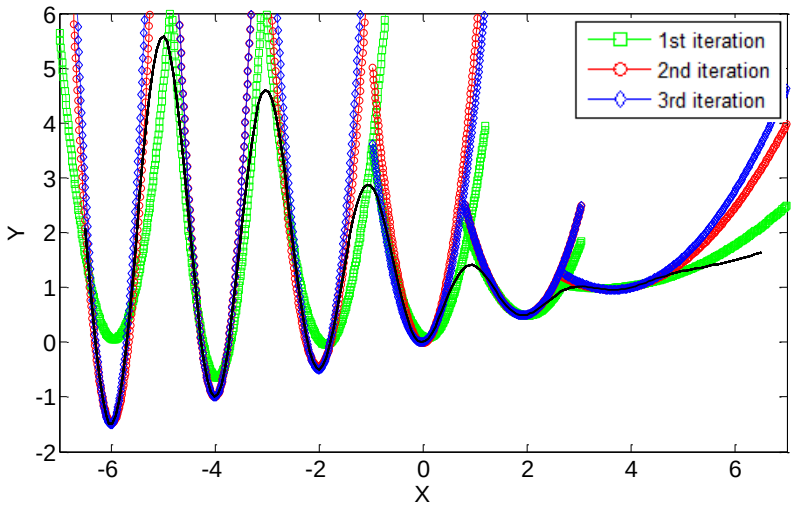


Fig. 4.4. Adaptive surrogate modeling

As seen in the above figure, the third fitted functions closely follow the variations of the objective function, especially in regions sufficiently close to local minima.

4.7 Global Optimization

Although the main objective of the proposed algorithm is to develop a platform for the optimization of nonlinear multi-modal functions, it can also be used as a global optimizer. In fact, if only the global optimum of a function is needed, the proposed optimization algorithm is capable of locating it and developing surrogate models within a few iterations.

The effectiveness of the algorithm in discovering a global minimum would be even more intensified if the global minimum is resting in an overall convex region. Fig. 4.5 shows an example of a single variable function with several local minima and one global minimum; the function has an overall convex shape. The dashed curve is the first order quadratic function developed as a surrogate model whereas the actual function is represented by a solid line.

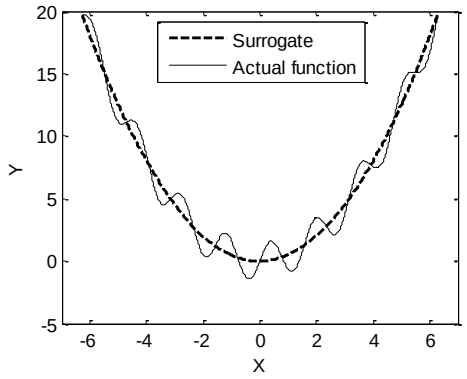


Fig. 4.5. Exploring a multi-modal function for its global minimum

Some work has been done aiming to use the response surface method as a global optimizer [7]. The proposed algorithm in this thesis, as highlighted here, can be also favourable for such purposes.

Chapter 5

Numerical Results and Comparative Assessments

This chapter scrutinizes and justifies the effectiveness of the proposed optimization algorithm using objective functions with known explicit forms. Four different studies are done including the assessment of the computational intensity of the algorithm (in case 1), accuracy analysis of fitting process and surrogate models (in case 2), effects of force displacement (in case 3), and optimization of benchmark multi-modal functions (in case 4). Since the multi-modal functions used in these cases have explicit forms, it becomes possible to verify how successful the algorithm has been in finding the expected local minima with adequate accuracy.

5.1 Case 1: Computational Intensity Analysis

The purpose of this case is to show the effectiveness of the proposed algorithm particularly with regard to the number of function evaluations required in the course of optimization of a multi-modal objective function, namely the Himmelblau's two-variable function. It also shows that with a suitably fine initial mesh, the proposed algorithm is able to detect all the local minima of a multi-modal function.

Himmelblau's two-variable function which is defined in Eq. (5.1) is widely used as a multi-modal test function for optimization algorithms. It has four analytically available local minima shown in the Table 5.1.

$$f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \tag{5.1}$$

Table 5.1. Local minima of Himmelblau's two-variable function

$f(x,y) =$ 0 for all four local minima			
(-3.78,-3.28)	(+3.58,-1.85)	(+3.00,+2.00)	(-2.80,+3.13)

Two variants of the proposed algorithm, one with quadratic and one with Gaussian surrogate functions, are used along with the previously introduced adaptive multi-modal optimization algorithm [66].

Each of the solvers is used with three different initial meshes, with average link lengths of 1.0 (fine mesh), 2.0 and 3.0 (sparse mesh). This is done to investigate the impact of the initial mesh sparsity on the subsequent discovery of local minima. For all three meshes, the initial points are randomly distributed in the [-6.0, +6.0] interval for both *x* and *y* variables (uniform distribution). Each algorithm is executed 100 times (i.e. with 100 such randomly initial meshes) to obtain statistically reliable information. Table 5.2 shows the percentage of the experiments that led to the discovery of all local minima for the three algorithms and for the three initial meshes.

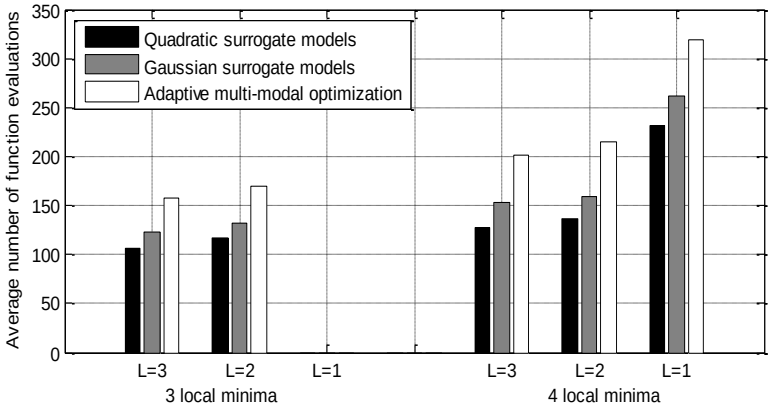
Table 5.2. Percentage of success in finding all four local minima

Algorithm	*Average initial link length*		
	L=1 (fine mesh)	*L=2*	*L=3 (sparse mesh)*
Proposed algorithm with quadratic surrogate functions	100%	80%	43%
Proposed algorithm with Gaussian surrogate functions	100%	76%	38%
Adaptive multi-modal optimization algorithm	100%	77%	53%

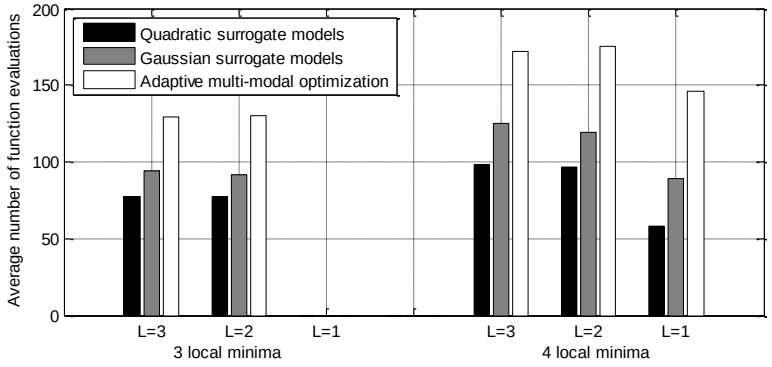
As seen all three algorithms find all four local minima when used with a fine initial mesh of L = 1. The likelihood of capturing all local minima drops almost uniformly when

more sparse meshes are used, with the Gaussian-based algorithm having a slightly lower chance of detection. This rate of success, however, is not obtained at the same computational expense for all three solvers. Fig. 5.1(a) shows the average number of function evaluations used by the algorithms for the above meshes. For example, discovery of all four local minima with a quadratic version of the proposed algorithm requires well below an average of 250 simulations, whereas it requires much larger than 300 simulations (on the average) using the adaptive multi-modal optimization algorithm.

The figure shows average values for both the cases where 4 and 3 local minima are detected. The proposed algorithm (its two variations) and the adaptive multi-modal optimization algorithm use the same meshes in the initial stage of their search; however the algorithm developed here requires a much smaller number of additional points in its subsequent iterations resulting in appreciable savings in the number of function evaluations, this is clearly shown in Fig. 5.1(b).



(a) Including the initial mesh points



(b) Without the initial mesh points

Fig. 5.1. Average number of function evaluations used by each algorithm

5.2 Case 2: Fitting and Surrogate Models

Surrogate functions are used in the proposed algorithm to aid in the identification of suitable experiments (points) in the vicinity of discovered local minima. These surrogate models are progressively tuned and fitted to the surface of the actual objective function and thus are reliable local indicators of its behaviour. In the present case, a single-variable multi-modal objective function is used to demonstrate the quality of the surrogate functions in describing the local variations of the objective function.

Consider the following objective function on the interval [-6.5, 6.5].

$$f(x) = \frac{x}{4} + (1 - \cos(\pi x)) \left(\tanh\left(\frac{x}{4}\right) - 1 \right)^2 \qquad (5.2)$$

This function has six local minima in the stated interval, and they are available analytically as shown in Table 5.3. The table also shows the local minima discovered by the proposed optimization algorithm with both quadratic and Gaussian surrogate functions using adequately fine meshes. The results show great agreement between numerically and analytically obtained minima.

Table 5.3. Local minima of the objective function in (5.2)

	Min #1	*Min #2*	*Min #3*	*Min #4*	*Min #5*	*Min #6*
Actual	-6.007	-4.0082	-2.0118	-0.0249	1.9189	3.6528
Quadratic	-6.007	-4.0081	-2.0126	-0.0248	1.9190	3.6527
Gaussian	-6.007	-4.0081	-2.0118	-0.0247	1.9191	3.6527

Table 5.4 and Table 5.5 show the actual values of the first and second derivatives of the objective function, respectively. The values are obtained by direct differentiation of (5.2) at its six local minima. Given that the quadratic and Gaussian surrogate functions formed around the local minima are expected to be faithful estimators of the objective function, their first and second order derivatives are also calculated and shown in the corresponding tables.

As seen the two surrogate functions do produce close approximations of the first and second order derivatives of the objective function. The second order derivatives are of particular importance as they are not necessarily vanishingly small around a minima (as are the first order derivatives), and they also show the general local shape of the objective function. Second order derivatives are also important in sensitivity assessment of optimized parameters and have been used in engineering situations [4].

The results in the following tables indicate that the Gaussian surrogate models are at times slightly less accurate than the quadratic ones, evidenced by their larger error in estimating second order derivatives. This is due to the shape of a Gaussian that tends to roll flat where sufficiently far from its minima. This feature may not be favourable with a

large class of objective functions. Use of Gaussian models is therefore recommended for cases when only sufficiently small deviations around the local minima are to be studied. Fig. 5.2 shows the actual objective function and the six quadratic surrogate models that are fitted to the function around its local minima.

Table 5.4. First order derivatives of the OF and surrogate functions

	Min #1	*Min #2*	*Min #3*	*Min #4*	*Min #5*	*Min #6*
Actual value	0	0	0	0	0	0
Quadratic	-0.0016	0.0016	-0.0183	0.0009	0	-0.0001
Gaussian	0	0.0009	-0.0011	0.0016	0.0004	-0.0001

Table 5.5. Second order derivative of the OF and surrogate functions

	Min #1	*Min #2*	*Min #3*	*Min #4*	*Min #5*	*Min #6*
Actual value	35.8495	30.7069	21.2852	10.2097	3.2867	0.7480
Quadratic	35.8092	30.7041	21.2877	10.1575	3.3117	0.7455
Gaussian	36.7413	31.5063	21.8733	10.7011	3.4226	0.7595

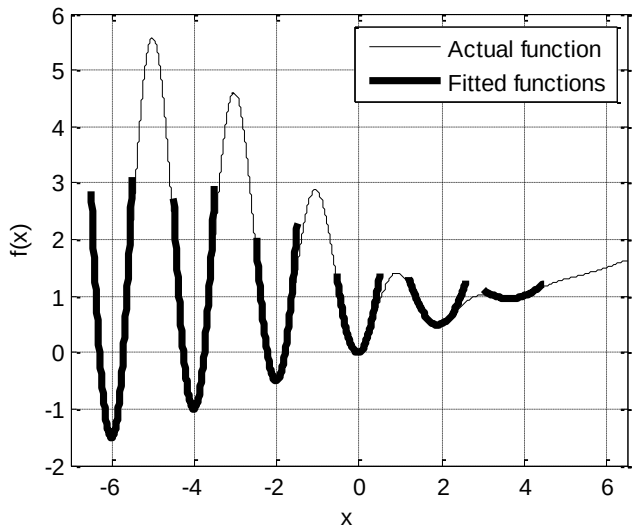


Fig. 5.2. The objective function in (5.2) together with its final quadratic local surrogates

5.3 Case 3: Effects of Force Displacement

In this section the effect of applying force displacement has been studied. An example of a two-dimensional case is used to demonstrate the impacts of force displacement on the number of function evaluations and the success of the algorithm in finding all local minima. Previously suggested algorithm [66] is also implemented in order to markedly compare the results.

5.3.1 A mathematical example

The following two-variable function with three local minima is considered to investigate the effect of force displacement on the computational burden of the proposed algorithm. As stated earlier, the recently proposed algorithm by Kobravi [66] is also tested.

$$\begin{aligned} f(x,y) = {} & 3(1-x)^2 e^{-(x^2+(y+1)^2)} \\ & - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{-(x^2+y^2)} \\ & - \frac{1}{3} e^{-(x+1)^2+y^2} \end{aligned} \tag{5.3}$$

The objective is to minimize the function in the interval of [-3, 3] for both variables so that they are accurate within two decimal places; i.e. an error of 0.01. The average length of links is chosen to be 0.5 and an initial random distribution of points is considered.

In order to obtain a statistically reliable result, the optimization programs have been run 100 times for each algorithm. The following table shows the percentage of the runs that resulted in the discovery of all local minima.

Table 5.6. Percentage of success in finding all local minima

	Algorithm	*Force displacement*	
		Yes	No
1-	Proposed algorithm with quadratic surrogate functions	100%	97%
2-	Proposed algorithm with Gaussian surrogate functions	98%	94%
3-	Adaptive multi-modal optimization algorithm	99%	95%

As seen, all the algorithms show enhanced results when the force displacement technique is applied. Besides, the quadratic surrogate modeling has slightly better outcome, in both cases, than the other two algorithms.

Table 5.7 shows the average number of function evaluations together with their range of variations in 100 runs presented for each algorithm with and without the influence of force displacement. Similarly, the force displacement technique can have valuable impacts and significantly reduce the number of function evaluations.

Table 5.7. The effect of force displacement on the computational burden of algorithms

	Function evaluations			
Algorithm	*Range*	*Average*	*Range*	*Average*
	With force displacement		*Without force displacement*	
1	284 - 360	316	365 – 420	394.2
2	274 - 385	326.9	379 - 432	401.4
3	342 - 427	378.4	423 - 473	445.2

Note that the numbers presented as for algorithms are substituted from Table 5.6.

5.4 Case 4: Optimization of Benchmark Multi-Modal Functions

In this section, the proposed algorithm is used in the optimization of a number of well-known multi-modal objective functions with explicit analytical formulation. The functions selected differ in the number of variables to be optimized, number of local minima, and also the complexity of the objective function surface. The mathematical formulation of each function together with the optimization results are presented in the following subsections.

Since the success of the algorithm partly depends on the quality of the initial mesh, the optimization for each function is repeated 50 times with 50 different initial meshes (all randomly generated) to allow drawing statistically sound conclusions. More details on the optimization parameters and the results are presented in the subsequent sub-sections.

5.4.1 Branin function

The Branin function is a two-variable multi-modal function defined as (5.4). It has three local minima which are presented in the Table 5.8.

$$B(x_1, x_2) = \left(x_2 - \left(\frac{5}{4\pi^2}\right)x_1^2 + \left(\frac{5}{\pi}\right)x_1 - 6\right)^2 + 10\left(1 - \left(\frac{1}{8\pi}\right)\right)cos(x_1) + 10 \qquad (5.4)$$

Table 5.8. Local minima of the Branin function

$B^*(x, y) \cong$ 0.397887 for all local minima		
$(-\pi, 12.275)$	$(\pi, 2.275)$	$(3\pi, 2.475)$

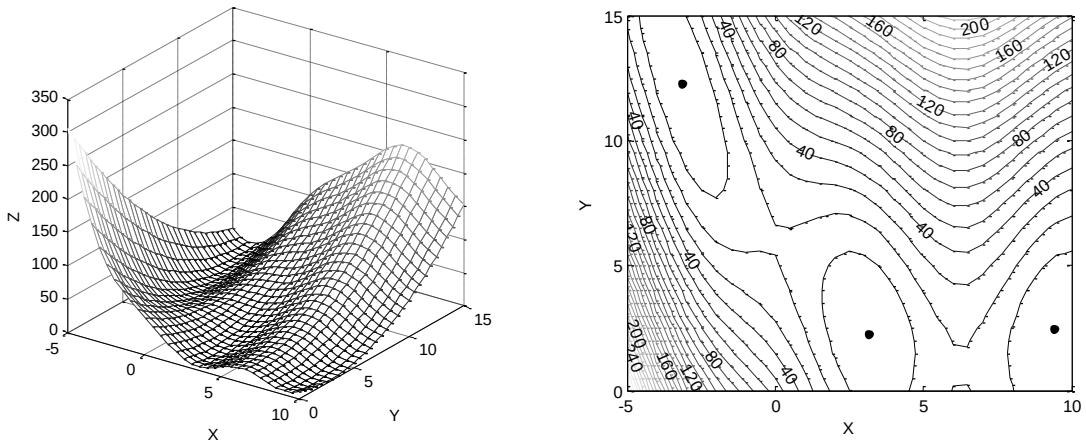
The first variable (x_1) has been changed within [-5, 10]; whereas, the second one (x_2) varied in the range of [0, 15]. Two different initial lengths of the links are considered and the summary of the result for 50 optimization runs are presented in the following table.

Table 5.9. Summary of the optimization results for Branin function

Success rate	*Average link length*	*Function evaluations*		
		Average	*Min/Max*	*Standard deviation*
100%	3	100.8	83/119	6.7
80%	5	74.7	59/90	7.5

As seen, the algorithm successfully finds all local minima of the function with the average link length of three. Besides, the number of function evaluations in both cases is reasonably small, which emphasises the effectiveness of algorithm

Fig. 5.3 shows a contour plot and a 3D view of the Branin function. Three local minima of this function are also highlighted in the contour plot using circles.



(a) Mesh plot (b) Contour plot

Fig. 5.3. The graphical view of the Branin function

5.4.2 Six-hump camel back function

As the name implies, it is a function of two variables with six local minima, two of them are the global ones. The mathematical formula of the function is shown in Eq. (5.5).The global minima and the function's value at those points together with the search space for each variable are presented in Table 5.10.

$$SH(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1 x_2 + (-4 + 4x_2^2)x_2^2 \qquad (5.5)$$

Table 5.10. The global minima and the search space for the Six-hump function

$SH^*(x, y) \cong -1.0316$, Search space: $x_1 \epsilon (-3,3), x_2 \epsilon (-2,2)$	
$(-0.0898, 0.7126)$	$(0.0898, -0.7126)$

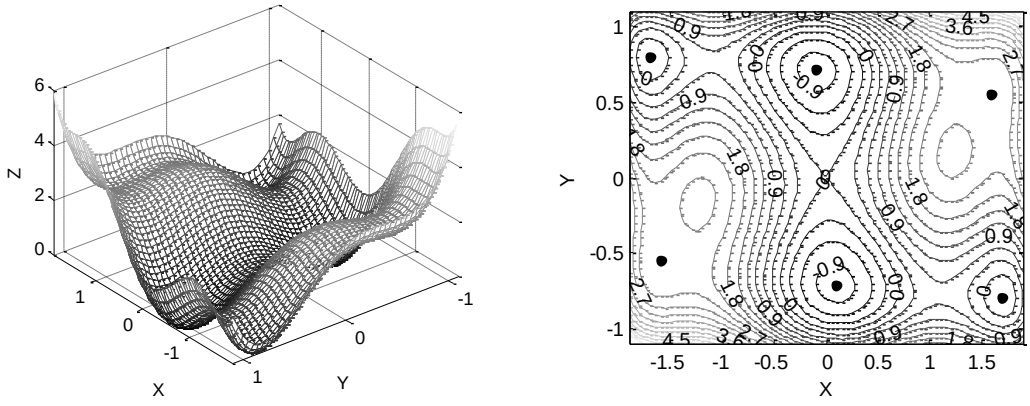
The following table shows the optimization results of the function. The algorithm satisfactorily locates all local minima. However, since a smaller links length is chosen, because of the complexity of the function, the number of function evaluations increases.

Table 5.11. Summary of the optimization results for Six-hump camel back function

Success rate	*Average link length*	*Function evaluations*		
		Average	*Min/Max*	*Standard deviation*
100%	0.2	794.6	775/811	9.2
78%	0.3	435.3	413/452	9.7

In order to see the variations of the function more clearly, the function is only presented in the interval of [-1.9 1.9] for x, and [-1.1 1.1] for y in Fig. 5.4; note that the simulation results are obtained using the search domain presented in Table 5.10.

The circles in Fig. 5.4(b) show the location of each minima.



(a) Mesh plot (b) Contour plot

Fig. 5.4. The graphical view of the Six-hump camel back function

5.4.3 Valleys function

This is also a two-variable function with three local minima. The function is defined as follows.

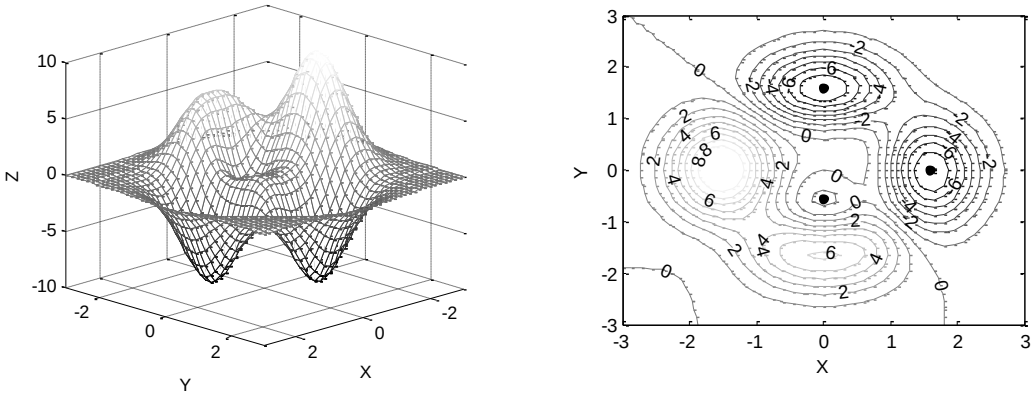
$$V(x,y) = -3(1-x^2)^2 \mathrm{e}^{-\left(x^2+(y+1)^2\right)} + 10\left(\frac{x}{5} - x^5 - y^5\right) e^{-\left(x^2+y^2\right)} + 3e^{-\left((x+1)^2+y^2\right)} \quad (5.6)$$

The function has three local minima within the range of [-3 3] for both variables. Table 5.12 summarizes the result of the optimization procedure with two different initial lengths of the links.

Table 5.12. Summary of the optimization results for Valleys function

Success rate	*Average link length*	*Function evaluations*		
		Average	*Min/Max*	*Standard deviation*
100%	0.5	258.9	246/267	5.4
80%	0.7	181.2	169/190	5.8

The following figure also illustrates the behaviour of the function with the local minima represented by circles in the corresponding contour plot.



(a) Mesh plot (b) Contour plot

Fig. 5.5. The graphical view of the Valleys function

5.4.4 Shekel function

Shekel functions are highly used as model functions for testing optimization techniques. Three different versions of the Shekel, each with different number of local minima, are available. In this section, a challenging form of the function with four variables and seven local minima is considered. Eq. (5.7) presents the formula of this function; the coefficients of the function ($\boldsymbol{a}_i$ and c_i) are shown in Table 5.13. The optimization uses an interval of [0 10] for all variables ($x_1, \dots, x_4$).

$$S_{4,7}(x_1, x_2, x_3, x_4) = -\sum_{i=1}^{7} [(\boldsymbol{x} - \boldsymbol{a}_i)^T (\boldsymbol{x} - \boldsymbol{a}_i) + c_i]^{-1}$$
$$\boldsymbol{x} = (x_1, x_2, x_3, x_4)^T; \boldsymbol{a}_i = (a_i^1, a_i^2, a_i^3, a_i^4)^T \tag{5.7}$$

Table 5.13. Coefficients of the Shekel function

i	$\boldsymbol{a}_i^T$				c_i
1	4.0	4.0	4.0	4.0	0.1
2	1.0	1.0	1.0	1.0	0.2
3	8.0	8.0	8.0	8.0	0.2
4	6.0	6.0	6.0	6.0	0.4
5	3.0	7.0	3.0	7.0	0.4
6	2.0	9.0	2.0	9.0	0.6
7	5.0	5.0	3.0	3.0	0.3

The location of the i-th local minimum approximately is equal to $\boldsymbol{a}_i^T$; whereas the $S(\boldsymbol{a}_i^T) \cong -1/c_i$. The global minimum located at $\boldsymbol{x}^* \cong (4,4,4,4)$ with the function value of $S^* \cong -10.4029$.

The following table shows the result of the proposed optimization algorithm using the Shekel function.

Table 5.14. Summary of the optimization results for Shekel function

Success rate	*Average link length*	*Function evaluations*		
		Average	*Min/Max*	*Standard deviation*
100% (5,6,7)	1.5	3632.2	3589/3707	24.9
80% (5,6,7)	2	1450.7	1427/1493	15.9

As seen, with a sufficiently fine initial mesh, the algorithm has a high likelihood of capturing more than four of the local minima for the Shekel function with a reasonably small number of function evaluations.

Chapter 6

Application Examples in High-Power Electronic Design

In this chapter the surrogate function based multi-modal optimization algorithm developed in the thesis is applied to several design examples involving power-electronic circuits. In the first example, a simple dc motor drive is considered. It comprises an uncontrolled ac-dc diode bridge in parallel with a low-pass LC filter. The dc-dc chopper is used to regulate the voltage across the dc motor and therefore maintain the desired speed. The design example aims to suitably adjust the proportional gain and integral time constant of the PI controller of the dc-dc chopper together with the filter elements at the output of the ac-dc diode bridge (a capacitor and two inductors) that result in a close match between the actual and the reference speed.

The second example demonstrates a highly complex speed control of an induction motor. Vector control or field-oriented control strategy has been used to provide a robust control for the induction motor. The load torque and the reference speed of the motor are changed repeatedly necessitating a vigorous and flexible control setting of the design parameters to maintain the speed, while the motor current is limited to its nominal value. The employment of the hysteresis current control loop and its naturally nonlinear switching frequency variations necessitate an adequately small simulation time step; i.e. a computationally intensive simulation case. Four different partial objectives are tackled

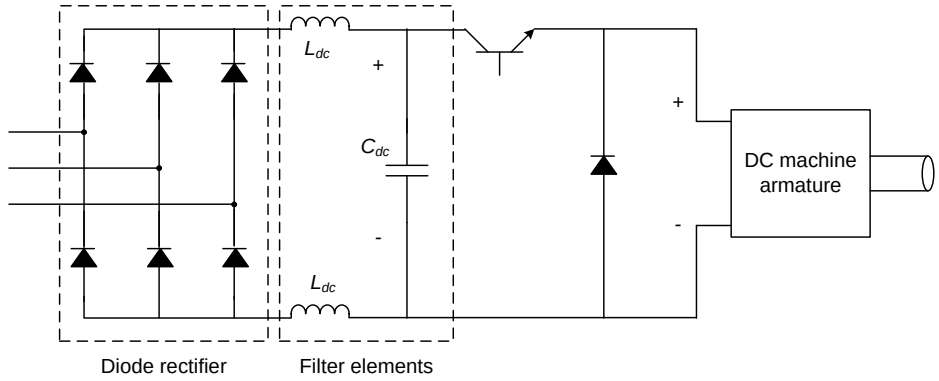
simultaneously for this case; namely, (i) the speed deviation; (ii) electrical torque ripples, (iii) harmonics contents of the input ac current; and (iv) potential cost of the designed elements. The aggregate objective function has been calculated by applying appropriate weights to each of the partial objectives.

In the last example, the Cigre HVDC benchmark system is studied. The proportional gains and time constants of the PI controllers on the rectifier and inverter sides are considered as the design parameters and the final results are assessed based on the value of the corresponding objective functions; the objective function monitors how closely the rectifier side current follows the current order.

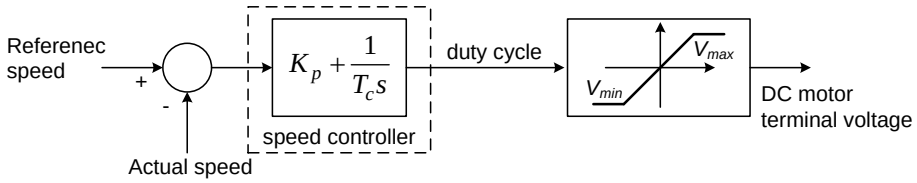
In all the above cases, due to the distinctive nonlinear and switching characteristic of the embedded power electronic devices, it is impossible to drive a detailed closed-form mathematical representation for the systems. They are therefore considered as black box systems and the simulation-based design approach has been used instead.

6.1 DC Motor Drive

The system under consideration is a switching dc motor drive, as shown in Fig. 6.1(a). The design problem is to determine optimal parameters for the speed controller (see the dashed box in Fig. 6.1(b)) as well as the inductor and capacitor values of the low-pass LC filter. Note that an explicit objective function for this circuit is exorbitantly difficult to formulate due to the nonlinear and switching nature of the underlying circuit. This necessitates use of a simulation-based design approach.



(a) Power circuitry



(b) Controller loop for the DC motor drive

Fig. 6.1. AC-DC power electronic dc motor drive

6.1.1 Description of the operation of the circuit

The drive system shown in Fig. 6.1(a) comprises an ac-dc diode bridge (rectifier), which serves to create an uncontrolled dc voltage from a three-phase ac system. The output of the rectifier is then fed to a low pass LC filter in order to remove its high-frequency ripple components and to obtain an essentially constant dc voltage across the terminals of the subsequent dc-dc converter. The dc-dc converter operates under a fixed frequency (selected to be 1800 Hz in this case), variable duty cycle regime [75]. Speed control is attained by suitably adjusting the duty cycle of the dc-dc converter, thereby creating different voltages at the motor terminals.

In order to automatically react to the changes in the reference speed of the motor or to compensate for its shaft load variations, a closed loop feedback system is employed,

where the difference between the reference and actual shaft speed is used to modulate the duty cycle of the dc-dc converter.

6.1.2 Design goals and development of an objective function

The intermediate LC low-pass filter and the duty cycle controller have profound impacts on the operation of the dc drive system both during transients and in steady state. Generally larger energy-storage elements (i.e. the inductor (L_{dc}) and the capacitor (C_{dc})) result in smoother rectified dc voltages and currents; however larger elements also tend to slow down the transient response of the closed loop controller. Although selection of suitable control system parameters can expedite the response time, it is often observed that a unified approach to the optimal selection of both the filter and controller parameters results in finer operation [4].

The objectives of the dc drive design problem are to select optimal values for the filter elements and the controller such that

1. the control system follows the *dynamic* changes in the reference speed as tightly as possible and with minimum deviation, and
2. the shaft speed in *steady state* is tightly regulated around its reference.

The following objective function is a simple representation of the above objectives as a mathematical expression.

$$OF\left(K_p, T_c, L_{dc}, C_{dc}\right) = \int_0^{t_f} \left(\omega_{ref} - \omega_{act}\right)^2 dt \qquad (6.1)$$

Note that the above OF penalizes any deviation between the reference and actual shaft speed both in transients and in steady state. More elaborate objective functions, which treat steady state and transient deviations differently (for example by giving unequal weights to these periods) can also be used [76]; however for this example, such more advanced objectives functions are not discussed.

Note that under ideal conditions where the actual speed of the machine exactly matches the reference speed both during transients and in steady state, the OF in (6.1) attains its absolute minimum of zero. For other conditions, minimization of (6.1) results

in the closest match. Further note that it is not easily possible to write an explicit formula for the objective function in (6.1) in terms of its four parameters. Its evaluation is rather done using computer simulation of the drive and its control system for various combinations of the four parameters as specified by the proposed optimization algorithm.

A computer model of the drive system is therefore created in the PSCAD®/EMTDC™ transient circuit simulator and is then linked with an implementation of the proposed optimization algorithm in MATLAB®. Table 6.1 shows the parameters of the drive system considered and simulated in the PSCAD®/EMTDC™.

Table 6.1. Drive system parameters

Moment of inertia of the rotor (J) = 0.002 [kg.m^2]			
Damping ratio of the mechanical system (B) = 0.001 [N.m.s]			
Rated armature voltage = 100 [V]		Rated armature current = 20 [A]	
Motor rated speed = 1800 [rpm]		Rated field current = 2 [A]	
$R_a = 0.5$ [Ω]	$L_a = 0.1$ [H]	$R_f = 25$ [Ω]	$L_f = 1.5$ [H]

In order to evaluate the dynamics of the drive system, a sequence of changes in the reference speed as shown in Fig. 6.2 is applied to its simulation model. The load torque for the period of simulation is maintained at 0.1 per-unit.

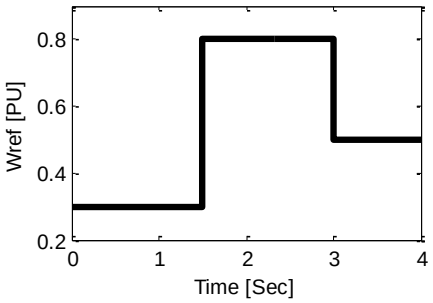


Fig. 6.2. Sequence of changes in the reference speed

6.1.3 Optimization results and discussion

The optimization algorithm is used for search within the boundaries shown in Table 6.2, and with a random distribution of seven initial samples for each variable (for a total of 2401 points in the initial mesh).

Table 6.2. Boundaries of each parameter

Parameter	K_p	T_c [s]	L_{dc} [H]	C_{dc} [mF]
Range	1000 - 10000	0.005 – 0.05	0.1 - 1	10 – 100

After 10 successive iterations of the algorithm (each iteration involves evaluation of the OF for the present points, localization, mesh refinement and surrogate model fitting), the algorithm detected a total number of 34 local minima. Of this number, 27 local minima reside either on or marginally away from the boundaries of the search area and seven are well within the boundaries. For the purpose of abridgement the following table lists only the local minima that are within the boundaries, along with their respective objective functions.

Table 6.3. Local minima and corresponding OF values

No.	K_p	T_c	L_{dc} [mH]	C_{dc} [mF]	*OF*
1	2122.715	0.012	393.8254	85.51425	0.013713
2	3487.423	0.043	753.4575	71.14979	0.014155
3	4017.187	0.012	853.4933	39.89997	0.015211
4	5018.060	0.019	314.1267	80.04724	0.012964
5	6999.240	0.027	546.0442	69.94061	0.013164
6	8502.411	0.035	550.8668	40.10422	0.012514
7	8532.914	0.020	850.5691	70.30985	0.012523

As seen the seven local minima have close objective function indicating that their overall performance in providing acceptable transient and steady state behaviour is close; however the corresponding parameter sets are different in some other aspects. For example, the local minimum 2 comprises a large inductor and relatively large capacitor,

which in practice imply higher cost and larger physical volume. From this perspective, local minima 1, 3 and 6 are preferred as they have smaller inductor values and much smaller capacitors. Comparative assessments such as this are only possible if the designer has access to all local optima of a design objective function, which is enabled through the proposed optimization algorithm.

Another important consideration in the design of complex systems is to assess the sensitivity of the optimized performance to the inevitable changes that may occur in the actual implementation of the system. The surrogate models formed and fitted around the local minima by the proposed algorithm are useful in providing sensitivity information without the need to re-simulate the system with perturbed parameters (such as in Monte-Carlo procedure). The following table shows the seven final surrogate models that are obtained by the algorithm around the local minima of Table 6.3.

Table 6.5 shows the second-order derivatives of the surrogate functions (obtained analytically) around the local minima. The first order derivatives of the objective function tend to zero for local minima not located on boundaries defined by the constraints; however the second order derivatives are not necessarily zero and show how sharply the objective function varies around the minima. As seen the local minimum 1 has by far the least sensitivity, and is therefore expected to result in a robust performance even in the face of changes in the optimized parameters.

Table 6.4. Final surrogate models in the form of $f(\mathbf{X}) = \frac{1}{2}\mathbf{X}^T\mathbf{A}\mathbf{X} + \mathbf{B}^T\mathbf{X} + c$

No.	**A**	**B**	*c*
1	$\begin{bmatrix} 1.1163 & -0.0221 & 0.0469 & -0.0608 \\ -0.0221 & 0.3870 & 0.0494 & 0.0340 \\ 0.0469 & 0.0494 & 0.2924 & -0.0226 \\ -0.0608 & 0.0340 & -0.02226 & 0.4461 \end{bmatrix}$	$\begin{bmatrix} -2.0781 \\ -1.4111 \\ -1.2007 \\ -3.7085 \end{bmatrix}$	22.4310
2	$\begin{bmatrix} 11.8647 & -0.0297 & 0.4066 & -0.765 \\ -0.0297 & 17.6914 & -0.3436 & -0.0921 \\ 0.4066 & -0.3436 & 14.2838 & 0.0608 \\ -0.765 & -0.0921 & 0.0608 & 16.0392 \end{bmatrix}$	$\begin{bmatrix} -38.97 \\ -153.2316 \\ -104.9916 \\ -111.3602 \end{bmatrix}$	1532.5
3	$\begin{bmatrix} 23.1034 & 0.0229 & -0.0514 & -0.0814 \\ 0.0229 & 28.0671 & -0.0787 & 0.0585 \\ -0.0514 & -0.0787 & 23.2971 & -0.036 \\ -0.0814 & 0.0585 & -0.036 & 17.4451 \end{bmatrix}$	$\begin{bmatrix} -92.9132 \\ -68.9371 \\ -198.7707 \\ -69.1133 \end{bmatrix}$	1259.7
4	$\begin{bmatrix} 14.9533 & 0.0697 & -0.3711 & -0.0826 \\ 0.0697 & 7.8154 & 0.6262 & -0.0407 \\ -0.3711 & 0.6262 & 11.4037 & -0.2666 \\ -0.0826 & -0.0407 & -0.2666 & 18.0923 \end{bmatrix}$	$\begin{bmatrix} -74.0429 \\ -31.6609 \\ -35.9781 \\ -143.6946 \end{bmatrix}$	881.3628
5	$\begin{bmatrix} 1.9949 & 1.1045 & 0.0149 & 0.0514 \\ 1.1045 & 2.661 & 0.2707 & 1.1249 \\ 0.0149 & 0.2707 & 2.4806 & -0.2653 \\ 0.0514 & 1.1249 & -0.2653 & 2.8198 \end{bmatrix}$	$\begin{bmatrix} -19.3859 \\ -30.8779 \\ -12.5435 \\ -24.9459 \end{bmatrix}$	265.872
6	$\begin{bmatrix} 52.2187 & -0.2801 & 0.0001 & -0.0895 \\ -0.2801 & 31.7515 & -0.0405 & 0.1674 \\ 0.0001 & -0.0405 & 42.2217 & -0.0509 \\ -0.0895 & 0.1674 & -0.0509 & 56.7355 \end{bmatrix}$	$\begin{bmatrix} -442.9843 \\ -222.2488 \\ -233.0925 \\ -227.5665 \end{bmatrix}$	3772.6
7	$\begin{bmatrix} 43.6031 & 0.0869 & -0.398 & -0.2023 \\ 0.0869 & 28.1231 & -0.018 & -0.3262 \\ -0.398 & -0.018 & 43.4643 & -0.0682 \\ -0.2023 & -0.3262 & -0.0682 & 44.6815 \end{bmatrix}$	$\begin{bmatrix} -369.154 \\ -109.8365 \\ -367.5411 \\ -310.267 \end{bmatrix}$	4458.1

Table 6.5. Second derivatives of the surrogate functions

No.	$\frac{\partial^2 OF}{\partial {K_p}^2}$	$\frac{\partial^2 OF}{\partial {T_c}^2}$	$\frac{\partial^2 OF}{\partial {L_{dc}}^2}$	$\frac{\partial^2 OF}{\partial {C_{dc}}^2}$
1	1.1163×10^{-6}	15480	29.24	4.461×10^{3}
2	11.8647×10^{-6}	707656	1428.38	160.392×10^{3}
3	23.1034×10^{-6}	1122684	2329.71	174.451×10^{3}
4	14.9533×10^{-6}	312616	1140.37	180.923×10^{3}
5	1.9949×10^{-6}	106440	248.06	28.198×10^{3}
6	52.2187×10^{-6}	1270060	4222.17	567.355×10^{3}
7	43.6031×10^{-6}	1124924	4346.43	446.815×10^{3}

By combining the properties of the local minima, from both parameter values and sensitivity standpoints, one can conclude that local minimum 1 is perhaps the most suitable option, providing a cost-effective filter (due to its small inductor and capacitor sizes) with least sensitivity. Fig. 6.3 shows the time response of the closed loop control system with the parameters of local minimum 1. As seen the reference speed is followed closely both in transient and steady state. The motor current waveform shows overshoots when accelerating the shaft, but does not exceed the rated 20A current of the motor.

The four-variable design example studied in this section initiated with 2,401 set of trial points and the final results obtained within the total number of 3,143 simulation runs. The conventional multiple-run approach, on the other hand, to acquire 1% accuracy for each variable requires $100^4 = 100{,}000{,}000$ number of simulation runs which is orders of magnitude larger than the proposed optimization algorithm.

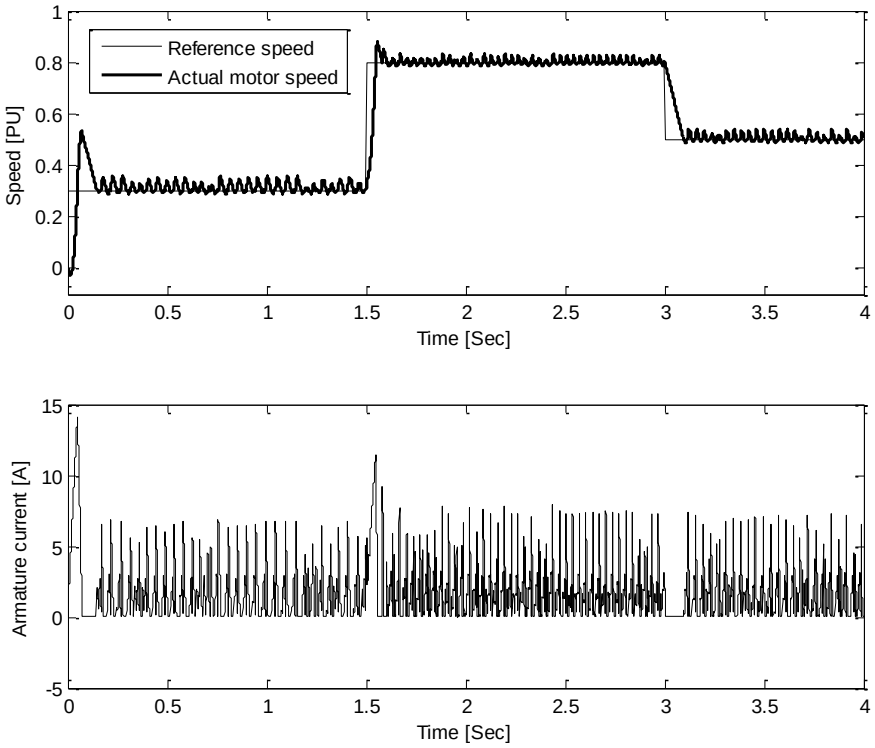


Fig. 6.3. Speed and current waveforms

6.2 Induction Machine Drive

Even with a relatively poor dynamic behaviour, induction motors have been extensively used in electric power networks. They are highly reliable due to the absence of commutators and brushes, mechanically simple, rugged, and economical [77]. However, with the considerable progress in the field of power electronics and available control strategies, it is now possible to use induction motors in high-performance drives, where only DC motors were traditionally used. With field-oriented or vector control methods, AC induction motors can be controlled in the same way as separately excited DC motors and hence easily control their dynamic behaviour. They bring about the added benefits of rugged construction and low maintenance requirement.

In the following sub-sections, first a mathematical review of the indirect vector control methodology is presented; it is then followed by a description of the circuit

structure and development of the objective function. Finally, the appraisal of the optimization results will conclude this section.

6.2.1 Mathematical review of the vector control methodology

The aim of vector control methodology is to implement control schemes that produce high-quality dynamic performance and are similar to those used to control DC motors. Unlike scalar controls, which control each variable individually without any concern about their coupling effects, field-oriented or vector control involves adjusting the magnitude and phase of the vector quantities of the induction motor. More importantly, scalar control methods use steady state circuit observations to develop a drive methodology, whereas vector control techniques are based on transient models.

In the vector control schemes, motor quantities (such as currents, voltages, magnetic flux, etc.) are described using complex space vectors. These vectors are valid not only for the steady state operation of the motor, but also during the transients. This will rectify the sluggish response of the scalar controls, such as *constant volt/Hertz,* during the transient period [78, 79].

Complex space vectors can be fully determined by only two components of the real and imaginary parts; the motor also can be considered as a 2-phase machine. This simplifies the equations and reduces the complexity of the control strategy. The following equation describes the instantaneous three phase stator current in a balanced mode:

$$i_{sa}(t) + i_{sb}(t) + i_{sc}(t) = 0 \tag{6.2}$$

The stator current space vector can be defined as in Eq. (6.3).

$$\overline{\boldsymbol{i}_s} = k(i_{sa} + a i_{sb} + a^2 i_{sc}) \tag{6.3}$$

where k is the transformation constant equal to 2/3 and $a = e^{\frac{j2\pi}{3}}, a^2 = e^{\frac{j4\pi}{3}}$.

Equivalently the current space vector can be presented by Eq. (6.4) in the two-axis stationary reference frame attached to the stator in which $i_{s\alpha}$ and $i_{s\beta}$ represents the real (direct) and imaginary (quadrature) components, respectively.

$$\overline{\boldsymbol{i}_s} = i_{s\alpha} + j i_{s\beta} \tag{6.4}$$

The fictitious 2-phase components of the stator current can then be easily calculated using the 3-phase stator current, as follows:

$$\begin{aligned} i_{s\alpha} &= k\left(i_{sa} - \frac{1}{2}i_{sb} - \frac{1}{2}i_{sc}\right) \\ i_{s\beta} &= \frac{k\sqrt{3}}{2}(i_{sb} - i_{sc}) \end{aligned} \tag{6.5}$$

Having a 3-phase balanced system as shown in Eq. (6.2), the above equation can be rewritten as:

$$\begin{aligned} i_{s\alpha} &= i_{sa} \\ i_{s\beta} &= \frac{1}{\sqrt{3}}(i_{sa} + 2i_{sb}) \end{aligned} \tag{6.6}$$

Eq. (6.6) represents the transformation of the three phase stator currents into a two phase orthogonal vector and is known as a *Clarke transform*. As a result, any motor quantity can be replaced by its space vector representation in the same way as described here. The motor quantities can also be formulated in a general reference frame rotating at a speed of $\omega_g = \frac{d\theta_g}{dt}$; where θ_g is the angle between the direct axis of the stationary reference frame (α) attached to the stator and the real axis (x) of the general reference frame. Fig. 6.4 illustrates the transformation between these two reference frames given by Eq.(6.7).

$$\overline{\boldsymbol{i}_{sg}} = \overline{\boldsymbol{i}_s} e^{-j\theta_g} = i_{sx} + j i_{sy} \tag{6.7}$$

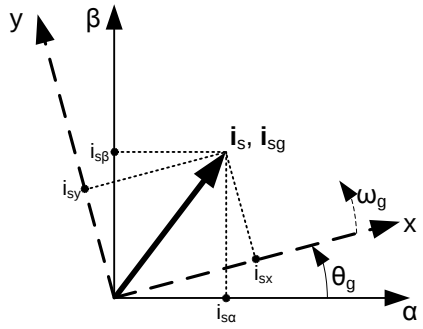


Fig. 6.4. Conversion to a general reference frame

Similarly all the other stator quantities can be obtained in the general reference frame. On the other hand, assuming the real axis of the reference frame to be attached to the rotor, all rotor quantities can be expressed by space vectors. Afterwards these quantities can be easily transferred to the general reference frame. The following equation describes such conversion for the space vector of the rotor current:

$$\overline{\boldsymbol{i}_{rg}} = \overline{\boldsymbol{i}_r} e^{-j(\theta_g - \theta_r)} = i_{rx} + j i_{ry} \tag{6.8}$$

where θ_r is the angle between the real axes of the reference frames attached to the stator and the rotor; and $\overline{\boldsymbol{i}_r}$ is the space vector of the rotor current using the rotor reference frame.

In vector control, all quantities must be expressed in the same reference frame. There are several choices for the reference frame such as stator flux-linkage space vector, rotor flux-linkage space vector or the magnetizing space vector. It is revealed that the reference frame (d-q) attached to the rotor flux-linkage space vector is the most suitable choice for the speed control of the induction motors. The direct axis (d) of this frame is aligned with the rotor flux space vector; i.e. the q axis component the rotor flux space vector is always zero.

After transformation into d-q coordinates the motor model would be as follows:

$$
\begin{aligned}
v_{sd} &= r_s i_{sd} + \frac{d}{dt}\lambda_{sd} - \omega_s \lambda_{sq} \\
v_{sq} &= r_s i_{sq} + \frac{d}{dt}\lambda_{sq} - \omega_s \lambda_{sd} \\
v_{rd} &= 0 = r_r i_{rd} + \frac{d}{dt}\lambda_{rd} - (\omega_s - \omega)\lambda_{rq} \\
v_{rq} &= 0 = r_r i_{rq} + \frac{d}{dt}\lambda_{rq} + (\omega_s - \omega)\lambda_{rd} \\
\lambda_{sd} &= (L_{ls} + L_m) i_{sd} + L_m i_{rd} \\
\lambda_{sq} &= (L_{ls} + L_m) i_{sq} + L_m i_{rq} \\
\lambda_{rd} &= (L_{lr} + L_m) i_{rd} + L_m i_{sd} \\
\lambda_{rq} &= (L_{lr} + L_m) i_{rq} + L_m i_{sq} \\
T_e &= \frac{3}{2}\frac{P}{2}\frac{L_m}{L_{lr} + L_m}(i_{sq}\lambda_{rd} - i_{sd}\lambda_{rq})
\end{aligned}
\tag{6.9}
$$

where:

$v_{sd,q}$	= d-q components of the stator voltages [V]
$v_{rd,q}$	= d-q components of the Rotor voltages [V]
$\lambda_{sd,q}$	= d-q components of the stator magnetic flux [Vs]
$\lambda_{rd,q}$	= d-q components of the rotor magnetic flux [Vs]
T_e	= electromagnetic torque [Nm]
$i_{sd}, i_{sq}, i_{rd}, i_{rq}$	= stator and rotor currents d-q components [A]
r_s, r_r	= stator and rotor phase resistances [Ω]
L_{ls}, L_{lr}	= stator and rotor phase inductances [H]
L_m	= mutual inductance between rotor and stator [H]

ω/ω_s = electrical rotor speed/synchronous speed [rad/s]

P = number of poles

Using the reference frame mentioned earlier, the q-axis component of the rotor flux vector becomes zero; that is, $\lambda_{rq} = 0$; the torque equation then would be as follows:

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_{lr} + L_m} i_{sq}\lambda_{rd} \tag{6.10}$$

The above equation resembles the behaviour of a DC motor where the q-axis component is used to control the torque while the d-axis component is adjusting the field. If the d-axis stator current is fixed at a value (corresponds to a desired field order), it can be shown that the d-axis rotor current will become zero. This will result in $\lambda_{rd} = L_m i_{sd}$ and consequently:

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m^2}{L_{lr} + L_m} i_{sq} i_{sd} \tag{6.11}$$

Referring back to the DC motor control, i_{sd} is analogous to the field current (the flux component) and i_{sq} is analogous to armature current (the torque component). Note that i_{sd} in Eq. (6.11) is aligned with the rotor's flux vector and i_{sq} is 90° in advance of i_{sd}. Applying the vector control concept, theses components of the stator current can be controlled independently resulting in a high-dynamic performance drive.

Fig. 6.5 summarizes the indirect vector control of an induction motor.

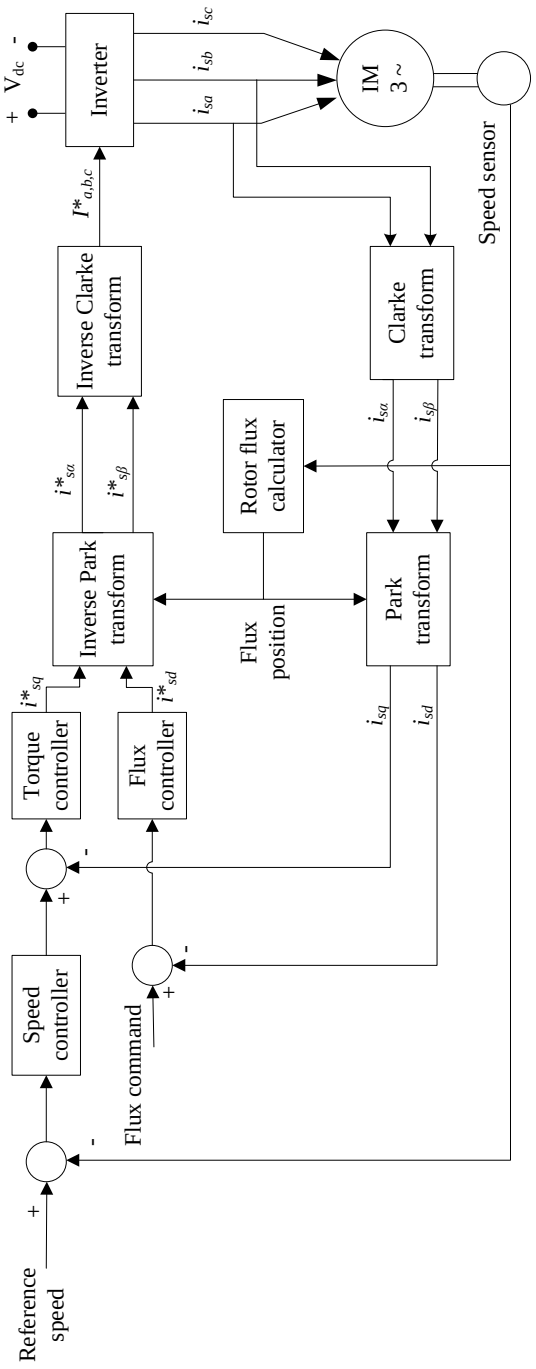


Fig. 6.5. Indirect vector control of an induction machine

6.2.2 Drive system and controller design

The system considered for this example consists of three major parts, as they are shown in Fig. 6.6. The six-pulse diode bridge is used to create an uncontrolled dc voltage from a three-phase ac system. Then a low-pass LC filter is employed to further alleviate the dc ripples and provide an essentially constant dc voltage for the following stage of dc-ac inversion. The higher the value of the L and C components, the better the output of the dc link in steady state; however, it may have undesirable effects on the transient period.

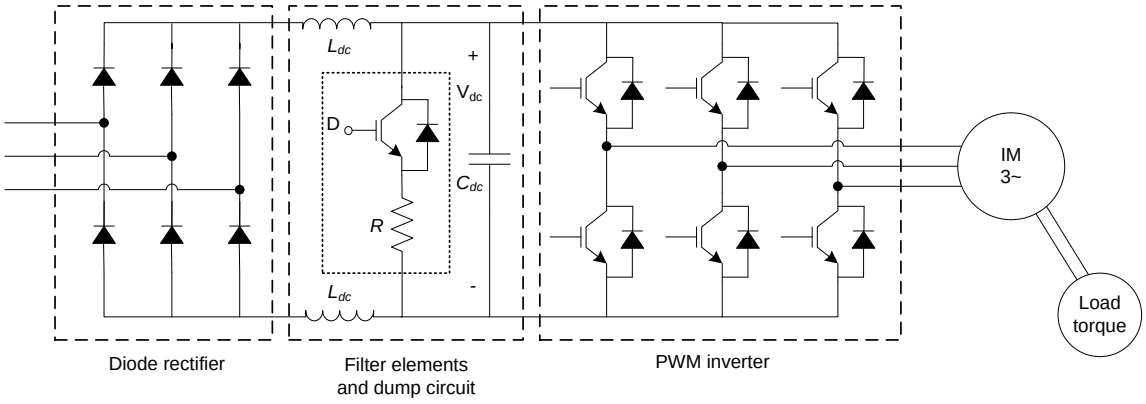


Fig. 6.6. Induction machine drive system

During breaking periods the energy stored in the motor (moving inertia) has to be transferred to a sink; however, the diode bridge used in this circuit is a one-directional gateway, which only conveys the energy from the ac source to the dc link. The only component in the circuit that is able to absorb the energy is the capacitor. As a result, the voltage across the capacitor will increase drastically to provide a rapid speed drop; this may have destructive effects on the capacitor. To avoid this, a bypass circuit is used in parallel with the capacitor to dissipate the energy only in conditions when the capacitor cannot withstand the high energy rush.

Fig. 6.7 shows the switching strategy used for this part of circuit. Once the voltage across the capacitor increases, the variable duty cycle of the series switch is accordingly increased to provide a path for the dump load resistor to dissipate the excessive energy and to compensate for the over-voltage.

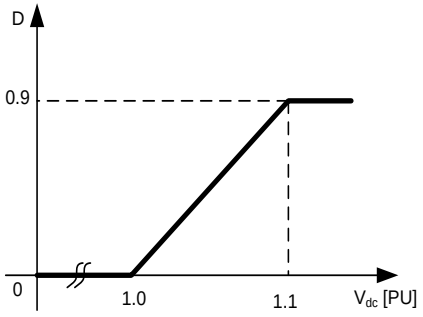


Fig. 6.7. Switching pattern of the dissipation circuit

Insulated-Gate Bipolar Transistors (IGBTs) are used as the last part of the circuit to produce approximately sinusoidal voltages and currents of controllable magnitude and frequency across the motor's terminal. The IGBT is usually used in high power applications due to its low drive current (controlled by MOSFET on its gate), high switching frequency capability and fast switching transient time. On the other hand, the voltage drop in IGBT is larger than the bipolar transistor, resulting in higher conduction losses.

The hysteresis current control loop manages the switching pattern for IGBTs; i.e. if the current drawn to a phase is smaller than the current order, the corresponding top switch is turned on, causing a positive voltage ($+V_{dc}/2$) to be placed across the terminals and raising the current; on the contrary, in case of a larger current than the required one, the bottom switch can apply a negative voltage ($-V_{dc}/2$) across the terminals and consequently reducing the current.

In essence, the switching frequency totally depends on the instantaneous value of the current on the sinusoidal reference waveform as well as the hysteresis band as shown in Fig. 6.8. A shorter hysteresis band requires more switching commands and consequently higher losses; however, the resultant waveform would have less low harmonic contents.

The control loop considered in this circuit is similar to Fig. 6.5. The flux command is set to its nominal value and speed control is attained by suitably adjusting the reference current at the motor terminals. In this example, the input current of the motor is tightly limited by its nominal value.

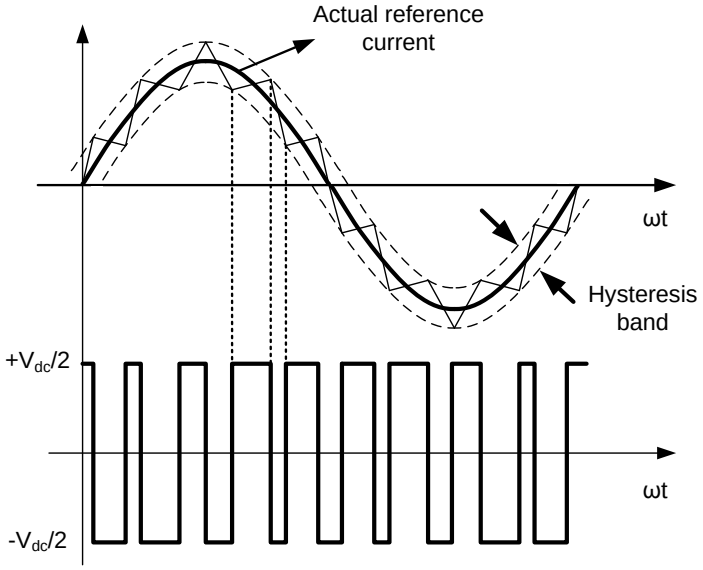


Fig. 6.8. Hysteresis current control

6.2.3 Design goals and development of an objective function

As stated earlier, the values of the L and C low-pass filter can have intensive impacts on the final result; as the large components in steady state provide superior dc voltage and current, they are inclined to slow down the transient. The values of L and C components together with the gain and time constant of the PI controller, which is used in the speed control loop, are deemed to be the input control parameters for the optimization of the system.

Several objectives are targeted for the above induction machine drive case as follows:

1. The control system tightly follows the reference speed in both *transient* and *steady state* conditions.
2. The electrical torque ripple is minimized in the steady state; this would result in a smooth stable rotation of the motor's shaft.
3. Minimum harmonic contents for the current drawn from the ac source.
4. Cost factor for the designed elements' size of the circuit (L and C).

To address all the above factors, the following aggregate objective function, OF_t(X), is considered. It permits the assignment of different weights (w_i) to each of the partial objectives (of_i) as shown in Eq. (6.12).

$$OF_t(K_p, T_c, L_{dc}, C_{dc}) = w_1 of_1 + w_2 of_2 + w_3 of_3 + w_4 of_4 \tag{6.12}$$

The partial objective functions are defined in Eq (6.13).

$$\begin{aligned} of_1 &= \int_0^{t_f} (\omega_{ref} - \omega_{act})^2 dt \\ of_2 &= \int_{ss} (T_{e-rip})^2 dt \\ of_3 &= \int_{ss} \left(\sum_{i=2}^{7} I_i^2 \right) dt \\ of_4 &= k_1 L_{dc}[mH] + k_2 C_{dc}[\mu F] \end{aligned} \tag{6.13}$$

As it is shown, the first partial objective function calculates the deviation of the actual speed from the reference speed during both transient and in steady state periods. The second one aims to represent a quantitative value for the electrical torque ripples during steady state, whereas the third one considers the second and high-order harmonic contents of the ac input current in steady state. The last objective function is intended to signify the size and cost of the designed circuit elements (L and C); where k_1 and k_2 are used to scale the values accordingly.

Table 6.6 shows the weighting and scaling factors used for each of the partial objective functions. These factors are obtained empirically by running several experiments and observing the range of variations for each of the of_i. It is tried that all partial of_i's proportionally contribute to the final aggregate objective function.

Table 6.6. Weighting factors of the partial objective functions

w_1	w_2	w_3	w_4	k_1	k_2
100	20000	70000	1	0.1	0.001

In order to evaluate the drive system and control parameters, a sequence of speed and load torque variations are applied to the motor. As it is shown in Fig. 6.9, the load torque has been increased linearly during one second and then it is limited to 0.1 [PU] for the next six seconds; this resembles a speed-dependent load torque during the start-up period followed by an active load. The total simulation time is 11 seconds during which dynamic changes applied for the reference speed and load torque as illustrated in Fig. 6.9.

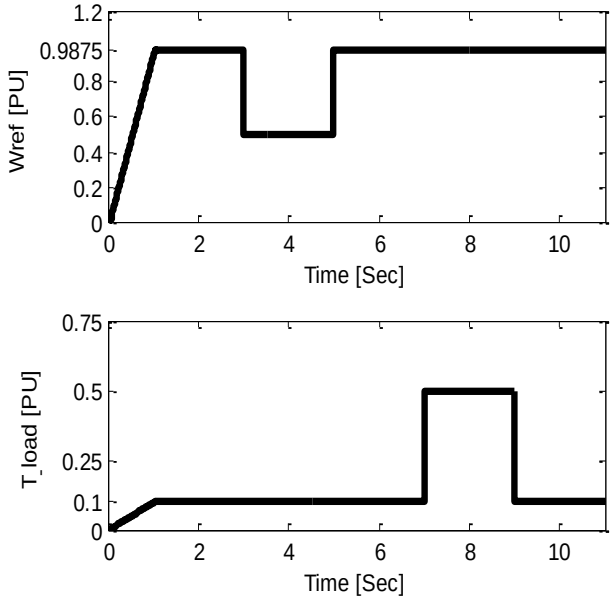


Fig. 6.9. Reference Speed and load torque variations

Note that, because of the employment of the hysteresis current control loop and its nonlinear switching characteristic, it is practically impossible to obtain an explicit closed-form mathematical function for the OF. The objective function evaluation is only possible either through the simulation or real world experiments. Therefore the drive system is implemented in PSCAD®/EMTDC™ and is linked to MATLAB® where the proposed optimization algorithm is hosted. Table 6.7 summarizes the induction machine parameters used in PSCAD®/EMTDC™.

Table 6.7. Induction Machine parameters

4 pole	f = 60 [Hz]	V_{LL} = 2300 [V]	P_{rated} = 500 [hp]	J = 11.06 [kg.m^2]
$r_s = 0.262$ [Ω]	$X_{ls} = 1.206$ [Ω]	$X_M = 54.02$ [Ω]	$X_{lr} = 1.206$ [Ω]	$r_r = 0.187$ [Ω]

Another prime consideration, due to the hysteresis current control, is the variable switching frequency of the system which in turn necessitates an adequately small simulation time step (20 [μs]); i.e. a computationally intensive design procedure.

6.2.4 Optimization results and discussions

The following table shows the range of variation considered for each design variable. Six points surrounded by the specified boundaries are uniformly distributed on each axis and the initial mesh has been constructed accordingly with the total number of 1296 points.

Table 6.8. The design parameters' boundaries

Parameter	K_p	T_c [s]	L_{dc} [mH]	C_{dc} [μF]
Range	5 -50	0.1 – 1	2 - 20	300 - 3000

The steps of mesh refinement, localization and surrogate modeling and optimization have been applied alternatively and the proposed algorithm has discovered 18 local minima after 10 successive iterations. The results tabulated here are five distant local minima from the boundaries accompanied by their corresponding partial and aggregate objective functions' values.

It should be noted that, our study shows most of the results are obtained within the first few iterations (three to five iterations in this case); this further proves the rapid convergence property of the proposed algorithm and suggests some modifications such as incorporation of the *OF* values into the termination criteria of the algorithm in order to expedite the design procedure.

Table 6.9. Local minima and corresponding partial and aggregate *OF* values

					Partial objectives				OF_t
	K_p	T_c	L_{dc} [mH]	C_{dc} [μF]	of_1	of_2	of_3	of_4	
1	22.93	0.46	12.7	1920.6	15.67	10.45	3.28	3.19	32.60
2	25.62	0.69	6.2	1519.0	15.30	10.59	4.02	2.14	32.06
3	31.67	0.27	12.8	1321.8	15.55	10.55	3.69	2.60	32.39
4	37.23	0.19	17.2	1707.2	15.66	10.70	3.76	3.43	33.55
5	41.06	0.46	9.2	1379.6	15.04	10.62	3.67	2.30	31.63

As it can be seen, the fifth solution is the global minimum and the local minimum number 4 has the largest aggregate *OF*. Also it is possible to compare the solutions based on their partial *of* values; e.g. the second local minimum has the smallest components' size (of_4 = 2.14) while it implies the largest harmonic contents of the input ac current (of_3 = 4.02); on the contrary, the first local minimum has the lowest input current harmonics (of_3 = 3.28) while its speed deviation has the largest value (of_1 = 15.67); it can be inferred that it may not be able to pursue the reference speed satisfactorily (this will be clarified later when the simulation results are presented).

A practically feasible solution should also have low sensitivity to the inevitable variations that may occur during the design procedure; this may happen after the implementation of the design due to the changes in the environmental conditions or components' structure such as temperature or aging; or even before that where the exact values of design parameters such as capacitor and inductor may not be physically available. The presence of surrogate models that accurately describe the system behaviour around its local minima can be useful in this stage and provide the necessary information without any prerequisite of additional simulations. Table 6.10 shows the second-order derivatives of the final surrogate models obtained for this case.

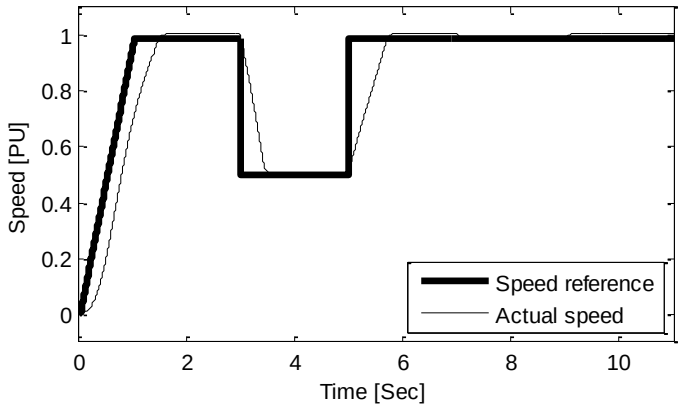
Table 6.10. Second-order derivatives of surrogate models

No.	$\frac{\partial^2 OF_t}{\partial {K_p}^2}$	$\frac{\partial^2 OF_t}{\partial {T_c}^2}$	$\frac{\partial^2 OF_t}{\partial {L_{dc}}^2}$	$\frac{\partial^2 OF_t}{\partial {C_{dc}}^2}$
1	0.33505	619.4378	2.817146	9.12×10^{-5}
2	0.051092	68.51258	0.493263	1.03×10^{-5}
3	0.067153	335.7926	0.547967	3.07×10^{-5}
4	0.230015	823.0202	1.320563	7.84×10^{-5}
5	0.153821	790.859	0.947558	3.76×10^{-5}

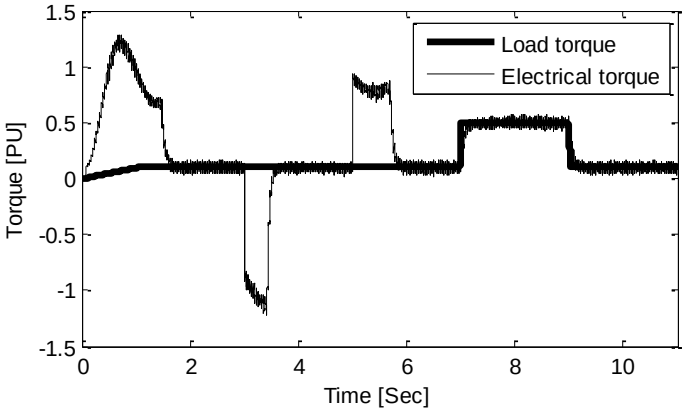
As it can be seen, the second local minimum has the lowest sensitivity to all the design parameters and hence is the best sensitivity wise option. In addition, the third local minimum presents reasonably small second derivatives, which imply an insensitive design; the fourth local minimum, on the contrary, is not a good option. A compromise should be made to choose the best suited solution considering all the above factors.

In this study, an overall acceptable performance of the machine is required; i.e. all the partial objectives are fairly satisfied and the aggregate objective function to be relatively small. It is also desired to have a solution that manifests the least sensitivity and claims low costs. As a result, the second local minimum has been chosen for this case considering all the above conditions.

The speed and torque transient responses of all local minima are depicted in the following figures. As expected, the actual speed of the motor has small deviations for the first, third and fourth local minima; this is consistent with the results obtained for the corresponding partial objective function (of_1) presented in Table 6.9.

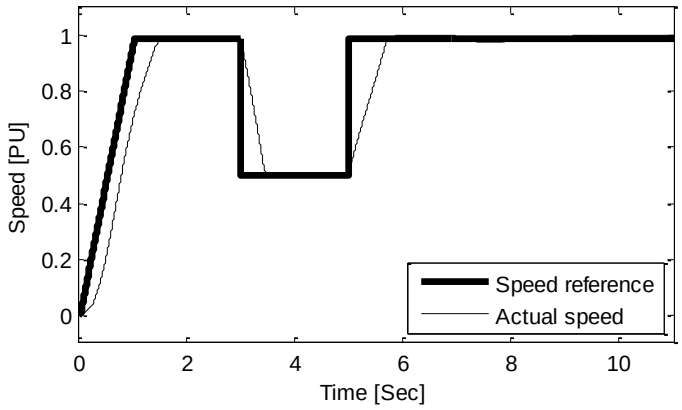


(a) Speed variations

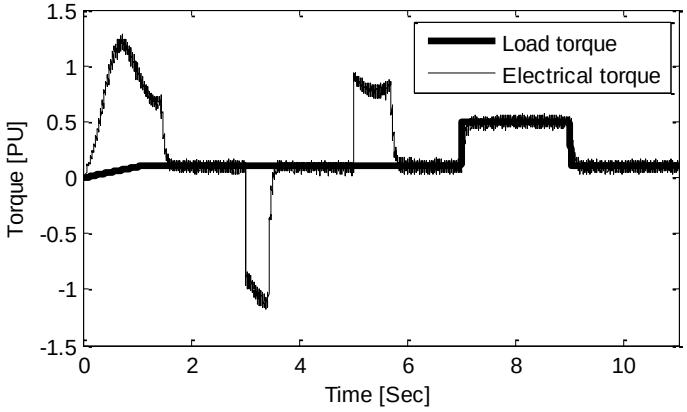


(b) Torque variations

Fig. 6.10. Transient response of the induction motor for the first local minimum

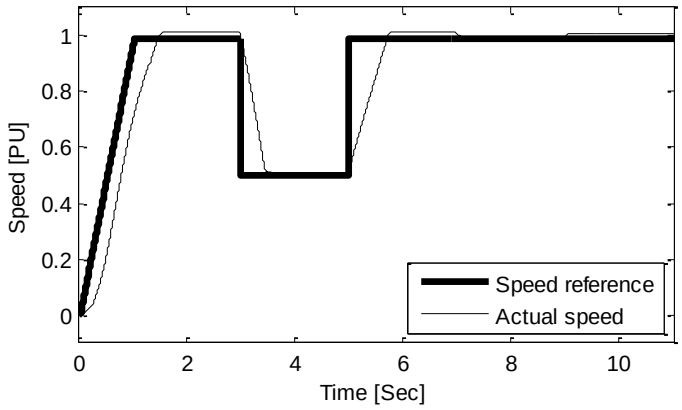


(a) Speed variations

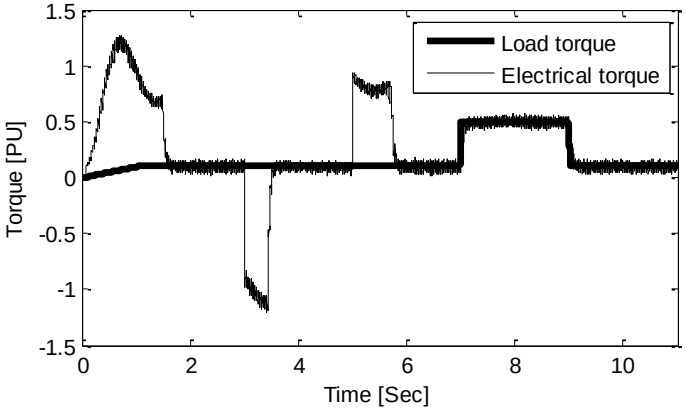


(b) Torque variations

Fig. 6.11. Transient response of the induction motor for the second local minimum

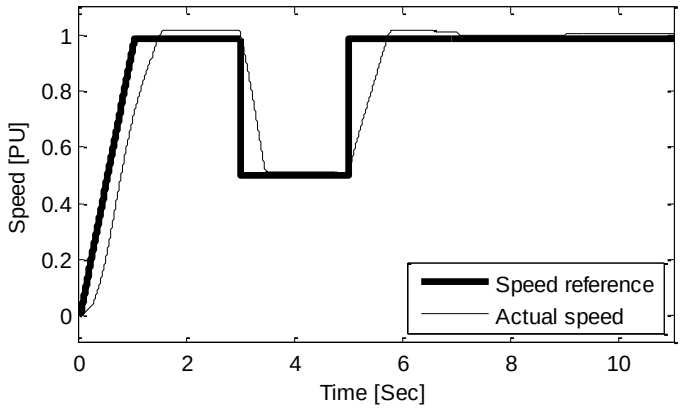


(a) Speed variations

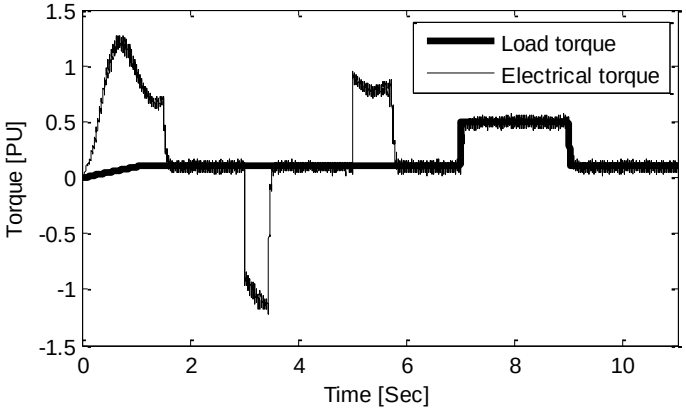


(b) Torque variations

Fig. 6.12. Transient response of the induction motor for the third local minimum

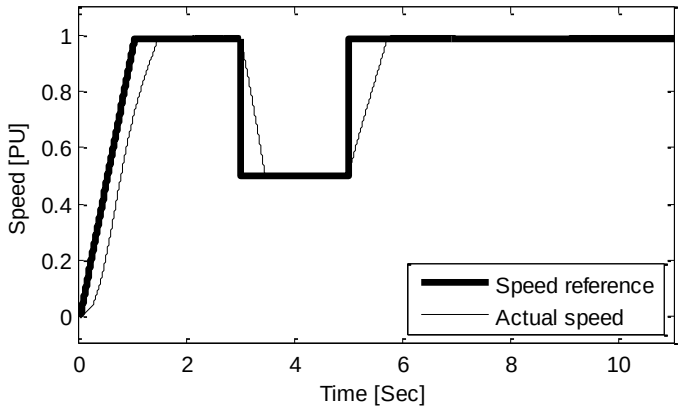


(a) Speed variations

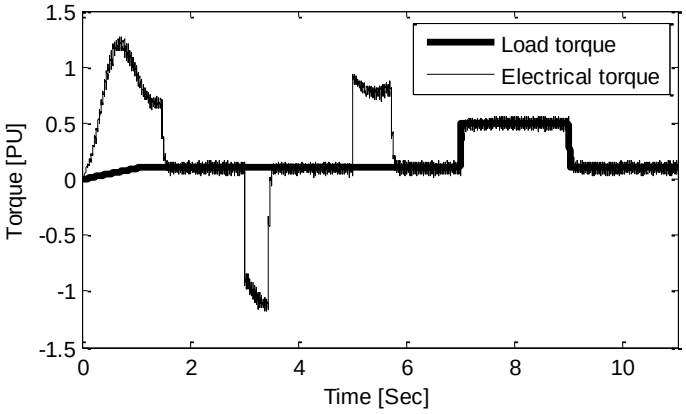


(b) Torque variations

Fig. 6.13. Transient response of the induction motor for the fourth local minimum



(a) Speed variations



(b) Torque variations

Fig. 6.14. Transient response of the induction motor for the fifth local minimum

6.3 Control System Design for an HVDC Transmission System

The developed optimization algorithm is applied to a larger case with more complexity of a high voltage direct current (HVDC) transmission system. The monopolar, 12-pulse CIGRE HVDC benchmark model [80], as shown in Fig. 6.15, is considered in this section.

The system is rated at 1000 MW with the rated dc voltage of 500 kV and the rated dc current of 2 kA. The monopolar dc link is modeled as a back-to-back system with 5 Ω resistance and the smoothing reactance on each side coincide with a capacitor in the middle. The ac system on both sides represented by Thevenin equivalent circuit with a short circuit ratio (SCR) of 2.5, giving an effective short circuit ratio (ESCR) of 1.9 which is considered as a weak ac system. Such condition makes the ac voltage more sensitive to possible variations of the ac network and consequently more difficult to control the HVDC system which is bound to operate near the stability margin.

To provide reactive power at the converter terminals, harmonic absorption filters (11-th and 13-th harmonics) together with a fixed capacitor are presented on each side. The filters will provide large amount of reactive power at the fundamental frequency. The exploitation of synchronous machines on the dc buses not only could provide the required reactive power, but also would increase the ESCR of the system.

Table 6.11 summarizes the parameters of the HVDC CIGRE benchmark model.

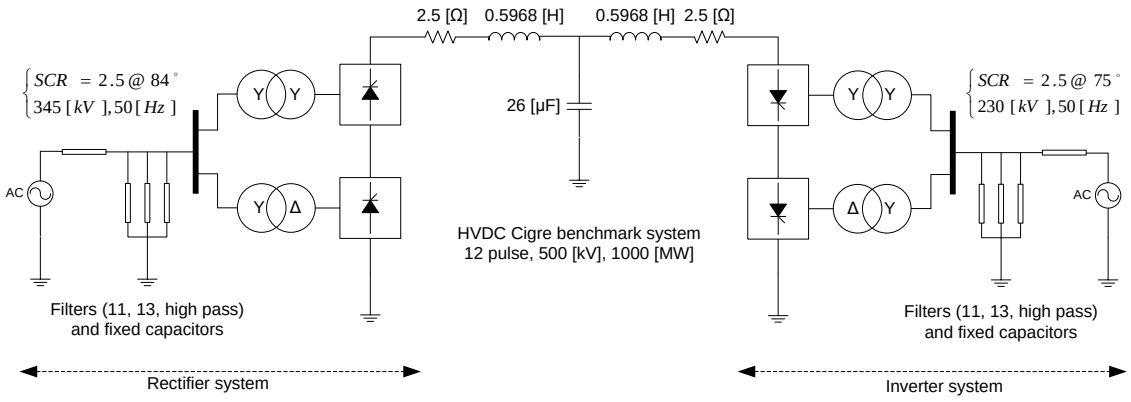


Fig. 6.15. Schematic diagram of CIGRE HVDC benchmark model

Table 6.11. Data related to the CIGRE HVDC benchmark model

Rectifier side parameters			
AC system		*Transformers (each)*	
345 kV, 50 Hz, SCR = 2.5 < 84°, ESCR = 1.9		603.73 MVA, 213.4557/345 kV, 18%	
Inverter side parameters			
AC system		*Transformers (each)*	
230 kV, 50 Hz, SCR = 2.5 < 75°, ESCR = 1.9		591.79 MVA, 209.2288/230kV, 18%	
Filters and fixed capacitors [MVAR] *(for both sides)*			
11-th harmonic	*13-th harmonic*	*Fixed capacitors*	
250	250	125	
DC link			
Resistance	*Rated power*	*Rated voltage*	*Rated current*
5 [Ω]	1000 [MW]	500 [kV]	2 [kA]

In general, the rectifier controller tends to maintain the current (current control (CC) mode) whereas the inverter is likely to maintain the voltage (constant extinction angle (CEA) control mode); that is, if the dc current exceeds the set point (I^*_d) the controller will increase the firing angle and consequently the current will be decreased; on the other hand, in the CEA mode of operation the value of the extinction angle (γ) is kept constant to prevent commutation failures. The control loops for the converters of the HVDC system is presented in Fig. 6.16.

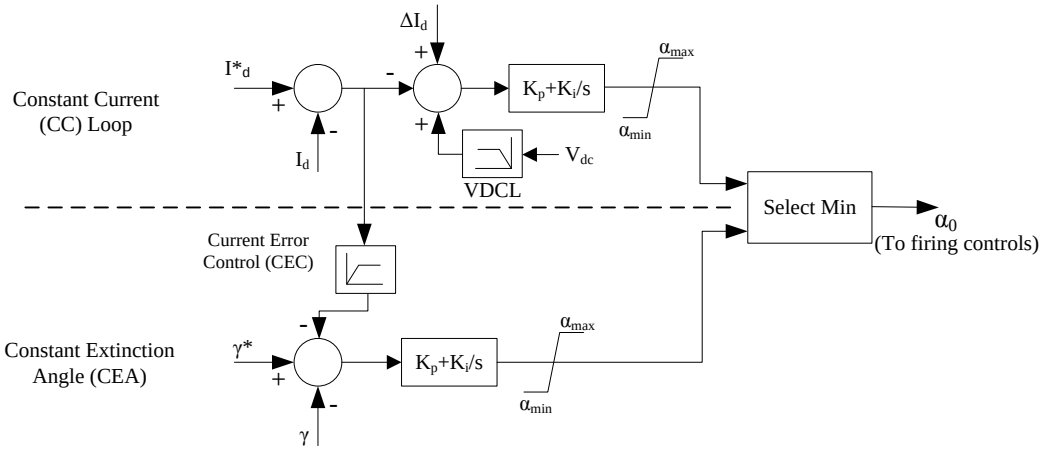


Fig. 6.16. Converter control system

The current error control (CEC) in Fig. 6.16 is used to smooth the transition between CC and CEA control modes. The overall characteristics of the converters are presented in Fig. 6.17 where the rectifier and inverter characteristics are shown by the dashed and bold lines, respectively.

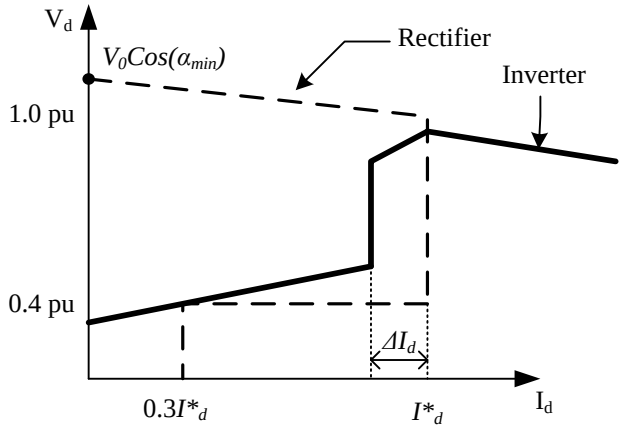


Fig. 6.17. Overall converter characteristics

The voltage dependent current limit (VDCL) will take over operation when an ac fault happens on the inverter side. It can reduce the overheating of the switches during the overlapped conduction periods by suitably adjusting the voltage and current orders [81].

6.3.1 Objective function selection

Proportional gains and integral time constants of the both controllers on the rectifier and inverter sides are considered as the input design parameters ($K_{p\text{-}R}$, $T_{c\text{-}R}$, $K_{p\text{-}I}$, $T_{c\text{-}I}$), the ranges of variations of which are presented in Table 6.12.

Table 6.12. Defining of the design space

	Rectifier side controller		*Inverter side controller*	
Parameter	$K_{p\text{-}R}$	$T_{c\text{-}R}$ [s]	$K_{p\text{-}I}$	$T_{c\text{-}I}$ [s]
Range	0.5 - 5	0.005 – 0.05	0.2 – 2	0.005 – 0.05

In order to effectively evaluate the system performance, a sequence of variations for the power order (I_{dc}) is considered, as shown in Fig. 6.18. The reference current is kept to the 80% of its nominal value for one second and after restoring back to the nominal value (1 pu) in the following cycle, it has been dropped to 50%; eventually, the set point is fixed at 0.9 pu.

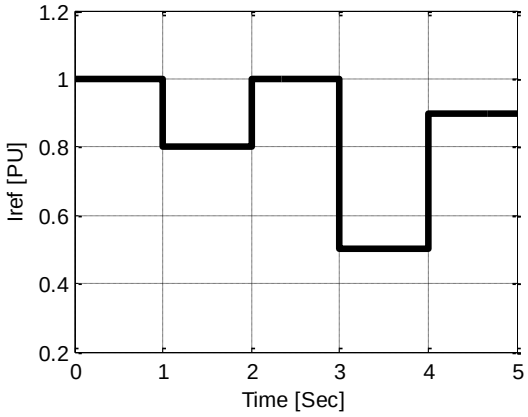


Fig. 6.18. Sequence of changes for the current order

The objective is to minimize the deviations between the actual dc link current (I_d) and the reference current ($I_{d\text{-}ref}$), subject to the variations of the reference current, by suitably selecting the controllers parameters. The mathematical definition of this objective function is presented in Eq. (6.14).

$$OF\left(K_{p-R}, T_{c-R}, K_{p-I}, T_{c-I}\right) = \int_0^{t_f} \left(I_{d-ref} - I_d\right)^2 dt \tag{6.14}$$

As it is shown, the defined objective function penalizes any deviation, both in transient and steady state, between the reference and actual dc current. Ideally, the absolute minimum of this *OF* is reached when I_d precisely follows $I_{d\text{-}ref}$, result in zero of the *OF*. However, any value that minimizes the difference between these two currents is of interest.

The first attempt, which used the above *OF*, failed to provide a satisfactory result. Fifteen local minima were detected by the algorithm; none of them showed a promising solution both in transient and steady state periods. Even though the value of the *OF* for some of the local minima was considerably small, the results were indicating sustained steady-state oscillations.

To rectify the mentioned situation, a modified *OF*, which penalizes rigorously any deviations within the steady state period, is used. The objective function also contains the transient response. The following *OF* considers both transient and steady state deviations where more contribution of the steady state values is deemed in the final aggregate *OF*.

$$OF\left(K_{p-R}, T_{c-R}, K_{p-I}, T_{c-I}\right) = \int_{t_{tr}} \left(I_{d-ref} - I_d\right)^2 dt + 100 \int_{t_{ss}} \left(I_{d-ref} - I_d\right)^2 dt \tag{6.15}$$

where t_{tr} and t_{ss} correspond to transient and steady state intervals, respectively. The first 10 cycles following each step change of the current reference, illustrated in Fig. 6.18, is considered as the transient periods; i.e. $t_{tr} = 200\ [msec]$.

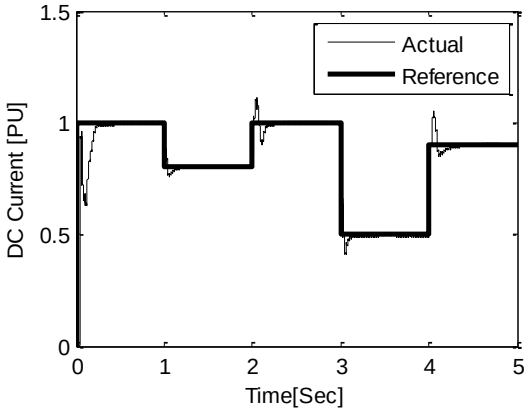
6.3.2 Optimization results

The proposed multi-modal optimization algorithm has been employed to minimize the objective function presented in Eq. (6.15). In the same vein as other cases discussed in this chapter, extraction of a closed-form formula for the objective function is prohibitively difficult. Therefore, the *OF* is searched within the intervals presented in Table 6.12 considering seven randomly placed points on each axis (with the total number of 2401 points). The simulation case has been run for 10 successive iterations and the following table summarizes the results of the optimization procedure. As it can be seen, three local minima have been detected from which the last one has a relatively large *OF* value.

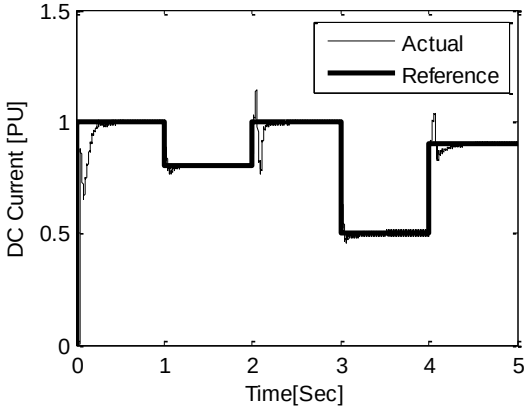
Fig. 6.19 demonstrate the result of each local minimum and as expected the third parameter set does not show an acceptable performance. It should be noted that, some of the parameters of the third local minimum slightly exceed the design space boundaries; this is due to the step changes that are embedded in the force displacement technique.

Table 6.13. Local minima detected by the proposed algorithm

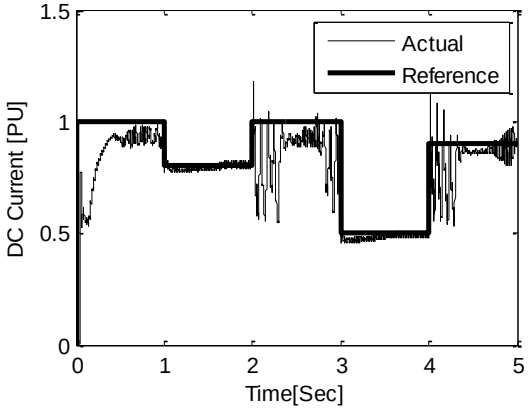
No.	$K_{p\text{-}R}$	$T_{c\text{-}R}$ [s]	$K_{p\text{-}I}$	$T_{c\text{-}I}$ [s]	*OF*
1	0.664	0.014	0.557	0.04	0.07
2	2.034	0.016	0.333	0.021	0.073
3	5.024	0.051	2	0.043	1.71



(a) First local minimum



(b) Second local minimum



(c) Third local minimum

Fig. 6.19. The corresponding dc current curves for local minima of Table 6.13

As seen, the difference between the first and second local minima is negligible, both in terms of the *OF* value and the system performance. Hence, taking account of some other factors is necessary to reveal the preeminent solution. One possible consideration is the sensitivity analysis of the *OF* with respect to the design parameters; as a less sensitive solution has advantages over a highly sensitive one. The surrogate models developed by the proposed algorithm are used to drive the sensitivity indices. Table 6.14 represents second order derivatives of the surrogate models around the two cited local minima.

Table 6.14. Second derivatives of the surrogate functions for the first two local minima

No.	$\frac{\partial^2 OF}{\partial K_{p-R}^2}$	$\frac{\partial^2 OF}{\partial T_{c-R}^2}$	$\frac{\partial^2 OF}{\partial K_{p-I}^2}$	$\frac{\partial^2 OF}{\partial T_{c-I}^2}$
1	0.294477	6358.782	2.364609	2915.306
2	55.43142	148733.5	80.12969	358606.8

The first operating point poses a dominant stable position whereas the second point highly depends on the control parameters set points, i.e. a highly sensitive solution. Consequently the first local minimum has been chosen for this design case.

Chapter 7

Conclusions, Contributions, and Future Directions

Although a number of multi-modal optimization algorithms are available, none of them are computationally efficient nor are offered in simulation based design class of optimization. The nonlinear multimodal optimization algorithm proposed in this thesis, on the other hand, is capable of locating local minima of a design case with an affordable number of simulation runs. Particularly, in the field of power systems with computationally intensive objective function evaluation, aligned with the previous study [77], it is for the first time that such an approach has been proposed, implemented and testified.

7.1 Thesis Contributions

The thesis contribution can be categorized as follows:

1. The thesis presented a thorough review of the currently available simulation based design tools with a focus on the advantages and disadvantages of each algorithm in terms of their ability to discover several minima, the required

number of simulation runs and their convergence rate for each algorithm. It also addressed some of the existing multi-modal optimization algorithms such as e*volutionary algorithms* and s*imulated annealing*.

2. In this thesis, a novel algorithm for multi-modal black box optimization of expensive objective functions was proposed. The new algorithm offers major enhancements to the recently-developed adaptive multi-resolution optimization algorithm [66]. Significant improvements in the computational intensity of the new algorithm are obtained by using surrogate modelling techniques, inherited from conventional response-surface methods (RSM) [7, 82]. The new algorithm benefited from the following facts:

   (i) Use of local surrogate models allows the new algorithm to judiciously select only a small number of new experiments in each of its iterations, thus relieving the computational intensity of the design.

   (ii) Besides shortening the design cycle, the new algorithm also generates progressively accurate surrogate models of the objective function around its discovered 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 replace intensive simulations of the original system.

3. The thesis also presented comparative studies of optimization of several multi-modal objective functions of known forms, and demonstrated the performance of the algorithm. The results clearly showed that the proposed algorithm is capable of capturing all local minima in an affordable number of simulations provided that a sufficiently small initial mesh is considered

4. Several engineering examples of design were also presented, for them an explicit objective function was practically unattainable. It was discovered that, at least for the studied cases, the multi-modality of the objective functions is a common feature among the power system cases; this can be readily addressed by the proposed algorithm.

7.2 Suggestions for Future Work

The thesis proposed a new technique for the optimal design of nonlinear multi-modal complex systems. It has paved the way for future research and development in the simulation based design area. Possible expansions of this work can be summarized as follows:

1. The thesis suggests quadratic and Gaussian functions as the simplest surrogate models for the system. However, implementation of more complex surrogate models (e.g. higher order polynomials) or surrogate models that can adaptively change during the course of optimization may improve the design procedure.

2. In the first stage of the proposed algorithm where an initial mesh is created in the multi-dimensional space of design parameters, a triangulation algorithm is to be deployed for establishment of links. Triangulation in higher dimensions, i.e. when the number of parameters to be optimized is large, becomes numerically intensive and computationally demanding. For example the Delaunay triangulation method used here becomes slow and inefficient when the dimension is larger than six or seven variables. This is a limitation on the current implementation of the algorithm; however it can be alleviated if a more computationally efficient triangulation algorithm is used.

3. The algorithm developed here regenerates the entire connections (links) subsequent to each iteration. Local mesh refinements can expedite the procedure by introducing new connections in the vicinity of the recently added points.

4. As proposed earlier, parallel processing can effectively expedite the design procedure. The intrinsic independence of the simulation runs at the first stage of mesh generation when a large number of simulations has to be conducted can be favourable to deploy parallel computation platforms. This can further be used when several local minima have been detected. The implementation of parallel processing as a quick way for the design of highly complex systems is highly recommended.

5. Another possible solution to expedite the design procedure is to replace highly detailed components models with computationally simplified equivalent models, which can closely imitate the actual system behaviour. The final design parameters can be further improved or verified by the use of the actual simulation cases.

6. A robust design of a system can also be implemented; it includes two strategically different categories of simulations, namely single-simulation and multi-simulation cases. The former includes a number of changes for values of circuit elements of a specific case; e.g. load torque or reference speed changes of a motor drive case or reference current changes of an HVDC system. This category of robust design has been implemented and tested for several cases in Chapter 6. On the other hand, the latter incorporates different simulation cases that are structurally different and the aggregate objective function comprises weighted values of a series of runs; i.e. a value for the aggregate objective function is obtained after several simulation runs. This would result in a

solution that may not be an optimum for all different scenarios, but is collectively the best. As an example, an HVDC design case can be targeted for design parameters that are suitable for several ESCR values. Although the multi-simulation robust optimization is not presented in the thesis, but it can be simply implemented by modifying the interface component between the simulation engine and the optimization program; i.e. instead of sending the value of the *OF* at the end of each run to the optimization program, the partial *OF* values to be obtained after each run and only the value of aggregate *OF* ,which will be available after a pre-defined number of simulation runs, to be sent to the optimization engine.

Appendix A

Stochastic Approximation

Stochastic approximation (SA) methods refer to optimization techniques which iteratively update the set of continuous design parameters known as "*best guess*" and search for a local optimum.

Consider the following general, optimization problem:

$$\min_{\theta \in \Theta} f(\theta) \tag{A.1}$$

where $f(\theta)$ is the objective function value estimated by simulation, θ is a vector of input parameters and $\Theta \subset \mathbb{R}^d$ is the feasible region.

Stochastic approximation algorithm first suggested by Robbins and Monro [83] using the general following form [84]:

$$\theta_{n+1} = \Pi_\theta(\theta_n - a_n \widehat{\nabla} f(\theta)) \tag{A.2}$$

where θ_n is the parameter value at the beginning of iteration n; $\widehat{\nabla} f(\theta)$ is the estimated gradient of the objective function in (A.1); Π_θ is a projection onto Θ; and $\{a_n\}$ is a sequence of positive real numbers.

To assure the convergence of the algorithm the following two assumptions are necessary: (i) a moderate-speed decreasing of the step sizes to eventually become zero; i.e. an unreasonably fast reduction may result in a convergence to a wrong point while a

slow one might never be converged, (ii) an unbiased gradient estimation; or the bias of the gradient estimate tends to zero [83]. A common set of the step sizes with the above two conditions is $\sum_{n=1}^{\infty} a_n = \infty$ and $\sum_{n=1}^{\infty} a_n^2 < \infty$ [84].

When finite difference technique is employed in Equation (A.2) to obtain an estimation for the gradient of the objective function, the resultant approach is called the Kiefer-Wolfowitz algorithm [85]. For more details on this subject refer to [86-88].

In the entire thesis, for simplicity, the stochastic property of the simulation model is ignored; that is, the results of multiple-run are deterministic. This is consistent with the simulation software and models commonly used in power systems.

References

[1] P. R. Marur and S. Srinivas, "A reduced-order finite element model for the simulation of automotive side structure crash response," *International Journal of Crashworthiness,* vol. 13, pp. 211, 2008.

[2] J. C. Vassberg, "Expectations for computational fluid dynamics," *International Journal of Computational Fluid Dynamics,* vol. 19, pp. 549, 2005.

[3] T. Aittokoski and K. Miettinen, "Cost effective simulation-based multiobjective optimization in the performance of an internal combustion engine," *Engineering Optimization,* vol. 40, pp. 593, 2008.

[4] M. Heidari, S. Filizadeh and A. M. Gole, "Support Tools for Simulation-Based Optimal Design of Power Networks With Embedded Power Electronics," *IEEE Transactions on Power Delivery,* vol. 23, pp. 1561-1570, 2008.

[5] A. M. Gole, S. Filizadeh, R. W. Menzies and P. L. Wilson, "Optimization-enabled electromagnetic transient simulation," *IEEE Transactions on Power Delivery,* vol. 20, pp. 512-518, 2005.

[6] J. Kostrowicki and L. Piela, "Diffusion equation method of global minimization: Performance for standard test functions," *Journal of Optimization Theory and Applications,* vol. 69, pp. 269-284, 1991.

[7] G. G. Wang, Z. Dong and P. Aitchison, "Adaptive response surface method - a global optimization scheme for approximation-based design problems," *Engineering Optimization,* vol. 33, pp. 707, 2001.

[8] A. Younis and Z. Dong, "Trends, features, and tests of common and recently introduced global optimization methods," *Engineering Optimization,* to appear in a forthcoming issue, 2010.

[9] "EMTDC manual," Winnipeg: Manitoba HVDC Research Center, 2003.

[10] S. Filizadeh, "Optimization-Enabled Electromagnetic Transient Simulation," *Ph. D. Dissertation, University of Manitoba, Winnipeg, Canada,* 2004.

[11] A. Gole, S. Filizadeh, R. Menzies and P. Wilson, "Optimization-enabled electromagnetic transient simulation," in *IEEE Power Engineering Society General Meeting,* 2004, pp. 1133 Vol.1.

[12] A. M. Gole, S. Filizadeh, R. W. Menzies and P. L. Wilson, "Optimization-enabled electromagnetic transient simulation," *IEEE Transactions on Power Delivery,* vol. 20, pp. 512-518, 2005.

[13] S. Filizadeh and A. M. Gole, "Optimal design of power electronic systems using electromagnetic transient simulation," in *Canadian Conference on Electrical and Computer Engineering,* 2005, pp. 450-453.

[14] F. Azadivar and Y. Lee, "Optimization of discrete variable stochastic systems by computer simulation," *Math. Comput. Simul.,* vol. 30, pp. 331-345, 9, 1988.

[15] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization.* John Wiley and Sons, 2001.

[16] F. Azadivar, "A tutorial on simulation optimization," in *Proceedings of the 24th Winter Simulation Conference,* 1992, pp. 198-204.

[17] S. Andradóttir, "A review of simulation optimization techniques," in *Proceedings of the 30th Winter Simulation Conference,* 1998, pp. 151-158.

[18] P. W. Glynn, "Optimization of stochastic systems via simulation," in *Proceedings of the 21st Winter Simulation Conference,* 1989, pp. 90-105.

[19] Y. Carson and A. Maria, "Simulation optimization: Methods and applications," in *Proceedings of the 29th Winter Simulation Conference,* 1997, pp. 118-126.

[20] R. Suri, "Infinitesimal perturbation analysis for general discrete event systems," *J.ACM,* vol. 34, pp. 686-717, 1987.

[21] A. Shapiro and Y. Wardi, "Convergence analysis of gradient descent stochastic algorithms," *J. Optimiz. Theory Appl.,* vol. 91, pp. 439-454, 11/30, 1996.

[22] P. W. Glynn, "Likelihood ratio gradient estimation for stochastic systems," *Commun ACM,* vol. 33, pp. 75-84, 1990.

[23] L. Schruben, "Simulation optimization using frequency domain methods," in *Proceedings of the 18th Winter Simulation Conference,* 1986, pp. 366-369.

[24] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal,* vol. 7, pp. 308-313, January 1, 1965, 1965.

[25] K. I. M. McKinnon, "Convergence of the Nelder-Mead Simplex Method to a Nonstationary Point," *SIAM J. Optim.,* vol. 9, pp. 148-158, 1998, 1998.

[26] J. C. Lagarias, J. A. Reeds, M. H. Wright and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," *SIAM J. Optim.,* vol. 9, pp. 112-147, 1998, 1998.

[27] M. J. Box, "A New Method of Constrained Optimization and a Comparison With Other Methods," *The Computer Journal,* vol. 8, pp. 42-52, April 1, 1965, 1965.

[28] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

[29] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing,* vol. 4, pp. 65-85, 06/01, 1994.

[30] H. P. Schwefel, *Evolution and Optimum Seeking: The Sixth Generation.* John Wiley & Sons, Inc., 1993.

[31] T. Bäck, "An Overview of Evolutionary Algorithms for Parameter Optimization," *Evolutionary Computation,* vol. 1, pp. 1-23, 1993.

[32] R. A. Rutenbar, "simulated annealing algorithms: An overview," *IEEE Circuits and Devices Magazine,* vol. 5, pp. 19-26, 1989.

[33] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," *Journal of Statistical Physics,* vol. 34, pp. 975-986, 03/01, 1984.

[34] F. Glover, E. Taillard and E. Taillard, "A user's guide to tabu search," *Annals of Operations Research,* vol. 41, pp. 1-28, 03/01, 1993.

[35] I. H. Osman, "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Annals of Operations Research,* vol. 41, pp. 421-451, 12/01, 1993.

[36] F. Glover, "Tabu Search: A Tutorial," *Interfaces,* vol. 20, pp. 74-94, Jul. - Aug., 1990.

[37] D. Goldsman, "Ranking and selection in simulation," in *Proceedings of the 15th Winter Simulation Conference - Volume 2,* 1983, pp. 387-394.

[38] J. R. Swisher, S. H. Jacobson and E. Yucesan, "Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey," *ACM Trans. Model. Comput. Simul.,* vol. 13, pp. 134-154, 2003.

[39] D. THOMPSON, "Response surface experimentation," *J. Food Process. Preserv.,* vol. 6, pp. 155-188, 1982.

[40] R. H. Myers and D. C. Montgomery, *Response Surface Methodology : Process and Product Optimization using Designed Experiments.* New York: Wiley, 1995.

[41] G. Vicente, A. Coteron, M. Martinez and J. Aracil, "Application of the factorial design of experiments and response surface methodology to optimize biodiesel production," *Industrial Crops and Products,* vol. 8, pp. 29-35, 3, 1998.

[42] C. G. Bucher and U. Bourgund, "A fast and efficient response surface approach for structural reliability problems," *Struct. Saf.,* vol. 7, pp. 57-66, 1, 1990.

[43] F. P. Pike, J. R. Nelli, E. E. Erickson, D. S. Arnold, G. E. P. Box and J. S. Hunter, "The application of statistical procedures to a study of the flooding capacity of a pulse column," North Carolina State Coll., Raleigh, Tech. Rep. ORO-140, 1954.

[44] R. Hooke and T. A. Jeeves, ""Direct Search" Solution of Numerical and Statistical Problems," *J.ACM,* vol. 8, pp. 212-229, 1961.

[45] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal,* vol. 7, pp. 149-154, 1964.

[46] B. V. Shah, R. J. Buehler and O. Kempthorne, "Some Algorithms for Minimizing a Function of Several Variables," *Journal of the Society for Industrial and Applied Mathematics,* vol. 12, pp. 74-92, Mar., 1964.

[47] G. Singh and D. Kalyanmoy Deb, "Comparison of multi-modal optimization algorithms based on evolutionary algorithms," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation,* 2006, pp. 1305-1312.

[48] P. Ballester and J. Carter, "Real-Parameter Genetic Algorithms for Finding Multiple Optimal Solutions in Multi-modal Optimization," *Genetic and Evolutionary Computation - GECCO 2003,* pp. 199-199, 2003.

[49] I. G. Tsoulos and I. E. Lagaris, "MinFinder: Locating all the local minima of a function," *Comput. Phys. Commun.,* vol. 174, pp. 166-179, 1/15, 2006.

[50] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and their Application*, 1987, pp. 41-49.

[51] A. Corana, M. Marchesi, C. Martini and S. Ridella, "Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm," *ACM Trans.Math.Softw.,* vol. 13, pp. 262-280, 1987.

[52] S. W. Mahfoud, "Niching methods for genetic algorithms," *Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA*, 1995.

[53] A. Petrowski, "A clearing procedure as a niching method for genetic algorithms." in *Proceedings of Third IEEE International Conference on Evolutionary Computation(ICEC'96),* Piscataway, NJ, 1996, pp. 798-803.

[54] X. Yin and N. Germay, "A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization," in *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms,* 1993, pp. 450-457.

[55] O. Mengsheol and D. Goldberg, "Probabilistic crowding: Deterministic crowding with probabilistic replacement," in *Proceedings of the Genetic and Evolutionary Computation Conference-1999(GECCO-99),* 1999, pp. 409-416.

[56] J. Li, M. E. Balazs, G. T. Parks and P. J. Clarkson, "A Species Conserving Genetic Algorithm for Multimodal Function Optimization," *Evol. Comput.,* vol. 10, pp. 207-234, 09/01, 2002.

[57] S. Salhi and N. M. Queen, "A hybrid algorithm for identifying global and local minima when optimizing functions with many minima," *Eur. J. Oper. Res.,* vol. 155, pp. 51-67, 5/16, 2004.

[58] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen and J. Sondergaard, "Space mapping: the state of the art," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, pp. 337-361, 11/21, 2004.

[59] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan and P. Kevin Tucker, "Surrogate-based analysis and optimization," *Prog. Aerospace Sci.,* vol. 41, pp. 1-28, 1, 2005.

[60] A. I. J. Forrester, N. W. Bressloff and A. J. Keane, "Optimization using surrogate models and partially converged computational fluid dynamics simulations," *Proceedings*

of the Royal Society A: Mathematical, Physical and Engineering Science, vol. 462, pp. 2177-2204, July 8, 2006, 2006.

[61] Zongzhao Zhou, Yew Soon Ong, Prasanth B. Nair, Andy J. Keane and Kai Yew Lum, "Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 37, pp. 66-76, 2007.

[62] D. C. Montgomery, *Design and Analysis of Experiments.* USA: John Wiley and Sons, 2001.

[63] J. Sacks, W. J. Welch, Toby J. Mitchell and H. P. Wynn, "Design and Analysis of Computer Experiments," *Statistical Science,* vol. 4, pp. 409-423, Nov., 1989.

[64] S. Filizadeh, A. M. Gole, D. A. Woodford and G. D. Irwin, "An Optimization-Enabled Electromagnetic Transient Simulation-Based Methodology for HVDC Controller Design," *IEEE Transactions on Power Delivery,* vol. 22, pp. 2559-2566, 2007.

[65] A. M. Gole, S. Filizadeh, R. W. Menzies and P. L. Wilson, "Optimization-enabled electromagnetic transient simulation," *IEEE Transactions on Power Delivery,* vol. 20, pp. 512-518, 2005.

[66] K. Kobravi and S. Filizadeh, "An adaptive multi-modal optimization algorithm for simulation-based design of power-electronic circuits," *Engineering Optimization,* vol. 41, pp. 945-969, 2009.

[67] J. R. Shewchuk, "Delaunay refinement algorithms for triangular mesh generation," *Computational Geometry,* vol. 22, pp. 21-74, 5, 2002.

[68] P. O. Persson, "Mesh generation for implicit geometries," *Ph. D. Dissertation, Massachusetts Institute of Technology, Department of Mathematics, Cambridge, United States,* 2004.

[69] M. Bern and P. Plassmann, "Mesh generation," in *Handbook of Computational Geometry,* J. -R. Sack and J. Urrutia, Eds. Elsevier Science, 2000.

[70] S. J. Owen, "A survey of non structural mesh generation technology," in *Proceedings of the 7th International Meshing Roundtable,* Sandia Nat. Lab., 1998, pp. 239-267.

[71] K. Kobravi, "Optimization-enabled transient simulation for design of power circuits with multi-modal objective functions," *M. Sc. Thesis, University of Manitoba, Winnipeg, Canada,* 2007.

[72] I. N. Bronshtein and K. A. Semendyayev, *Handbook of Mathematics.* New York: Van Nostrand Reinhold, 1985.

[73] P. O. Persson and G. Strang, "A Simple Mesh Generator in MATLAB," *SIAM Rev,* vol. 46, pp. 329-345, 2004, 2004.

[74] Å. Björck, *Numerical Methods for Least Squares Problems.* SIAM, 1996.

[75] N. Mohan, T. M. Undeland and W. P. Robbins, *Power Electronics: Converters, Applications, and Design.* USA: John Wiley & Sons, 2002.

[76] S. Filizadeh, A. M. Gole, D. A. Woodford and G. D. Irwin, "An Optimization-Enabled Electromagnetic Transient Simulation-Based Methodology for HVDC Controller Design," *IEEE Transactions on Power Delivery,* vol. 22, pp. 2559-2566, 2007.

[77] P. C. Sen, *Principles of Electric Machines and Power electronics.,* Wiley, 1996.

[78] P. C. Krause, O. Wasynczuk and S. D. Sudhoff, *Analysis of Electric Machinery and Drive Systems.* Wiley Inter-science, 2002.

[79] N. P. Quang and J. Dittrich, *Vector Control of Three-Phase AC Machines: System Development in the Practice.* Springer, 2008.

[80] M. Szechtman, T. Wess and C. V. Thio, "A benchmark model for HVDC system studies," in *International Conference on AC and DC Power Transmission,* 1991, pp. 374-378.

[81] J. Arrillaga, *High Voltage Direct Current Transmission.* IEE Power Eng. Series, 1998.

[82] R. H. Myers, D. C. Montgomery and C. M. Anderson-Cook, *Response Surface Methodology : Process and Product Optimization using Designed Experiments.* New York: WileyBlackwell, 2009.

[83] H. Robbins and Sutton Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics,* vol. 22, pp. 400-407, Sep., 1951.

[84] M. Fu, "Optimization via simulation: A review," *Annals of Operations Research,* vol. 53, pp. 199-247, 12/28, 1994.

[85] J. Kiefer and J. Wolfowitz, "Stochastic Estimation of the Maximum of a Regression Function," *The Annals of Mathematical Statistics,* vol. 23, pp. 462-466, Sep., 1952.

[86] L. Ljung, G. Pflug and H. Walk, *Stochastic Approximation and Optimization of Random Systems.* Basel: Birkhauser Verlag, 1992.

[87] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications.* New York: Springer-Verlag, 1997.

[88] H. J. Kushner and D. C. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* New York: Springer, 1978.