Optimization-Enabled Electromagnetic Transient Simulation

By

Shaahin Filizadeh

A Thesis

Submitted to the Faculty of Graduate Studies in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

Optimization-Enabled Electromagnetic Transient Simulation

BY

Shaahin Filizadeh

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

Of

DOCTOR OF PHILOSOPHY

Acknowledgement

The author would like to express his deepest gratitude to his Thesis Advisor, Prof. Ani Gole, for his guidance, insightful comments and support throughout all the stages of this research. This research could not be completed as in its present form without Prof. Gole's encouragement, and his efforts in providing the author with several opportunities to present this work as widely as possible.

I would also like to thank the Manitoba HVDC Research Center for their financial and invaluable technical support. Special thanks go to Mr. Paul Wilson, Mr. Randy Wachal, Mr. John Nordstrom and Dr. Rohitha Jayasinghe for their support and help.

Financial support from the Manitoba Hydro is also deeply acknowledged.

I would also like to thank my parents for their continuous support and encouragement over all the long years I spent in school.

Finally I should express my most sincere appreciation to my beloved wife, Leila, who never failed to encourage me in all the stages of my Ph.D. studies, and while she was a full-time graduate student, she always showed priceless support beyond my imagination.

To Leila

Abstract

This thesis deals with the development of a new set of tools and techniques for the optimal design of nonlinear circuits, such as power electronic systems. The incentive for this work arises from the fact that currently used design procedures show little practical capability in modern nonlinear design problems, where several parameters have to be optimally selected.

The new tools are based on interfacing an optimization algorithm with a transient simulation program. The former is used to strategically select parameter sets, while the performance of the system is evaluated using the latter. The thesis describes the methods adopted to interface the parts and also reviews some representative optimization algorithms that are interfaced with the simulation program PSCAD/EMTDC. Advanced issues such as inclusion of robustness into the design procedure as well as interfacing gradient-based optimization algorithms are also investigated. The proposed tools are very efficient even when no explicit mathematical objective function is available.

The proposed tools are used to design several nonlinear and switching circuits with different design specifications. The results confirm that the developed tools are orders of magnitude faster than the conventional methods in converging to the optimal parameter set while producing results of much higher accuracy.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Power System Simulation Tools

Modern electric power networks, which carry out the tasks of generation, transmission and distribution of electric energy, are large-scale, nonlinear systems that exhibit complicated dynamic behavior. This arises from the fact that such systems usually contain a large number of both linear and nonlinear elements, different types and layers of controls, and in an increasing number of cases, power electronic equipment used for purposes such as real and reactive power control and voltage regulation. Analysis, design and operation of such complex systems require various types of advanced tools and techniques to be employed.

Power system simulation software consists of tools that can be used for both the analysis and design of power networks. In these tools, mathematical equations of the system are formed and solved using analytical or numerical methods, and thereby they provide a sound understanding of the behavior of the system without necessarily recourse to costly or even destructive laboratory or field experiments. The type of the mathematical formulation and solution methods used in various types of simulation tools differ depending on the type of the studies to be carried out. The level of detail considered in modeling power system apparatus is also different and depends on the nature of the problem(s) under consideration, mathematical modeling complexities and of course the simulation platform(s) available [1,32].

1.2 Electromagnetic Transient Simulation Programs

Electromagnetic transients simulation programs, also referred to as emtp-type programs, form an important category of power system simulation tools, and are used to precisely describe the short term transient behavior of a power system. In these tools, time domain differential equations of the system, including individual elements, controls, semiconductor switches, etc., are set up and solved using numerical integration methods, such as the trapezoidal rule [2].

Emtp-type programs are used in a wide range of applications such as tuning of power system controls, study of stress on components during transients, harmonic analysis and power quality studies, etc. [3,4,5,6]. Since the level of detail used in the modeling of components is very high, emtp programs are usually computationally demanding and completion of simulations, even for a relatively short period of time, can be quite time-consuming. Using larger simulation time steps can speed up the process, however it will decrease the accuracy of the results and in case inappropriate integration methods are used, will even result in numerical instability of the solution.

Since emtp-type programs can provide a detailed description of the transient behavior of a power system, they are extensively used in design problems, where it is necessary to accurately study the performance of a designed component, and to examine its impact on and its interactions with the rest of the network. For example, in a controller tuning problem, a transients simulation program can precisely simulate the behavior of the controlled variable for a given set of controller gains, and provide the designer with an accurate estimation of the performance of the designed controller. Such detailed analysis of a design can provide a high degree of confidence in its proper operation when it is actually implemented.

1.3 Traditional Approaches to the Design Using Simulation Tools

The major concern associated with the use of emtp-type programs in design problems is that the design process is often accompanied with a large amount of trial and error. The designer often has to simulate the system several times before reaching the desired set of design parameters so that design specifications and constraints are best met. With the increasing complexity of power systems, their design issues are becoming more challenging and often include multi-objective, multi-variable problems. Many such design problems require repeated application of transients simulation to ensure that all the details and interactions are effectively considered; but this on the other hand, causes the computational burden to become overwhelmingly large.

Two traditional approaches widely used in simulation tools are Monte Carlo analysis and the multiple-run approach. The Monte Carlo method, which is a branch of experimental mathematics, is based on experiments with random numbers and has numerous applications in various fields of engineering. This type of analysis is used to study the behavior of both stochastic and deterministic systems and usually requires several simulation runs to be conducted [7]. Another attempt to simplify the design process using emtp-type programs is an automated search engine called 'multiple-run', which has been added to some of the simulation programs [8]. In the method of multiple-run, design parameters are varied on a user-specified basis, e.g. in linear, logarithmic or even random steps, and for every parameter set so obtained, a simulation run is conducted. At the end of each run, an index is assigned to the parameter set used, which shows the conformity of its performance to the design objectives set by the user. At the end of the whole search process, the best parameter set is chosen simply by selecting the one that has the best figure of merit for its performance.

The method of multiple-run, which is usually used in emtp-type programs, is obviously inefficient in searching for the optimal value of design parameters; firstly because it deploys no intelligence in the way it spans the search space. The search process can become very lengthy and may include regions of the space that do not contain the optimum. The pre-specified method for varying the parameters to span the desired range, e.g. linear or logarithmic, searches the areas that could have been otherwise identified and eliminated if better search algorithms could be used.

The second major drawback of the multiple-run approach is the poor accuracy of its results, caused by the discrete nature of the steps taken during the search process. While taking large steps can decrease the total search time, it will result in poor approximations of the optimum. Finer searching by reducing the increments of design variables produces results of better accuracy, but at the cost of increased computer search time.

1.4 Research Objectives of the Thesis

The inefficiency of the method of multiple-run is a direct consequence of its search philosophy. Advanced optimization algorithms, on the other hand, provide a powerful optimum-seeking means. In an optimization algorithm, an Objective Function (OF), which is a figure of merit for a point (parameter set) and measures the conformity of the performance of the point with the user-specified objectives, is tested for a number of trial points (parameter sets). The ultimate goal of an optimization algorithm is to find a point that has the best (optimal) OF value [9]. The parameter sets to be tested are strategically chosen by the optimization algorithm, and hence the optimal parameter set can be determined with less computation.

The combination of a transient simulation program with an optimization algorithm has the potential to be an extremely powerful design tool. Development of this new tool, studying the performance of the new design approach in real-world design problems, and investigating the opportunities unlocked by such a tool are the main objectives of this thesis.

The objective of this study is to use transient simulation as an engine for evaluating the objective function, which is optimized using an external optimization algorithm. Since in this tool, the search will be done through a dedicated algorithm, the entire design process can be completed in a number of runs that is orders of magnitude smaller than what is possible with the method of multiple-run. The accuracy of the results can also considerably be improved and it is determined by the convergence (search termination) criteria in the optimization algorithm. The true power of the new tool will be in providing the capability of optimally designing a nonlinear system where no explicit formula for the design objective function in terms of the design parameters is available.

Alleviation of the undesirable features of the multiple-run approach boosts the applicability of transient simulation programs in design problems by not only speeding up the design process but also increasing the accuracy of the results.

1.5 Thesis Organization

As stated above, the thesis aims at development of a new design tool based on interfacing electromagnetic transient simulation programs with optimization algorithms. The thesis is thus organized in such a way to provide a thorough understanding of the methods and techniques deployed and also to demonstrate the power of the new tool in solving real-life design problems.

Following the introduction given in this chapter, chapter 2 presents an overview of the electromagnetic transient simulation programs, which are one of the two building blocks of the new tool. The chapter reviews the general basic theory of transient simulation of power systems, and in particular describes the mechanics of the PSCAD/EMTDC simulation tool, which is a widely-used electromagnetic transients simulation program [8]. PSCAD/EMTDC is the simulation software that is used in this research for interfacing with optimization algorithms. The chapter will also examine the design procedure using PSCAD/EMTDC and will demonstrate the shortcomings of the conventional methods currently used. Later in chapter two, the applicability of optimization algorithms in overcoming these drawbacks will be investigated. Statement of design specifications in the form of an optimization problem, which in fact is the key to the application of optimization methods in power system design problems, will conclude chapter two.

Chapter 3 focuses on the development of the interface between transient simulation and optimization. This chapter will discuss some representative methods for inclusion of design specifications into optimization objective functions; these methods will later be used in the design of a number of example cases.

In chapter 4, a number of representative optimization algorithms, namely Simplex method of Nelder and Mead, Hooke-Jeeves optimization method, Genetic Algorithms and Golden Section optimization method will be discussed. These methods are used for interfacing with PSCAD/EMTDC in this research. This chapter will discuss the reasons why non-gradient-based methods are generally preferred for interfacing with simulation programs.

Chapter 5 will present a number of design examples tackled with the new tool. It will address not only different types of design problems and objectives, but also the applicability of various non-gradient-based algorithms. The results will be presented along a discussion of their significance where necessary.

In chapter 6, more advanced design issues will be discussed. Specifically the methods and techniques required for inclusion of robustness into the optimal design procedure are considered. A modified interface between optimization and transient simulation, developed to include robustness, will be presented in this chapter. Examples of the application of this modified tool are also given.

Chapter 7 is devoted to the application of gradient-based methods in the design using transient simulation tools. The chapter will discuss, among other things, the limitations and computational complexities of such algorithms. A number of example cases of the application of these methods will be presented in this chapter.

Chapter 8 will provide a summarizing overview of the entire thesis and will also signify the thesis contributions and will provide recommendations for future research on the topic.

A list of references cited throughout the thesis will conclude the thesis.

Chapter 2

Electromagnetic Transient Simulation Tools

2.1 General Structure of an Electromagnetic Transient Simulation Tool - PSCAD/EMTDC

Simulation of the behavior of power systems is an inevitable task in their analysis, design and operation. This is because an accurate simulation can greatly reduce the need for laboratory or field experiments that are usually expensive, time-consuming and in some cases even destructive.

As mentioned earlier, the type of modeling and the level of detail considered in power system simulation tools depend largely on the type of phenomena to be studied. For example, while steady-state phasor models can adequately describe a power system for load flow studies, transient simulation of the same system requires detailed modeling of individual components considering the dynamics of all the elements involved.

The focus throughout the rest of this chapter will be on a class of power system simulation tools, namely electromagnetic transients simulation programs (hereinafter also referred to as emtp-type programs). These programs are used to study the short-term transient behavior of power systems of virtually any complexity. In the following, we will mainly focus on the general structure of PSCAD/EMTDC simulation tool.

Developed originally in early-80's by Dennis Woodford at Manitoba Hydro with input from the University of Manitoba, EMTDC was primarily a tool for simulating transient phenomena in HVDC systems [8]. Later versions of the program were expanded to include ac systems, control circuitry, transmission lines, machines, etc. [10]. Currently EMTDC is capable of simulating electromagnetic and electromechanical transients in both ac and dc power systems, including systems with switching devices such as power electronic converters. Besides a graphical user interface (PSCAD) enhances the interface of the user with the program through a graphical draft on which the circuit elements are placed and connections are made [8].

The internal structure of EMTDC follows the method originally introduced by Dommel [2] and is based on time domain equations of individual network elements, which are subsequently formed into the admittance matrix (**Y**-matrix) of the network. Sources and electric machines are modeled as current injections to the nodes. Control signals are also interfaced with the electrical network during the simulation.

The solution method used in EMTDC is the trapezoidal integration rule [2]. The most important feature of this integration rule is its stability-preservation in linear systems, i.e. large time steps will not cause numerical instability in the solution, although they may deteriorate the accuracy of the results.

2.2 Applications of EMTDC

Transient simulation programs, such as PSCAD/EMTDC, are used in a wide variety of power system analysis and design problems [8,10,11]. Their ability to provide a detailed description of the short-term transient behavior of complex systems is a major advantage that can be used to increase the functionality and reliability of a new design. Some typical applications of PSCAD/EMTDC are as follows:

- Control systems design and coordination of power system controls;
- Power electronic applications in power systems and HVDC;
- Filter design and harmonic analysis;
- Over-voltage studies in power systems due to lightning and other disturbances;
- Transient behavior of power transformers, etc.

An important feature of PSCAD/EMTDC is that it allows for relatively easy connection to user-defined components. Using this feature, users can design their own components and interface them with EMTDC and in this way, the program can be used in cases other than what is possible using its standard master library of components. This feature makes PSCAD/EMTDC an ideal host for external programs to be interfaced.

2.3 EMTDC Structure

In order to understand the interfacing mechanism to the externally designed optimization algorithms, it is important to know the internal structure of the EMTDC. Fig. 2.1 shows a simplified flowchart of the main body of EMTDC [8]. The subroutines DSDYN (master dynamics subroutine) and DSOUT (output definition subroutine) are the main sections where user-defined codes can be placed and interfaced with the program. Besides hosting user-defined code segments, these subroutines handle the dynamics of the circuit and generate the output signals to be displayed or saved. It is also clear that multiple-run is an integrated part of the main body of the program that can be enabled if so needed.

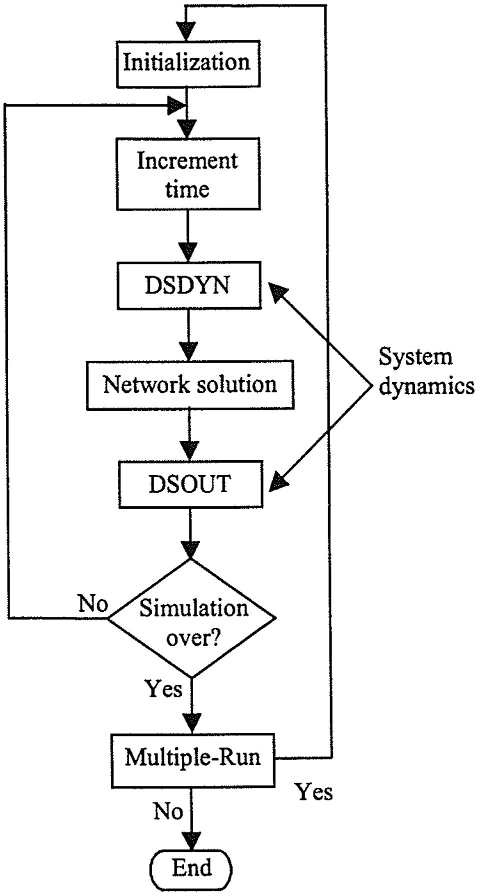


Fig. 2.1 General structure of PSCAD/EMTDC

The connections to external programs are made through either of DSDYN or DSOUT subroutines depending on the type of the external program. Output processing programs are usually interfaced using DSOUT, while other programs such as optimization are linked through DSDYN. The process of interfacing external programs may sometimes have programming complexities and require extensive coding to ensure that variables are properly stored and retrieved, and that the external code does not interfere with the main body of EMTDC.

2.4 Conventional Design Procedures Using EMTDC

As mentioned earlier, the capability of EMTDC in providing a detailed description of the transient as well as the steady-state behavior of complex systems, is a major incentive for its application in design of such systems. Transient simulation of a network for a given set of design parameters yields an accurate measure of the closeness of the system performance to the pre-specified design objectives. The question challenging the designer is how to determine the optimum set of design parameters such that design objectives are satisfied as closely as possible. One can think of the design procedure as a search problem, in which a number of parameters (design parameters) are to be determined.

Multiple-run (MR) and random-based search methods, e.g. Monte-Carlo (MC), are the major approaches conventionally used in transient simulation programs. In the following we review these two methods and give an example of their application in a simple design problem. The example will not only show how these methods are practically used, but also serves to pinpoint the major drawbacks of these methods. Developing methods to remedy these shortcomings is the main incentive for this research.

2.4.1 Multiple-Run (MR)

In the method of multiple-run, design parameters are varied in a discrete user-specified manner, e.g. linear or logarithmic, to span a feasible range of variation for each of them. For a design parameter x_i, a number of samples equal to n_i are chosen (in linear or logarithmic steps, for example) from the interval $[x_i(min), x_i(max)]$. In an N-parameter design problem, where the set of design parameters is $\mathbf{x} = (x_1, x_2, \cdots, x_N)$, the total number of sets generated using the above discrete method will be

$$M = n_1 \times n_2 \times \cdots \times n_N \tag{2.1}$$

For example, in a 2-parameter design problem, where x_1 and x_2 have n_1 and n_2 samples, respectively, the total number of sets of design parameters (points in the form of (x_1,x_2)) will become n_1n_2, and they will form a 2-D grid of points in the x_1-x_2 plane, as shown in Fig. 2.2, which shows the grid points as well as the contour plot of the respective objective function. Note that the contour plot is not known a-priori, and it is shown in Fig. 2.2 merely to facilitate the description of the method of MR.

Having formed the N-dimensional grid of candidate sets of design parameters, a simulation run is conducted for each of them. At the end of each simulation, a numerical value indicating the conformity of the performance of the system under the current parameter set to the design objectives will be assigned to the parameter set. In this thesis the assignment of this numerical index is such that a low value of the index shows a high level of conformity, i.e. a desirable design. The ultimate objective will naturally be to find a set of design parameters with as low of index value as possible. Selection of the best performing parameter set is done upon completion of simulations for the entire set of parameter sets in the grid.

The method of multiple-run can provide an insight into the performance of the system for a range of variations of the design parameters. This can later on lead to a finer search over a region, where the best set of parameters is most likely located. One appealing feature of this method is that, the chance of estimating global optimum is quite high provided that the original search region includes the global optimum.

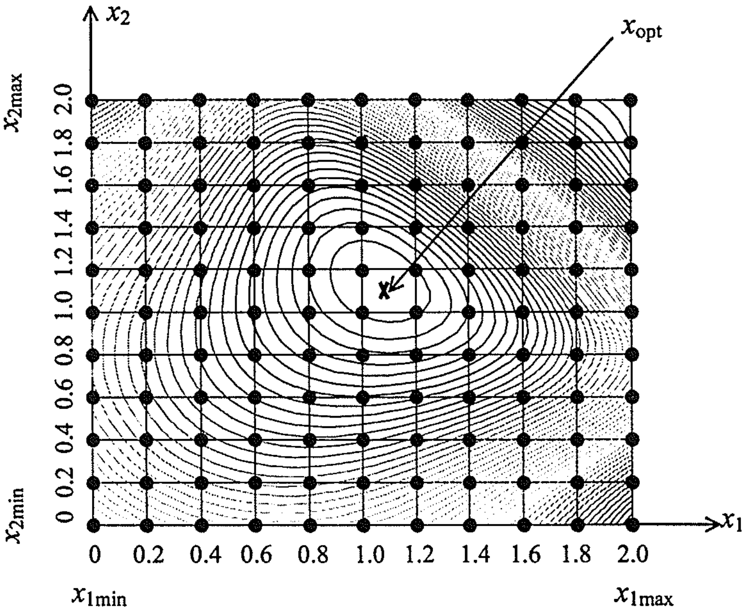


Fig. 2.2 Grid established for an MR design of a two-variable problem

The most severe drawback of this method, however, is that the completion of the simulations for the entire points in a grid, even for a very coarse grid, can be extensively time-consuming. Taking into consideration the fact that most design problems involve several design parameters, it is evident from (2.1) that even with a few samples for each of the parameters, the resulting grid will contain an excessively large number of points. For example, in a four variable design problem, where each of the parameters takes on only 10 values, the resulting grid will contain 10,000 points, for each of which a simulation run has to be carried out. Also note that with such a coarse grid, the final optimal solution may be a coarse approximation of the true optimum. For example, the MR design for the case shown in Fig. 2.2 results in the selection of one of the four points surrounding the optimum denoted by X.

These two shortcomings, namely extensive simulations required and limited accuracy of the results, limit the application of this method in most of the modern day design problems. Later in this chapter, these issues will be illustrated through a simple design problem.

2.4.2 Random-Based Methods

Random-based search methods form a large category of design and analysis tools, and are especially used in sensitivity analysis of complex designs. These methods are often classified under the general theme of Monte Carlo (MC) analysis [7].

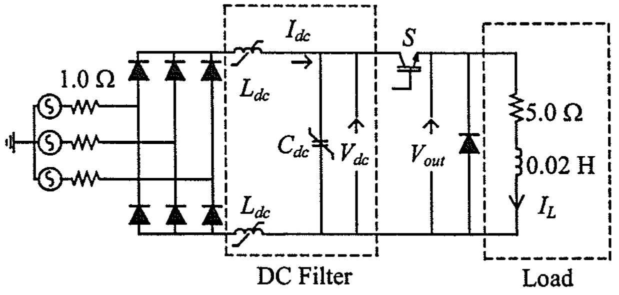
In EMTDC a random search can be carried using the core engine of the multiple-run; however, instead of the systematic, grid-based method of generating points in the search space used in MR, candidate points are generated using a random number generator. This type of analysis can be very useful in determining the statistical properties of a design. In such an approach, each of the design parameters is allowed to vary in a specified range, which shows the tolerance band of that parameter around its nominal value. Extensive simulations are carried out while design parameters are varied randomly (based on a given probability density function such as normal or uniform) in their respective ranges. The behavior of the design is finally expressed in terms of the statistical properties (e.g. mean and variance) of the variables of interest. The results can reveal sensitivity of the design with respect to the variations of the design parameters in their allowable tolerance bands.

Although the above description mostly applies to a case where design parameters are already selected, the same approach can be used to find the appropriate set of design parameters as well. To do so, one can specify permissible ranges for the parameters and using uniform distribution for each of them, carry out the random search. The setting of parameters that has the most favorable performance can be singled out at the end of the simulations.

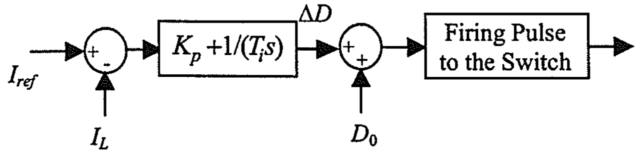
An important feature, shared by both the random search methods and the method of MR, is that each of the design parameters is searched in its own respective range and there is no need to introduce further constraints or scale factors. We will show that these overhead manipulations are unavoidable companions to most of the optimization methods described in later chapters of the thesis.

2.5 An Example of Design Using EMTDC

In this section, we demonstrate the use of conventional design procedures, the method of MR and random-based search strategies, in a simple design problem. The problem considers design of a proportional-integral (PI) controller for a dc-dc converter, as shown in Fig. 2.3, in which the proportional gain (K_p) and integral time constant (T_i) are to be selected.



(a) Dc-Dc converter



(b) Load current control system

Fig. 2.3 Schematic diagram of the circuit and its control system

The objective of the design is to obtain the optimum dynamic response of the load current to a specified step change in the reference current, in terms of the following specifications.

1. The load current should follow the step change as fast as possible;
2. Dynamic response of the load current should have minimum overshoot;
3. Steady-state error between the actual and reference currents should be minimized.

The above objectives can be quantitatively incorporated into an optimization objective function as given below.

$$ISE(K_p, T_i) = \int_{t=T_1}^{T_F} \left(1 - \frac{I_L}{I_{ref}}\right)^2 dt \qquad (2.2)$$

where T_1 and T_F are the step time and final simulation time, respectively. In this problem T_1 and T_F are equal to 0.1 sec and 1.0 sec, respectively. The reference current is set to 30 A. Note that under situations where the actual and reference currents closely match, the Integral Square Error (*ISE*) objective function defined above will have a small value. Its minimum possible value of zero occurs when the two currents exactly match at every instant of time. The objective of the search will therefore be to find a pair of (K_p, T_i) such that *ISE* attains as small a numerical value as possible.

Table 2.1 shows the summary of the design procedure carried out using the method of MR as well as a random search method. Fig. 2.4 shows the contour plot of the optimization objective function, which is obtained using the results of the MR simulation. Fig. 2.5 shows the results of the random search method. The trial points generated through the MR and random search are marked in the figures.

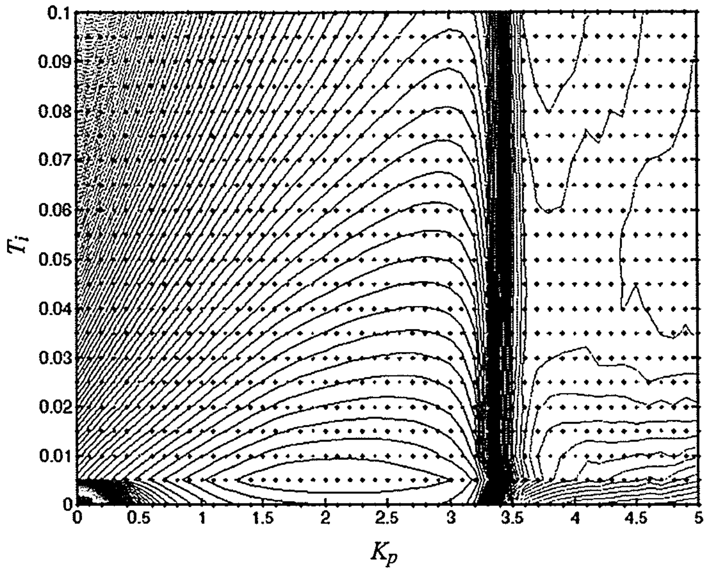


Fig. 2.4 Grid of points and the contour plot generated by MR

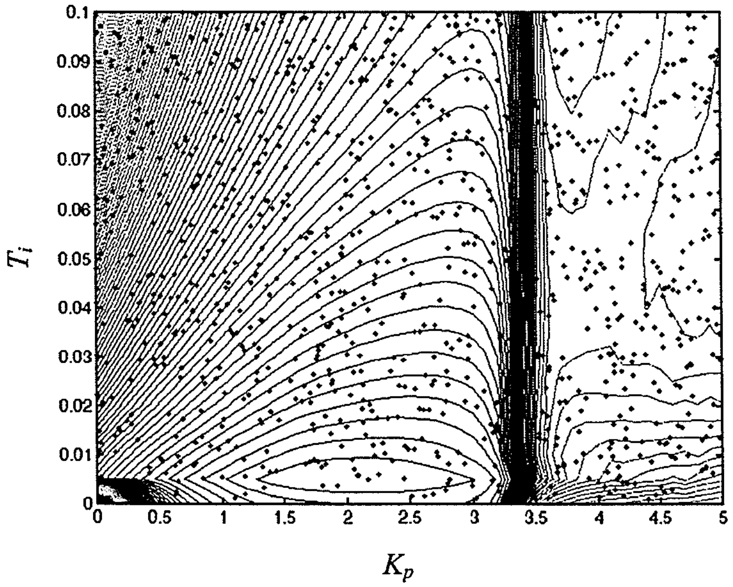


Fig. 2.5 Points generated by the random search method

It is seen that the design, even for a two-variable case, can require extensive simulations. One other point is that the success of the design using these methods depends on the optimum being in the search area. In cases where no such a-priori knowledge is available, an overly large search area should be selected to ensure the optimum is encapsulated. This will, in its turn, exacerbate the intensity of the simulation burden.

This example is later reconsidered in chapter 5, and an alternative design procedure will be presented, which not only improves the speed of the design but also generates results of much higher accuracy.

Table 2.1 Summary of design for the dc-dc converter

Design using MR				
Search interval for K_p [start : increment : end]	Search interval for T_i [start : increment : end]	Optimal parameter set	Number of simulations	*ISE* (optimum)
[0.0 : 0.1 : 5.0]	[0.001 : 0.005 : 0.1]	[1.9,0.006]	1071	0.00177

Design using random search					
Search interval for K_p	Search interval for T_i	PDF	Optimal parameter set	Number of simulations	*ISE* (optimum)
[0.0,5.0], 50 samples	[0.001,0.1], 20 samples	Uniform	[2.09,0.0058]	1000	0.00177

2.6 Statement of Design Specifications as an Optimization Problem

As mentioned earlier, a given design problem involves selection of the settings of a number of parameters in such a way that the design objectives are met as closely as possible. The conventional methods of design using transient simulation programs, namely the method of MR and the random search, essentially seek to solve this problem by searching for a parameter set, for which the performance of the system satisfies the design specifications.

Further examination of the setup of a design problem reveals that the process of design is in fact very similar to that of an optimization problem. Generally speaking a design problem can be stated as follows.

Find $\mathbf{x} = (x_1, x_2, \cdots, x_N)$
such that
design specification 1, design specification 2,$\cdots$, design specification M,
are met, (2.3)
subject to
constraints

We note that, design specifications are expressed in terms of measurable quantities of the system under consideration. For example, a design may specify the maximum overvoltage on a busbar to be less 5%, or the steady state error of a closed loop system to be less than 2%, etc. A well-performing set of design parameters is identified by how close the corresponding actual performance of the system will be to the design objectives. Therefore, one may state the design problem in the following equivalent form.

Find $\mathbf{x} = (x_1, x_2, \cdots, x_N)$
such that
$\|\text{actual performance of the system - design specifications}\|$
is minimized, (2.4)
subject to
constraints

where $\| \ \|$ is a suitably defined norm. The above statement of the design problem is apparently a classic optimization problem, in which the actual performance of the system is a function of the design parameter set $\mathbf{x}$, and its solution yields the optimal $\mathbf{x}$ resulting in the closest match between the actual and specified performances.

We note that in the design using transient simulation programs, the performance of the system for a given parameter set is evaluated using transient simulation of the network under consideration. Using the formulation in (2.4), it is apparent that should an optimization problem be properly linked to the transient simulation program, the search for the optimal parameter set can be significantly enhanced. This is because the theory of nonlinear optimization provides an abundance of elegant methods that can be used to optimize complicated objective functions.

The above potential is the main incentive behind this research. The following chapter provides details of the methods deployed to link optimization algorithms with PSCAD/EMTDC.

Interfacing between Electromagnetic Transient Simulation and Optimization Algorithms

Chapter 3

3.1 Introduction and Chapter Overview

This chapter discusses the methods of coupling optimization algorithms with transient simulation programs. As mentioned in the previous chapter, design of power systems using transient simulation of the network can be enhanced by formulating the design objectives as an optimization problem. An optimization algorithm can be used to strategically select the parameter sets to be tested, while the actual evaluation of the performance of the system for a selected parameter set is carried out through transient simulation of the network.

The above constitutes the basic interface between transient simulation and optimization; however, the mechanics of how to invoke transient simulation within an optimization algorithm can be designed according to the type of the problem to be handled. Two different types of such mechanisms are addressed in this thesis.

One other important issue in all optimization problems is the design of objective functions, which provide a quantitative approach to the evaluation of the norm of the difference between the actual and specified performances (see (2.4)). A meaningful, well-designed objective function can contribute to the well-posed-ness of the optimization problem under consideration and can increase the chance of finding a solution.

This chapter reviews some of the methods used in this research for designing objective functions. It is however important to note that an objective function should be custom-designed for an optimization problem to include its specific features and objectives, and as such, the methods presented in this chapter can only serve as guidelines that can be useful in designing other objective functions as well.

3.2 Basic Interface between Transient Simulation and Optimization - An Alternative Approach to Multiple-Run [22,23]

The basic interface between transient simulation and optimization is intended to alleviate the two major drawbacks, namely the blind search approach and the poor accuracy of the results, of the conventional method of multiple-run (both in its deterministic and random forms).

Fig. 3.1 shows the basic interface between transient simulation and optimization. As shown, the evaluation of the objective function is done through transient simulation of the network, while nonlinear optimization steers the selection of the candidate points to be tested.

It is stressed that the proposed interface is general in the sense that it can actually be used to link transient simulation to virtually any optimization algorithms, as long as the objective function derivatives are not required (non-gradient-based optimization algorithms). We defer the study of some such algorithms to the next chapter. Gradient-based optimization methods require a modified interface, which will be introduced in chapter 7.

As shown in Fig. 3.1, the simulation runs are successively carried out until the optimization convergence criteria are met. Corresponding to every point (parameter set) **x**, a simulation run is conducted and at the end of each simulation, an objective function evaluation (*OF*(**x**)) is assigned to the point. The objective function is set up so that smaller values indicate a better fit of the objective (recall (2.4)); hence the algorithm aims to minimize the OF. The process continues until successive evaluations of the objective function are essentially unchanged, indicating convergence to at least a local minimum.

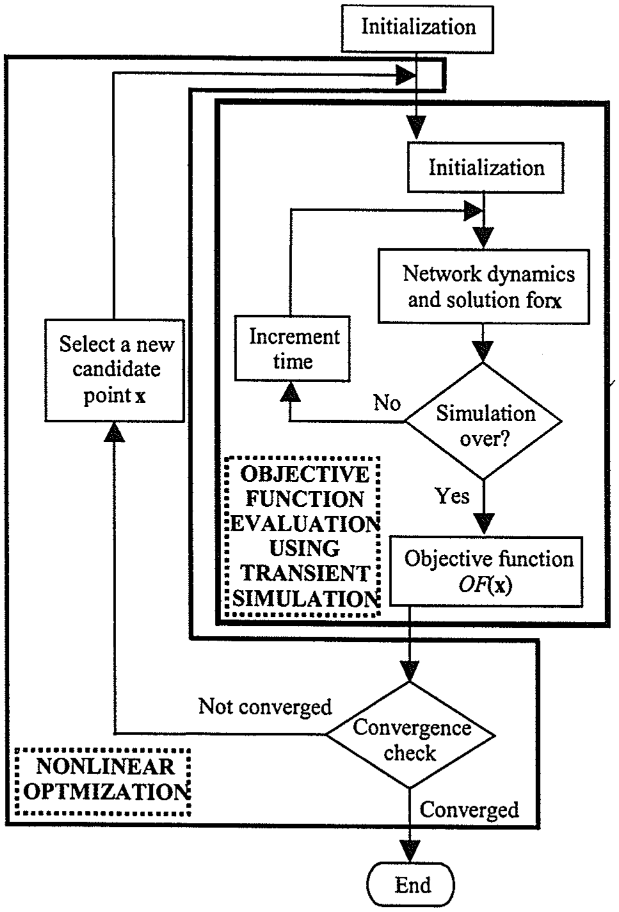


Fig. 3.1 Basic interface between optimization and transient simulation

The choice of optimization algorithms to be used is partially dependent on the problem and partially on the simulation techniques and facilities. A number of algorithms that have been used in this research are reviewed in chapter 4.

By enhancing the optimum-seeking method, this interface serves as a powerful tool in the design process of complex power systems. A number of sample case studies using this tool are given in chapter 5.

Some of the areas where the new design tool can be used are as follows.

- Determination of the optimum setting of control system parameters;
- Determination of the optimum values of power system components;
- Optimum operating conditions of power electronic converters to minimize the harmonic content of generated waveforms, etc.

3.3 Inclusion of Design Objectives into Optimization Objective Functions

Previously in chapter 2, we formulated the design problem as an optimization problem, in which the norm of the difference between the actual performance of a system with the objectives specified is to be minimized to yield the optimal parameter set. The interface developed in this chapter serves this purpose. The question however remains as how to effectively incorporate the design objectives into the optimization objective function, and also what kind of norm should be used.

3.3.1 Design of Objective Functions

Objective functions are central to optimization problems. The mathematical OF must be carefully selected so that it is an accurate measure of the conformity of the response to the requirements specified by the user. In a control system design for example, it may be required that the controlled variable closely matches the specified order in the steady state, and also has a good transient response. Fig. 3.2 shows a typical system response for the controlled variable $x(t)$. Design objectives require that the response should reach the ordered value within the rise time T_r, have a peak overshoot less than M_p and settle within a steady state error tolerance E_s close to the desired steady-state set point for $t > T_s$. In graphical terms, this means that the response should lie within the white (unshaded) regions.

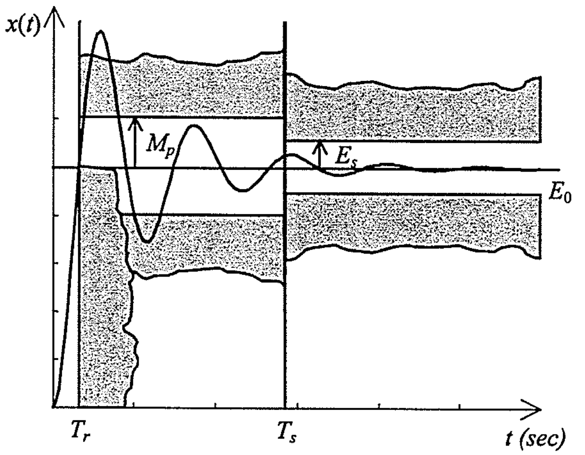


Fig. 3.2 Design of objective functions to meet specific performance measures

A single mathematical OF that attempts to embody all the above requirements can be generated based on the well-known Integral Square Error (ISE) measure popular in control systems. The OF selected in (3.1) has three components, one for each of the regions $[0, T_r)$, $[T_r, T_s]$ and (T_s,∞). In any of these regions, the closer the response $x(t)$ is to the desired, the smaller the value of the OF.

$$OF = ISE = K_1 \int_0^{Tr} (x(t) - E_0)^2 dt + K_2 \int_{Tr}^{Ts} g(x(t), E_0, Mp) dt + K_3 \int_{Ts}^{\infty} g(x(t), E_0, Es) dt \tag{3.1}$$

where

$$g(x,E_1,E_2)=\begin{cases}(x-E_1)^2 & \text{if } |x-E_1|\geq E_2\\ 0 & \text{otherwise}\end{cases} \tag{3.2}$$

Multi-objective optimization problems often converge to a solution that is a compromise between the objectives. For example in a control system design problem as stated above, the requirement for a fast rise-time may conflict with the need for a low overshoot. The factors K_1, K_2 and K_3 can be used to give selective importance to the three different regions, depending on which objective is more important to the designer.

Selecting parameters in the simulation that minimize ISE thus provide the best fit of the desired objectives. Note that the selected OF is implicitly a function of the design variables, although it may be very difficult or even impossible to explicitly show the relationship. This captures the true power of the simulation-based optimization, where it is possible to optimize a design without an explicit formula for the relationship between the objective function and the design parameters.

For example, if a power converter, whose response is to be optimized, has a proportional-integral (PI) controller with proportional gain and integral time constant K_p and K_i, respectively (such as the example presented in the previous chapter), the corresponding optimization problem can be stated as follows.

$$\begin{aligned}&\text{Minimize } OF(K_p,T_i)\\ &\text{subject to } K_p>0, T_i>0\end{aligned} \tag{3.3}$$

3.3.2 Weighting Components of Combined Objective Functions

It is sometimes necessary to include various measures into a single objective function. This may happen for instance when the monetary cost of a given system as well as its dynamic performance is included in the objective function. Such sub-measures take on values that can be significantly different from one another. Should one of the sub-measures have numerical values that are considerably larger than those of the others, it will force the optimization to focus only on this sub-measure. In a well-defined objective function, different measures should have comparable numerical values and this can be achieved by properly weighting individual measures before they are combined into the OF. Such an objective function will therefore have the following form.

$$OF(\mathbf{x}) = \sum_i W_i \cdot M_i(\mathbf{x}) \tag{3.4}$$

where **x** and $OF(\mathbf{x})$ are the candidate point (parameter set) and its corresponding overall objective function evaluation and W_i is the weighting given to the i-th performance measure $M_i(\mathbf{x})$. Proper selection of weighting factors can result in comparable contribution of various measures to the overall objective function.

More details on this vital subject will be provided when the simulation results are presented in chapter 5.

3.3.3 Scaling and Constrained Optimization

Optimization parameters often need to be constrained to meet physical or design limitations (see (2.3) or (2.4)). For example, in a control system design problem, time constants of integrators need to be nonzero or gains may need to be only positive. As mentioned earlier in chapter 2, the method of MR and also the random search methods conventionally used in design using transient simulation programs, inherently address the issues of scaling and constraints. However, several optimization algorithms are originally unconstrained methods, and therefore their application in constrained optimization problems needs some external considerations.

One way to impose constraints is to externally limit the coordinates of the candidate point to their specified limits. This can be done by hard limiting or by introducing auxiliary variables. For example if a certain optimization variable x needs to be limited between $[a,b]$, one may define an auxiliary variable y where $x = (b+a)/2 + (b-a)/2 \sin(y)$. The optimization algorithm will generate the unconstrained variable y; however, the actual design variable x that is obtained from the above equation is automatically constrained to be in the interval $[a,b]$. Other methods such as use of absolute values or squaring the variable can also be used to ensure positive-ness of the variables.

Another issue of importance in optimization problems is proper scaling of variables. It is generally unfavorable if numerical values of the optimization variables are considerably different from one another. For example in a case where the value of a resistor and a capacitor are to be determined, the values may be orders of magnitude different (10^3 (ohms) and 10^{-6} (farads) for example). To avoid numerical complexities, optimization variables within the algorithm should be properly scaled ('per-unitized') to have comparable magnitudes. Subsequently, they can externally be scaled back to their original ranges before being submitted for objective function evaluation.

In the next chapter, we discuss some non-gradient-based algorithms that have been used in conjunction with PSCAD/EMTDC in the proposed design tool. Advanced design issues, such as inclusion of robustness, as well as interfacing gradient-based methods are presented in chapters 6 and 7, respectively.

Chapter 4

Optimization Algorithms

This chapter presents four representative optimization algorithms that have been used in this thesis for interfacing with PSCAD/EMTDC. The methods discussed are only typical examples of non-gradient-based optimization algorithms and are presented to provide an insight into the behavior of optimization algorithms. They also provide a flavor of how different optimization algorithms select new parameter sets during their search process.

It should be noted that the interfacing method described in chapter 3 is not limited to the following algorithms, and can actually be used to interface any non-gradient-based optimization method. Gradient-based methods are presented in chapter 7.

4.1 An Overview of Optimization Algorithms

An optimization algorithm is basically a specialized search algorithm that carries out two major tasks, namely generation of new trial points (parameter sets), and processing of optimization objective functions (OF) [9]. An objective function is a quantitative measure of the performance of a trial point in meeting the objectives set by the user. In a minimization problem for instance, parameter sets that have better fits to the specified objectives, will have lower objective function evaluations; the ultimate goal of a minimization problem will therefore be to find a parameter set that has the lowest (at least locally) OF evaluation.

The differentiating feature between various optimization methods is the manner in which they select new trial points based on the current point(s). Generally speaking, optimization algorithms can be classified into three large categories: analytical, geometric (heuristic) and random-based. In analytical algorithms, such as the gradient-based methods, an analytical formulation for direction of the steepest descent, establishes the basis on which current point(s) are manipulated to generate new candidates [9]. It is often possible to analytically investigate the convergence of such algorithms. Geometric methods, on the other hand, use intuitive geometric considerations as their search basis. Test of convergence in these algorithms is more difficult and is often limited to special cases. It should be noted that although these algorithms are geometry-based, they often imitate the same gradient-based search scheme as in analytical algorithms. In the last class of optimization algorithms, the random-based methods, search for the optimum involves some random movements that are somehow supervised by the algorithm [12,13]. The main feature of these algorithms is their higher chance of converging to the global optima, which occurs at a much higher simulation burden.

Fig. 4.1 shows the schematic diagram of a generic optimization algorithm. The variable $\mathbf{x}$ denotes the current trial point (parameter set). In general, $\mathbf{x}$ is a vector of N elements (in an N-variable optimization problem), each of which is an optimization variable. The goal of the algorithm is to find the point $\mathbf{x}_{opt}$ whose elements represent the best values for the optimization variables.

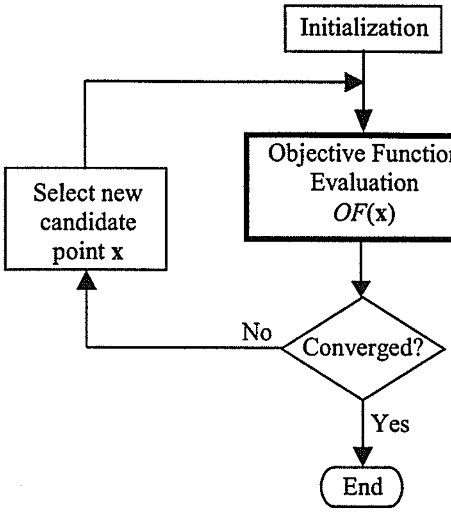

Initialization
Objective Function Evaluation $OF(\mathbf{x})$
Select new candidate point x
No
Converged?
Yes
End

Fig. 4.1 Schematic diagram of a generic optimization algorithm

The block for objective function evaluation returns a figure of merit for the performance $OF(\mathbf{x})$ of the current trial point. It is important to note that OF evaluation is not necessarily a task to be done by the optimization algorithm itself. Depending on the type of the problem under consideration, calculation of the $OF(\mathbf{x})$ can be as simple as evaluating an explicit mathematical function for $\mathbf{x}$, or may on the other hand need extensive laboratory experiments. In our case, it is done through transient simulation of a power network using PSCAD/EMTDC.

Before proceeding any further, another underlying difference between various optimization algorithms should be mentioned, and that is the type of information they require in order to generate new candidate points. While various gradient-based methods require first and higher order derivatives of the objective function, a large category of optimization algorithms merely rely on function evaluations in the process of selecting new parameter sets.

Most of the time in complicated optimization problems, such as ones encountered in the design of power systems, an explicit mathematical expression of the objective function in terms of design parameters is extremely hard or even impossible to obtain. In such cases, lack of the ability to calculate the derivatives, limits the application of gradient-based methods and the natural choice is to use the latter class of optimization algorithms, which only require function evaluations.

Based on the above observation, the initial focus in this thesis is on non-gradient-based class of algorithms. In the following sections, four nonlinear optimization algorithms of this kind are considered for interfacing with PSCAD/EMTDC. Each of these algorithms has a certain domain of application, which is referred to in the following.

A single variable optimization method is first presented followed by two nonlinear multi-variable methods, and finally a mixed-integer optimization method (Genetic Algorithms) concludes the chapter.

4.2 A Single-Variable Optimization Method - Golden Section Method

Single-variable optimization techniques comprise a fundamental part of optimization algorithms. This is because there are numerous applications in which single-variable optimization problems arise. These problems can be more efficiently solved using techniques specifically tailored for single-variable cases. Besides, there are several multi-variable optimization algorithms inherently based on single-variable methods. In these methods, single-variable routines are successively carried out for each of the variables.

An important assumption that many single-variable optimization techniques are based on is the unimodality of the objective function concerned. Consider an interval $[a,b]$ and suppose that the objective function $f(x)$ has an optimum x^* in this interval. The function is called unimodal if it is monotonic on either side of x^*. Obviously the function should be monotonically increasing on one side and monotonically decreasing on the other side. Fig. 4.2 shows a monotonic function with a minimum in the interval [-0.5,1.5]. It is seen that the function is monotonically decreasing on the interval $[-0.5,x^*]$ and monotonically increasing on the interval $[x^*,1.5]$.

Unimodality is an essential assumption in region-elimination methods such as Golden Section, Interval Halving, Fibonacci, etc. Suppose that function $f(x)$ is unimodal over $[a,b]$ with a minimum x^* occurring in the interval, and x_1 and x_2 are points within the interval such that we have $a < x_1 < x_2 < b$. Comparing function evaluations at x_1 and x_2, we may infer:

If $f(x_1) > f(x_2)$, then x^* does not lie in the interval $[a, x_1]$; this region can therefore be eliminated.

If $f(x_2) > f(x_1)$, then x^* does not lie in the interval $[x_2,b]$; this region can therefore be eliminated.

Region elimination can be continued with a new pair of x_1 and x_2 until the search interval is sufficiently small. The main question however is how to choose the intermediate points. Golden Section is a method for choosing intermediate points x_1 and x_2. It can be proved that it is the most efficient region elimination method [9].

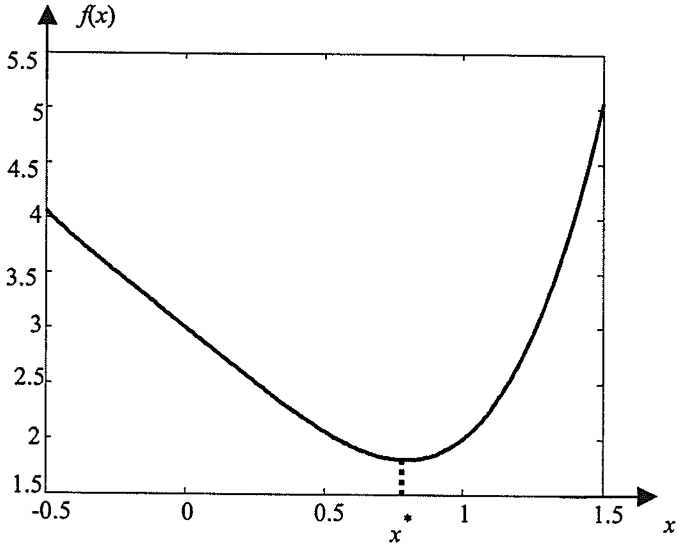
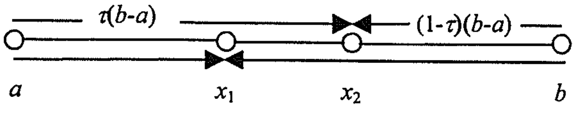


Fig. 4.2 Unimodality

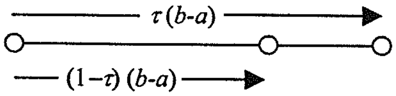
To apply the Golden Section method, one should first find a bracketing interval around the minimum. Consider a minimum bracketing interval [a,b] as shown in Fig. 4.3 (a). Also consider two intermediate points x_1 and x_2 that are located at distances $\tau(b-a)$ and $(1-\tau)(b-a)$ from a, respectively. Note that there is symmetry in the way the intermediate points are positioned between the two end points, i.e. $\overline{ax_1}/\overline{ab} = \overline{bx_2}/\overline{ab}$. The idea is now to select τ so that the symmetry is retained following the elimination of a subinterval, e.g. [x_2, b] as shown in Fig. 4.3 (b).

For that reason, τ should satisfy the equation $1-\tau = \tau \cdot \tau$; therefore we will have $\tau = \dfrac{-1 \pm \sqrt{5}}{2}$. The feasible solution $\tau = 0.618$ is used and it is known as the Golden ratio. Following the deletion of a sub-interval, only one new point is inserted with the appropriate distance and the new configuration of the points will still retain its Golden ratio. Successive operation of interval elimination and intermediate point insertion will ultimately result in a minimum bracketing interval that is smaller than a pre-specified value and the search can be stopped. More details about bracketing methods and interval elimination techniques can be found in [9].

Each of the methods discussed above obviously has its own strengths and drawbacks; appropriate use of any of them therefore requires careful examination of their features and potentials along with those of the optimization problem under consideration. The discussion in this chapter covered only those aspects that are deemed necessary to provide a basic understanding of the methods interfaced with PSCAD/EMTDC.



(a) Original placement of points



(b) Preservation of symmetry

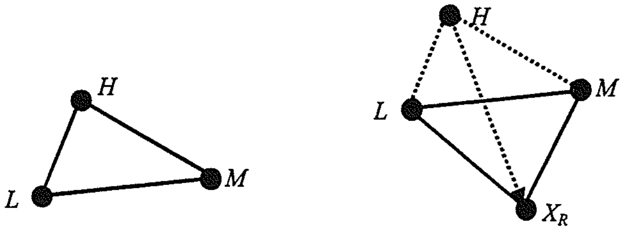
Fig. 4.3 Golden section method

4.3 Simplex Optimization Method of Nelder and Mead [9,14]

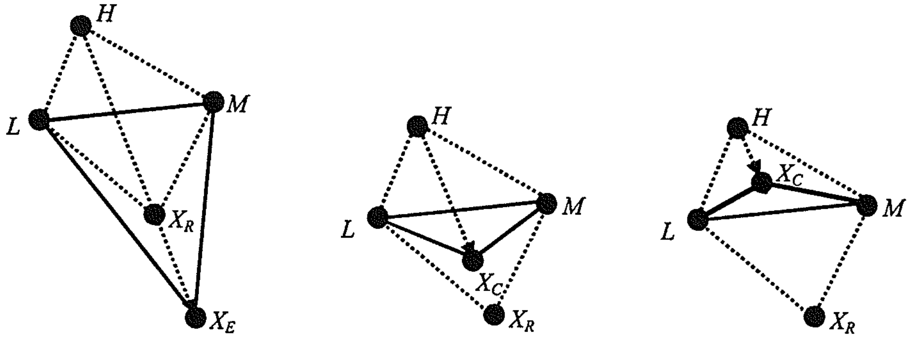
Simplex optimization method is a geometry-based, non-linear optimization algorithm, and is therefore classified as a heuristic method. This optimization method is usually very suitable for cases where the number of variables is limited to twenty or less. This method is based on successive intuitive re-shaping of an object called simplex. In N-dimensional space, a simplex is a geometric object comprising $N+1$ vertices. Should the distance between any two vertices be equal, the simplex is called a regular simplex. In two- and three-dimensional spaces, the simplex will be a triangle and a tetrahedron, respectively. To understand the underlying idea of the Simplex optimization method, consider a triangle (simplex) in a two-dimensional space, as shown in Fig. 4.4 (a). The nodes are labeled high (*H*) to low (*L*) according to their corresponding objective function evaluation. The idea is to move away from the high point. To do so, the high point (*H*) is reflected through the centroid of the face of simplex right across from it to generate a (hopefully) better point (X_R). This operation is called reflection (Fig. 4.4 (b)). Should the newly generated point be better than the current low point (*L*) (i.e. with a lower objective function evaluation in a minimization problem), an even larger step is taken in the same direction. This operation is called expansion (Fig. 4.4 (c)). If the reflected point is worse than the second highest point (*M*), a contraction occurs (Fig. 4.4 (d)). One other type of contraction is carried out when the reflection fails to generate a better point than the current high point (*H*). In that case, a contraction without reflection is performed, as shown in Fig. 4.4 (e). Successive operations of reflection, expansion and contraction will cause the original simplex to move towards points with better objective function evaluations. More details about this optimization method can be found in [9,14].

It is important to note that in this method the data pertinent to a simplex should always be available to the optimization algorithm, i.e. *N*+1 point (vertex) coordinates plus their corresponding objective function evaluations.

The most attractive feature of Simplex method is the simplicity of its underlying idea, which has made the method easy, yet powerful and reliable, to apply to optimization problems. However, it should be mentioned that Simplex method is not guaranteed to find the global optima, and depending on the initial simplex it might move towards and converge to local optima. It has been suggested that in such cases it may be a good idea to restart the algorithm with a new side length when it reaches an optimum. It causes the method to search for other optima as well.



(a) Initial simplex (b) Reflection



(c) Expansion ($X_R < L$) (d) Contraction ($X_R > M$) (e) Contraction ($X_R > H$)

Fig. 4.4 Fundamental operations on a simplex

Since Simplex is a heuristic method based on geometric interpretations, it is interesting to note how the geometry of a simplex evolves in a simple optimization problem. Consider a two-variable function as defined below:

$$f(x_1,x_2) = (x_1 - 1)^2 + (x_2 - 1)^2 + (\sin x_1)^2 x_2^2 \tag{4.1}$$

The function has a minimum of 0.38 at (0.79,0.69), as indicated by the low-density area of the associated contour map in Fig. 4.5. The simplex in this case is a triangle as the space has two dimensions. The initial simplex vertices have OF values of 9.3, 12.4 and 19.8, respectively.

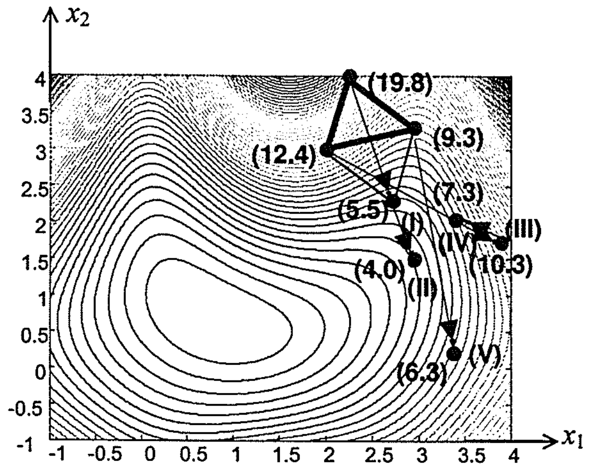


Fig. 4.5 Simplex evolution

In the first step, the highest vertex (19.8) is discarded and replaced with the reflected vertex (I) with the OF value 5.5. This vertex is even smaller (in OF value) than the smallest of the original simplex and thus indicates a favorable direction for movement, causing an expansion to a new vertex (II) with value 4.0. The procedure is continued, and generates successive vertices III, IV, V…and so on. Sometimes the reflected vertex has to be discarded and a contraction called for (i.e., vertex III is dropped in favor of IV as its function value was higher than at any of the vertices of the reflection centroid). The process continues until the function evaluations at the vertices differ by an amount smaller than the convergence criterion.

Since the Simplex optimization method is merely based on OF evaluations and does not require first or higher order derivatives, it is quite suitable for problems where it is not possible to explicitly evaluate derivatives, or when the objective function is not smooth and differentiable. Since in power system optimization problems, no a-priori knowledge of the nature of the optimization surface is usually available, derivative-independent method such as Simplex are very favorable.

4.4 Hooke-Jeeves Optimization Method

The Hooke-Jeeves (HJ) method is a nonlinear, multi-variable optimization method that, similar to the method of Simplex, is a heuristic method. Due to this fact, it comes in a variety of forms; however, the variations are essentially based on successive single variable searches along a set of directions. It is interesting to recall that even in Simplex method, the underlying idea is to find a path towards an optimum, and the re-shaping of the simplex is in fact a methodology to do so.

The HJ search method is comprised of two distinct parts, namely the Exploratory Search and the Pattern Move. In the former, search is started around a pattern point along some directions. The directions should be chosen such that the whole optimization space can be spanned. Therefore, it is reasonable for example to choose coordinate directions as the search directions. The directions are searched one at a time to find points with better objective function evaluations. Should the search in a certain direction fail, it is replaced with a search in the opposite direction. Once all the directions are searched, the exploratory search about the current pattern point is over. The resulting point is called a base point.

Pattern Move is the process of moving the current pattern point to a new one along the line from the current pattern point to the current base point. This is essentially a move in the direction that decreases the objective function evaluation (considering a minimization problem). The new pattern point is obtained from the following formula [9].

$$\mathbf{x}_p^{(k+1)} = \mathbf{x}^{(k)} + (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}) \tag{4.2}$$

where $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k\text{-}1)}$ are current and previous base points, respectively. The new pattern point is accepted only if it results in a better function evaluation that the current base point; otherwise the search is returned to the current base point, but with a smaller step size. The search continues until the step size becomes smaller than a pre-specified value.

The whole procedure is better illustrated with an example. Consider the two-variable function of (4.1). The problem is to find its minimum starting from the initial point $\mathbf{x}^{(0)}$ = [4,3]. In order to use the HJ method, the following parameters are assumed:

Initial step length, $\Delta = 1.0$

Step reduction factor, $\alpha = 2.0$

The exploratory search is started around the point $\mathbf{x}^{(0)}$, which is currently both the base and the pattern point. We first notice that $f(\mathbf{x}^{(0)}) = 18.15$. The search should therefore look for points with lower function evaluations. The result of the search is summarized in Table 4.1.

Table 4.1 Exploratory search around $\mathbf{x}^{(0)}$

Test Point (x)	[4,4]	[4,2]	[5,3]	[3,3]
f(x)	27.16	12.29	28.28	8.18

A better point ([3,3]) is found. The exploratory search is therefore successful. We will now have $\mathbf{x}^{(1)}$ = [3,3], $\mathbf{x}^{(0)}$ = [4,3]. The new temporary pattern point will therefore be $\mathbf{x}_p^{(1)} = \mathbf{x}^{(1)} + (\mathbf{x}^{(1)} - \mathbf{x}^{(0)})$ = [2,3]. The exploratory search is now performed around $\mathbf{x}_p^{(1)}$. The results are given in Table 4.2.

Table 4.2 Exploratory search around $\mathbf{x}_p^{(1)}$

Test Point (x)	[2,4]	[2,2]	[3,3]	[1,3]
f(x)	23.23	5.31	8.18	10.37

Since the search around $\mathbf{x}_p^{(1)}$ has resulted in a better point than $\mathbf{x}^{(1)}$, we will have $\mathbf{x}^{(2)}$ = [2,2] and a new pattern move should take place as $\mathbf{x}_p^{(2)} = \mathbf{x}^{(2)} + (\mathbf{x}^{(2)} - \mathbf{x}^{(1)})$ = [1,1]. The search should now be carried out around $\mathbf{x}_p^{(2)}$. If the search around a pattern point fails to result in a base point with lower objective function evaluation than the current base point, the exploratory search is deemed unsuccessful. The search is then re-started with the current base point and a new step size equal to the current step size divided by the step reduction factor, i.e. $\Delta^{new} = \Delta^{old} / \alpha$.

Fig. 4.6 shows the contour map of the function $f(x_1,x_2)$ and the first few points obtained by HJ method.

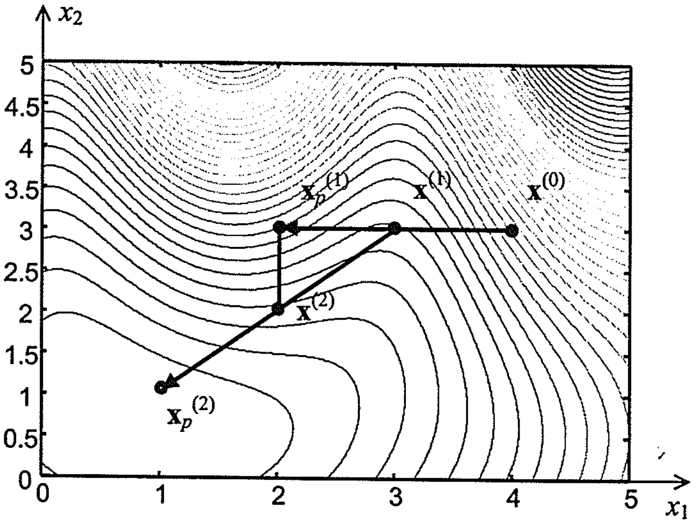


Fig. 4.6 Exploratory search and pattern move in HJ

It should be noted that HJ is essentially an optimization method suitable for finding local optima and does not necessarily converge to global optima. It is also interesting to note that it generally goes towards the optima fairly fast, but it may not converge to the exact optima very rapidly. There are a number of variations to HJ method that each focus on some aspects of the algorithm to improve its performance; however, they are beyond the scope of this research and will therefore be avoided.

4.5 Genetic Algorithms [29,31]

Genetic Algorithms (GA) are a fairly new class of optimization methods. GAs along with some other optimization methods such as Particle Swarm, Simulated Annealing, etc. are considered to be parts of what is called Evolutionary Computing [13]. The main areas of the application of GAs in power system design problems are cases that involve mixed-integer optimization, i.e. cases where some of the variables take on real values and others take on integer or on/off values. The coding procedure used in GAs (see below) can easily address such cases.

Similar to Simplex method, GAs are also based on intuitive interpretations rather than formal mathematics. The underlying idea of GAs is to approximate the natural evolution and selection of biological organisms. It is believed by some that biological organs are evolving towards generations of higher fitness, intelligence, physical capability, etc. It is done through the generation of new genes from the old ones that hopefully carry traits of higher fitness [12,13,15]. GAs are in fact attempting to mathematically capture the complicated process of evolution in nature.

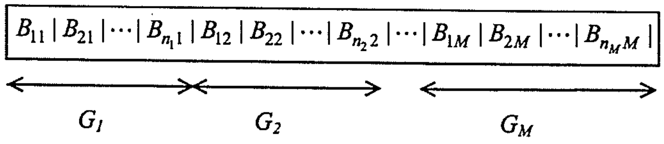
Generally, the process of optimization with GAs comprises the following steps:

- Coding the optimization variable into genes and chromosomes,
- Generation of the initial population,
- Objective function (fitness) evaluation and natural selection,
- Pairing,
- Mating,
- Mutations,

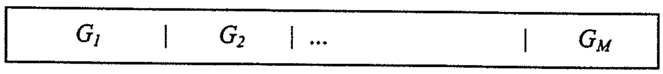
A brief overview of the above steps will be given here; however, more details can be found in [12].

4.5.1 Coding

The first step in solving problems using GAs is coding. By coding we mean an arrangement of optimization variables in an ordered way, which is usually referred to as a chromosome. The coding may be either binary or real, and the resulting GA is called binary-valued (binary-coded) or real-valued (real-coded), respectively. It is possible to solve real-valued optimization problems using a binary-coded GA; however, this requires representation of real variables with binary codes. The number of bits used to code each variable determines the quantization error. While it is possible to decrease the quantization error by using larger number of bits, the memory storage problems become more difficult as the chromosome length increases. Fig. 4.7 shows a schematic diagram of binary- and real-coded chromosomes. For mixed-integer optimization problems, which is the main intended use of GAs in this research, each chromosome consists of a number of binary-coded as well as some real-coded genes. In addition some single-bit variables, called switching (on/off) variables are also incorporated.


$B_{11} | B_{21} | \cdots | B_{n_1 1} | B_{12} | B_{22} | \cdots | B_{n_2 2} | \cdots | B_{1M} | B_{2M} | \cdots | B_{n_M M} |$
G_1
G_2
G_M

(a) Binary-coded chromosome



(b) Real-coded chromosome

Fig. 4.7 Coding

- *An Example of Coding*

To provide a better understanding of the coding process, a real-valued optimization problem with binary-coding is considered. Suppose one of the optimization parameters varies over the interval [0,1]. The optimal value of this parameter will therefore lie in this range. It is possible to code this parameter into a binary gene (for example G_1 in Fig. 4.7 (a)) using an appropriate number of bits. Suppose that three bits are used for coding of this parameter. The following table shows the quantization bands and their respective binary equivalents.

Table 4.3 Coding a real parameter

Real range	Binary representation
[0,1/8]	000
(1/8,2/8]	001
(2/8,3/8]	010
(3/8,4/8]	011
(4/8,5/8]	100
(5/8,6/8]	101
(6/8,7/8]	110
(7/8,1]	111

It is clear that each interval is 1/8 unit wide and the binary coding introduces a maximum error of 1/16 unit. It is possible to decrease the quantization error by using a larger number of bits, at the expense of larger memory storage and more intensive computations.

Successive genes G_2 and G_3, etc. are used for the remaining optimization parameters. Note that there may be integer as well as switching parameters in an optimization problem. Integer variables are coded using exactly the same procedure as outlined above except for the fact that the coding for these variables does not involve any approximations. Switching variables, which represent on/off states for example, are coded using single bits.

4.5.2 Initial Population and Natural Selection

The initial population comprises a fairly large number of chromosomes that are usually randomly initialized. The large number of chromosomes and their random initialization cause the GA to start the search at widely separated points in the search space. Therefore, the chance of finding global optima using GAs is potentially higher, although it is not guaranteed.

With a larger initial population, the likelihood of the optimization space being searched evenly increases. The population of the next generations is usually less than the initial population, i.e. from N_{ipop} initial chromosomes, a number N_{pop} are chosen to form the next generation. N_{pop} is usually kept constant for the rest of the process. It is therefore possible to discard a number of bad performers to prevent them from passing their traits to the next generations. Natural selection is the process of choosing a number of chromosomes of better performance to form the next generation. There are basically two ways to choose the winning chromosomes; one is to randomly choose N_{pop} ones from the initial population, and the other is to choose the top N_{pop} chromosomes when the initial population members are sorted in descending order according to their fitness. Although the random approach is very easy to implement, it obviously has no control on the passing of the bad performing chromosomes to the next generation. In this research, the sorting method has been adopted.

4.5.3 Pairing

In each of the generations following the initial one, N_{good} chromosomes are singled out to form the mating pool. Each couple of mating chromosomes usually produces two offspring. The total number of offspring should be equal to $N_{pop} - N_{good}$ to keep the population of the generation constant. There are several methods for selecting mating chromosomes. A number of these methods are addressed in the following.

- *Top to Bottom*

Once the chromosomes in the current population are sorted in descending order according to their fitness, the mating pairs can be chosen two by two from top to bottom; i.e. (1,2), (3,4), etc. This is a very simple approach to the mating process and is obviously very straightforward to implement.

- *Random*

In random pairing method, $N_{pop} - N_{good}$ random numbers between 1 and N_{good} are generated. Chromosomes corresponding to the numbers generated are mated two by two. So if the sequence of random numbers is $\{2,3,4,6,1,8\}$, the mating pairs will be (2,3), (4,6) and (1,8).

- *Rank Weighting*

Rank weighting and cost weighting methods, which will be introduced afterwards, are computationally more demanding than the methods already discussed.

Suppose that the sorted population is available. Each of the top N_{good} chromosomes is assigned a unique number based on its rank n in the list as,

$$P_n = \frac{N_{good} - n + 1}{N_{good}(N_{good} + 1)/2} \tag{4.3}$$

Cumulative probabilities for each chromosome are then calculated as,

$$(CP)_n = \sum_{i=1}^{n} P_n \tag{4.4}$$

Note that the cumulative probability of the top N_{good} chromosomes will be less than or equal to one. To select the mating pairs, N_{pop} - N_{good} random numbers between zero and one are generated. Each of the random numbers is then compared with the cumulative probabilities of the chromosomes from top to bottom of the list. The first chromosome with a cumulative probability greater than the random number under consideration will become a parent.

- *Cost Weighting*

In this approach, the normalized cost for each of the top N_{good} chromosomes is calculated as follows.

$$C_n = (Fitness)_n - (Fitness)_{N_{good}+1} \tag{4.5}$$

where n is the rank of the chromosome.

Individual probabilities are then calculated as,

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{good}} C_i} \right| \tag{4.6}$$

Cumulative probabilities for the chromosomes are calculated as (4.4). N_{pop} - N_{good} random numbers between zero and one are then generated and using the same process as in rank weighting, mating pairs are selected.

The weighting methods outlined above are usually referred to as roulette wheel weighting methods.

- *Tournament*

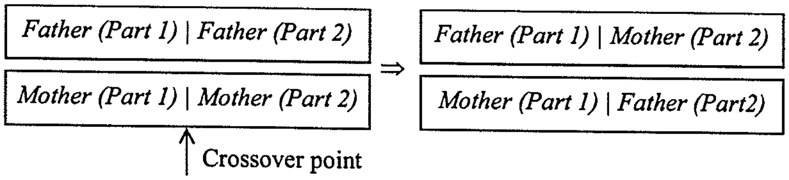
This approach is quite similar to the mating process in nature. In this approach, two random numbers between one and N_{good} are generated at a time. The chromosomes corresponding to these numbers are picked up, and their fitness is compared. The fitter chromosome will be a parent. The same process goes on until the required number of mating pairs are selected.

The methods outlined above obviously result in different sets of mating pairs and therefore the properties of the generations produced by them are not similar. Random nature of the GAs also contributes to this fact.

4.5.4 Mating

Mating is the process by which a number of offspring (usually two) are produced by a mating pair. The idea is to generate more chromosomes with (hopefully) better performance. The fundamental procedure used in mating is crossover. The simplest form of this operation is single-point crossover, which is described below. There are multi-point crossover alternatives as well. Crossover is slightly different for binary and real coded chromosomes. Both variations will be addressed here.

Consider two binary coded parent chromosomes as shown in Fig. 4.8 (a). A crossover point between the first and last bit is selected randomly. The two offspring are produced as shown in Fig. 4.8 (b).


Father (Part 1) | Father (Part 2)
Mother (Part 1) | Mother (Part 2)
Crossover point
Father (Part 1) | Mother (Part 2)
Mother (Part 1) | Father (Part2)

(a) Mating pairs (b) Offspring

Fig. 4.8 Binary crossover

Crossover in real coded chromosomes comes in many variations. The method adopted in this research is a fairly simple approach; however, other crossover methods can be easily incorporated.

Consider two real-coded mating chromosomes as follows:

$$\begin{aligned} Mother &= [m_1, m_2, \cdots, m_c, \cdots, m_N] \\ Father &= [f_1, f_2, \cdots, f_c, \cdots, f_N] \end{aligned} \tag{4.7}$$

where subscripts 'c' denotes the crossover point. A randomly generated constant β (from [0,1]) is also chosen. The offspring can be obtained as follows:

$$\begin{aligned} Child_1 &= [m_1, m_2, \cdots, m_{c-1}, x_{new1}, f_{c+1}, \cdots, f_N] \\ Child_2 &= [f_1, f_2, \cdots, f_{c-1}, x_{new2}, m_{c+1}, \cdots, m_N] \end{aligned} \tag{4.8}$$

where

$$\begin{aligned} x_{new1} &= m_c - \beta(m_c - f_c) \\ x_{new2} &= f_c + \beta(m_c - f_c) \end{aligned} \tag{4.9}$$

As mentioned earlier, there are other crossover methods that employ other blending strategies. An overview of these methods can be found in [12].

4.5.5 Mutations

Using crossover, it is possible to produce new chromosomes that have (hopefully) inherited good traits from their parents. The offspring carry traits that are more or less similar to those of their parents. From an optimization point of view, crossover is similar to searching around local optima when it is found. Therefore, although the spread of a GA's initial population is a promising feature in search for global optima, GA's relying on merely crossover operations can easily cause it to get stuck in local optima.

Mutations are the remedies proposed to overcome this problem. During mutations, a parameter in a chromosome is replaced with a totally random value of the same size. A user-specified mutation probability or rate determines the number of chromosomes taking part in mutation. It should be noted that choosing a suitable value for mutation probability is an essential factor in the performance of GAs. Should the rate be low, very few chromosomes are mutated and the chance of finding global optima decreases, while with high mutation rates, the possibility of losing chromosomes with high fitness increases and the good traits already accumulated in the population can be deteriorated.

The architecture of a GA approach to problem solving is very flexible and there are different variations to implement them. As mentioned, there are several methods proposed for each of the basic operations outlined above. Besides, the structural parameters of the method, such as the initial population, the surviving population, the number of mating pairs, the mutation rate, etc. also affect the performance of the algorithm. Further details can be found in [12,15].

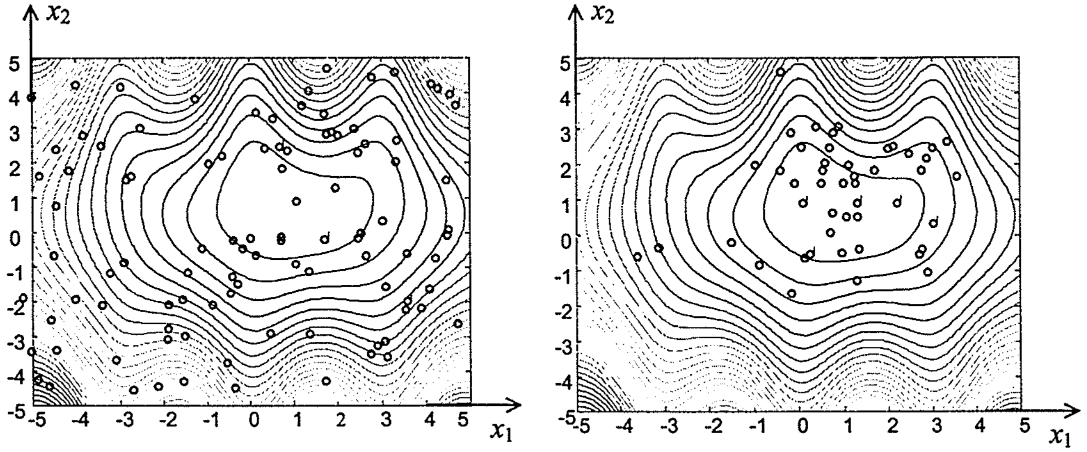
4.5.6 Optimization Using Genetic Algorithms – An Example

In this section we use GA to solve a two-variable optimization problem similar to the one we discussed in section 4.3. Although solution with Simplex and other optimization methods is quite straightforward to obtain, and there is little serious incentive to use GAs, this example serves to demonstrate the principles of the GA in a comprehensible way.

The coding for the problem involves using two genes in each chromosome, where each gene is a real number representing one of the optimization parameters (see Fig. 4.7). A chromosome contains a pair of such genes.

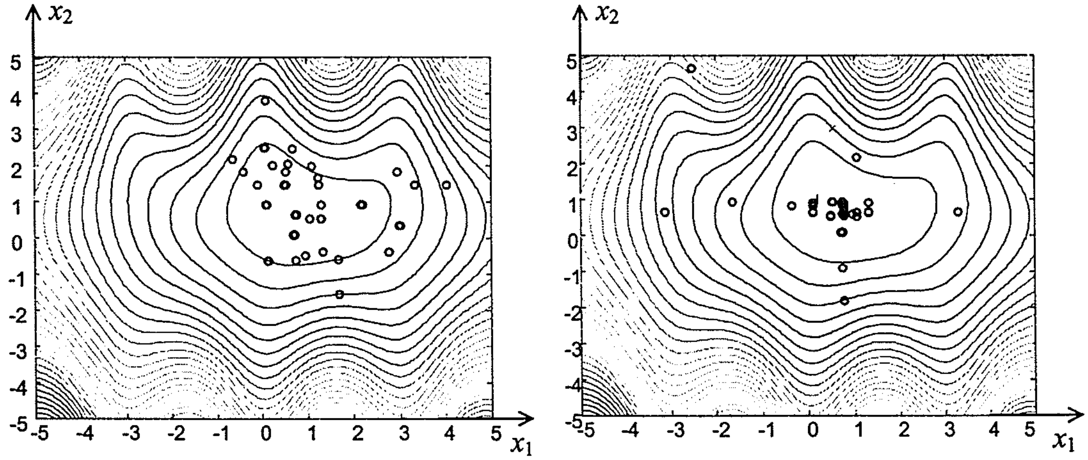
The initial population, number of surviving population and the number of mating pool members are $N_{ipop} = 100$, $N_{pop} = 50$ and $N_{good} = 30$, respectively. In other words, the search starts with randomly generating 100 instances of the 2-gene chromosomes described above; following the evaluation of the fitness of these 100 initial candidates, the top 50 ones are selected to form the next generation. The top 30 chromosomes are maintained and form the mating pool.

Optimization variables, x_1 and x_2 are varied over the range (-5,5). Cost weighting method is used to select the mating pool members and the mutation rate is chosen to be 10%. Fig. 4.9 shows the distribution of the points over the x_1-x_2 plane for the initial population as well as the first two successive generations. Distribution for the sixth generation is also shown.



(a) Initial population

(b) Second generation



(c) Third generation

(d) Sixth generation

Fig. 4.9 Distribution of points in various generations

It is seen that the initial population is randomly scattered all over the search space. The second generation consists of a smaller number of points, which are closer to the minimum. Subsequent generations contain the same number of points, however the points are mostly concentrated around the minimum. In each generation, there are points that are farther from the minimum. These points are mostly generated through the mutations and are used to ensure the trend of evolutions is not confined to a local minimum.

It is observed that the GA has managed to move its search region to an area quite close to the minimum. Since the minimum is not known before hand, the procedure should be terminated when most of the candidate points are within a sufficiently small area.

4.6 Gradient-Based Optimization Methods

Earlier in this chapter it was mentioned that the original focus in this research was on the application of non-gradient-based optimization algorithms. This was mainly because of the fact the design problems in power systems often involve cases where explicit representation of optimization objective functions in terms of design variables is not readily available. And therefore, optimization algorithms that do not require derivatives (of any order) are generally preferable. The optimization methods described above all fall in this category.

Later on, experiments were carried out with numerical implementation of gradient-based algorithms, for cases with and without explicitly represented objective functions. It was found that under certain circumstances, derivatives can be numerically approximated, and that the gradient-based methods even with approximate derivatives can be successfully implemented.

The results of these studies are separately reported in chapter 7, which addresses the numerical calculation of derivatives, general structure of gradient-based methods, algorithms used for the adjustment of step lengths, and also a number of example cases tackled with this type of optimization algorithms.

Application of Optimization-Based Simulation in Power Electronic Design Problems

Chapter 5

5.1 Chapter Overview

In this chapter, a number of examples are presented, in which the developed tools are used for different types of designs. The discussion is initially started with revisiting the two-variable dc-dc converter example of chapter 2, and the optimal design tool is used to find the parameters of the control system. Later a comprehensive design of a dc-dc converter is presented, in which not only control system parameters, but also power circuit elements and the switching frequency of the converter are also optimally selected. The next example will address optimal pulse-width modulation multilevel converters under practical condition where the dc bus voltage has ripple components of arbitrary frequency and amplitude. A mixed-integer optimization example is also presented.

5.2 Optimal Control System Design for a Dc-Dc Converter

The main purpose of this example is to provide an introduction to the issue of optimal design using the tools and techniques developed in chapter 3. Although the case is quite simple and even an exhaustive search (such as MR or random as reported in chapter 2) yields the optimal solution with an affordable simulation effort, it is instructive to do the design using the optimization-enabled simulation.

5.2.1 Operating Principles of Dc-Dc Converters

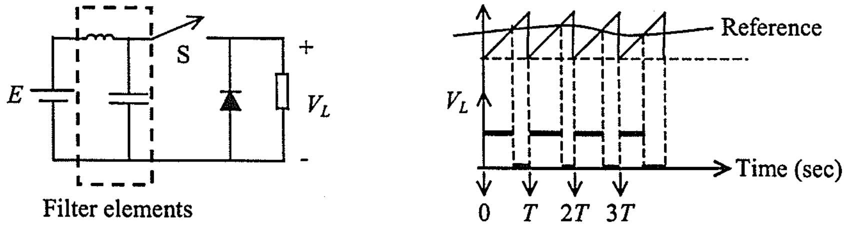
Dc-dc converters are used to produce controllable dc voltage from a fixed dc source. Fig. 5.1 (a) schematically shows the operation of a step-down dc-dc converter, which generates a controlled voltage V_L through switching of the semiconductor switch S. Capacitive and inductive filter elements are usually added to the input to provide a smooth input side voltage and also to prevent ac currents from flowing through the dc source E. Turning the switch ON and OFF results in voltages E and zero to appear across the load. Proper selection of ON and OFF intervals controls the average output voltage across the load.

The switching usually happens at a constant frequency ($1/T$). The ON/OFF intervals of the switch are determined by comparing a reference waveform with a triangular waveform as shown in Fig. 5.1 (b). The duty cycle of the switch is determined using the following formula [16].

$$D = \frac{T_{ON}}{T} \tag{5.1}$$

where T_{ON} is the length of the ON interval of the switch within the switching period. It can be shown that for a given duty cycle (D), the average output voltage is as follows.

$$\langle VL \rangle = D \cdot E \tag{5.2}$$

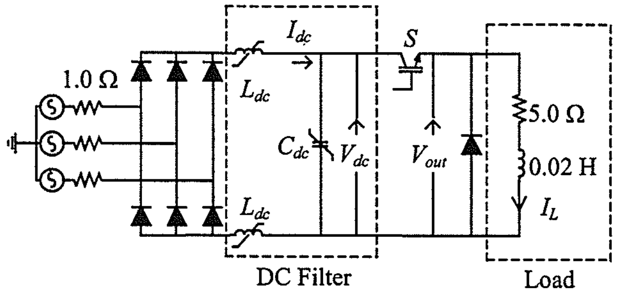


(a) Schematic diagram of a dc-dc converter

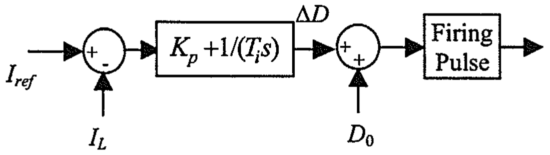
(b) Firing pulse generation and output voltage

Fig. 5.1 Principles of the operation of a step-down converter

As shown in Fig. 5.2, the input side voltage is usually obtained through a rectifier; a closed loop control system is usually used to determine the duty cycle to regulate the load current (or voltage).



(a) Dc-Dc converter



(b) Load current control system

Fig. 5.2 Schematic diagram of the dc-dc converter and its control system

5.2.2 Optimal Design Setup

Dynamic response of the load current depends, among other things, on the control system parameters. Design of the control system for a dc-dc converter should therefore take into account the desirable performance characteristics specified for the load current dynamics.

Design objectives in this example are identical to those specified in chapter 2, and hence the design problem can be stated as follows.

Find $\mathbf{x} = (K_p, T_i)$
such that load current dynamic response
1) has minimum overshoot, (5.3)
2) follows the step reference as fast as possible,
3) has minimum steady state error

These objectives have to be embedded into a suitably defined objective function. This objective function quantitatively measures the closeness of the actual response to the specified goals. The Integral Square Error (ISE) objective function given below serves such a purpose.

$$ISE(K_p, T_i) = \int_{t=T_1}^{T_F} (1 - \frac{I_L}{I_{ref}})^2 dt \qquad (5.4)$$

We note that when the actual load current closely matches the reference current at every point, the ISE objective function attains a low value (exact match at every instant of time causes ISE to attain its lowest possible value of zero). Therefore, the problem can be re-stated in the form of an optimization problem as follows.

$$\text{Minimize } ISE(\mathbf{x}), \ \mathbf{x} = (K_p, T_i) \qquad (5.5)$$

The time interval over which the ISE is evaluated should be long enough to allow steady state to be reached. To ensure positive-ness of the controller gain and time constant, the outputs of the optimization are squared, and the results are used as the actual gain and time constant. Table 5.1 summarizes the design procedure.

Table 5.1 Summary of the design using optimization-enabled transient simulation

Reference current (A)	T_1 (sec)	T_F (sec)	Initial parameter set (K_p,T_i)	Optimized Parameter set (K_p,T_i)	Total number of simulations	Optimal ISE
30	0.1	0.5	[0.16,0.11]	[1.75,0.50E-02]	37	0.00175

The optimization method used for this design is the nonlinear simplex method of Nelder and Mead. Fig. 5.3 shows the dynamic response of the load current for the initial and optimized parameter sets listed above.

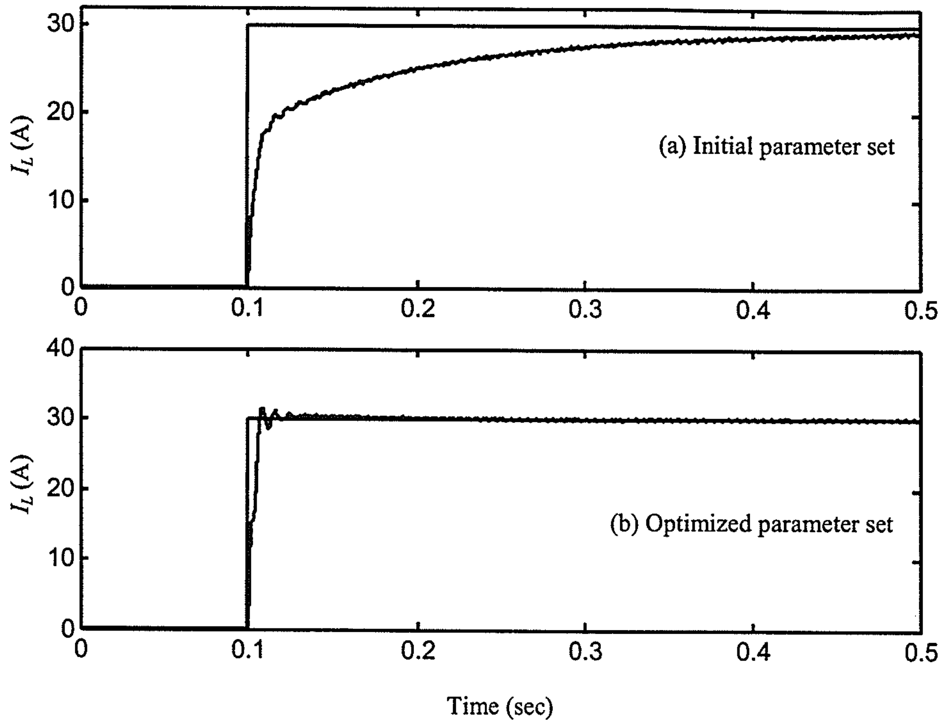


Fig. 5.3 Dynamic response of the load current

The entire design is carried out in a considerably lower number of simulations, than what was obtained using the conventional methods of MR or random search presented in chapter 2 (over 1000 runs in both approaches). This constitutes the major benefit of combining optimization algorithms with the power of transient simulation in the design process.

5.3 A Comprehensive Design of a Dc-Dc Converter

This example demonstrates the use of optimization based simulation to do a composite design of a dc-dc converter. In this exercise, the electrical components such as inductors and capacitors, as well as control system parameters are simultaneously optimized. The circuit under consideration is shown in Fig. 5.2 (a). The 300 V, 60 Hz three phase ac supply is rectified and produces the input side dc supply V_{dc}. Capacitive and inductive filter elements are used to provide smooth dc voltage and rectifier dc current. The performance of the system is not only a function of control system parameters but also a function of filter element components and the switching frequency. The design problem of this converter can be stated as given in (5.6).

Find $\mathbf{x} = (K_p, T_i, L_{dc}, C_{dc}, f_{sw})$
such that we obtain
1) A good steady state load current with minimal harmonic ripple,
2) A good transient response, with the specified rise time, without excessive overshoot,
3) Minimum ripple voltage V_{dc} on the input capacitor and ripple current on the dc current I_{dc}, (the latter is to prevent rectifier current chopping related stresses),
4) As low a switching frequency as possible, in order to limit the losses,

subject to
K_p, T_i, L_{dc}, C_{dc} and f_{sw} being positive (5.6)

Note that the objective (4) conflicts with (1), as a lower switching frequency results in a larger harmonic ripple. Therefore, the process of optimal design is a compromise between competing objectives, whose importance is determined through the design of objective functions.

- *Design of the Objective Function*

In this problem, the OF contains several terms, each of which may have different units or scales. For example, the term representing the penalty for a higher switching frequency is quite different from the term penalizing a large steady-state ripple. In order to make a meaningful OF, these different measures have to be combined using suitable weighting functions, as shown below.

$$OF(\mathbf{x}) = \sum_i W_i \cdot M_i(\mathbf{x}) \tag{5.7}$$

where **x** and *OF*(**x**) are the candidate point and its overall function evaluation and W_i is the weighting given to the *i*-th performance measure $M_i(\mathbf{x})$.

The objective function given below is chosen to embed the design objectives as specified in (5.6). The reference load current for this design is 45 A.

$$OF(K_p, K_i, f_{sw}, L_{dc}, C_{dc}) = W_1 \cdot ISE_1 + W_2 \cdot f_{sw} + W_3 \cdot ISE_2 \tag{5.8}$$

- *Design of Individual Measures of Performance*

The first term in (5.8) contains the sub-measure ISE_1 for the performance of the control system, which is identical to (3.1) with x replaced by I_L and addresses objectives (1) and (2) of (5.6). The 'desired' maximum permissible overshoot (M_p) as well as rise and settling times (T_r,T_s) are as shown in Table 5.2. The weightings given to the three parts of ISE_1 (K_1, K_2 and K_3 in (3.1)) are equal to unity. Note that there is no guarantee that the final optimal design has all the above characteristics; fulfillment of the design specifications in some cases may not be possible due to the fact the selected structure of the control system (PI in this case) is not capable of meeting the objectives. The optimal design procedure however, tries to determine the parameters of the specified structure so that the best possible performance is obtained. It is also possible to penalize deviations in more critical factors by using larger relative weights.

The second term $W_2 \cdot f_{sw}$ penalizes a high switching frequency. The third term with ISE_2 satisfies objective (3) by taking into account the smoothness of the input side voltage and current waveforms, and has the following form.

$$ISE_2 = \int_{T_1}^{T_F} (A_1(I_{dc} - \langle I_{dc} \rangle)^2 + A_2(V_{dc} - \langle V_{dc} \rangle)^2)dt \tag{5.9}$$

where T_1 is the initial time of ISE_2 calculation and T_F is the final simulation time. T_1 is chosen such that to ensure only steady state ripple of the voltage and current waveforms are taken into account. A_1 and A_2 are the weightings given to the two parts to ensure their comparable contribution to ISE_2. The quantities $\langle I_{dc} \rangle$ and $\langle V_{dc} \rangle$ represent the average steady state values of the input dc current I_{dc}, and voltage V_{dc}, respectively. Since $\langle V_{dc} \rangle$ (=337.5 V) is approximately 11 times larger than $\langle I_{dc} \rangle$ (=30 A), the weighting factors A_1 and A_2 are selected to be 1 and $1/11^2$, respectively.

- *Constraints*

In this problem, it is necessary to scale the search variables themselves. For example, the inductance, capacitance, and switching frequency all have different units, and hence different numerical value ranges, i.e., f_{sw} is in the range of 1000 Hz, L_{dc} in the range of mH (10^{-3}) and C_{dc} in the range of 100 μF (10^{-4}). In order to improve the efficiency of the search, it is desirable to scale (or per-unitize) the search variable. Note that the scaling only applies to the points generated in the optimization algorithm; they must be re-scaled to their original system of units and ranges when used in the EMTDC simulation.

Optimization literature indicates a variety of approaches that can be used to ensure that the optimization program does not select candidate points in a non-feasible range [9]. Simply assigning a fixed large value to the OF for such points is not recommended, as it makes the function discontinuous at the boundaries and degrades the numerical performance of the search. Sometimes the constraints can be eliminated by a variable substitution. In this example we have used squaring of the internal variables to ensure positive-ness of the actual optimization parameters.

- *Numerical Results*

Each of these objectives in (5.8) is given a weight (W_1, W_2 and W_3) according to their relative importance (as judged by the user). The weights selected here are shown in Table 5.2, which also shows the optimization parameters, the starting points given and final solution obtained.

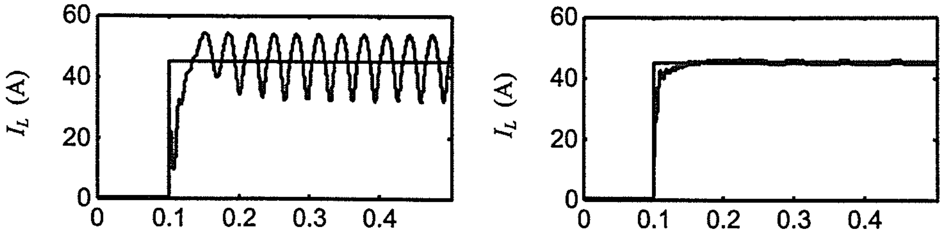
Table 5.2 Optimization parameters and results for the dc-dc converter design

<table>
<tr><td colspan="7">Overall objective function</td></tr>
<tr><td colspan="2">W_1</td><td colspan="2">W_2</td><td colspan="3">W_3</td></tr>
<tr><td colspan="2">1.5</td><td colspan="2">0.005</td><td colspan="3">5.0</td></tr>
<tr><td colspan="7">ISE_1</td></tr>
<tr><td>K_1</td><td>K_2</td><td>K_3</td><td>T_r (sec)</td><td>T_s (sec)</td><td>M_p</td><td>E_s</td></tr>
<tr><td>1.0</td><td>1.0</td><td>1.0</td><td>0.02</td><td>0.05</td><td>6.75 (15%)</td><td>2.25 (5%)</td></tr>
<tr><td colspan="7">ISE_2</td></tr>
<tr><td colspan="2">T_1 (sec)</td><td colspan="2">T_F (sec)</td><td colspan="2">A_1</td><td>A_2</td></tr>
<tr><td colspan="2">0.4</td><td colspan="2">0.5</td><td colspan="2">1.0</td><td>0.0083</td></tr>
<tr><td colspan="7">Optimization initial guess</td></tr>
<tr><td>K_p</td><td>K_i ($1/T_i$)</td><td colspan="2">f_{sw} (Hz)</td><td colspan="2">L_{dc} (mH)</td><td>C_{dc} (µF)</td></tr>
<tr><td>2.0</td><td>90.0</td><td colspan="2">2000</td><td colspan="2">50.0</td><td>150</td></tr>
<tr><td colspan="7">Optimization final results</td></tr>
<tr><td>K_p</td><td>K_i ($1/T_i$)</td><td colspan="2">f_{sw} (Hz)</td><td colspan="2">L_{dc} (mH)</td><td>C_{dc} (µF)</td></tr>
<tr><td>1.83</td><td>101.3</td><td colspan="2">819.9</td><td colspan="2">5.5</td><td>172.4</td></tr>
<tr><td colspan="7">Variable scale factors</td></tr>
<tr><td>For K_p</td><td>For K_i ($1/T_i$)</td><td colspan="2">For f_{sw} (Hz)</td><td colspan="2">For L_{dc} (mH)</td><td>For C_{dc} (µF)</td></tr>
<tr><td>0.02</td><td>1.0</td><td colspan="2">100</td><td colspan="2">0.001</td><td>1.0</td></tr>
</table>

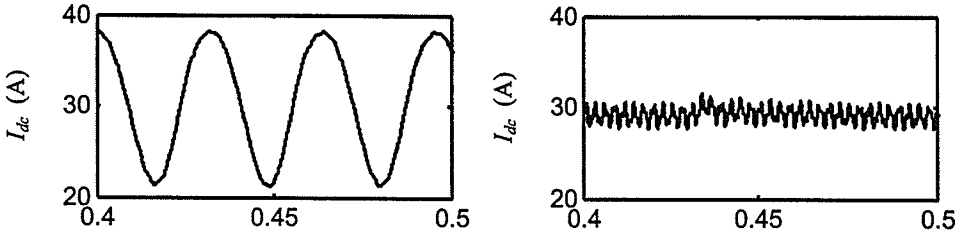
Fig. 5.4 shows the load current, and steady state dc link voltage and current waveforms for initial and optimized values of the parameters, respectively.

It is observed that significant improvement in the performance of the system is obtained, while switching frequency is reduced more than half of its initial value and therefore switching losses are also considerably lowered.

(a) Dynamic response of the load



(b) Steady state inductor current



(c) Steady state dc link voltage

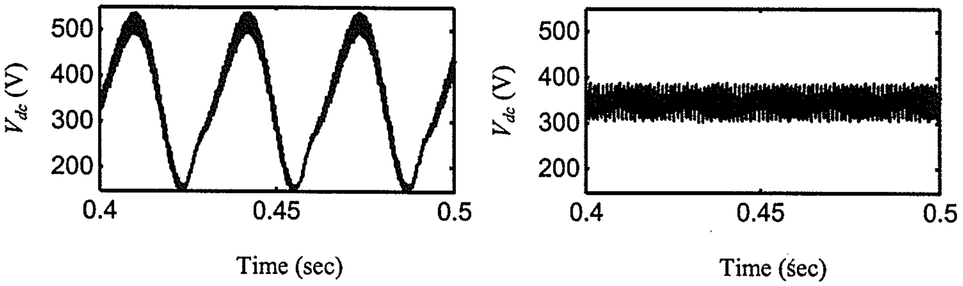


Fig. 5.4 Pre-optimized (left) and optimized (right) performance of the system for the 45 A operating point

It is interesting to note that using the conventional multiple-run approach, the same optimization problem with only 10 steps for each of the 5 variables would require 100,000 (=10^5) runs. The accuracy of the solution would also be limited by this relatively coarse search grid. With optimization-enabled emtp simulation the final solution is obtained in less than 300 runs and provides a considerably higher level of accuracy as well.

5.4 Optimal Pulse-Width Modulation for Multi-Level Converters

In this section, we use optimization based transient simulation to calculate the switching angles of a pulse-width modulated (PWM) voltage sourced converter (VSC) in order to improve the harmonic quality of its output waveform.

5.4.1 Principles of OPWM

Optimal PWM, also known as Selective Harmonic Elimination (SHE), is a method for generating switching patterns for VSCs. Unlike ordinary PWM scheme, which require high switching frequencies to shift undesirable harmonics to higher frequency bands [16], OPWM operates based on a few chopping angles per quarter cycle, which are optimally selected to shape the harmonic spectrum of the output voltage [17]. The main advantage of SHE over ordinary PWM is its lower switching losses. The disadvantage lies in fact that SHE is inherently an offline method and the chopping angles should be pre-calculated for a wide range of operating points and stored in a memory [18].

Fig. 5.5 shows an idealized waveform for the three-level VSC of Fig. 5.6. Through the switching of the appropriate semiconductor switches, a waveform with three levels (*E*, 0, -*E*), where *E* is the dc bus voltage, can be applied to the load.

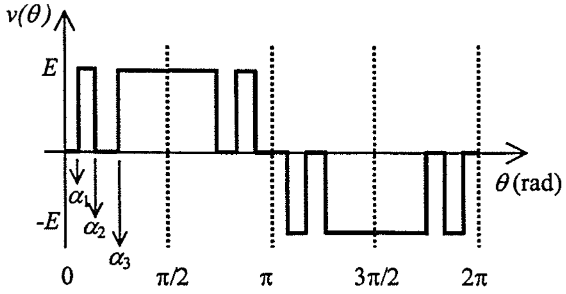


Fig. 5.5 A typical three-level waveform

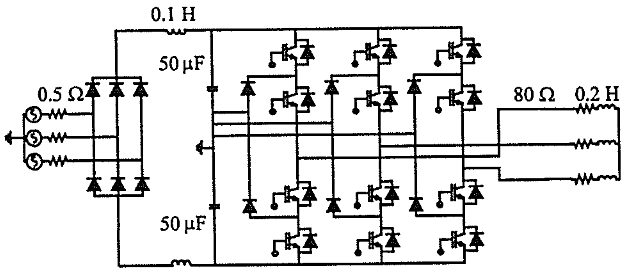


Fig. 5.6 Three-level voltage-sourced converter in a back-to-back arrangement

Of course, it should be noted that the waveform is idealized, in that the dc bus voltage is actually the output of a rectifier and has additional ripple, which is not shown. The number of switchings in a quarter cycle is controlled by the N (N=3 in Fig. 5.5) switching angles $\alpha_1, \alpha_2, \ldots, \alpha_N$.

These N angles can be optimally selected to achieve N different objectives, such as the elimination of N-1 specified harmonics and the regulation of the fundamental voltage. For the ideal case of constant dc link voltage, the analytical expression for the amplitude of the k-th harmonic of the voltage waveform in terms of the chopping angles is as follows [19]:

$$V_k = \frac{4E}{k\pi}\sum_{i=1}^{N}(-1)^i \cos(k\alpha_i)\,, \quad N\text{: odd, } k=1,5,7,11,13,\ldots \tag{5.10}$$

where E is half of the dc link voltage. Setting $V_1=V_{ref}$ and $V_i = 0$ for i in the list of N-1 harmonics targeted for removal, it is possible to solve (5.10) to analytically obtain solutions for $\alpha_1, \alpha_2, \ldots, \alpha_N$. However due to the fact that in reality the dc voltage is not constant and has ripple on it, the angles calculated from (5.10) are only approximate.

There has been research on developing techniques for the reduction of ac side harmonics in the presence of dc side ripple. Techniques based on variable modulating functions [20] and correction of the OPWM patterns [21] are proposed; however these methods mostly require a precise knowledge of the amplitude and phase angle of the ripple component, which is usually not known. In contrast, nonlinear optimization based EMTDC simulation can be used to determine the true chopping angles under such conditions.

5.4.2 Optimal PWM under Practical Situations

In the example discussed here the 20 kV(ac, l-l), 60 Hz dc-link fed 3-level VSC in Fig. 5.6 is controlled with three chops per quarter cycle. The resulting three degrees of freedom are used to regulate the magnitude of the fundamental component and to eliminate the two most problematic lowest order harmonics in the line voltage waveform, namely the 5th and the 7th. The diode bridge rectifier results in an average nominal dc link voltage of 27 kV.

The OF used for this problem is an ordinary ISE function as follows.

$$OF(\alpha_1,\alpha_2,\alpha_3)=\int_{t=t_0}^{T}((v_1-V_{1-ref})^2+v_5^2+v_7^2)dt \tag{5.11}$$

where v_1, v_5 and v_7 are magnitudes of the fundamental, 5th and 7th components of the ac phase voltage, and V_{1-ref} is the reference fundamental voltage. T and t_0 are total simulation time and start time of ISE calculation, which in this case are set to 0.4 sec and 0.1 sec, respectively. Note that if all objectives i.e., $v_1=V_{1-ref}$ and $v_5=v_7=0$ are met, the OF in (5.11) achieves its minimum possible value of zero.

- *Optimization Results*

Table 5.3 shows the optimization results for this case, when the fundamental phase voltage is regulated to 10 kV (rms). Chopping angles from (5.10) (calculated with the assumption of constant dc bus voltage) as well as the starting and converged values from optimization are shown. The system is simulated with a 10 μs time step. The interpolation function of PSCAD/EMTDC is active, for further reducing time-step related jitter in the implementation of the firing angles [28].

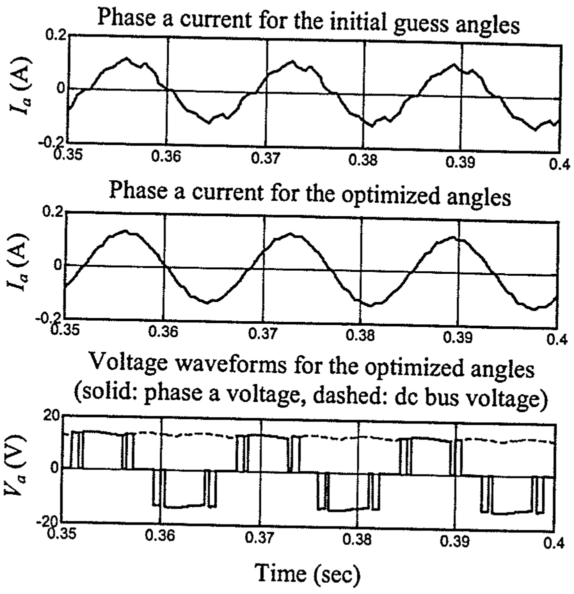
Table 5.3 Optimization results for the voltage sourced converter

	α_1	α_2	α_3
Angles using (5.9)	22.4°	37.7°	46.4°
Optimization initial guess	15°	30°	50°
Optimization converged results	22.43°	37.44°	46.24°

The results show that the method is indeed able to achieve the desired objectives and that the presence of ripple in this case makes a very marginal difference from the solution obtained analytically assuming constant dc link voltage. As an additional confirmation, simulation with the dc link replaced by a dc source yielded exactly the same solution as the analytical one.

Fig. 5.7 shows the steady state phase voltage and current as well as the bus voltage across the upper capacitor, which clearly shows the ripple imposed on it. The current waveform is shown for both initial and optimized angles. Due to the elimination of lower order harmonics, the converged waveform is smoother.

Fig. 5.8 shows the harmonic spectrum of the voltage waveform before and after optimization. It is observed that the optimization process yields the desired fundamental voltage (10 kV) and reduces the targeted 5th and 7th harmonics to near zero. Higher order harmonics not targeted for elimination, are however amplified- a well-known property of PWM converters. Nevertheless, they have a smaller impact due to the naturally higher impedance of the ac system at these higher frequencies.


Phase a current for the initial guess angles

Phase a current for the optimized angles

Voltage waveforms for the optimized angles
(solid: phase a voltage, dashed: dc bus voltage)

Fig. 5.7 Voltage and current waveforms before and after optimization

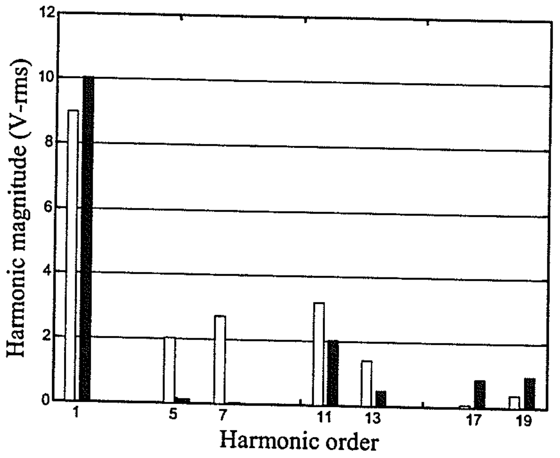


Fig. 5.8 Spectrum of the ac voltage waveform before and after optimization
(white bars: initial guess angles; black bars: optimized angles)

The variation of the OF (ISE) versus run number is shown in Fig. 5.9, which shows the evolution of the optimization process. Successive trial points essentially show a monotonic decrease, with only a few jumps due to the search process. Using the traditional approach of sequential or multiple-runs on the other hand would not have resulted in such a monotonic decrease. This demonstrates that the trial points selected through optimization are better at consistently lowering the objective function, and hence result in a much smaller number of runs.

The same simulation case can be easily modified to study related problems such as harmonic elimination solutions for ac system unbalances, unequal parameters in each phase, the presence of non-characteristic harmonics and so on.

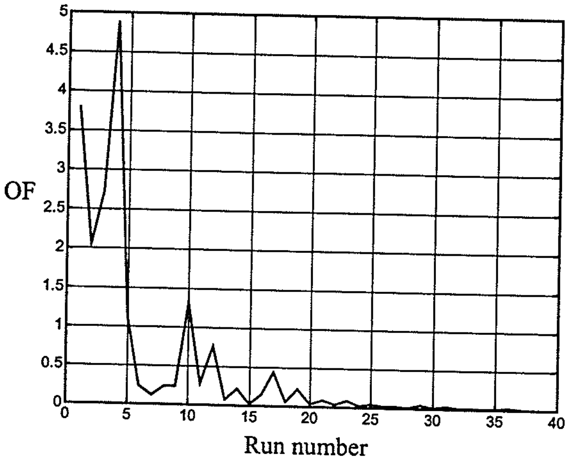


Fig. 5.9 ISE variation versus run number

5.5 OPWM with DC Bus Voltage Ripple - Further Investigations

The optimization results of the previous section show that under practical situations where the dc bus capacitance is large enough (hence the ripple is reasonably small), the optimized angles are very close to those obtained from idealized analysis (see (5.10)). In other words, the ideally calculated angles continue to function satisfactorily under practical situations as long as the ripple on the dc bus voltage is kept small.

In this section we investigate this issue in more depth by considering the three-level converter of the previous section with a non-ideal dc bus, which can contain various amounts of harmonics of different orders.

Nonlinear optimization is used to find three chopping angles required for regulating the fundamental component of the phase voltage as well as eliminating 5th and 7th order harmonics, when the dc bus voltage has a given harmonic content.

Three sets of test are carried out on the system: (1) ideal dc bus voltage, i.e. dc bus voltage is constant at 13 kV, (2) 5% of the 6th order harmonic is present on the dc bus voltage, (3) 5% of the 5th order harmonic is present on the dc bus voltage.

Note that for balanced three-phase operation, (2) and (3) represent characteristic and non-characteristic harmonics, respectively.

For each of the above cases, the fundamental is varied from 6.0 kV to 10.5 kV, and the angles are calculated using the optimization-enabled transient simulation. The results are shown on Fig. 5.10.

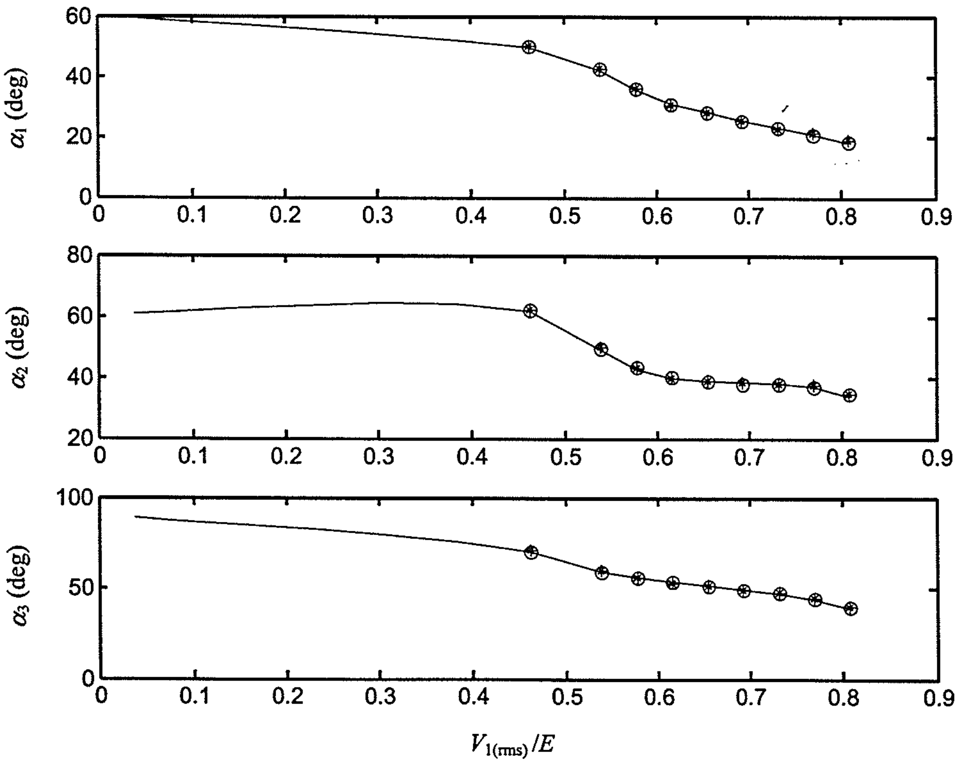


Fig. 5.10 Variation of angles versus fundamental output voltage
Solid line: ideal case, *: 5% of 6th harmonic, o: 5% of 5th harmonic

The results, even with 5% of characteristic and non-characteristic harmonics on the dc bus, are close to those of the ideal case. Further investigations have shown that the ideally calculated angles still eliminate the target harmonics as long as the amount of the ripple or system unbalance is reasonably small [24].

5.6 Mixed-Integer Optimization Using Genetic Algorithms - A Demonstrative Example

So far, the examples presented involved problems in which real-valued variables were to be optimized. The choice of the optimization method to use was also based on this essential fact.

In this section an optimization problem of a different nature is considered. The problem consists of selecting the proper structure as well as component values for a load that absorbs maximum power from an ac source. The problem of choosing the proper structure among a given number of load structures is an integer-valued optimization, whereas selecting the optimal value of parameters involves real variables, hence a mixed-integer problem.

A simple problem with known analytical solution has been chosen for the demonstrative example, as then the validity of the GA-based method can be readily established.

5.6.1 Problem Setup and the Analytical Solution

Fig. 5.11 shows the ac network as well as two different structures for the load, being an *RL* and an *RC* one. The solution can be easily found using the maximum power transfer theorem. According to this theorem to absorb maximum power from the source, the load impedance should be the complex conjugate of the source impedance at the frequency of operation, i.e., $\mathbf{Z_L} = \overline{\mathbf{Z_S}}$. To be able to compensate for the inductive imaginary part of the source impedance, the load should be an *RC* one; therefore, the load structure that can absorb maximum power from the source is the *RC* combination (Load 2), and the component values will be as follows.

$$R_2 = R_S = 5\ \Omega$$
$$C_2 = \frac{1}{\omega^2 L_S} = \frac{1}{(2\pi 60)^2 1} = 7.036\ \mu\text{F}$$

The GA solution will essentially demonstrate the capability of the algorithm in finding the appropriate load structure as well determination of parameter values.

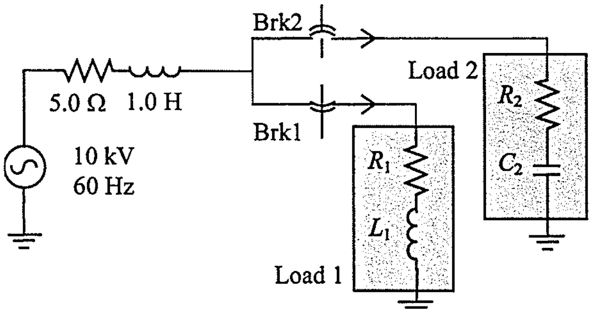


Fig. 5.11 Maximum power transfer theorem using GA

5.6.2 GA Optimization Setup and Results

The GA solution to the above problem starts with the coding of the parameters into a chromosome. The chromosome includes a switching (on/off) variable for the state of breaker 1 (the switches have complementary states), and four real variables being the values of R_1 (Ω), L_1 (H), R_2 (Ω) and C_2 (μF), respectively. Therefore the encoded variables form a chromosome as shown in Fig. 5.12.

Brk_1	R_1	L_1	R_2	C_2

Fig. 5.12 Chromosome structure

The solution was started with 100 initial population instances of the chromosomes; subsequent generations each had 50 chromosomes and the mating pool was comprised of the top 30 chromosomes in each generation. The mutation rate for the switching and the real sections of the chromosomes were set to 5% and 20%, respectively. For the pairing of chromosomes the tournament method was used. Table 5.4 shows the summary of the GA solution.

Table 5.4 GA solution to the maximum power transfer problem

Number of simulations	Optimal state of switch 1	Optimized R_1 (Ω)	Optimized L_1 (H)	Optimized R_2 (Ω)	Optimized C_2 (μF)
900	off	6.93 (irrelevant)	0.053 (irrelevant)	4.998	7.039

With the specified numbers of the initial and surviving populations, the solution is obviously found in the 17th generation following the initial generation. It is also instructive to see the average fitness of the generations as the evolution of generation proceeds. Fig. 5.13 shows the average fitness of the generations as a function of the generation number. The fitness is a measure of the average power delivered to the load in the steady state. The graph shows that the successive generations are evolved so that the load absorbs more power from the source.

As a general conclusion, one may observe that the solution using GAs is intensive in terms of computational effort, and requires several simulations before the solution is found. However, in cases where multiple optima exist or optimization variables take on integer as well as real values, the GA solution is more justified as it increases the chance of finding the global optimum and also provides a simple method for mixed-integer problem solving.

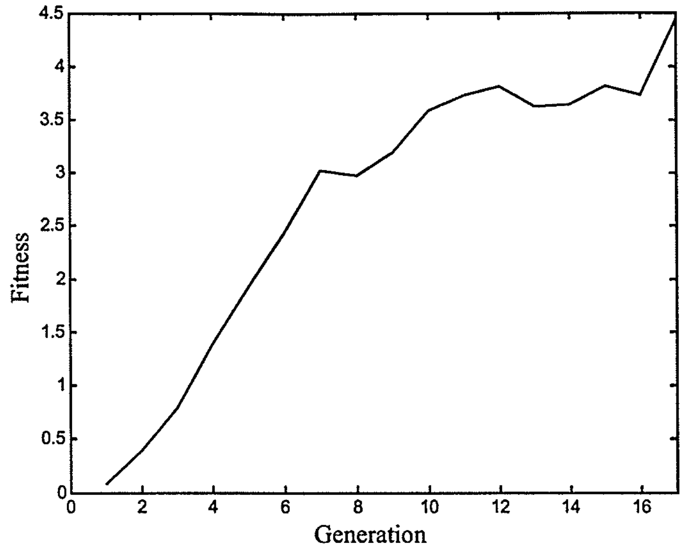


Fig. 5.13 Average fitness of generations

It is important to note that the GA approach shows a high level of parallelism. Evaluation of the fitness of the members of a certain generation is essentially a parallel task and can be assigned to a number of processors. In this sense, the GA solution in the above example essentially requires 17 steps, each representing a generation.

In a parallel implementation of a GA-based approach, the main processor can be used to generate the chromosomes in a generation and assign the fitness evaluation to individual processors working in parallel.

Chapter 6

Inclusion of the Robustness into the Design Procedure

6.1 Introduction and Chapter Overview

Throughout the previous chapters, a general framework was established for interfacing nonlinear optimization algorithms with electromagnetic transient simulation programs with the aim of developing a powerful tool for the optimal design of complicated nonlinear systems, such as power electronic converters. Methods were proposed for incorporating design objectives into meaningful, well-defined objective functions, and later on they were used along with the optimal design tool for the design of some nonlinear, switching systems. The proposed tool proved to be significantly more efficient than the conventional design methods, e.g. MR and random search methods, in terms of computer resources and accuracy of the results.

It is important to note that power systems and components are usually nonlinear dynamical systems that can show significantly different behavior under different operating points or configurations. Therefore, due to these non-linearities inherent in any large power network, the optimized parameter set, obtained using the basic interface described previously, may be sub-optimal or even inappropriate for other system operating conditions. In other words, the process of optimal design, as described so far, is operating point dependent and the optimal parameter set so obtained does not necessarily guarantee optimal performance for operating points other than its designated one.

This chapter will introduce a modified interface between transient simulation and optimization to enable robust optimal design. Example cases for the application of the new tool will also be presented.

6.2 Robust Optimal Design Interface [25]

The main reason why the optimal setting of parameters as obtained by using the basic interface is not robust to the changes in the operating point is that the process of optimization in that case only considers a single operating point or configuration. The performance of the system for a given parameter set under other operating points was not incorporated into the design.

This section introduces a modified interface between transient simulation and an optimization algorithm, which enhances the design of optimal systems by searching for a parameter set that has the best overall performance over a range of operating points and hence remains robust to operating point variations. In this tool the performance of a parameter set is evaluated not merely at one operating point (like the original interface), but over a range of such points, resulting in a cumulative or aggregate objective function. The nonlinear optimization algorithm then adjusts the candidate set of design parameters based on how well it minimizes this cumulative OF. The optimized parameter set will thus be the best performer for the entire range of operating points considered and its optimal performance will therefore remain robust.

It is important to note that the parameter set optimized with such a view will not necessarily be the best performer for any of the operating points, if considered individually, but will have the best *overall* performance.

Fig. 6.1 shows the proposed schematic diagram of the robust optimal design tool.

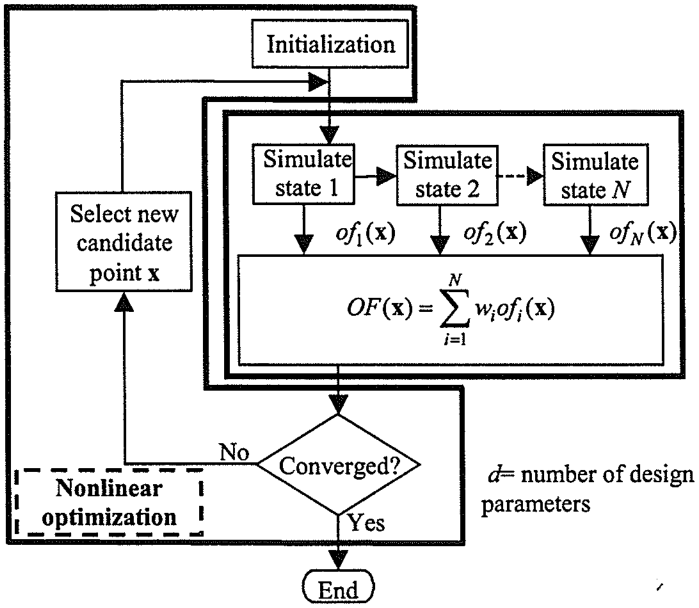


Fig. 6.1 Robust optimal design interface

As it is shown, the objective function evaluation block now consists of a number of transient simulation runs, each corresponding to the same trial point (parameter set) but for a different operating point or condition (state). The aggregate objective function ($OF(\mathbf{x})$) is the weighted sum of the individual objective functions $of_i(\mathbf{x})$ (referred to as "partial objective functions") for each run (note lowercase letters). Obviously optimizing the parameter set $\mathbf{x}$ to minimize $OF(\mathbf{x})$ gives a solution that is optimized to the aggregate, but is not necessarily optimal for any of the individual operating conditions (respectively, with objective functions $of_i(\mathbf{x})$).

The weights w_i are chosen to weight the individual runs differently if so desired. They can be used to emphasize the importance of one or some of the operating points or conditions.

Design of partial objective functions corresponding to individual operating conditions follows the same general rules as outlined in the previous sections. Their aggregation into the cumulative $OF(\mathbf{x})$, however determines the nature of the compromise the optimization algorithm has to make in order to minimize the aggregate OF. Similar to a conventional optimization problem, the robust optimal design is also a compromise between competing factors not only within the partial objective functions, but also within the aggregate one. Larger gains or weights generally show a more important index, and therefore the focus of the compromise will be towards the indices with larger weights.

6.3 Basic Interface versus Robust Optimal Design

We mentioned that the basic interface of Fig. 3.1 can be used to determine the optimal setting of design parameters for a given operating condition, and that its results are usually sensitive to the changes in the operating point. The process of inclusion of robustness into the design was done through the modified interface of Fig. 6.1.

A question that can logically be asked is whether the basic interface can be of any use when a seemingly more powerful tool is also available. The answer to this question is affirmative. Indeed, the robust optimal design tool cannot replace the basic tool in all design problems.

We firstly re-assert that, the solution obtained by the robust optimal design tool is the best performer overall and is not necessarily optimal for any single operating point within the range. This can potentially be undesirable, because it is possible to achieve better performance for a given operating point if a specific design is carried out for that point using the basic interface. With the decreasing cost of memory storage, it is easily possible to design the optimum parameter sets for a wide range of operating points and store them in a large table. A higher level control can adaptively extract the optimum parameter set (from the table) corresponding to the current operating point so that for each operating point the best performance is attained (a simple example of this approach is the method of gain scheduling that finds numerous applications in control systems).

The second point is that there are some applications where robustness is not an issue of concern at all. An example of this case is the optimal pulse-width modulation of power electronic converters, in which optimal angles have to be specifically designed for each operating point and robustness is out of question. In this case, a single set of chopping angles designed with a robustness view will deteriorate the essential assumptions of OPWM.

Altogether, it can be inferred that the design tools introduced in chapter 3 and this chapter, are complementary and their application in various design problems should be done with careful examination of the problem under consideration as well as the goals sought after through the design.

6.4 Example Cases

6.4.1 Dc Power Supply

The schematic diagram of a dc-dc converter with its closed loop load current control system was shown in Fig. 5.2(a) and 5.2(b) [16]. Unlike the design presented in section 5.2, in this section we consider a robust optimal design of the converter. The design objectives are listed below. The switching frequency is kept constant at 1800 Hz and is therefore omitted from the list of design parameters.

Find $\mathbf{x} = (K_p, T_i, L_{dc}, C_{dc})$
such that we obtain
1) A good steady state load current with minimal harmonic ripple,
2) A good transient response, with the specified rise time, without excessive overshoot,
3) Minimum ripple voltage V_{dc} on the input capacitor and ripple current on the dc current I_{dc}, (the latter is to prevent rectifier current chopping related stresses), (6.1)
4) Robust operation, i.e. the above objectives should be maintained when operating at large, medium or small load currents,

subject to
K_p, T_i, L_{dc}, and C_{dc} being positive

The desired objectives listed above can be encapsulated into a partial objective function for each of the operating conditions as follows:

$$of(\mathbf{x}) = M_1 \times ISE_1 + M_2 \times ISE_2 \quad \text{where } \mathbf{x} = (K_p, K_i, L_{dc}, C_{dc})^T \tag{6.2}$$

where ISE_1 and ISE_2 are the performance indices for the load current dynamic response and dc bus voltage and current steady state ripple, respectively. M_1 and M_2 are weighting factors given to the indices to ensure that they contribute evenly to the objective function. The performance measures used are defined below.

$$\begin{aligned} ISE_1 &= \int_{T_0}^{T_F} (1 - \frac{I_L}{I_{L-ref}})^2 dt \\ ISE_2 &= \int_{T_1}^{T_F} \left[K_1 (1 - \frac{I_{dc}}{\langle I_{dc} \rangle})^2 + K_2 (1 - \frac{V_{dc}}{\langle V_{dc} \rangle})^2 \right] dt \end{aligned} \tag{6.3}$$

where $\langle I_{dc} \rangle$ and $\langle V_{dc} \rangle$ are the average values of the steady state dc bus voltage and current. Weighting factors K_1 and K_2 determine the contribution of voltage and current ripple to ISE_2, respectively. Notice that if the performance is ideal, i.e., the reference and actual values match everywhere and there is no ripple voltage or current, the objective function of (6.2) reduces to its minimum possible value of zero.

In order to demonstrate why a robust procedure is necessary, the following sub-section first shows that optimizing the parameters for one operating point can often lead to a poor response for other operating points. In the subsequent sub-section, the robust method is used to rectify this problem.

A. Design for a Single Operating Point - Lack of Robustness

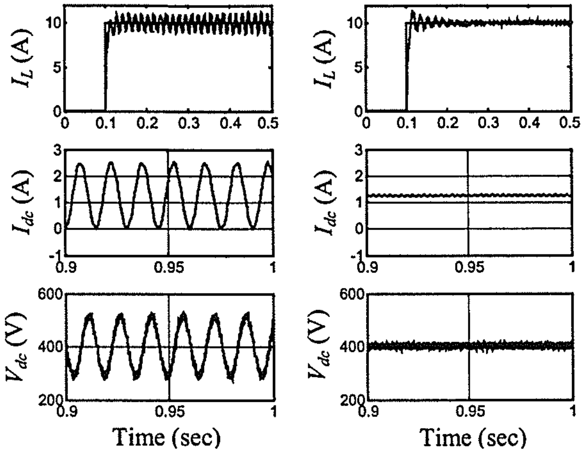
Table 6.1 summarizes the results obtained from optimal design for the single operating condition corresponding to I_{ref} = 10 A (a step waveform applied at 0.1 sec), which is considered as a "small" load current.

Table 6.1 Optimization parameters and results for the operation at 10 A

<table>
<tr><td colspan="60">ISE_1</td></tr>
<tr><td colspan="20">M_1</td><td colspan="20">T_0</td><td colspan="20">T_1</td></tr>
<tr><td colspan="20">150</td><td colspan="20">0.1 sec</td><td colspan="20">0.5 sec</td></tr>
<tr><td colspan="60">ISE_2</td></tr>
<tr><td colspan="15">M_2</td><td colspan="15">T_1</td><td colspan="15">K_1</td><td colspan="15">K_2</td></tr>
<tr><td colspan="15">1000</td><td colspan="15">0.4 sec</td><td colspan="15">1.69</td><td colspan="15">1.69</td></tr>
<tr><td colspan="60">Initial guess</td></tr>
<tr><td colspan="12">K_p</td><td colspan="12">$T_i = 1/K_i$ (sec)</td><td colspan="12">L_{dc} (H)</td><td colspan="12">C_{dc} (μF)</td><td colspan="12">of</td></tr>
<tr><td colspan="12">2.08</td><td colspan="12">0.48e-2</td><td colspan="12">0.11</td><td colspan="12">23.4</td><td colspan="12">83.6</td></tr>
<tr><td colspan="60">Optimized results</td></tr>
<tr><td colspan="12">K_p</td><td colspan="12">$T_i = 1/K_i$ (sec)</td><td colspan="12">L_{dc} (H)</td><td colspan="12">C_{dc} (μF)</td><td colspan="12">of</td></tr>
<tr><td colspan="12">1.27</td><td colspan="12">0.40e-2</td><td colspan="12">0.13</td><td colspan="12">29.8</td><td colspan="12">0.78</td></tr>
</table>

Fig. 6.2(a) and (b) show the step response of the dc load current as well as steady state values of the rectifier side dc current and voltage with the initial and optimized parameters respectively. It is evident that the optimized parameters result in a much improved transient step response as well as in significant reduction in ripple (of = 0.78) as compared with the pre-optimized results (of = 83.6).

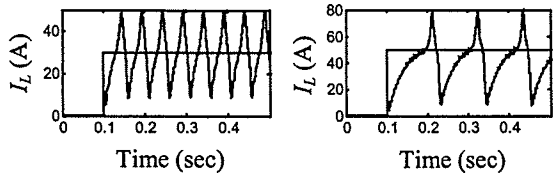
Although the performance with 10 A dc current is very good, the design is not robust as seen from the extremely poor dynamic response shown in Fig. 6.3 for the "medium" (30 A) and "large" (50 A) values of load current. The next section describes a solution that overcomes this serious drawback with the proposed robust optimization method.



(a) Pre-optimized parameter set

(b) Optimized parameter set

Fig. 6.2 Optimal design for the operating point corresponding to I_{ref} = 10 A



(a) I_{ref} = 30 A

(b) I_{ref} = 50 A

Fig. 6.3 Dynamic response of the system for two other operating points

B. Development of the Robust Optimal Design

As it was pointed out earlier, the robust optimization method requires the optimization of an aggregate objective function over several operating points. In this particular example we have selected three points that span the operating range, namely 10 A (low current), 30 A (medium current) and 50 A (high current).

The individual objective functions $of_i(\mathbf{x})$ are calculated exactly as mentioned above according to (6.2) and (6.3). The respective values of the average steady state current $\langle I_{dc} \rangle$ and voltage $\langle V_{dc} \rangle$ used in (6.3) for these cases are shown in Table 6.2. The aggregate OF is calculated as shown in Fig. 6.1, with equal weights $w_i = 1.0$, where $i = 1,2,3$ corresponds to the operating points at 10A, 30 A and 50 A, respectively. This weight distribution indicates that the performance at all these operating points is considered to be equally important. If desired, other weights can be chosen; for example, the operation at 10 A may happen very infrequently and may be assigned a relatively smaller weight.

Table 6.2 Steady state quantities for the dc-dc converter circuit

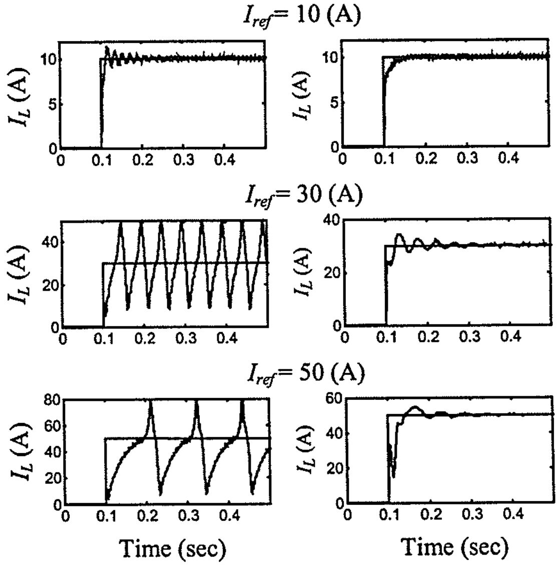
Operating point (I_L)	$\langle I_{dc} \rangle$	$\langle V_{dc} \rangle$
10 A	1.3 A	400 V
30 A	11.7 A	384 V
50 A	37.5	333 V

The aggregate objective function $OF(\mathbf{x})$ thus calculated is optimized and the optimal parameters so obtained are shown in Table 6.3. Note that they are quite different from the ones calculated earlier without consideration of robustness as shown in Table 6.1. The final results from the non-robust optimization (Table 6.1) were used as the starting point for the parameter set $\mathbf{x}$. The convergence to the robust parameter set took 231 simulations. The results for the response of the load current are shown in the second column of Fig. 6.4. Note that as the starting point for the robust optimization was the earlier non-robust parameter set, comparison with the left hand column of Fig. 6.4 shows the improvement introduced by the robust algorithm. Notice that the performance at 30 A and 50 A is now significantly improved. The response for the 10 A case is also good, although it has a larger rise time compared to the earlier case, which was optimized for

10 A (the partial objective function is 1.43 as opposed to 0.78 obtained earlier). This shows that the new result is slightly less optimal for the 10 A condition, but is superior overall.

Table 6.3 Robust optimization results for the dc-dc converter

Optimization final results				
K_p	$T_i = 1/K_i$ (sec)	L_{dc} (H)	C_{dc} (μF)	*of*
1.511	0.769e-2	0.415e-1	491.8	3.06

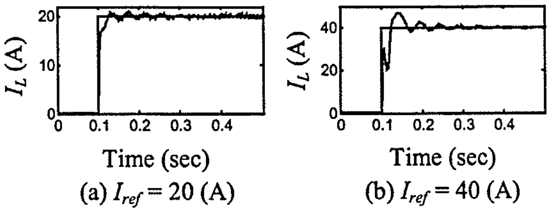


(a) Pre-optimized parameter set (b) Optimized parameter set

Fig. 6.4 Robust optimal design

The robust optimization procedure only allows a finite number of operating points (10A, 30 A and 50 A in the example) to be considered. Fig. 6.5 shows the response for I_{ref}= 20 A and 40 A respectively; which were not explicitly included in the objective function. The figure shows that the parameter set still gives acceptable results.

It should be pointed out that it is not at all necessary to use the non-robust optimized parameters as the seed parameters for the robust optimization procedure. If one starts from the same starting point as in Table 6.1, the robust point is obtained in 228 runs. Note that this number is still orders of magnitude smaller than what would be required with a multiple-run search. For example, with just 10 equally spaced trial values for each of the four search parameters there would be a total of 10^4=10,000 runs.



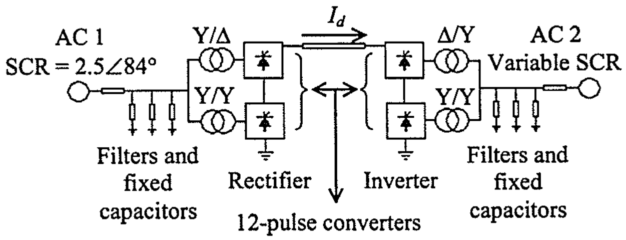
(a) I_{ref} = 20 (A) (b) I_{ref} = 40 (A)

Fig. 6.5 Dynamic response of the robust optimal parameter set for two other operating points

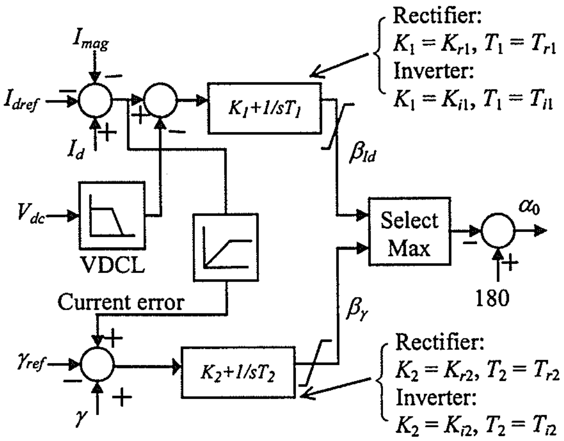
6.4.2 HVDC Control System Design

This example demonstrates the effectiveness of the robust optimization approach in the design of a much larger system. The system under consideration is a monopolar representation of a high voltage direct current (HVDC) transmission system, with a relatively detailed representation of the converters, controls and the sending/receiving end ac networks. The variables to be optimized are the controller gains, which are four in number. In the previous example the performance of the system was optimized to be uniformly good at different operating points. On the other hand, in this example the controls are optimized to provide uniformly good performance with different receiving end ac networks.

Fig. 6.6 shows schematic diagram of an HVDC system and converter controls, which is based on the First CIGRE HVDC benchmark system [27], with network data as shown in Table 6.4. The 12-pulse inverter and the rectifier are each rated at 500 kV and 2 kA. The ac systems on both sides are represented using Thevenin equivalent circuits. The ac system at the rectifier side has a short circuit ratio (SCR) of 2.5.



(a) CIGRE HVDC benchmark model



(b) Converter control system

Fig. 6.6 Schematic diagram of CIGRE HVDC benchmark model

Table 6.4 Data pertinent to the CIGRE HVDC benchmark model

<table>
<tr><td colspan="6">Rectifier side</td></tr>
<tr><td colspan="3">Ac system</td><td colspan="3">Transformers (each)</td></tr>
<tr><td colspan="3">373.3 kV, SCR = 2.5∠84°</td><td colspan="3">600 MVA, 211.3/345 kV, 18%</td></tr>
<tr><td colspan="6">Inverter side</td></tr>
<tr><td colspan="3">Ac system</td><td colspan="3">Transformers (each)</td></tr>
<tr><td colspan="3">218.2 kV, SCR = 2.0∠75°
222.3 kV, SCR = 3.0∠80°
226.3 kV, SCR = 5.0∠85°</td><td colspan="3">600 MVA
206.5/230 kV
18%</td></tr>
<tr><td colspan="6">Filters and fixed capacitors (MVAR) (for both sides)</td></tr>
<tr><td colspan="2">11-th harmonic</td><td colspan="2">13-th harmonic</td><td colspan="2">Fixed capacitors</td></tr>
<tr><td colspan="2">250</td><td colspan="2">250</td><td colspan="2">150</td></tr>
<tr><td colspan="6">Dc link</td></tr>
<tr><td colspan="2">Dc line resistance</td><td colspan="2">Rated dc voltage (rectifier side)</td><td colspan="2">Rated dc current</td></tr>
<tr><td colspan="2">5Ω</td><td colspan="2">500 kV</td><td colspan="2">2 kA</td></tr>
</table>

The nominal SCR on the inverter side is 2.5; however, in this study, three different SCR values are used corresponding to the possible variations in the receiving end network strengths (SCR=2.0 for a "weak" network, SCR=3.0 for a "medium strength" network, and SCR=5.0 for a "strong" network). Such variations in short circuit levels can occur in real HVDC systems for several reasons, such as reconfigurations of the ac network due to the switching of transmission lines and changes in the number of synchronous compensators on the ac busbar.

A. Selection of the Objective Function

The four parameters to be set for robust performance are the proportional gain (K_{r1}) and integral time constant (T_{r1}) on the rectifier side current controller, and proportional gain (K_{i2}) and integral time constant (T_{i2}) of the inverter extinction angle controller.

The objective of the design is to select values of these parameters so that the deviation between the actual dc current I_d and the current order I_{dref} is as small as possible, when changes are made to the current order. The design should be robust, i.e., the performance should be good even with changes in the strength of the receiving end ac network. To meet the above requirements, the objective function in (6.4) is selected, which penalizes differences between I_d and I_{dref}.

$$\begin{aligned} of_i(\mathbf{x}) &= ISE(\mathbf{x}) = \int_{T_0}^{T_F} (1-\frac{I_d}{I_{dref}})^2\,dt \\ \mathbf{x} &= (K_{r1}, T_{r1}, K_{i2}, T_{i2}) \end{aligned} \tag{6.4}$$

The test comprises of a -20% step change in the current order from 1 pu to 0.8 pu, followed by another +20% change to bring the set point back to 1.0 pu.

The form of the aggregate objective function $OF(\mathbf{x})$ shown in Fig. 6.1, permits the assignment of different weights to each of the partial objectives $of_i(\mathbf{x})$. In this example, the weights w_i are selected as shown in (6.5). The reason for this choice is that the system is more susceptible to commutation failure at the lower short circuit ratios, if the current overshoot is too large. Hence the weights (w_1 =1, w_2=1) assigned to the lower short circuit ratio systems with SCR =2.0 and SCR= 3.0 respectively is twice that of the weight ($w_3 = 0.5$) assigned to the higher short circuit system with SCR=5.0.

$$OF(\mathrm{x}) = \underbrace{1.0}_{w_1} \times of_1(\mathrm{x}) + \underbrace{1.0}_{w_2} \times of_2(\mathrm{x}) + \underbrace{0.5}_{w_3} \times of_3(\mathrm{x}) \qquad (6.5)$$

B. Optimization Results

The above form of the objective function causes the optimization to search for parameter sets that are more specifically designed for weak and medium strength ac system, which are naturally more difficult to control.

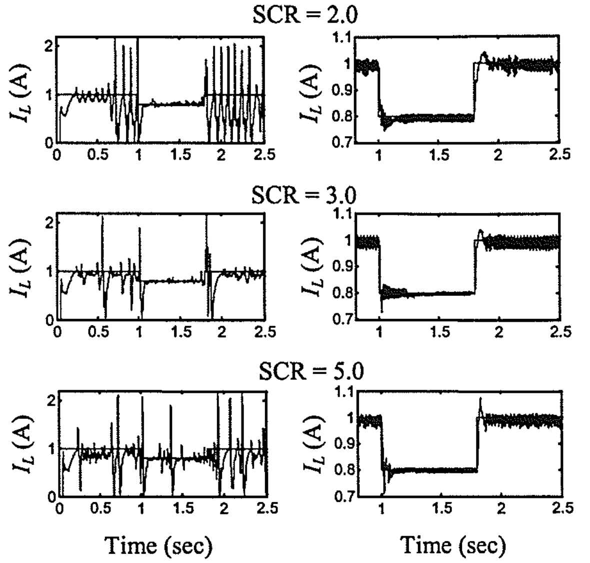
Table 6.5 shows the initial and optimal parameters as well as the corresponding partial and aggregate objective values. The corresponding time-domain simulation plots for the dc current are shown in Fig. 6.7.

Table 6.5 HVDC control system optimization results

<table>
<tr><td colspan="8">Initial guess</td></tr>
<tr><td>K_{r1}</td><td>T_{r1}</td><td>K_{i2}</td><td>T_{i2}</td><td>of_1</td><td>of_2</td><td>of_3</td><td>OF</td></tr>
<tr><td>8.1</td><td>0.01</td><td>2.5</td><td>0.04</td><td>453</td><td>148</td><td>324</td><td>763</td></tr>
<tr><td colspan="8">Optimized results ($w_1 = w_2 = 1$, $w_3 = 0.5$)</td></tr>
<tr><td>K_{r1}</td><td>T_{r1}</td><td>K_{i2}</td><td>T_{i2}</td><td>of_1</td><td>of_2</td><td>of_3</td><td>OF</td></tr>
<tr><td>15.92</td><td>0.0091</td><td>0.215</td><td>0.075</td><td>1.56</td><td>1.2</td><td>1.36</td><td>3.44</td></tr>
</table>

The pre-optimized simulations show an extremely poor response with repeated occurrences of commutation failure. The converged parameter set demonstrates a significantly improved response. This is also quantitatively evident from the objective function tabulations in Table 6.5. The pre-optimization $OF = 763$ is reduced to 3.44.

It is interesting to compare this result with the parameters that are obtained when the weights in (6.5) are all selected to be equal. The optimized parameters, together with the numerical OF values are shown in Table 6.6. It can be seen that the partial objective function of_3 (corresponding to SCR=5) is smaller (1.28) in comparison with the value (1.36) reported in Table 6.5. On the other hand, the values for of_1 and of_2 are now larger.



(a) Initial parameter set (b) Optimized parameter set

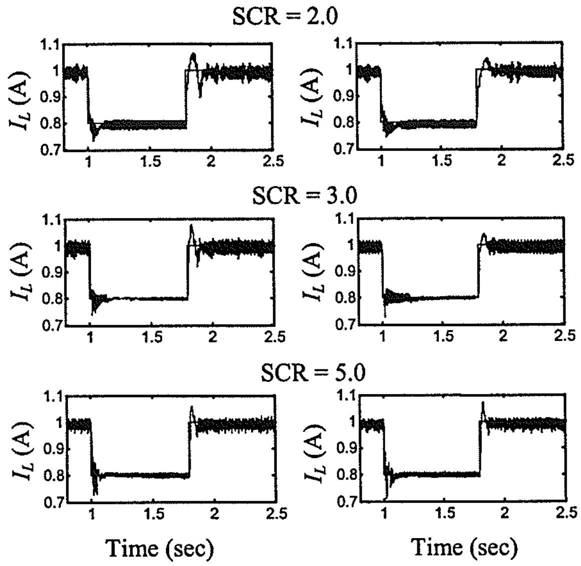
Fig. 6.7 Dc current dynamic response (with unequal weights)

This means that imposing a more stringent weight on of_3, does indeed improve the response for this operating condition; however it does so at the expense of the response at the other two conditions. Robust optimization is a compromise between the competing partial objectives. Hence it becomes important to promote those objectives which are vital while assigning smaller relative weights to the non-critical objectives. As argued above, the vulnerability of the system for the SCR=5 condition was expected to be less severe in comparison with the smaller SCR systems; therefore a smaller weight was selected for w_3.

Table 6.6 Optimization results with equal weights

Optimized results ($w_1 = w_2 = w_3 = 1.0$)							
K_{r1}	T_{r1}	K_{i2}	T_{i2}	of_1	of_2	of_3	OF
14.54	0.0103	0.23	0.031	1.65	1.44	1.28	4.37

The above quantitative results can also be visualized with the aid of Fig. 6.8, which shows the simulation results with optimized parameter settings obtained using equal weights side by side with the earlier results (presented in Fig. 6.7) obtained using unequal weights. It is evident that the dc current for SCR=2.0,3.0 shows larger deviations from the reference for the equal weights, as compared with the waveforms with unequal weights. On the other hand, the results for the SCR=5 case are closer with the equal weights.



(a) Optimized parameter set with equal weights

(b) Optimized parameter set with unequal weights

Fig. 6.8 Optimized system performance with equal and unequal weights

C. Discussion of the Results

This section demonstrated that the design of robust systems is essentially a compromise between competing performance measures. Assignment of weights to these factors before aggregation into the overall objective function shows their relative importance and determines their relative strength in diverting the design towards measures of higher importance (denoted by higher weights). It is therefore important to decide on the relative importance of different operating points or system conditions before assigning weights so that the final aggregate OF is a balanced combination of all system conditions.

The same argument can be repeated for the design of partial (individual) objective functions when they are composed of a number of performance indices. The weights assigned to individual measures show their relative significance and therefore they should be carefully chosen to avoid unnecessarily overemphasized factors.

Chapter 7

Application of Gradient-Based Optimization Algorithms

7.1 Chapter Overview

In chapter four, general principles of optimization algorithms were reviewed and it was noted that they are divided into two categories based on the type of information they need to generate new candidate points. It was also mentioned that in gradient-based optimization methods first or higher order derivatives are required as well as objective function evaluations.

Consideration of the gradient-based methods was deferred to this chapter, because computational complexities associated with numerical implementation of these methods are usually quite high and therefore they are not usually prime choices for being interfaced with transient simulation programs.

This chapter deals with the methods and techniques required for this purpose. It will be shown that the process of optimal design using gradient-based optimization algorithms can be divided into two successive parts, namely numerical evaluation of the gradient and determination of the step length. It will be shown that the former has a significant potential for parallel processing, which is the key motivation for the application of these methods on parallel processors.

7.2 Gradient-Based Optimization Algorithms

In gradient-based optimization methods, selection of a new parameter set is carried out based on objective function evaluation at the current parameter set as well as first and/or higher order derivatives. In general, the iteration formula for obtaining the new trial parameter set in gradient-based methods is as follows [9].

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{s}(\mathbf{x}^{(k)}) \tag{7.1}$$

where $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k+1)}$ are current and new parameter sets, respectively. $\mathbf{s}(\mathbf{x}^{(k)})$ is the search direction in the N-dimensional space, and $\alpha^{(k)}$ is the length of the step in that direction. The search direction $\mathbf{s}(\mathbf{x}^{(k)})$ is determined using gradient information at the current point $\mathbf{x}^{(k)}$, and depends on the type of the information required by the algorithm. The most straightforward approach to choose the search direction is to use the direction of the largest descent based on the local information at $\mathbf{x}^{(k)}$, which is the opposite of the gradient of the objective function at $\mathbf{x}^{(k)}$. The iteration formula will therefore become.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)}\nabla f(\mathbf{x}^{(k)}) \tag{7.2}$$

where $\nabla f(\mathbf{x}^{(k)})$ is the gradient of the objective function at $\mathbf{x}^{(k)}$, and it is obtained from the following formula.

$$\nabla f = [\frac{\partial f}{\partial x_1} \;\; \frac{\partial f}{\partial x_2} \;\; \cdots \;\; \frac{\partial f}{\partial x_N}]^T \tag{7.3}$$

The above formula is known as Cauchy optimization method [9]. The algorithm can be stopped when the gradient (derivative) becomes smaller than a pre-specified threshold (ε).

Further examination of (7.2) reveals that the movement towards the optimum becomes slow as the search gets close enough to the optimum. The reason is that the gradient of the objective function is small around the optimum, therefore the new point $\mathbf{x}^{(k+1)}$ will be marginally different from the current point $\mathbf{x}^{(k)}$. Modifications are introduced to the Cauchy method to improve its behavior at regions both far from and close to the optimum. These include use of second order as well as first order derivatives [30]; however due to the lack of an explicit objective function, these methods are not used in this thesis.

The step length parameter $\alpha^{(k)}$ is usually chosen to minimize the objective function at the new point $\mathbf{x}^{(k+1)}$. Should the explicit representation of the objective function in terms of design parameters be available, the parameters $\mathbf{s}(\mathbf{x}^{(k)})$ and $\alpha^{(k)}$ can be obtained using partial derivatives and a single variable optimization, respectively (see below for details). However, in most of the design (optimization) problems, such as those encountered in power system transient simulation, this explicit mathematical representation is not available, and so the only feasible approach is to use numerical techniques to determine search direction and step length. Lack of this vital information was the main reason why non-gradient-based optimization methods were originally chosen in this research.

In the following, the interfacing method developed to link transient simulation with the Cauchy optimization method is presented. The interface developed handles the numerical calculation of derivatives as well as the single variable optimization needed to determine the optimum step length at each iteration.

7.3 Interfacing EMTDC and Cauchy Optimization Method [26]

Fig. 7.1 shows the schematic diagram of the interface between transient simulation and the Cauchy optimization algorithm. As mentioned above, lack of an explicit representation of the objective function in terms of individual design parameters, leaves us with no choice other than numerically finding the partial derivatives and the appropriate step length, as shown in blocks (2) and (3) in the diagram of Fig. 7.1.

In the following two subsections, we will present the numerical algorithm used for gradient calculation as well as the method for determining the step length. Although Fig. 7.1 shows only one explicit objective function evaluation (block (1)), the following discussion reveals that several more intermediate objective function evaluations are also required.

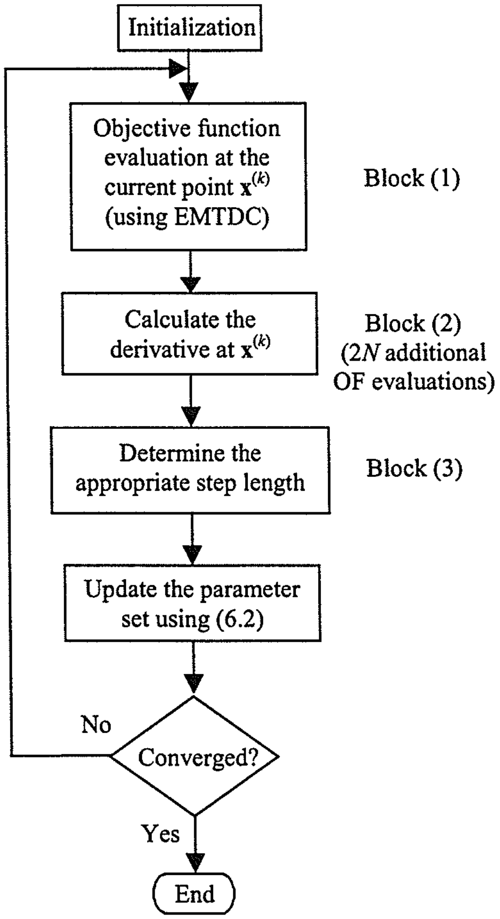

Initialization
Objective function evaluation at the current point $\mathbf{x}^{(k)}$ (using EMTDC)
Block (1)
Calculate the derivative at $\mathbf{x}^{(k)}$
Block (2) (2N additional OF evaluations)
Determine the appropriate step length
Block (3)
Update the parameter set using (6.2)
No
Converged?
Yes
End

Fig. 7.1 Interface between Cauchy optimization and transient simulation

7.3.1 Calculation of Partial Derivatives

The gradient vector given in (7.3) can be formed by evaluating partial derivatives of the objective function at the current point (parameter set). Numerical partial derivatives can be evaluated using the following formula.

$$\frac{\partial f}{\partial x_j}(\mathbf{x}^{(k)}) = \frac{f(\mathbf{x}^{(k)} + \mathbf{h}_j) - f(\mathbf{x}^{(k)} - \mathbf{h}_j)}{2h} \tag{7.4}$$

where h is an adequately small increment, and the vector $\mathbf{h}_j$ is defined as follows.

$$\mathbf{h}_j = \begin{bmatrix} \underbrace{0 \ \cdots \ 0}_{1 \text{ to } j-1} & h & \underbrace{0 \ \cdots \ 0}_{j+1 \text{ to } N} \end{bmatrix}_{1\times N} \tag{7.5}$$

Note that the formula in (7.4) practically yields the average of the right and left partial derivatives of the function at the point $\mathbf{x}^{(k)}$. Technically, the increment h has to be as small as possible; however, to avoid numerical errors, we may choose h to be a fraction of the corresponding $x_j^{(k)}$, e.g. 5% of $x_j^{(k)}$, which of course is not necessarily extremely small and will therefore result in an approximation of the actual partial derivative.

We note that each of the partial derivatives in (7.3) can be calculated using (7.4) and each will require two objective function evaluations (in our case, using PSCAD/EMTDC). Therefore for an N-variable optimization problem, calculation of the gradient will require $2N$ objective function evaluations.

7.3.2 Determination of Appropriate Step Length

The step length factor in (7.2) determines how far the new point (parameter set) will be from the original point (in the direction specified by $\mathbf{s}(\mathbf{x}^{(k)})$). In a minimization problem, it is obviously desirable to have $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$. Since $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)} \nabla f(\mathbf{x}^{(k)})$, one could argue that $f(\mathbf{x}^{(k+1)})$ is a function of $\alpha^{(k)}$, i.e. $f(\mathbf{x}^{(k+1)}) = g(\alpha^{(k)})$. Therefore, the best choice for $\alpha^{(k)}$ is to find it such that $f(\mathbf{x}^{(k+1)})$ is minimized. Normally single variable optimization methods are used to determine $\alpha^{(k)}$ in each iteration. The algorithm used in this work is given below.

The basic idea in this algorithm is to find a value for the current step length so that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$. Since the OF evaluation at the new point is essentially a function of the step length (see above), it is technically possible to find the optimum value of the step length such that the new point has the lowest possible OF evaluation. This however requires a single variable 'optimization' in terms of the step length, which in turn will require several intermediate steps (OF evaluations) before the optimal step length is obtained. Besides most of the single variable optimization methods require a pre-specified interval over which the objective function is unimodal (see section 4.5). Such an interval is not usually known a-priori, and this also leaves little incentive for using a single variable optimization method to find the optimal step length. It is asserted that use of single variable optimization methods is quite possible, but it is not investigated in this research.

Unlike an optimization algorithm, the following algorithm merely tries to find a value for the step length such that the next point has a lower OF evaluation (not necessarily the lowest possible).

Algorithm for determination of an appropriate step length

```
$\alpha^{(k)} = \alpha_0$
$\mathbf{x}_{test1} = \mathbf{x}^{(k)} - \alpha^{(k)}\nabla f(\mathbf{x}^{(k)})$
IF $f(\mathbf{x}_{test1}) < f(\mathbf{x}^{(k)})$
    $\mathbf{x}_{test2} = \mathbf{x}^{(k)} - 1.5\alpha^{(k)}\nabla f(\mathbf{x}^{(k)})$, it is a good direction, so we can increase the step length
    IF $f(\mathbf{x}_{test2}) < f(\mathbf{x}^{(k)})$
        $\alpha^{(k)} = 1.5\alpha^{(k)}$, we choose the elongated step length
    ELSE
```

```
8                 $\alpha^{(k)} = \alpha^{(k)}$, the original step length is good enough
9           END
10      ELSE, the original step length is too large
11          $\mathbf{x}_{test2} = \mathbf{x}^{(k)} - 0.5\alpha^{(k)}\nabla f(\mathbf{x}^{(k)})$
12          IF $f(\mathbf{x}_{test2}) < f(\mathbf{x}^{(k)})$
13                $\alpha^{(k)} = 0.5\alpha^{(k)}$, the halved step length works well
14          ELSE
15                $\alpha^{(k)} = 0.25\alpha^{(k)}$, the shortened step length is still too large
16                GOTO 2
17          END
18      END
```

7.4 Example Cases

7.4.1 A Simple Two-Variable Example

In this section we study the performance of the proposed method in finding the minimum of a simple function of two variables, as given below. Since the mathematical expression of the function is available, its contour plot can be easily generated using mathematical software, such as MATLAB.

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2 + (\sin x_1)^2 x_2^2 \tag{7.6}$$

Note that since the function is explicitly defined, it is possible to find the gradient and the optimal step length analytically, but here we intentionally use the numerical derivatives and the algorithm given above to demonstrate the procedure that will later on be used for objective functions of arbitrary structure.

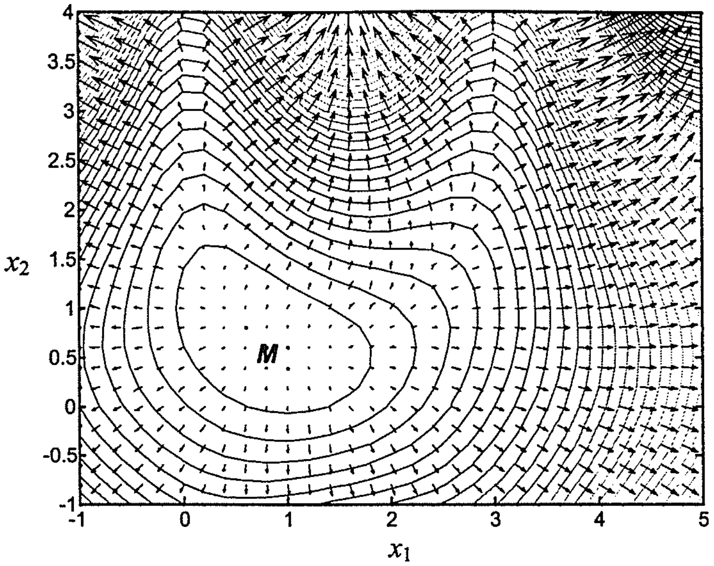


Fig. 7.2 Contour plot of the function in (7.6)

The arrows on the contour plot of the function (Fig. 7.2) show the direction of the gradient at each point. The function attains its minimum of 0.38 at the point M=(0.79,0.69). The optimization method follows the general outline of the Fig. 7.1, except for the fact that objective function evaluation does not require simulation, and so the block (1) in Fig. 7.1 is simply reduced to evaluating (7.6) for a given (x_1,x_2).

The algorithm is executed using the following parameters.

Table 7.1 Parameters used in the algorithm

Starting point	Initial step length (α)	Termination criteria (ε)	Increment (h)
(2,3)	0.1	0.05	2%

Fig. 7.3 shows the first few iterations of the algorithm. It is seen that firstly the numerical gradient of the objective function at (2,3) (Point 1) is calculated, and then an appropriate step is taken at the direction of the steepest descent, which is correctly computed to be in the opposite direction of the local gradient. The objective function evaluation at the starting point $f(2,3)$, is equal to 12.44. Following the first step, a new

point (2.18,2.11) (Point 2) is obtained with an OF evaluation of 5.08. Since the new point has a lower OF evaluation than the starting point, the step length adjustment algorithm takes a larger step in the same direction, which results in a new point (2.72,1.65) (Point 3) with an OF evaluation equal to 3.85, which again is lower than the previous point. This point is adopted as the current base point and gradient is calculated to determine the appropriate direction for the movement starting form this point.

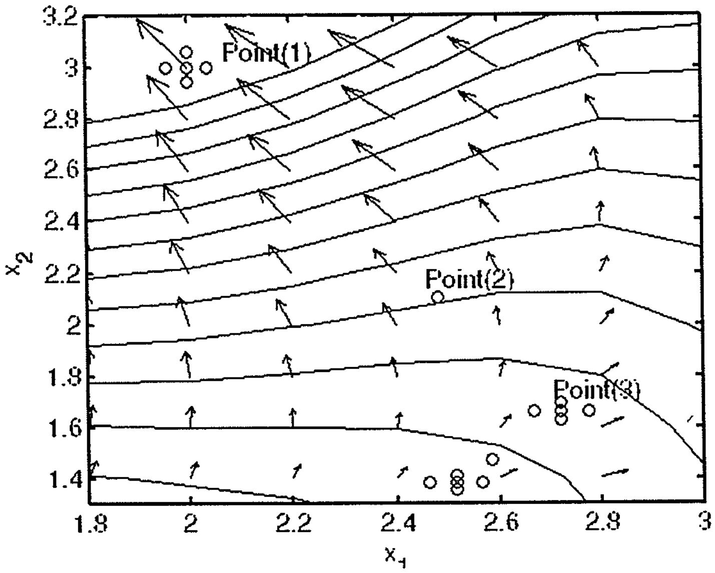


Fig. 7.3 The first few iterations of the Cauchy method

The entire optimization is carried out in 154 objective function evaluations. Note that this number is not the number of iterations required by the gradient-based algorithm, because it also contains all the objective function evaluations that are necessary for gradient evaluation and also for finding the appropriate step length as well.

7.4.2 Control Design for an HVDC System [27]

In this section we present the results of the optimal design of the HVDC control system (see 6.4.2). The design is carried out for a single terminating ac system strength of

SCR = 3.0. Schematic diagram of the system as well as its control system layout is shown in Fig. 6.6 (a) and (b). Table 6.4 summarizes the data for the system.

A. Design Specifications and the Objective Function

The objective of the design is to find four control system parameters, K_{r1} and T_{r1} for the rectifier side current controller, and K_{i2} and T_{i2} for the inverter extinction angle controller, so that the dynamic response of the dc link current around its nominal value becomes as smooth as possible. The objective function given below is used as a measure of the deviation between the actual and the reference dc link currents.

$$ISE = 1000 \int_{0.8}^{3.0} (1 - \frac{I_d}{I_{dref}})^2 \, dt \qquad (7.7)$$

The OF is a function of control system parameters and penalizes any instantaneous difference between these two quantities. Therefore minimizing this OF yields a set of control system parameters for which the actual dc link current is closest to the reference current.

B. Optimization Results

Table 7.2 shows the pre-optimized parameter set as well as the optimized parameter set obtained using the proposed tool. The corresponding dynamic responses of the dc current are shown in Fig. 7.4.

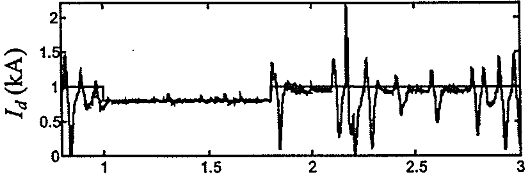
Table 7.2 Optimization results

<table>
<tr><td colspan="5">Initial parameter set</td></tr>
<tr><td>K_{r1}</td><td>T_{r1}</td><td>K_{i2}</td><td>T_{i2}</td><td>ISE</td></tr>
<tr><td>8.1</td><td>0.01</td><td>2.5</td><td>0.04</td><td>124.86</td></tr>
<tr><td colspan="5">Optimized parameter set</td></tr>
<tr><td>K_{r1}</td><td>T_{r1}</td><td>K_{i2}</td><td>T_{i2}</td><td>ISE</td></tr>
<tr><td>1.33</td><td>0.0029</td><td>0.67</td><td>0.027</td><td>0.88</td></tr>
</table>

It is seen that the pre-optimized parameter set results in severe sustained current fluctuations (commutation failure), whereas the optimized parameter set causes a smooth tracking of the current reference with minimal overshoot and excellent steady state match between the actual and the reference current.

The entire design is done in 430 simulation runs. We again compare the design with a conventional MR design for a 4-variable case, which with only 10 steps in each direction, would require 10^4 simulations. As in the previous example, we also note that the 430 simulations in this case is not the actual number of iterations done by the gradient-based method. It contains all the intermediate simulations required for the numerical calculation of the derivatives as well as those carried out for the determination of the appropriate step length at each point.

(a) Initial parameter set



(b) Optimized parameter set

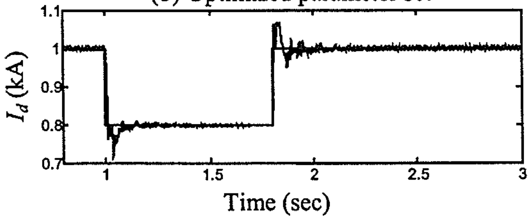


Fig. 7.4 Dynamic response of the system

7.5 Potential for Parallel Processing

Earlier in this thesis, we mentioned that non-gradient-based optimization methods, i.e. those which require only function evaluations, are generally easier to be interfaced with transient simulation programs due to the fact that they do not require any numerical derivatives. However, generally superior performance of the gradient-based algorithms is a strong enough justification to consider them for interfacing too.

The treatment of the subject presented throughout this chapter shows that this perception is indeed valid and these algorithms do perform satisfactorily in practice. However, further examination of the routines used for numerical calculation of derivatives and also determination of appropriate step length reveals that a large number of intermediate simulations are to be carried out to find the gradient as well the appropriate step length merely to enable finding a new parameter set (see equation (7.2)). Although gradient-based algorithms are generally expected to be able to find the optimum in a fewer number of iterations than non-gradient-based algorithms, these intermediate simulations, which are necessitated by the absence of an explicit representation of the objective function in terms of design variables, can severely impact the performance of these algorithms. For example a gradient-based algorithm may be able to find the optimum of a function in handful of iterations as in (7.2), but we note that in a 5-variable optimization problem for instance, each of the iterations will require 10 simulations to find the gradient as well as a number of simulations to determine the step length required.

The combined number of simulations may become so large that they degrade the performance of gradient-based algorithms to below that of non-gradient-based ones. However, it is easy to note that the numerical evaluation of derivatives as presented in (7.4) is an inherently parallel procedure, because it involves independent function evaluations that can be carried out on different processors. Therefore, in an N-dimensional problem, for which $2N$ objective function evaluations are required to calculate the gradient, each of these evaluations can be assigned to a different processor and the final results are sent back to the main processor, which does the job of calculating the partial derivatives and forming the gradient. Exploiting this inherent parallelism can significantly speed up the process of optimization and therefore enhances the design loop by lowering the burden on the main processor.

Chapter 8

Conclusions and Future Work

This thesis dealt with the development of a new tool for the optimal design of complex linear and nonlinear circuits. The new tool is based on the combination of an optimization algorithm with an electromagnetic transient simulation program (PSCAD/EMTDC in this case). The former is used to strategically select candidate parameter sets and steer the design towards optimality, while the latter evaluates the respective objective function by accurately simulating the time domain behavior of the system.

This chapter summarizes the major contributions of the thesis and also identifies some other related areas where further research may be carried out.

8.1 Thesis Contributions

The contributions of the thesis can be categorized into two sections being (1) the development of the new tool and (2) the applications investigated.

8.1.1 Development of the New Tool

The new tool was developed recognizing a significant need for a versatile platform for the optimal design of complex power networks. Modern networks are becoming increasingly complex and the introduction and proliferation of switching devices has intensified this issue even more.

Design in modern power networks usually includes multi-variable, multi-objective problems. Taking into account the complexity of the system, it is nearly impossible in many such cases to obtain an explicit mathematical representation of the design objective function in terms of the design parameters. This limits the application of many traditional design approaches, and leaves the designer with the only choice of recourse to numerical methods.

Conventional methods of Multiple-Run and Random Search are found to be extremely inefficient in terms of computer resources and accuracy of the results when the number of design parameters or complexity of the system increases.

The new design tool developed in this thesis not only improves the accuracy of results but also expedites the design process by orders of magnitude.

The distinguishing feature of the new design tool is the ability of optimally designing a nonlinear system of virtually any complexity *even when no explicit mathematical representation of the interdependence of the design variables and the objective function is available.*

8.1.2 Development of the Robust Optimal Design Tool

Robustness of the optimal design obtained by the developed tool was also investigated in the thesis. It was shown that the nonlinear nature of the power systems causes the optimal design to be dependent on the operating point or condition and therefore the optimal parameter set obtained for a given condition may be sub-optimal or even inappropriate for other operating points or conditions.

A modified interface between transient simulation and optimization algorithms was developed to include robustness into the design procedure. The robust optimal design tool evaluates the performance of the system for a given parameter set over a range of operating points and attempts to optimize the overall performance of the system by minimizing the cumulative objective function. The parameter set so optimized is the best overall performer and retains robustness to the changes in the operating point within the range considered.

The robust optimal design tool also benefits from the main feature of the proposed optimal design scheme in not relying on an explicit representation of the objective function in terms of the design variables.

8.1.3 Applications

The thesis also investigated a fairly large number of applications of the developed tools.

The developed tools were applied to optimally design dc-dc converters under various scenarios such as design of the control system, overall design including the controls and the electrical components, and design including robustness.

Investigations were done into the optimal design of switching patterns for voltage sourced converters to eliminate harmonics from their output voltage while keeping the switching rate as low as possible to limit the associated losses.

The studies included both ideal and non-ideal cases where the dc bus voltage is fixed or has certain amounts of characteristic and non-characteristic harmonic components. While it is possible to analytically formulate and solve the problem under the ideal situations, the analytical solution of the problem for the non-ideal case is virtually impossible to obtain if exact information of the nature of the ripple is not available. The optimal design tool provides a method for the solution of the non-ideal problem where no such information is known. The results surprisingly showed that angles calculated under the idealized assumption of fixed dc bus voltage perform satisfactorily even under dc bus voltage fluctuations as long as the amplitudes of the ripple components are reasonably small.

Other applications such as the design of control systems for HVDC systems were also investigated.

The studies carried out in the thesis reconfirmed that the anticipated improvements of higher accuracy of the results and faster convergence to the optimal parameter set are actually achieved by using the developed tool.

8.2 Recommended Future Directions

The tools and techniques developed in this thesis show great potentials for further research in this area both in the development of more advanced tools and the applications.

8.2.1 Development of a Mathematical Framework for Handling Constraints

Constrained optimization forms an important category of optimal design problems. This arises from the fact most engineering design applications are accompanied by several (usually stringent) constraints imposed by physical or contractual limitations.

As it was mentioned earlier, evaluation of the objective function is not necessarily done through simulation and may also require tests to be carried out on the actual system. These tests are carried out on the system using the parameters specified by the optimization algorithm. The process of optimization, on the other hand, requires several experiments with intermediate candidate points before the optimal setting of parameters is determined. It is worth noting that the process of supervised trial and error conducted by an optimization algorithm may involve parameter sets that are inappropriate for the system and may for example cause instability of the system.

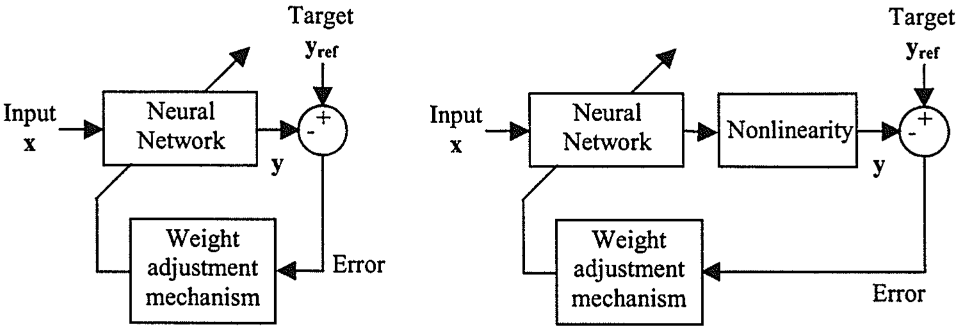
Research can be done into devising methods for constraining variables to regions where certain performance criteria such as the stability are preserved. This may be easy in case where controller gains are to be optimized, but it is generally an open problem for cases where severe nonlinearities exist. The research in this area targets development of the mathematical framework for identifying 'safe' regions of operation and development of methods so that the optimization steers the optimal design process through such regions only.

8.2.2 Application in the Design of Neural Networks (NN)

Neural networks have attracted much attention in various applications in power systems. They can be used to mimic a highly nonlinear mapping using nonlinear neurons interconnected through adjustable weights.

The process of training of a neural network is essentially an optimization problem, in which the weights are adjusted so that the average error between the actual outputs of the network and the target outputs in the training set is minimized. Commonly used training algorithms such as the back propagation method use gradient-based optimization methods and propagate the error between the actual outputs of the NN and the targets through the layers of the network and compute the necessary adjustments to the weights (see Fig. 8.1 (a)).

There are cases however, where the target outputs are not expressed directly as the outputs of the NN. This situation happens for example when the NN is part of a larger system and the measurements are done at the actual outputs of the combined system, which are not necessarily those of the NN (see Fig. 8.1 (b)).



(a) Targets correspond to the actual NN outputs

(b) Targets do not correspond to the actual NN outputs

Fig. 8.1 Training of a neural network

In such cases, the back propagation algorithm has to be modified, as it is not possible to directly calculate the error and propagate it through the layers. Methods such as approximating the corresponding target outputs of the NN or inverting the nonlinear function have been proposed.

The optimal design tool proposed in this thesis can be used in such cases without recourse to approximate methods or trying to invert nonlinear functions. The main idea is to use alternative optimization methods such as those presented in this thesis to find the optimal setting of the weights so that the average error between the actual and target outputs is minimized. Since in the proposed method, no explicit mathematical expression of the objective function in terms of the design parameters is required, it is possible to train the network in its original setting within the combined system.

As an example of an NN-based system for which the above method can be used, one may consider an OPWM-based active filter, whose inputs are the settings of the harmonics and the fundamental voltage and whose outputs are the firing angles necessary to yield the specified spectrum.

NN-based control systems may also be investigated under the same category.

8.2.3 Expansion to Other Optimization Algorithms

The algorithms used in this research are only representatives of a wide range of optimization algorithms. This can be expanded to interface more optimization algorithms with transient simulation in order to provide the user with more choices for specific problem solving requirements.

8.2.4 Implementation on Parallel Processors

As mentioned in previous chapters, genetic algorithms and gradient-based optimization show very promising features for parallel processing. Implementation of these methods on parallel computing platforms extensively facilitates their computational efficiency by reducing the burden on a single processor. Development of the appropriate protocols for the communication between the processors and the main supervisor as well as investigating the potentials of the parallel tools can be pursued as an extension to the current research.

References

[1] N. Mohan, W. P. Robbins, T. M. Undeland, R. Nilssen, O. Mo, "Simulation of power electronic and motion control systems - an overview," *Proceedings of the IEEE*, vol. 82, no. 8, pp. 1287-1302, August 1994.

[2] H. W. Dommel, "Digital computer solution of electromagnetic transients in single- and multiphase networks," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-88, no. 4, pp. 388-399, April 1969.

[3] O. Anaya-Lara, E. Acha, "Modeling and analysis of custom power system by PSCAD/EMTDC," *IEEE Trans. Power Delivery*, vol. 17, no. 1, pp. 266-272, January 2002.

[4] Y. H. Liu, J. Arrillaga, N. R. Watson, "Multi-level voltage sourced conversion by voltage re-injection at six times the fundamental frequency," *IEE Proc.-Electric Power Applications*, vol. 149, no. 3, pp. 201-207, May 2002.

[5] W. Shi, M. R. Iravani, "Effect of HVDC line faults on transient torsional torques of turbine generator shafts," *IEEE Trans. Power Systems*, vol. 9, no. 3, pp. 1457-1464, May 1999.

[6] J. R. Lucas, P. G. McLaren. W. W. L. Keerthipala, R. P. Jayasinghe, "Improved simulation models for current and voltage transformers in relay studies," *IEEE Trans. Power Delivery*, vol. 7, n. 1, pp. 152-159, January 1992.

[7] J. M. Hammersley, D. C. Handscomb, *Monte Carlo Methods*, London: Methuen & Co. Ltd., 1964.

[8] *EMTDC Manual*, Winnipeg: Manitoba HVDC Research Center, 2003.

[9] G. V. Reklaitis, A. Ravindran, K. M. Ragsdell, *Engineering Optimization - Methods and Applications*, New York: Wiley-Interscience, 1983.

[10] D. A. Woodford, A. M. Gole, R. W. Menzies, "Digital simulation of dc links and ac machines," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-102, no. 6, pp. 1616-1623, June 1983.

[11] A. M. Gole, M. Meisingset, "An active filter for use at capacitor commutated HVDC converter," *IEEE Ttrans. Power Delivery*, vol. 16, no. 2, pp. 335-341, April 2001.

[12] R. L. Haupt, S. E. Haupt, *Practical Genetic Algorithms*, New York: Wiley-Interscience, 1998.

[13] K. Y. Lee, M. A. El-Shakrawi (editors), *Tutorial on Modern Heuristic Optimization Techniques with Applications to Power Systems*, IEEE Power Engineering Society 02TP160, 2002.

[14] J. A. Nelder, R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308-313, 1965.

[15] D. S. Weile, E. Michielssen, "Genetic algorithm optimization applied to electromagnetics: a review," *IEEE Trans. Antennas and Propagation*, vol. 45, no. 3, pp. 343-353, March 1997.

[16] N. Mohan, W. P. Robbins, T. M. Undeland, *Power Electronics: Converters, Applications and Design* (2nd edition), New York: John Wiley & Sons, Inc., 1995.

[17] H. S. Patel, R. G. Hoft, "Generalized techniques of harmonic elimination and voltage control in thyristor inverters: Part I: harmonic elimination," *IEEE Trans. Industry Applications*, vol. IA-9, no. 3, pp. 310-317, May/June 1973.

[18] P. N. Enjeti, P. D. Ziogas, J. F. Lindsay, "Programmed PWM techniques to eliminate harmonics: a critical evaluation," *IEEE Trans. Industry Applications*, vol. 26, no. 2, pp. 302-316, March/April 1990.

[19] D. Czarkowski, D. V. Chundnovsky, G. V. Chundnovsky, I. W. Selesnick, "Solving the optimal PWM problem for single-phase inverters," *IEEE Trans. Circuits and Systems - I: Fundamental Theory and Applications*, vol. 49, no. 4, pp. 465-475, April 2002.

[20] J. Y. Lee, Y. Y. Sun, "Adaptive harmonic control in PWM inverters with fluctuating input voltage," *IEEE Trans. Industrial Electronics*, vol. IE-33, no. 1, pp. 92-98, February 1986.

[21] P. N. Enjeti, W. Shireen, "A new technique to reject dc-link voltage ripple for inverters operating on programmed PWM waveforms," *IEEE Trans. Power Electronics*, vol. 7, no. 1, pp. 171-180, January 1992.

[22] A. M. Gole, S. Filizadeh, R. W. Menzies, P. L. Wilson, "Electromagnetic transients simulation as an objective function evaluator for optimization of power system performance," in *Proc. International Conference on Power System Transients-IPST 2003*.

[23] A. M. Gole, S. Filizadeh, R. W. Menzies, P. L. Wilson, Optimization-enabled electromagnetic transient simulation," *IEEE Trans. Power Delivery*, to appear.

[24] S. Filizadeh, A. M. Gole, "Harmonic performance analysis of an OPWM controlled STATCOM in network applications," *IEEE Trans. Power Delivery*, to appear.

[25] A. M. Gole, S. Filizadeh, P. L. Wilson, "Inclusion of robustness into design using optimization-enabled transient simulation," *IEEE Trans. Power Delivery*, under review.

[26] S. Filizadeh, A. M. Gole, "Interfacing gradient-based optimization methods with simulation programs: a novel tool for the design of high performance systems," in *Proc. 18th International Conference on High Performance Computing Systems and Applications-HPCS04.*

[27] M. Sczechtman, T. Wess, C. V. Thio, "First benchmark model for HVDC control studies," *Electra*, no. 135, pp. 54-73, April 1991.

[28] A. M. Gole, I. T. Fernando, G. D. Irwin, O. B. Nayak, "Modeling of power electronic apparatus: additional interpolation issues, in *Proc. International Conference on Power System Transients-IPST 1997.*

[29] C. Z. Janikow, D. St. Clair, "Genetic algorithms: simulating nature's methods of evolving the best design solution," *IEEE Potentials*, vol. 14, no. 1, pp. 31-35, February/March 1995.

[30] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear functions," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431-441, June 1963.

[31] K. F. Man, K. S. Tang, S. Kwong, "Genetic algorithms: concepts and applications," *IEEE Trans. Industrial Electronics*, vol. 43, no. 5, pp. 519-534, October 1996.

[32] A. M. Gole, et al., "Guidelines for modeling power electronics in electric power engineering applications," *IEEE Trans. Power Delivery*, vol. 12, no. 1, pp. 505-514, January 1997.