AUTONOMOUS MULTI-STAGE FLEXIBLE OPTIMAL POWER FLOW

A Dissertation
Presented to
The Academic Faculty

by

Chiyang Zhong

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2019

AUTONOMOUS MULTI-STAGE FLEXIBLE OPTIMAL POWER FLOW

Approved by:

Dr. A. P. Sakis Meliopoulos, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Andy Sun
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Maryam Saeedifard
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Rui Fan
School of Engineering and Computer Science
University of Denver

Dr. Daniel Molzahn
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: [November 07, 2019]

ACKNOWLEDGEMENTS

Over the past four years of my PhD studies, there has been a lot of ups and downs. Although the process is accompanied by stress and failures, there is definitely laughter and successes along the way. I am glad that I started this exciting and rewarding journey in 2015. Now that this wonderful journey has come to an end, I would like to give thanks to the people who have supported me along the way.

First of all, I would like to express my sincere gratitude to my advisor, Dr. A. P. Sakis Meliopoulos, for his guidance and support throughout the years. Dr. Meliopoulos is extremely smart and knowledgeable, as he could answer every question I had during my PhD studies. He is always motivated, passionate and enthusiastic about his work, which encouraged me to overcome many challenges in my research. I have also learned a lot from him on how to work in teams, solve problems, manage my time, deal with stress, etc. I could not imagine having a better advisor and mentor than Dr. Meliopoulos, whose precious lessons will definitely influence my whole life.

A special thank you goes to Dr. George Cokkinides for his help in my research. He is an expert in both power systems and software development. Many of the coding techniques I have learned came from him. Without his help, I would not be able to write codes as efficient and elegant as they are now.

In addition, I would like to thank my committee members, Dr. Maryam Saeedifard, Dr. Andy Sun, Dr. Daniel Molzahn and Dr. Rui Fan for agreeing on serving on my PhD

defense committee. Their constructive and insightful suggestions on my work, as well as their contributions to the completion of my dissertation are greatly appreciated.

My thanks also go to my fellow labmates in the Power Systems Control and Automation Laboratory. I am grateful for the help and support I got from Dr. Liangyi Sun, Dr. Yu Liu and Dr. Bai Cui, as they were the ones I could turn to when I joined the group. I was able to get familiar with the PhD life quickly because of them. I would like to thank Boqi Xie, Jiahao Xie, Maad AlOwaifeer and Gad Monga Ilunga for the projects we worked on together, the discussions we had, and the help I got from them. I will never forget all those late nights that we stayed in the lab to make the deadlines. I would also like to thank Kaiyu Liu, Dr. Yuan Kong, Dr. Zhengkai Wu, Yi Du, Orestis Vasios, Abdullah Alamri, Seyyedmohammadsadegh Vejdan and Dr. Hussain Albinali for their friendship and support. The memories we had together will be treasured forever.

Most of all, no words can express my gratitude to my family, in particular my parents Guoqiang Zhong and Yuning Peng, for their unconditional love. They have always stood by my side and supported the decisions I made. None of my achievements would be possible if it was not for them, to whom this dissertation is dedicated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Optimal power flow (OPF) has always played an important role in the operation of power systems, which are highly nonlinear and complex networks. OPF provides an optimal or suboptimal solution to coordinate the power flows in the electrical network by controlling different devices. To ensure a safe and reliable operation of the system, constraints, such as the generator capacities, line thermal limits and bus voltage limits, are added into the optimization problem. By solving the OPF problem, the objective is achieved and at the same time the constraints imposed on the system are satisfied.

In modern power systems, an increasing number of renewable resources and controllable devices are implemented every year. The conventional OPF that mainly models the generators, lines and loads, as well as some other devices considered due to specific reasons, is not suited for the modern networks. To deal with these new challenges, this proposed research develops a systematic way to formulate and solve the OPF problem autonomously. We introduce two specific problems facing modern power systems: the multi-stage quadratic flexible OPF (MQFOPF) and the security constrained quadratic OPF (SCQOPF).

The presence of renewables and their associated uncertainties have created the need to utilize load flexibility and storage. This need makes the optimization of the system time dependent, i.e. the optimal operation at one instant is linked to the optimal operation at another instant. We are proposing a formulation for the optimization of the power system over a horizon by segmenting the horizon into a number of time periods or stages. This formulation results in a multi-stage OPF. We formulate the OPF as a quadratized problem,

yielding the multi-stage quadratic flexible OPF. The MQFOPF optimizes the operation of the system over multiple stages into the future, where a stage is defined as a time period. The solution of this problem provides the optimal operating points of all the flexible loads and storage resources of the system over the horizon of the MQFOPF defined as the total time spanned by all the stages. Flexible resources can be smart appliances, electric vehicles, thermostatically controlled loads, batteries, etc. The MQFOPF is important in modern power systems as it coordinates and optimizes the operation of all distributed resources in the system, whether they are owned by the utility or the customer.

The objective of the optimal power flow (OPF) problem is to provide set points for power system controls that will optimize a given objective function without violating operating constraints. Furthermore, to ensure a reliable and secure operation, the system is often required to withstand the loss of one or more apparatus depending on the desired security criterion (e.g., N-1 criterion). When such requirement is included in the optimization problem, the problem is termed security constrained OPF (SCOPF).

The SCOPF optimizes the operation of the system for the present conditions as well as for a number of critical contingencies and drastically improves operational security. We propose a new formulation of the SCOPF that models the physics of the modern power system rigorously. The models are cast into objects in a universal syntax, state and control algebraic companion form (SCAQCF), that consists of a number of linear and quadratic equations. The optimization problem is constructed via an object-oriented procedure operating on the device SCAQCF models. The resulting optimization problem consists of linear and quadratic constraints and a quadratic objective function. For this reason, we named it security constraint quadratic optimal power flow (SCQOPF). To ensure reliable

operation of the system, an SCQOPF problem usually includes many contingencies, making it computationally expensive.

To accommodate a huge number of devices, both old and new, in power systems, a physically based object-oriented modeling approach is utilized. A unified general expression is introduced for the device models, based on which the network model is constructed. Together with the objective function, an OPF problem is formed and a tailored sequential linear programming algorithm is used to compute the optimal solution. During the solution process, the constraints are included gradually and the efficient costate method is applied to linearizing the OPF model with respect to the control variables only. Due to object orientation, the whole formulation and solution process of the selected OPF problem is fully autonomous.

The performance of the proposed research method is tested in two different numerical cases. Detailed results are presented, including comparisons with a commercial optimization software. The autonomous OPF approach is proven to be promising and the SLP method outperforms the commercial solver.

CHAPTER 1. INTRODUCTION

This chapter describes the problem statement, research objectives and outline of the thesis. The problem statement briefly discusses the current status of the optimal power flow and the existing shortcomings in this field. The research objectives are described next to address the possible solutions to these problems. Finally, the structure of the thesis is presented.

1.1 Problem Statement

The variability of renewable resources creates challenges for the operation and control of power systems. Renewable energy sources include wind farms, photovoltaic (PV) panels, battery storage units, etc. Figure 1 shows how wind and PV power outputs can vary in a day. These drastic power variations may cause disruption in conventional generation planning, imbalance between supply and demand, and voltage instability issues. The stress imposed on the grid by renewable generation can be mitigated by using the flexibility of customer resources, which include rooftop PV panels, thermostatically controlled loads (TCLs), batteries, smart appliances, etc.

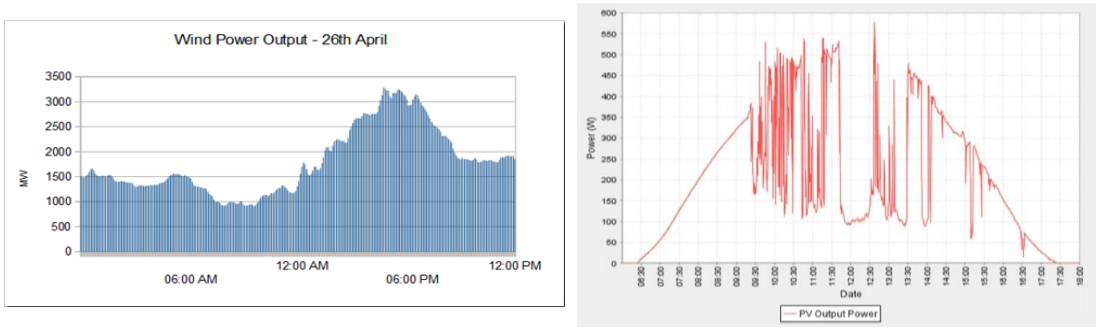

Figure 1 – Wind and PV power outputs.

Optimal power flow (OPF) has always played an important role in coordinating the operation of different devices so that the system remains in a healthy state. It is a term that describes a class of problems that optimizes the operation of power systems with respect to a certain objective function while satisfying a set of physical constraints and engineering limits. The physical constraints are normally defined as the hard constraints imposed by individual devices and materials, for instance, line thermal limits and generator capacities. As for the engineering limits, they are usually the limits that are introduced to achieve certain goals, like the acceptable range of bus voltages.

Traditionally, OPF is used to optimize the power flows in transmission systems. The equalities are defined as real and reactive power flow equations derived from conservation of power at each node in the network. The solution mainly contains the flows of real and reactive powers on each branch. The power injection at each node is also obtained, thus giving information on the generation and demand. Nowadays, the OPF application has been extended to distribution systems. The problem consists of many other types of equations besides the traditional power flow equations, especially when more and more new devices are implemented and taken into account. However, today's OPF formulations are still mostly based on nonlinear power balance equations, which makes the optimization problem extremely complex. In addition, the devices modeled are usually not detailed enough so the power networks are not formulated accurately. This is especially true in distribution networks, where individual devices may have huge impacts on the power grid. When solving the nonlinear and nonconvex OPF problems, many of the existing methods either simplify the problem through approximation or take too long to converge.

In order to deal with the above-mentioned issues in the existing OPF applications, a more accurate OPF formulation and a more efficient solution algorithm is required to accommodate the fast-changing power system.

1.2 Research Objectives

The existence of a variety of new resources calls for a systematic computational framework to integrate the new devices into the power system operations by optimally coordinating their control and operation. Due to the complexity of modern systems with many types of distributed energy resources (DERs), a complete automated process is required to formulate and solve the OPF problem in an autonomous and efficient way based on standardized device modeling.

The proposed research develops a novel approach to optimally coordinate the control of different devices through autonomously formulating and solving OPF problems of the corresponding systems. Both multi-stage quadratic flexible OPF (MQFOPF) and security constrained quadratic OPF (SCQOPF) problems can be solved through the proposed method. Note that a stage is defined as a time period with a user-chosen interval, and “flexible” means that customer controllable resources such as rooftop solar panels, thermostatically controlled loads, smart appliances, etc. are included for the operation of the system, thus customer flexibility is incorporated. The term “quadratic” indicates that the problems are quadratized so that the highest order of nonlinearity is 2.

The formulations of both the MQFOPF and SCQOPF problems build their foundations on a physically based object-oriented modeling approach. Every device is modeled as a mathematical object in a standard syntax called the state and control algebraic

quadratic companion form (SCAQCF), describing the physical laws the device satisfies. When different devices are connected to form a system, a network formulation procedure is performed to construct the network model presented in the same standard form. Combining the network model with a user-defined objective function yields either a multi-stage quadratic flexible OPF model (MQFOPFM) or a security constrained quadratic OPF model (SCQOPFM) is formulated.

The MQFOPFM is constructed by stacking up the network model over multiple stages in the quasi-dynamic domain, which neglects fast electrical transients but slower dynamics like electromagnetic transients are considered. This is because each network model describes the system in only one time period. Due to the existence of slow dynamics and time-dependent elements in the network, various stages are coupled. Therefore, the final MQFOPF problem can only be solved as a whole with the network models from different stages stacked together, which optimizes a multi-stage problem and gives the optimal system controls for each stage into the future horizon.

The SCQOPFM is formulated by combining the base case and contingency network models in the frequency domain, where only a single instant of time is captured. The base case network model is formulated first with all the devices operational, while the contingency network models are copies of the base case network with the corresponding outage device models removed. Note that the frequency domain models are the steady-state models, so there are no dynamics in the SCQOPFM. Solving it provides the optimal controls that achieve the objective and guarantee a safe operation of the system at the same time, even when a selected contingency occurs.

To solve the formulated OPF problem, a tailored sequential linear programming (SLP) method incorporating the costate method is utilized. Only the violated constraints are included in a set of constraints called model constraints, which are considered in the linearized OPF problem. The model constraints are added gradually in every SLP iteration. The whole process from device and network modeling to OPF problem formulation and solution is fully autonomous.

Compared to the existing methods used for OPF, the proposed research has the following merits.

1) Detailed physically based device models form the basis of the OPF formulation, giving more accurate information of the system, which is especially important in distribution systems since they can be impacted by individual devices easily due to smaller network scales. This is an advantage over the existing formulations that mainly use aggregate power injection information at each node.
2) The devices are modeled in an object-oriented way using a standard syntax. This feature allows various DERs to be integrated into the system efficiently. Any new resources can be incorporated into the OPF formulation by simply modeling them in the standard form, thus achieving the flexibility of controlling various devices on the customer side.
3) Object orientation is also applied to the network model by using the same standard modeling syntax, which makes it easy to be incorporated into different OPF applications. For instance, the same network model can be used to construct both the MQFOPF and SCQOPF problems as demonstrated in this thesis.

4) The proposed formulation uses current conservation equations at each node instead of power balance equations in conventional OPF, in which most equations are nonlinear. As a result, most of the equations in the OPF model are linear and the highest nonlinearity is 2. The objective function and inequality constraints have the same characteristics as the nonlinear parts are at most quadratic. This method gives a less complex OPF problem.
5) When solving the OPF problem in SLP, constraints are gradually added to the linearized problem to be solved. In addition, the costate method is used to linearize the OPF model with respect to only the control variables. In addition, the costate method is extremely efficient as the computationally expensive Jacobian matrix used has already been computed in the previous iteration. These mechanisms significantly reduce the size of the problem and increase the algorithm efficiency.
6) The whole OPF formulation and solution process achieves full autonomy. The method is free of human inputs, greatly reducing the possibility of human errors. Once the individual device models and the information of what is measured at various meters and relays in the system are given, the corresponding OPF problem is created autonomously from the output of either a simulator or a state estimator and the solution is subsequentially computed automatically.

1.3 Thesis Outline

The remainder of the thesis is structured as follows.

Chapter 2 presents a literature review of OPF formulation and solution methodologies. Various methods in the existing literature are looked at and they are

compared with each other. The pros and cons of the popular approaches are discussed in detail.

Chapter 3 contains the overview of the proposed research, which includes object-oriented device modeling and network formulation, as well as OPF problem generation and solution. The whole process is fully autonomous.

Chapter 4 describes a physically based object-oriented modeling approach. Any device can be modeled using the SCAQCF standard syntax. A detailed example is presented to demonstrate the device modeling process. When the SCAQCFs of all devices in a system are given, the network model is formulated based on the device models, yielding the network SCAQCF model.

Chapter 5 introduces the automatic formulation of the MQFOPFM in the quasi-dynamic domain. The OPF model is constructed based on a user-defined objective function and a pre-obtained network model in the SCAQCF expression. Two specific objective functions are discussed in detail: network voltage profile optimization and total generation cost minimization. Most of the equality constraints in the MQFOPFM is linear and the rest of the constraints are at most quadratic. Therefore, the MQOPFM is less complex than the conventional OPF problems.

Chapter 6 develops an SLP method that is tailored for solving the MQFOPF problem. In each iteration, the algorithm goes through constraint violation test, convergence check, OPF linearization, linearized problem solving and operating point update. Adding model constraints incrementally and linearizing the problem with respect

to the control variables significantly reduces the size of the OPF problem. The linearization step is described in detail as it is the most important part of the process.

Chapter 7 introduces the SCQOPF problem, which is formulated in the frequency domain for a specific time instant. In addition to the device SCAQCF models in the frequency domain, every device also has an outage model. The SCQOPF problem formulated includes one base case and many imported contingencies, with each being a separate network model. The contingency network models are formed directly from the base case network model with the corresponding outage device models removed. Stacking the base case and contingency network models together with a user-defined objective function yields the SCQOPFM, which is also solved through an SLP approach.

Chapter 8 contains a numerical case used to demonstrate the performance of the proposed research method on the MQFOPF. The test system is a network model that simulates an actual feeder in the Public Service Company of New Mexico (PNM) power network. The voltages at selected nodes in the system are levelized in the MQFOPFM. The MQFOPF optimization results are also listed, with different cases run and compared.

Chapter 9 performs the SCQOPF on the IEEE 57-bus system with several selected contingencies (possible device outages) included. The objective function of the SCQOPFM formulated is the minimization of total generation cost in the base case of the network. The optimal SCQOPF solution is given, together with comparisons between different cases.

Chapter 10 concludes the proposed research and summarizes contributions of this thesis. In addition, possible future work directions are also discussed.

CHAPTER 2. LITERATURE REVIEW

OPF is a term that describes a class of problems that optimizes the operation of power systems with respect to a certain objective function while satisfying a set of physical constraints and engineering limits. In 1962, Carpentier included power flow equations in formulating an economic dispatch problem [1], which is generally considered as the first publication on OPF. The paper gives the optimality conditions for OPF problems including variable bounds based on the Karush-Kuhn-Tucker conditions. Since then, different types of OPF formulation and optimization methods have been introduced in the literature.

This chapter presents a thorough literature review that categorizes the existing literature based on formulation and solution in the field of OPF. Some popular approaches are discussed in detail.

2.1 Problem Formulation

Any OPF problem is composed of variables, objective function and constraints. The general structure of an OPF problem [2] is given by

$$\begin{aligned} \min \quad & J(\mathbf{x},\mathbf{u}) \\ \text{s.t.} \quad & g(\mathbf{x},\mathbf{u}) = 0 \\ & h(\mathbf{x},\mathbf{u}) \le 0 \end{aligned} \tag{1}$$

where $\mathbf{x}$ and $\mathbf{u}$ are the state and control variables of the problem, respectively. Together, they become the operating point $(\mathbf{x},\mathbf{u})$. The objective function in this case is $J(\mathbf{x},\mathbf{u})$, while $g(\mathbf{x},\mathbf{u}) = 0$ are the equality constraints and $h(\mathbf{x},\mathbf{u}) \le 0$ are the inequality

constraints. Note that the variable bounds of the states and controls can be written in the form of the inequality constraints.

The variables represent the operational state of the electrical system. In early OPF formulations, the state variables $\mathbf{x}$ are usually the voltage magnitudes and angles, as well as the real and reactive power injections at network nodes [3]-[6]. This kind of voltage representation is called polar form, which is trigonometric. An alternative representation used is the rectangular form, where real and imaginary parts of the voltages are the system states [7]-[9]. In this case, the power flow equations become quadratic. Besides bus voltages and power injections, other types of state like current injection [9] and battery state of charge (SOC) [10] are seen in recent years. The states are generally continuous variables, but this is not the case for control variables $\mathbf{u}$. Some examples of the continuous controls are generator voltage setpoints and power outputs, while the discrete controls may include transformer tap settings and shunt reactive device switching controls [2]. The control is normally defined by the device or application, so new types of control variables may emerge when more and more new devices are introduced to modern power systems. In many OPF problems, states and controls are not treated distinctively different, as they are all OPF variables.

The objective function defines the main goal of the OPF problems. The most common one is the minimization of total generation cost. Since the cost curves are generally nonlinear, they are often represented by piecewise linear functions [11] or quadratic functions [12]. Other kinds of objective function include minimization of power system losses [13], optimization of voltage profile [14], optimization of post-contingency corrective rescheduling actions [15], etc. Some publications have objective functions with

multi-objectives, which use different weights to prioritize various goals. For example, minimization of the weighted summation of voltage violation, generator reactive power violation and power losses is the objective in [16].

For any optimization problem, there are two categories of constraints: equalities and inequalities. It is the same for OPF problems, in most of which equality constraints are the power flow equations. Every electrical node has both real and reactive power injections, corresponding to two equality constraints. In [9], they become the current injections, which are derived from Kirchhoff's current law (KCL). Due to the integration of renewable sources and energy storage units, other equations such as those that describe converter operation [17] and storage dynamics [18] may also be included in the OPF formulation. As for the inequality constraints, they are the bus voltage bounds, line thermal limits, generator capacities, etc. They are either imposed by material physical properties or defined by engineers to ensure desirable system operation. Hence, the introduction of new devices has brought some new inequality constraints into the picture, an example of which is the battery SOC limits. When solving an OPF problem, not only the objective needs to be achieved, but also all the constraints should be satisfied. If any constraint violation exists after the problem is solved, the OPF solution is said to be infeasible. In many cases, to drive the solution to feasibility, penalties are added to the objective function. Therefore, in this sense, constraints are the most important part of an optimization problem.

The concept of OPF formulation is based upon the three above-mentioned pieces: variables, objective function and constraints. Some existing OPF formulations are described in detail in the following sections.

2.1.1 Nonlinear Formulation

The earliest OPF formulation is nonlinear, as first introduced in [1] and used by other researchers in the 1960s and 1970s [3]-[5]. The nonlinearity is introduced by power balance equations at each node. With respect to the voltage magnitude $|V|$ and angle θ, the real and reactive power injections P and Q at node k are given by

$$\begin{aligned} P_k &= \sum_{j \in S_k} |V_k||V_j|\left(G_{kj}\cos\left(\theta_k - \theta_j\right) + B_{kj}\sin\left(\theta_k - \theta_j\right)\right) \\ Q_k &= \sum_{j \in S_k} |V_k||V_j|\left(G_{kj}\sin\left(\theta_k - \theta_j\right) - B_{kj}\cos\left(\theta_k - \theta_j\right)\right) \end{aligned} \tag{2}$$

where subscript j denotes node j and S_k is the set of other nodes that are connected to node k. G_{kj} and B_{kj} are respectively the branch conductance and susceptance between nodes k and j. The voltages are represented in polar form in (2), resulting in the equations being trigonometric. When the voltages are expressed using rectangular coordinates as shown in [7], (2) becomes

$$\begin{aligned} P_k &= \sum_{j \in S_k} \left(G_{kj}V_{kr}V_{jr} + B_{kj}V_{kr}V_{ji} + G_{kj}V_{ki}V_{ji} - B_{kj}V_{ki}V_{jr}\right) \\ Q_k &= \sum_{j \in S_k} \left(G_{kj}V_{ki}V_{jr} + B_{kj}V_{ki}V_{ji} - G_{kr}V_{ki}V_{ji} + B_{kj}V_{kr}V_{jr}\right) \end{aligned} \tag{3}$$

where r and i in the subscripts denote the real and imaginary parts of the voltages, respectively. Compared to (2), power equations in (3) are quadratic, making the formulation less complicated.

Generally speaking, nonlinear OPF formulation is able to accurately simulate the operation of the corresponding network. However, the resulting problem is nonconvex and hard to solve.

2.1.2 Linear Formulation

Although the nonlinear formulation of OPF accurately describes a system, it is nonconvex and solving it is computationally expensive. In contrast, linear formulation is convex and has a great advantage in computational speed. Direct current (DC) optimal power flow is an example of linear formulation and it has been extensively used in the electric power industry. In DC power flow introduced in [19], voltage magnitudes are assumed to be 1 pu and bus voltage angles are close to 0. Low resistance to reactance ratio is utilized in long transmission lines and losses in the system are neglected. Therefore, with $G_{kj} \approx 0$ and $\sin\left(\theta_k - \theta_j\right) \approx \theta_k - \theta_j$, the real power injection at node k is approximated by

$$P_k = \sum_{j \in S_k} B_{kj}\left(\theta_k - \theta_j\right) \tag{4}$$

In distribution networks, since the lines are much shorter in length, the resistance to reactance ratio is no longer low. Hence, the real and reactive power flows cannot be decoupled. As a result, DC power flow does not hold in distribution systems.

Another kind of OPF linear formulation is the linearization of power flow equations around the current operating point, which is widely seen in literature using SLP techniques to solve OPF problems [20]-[22].

2.1.3 Quadratic Formulation

Quadratic formulation falls between nonlinear formulation and linear formulation as it is made up of quadratic objective function $0.5\mathbf{x}^T \bar{A}\mathbf{x} + \bar{b}^T\mathbf{x}$ and linear constraints [23]. The problem is convex only when matrix $\bar{A}$ is positive semidefinite. Sequential quadratic programming (SQP) is applied when the constraints are linearized around the operating point iteratively [13].

If the OPF constraints are also quadratic like those presented in [24] in addition to the quadratic objective function, the problem is said to be formulated as a quadratic constrained quadratic program, which is suited for many optimization problems in power systems [25]. Because of the computationally difficult nature, this type of problems is usually solved after relaxations are performed.

2.1.4 Mixed Integer Formulation

Due to the existence of discrete control elements in power systems, such as tap changers fitted to transformers and switched capacitor banks, the original nonlinear, linear and quadratic formulations of the OPF problems are not suitable anymore. Therefore, mixed integer linear and mixed integer nonlinear formulations are considered. Paper [26] proposes a two-level decomposition scheme that contains a master problem and a subproblem, with transformer tap settings and volt-ampere reactive sources modeled as discrete elements. The master problem is formulated as a mixed integer linear programming (LP) problem, whereas the subproblem uses a nonlinear formulation. Mixed integer nonlinear formulation is the most accurate representation of a power system, but also the most sophisticated one. Bai and Wei have formulated a security-constrained unit

commitment (SCUC) problem as a mixed integer nonlinear programming problem, incorporating operational and optimal power flow constraints [27].

2.2 Solution Methodology

Solution methodologies are usually selected depending on the formulation of the OPF problem. They can be categorized into deterministic and non-deterministic approaches.

2.2.1 Deterministic Methods

The first efficient solution to OPF was obtained through gradient methods. The reduced gradient (RG) method was applied to OPF by Dommel and Tinney [4], where penalty functions are used for inequality constraints. An extension of the RG is the generalized reduced gradient (GRG) method, which was first applied to OPF in [28]. The GRG procedure repartitions the variables whenever a dependent variable reaches the boundary, thus eliminating penalty terms. The Newton-Raphson method is used for unconstrained optimization problems based on the Taylor's theorem. It solves the nonlinear power flow equations by linearizing them at each successive point reached [3]. To avoid having to compute the inverse of the Hessian matrix, which is computational expensive, quasi-Newton methods were developed to approximate the Hessian iteratively [29], but they are rarely used today due to the increase in computational power of computers. Convexity is required for global convergence of Newton-Raphson and quasi-Newton methods. When OPF is formulated as a linear program like DC optimal power flow, simplex method can be directly applied. Robust and easy to implement, simplex method performs well on LP problems by searching through the extreme points of the polytopes.

When the nonlinear constraints of an OPF problem are linearized with respect to the operating point successively, SLP or SQP are applied depending on whether the objective function is linear or quadratic, respectively. In each iteration, a linear program or a quadratic program around the current operating point is optimized, thus giving the operating point for the next iteration. The convergence is achieved at some point depending on the criteria set, but oscillations may be observed near the optimal solution [30]. However, the optimum is not guaranteed to be the optimal solution of the original nonlinear problem due to linearization errors. SLP can still be implemented to a problem with quadratic objectives by linearizing the objective function [31]. Nine years after the introduction of the first interior point method (IPM) for linear programs in 1984, this projective scaling algorithm made its debut in the OPF field in [32]. IPM converges faster than simplex method in large-scale LP problems, so it has been used to replace simplex method to solve the linearized program in each iteration of SLP. The use of IPM was extended to SQP in [33] and some other enhancements were evaluated in [34].

Due to the nonlinear and non-convex nature of accurate OPF formulations, new relaxation techniques have been adopted in recent years, such as semidefinite programming (SDP) and second-order cone programming (SOCP). An SDP relaxation is seen in [25] for QCQP problems and an SDP relaxed SCUC problem solved by IPM is developed in [27]. Lavaei and Low have proposed an SDP optimization approach and provided the conditions that guarantee a zero duality gap for OPF problems [35]. Literature [36] concludes that SOCP relaxation is faster and more reliable than SDP relaxation while claiming that quadratic convex relaxation is even stronger. These convex relaxation methods introduce

approximations for the non-convex OPF problems, which makes it difficult to obtain the operating condition of the actual power system.

2.2.2 Non-Deterministic Methods

When using deterministic methods, the results produced by every run of the algorithm are the same. There also exist a number of non-deterministic optimization approaches, where the solutions may differ from run to run. Some of these methods are described as follows. Paper [16] uses genetic algorithm (GA), which is a type of evolutionary algorithm (EA), for optimal reactive power planning. Inspired by biological evolution, EAs usually go through initialization, iterations of selection including crossover and mutation, and termination to find the solution. The fitness of individual solution in a population is evaluated every iteration and the least-fit individuals are weeded out during the process. Besides GA, evolutionary programming [37] and differential evolution [38] also belong to the EA category. Swarm intelligence algorithms such as artificial bee colony algorithm [39], particle swarm optimization (PSO) [40], ant colony optimization [41], etc. have also been applied to solving OPF problems. These methods mimic the collective behaviors of decentralized and self-organized biological systems. Recently, machine learning (ML) has drawn much attention in academia, including in the OPF field of research. The idea is to give computers the ability to learn with large data sets. The computers are expected to find patterns and improve the performance on specific tasks using big data. ML is a subset of artificial intelligence and one of the tools used is artificial neural network (ANN), which is based on the operation of human brain. Similar to a biological neural network, elements in ANN are called artificial neurons. When a neuron receives an input, the neuron is activated by changing its internal state accordingly, and

then an output is produced by the neuron based on the input and the internal state. In [42], ANN is used to model stability and security constraints in OPF problems. Simulated annealing was proposed in 1983 as a nature-inspired metaheuristic and applied to OPF by Hsiao [43]. Another method belonging to this category is cuckoo search (CS) [44], in which a solution is represented by an egg in a nest. In every iteration, given that the total number of nests is constant, each cuckoo lays one egg in a random nest and the best nests with high-quality eggs are selected for the next iteration, while the worst nests are thrown away with respect to a probability. PSO and CS methods for unbalanced distribution system optimization are compared in [45]. There are also hybrid algorithms that combines both deterministic and non-deterministic methods. For instance, SQP and differential evolution algorithms are combined in [46] for OPF solving. The metaheuristics described are suitable for all types of OPF problems, including nonconvex ones with complicated constraints.

Most non-deterministic methods are able to break away from local optima and better solutions may be reached compared to deterministic methods given sufficient time. However, the performance of metaheuristics is highly dependent on fine parameter tuning and suffers from scalability issues. In addition, the same initial conditions for the algorithms do not guarantee the same optimization results. As a result, these methods are not presently competitive for the solution of OPF problems.

2.3 Summary

In general, the OPF formulations in most existing papers mainly use power flow equations represented by voltages and currents in either rectangular or polar form, causing the OPF equalities to be mostly quadratic or nonlinear with trigonometric functions,

respectively. Using KCL equations at each node [24] significantly reduces the complexity of the formulation by making most of the OPF equalities linear. Moreover, the device models used in the existing OPF models are not accurate enough. For example, the energy storage system in [47] is only modeled with charging power, discharging power and state of charge, neglecting the possible circuit inside that may limit its performance. Detailed physically based device modeling is important in coordinating the controls of various distributed generation in a complex system. In addition, no standardization of device modeling is seen in the literature, which makes it difficult to incorporate various DERs and solve the OPF problem efficiently in an autonomous manner without any human involvement during the process. Last but not least, many of the OPF solution methods use convex relaxations or metaheuristic approaches, which results in an approximation of the actual problem or a computationally expensive solution, respectively. Better techniques are needed to solve the OPF problems efficiently and accurately.

CHAPTER 3. PROPOSED RESEARCH OVERVIEW

In this dissertation, an autonomous formulation and solution method for object-oriented based OPF problems is developed. This chapter goes through the overview of the research method.

3.1 Method Overview

The overview of the proposed research is shown in Figure 2. The method starts from device modeling and then to network modeling. Together with a pre-defined objective function, the users can choose to construct either an MQFOPFM or an SCQOPFM. After the selected OPF problem is solved, the corresponding solution is obtained.

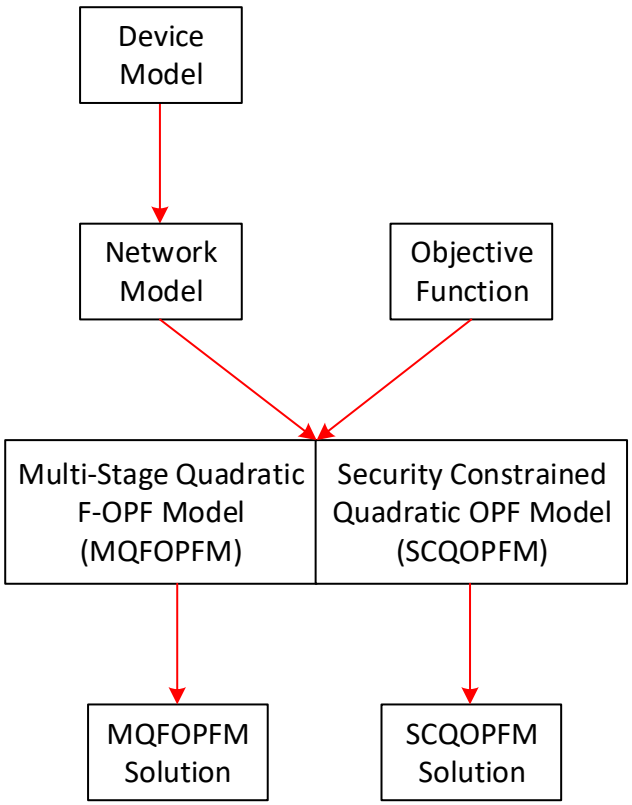

Figure 2 – Overview of the proposed research.

A physically based object-oriented modeling approach is used to simulate device and network operations [48]. Any device can be presented in the SCAQCF standard modeling syntax as long as its physics are accurately written in mathematical equations. Quadratization and quadratic integration are involved during the process. Quadratization not only makes the highest order of terms 2, but also converts discrete variables continuous. The resulting models have the same form, including states, controls, equations and functional constraints. Through variables are defined and they serve as interface media that connect one device to another. The corresponding through equations are linear, while all nonlinear terms are converted into quadratic terms and grouped together as device internal equations. As a result, given all the interconnected devices in a network, the network model is automatically formulated, also presented in the SCAQCF expression. The standard modeling syntax achieves object orientation, which allows the models to be manipulated the same way without any knowledge of the model details.

The device and network models are the foundation of the OPF problem in the proposed formulation process. The network model directly becomes the equality and inequality constraints of the optimization problem. With a user-selected objective function defined, users are able to choose from two OPF applications: MQFOPFM and SCQOPFM. The MQFOPFM is created in the quasi-dynamic domain where fast electrical transients are neglected, while the SCQOPFM is formed in the frequency domain where no transient is considered as only one single time instant is concerned. When the MQFOPFM is formed, the network model is stacked up over multiple time stages. The past history vectors in the network SCAQCF model link different stages together. As for the SCQOPFM, the base case and contingency network models are stacked together. The base case network model

is solved first and its power flow solution is used to solve for the initial operating conditions of the imported contingencies to be included in the SCQOPF. Each contingency network model is generated by removing the corresponding outage devices from the base case network model. It has its own set of states, equations and constraints, while there is only one set of controls shared among all networks. Due to the way the SCAQCF syntax is defined, Kirchhoff's current law (KCL) is applied at every node in an electrical network when the network model is constructed. Thus, most of the equations in the network model are linear KCL equations, while rest of the equations are quadratic. Hence, no matter which application is selected, the proposed OPF formulation gives a less complicated problem to solve compared to the conventional formulation.

The MQFOPFM and SCQOPFM formulated are solved by tailored SLP methods. Initial operating point and past history vectors at the first stage are required for solving the MQFOPFM, while solving the SCQOPFM needs the initial operating point of each network model, including the base case and contingencies. In every SLP iteration, model constraints are first defined. The violated constraints that are not in the model constraint set are added as new model constraints, which are considered in the linearized OPF problem to be solved. Note that each network model in the SCQOPFM has its own model constraints. Due to the gradual inclusion of model constraints, the linearized problem size is the smallest possible in each iteration. The next step is the OPF linearization, in which the costate method is used to linearize the objective function and model constraints with respect to only the control variables. Hence, the states are eliminated from the linearized OPF model and the size of the problem to be solved is further reduced significantly. During the linearization process, inversing the Jacobian matrix is the most time-consuming step,

but it has already been computed in the previous iteration. Except for that, only a few additions and multiplications are required. As a result, the costate method is extremely efficient. In either OPF case, the linearized problem is solved through an LP solver. The results are the control movements that are used to update the controls. Then, the states are computed via Newton's method, thus obtaining the new operating point. The SLP algorithm proceeds to the next iteration after the maximum control movement limits are adjusted based on the change in the OPF objective function value. The new states and controls are substituted into all the constraints to check for violations. The SLP convergence is determined following the model constraint defining process. The SLP algorithm converges only when the linearized problem gives feasible solution and all constraints satisfied. The optimal solution obtained is the control movements that drive the system to the operating condition that achieves the OPF objective while satisfying all the constraints. Note that for the SCQOPFM, the linearization and state update procedures are performed network by network iteratively using the corresponding Jacobian matrix.

In general, it is expected that the user selects the objective function and chooses which OPF problem to formulate and solve, the MQFOPFM or the SCQOPFM. No matter which problem is chosen, the process starts from device and network modeling to the corresponding OPF formulation and solution. The whole process is fully autonomous.

3.2 Summary

An overview of the proposed autonomous OPF method is provided in this chapter. The main advantages of the research proposed is given as follows. First, the OPF problems formulated are more accurate than the conventional ones as the devices are modeled

physically in detail. Second, due to object orientation, any new DER can be easily integrated into the OPF formulation. Third, most equations in the proposed OPF models are linear while the rest are at most quadratic, so the they are less complex and easier to solve. Fourth, constraints are gradually added and the costate method is used during the OPF solution process to reduce the size of the problem to be solved and increase algorithm efficiency. Last but not least, the proposed method autonomously formulates and solves the OPF problems, thus reducing possible human errors.

CHAPTER 4. PHYSICALLY BASED OBJECT-ORIENTED MODELING

Physically based object-oriented modeling is the foundation supporting the autonomous formulation and solution of any multi-stage quadratic flexible optimal power flow problem. It is also the starting point of the whole process, including device modeling and network formation. A standard syntax is utilized to represent the device models based on their physical properties and subsequently form the network model as general mathematical objects. By doing this, each object is accurately constructed and manipulated the same way regardless of its characteristics. Everything is treated as a black box, thus achieving object orientation.

This chapter describes how the devices are modeled and subsequently how the network model is formed based on the device models. Details of the standard syntax used for modeling is provided.

4.1 Device Modeling

The first step of the MQFOPF autonomous process is the modeling of devices. For every device, based on its physical properties, the device compact model is constructed. Since the compact model may contain many high order terms, which complicates the simulation process, quadratization is introduced. If any term has an order of 3 or above, new variables are introduced to bring down its order to 2, hence making the highest order of terms quadratic [49]. After quadratization, the state and control quadratized device model (SCQDM) is obtained. In order to simulate the operation of the device, quadratic

integration is then applied, yielding the state and control algebraic quadratic companion form (SCAQCF), which is the standard syntax for object orientation. Every device considered in the MQFOPF process is modeled in the SCAQCF syntax. The whole device modeling process is presented in Figure 3.

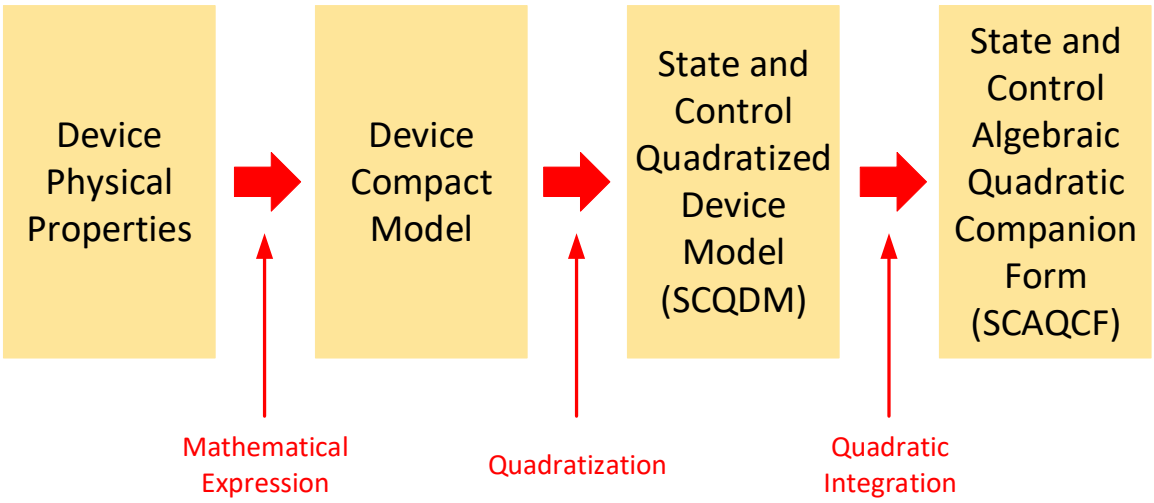

Figure 3 – Device modeling process.

4.1.1 Device Compact Model

Any device can be described mathematically using a set of equations and inequalities. The device compact model is the accurate mathematical description of an actual device according to the laws of physics it obeys. The general expression of the compact model of a device operating at time t is given by

$$\begin{aligned} \mathbf{i}(t) &= f_1\left(\mathbf{x}(t),\mathbf{u}(t)\right) \\ 0 &= f_2\left(\mathbf{x}(t),\mathbf{u}(t)\right) \\ h\left(\mathbf{x}(t),\mathbf{u}(t)\right) &\le 0 \\ \mathbf{u}_{\text{cmin}} \le \mathbf{u}(t) &\le \mathbf{u}_{\text{cmax}} \end{aligned} \tag{5}$$

where $\mathbf{i}$, $\mathbf{x}$ and $\mathbf{u}$ are the vectors of through variables, state variables and control variables, respectively. f_1 describes the through equations and f_2 describes the internal equations of the device. The through equations are also called the terminal equations of the device. h is a set of functions that represent the device functional constraints, which are defined as $h \le 0$. The control lower and upper bounds are respectively $\mathbf{u}_{\text{cmin}}$ and $\mathbf{u}_{\text{cmax}}$.

Since (5) is a generalized model for any type of device, it is not limited to electrical devices, in which the through variables are typically currents. For thermal and mechanical devices, their through variables are heat flows and torques respectively. Therefore, the modeling of any device starts from the construction of its device compact model.

4.1.2 State and Control Quadratized Device Model

Although the device compact model may contain high order polynomials or even trigonometric and exponential terms, quadratization is able to convert (5) into a form with the highest order being 2, which is called the state and control quadratized device model (SCQDM). During the quadratization process, new state variables are introduced. For example, applying quadratization to term x_1^n with $n \ge 3$ yields

$$x_1^n = \begin{cases} x_2^2, & \text{if } n \text{ is even} \\ x_3 x_4, & \text{if } n \text{ is odd} \end{cases} \tag{6}$$

where x_2, x_3 and x_4 are new variables introduced for quadratization. They have the following relationships with x_1.

$$
\begin{aligned}
x_2 &= x_1^{\frac{n}{2}} \\
x_3 &= x_1^{\frac{n+1}{2}} \\
x_4 &= x_1^{\frac{n-1}{2}}
\end{aligned}
\tag{7}
$$

The process of (6) is repeated until all terms become either linear or quadratic. Similarly, (6) can be applied to trigonometric and exponential terms when they are approximated as polynomials with a certain number of terms depending on the required accuracy.

Converting discrete binary variables into continuous variables is another advantage of quadratization. For any binary variable x_b, a new state y is introduced so that $y^2 = x_b\left(1-x_b\right)$. By adding y^2 with a large weight to the OPF objective function to be minimized, y^2 is driven to 0, thus making the value of x_b either 0 or 1. This method is used on devices with binary variables like switched capacitor banks.

In the model, everything is in metric unit. For electrical devices, the voltage and current variables are represented in the rectangular form, including real and imaginary parts. The standard SCQDM is written as

$$
\begin{aligned}
&\mathbf{i}(t) = Y_{x1}\mathbf{x}(t) + Y_{u1}\mathbf{u}(t) + D_1\frac{d\mathbf{x}(t)}{dt} + C_1 \\
&0 = Y_{x2}\mathbf{x}(t) + Y_{u2}\mathbf{u}(t) + D_2\frac{d\mathbf{x}(t)}{dt} + C_2 \\
&0 = Y_{x3}\mathbf{x}(t) + Y_{u3}\mathbf{u}(t) + \left\{\begin{array}{c}\vdots \\ \mathbf{x}(t)^T F_{x3}^k \mathbf{x}(t) \\ \vdots\end{array}\right\} + \left\{\begin{array}{c}\vdots \\ \mathbf{u}(t)^T F_{u3}^k \mathbf{u}(t) \\ \vdots\end{array}\right\} + \left\{\begin{array}{c}\vdots \\ \mathbf{u}(t)^T F_{ux3}^k \mathbf{x}(t) \\ \vdots\end{array}\right\} + C_3 \\
&h = Y_{cx}\mathbf{x}(t) + Y_{cu}\mathbf{u}(t) + \left\{\begin{array}{c}\vdots \\ \mathbf{x}(t)^T F_{cx}^k \mathbf{x}(t) \\ \vdots\end{array}\right\} + \left\{\begin{array}{c}\vdots \\ \mathbf{u}(t)^T F_{cu}^k \mathbf{u}(t) \\ \vdots\end{array}\right\} + \left\{\begin{array}{c}\vdots \\ \mathbf{u}(t)^T F_{cux}^k \mathbf{x}(t) \\ \vdots\end{array}\right\} + C_c \le 0 \\
&\mathbf{u}_{\text{cmin}} \le \mathbf{u}(t) \le \mathbf{u}_{\text{cmax}}
\end{aligned}
\tag{8}
$$

where matrices Y and D contain the linear and differential terms associated with the device states and controls, respectively. Quadratic coefficients are stored in matrices F, while constant terms are contained in vectors C. Subscripts x and u denote whether the coefficients correspond to states or controls, or even both for ux. 1, 2 and 3 in the subscripts denote equation sets 1, 2 and 3, respectively, while c corresponds to constraints. Superscript k of matrices F corresponds to equation k. For the control limits in (8), the lower and upper physical control bounds respectively are $\mathbf{u}_{\text{cmin}}$ and $\mathbf{u}_{\text{cmax}}$. In addition, each device model also has a set of algorithmic constraints $-\mathbf{u}_{\text{cllim}} \le \mathbf{d}(t) \le \mathbf{u}_{\text{cllim}}$, where $\mathbf{d}$ is defined as the vector of control movements to keep the linearization errors below a certain threshold in the SLP algorithm and $\mathbf{u}_{\text{cllim}}$ contains the initial non-negative maximum control excursion limits (step sizes).

The SCQDM models the operation of the corresponding device at time t using 3 sets of equations, functional constraints and control limits. Equation set 1 is the through

equations, while equation sets 2 and 3 are the internal equations. All quadratic terms are only present in equation set 3.

4.1.3 State and Control Algebraic Quadratic Companion Form

To simulate the operation of a device, quadratic integration is applied to the first two equation sets of the SCQDM to acquire the SCAQCF, which is the standard syntax for device modeling.

Quadratic integration is an integration method that assumes a function varies quadratically between two consecutive time steps. Different from the widely used trapezoidal integration, quadratic integration has been proven to be more accurate and numerical stable [50]. Comparison of the two integration methods are shown in Figure 4.

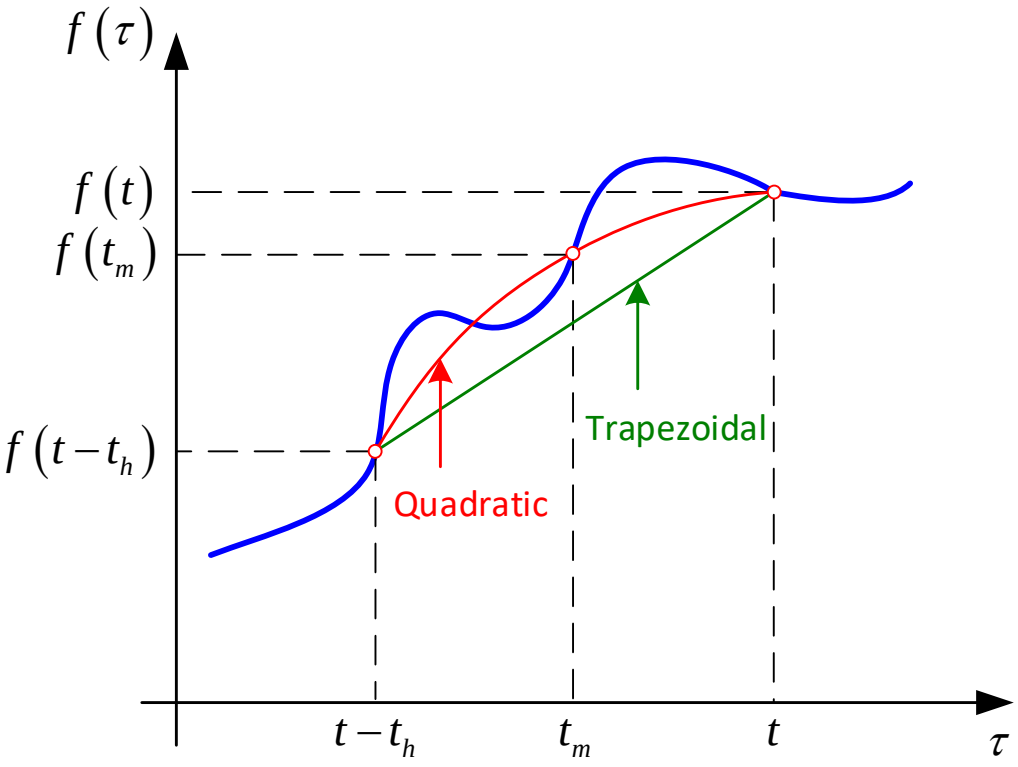

Figure 4 – Quadratic and trapezoidal integration methods.

For any function $f(\tau)$, its curve between two time steps $t-t_h$ and t can be approximated either by a line or a quadratic curve, where t_h is the time step size. As shown in Figure 4, trapezoidal integration assumes that $f(\tau)$ varies linearly between $t-t_h$ and t, which gives

$$\int_{t-t_h}^{t} f(\tau)d\tau = \frac{t_h}{2} f(t) + \frac{t_h}{2} f(t-t_h) \tag{9}$$

In equation (9), only $f(t)$ and $f(t-t_h)$ are needed to approximate the integration of $f(\tau)$ from $t-t_h$ to t. As for quadratic integration, $f(\tau)$ varying quadratically between the two points is assumed. As a result, a third point is required during the integration process. In Figure 4, the midpoint t_m between time steps $t-t_h$ and t is selected, yielding

$$\begin{aligned}
\int_{t-t_h}^{t} f(\tau)d\tau &= \frac{t_h}{6} f(t) + \frac{2t_h}{3} f(t_m) + \frac{t_h}{6} f(t-t_h) \\
\int_{t-t_h}^{t_m} f(\tau)d\tau &= -\frac{t_h}{24} f(t) + \frac{t_h}{3} f(t_m) + \frac{5t_h}{24} f(t-t_h)
\end{aligned} \tag{10}$$

As shown in (10), there are two equations for quadratic integration. One is the integration of $f(\tau)$ from $t-t_h$ to t, while the other one integrates $f(\tau)$ from $t-t_m$ to t. Note that

$$t_m = t - \frac{t_h}{2} \tag{11}$$

It can be noticed from (10) that besides points $f(t)$ and $f(t-t_h)$, $f(t_m)$ is also required during the integration process. $f(t-t_h)$ is called the past history term in this case.

The general expression for the SCAQCF syntax obtained after the quadratic integration procedure is given by

$$\begin{Bmatrix} \mathbf{i}(t) \\ 0 \\ 0 \\ \mathbf{i}(t_m) \\ 0 \\ 0 \end{Bmatrix} = Y_x\mathbf{x} + Y_u\mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T F_{xx}^k \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{uu}^k \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{ux}^k \mathbf{x} \\ \vdots \end{Bmatrix} - B$$

$$B = -N_x\mathbf{x}(t-t_h) - N_u\mathbf{u}(t-t_h) - M\mathbf{i}(t-t_h) - K \tag{12}$$

$$h = Y_{fx}\mathbf{x} + Y_{fu}\mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T F_{fxx}^k \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{fuu}^k \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{fux}^k \mathbf{x} \\ \vdots \end{Bmatrix} + C_f \le 0$$

$$\mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max}$$

where B is the past history vector that stores the device model information from the previous time step $t - t_h$. Matrices N and M contain linear coefficients of the variables at $t - t_h$, while vector K has the constant terms in the device equations. Subscript f in (12) denotes the functional constraints. In the SCAQCF model, the algorithmic constraints become $-\mathbf{u}_{\text{llim}} \le \mathbf{d} \le \mathbf{u}_{\text{llim}}$. Including variables at time steps t and t_m, vectors $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{d}$ are defined as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}(t_m) \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{u}(t_m) \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} \mathbf{d}(t) \\ \mathbf{d}(t_m) \end{bmatrix} \tag{13}$$

Via quadratic integration, the matrices and vectors in the SCAQCF model are constructed as follows by those in the SCQDM.

$$
Y_x = \begin{bmatrix} \frac{4}{t_h}D_1 + Y_{x1} & -\frac{8}{t_h}D_1 \\ \frac{4}{t_h}D_2 + Y_{x2} & -\frac{8}{t_h}D_2 \\ Y_{x3} & 0 \\ \frac{1}{2t_h}D_1 & \frac{2}{t_h}D_1 + Y_{x1} \\ \frac{1}{2t_h}D_2 & \frac{2}{t_h}D_2 + Y_{x2} \\ 0 & Y_{x3} \end{bmatrix} \quad Y_u = \begin{bmatrix} Y_{u1} & 0 \\ Y_{u2} & 0 \\ Y_{u3} & 0 \\ 0 & Y_{u1} \\ 0 & Y_{u2} \\ 0 & Y_{u3} \end{bmatrix} \quad K = \begin{bmatrix} 0 \\ 0 \\ C_3 \\ \frac{3}{2}C_1 \\ \frac{3}{2}C_2 \\ C_3 \end{bmatrix}
$$

$$
F_{xx} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ F_{x3} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & F_{x3} \end{bmatrix} \quad F_{uu} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ F_{u3} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & F_{u3} \end{bmatrix} \quad F_{ux} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ F_{ux3} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & F_{ux3} \end{bmatrix}
$$

$$
N_x = \begin{bmatrix} -Y_{x1} + \frac{4}{t_h}D_1 \\ -Y_{x2} + \frac{4}{t_h}D_2 \\ 0 \\ \frac{1}{2}Y_{x1} - \frac{5}{2t_h}D_1 \\ \frac{1}{2}Y_{x2} - \frac{5}{2t_h}D_2 \\ 0 \end{bmatrix} \quad N_u = \begin{bmatrix} -Y_{u1} \\ -Y_{u2} \\ 0 \\ \frac{1}{2}Y_{u1} \\ \frac{1}{2}Y_{u2} \\ 0 \end{bmatrix} \quad M = \begin{bmatrix} I_{size(\mathbf{i}(t))} \\ 0 \\ 0 \\ -\frac{1}{2}I_{size(\mathbf{i}(t))} \\ 0 \\ 0 \end{bmatrix}
$$

$$
Y_{fx} = \begin{bmatrix} Y_{cx} & 0 \\ 0 & Y_{cx} \end{bmatrix} \quad Y_{fu} = \begin{bmatrix} Y_{cu} & 0 \\ 0 & Y_{cu} \end{bmatrix} \quad C_f = \begin{bmatrix} C_c \\ C_c \end{bmatrix}
$$

$$
F_{fxx} = \begin{bmatrix} F_{cx} & 0 \\ 0 & F_{cx} \end{bmatrix} \quad F_{fuu} = \begin{bmatrix} F_{cu} & 0 \\ 0 & F_{cu} \end{bmatrix} \quad F_{fux} = \begin{bmatrix} F_{cux} & 0 \\ 0 & F_{cux} \end{bmatrix}
$$

$$
\mathbf{u}_{\min} = \begin{bmatrix} \mathbf{u}_{\text{cmin}} \\ \mathbf{u}_{\text{cmin}} \end{bmatrix} \quad \mathbf{u}_{\max} = \begin{bmatrix} \mathbf{u}_{\text{cmax}} \\ \mathbf{u}_{\text{cmax}} \end{bmatrix} \quad \mathbf{u}_{\text{llim}} = \begin{bmatrix} \mathbf{u}_{\text{cllim}} \\ \mathbf{u}_{\text{cllim}} \end{bmatrix}
$$

(14)

Note that $I_{size(\mathbf{i}(t))}$ in (14) is an identity matrix with dimensions equal to the size of $\mathbf{i}(t)$.

The SCAQCF in (12) is able to simulate the operation of any device with each time period containing the model information of both time t and t_m. Different time steps are connected through past history vector B. Paper [51] shows that the physically based TCL model represented in the SCAQCF syntax is much more detailed and accurate compared to the commonly used TCL model given in [52]-[54]. The general form shown in (12) is the object-oriented standard syntax that models devices, which are utilized to construct the network model and subsequently form the MQFOPF problem. Note that the process of forming the SCAQCF from the SCQDM is fully automatic.

4.1.4 Example of Device Modeling

A switched capacitor with capacitance C shown in Figure 5 is considered as an example of device modeling in the quasi-dynamic domain, in which phasors are used. It is assumed to be switched on and off by a binary control variable u_s, where $u_s = 1$ means on and $u_s = 0$ means off.

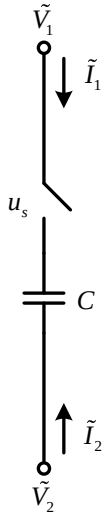

Figure 5 – Example switched capacitor.

The capacitor terminal voltages are $\tilde{V}_1$ and $\tilde{V}_2$, while their corresponding terminal currents $\tilde{I}_1$ and $\tilde{I}_2$. Assuming that a functional constraint of $0.95V_{nom} \le \tilde{V}_1 \le 1.05V_{nom}$ with $V_{\min}$ and $V_{\max}$ being the minimum and maximum voltage allowed, respectively, the device compact model of the switched capacitor is

$$\begin{aligned}
&\tilde{I}_1(t) = j\omega C\left(\tilde{V}_1(t) - \tilde{V}_2(t)\right)u_s(t) \\
&\tilde{I}_2(t) = -j\omega C\left(\tilde{V}_1(t) - \tilde{V}_2(t)\right)u_s(t) \\
&0.95V_{\min} \le \tilde{V}_1(t) \le 1.05V_{\max} \\
&u_s(t) \in \{0,1\}
\end{aligned} \tag{15}$$

where $\omega = 2\pi f_0$ with base frequency f_0 being 60 Hz. To convert (15) to the SCQDM, the voltage and current phasors are represented by real and imaginary parts. Since all quadratic terms are moved to equation set 3 in the SCQDM, new state variables $s_1 = (V_{1r} - V_{2r})u_s$ and $s_2 = (V_{1i} - V_{2i})u_s$ are introduced for quadratization. Another variable y is added for the binary control variable so that $y^2 = u_s(1-u_s)$. By adding y^2 to the objective function of an OPF problem with a large weight, minimizing y^2 drives u_s to a binary value. The maximum control excursion limit for the control movement d_s is 1 in this case. As a result, the SCQDM of the example switched capacitor is given by

$$
\begin{aligned}
&I_{1r}(t) = -\omega C s_2(t) \\
&I_{1i}(t) = \omega C s_1(t) \\
&I_{2r}(t) = \omega C s_2(t) \\
&I_{2i}(t) = -\omega C s_1(t) \\
&0 = s_1(t) - V_{1r}(t) u_s(t) + V_{2r}(t) u_s(t) \\
&0 = s_2(t) - V_{1i}(t) u_s(t) + V_{2i}(t) u_s(t) \\
&0 = y^2(t) - u_s(t) + u_s^2(t) \\
&-V_{1r}^2(t) - V_{1i}^2(t) + V_{\min}^2 \le 0 \\
&V_{1r}^2(t) + V_{1i}^2(t) - V_{\max}^2 \le 0 \\
&0 \le u_s(t) \le 1
\end{aligned}
\tag{16}
$$

where r and i in the subscripts denote the real and imaginary parts of the variables. The control movement constraints at time t in this example are $-1 \le d_s(t) \le 1$. The SCQDM states are

$$
\mathbf{x}(t) = \begin{bmatrix} V_{1r}(t) \\ V_{1i}(t) \\ V_{2r}(t) \\ V_{2i}(t) \\ s_1(t) \\ s_2(t) \\ y(t) \end{bmatrix}
\tag{17}
$$

Note that (16) only has equation sets 1 and 3. After quadratic integration, according to (14), the SCAQCF of the switched capacitor is constructed as

$$
\begin{aligned}
&I_{1r}(t) = -\omega C s_2(t) + \omega C s_2(t-t_h) + I_{1r}(t-t_h)\\
&I_{1i}(t) = \omega C s_1(t) - \omega C s_1(t-t_h) + I_{1i}(t-t_h)\\
&I_{2r}(t) = \omega C s_2(t) - \omega C s_2(t-t_h) + I_{2r}(t-t_h)\\
&I_{2i}(t) = -\omega C s_1(t) + \omega C s_1(t-t_h) + I_{2i}(t-t_h)\\
&0 = s_1(t) - V_{1r}(t)u_s(t) + V_{2r}(t)u_s(t)\\
&0 = s_2(t) - V_{1i}(t)u_s(t) + V_{2i}(t)u_s(t)\\
&0 = y^2(t) - u_s(t) + u_s^2(t)\\
&I_{1r}(t_m) = -\omega C s_2(t_m) - \frac{1}{2}\omega C s_2(t-t_h) - \frac{1}{2} I_{1r}(t-t_h)\\
&I_{1i}(t_m) = \omega C s_1(t_m) + \frac{1}{2}\omega C s_1(t-t_h) - \frac{1}{2} I_{1i}(t-t_h)\\
&I_{2r}(t_m) = \omega C s_2(t_m) + \frac{1}{2}\omega C s_2(t-t_h) - \frac{1}{2} I_{2r}(t-t_h)\\
&I_{2i}(t_m) = -\omega C s_1(t_m) - \frac{1}{2}\omega C s_1(t-t_h) - \frac{1}{2} I_{2i}(t-t_h)\\
&0 = s_1(t_m) - V_{1r}(t_m)u_s(t_m) + V_{2r}(t_m)u_s(t_m)\\
&0 = s_2(t_m) - V_{1i}(t_m)u_s(t_m) + V_{2i}(t_m)u_s(t_m)\\
&0 = y^2(t_m) - u_s(t_m) + u_s^2(t_m)\\
&-V_{1r}^2(t) - V_{1i}^2(t) + V_{\min}^2 \le 0\\
&V_{1r}^2(t) + V_{1i}^2(t) - V_{\max}^2 \le 0\\
&-V_{1r}^2(t_m) - V_{1i}^2(t_m) + V_{\min}^2 \le 0\\
&V_{1r}^2(t_m) + V_{1i}^2(t_m) - V_{\max}^2 \le 0\\
&\begin{bmatrix} 0 \\ 0 \end{bmatrix} \le \begin{bmatrix} u_s(t) \\ u_s(t_m) \end{bmatrix} \le \begin{bmatrix} 1 \\ 1 \end{bmatrix}
\end{aligned}
\tag{18}
$$

with the algorithmic constraints being

$$
\begin{bmatrix} -1 \\ -1 \end{bmatrix} \le \begin{bmatrix} d_s(t) \\ d_s(t_m) \end{bmatrix} \le \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{19}
$$

The capacitor SCAQCF states, controls and control movements are

$$\mathbf{x}=\begin{bmatrix} V_{1r}(t) \\ V_{1i}(t) \\ V_{2r}(t) \\ V_{2i}(t) \\ s_1(t) \\ s_2(t) \\ y(t) \\ V_{1r}(t_m) \\ V_{1i}(t_m) \\ V_{2r}(t_m) \\ V_{2i}(t_m) \\ s_1(t_m) \\ s_2(t_m) \\ y(t_m) \end{bmatrix} \quad \mathbf{u}=\begin{bmatrix} u_s(t) \\ u_s(t_m) \end{bmatrix} \quad \mathbf{d}=\begin{bmatrix} d_s(t) \\ d_s(t_m) \end{bmatrix} \tag{20}$$

With the SCAQCF in (18) obtained, the capacitor modeling process is complete.

4.2 Network Formulation

Given the device SCAQCF models in a system, the model of the system or network is generated automatically and it is also represented in the SCAQCF standard syntax [55]. The states, controls, equations and constraints are mapped from the device level to the network level. The through variables of different devices connected to the same network node sum up to 0. In an electrical network, this corresponds to the Kirchhoff's current law (KCL). In a mechanical system or a thermal system, it is the sum of torques or heat flows at the same node, respectively. Therefore, object orientation being one of the advantages of the SCAQCF is fully reflected in this case. Given n device models in the SCAQCF syntax, Figure 6 illustrates the construction of the network SCAQCF model. The device number is denoted by superscript i in the device models, where $i=1,2,\cdots,n$.

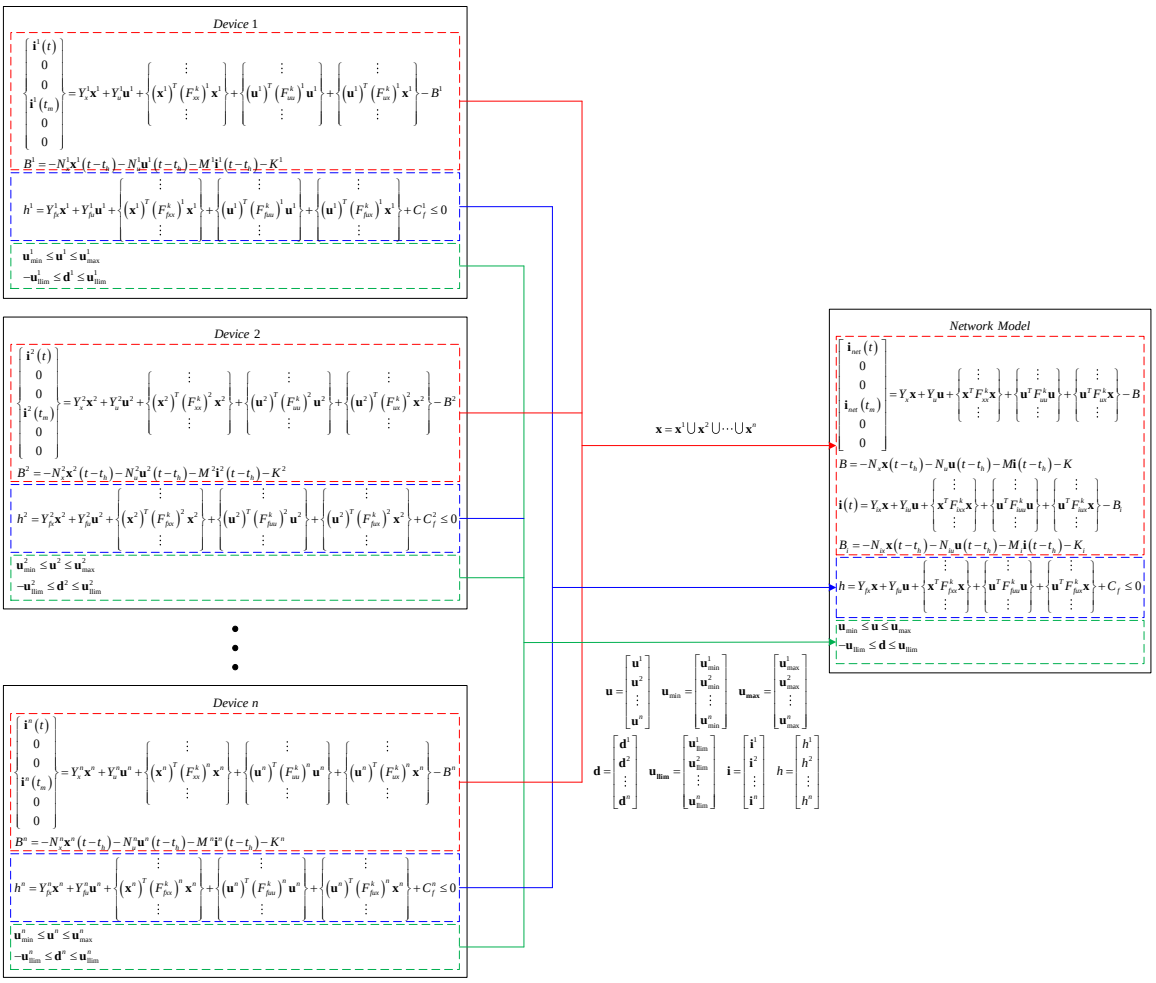

Figure 6 – Network model construction process.

The device equations corresponding to each network node are combined into one network internal equation using KCL by eliminating the through variables of the devices connected to that node. These equations are expressed in terms of device state variables, which now become the network states. Note that the network states are the union of all device states, as the terminal voltages of connected devices are the same variables. The network through variables at the interface nodes are denoted by $\mathbf{i}_{net}$. The internal equations, functional constraints and control limits of the network model directly come from those of the individual device models. Through equations $\mathbf{i}^i(t) = f\left(\mathbf{x}^i, \mathbf{u}^i, \mathbf{i}^i(t-t_h)\right)$ of each device

i are also formed as part of the network model. Subscript i denotes the matrices and vectors for the device through equations. Note that mapping lists for the equations, states, controls and constraints are created and utilized during the network formulation process.

Since every variable has its own unit, which may result in different orders of magnitude for different variables in metric unit, normalization factors are defined for the states, controls, equations and constraints. They are used to normalize the network SCAQCF model shown in Figure 6. The general network model after normalization is

$$
\begin{aligned}
&\begin{bmatrix} \mathbf{i}_{net}(t) \\ 0 \\ 0 \\ \mathbf{i}_{net}(t_m) \\ 0 \\ 0 \end{bmatrix} = Y_x\mathbf{x} + Y_u\mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T F_{xx}^k \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{uu}^k \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{ux}^k \mathbf{x} \\ \vdots \end{Bmatrix} - B \\
&B = -N_x\mathbf{x}(t-t_h) - N_u\mathbf{u}(t-t_h) - M\mathbf{i}(t-t_h) - K \\
&\mathbf{i}(t) = Y_{ix}\mathbf{x} + Y_{iu}\mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T F_{ixx}^k \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{iuu}^k \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{iux}^k \mathbf{x} \\ \vdots \end{Bmatrix} - B_i \\
&B_i = -N_{ix}\mathbf{x}(t-t_h) - N_{iu}\mathbf{u}(t-t_h) - M_i\mathbf{i}(t-t_h) - K_i \\
&h = Y_{fx}\mathbf{x} + Y_{fu}\mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T F_{fxx}^k \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{fuu}^k \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{fux}^k \mathbf{x} \\ \vdots \end{Bmatrix} + C_f \le 0 \\
&\mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max}
\end{aligned}
\tag{21}
$$

The network SCAQCF model is almost the same as the device SCAQCF model, except that it has one more set of equations, which consists of the through equations of all devices in the network. These equations are needed for the formation of the multi-stage quadratic flexible OPF model (MQFOPFM), as the device through variables become a part

of the network past history. The network model is also object-oriented and it can be treated like a large virtual device. Through vector $\mathbf{i}_{net}$ contains the currents flowing into the network and all its elements are zero if the network is a close system, meaning that it does not have any interface nodes that are connected to other systems. With the network SCAQCF model formulated, the MQFOPFM can be formed after an objective function is defined.

4.3 Summary

A physically based object-oriented modeling approach is presented in this chapter. The operation of any device can be represented using a set of mathematical equations based on the device physical properties. Due to possible nonlinearity in the equations, quadratization is applied to ensure that the highest order of terms in the model is 2, thus producing the SCQDM of the device. Since transients exist in the device operation and they are expressed in differential terms, quadratic integration is utilized to convert the SCQDM to the SCAQCF standard syntax. When all devices in a system or network are represented in SCAQCF, the corresponding network model is constructed automatically, also in SCAQCF.

All types of devices and networks, including those that are not electrical, can be modeled in the standard form, so object orientation is achieved. The advantage is that every model is in the same format and hence can be manipulated the same way without knowing what it really is. Therefore, the object-oriented modeling method provides the basis for the formation of the MQFOPFM.

Besides the OPF formulations, the SCAQCF unified syntax also has important applications in other areas of the power systems, such as state estimation [56] – [60], protection [61] – [68], harmonic analysis [69] – [71], and cyber security [72] – [74].

CHAPTER 5. AUTONOMOUS MULTI-STAGE QUADRATIC FLEXIBLE OPTIMAL POWER FLOW FORMULATION

With the network SCAQCF model acquired, a multi-stage quadratic flexible optimal power flow model (MQFOPFM) is formed autonomously. This chapter illustrates the process of the MQFOPFM formation. First, an objective function is selected by the users, defined and generated accordingly. Then, a quadratic OPF problem is formulated by combining the objective function with the network model. Finally, by stacking the OPF model over multiple time stages, an MQFOPFM is obtained. The MQFOPF is performed in the quasi-dynamic domain, where fast electrical transients are neglected but slower dynamics such as electromechanical transients are considered [71].

5.1 Objective Function

The objective function is a part of the MQFOPFM that specifies the objective or goal of the OPF problem. It is selected by the users and represented in a general form, which serves an object-oriented purpose. In this dissertation, specifically two types of objective function are described in detail: network voltage profile optimization and total generation cost minimization.

5.1.1 General Form of Objective Function

Similar to the SCAQCF, a normalization factor is defined for the objective function. To achieve object orientation, the normalized objective function J of the MQFOPFM also has a general form, whose mathematical expression is given by

$$J = Y_{ox}^T \mathbf{x} + Y_{ou}^T \mathbf{u} + \mathbf{x}^T F_{oxx} \mathbf{x} + \mathbf{u}^T F_{ouu} \mathbf{u} + \mathbf{u}^T F_{oux} \mathbf{x} + C_o \tag{22}$$

where subscript o denotes the objective function. Similar to the SCAQCF, in (22) vectors Y have the linear coefficients and matrices F contain the quadratic coefficients. There is only one constant term C_o.

With the objective function general expression introduced, every user-selected objective function can be written in the form of (22) and manipulated the same way no matter how it is defined, thus achieving object orientation.

5.1.2 Network Voltage Profile Optimization

Optimization of the voltage profile in an electrical network is defined as the levelization of selected node voltages [75]. Therefore, the objective is to minimize the sum of squared voltage magnitude mismatches at selected nodes, which is mathematically expressed as

$$\min \quad J = \sum_{i \in S_{bus}} \left(\frac{V_{mag,i} - V_{tar,i}}{\eta_i V_{tar,i}} \right)^2 \tag{23}$$

where S_{bus} is the set of selected buses. $V_{mag,i}$ and $V_{tar,i}$ are the voltage magnitude and targeted voltage value at node i, respectively, while η_i is a pre-defined tolerance value (e.g. 5%) at the node. In a three-phase system, the voltages of all three phases at a bus can be selected for levelization.

When devices with binary control exist in the network, for example switched capacitors described in Section 4.1.4, each continuous variable y_i related to binary variable $u_{b,i}$ is added to the objective function with a big number W_i as its coefficient. As a result, the objective function for voltage profile optimization becomes

$$\min \quad J = \sum_{i \in S_{bus}} \left(\frac{V_{mag,i} - V_{tar,i}}{\eta_i V_{tar,i}} \right)^2 + \sum_{i \in S_{bv}} W_i y_i^2 \tag{24}$$

where S_{bv} is the set of binary variables. Note that coefficients W are large numbers to make sure that states y are driven to 0, thereby making the binary variables equal to either 0 or 1.

Objective function (24) can be written in the form of (22), in which every voltage magnitude V_{mag} is a state of the network. For every node that is selected for voltage levelization, V_{mag} is introduced by a fictious voltage magnitude device connected to that node. In addition, the following equations are included as the device internal equations.

$$\begin{aligned} 0 &= V_r^2 + V_i^2 - V_{mag}^2 \\ 0 &= V_{mag} - s_d^2 \end{aligned} \tag{25}$$

States V_r and V_i are respectively the real and imaginary parts of the selected voltage, while s_d is a dummy variable added to make the computed V_{mag} non-negative. Although vector $\mathbf{x}$ includes states at both time t and t_m, the voltage magnitudes considered in (24) only include the time t variables.

5.1.3 Minimization of Total Generation Cost

The total generation cost minimization is to minimize the sum of the costs of all generator real power outputs, which is given by

$$\min \quad J = \sum_{i \in S_{gen}} C_{gen,i} \tag{26}$$

where S_{gen} is the set of generators in the network and $C_{gen,i}$ is the generation cost of generator i. In order to generate (26), every generator has C_{gen} as a state and its corresponding internal equation is

$$0 = c_g P_{gen}^2 + b_g P_{gen} + a_g - C_{gen} \tag{27}$$

where P_{gen} is the real power output of the generator, which is also a state. c_g, b_g and a_g are parameters entered by the users. For different generators, the parameters may be different.

Similar to the voltage profile optimization objective function, when binary related variables y are included, (26) becomes

$$\min \quad J = \sum_{i \in S_{gen}} C_{gen,i} + \sum_{i \in S_{bv}} W_i y_i^2 \tag{28}$$

where the generation costs are from time t. Note that (28) can also be expressed in the general form.

5.2 Quadratic Optimal Power Flow

Combining the network SCAQCF model in (21) and the objective function in (22) yields the quadratic optimal power flow problem, which is given by

$$
\begin{aligned}
\min \quad & J = Y_{ox}^T\mathbf{x} + Y_{ou}^T\mathbf{u} + \mathbf{x}^T F_{oxx}\mathbf{x} + \mathbf{u}^T F_{ouu}\mathbf{u} + \mathbf{u}^T F_{oux}\mathbf{x} + C_o \\
\text{s.t.} \quad & 0 = Y_x\mathbf{x} + Y_u\mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T F_{xx}^k\mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{uu}^k\mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{ux}^k\mathbf{x} \\ \vdots \end{Bmatrix} - B \\
& B = -N_x\mathbf{x}(t-t_h) - N_u\mathbf{u}(t-t_h) - M\mathbf{i}(t-t_h) - K \\
& 0 = Y_{ix}\mathbf{x} + Y_{iu}\mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T F_{ixx}^k\mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{iuu}^k\mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{iux}^k\mathbf{x} \\ \vdots \end{Bmatrix} - B_i - \mathbf{i}(t) \\
& B_i = -N_{ix}\mathbf{x}(t-t_h) - N_{iu}\mathbf{u}(t-t_h) - M_i\mathbf{i}(t-t_h) - K_i \\
& h = Y_{fx}\mathbf{x} + Y_{fu}\mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T F_{fxx}^k\mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{fuu}^k\mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T F_{fux}^k\mathbf{x} \\ \vdots \end{Bmatrix} + C_f \le 0 \\
& \mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max}
\end{aligned}
\tag{29}
$$

Note that $\mathbf{i}_{net}$ and $\mathbf{i}$ in (21) are moved to the right hand side of the equations in (29) so that the left hand sides are left with 0. Since OPF is performed on a closed or isolated network without interface nodes, $\mathbf{i}_{net}$ does not exist in (29). The algorithmic constraints are $-\mathbf{u}_{\text{llim}} \le \mathbf{d} \le \mathbf{u}_{\text{llim}}$. As a result, the problem is subject to two sets of equality constraints and one set of inequality constraints with bounds on the controls and control movements.

The network states $\mathbf{x}$, controls $\mathbf{u}$ and control movements $\mathbf{d}$ include both time t and t_m variables. However, in the quadratic OPF problem, the optimization states consist of $\mathbf{x}(t)$, $\mathbf{x}(t_m)$ and $\mathbf{i}(t)$. The optimization controls remain the same including $\mathbf{u}(t)$ and

$\mathbf{u}(t_m)$. The optimization control movements are also not changed as they contain $\mathbf{d}(t)$ and $\mathbf{d}(t_m)$. Time $t - t_h$ is seen as the previous time step, so past history vectors B and B_i are constant vectors computed by the states and controls at time $t - t_h$. With (29) obtained, the MQFOPFM is directly formed [55].

5.3 Multi-Stage Quadratic Flexible Optimal Power Flow Model

The multi-stage quadratic flexible OPF model is assumed to be constructed over n stages or time intervals. The timeline is shown in Figure 7. Time t_{im} represents the intermediate time step between t_i and t_{i+1}, where $t_{i+1} = t_i + t_h$ and $t_{i,m} = t_i + h/2 = t_{i+1} - h/2$, with t_h being the time interval between stages as well as the modeling time step size for the device and network SCAQCF models.

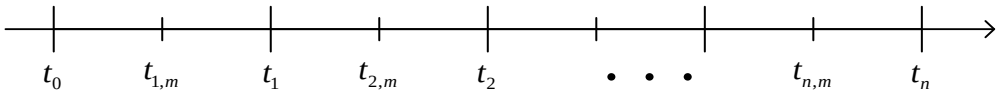

Figure 7 – Timeline of stages.

In the timeline, t_0 is the present. The goal is to optimize the power flow over a horizon of n stages into the future. At every time stage i where $i = 1, 2, \cdots, n$, let the corresponding states, controls and control movements be

$$\mathbf{x}_{t_i} = \begin{bmatrix} \mathbf{x}(t_i) \\ \mathbf{x}(t_{i,m}) \end{bmatrix} \quad \mathbf{u}_{t_i} = \begin{bmatrix} \mathbf{u}(t_i) \\ \mathbf{u}(t_{i,m}) \end{bmatrix} \quad \mathbf{d}_{t_i} = \begin{bmatrix} \mathbf{d}(t_i) \\ \mathbf{d}(t_{i,m}) \end{bmatrix} \tag{30}$$

Then, the OPF problem in (29) is rewritten in (31), which is called the single-stage quadratic flexible OPF model (SQFOPFM) at stage i.

$$
\begin{aligned}
\min \quad & Y_{ox}^T \mathbf{x}_{t_i} + Y_{ou}^T \mathbf{u}_{t_i} + \mathbf{x}_{t_i}^T F_{oxx} \mathbf{x}_{t_i} + \mathbf{u}_{t_i}^T F_{ouu} \mathbf{u}_{t_i} + \mathbf{u}_{t_i}^T F_{oux} \mathbf{x}_{t_i} + C_o \\
\text{s.t.} \quad & 0 = Y_x \mathbf{x}_{t_i} + Y_u \mathbf{u}_{t_i} + \begin{Bmatrix} \vdots \\ \mathbf{x}_{t_i}^T F_{xx}^k \mathbf{x}_{t_i} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}_{t_i}^T F_{uu}^k \mathbf{u}_{t_i} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}_{t_i}^T F_{ux}^k \mathbf{x}_{t_i} \\ \vdots \end{Bmatrix} - B\left(t_{i-1}\right) \\
& B\left(t_{i-1}\right) = -N_x \mathbf{x}\left(t_{i-1}\right) - N_u \mathbf{u}\left(t_{i-1}\right) - M\mathbf{i}\left(t_{i-1}\right) - K \\
& 0 = Y_{ix} \mathbf{x}_{t_i} + Y_{iu} \mathbf{u}_{t_i} + \begin{Bmatrix} \vdots \\ \mathbf{x}_{t_i}^T F_{ixx}^k \mathbf{x}_{t_i} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}_{t_i}^T F_{iuu}^k \mathbf{u}_{t_i} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}_{t_i}^T F_{iux}^k \mathbf{x}_{t_i} \\ \vdots \end{Bmatrix} - B_i\left(t_{i-1}\right) - \mathbf{i}\left(t_i\right) \qquad (31) \\
& B_i\left(t_{i-1}\right) = -N_{ix} \mathbf{x}\left(t_{i-1}\right) - N_{iu} \mathbf{u}\left(t_{i-1}\right) - M_i \mathbf{i}\left(t_{i-1}\right) - K_i \\
& Y_{fx} \mathbf{x}_{t_i} + Y_{fu} \mathbf{u}_{t_i} + \begin{Bmatrix} \vdots \\ \mathbf{x}_{t_i}^T F_{fxx}^k \mathbf{x}_{t_i} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}_{t_i}^T F_{fuu}^k \mathbf{u}_{t_i} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}_{t_i}^T F_{fux}^k \mathbf{x}_{t_i} \\ \vdots \end{Bmatrix} + C_f \le 0 \\
& \mathbf{u}_{\min} \le \mathbf{u}_{t_i} \le \mathbf{u}_{\max}
\end{aligned}
$$

The control movements are bounded as $-\mathbf{u}_{\text{llim}} \le \mathbf{d}_{t_i} \le \mathbf{u}_{\text{llim}}$. Note that when $i = 1$, past history vectors $B\left(t_0\right)$ and $B_i\left(t_0\right)$ are constants that can be provided either by simulation or state estimation [56].

In order to form a quadratic flexible OPF model with n stages as shown in Figure 7, the SQFOPFM in (31) with $i = 1, 2, \cdots, n$ are stacked together. The general expression of the MQFOPFM is given by

$$
\begin{aligned}
\min \quad & J = Y_{Mox}^T \mathbf{x} + Y_{Mou}^T \mathbf{u} + \mathbf{x}^T F_{Moxx} \mathbf{x} + \mathbf{u}^T F_{Mouu} \mathbf{u} + \mathbf{u}^T F_{Moux} \mathbf{x} + C_{Mo} \\
\text{s.t.} \quad & g = Y_{Mx} \mathbf{x} + Y_{Mu} \mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T \left\langle F_{Mxx}^k \right\rangle \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \left\langle F_{Muu}^k \right\rangle \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \left\langle F_{Mux}^k \right\rangle \mathbf{x} \\ \vdots \end{Bmatrix} + K_M = 0 \\
& h = Y_{Mfx} \mathbf{x} + Y_{Mfu} \mathbf{u} + \begin{Bmatrix} \vdots \\ \mathbf{x}^T \left\langle F_{Mfxx}^k \right\rangle \mathbf{x} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \left\langle F_{Mfuu}^k \right\rangle \mathbf{u} \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \mathbf{u}^T \left\langle F_{Mfux}^k \right\rangle \mathbf{x} \\ \vdots \end{Bmatrix} + C_{Mf} \le 0 \\
& \mathbf{u}_{\text{Mmin}} \le \mathbf{u} \le \mathbf{u}_{\text{Mmax}}
\end{aligned}
\tag{32}
$$

where M in the subscripts denotes the model with multiple stages. Similarly, the algorithmic constraints on $\mathbf{d}$ are $-\mathbf{u}_{\text{Mllim}} \le \mathbf{d} \le \mathbf{u}_{\text{Mllim}}$. The states, controls and control movements for model (32) are

$$
\mathbf{x} = \begin{bmatrix} \mathbf{x}(t_1) \\ \mathbf{x}(t_{1,m}) \\ \mathbf{i}(t_1) \\ \mathbf{x}(t_2) \\ \mathbf{x}(t_{2,m}) \\ \mathbf{i}(t_2) \\ \vdots \\ \mathbf{x}(t_n) \\ \mathbf{x}(t_{n,m}) \\ \mathbf{i}(t_n) \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}(t_1) \\ \mathbf{u}(t_{1,m}) \\ \mathbf{u}(t_2) \\ \mathbf{u}(t_{2,m}) \\ \vdots \\ \mathbf{u}(t_n) \\ \mathbf{u}(t_{n,m}) \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} \mathbf{d}(t_1) \\ \mathbf{d}(t_{1,m}) \\ \mathbf{d}(t_2) \\ \mathbf{d}(t_{2,m}) \\ \vdots \\ \mathbf{d}(t_n) \\ \mathbf{d}(t_{n,m}) \end{bmatrix}
\tag{33}
$$

As mentioned in Section 4.2, the through variables $\mathbf{i}$ become a part of the states. J, g and h are all functions of $\mathbf{x}$ and $\mathbf{u}$.

When the objective function in (32) is formed, the coefficient matrices and vectors as well as the constant value are given by

$$Y_{Mox}=\begin{bmatrix} Y_{ox} \\ 0 \\ Y_{ox} \\ 0 \\ \vdots \\ Y_{ox} \\ 0 \end{bmatrix} \quad Y_{Mou}=\begin{bmatrix} Y_{ou} \\ Y_{ou} \\ \vdots \\ Y_{ou} \end{bmatrix} \quad C_{Mo}=nC_o \quad F_{Moxx}=\begin{bmatrix} F_{oxx} & & & & & & \\ & 0 & & & & & \\ & & F_{oxx} & & & & \\ & & & 0 & & & \\ & & & & \ddots & & \\ & & & & & F_{oxx} & \\ & & & & & & 0 \end{bmatrix} \tag{34}$$

$$F_{Mouu}=\begin{bmatrix} F_{ouu} & & & \\ & F_{ouu} & & \\ & & \ddots & \\ & & & F_{ouu} \end{bmatrix} \quad F_{Moux}=\begin{bmatrix} F_{oux} & 0 & & & \\ & F_{oux} & 0 & & \\ & & \ddots & & \\ & & & F_{oux} & 0 \end{bmatrix}$$

When the equality constraints of n stages are stacked up, past history vectors B and B_i from (31) link variables among different stages. As long as they are not 0, the various stages are coupled. In the absence of coupling, the problem collapses to a problem of n independent OPF problems. Hence, with $B(t_0)$ and $B_i(t_0)$ known, the constant vector K_M of the equality constraints $g=0$ in (32) is

$$K_M=\begin{bmatrix} -B(t_0) \\ -B_i(t_0) \\ K \\ K_i \\ K \\ K_i \\ \vdots \\ K \\ K_i \end{bmatrix} \tag{35}$$

The other matrices of the equality constraints are formed as follows.

$$
Y_{Mx} = \begin{bmatrix}
Y_x & 0 & & & & & & & & \\
Y_{ix} & -I & & & & & & & & \\
N_x & M & Y_x & 0 & & & & & & \\
N_{ix} & M_i & Y_{ix} & -I & & & & & & \\
 & & N_x & M & Y_x & 0 & & & & \\
 & & N_{ix} & M_i & Y_{ix} & -I & & & & \\
 & & & & & & \ddots & & & \\
 & & & & & & & N_x & M & Y_x & 0 \\
 & & & & & & & N_{ix} & M_i & Y_{ix} & -I
\end{bmatrix}
$$

$$
Y_{Mu} = \begin{bmatrix}
Y_u & 0 & & & & & \\
Y_{iu} & 0 & & & & & \\
N_u & Y_u & & & & & \\
N_{iu} & Y_{iu} & & & & & \\
 & N_u & Y_u & & & & \\
 & N_{iu} & Y_{iu} & & & & \\
 & & & \ddots & & & \\
 & & & & N_u & Y_u & \\
 & & & & N_{iu} & Y_{iu}
\end{bmatrix}
\quad
F_{Mxx} = \begin{bmatrix}
F_{xx} & & & & & & \\
F_{ixx} & & & & & & \\
 & F_{xx} & & & & & \\
 & F_{ixx} & & & & & \\
 & & F_{xx} & & & & \\
 & & F_{ixx} & & & & \\
 & & & & \ddots & & \\
 & & & & & & F_{xx} \\
 & & & & & & F_{ixx}
\end{bmatrix}
\tag{36}
$$

$$
F_{Muu} = \begin{bmatrix}
F_{uu} & & & & & & \\
F_{iuu} & & & & & & \\
 & F_{uu} & & & & & \\
 & F_{iuu} & & & & & \\
 & & F_{uu} & & & & \\
 & & F_{iuu} & & & & \\
 & & & \ddots & & & \\
 & & & & & F_{uu} & \\
 & & & & & F_{iuu} &
\end{bmatrix}
\quad
F_{Mux} = \begin{bmatrix}
F_{ux} & & & & & & \\
F_{iux} & & & & & & \\
 & F_{ux} & & & & & \\
 & F_{iux} & & & & & \\
 & & F_{ux} & & & & \\
 & & F_{iux} & & & & \\
 & & & & \ddots & & \\
 & & & & & & F_{ux} \\
 & & & & & & F_{iux}
\end{bmatrix}
$$

As for the inequality constraints and control limits, the corresponding matrices and vectors in the MQFOPFM are constructed as

$$
\begin{aligned}
Y_{Mfx} &= \begin{bmatrix} Y_{fx} & 0 & & & \\ & Y_{fx} & 0 & & \\ & & \ddots & & \\ & & & Y_{fx} & 0 \end{bmatrix} \quad Y_{Mfu} = \begin{bmatrix} Y_{fu} & & & \\ & Y_{fu} & & \\ & & \ddots & \\ & & & Y_{fu} \end{bmatrix} \\
F_{Mfxx} &= \begin{bmatrix} F_{fxx} & & & \\ & F_{fxx} & & \\ & & \ddots & \\ & & & F_{fxx} \end{bmatrix} \quad F_{Mfuu} = \begin{bmatrix} F_{fuu} & & & \\ & F_{fuu} & & \\ & & \ddots & \\ & & & F_{fuu} \end{bmatrix} \\
F_{Mfux} &= \begin{bmatrix} F_{fux} & & & \\ & F_{fux} & & \\ & & \ddots & \\ & & & F_{fux} \end{bmatrix} \quad C_{Mf} = \begin{bmatrix} C_f \\ C_f \\ \vdots \\ C_f \end{bmatrix} \\
\mathbf{u}_{\text{Mmin}} &= \begin{bmatrix} \mathbf{u}_{\text{min}} \\ \mathbf{u}_{\text{min}} \\ \vdots \\ \mathbf{u}_{\text{min}} \end{bmatrix} \quad \mathbf{u}_{\text{Mmax}} = \begin{bmatrix} \mathbf{u}_{\text{max}} \\ \mathbf{u}_{\text{max}} \\ \vdots \\ \mathbf{u}_{\text{max}} \end{bmatrix} \quad \mathbf{u}_{\text{Mllim}} = \begin{bmatrix} \mathbf{u}_{\text{llim}} \\ \mathbf{u}_{\text{llim}} \\ \vdots \\ \mathbf{u}_{\text{llim}} \end{bmatrix}
\end{aligned}
\tag{37}
$$

where $\mathbf{u}_{\text{Mllim}}$ only contains the initial maximum control excursion limits. Their values are adjusted in the SLP algorithm utilized to solve the MQFOPF problem.

From (33) – (37), the MQFOPFM in (32) is successfully acquired. The duration of the dynamics included in the model cannot be less than the length of a time stage t_h. When the number of stages is equal to 1, model (32) becomes an SQFOPFM. Since the network model and objective function, upon which the MQFOPFM is built, are already normalized, all the variables are in per unit. The formation process of the MQFOPF problem is fully autonomous from device modeling, to network formulation, and to the construction of the SQFOPFM as well as the final MQFOPFM. The next step is to solve the multi-stage quadratic flexible OPF problem through SLP [55], which is described in detail in the following chapter.

5.4 Summary

This chapter describes how a MQFOPFM is formulated upon a network model in the quasi-dynamic domain. First, a user-selected objective function is defined in a general form based on the network states and controls. Two specific objective functions are discussed. Network voltage profile optimization levelizes the voltages at selected network nodes, while minimization of total generation costs minimizes the sum of the costs of all generator real power outputs. Second, the SQFOPFM is obtained with the network SCAQCF model providing the states, equations and inequality constraints, as well as the controls and their boundaries. Finally, the MQFOPFM is constructed by stacking the SQFOPFM over a horizon of multiple time stages or periods into the future.

Since the formulation of the network model is based upon device models in the SCAQCF syntax, whose connection to other devices is defined by through variables, most of the network equations are linear KCL equations. Therefore, most of the equality constraints in the resulting MQFOPFM are linear, while the rest quadratic. This structure of the OPF problem is different from the conventional formulation, where power balance equations dominate the model and higher nonlinearities like trigonometric terms exist. As a result, the MQFOPFM is less complex and easier to solve compared to the conventional OPF models.

CHAPTER 6. OPTIMAL POWER FLOW SOLUTION METHOD VIA SEQUENTIAL LINEAR PROGRAMMING

The method used to solve the OPF problem is sequential linear programming (SLP). The costate method is utilized to linearize the MQFOPFM with respect to the controls, generating the linearized flexible OPF model (LFOPFM), which is solved by a linear programming (LP) solver in every iteration. By updating the operating point and adjusting the maximum control excursion limits iteratively, eventually an optimal operating point is obtained with the corresponding controls being the optimal control actions for the network.

6.1 Overall Approach

As illustrated in Figure 8, the SLP algorithm starts after the formation of the MQFOPFM. Obtained through either simulation or state estimation, the initial system conditions at t_1 include states $\mathbf{x}^0(t_1)$, controls $\mathbf{u}^0(t_1)$ and device through values $\mathbf{i}^0(t_1)$, as well as past history vectors of the network $B(t_0)$ and $B_i(t_0)$. The initial operating point $(\mathbf{x}^0, \mathbf{u}^0)$ for the algorithm is

$$\mathbf{x}^0 = \begin{bmatrix} \mathbf{x}^0(t_1) \\ \mathbf{i}^0(t_1) \\ \mathbf{x}^0(t_1) \\ \mathbf{i}^0(t_1) \\ \vdots \\ \mathbf{x}^0(t_1) \\ \mathbf{i}^0(t_1) \end{bmatrix} \quad \mathbf{u}^0 = \begin{bmatrix} \mathbf{u}^0(t_1) \\ \mathbf{u}^0(t_1) \\ \vdots \\ \mathbf{u}^0(t_1) \end{bmatrix} \tag{38}$$

In addition, three small positive values ε_0, ε_1 and ε_2 used throughout the SLP algorithm are defined in the initialization process.

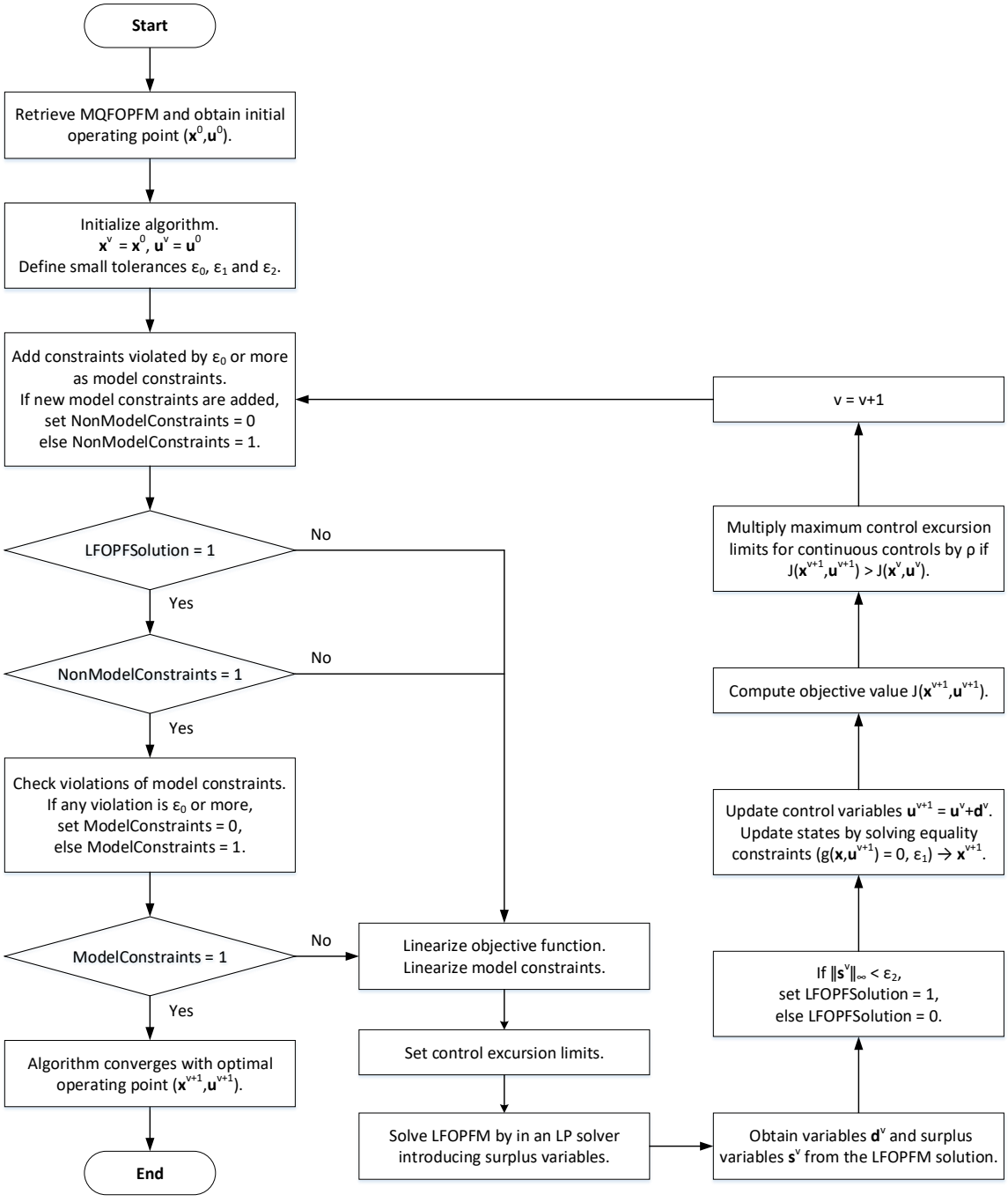

Figure 8 – Sequential linear programming algorithm.

In each SLP iteration ν, the solution method works in a subspace of the MQFOPF constraints, which is defined as the model constraint set. The model constraints are defined as the constraints that are considered for linearization and included in the linearized problem to be solved. At the start of the iteration, the operating point $\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)$ is substituted into all inequality constraints $h\left(\mathbf{x},\mathbf{u}\right)$ to check for violations. The newly violated constraints are added to the set of model constraints $h_m\left(\mathbf{x},\mathbf{u}\right)$. In other words, with tolerance ε_0 defined, constraint i becomes a new model constraint if $h_i\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)\geq\varepsilon_0$ while it is not already in the set. By doing this, the size of the OPF problem to be solved in each iteration is greatly reduced.

After the model constraints are picked out, the MQFOPFM is linearized with respect to only the control variables using the costate method. Both the objective function and the model constraints are linearized. The control excursion limits are set after the linearization process. With these steps completed, the LFOPFM is formed and then solved by an LP solver.

When solving the LFOPFM, non-negative surplus variables $\mathbf{s}^{\nu}$ are introduced and minimized to avoid infeasibility of the linearized problem. The solution is $\mathbf{d}^{\nu}$, which contains the control movements $\mathbf{u}^{\nu+1}-\mathbf{u}^{\nu}$. They are used to update the controls in the network. Then, the new states are computed from Newton's method with tolerance ε_2. Hence, the operating point is updated as $\left(\mathbf{x}^{\nu+1},\mathbf{u}^{\nu+1}\right)$, which is used to compute the objective function value J. The maximum control excursion limits $\mathbf{u}_{\text{Mllim}}$ are adjusted

according to the comparison between $J\left(\mathbf{x}^{\nu+1},\mathbf{u}^{\nu+1}\right)$ and $J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)$. The SLP algorithm proceeds to the next iteration afterwards, with the constraints tested for violations using the new operating point, followed by the convergence check. If the LFOPFM was successfully solved with the maximum surplus variable being less than ε_1 in the previous iteration and no constraint is violated in the current iteration, the SLP algorithm has converged.

The MQFOPFM is solved automatically through the SLP algorithm after the MQFOPFM is formed. Therefore, the whole process of the formulation and solution of the MQFOPF problem is fully autonomous. Note that metaheuristics like the EA implemented in [76] are not used at all in the proposed algorithm.

6.2 Linearization of Optimal Power Flow

Linearization of the MQFOPFM is a vital part of the SLP algorithm, as it generates a linearized problem that can be solved easily using an LP solver. The costate method is utilized during the linearization process, yielding the LFOPFM. It is an extremely efficient method and the size of the problem becomes much smaller after linearization.

6.2.1 Costate Method

The MQOFPFM consists of both states and controls. To simplify the problem, the costate method is applied so that the quadratic model becomes a linearized model represented by only the control variables. Note that the costate method is a sensitivity-based linearization approach, which utilizes the sensitivities of the OPF model with respect to controls $\mathbf{u}$ [77]. As shown in Figure 9, the process includes the linearization of both the

objective function and model constraints. After linearization, with the help of an LP solver, the MQFOPF problem can be solved easily.

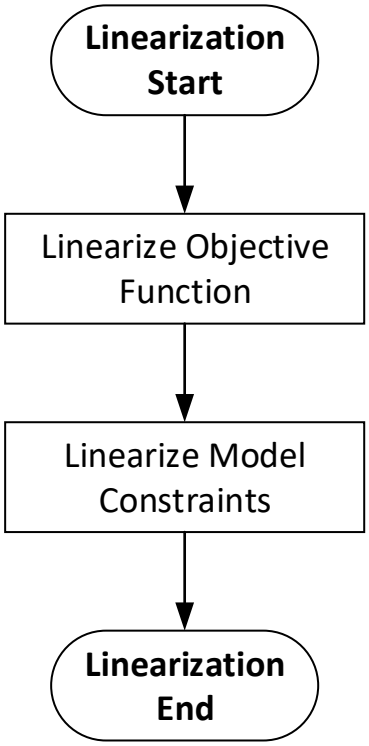

Figure 9 – Linearization process.

Given the objective function J in (32), based on Taylor's expansion with respect to the controls, at iteration ν around operating point $\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)$, J is approximated by

$$J \cong J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)+\frac{dJ\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{d\mathbf{u}}\mathbf{d} \tag{39}$$

where $\mathbf{d}=\mathbf{u}-\mathbf{u}^{\nu}$ are the control movements and $dJ\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)\big/d\mathbf{u}$ is given by

$$\frac{dJ\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{d\mathbf{u}}=\frac{\partial J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial\mathbf{u}}+\frac{\partial J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial\mathbf{x}}\frac{d\mathbf{x}}{d\mathbf{u}} \tag{40}$$

Since the objective function is quadratic and all the coefficient matrices have been defined in (34), the partial derivatives in (40) are computed as

$$\begin{aligned}\frac{\partial J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial\mathbf{u}}&=Y_{Mou}+\left(\mathbf{u}^{\nu}\right)^{T}F_{Mouu}+\left(F_{Mouu}\mathbf{u}^{\nu}\right)^{T}+\left(F_{Moux}\mathbf{x}^{\nu}\right)^{T}\\\frac{\partial J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial\mathbf{x}}&=Y_{Mox}+\left(\mathbf{x}^{\nu}\right)^{T}F_{Moxx}+\left(F_{Moxx}\mathbf{x}^{\nu}\right)^{T}+\left(\mathbf{u}^{\nu}\right)^{T}F_{Moux}\end{aligned} \tag{41}$$

Derivative $d\mathbf{x}/d\mathbf{u}$ is obtained from the equality constraints $g=0$. Differentiating both sides of the equations yields

$$\frac{\partial g\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial\mathbf{u}}+\frac{\partial g\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial\mathbf{x}}\frac{d\mathbf{x}}{d\mathbf{u}}=0 \tag{42}$$

Therefore, $d\mathbf{x}/d\mathbf{u}$ is calculated as

$$\frac{d\mathbf{x}}{d\mathbf{u}}=-\left(\frac{\partial g\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial\mathbf{x}}\right)^{-1}\frac{\partial g\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial\mathbf{u}} \tag{43}$$

where the partial derivatives are acquired from (36) as

$$\frac{\partial g\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}} = Y_{Mu} + \left\{ \begin{matrix} \vdots \\ \left(\mathbf{u}^{\nu}\right)^{T} F^{k}_{Muu} \\ \vdots \end{matrix} \right\} + \left\{ \begin{matrix} \vdots \\ F^{k}_{Muu}\,\mathbf{u}^{\nu} \\ \vdots \end{matrix} \right\}^{T} + \left\{ \begin{matrix} \vdots \\ F^{k}_{Mux}\,\mathbf{x}^{\nu} \\ \vdots \end{matrix} \right\}^{T}$$
$$\frac{\partial g\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{x}} = Y_{Mx} + \left\{ \begin{matrix} \vdots \\ \left(\mathbf{x}^{\nu}\right)^{T} F^{k}_{Mxx} \\ \vdots \end{matrix} \right\} + \left\{ \begin{matrix} \vdots \\ F^{k}_{Mxx}\,\mathbf{x}^{\nu} \\ \vdots \end{matrix} \right\}^{T} + \left\{ \begin{matrix} \vdots \\ \left(\mathbf{u}^{\nu}\right)^{T} F^{k}_{Mux} \\ \vdots \end{matrix} \right\} \tag{44}$$

Partial derivative $\partial g\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)/\partial \mathbf{x}$ is the Jacobian matrix. With (43) obtained, substituting (40) into (39) yields

$$J \cong J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right) + \left(\frac{\partial J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}} - \hat{\mathbf{x}}\frac{\partial g\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}} \right)\mathbf{d} \tag{45}$$

where the costate vector $\hat{\mathbf{x}}$ is given by

$$\hat{\mathbf{x}} = \frac{\partial J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{x}} \left(\frac{\partial g\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{x}} \right)^{-1} \tag{46}$$

Similarly, the model constraints are also linearized using the costate method. Functions h_m are linearized around operating point $\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)$ as

$$h_m \cong h_m\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right) + \frac{dh_m\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)}{d\mathbf{u}}\mathbf{d} \tag{47}$$

where $dh_m\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)/d\mathbf{u}$ has the following expression.

$$\frac{dh_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{d\mathbf{u}}=\frac{\partial h_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial\mathbf{u}}+\frac{\partial h_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial\mathbf{x}}\frac{d\mathbf{x}}{d\mathbf{u}} \tag{48}$$

The partial derivatives are also computed as follows based on the MQFOPFM constraint matrices in (37).

$$\begin{aligned}\frac{\partial h_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial\mathbf{u}}&=Y_{Mfu}+\left\{\begin{matrix}\vdots\\ \left(\mathbf{u}^\nu\right)^T F^k_{Mfuu}\\ \vdots\end{matrix}\right\}+\left\{\begin{matrix}\vdots\\ F^k_{Mfuu}\,\mathbf{u}^\nu\\ \vdots\end{matrix}\right\}^T+\left\{\begin{matrix}\vdots\\ F^k_{Mfux}\,\mathbf{x}^\nu\\ \vdots\end{matrix}\right\}^T\\ \frac{\partial h_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial\mathbf{x}}&=Y_{Mfx}+\left\{\begin{matrix}\vdots\\ \left(\mathbf{x}^\nu\right)^T F^k_{Mfxx}\\ \vdots\end{matrix}\right\}+\left\{\begin{matrix}\vdots\\ F^k_{Mfxx}\,\mathbf{x}^\nu\\ \vdots\end{matrix}\right\}^T+\left\{\begin{matrix}\vdots\\ \left(\mathbf{u}^\nu\right)^T F^k_{Mfux}\\ \vdots\end{matrix}\right\}\end{aligned} \tag{49}$$

Note that k in the subscripts represents constraint k, which is a model constraint. As a result, the expression in (47) becomes

$$h_m\cong h_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)+\left(\frac{\partial h_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial\mathbf{u}}-\hat{\mathbf{y}}\,\frac{\partial g\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial\mathbf{u}}\right)\mathbf{d} \tag{50}$$

where $\hat{\mathbf{y}}$ is the costate matrix in this case and it is given by

$$\hat{\mathbf{y}}=\frac{\partial h_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial\mathbf{x}}\left(\frac{\partial g\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial\mathbf{x}}\right)^{-1} \tag{51}$$

To sum up, the linearized expression of the objective function and model constraints produced by the costate method are (45) and (50), respectively. The detailed linearization procedures are shown in Figure 10. $n_{control}$ and n_{mconst} are the number of

controls and number of model constraints, respectively. i and j are indices for counting the controls and model constraints. Temporary matrix α and vector β are used during calculations. Note that the model constraints are linearized one at a time.

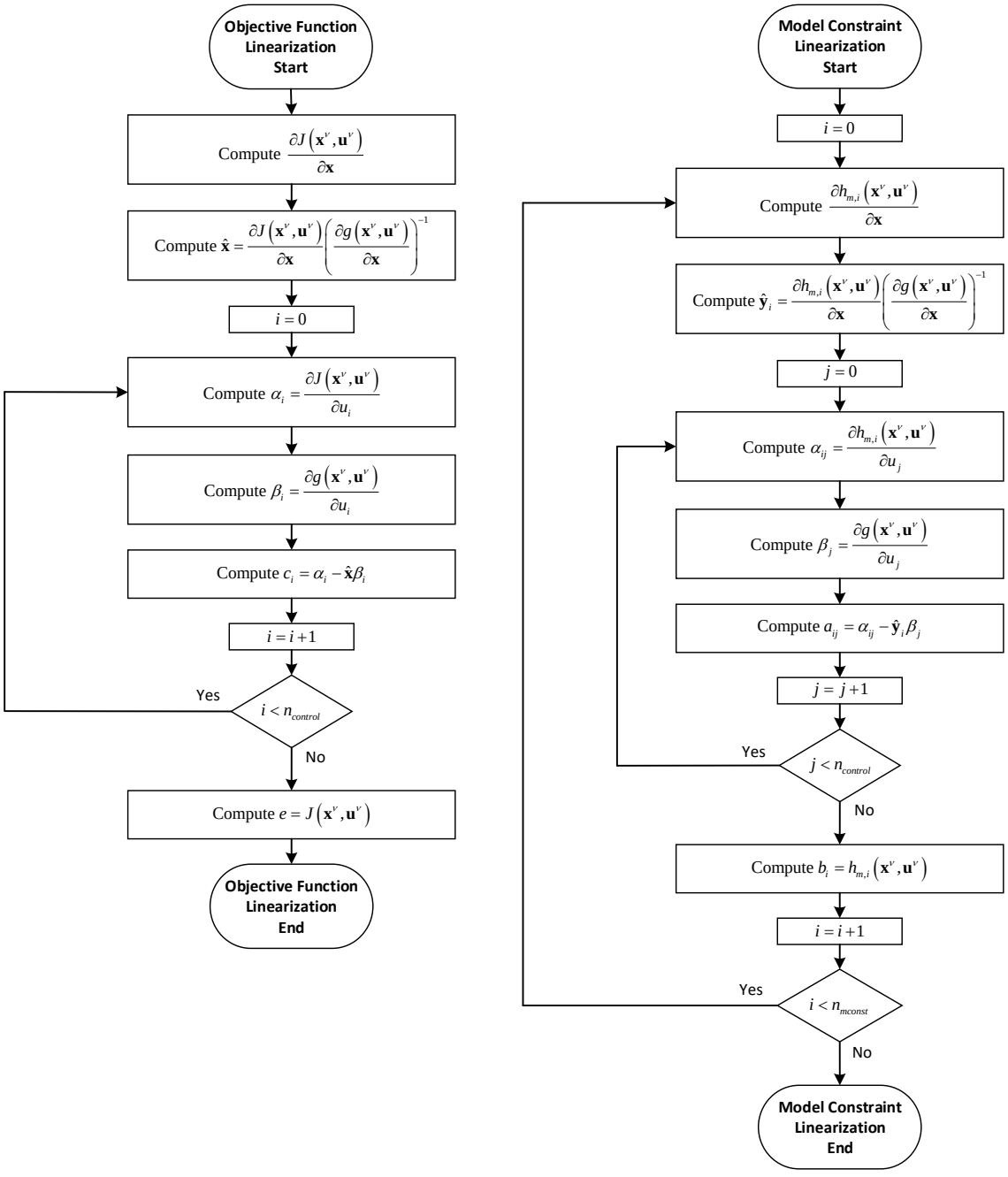

Figure 10 – Detailed linearization procedures.

By using the costate method to linearize the MQFOPFM with respect to only the controls, the size of the linearized problem is significantly reduced. Although inversing the Jacobian matrix is most time-consuming step in the linearization process, the costate method uses the Jacobian matrix available from the previous iteration. In addition, only a few partial derivatives need to be computed with simple additions and multiplications in every iteration. Therefore, the costate method is extremely efficient.

6.2.2 Linearized Flexible Optimal Power Flow Model

From the costate method, it is obvious that the control movements $\mathbf{d}$ are the variables of the linearized problem. Therefore, besides the linearized objective function and model constraints, the variable bounds are required to complete the LFOPFM. The lower and upper bounds of $\mathbf{d}$ are introduced as $\mathbf{d}_{\min}$ and $\mathbf{d}_{\max}$, respectively. Based on $\mathbf{d} = \mathbf{u} - \mathbf{u}^{\nu}$ in each iteration ν, as well as $\mathbf{u}_{\text{Mmin}} \le \mathbf{u} \le \mathbf{u}_{\text{Mmax}}$ and $-\mathbf{u}_{\text{Mllim}} \le \mathbf{d} \le \mathbf{u}_{\text{Mllim}}$ defined in (32), every entry i in $\mathbf{d}_{\min}$ and $\mathbf{d}_{\max}$ is computed as

$$\begin{aligned} d_{\min,i} &= \max\left(u_{\text{Mmin},i} - u_i^{\nu}, -u_{\text{Mllim},i}\right) \\ d_{\max,i} &= \min\left(u_{\text{Mmax},i} - u_i^{\nu}, u_{\text{Mllim},i}\right) \end{aligned} \tag{52}$$

Combining the objective function in (39), model constraints in (47) and control movement limits (52) yields the LFOPFM, whose general expression is given by

$$\begin{aligned} \min \quad & c^T\mathbf{d} + e \\ \text{s.t.} \quad & a\mathbf{d} + b \le 0 \\ & \mathbf{d}_{\min} \le \mathbf{d} \le \mathbf{d}_{\max} \end{aligned} \tag{53}$$

where c^T, e, a and b are

$$\begin{aligned} c^T &= \frac{dJ\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{d\mathbf{u}} = \frac{\partial J\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial \mathbf{u}} - \hat{\mathbf{x}}\frac{\partial g\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial \mathbf{u}} \\ e &= J\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right) \\ a &= \frac{dh_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{d\mathbf{u}} = \frac{\partial h_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial \mathbf{u}} - \hat{\mathbf{y}}\frac{\partial g\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right)}{\partial \mathbf{u}} \\ b &= h_m\left(\mathbf{x}^\nu,\mathbf{u}^\nu\right) \end{aligned} \tag{54}$$

6.3 Iterative Solution Process

Iteratively, an LP solver is utilized to solve the LFOPFM formed. Solution **d** obtained is used to update the controls and states, thus acquiring the new operating point for the next iteration.

6.3.1 Linear Programming Solution

After (53) is obtained, it is solved in an LP solver. To ensure the feasibility of the linearized problem, non-negative surplus variables **s** are introduced. A surplus variable is assigned to every model constraint and minimized in the objective function with a large weight. As a result, (53) becomes

$$\begin{aligned} \min \quad & c^T\mathbf{d} + e + w^T\mathbf{s} \\ \text{s.t.} \quad & a\mathbf{d} + b - \mathbf{s} \le 0 \\ & \mathbf{d}_{\min} \le \mathbf{d} \le \mathbf{d}_{\max} \\ & \mathbf{s} \ge 0 \end{aligned} \tag{55}$$

where w is the weight vector. Due to **s** being non-negative, the inequality constraints in (55) are always satisfied.

Linear problem (55) is fed to an LP solver, which produces vectors $\mathbf{d}$ and $\mathbf{s}$ as its solution. If all entries in $\mathbf{s}$ is driven to 0, the original problem in (53) is said to be feasible. Otherwise, the it is infeasible. No matter whether the LFOPFM is feasible or not, $\mathbf{d}$ is used to compute the new operating point.

6.3.2 Operating Point Update

By definition, the control movements $\mathbf{d}^{\nu}$ given by the solution of (55) is the difference between $\mathbf{u}^{\nu+1}$ and $\mathbf{u}^{\nu}$. Therefore, the controls are updated as $\mathbf{u}^{\nu+1} = \mathbf{u}^{\nu} + \mathbf{d}^{\nu}$. It is assumed that the controls are applied to the system at time t in each stage, so the time t_m values are the same as the time t values. Then, Newton's method is applied to the MQFOPFM equality equations $g = 0$ to solve for the updated states $\mathbf{x}^{\nu+1}$. Let κ be the iteration count, $\kappa_{\max}$ be the maximum number of iterations allowed, and ε_1 be a pre-defined tolerance value for Newton's method. The steps for updating the state variables are listed as follows.

1) Let $\kappa = 0$ and $\mathbf{x}^{\kappa} = \mathbf{x}^{\nu}$.
2) Substitute κ^{ν} and $\mathbf{u}^{\nu+1}$ into the MQFOPFM equations and compute $g\left(\mathbf{x}^{\kappa}, \mathbf{u}^{\nu+1}\right)$. If $\left\| g\left(\mathbf{x}^{\kappa}, \mathbf{u}^{\nu+1}\right) \right\|_{\infty} \le \varepsilon_1$, the procedure terminates and $\mathbf{x}^{\nu+1} = \mathbf{x}^{\kappa}$. Otherwise, go to the next step.
3) Compute Jacobian matrix $\partial g\left(\mathbf{x}^{\kappa}, \mathbf{u}^{\nu+1}\right) \big/ \partial \mathbf{x}$.
4) Calculate $\mathbf{x}^{\kappa+1}$ as

$$\mathbf{x}^{\kappa+1} = \mathbf{x}^{\kappa} - \left(\frac{\partial g\left(\mathbf{x}^{\kappa}, \mathbf{u}^{\nu+1}\right)}{\partial \mathbf{x}} \right)^{-1} g\left(\mathbf{x}^{\kappa}, \mathbf{u}^{\nu+1}\right) \tag{56}$$

5) Let $\kappa = \kappa + 1$. If $\kappa \le \kappa_{\max}$, go to step 2). Otherwise, non-convergence is reported for Newton's method.

With the states updated, new operating point $\left(\mathbf{x}^{\nu+1}, \mathbf{u}^{\nu+1}\right)$ is obtained. The SLP proceeds to the next iteration, which is $\nu+1$, after the maximum control excursion limits are adjusted. Operating point $\left(\mathbf{x}^{\nu+1}, \mathbf{u}^{\nu+1}\right)$ is used in iteration $\nu+1$ for constraint violation check, and subsequently linearization if the algorithm does not converge at this point.

6.4 Algorithm Convergence

As illustrated in Figure 8, the SLP algorithm convergence is determined after the check of constraint violations in every iteration. The linearization process follows if the algorithm continues.

In order to drive the algorithm to convergence, another step is required after the update of operating point. This step includes the computation of the objective function value, based on which the maximum control excursion limits $\mathbf{u}_{\text{Mllim}}$ are adjusted.

6.4.1 Adjustment of Maximum Control Excursion Limits

Once operating point $\left(\mathbf{x}^{\nu+1}, \mathbf{u}^{\nu+1}\right)$ is obtained, the objective function value $J\left(\mathbf{x}^{\nu+1}, \mathbf{u}^{\nu+1}\right)$ is calculated accordingly. The adjustment of $\mathbf{u}_{\text{Mllim}}$ is based on the result of

comparison between $J\left(\mathbf{x}^{\nu+1},\mathbf{u}^{\nu+1}\right)$ and $J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)$, which is from the previous iteration. If $J\left(\mathbf{x}^{\nu+1},\mathbf{u}^{\nu+1}\right) \geq J\left(\mathbf{x}^{\nu},\mathbf{u}^{\nu}\right)$, the maximum control excursion limits of all the continuous control variables are multiplied by ρ, a parameter between 0 and 1 that is tuned to get a better SLP performance. Otherwise, nothing needs to be done. Note that if ρ is too small, the algorithm may be stuck with an infeasible solution after several iterations. However, if ρ is too large, oscillations in the solution may be observed between iterations. Therefore, its value needs to be carefully selected.

Adjusting $\mathbf{u}_{\text{Mllim}}$ affects the computations of $\mathbf{d}_{\min}$ and $\mathbf{d}_{\max}$ in (52), which limit the control movements between two consecutive iterations. Since the initial values defined during the MQFOPFM formation already guarantee a small linearization error, halving $\mathbf{u}_{\text{Mllim}}$ makes sure that the error does not increase.

6.4.2 Convergence Criteria

The SLP convergence is determined after operating point $\left(\mathbf{x}^{\nu+1},\mathbf{u}^{\nu+1}\right)$ is substituted into all the MQFOPFM inequality constraints to check for violations in iteration $\nu+1$. There are three convergence criteria, which correspond to three binary indicators. As shown in Figure 8, the indicators are NonModelConstraints, ModelConstraints and LFOPFSolution. Indicator NonModelConstraints = 1 if all the constraints that are not in the model constraint set are satisfied. If any violation is detected, the corresponding constraint is added as a new model constraint and NonModelConstraints is set to 0. For indicator ModelConstraints, it is 1 when all the model constraints are satisfied. Otherwise, its value is equal to 0. The third indicator LFOPFSolution is set to 1 if the LFOPFM in (53)

is solved with a feasible solution in the previous iteration, meaning the surplus variables satisfy $\left\|\mathbf{s}^{\nu}\right\|_{\infty} < \varepsilon_2$, where ε_2 is a pre-defined small positive number. If any variable *s* is not driven to 0 during the LP solution process, LFOPFSolution = 0. The SLP algorithm converges at iteration $\nu+1$ only when all three indicators are 1. In this case, $\left(\mathbf{x}^{\nu+1}, \mathbf{u}^{\nu+1}\right)$ is the optimal operating point. The algorithm proceeds to the linearization process if any of the three indicators has a different binary value. However, if the iteration count exceeds the maximum number of iterations, non-convergence is reported.

In general, the SLP algorithm converges when the LFOPFM has a feasible solution and all inequality constraints in the MQFOPFM are satisfied. Note that the optimal operating point contains the normalized states and controls in all stages within the optimization horizon. The control values at each stage are the optimal control actions to be implemented back into the selected network at that time period to derive the system to the optimal operating condition. With the system operating conditions estimated with field measurements and used as the initial operating point for solving the MQFOPF problem, applying the optimal controls back into the field completes a full state feedback control loop [78].

6.5 Summary

In order to solve the MQFOPFM, an SLP algorithm is developed in this chapter. The operating point is substituted into the constraints to check for violations in each iteration. The violated constraints are added to the model constraint set that is considered in the linearized problem. The costate method is used to linearize the objective function and model constraints with respect to the control variables, giving the LFOPFM, which is

solved through an LP solver. The control movements obtained from the solution are used to update the controls, and then the new state values are computed via Newton's method. The maximum control excursion limits are adjusted at the end of the iteration based on the objective function value calculated. The algorithm convergence is determined after the model constraints are defined. When the LFOPFM in the previous iteration has a feasible solution and no constraint is violated, the algorithm converges.

Although the original MQFOPF problem has many states and controls, by adding model constraints incrementally, the size of the optimization problem becomes a lot smaller. The costate method further reduces the size of the problem by linearizing the model with respect to only the control variables and eliminate the states variables. Another advantage of the costate method is that it uses the Jacobian matrix already computed in the previous iteration. Only a few additions and multiplications are required for linearization. Hence, the process is extremely efficient. All the above-mentioned merits of the SLP algorithm makes it a good method for solving the MQFOPFM. If too many stages are considered and the OPF problem cannot be solved online, the model predictive control scheme used in [79] can be integrated so that the optimization becomes a moving-horizon problem with less stages in each horizon.

CHAPTER 7. SECURITY CONSTRAINED QUADRATIC OPTIMAL POWER FLOW

Another application of the proposed autonomous formulation and solution method is the security constrained quadratic OPF (SCQOPF). With the network SCAQCF model given, a security constrained quadratic OPF model (SCQOPFM) can be constructed. Besides those procedures required for the MQFOPFM formation, the SCQOPFM formulation process also includes the introduction of device outage models and contingency networks. However, unlike the MQFOPF problem, the SCQOPF problem uses device and network models in the frequency domain, which neglects all transients and only a single time instant is considered during the process.

The original network model formed from the device SCAQCF models is seen as the base case for the SCQOPF problem. When contingencies are imported, every contingency has a corresponding network model, which is adapted from the base case. First the base case model is copied over to the contingency. Then, the outage device is removed from the contingency network, before which an outage model is defined for each device. Note that the device outage models and contingency network models are also represented in the SCAQCF standard syntax. Together with a user-defined objective function, combining the base case and all contingency network models generates the SCQOPFM. Each network model has its own states, equations and constraints. However, all network models share the same set of controls. The SLP solution method is utilized to solve the SCQOPF problem. The costate method is applied to every contingency to linearize the network model, giving the linearized security constrained OPF model (LSCOPFM).

The overall autonomous formulation and solution process of the SCQOPFM is shown in Figure 11, in which N imported contingencies are included. The figure only shows a general flow of procedures. Every step in this process is discussed in detail in the following sections.

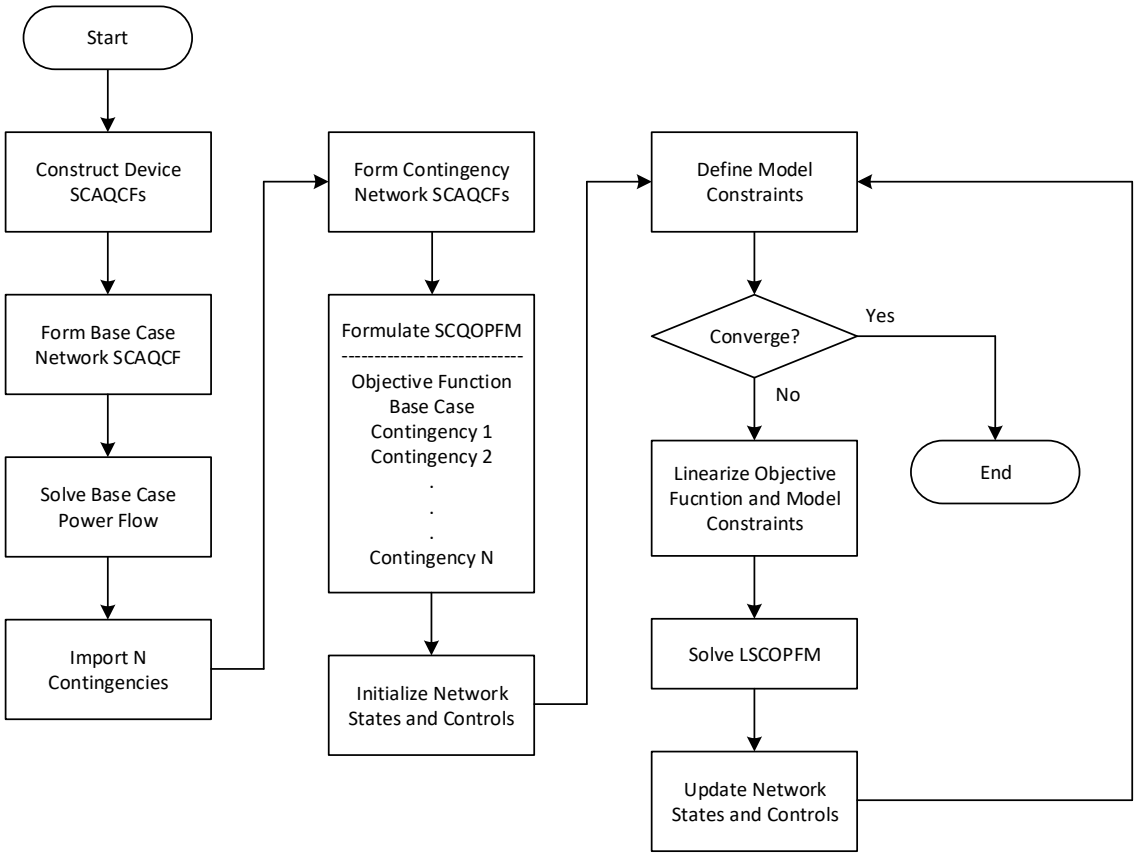

Figure 11 – Overall SCQOPFM formulation and solution process.

7.1 Frequency Domain Modeling

In the frequency domain, the modeling procedures are similar to those in the quasi-dynamic domain. The construction of device models follows the steps shown in Figure 3 except that quadratic integration is not applicable in this case because it is only for a single

instant in time. As a result, different from (12), the SCAQCF in frequency domain is derived from (8) as

$$\begin{aligned}
&\begin{bmatrix}\mathbf{i}\\0\\0\end{bmatrix} = Y_x\mathbf{x} + Y_u\mathbf{u} + \begin{Bmatrix}\vdots\\ \mathbf{x}^T F_{xx}^k \mathbf{x}\\ \vdots\end{Bmatrix} + \begin{Bmatrix}\vdots\\ \mathbf{u}^T F_{uu}^k \mathbf{u}\\ \vdots\end{Bmatrix} + \begin{Bmatrix}\vdots\\ \mathbf{u}^T F_{ux}^k \mathbf{x}\\ \vdots\end{Bmatrix} + K\\
&h = Y_{fx}\mathbf{x} + Y_{fu}\mathbf{u} + \begin{Bmatrix}\vdots\\ \mathbf{x}^T F_{fxx}^k \mathbf{x}\\ \vdots\end{Bmatrix} + \begin{Bmatrix}\vdots\\ \mathbf{u}^T F_{fuu}^k \mathbf{u}\\ \vdots\end{Bmatrix} + \begin{Bmatrix}\vdots\\ \mathbf{u}^T F_{fux}^k \mathbf{x}\\ \vdots\end{Bmatrix} + C_f \le 0\\
&\mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max}
\end{aligned} \tag{57}$$

with the algorithmic constraints on the control movements being $-\mathbf{u}_{\text{llim}} \le \mathbf{d} \le \mathbf{u}_{\text{llim}}$. The matrices and vectors are formed by those in (8) as follows.

$$\begin{aligned}
&Y_x = \begin{bmatrix}Y_{x1}\\Y_{x2}\\Y_{x3}\end{bmatrix} \quad Y_u = \begin{bmatrix}Y_{u1}\\Y_{u2}\\Y_{u3}\end{bmatrix} \quad K = \begin{bmatrix}C_1\\C_2\\C_3\end{bmatrix}\\
&F_{xx} = \begin{bmatrix}0\\0\\F_{x3}\end{bmatrix} \quad F_{uu} = \begin{bmatrix}0\\0\\F_{u3}\end{bmatrix} \quad F_{ux} = \begin{bmatrix}0\\0\\F_{ux3}\end{bmatrix}\\
&Y_{fx} = Y_{cx} \quad Y_{fu} = Y_{cu} \quad C_f = C_c\\
&F_{fxx} = F_{cx} \quad F_{fuu} = F_{cu} \quad F_{fux} = F_{cux}\\
&\mathbf{u}_{\min} = \mathbf{u}_{\text{cmin}} \quad \mathbf{u}_{\max} = \mathbf{u}_{\text{cmax}} \quad \mathbf{u}_{\text{llim}} = \mathbf{u}_{\text{cllim}}
\end{aligned} \tag{58}$$

Since (57) simulates the operation of the device at time instant t, there is no past history vector in the frequency domain SCAQCF. Note that (57) is regarded as the operational model of the device in the SCQOPF problem.

The frequency domain network SCAQCF model is formulated the same way as that shown in Figure 6 except that no past history vectors and device through vectors are

needed. Hence, the frequency domain network model is also given by (57), with the through variables being $\mathbf{i}_{net}$ instead of $\mathbf{i}$. This network model is called the base case model in the following SCQOPF problem.

7.2 Security Constrained Quadratic Optimal Power Flow Formulation

The formulation of a security constrained quadratic OPF problem is done in the frequency domain. It includes the modeling of devices and networks. Besides the original device and network models described in Section 7.1, the device outage and contingency network models are also required. When the objective function, base case network model and contingency network model are put together, the SCQOPFM is formed.

7.2.1 Device Outage Model

When a contingency network is formed, the outage devices need to be removed. Since all contingency network models are first copied directly from the base case network model, in order to maintain the network structure and make minimal changes, the concept of the device outage model is introduced. Instead of taking the outage device models completely out of a contingency network model, the original device operational models are changed to their outage models.

Every device in the network has an operational model and an outage model. Just like the operational model, the outage model is also written in the frequency domain SCAQCF standard syntax given in (57). The difference is that an outage control factor μ_c is added to the outage model. Its value is user-entered, which is usually a small percentage, denoting the influence of the device on the network. The device outage models are only

used in the contingency networks when the corresponding outage devices need to be removed.

The construction of a device outage model is the same as that of an operational model. The process goes through writing out the device compact model based on its physics, quadratization of the model, and eventually representing it in the frequency domain SCAQCF syntax. Only this time μ_c added into the model equations to reduce its operational impact on the network. Every type of device has μ_c included differently, depending on how the device operate. For a line or a transformer that is linear, μ_c is multiplied to all its terms. For a generator with real and reactive power outputs P_{gen} and Q_{gen}, they become $\mu_c P_{gen}$ and $\mu_c Q_{gen}$ in the outage model, thus reducing the generator outputs to a small percentage. In this case, when minimizing the generation costs, (27) with a_g being a constant in the operational model becomes (59) in the outage model.

$$0 = c_g \mu_c^2 P_{gen}^2 + b_g \mu_c P_{gen} + \mu_c a_g - C_{gen} \tag{59}$$

Note that (59) is an approximation of the cost of the outage generator. With μ_c being close to 0, C_{gen} ends up being a very small value as well. Hence, the impact of the outage generator on objective function (28) is close to none.

To construct the device outage model, not only the equations need to be changed, the functional constraints also require some modifications. When a device is removed from the contingency network, its constraints should also be neglected in solving the SCQOPF problem.

Using the same switched capacitor in Figure 5 as an example, its outage model in the frequency domain is similar to the SCQDM in (8), given by

$$
\begin{aligned}
&I_{1r} = -\omega C s_2 \\
&I_{1i} = \omega C s_1 \\
&I_{2r} = \omega C s_2 \\
&I_{2i} = -\omega C s_1 \\
&0 = s_1 - \mu_c V_{1r} u_s + \mu_c V_{2r} u_s \\
&0 = s_2 - \mu_c V_{1i} u_s + \mu_c V_{2i} u_s \\
&0 = y^2 - u_s + u_s^2 \\
&-V_{1r}^2 - V_{1i}^2 + V_{\min}^2 - \infty \le 0 \\
&V_{1r}^2 + V_{1i}^2 - V_{\max}^2 - \infty \le 0 \\
&0 \le u_s \le 1
\end{aligned}
\tag{60}
$$

Similarly, the device has to satisfy $0 \le d_s \le 1$ algorithmically.

In general, by introducing the device outage model and writing it in the frequency domain SCAQCF syntax, the contingency network models are formed automatically and the structure of the base case network is well maintained for more efficient computations in the following procedures.

7.2.2 Base Case and Contingencies

The SCQOPF base case network model is directly given by (53), whose power flow is solved. The base case solution is used to compute the initial operating states of the N contingencies imported. The contingency network models are constructed by first copying the base case model over and then remove the outage devices. Both the base case and contingency networks are modeled in the frequency domain SCAQCF syntax. For simplicity of representation, they are in general expressed as

$$
\begin{aligned}
& g^i\left(\mathbf{x}^i, \mathbf{u}\right) = 0 \\
& h^i\left(\mathbf{x}^i, \mathbf{u}\right) \le 0 \\
& \mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max}
\end{aligned} \tag{61}
$$

where $i = 0,1,\cdots,N$ is the network index. $i = 0$ denote the base case, while the model for contingency i is represented when $i \ge 1$. The control movement bounds are $-\mathbf{u}_{\text{llim}} \le \mathbf{d} \le \mathbf{u}_{\text{llim}}$. From (61) it can be noticed that every contingency network has its own states, equations and constraints. However, control vector $\mathbf{u}$ is shared by all network models. Thus, $\mathbf{u}_{\min}$, $\mathbf{u}_{\max}$ and $\mathbf{u}_{\text{llim}}$ are also shared among networks. Again, due to the networks being close, through vector $\mathbf{i}_{net}$ does not exist.

Since outage control factor μ_c defined for the device outage models only makes minimal changes to the network structure, the base case and contingency networks have almost the same structure.

7.2.3 Security Constrained Quadratic Optimal Power Flow Model

Objective functions in (24) and (28) also apply to the SCQOPF problem. With the objective function defined, stacking all the network models in (61) with $i = 0,1,\cdots,N$ together yields the SCQOPFM. Note that the objective is a function of the controls and the base case states only. The SCQOPFM is given by

$$
\begin{aligned}
\min \quad & J\left(\mathbf{x}^0, \mathbf{u}\right) \\
\text{s.t.} \quad & g^0\left(\mathbf{x}^0, \mathbf{u}\right) = 0 \\
& h^0\left(\mathbf{x}^0, \mathbf{u}\right) \le 0 \\
& g^1\left(\mathbf{x}^1, \mathbf{u}\right) = 0 \\
& h^1\left(\mathbf{x}^1, \mathbf{u}\right) \le 0 \\
& \vdots \\
& g^N\left(\mathbf{x}^N, \mathbf{u}\right) = 0 \\
& h^N\left(\mathbf{x}^N, \mathbf{u}\right) \le 0 \\
& \mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max}
\end{aligned}
\tag{62}
$$

where network 0 denotes the base case while networks 1 to N are contingencies. As a result, there are $N+1$ network models in total. Each network i has its own set of power flow equations $g^i\left(\mathbf{x}^i, \mathbf{u}\right) = 0$ and inequality constraints $h^i\left(\mathbf{x}^i, \mathbf{u}\right) \le 0$. All network models use the same set of controls $\mathbf{u}$, so there is only one set of control movement constraints $-\mathbf{u}_{\text{llim}} \le \mathbf{d} \le \mathbf{u}_{\text{llim}}$. When $\mathbf{u}$ is substituted into the equations in different networks, the state vectors computed are different.

Since the network models in (62) are defined in the frequency domain, the SCQOPFM can be viewed as an OPF problem that are constructed at time instant t including contingencies. The controls $\mathbf{u}$ are also for time t only.

7.3 Solution Method for Security Constrained Quadratic Optimal Power Flow

Figure 11 shows that the SCQOPF problem is solved through an SLP approach similar to the one used for the MQFOPF problem. However, instead of defining and linearizing the model constraints in multiple stages, the costate method linearizes the model

constraints defined in each network model. Combining all the linearized networks forms the LSCOPFM, which is solved by an LP solver. The solution is the control movement vector shared by all networks. The states of every network are updated independently after the update of controls in each iteration. The convergence criteria are the same as those defined in Figure 8, except that this time all constraints in all networks need to be satisfied to achieve SLP convergence.

7.3.1 Linearized Security Constrained Optimal Power Flow

At the beginning of each iteration ν, the algorithm goes through all the network models. For every network i, operating point $\left(\mathbf{x}^{i,\nu}, \mathbf{u}^{\nu}\right)$ is substituted into the constraints of that network model. Violated constraints that are not model constraints are added to the set. It not all the convergence criteria are met, the algorithm linearizes the model constraints in all networks as well as the objective function via the costate method.

Consider network model i given in (61). From (50) and (51), the following equations are valid.

$$\begin{aligned} h_m^i &\cong h_m^i\left(\mathbf{x}^{i,\nu}, \mathbf{u}^{\nu}\right) + \left(\frac{\partial h_m^i\left(\mathbf{x}^{i,\nu}, \mathbf{u}^{\nu}\right)}{\partial \mathbf{u}} - \hat{\mathbf{y}}^i \frac{\partial g^i\left(\mathbf{x}^{i,\nu}, \mathbf{u}^{\nu}\right)}{\partial \mathbf{u}} \right) \mathbf{d} \\ \hat{\mathbf{y}}^i &= \frac{\partial h_m^i\left(\mathbf{x}^{i,\nu}, \mathbf{u}^{\nu}\right)}{\partial \mathbf{x}^i} \left(\frac{\partial g^i\left(\mathbf{x}^{i,\nu}, \mathbf{u}^{\nu}\right)}{\partial \mathbf{x}^i} \right)^{-1} \end{aligned} \tag{63}$$

where according to (44) and (49) the partial derivatives derived from the frequency domain network model in (57) are given by

$$
\begin{aligned}
\frac{\partial g^i\left(\mathbf{x}^{i,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}} &= Y_u^i + \left\{\begin{matrix}\vdots\\ \left(\mathbf{u}^{\nu}\right)^T F_{uu}^{i,k}\\ \vdots\end{matrix}\right\} + \left\{\begin{matrix}\vdots\\ F_{uu}^{i,k}\,\mathbf{u}^{\nu}\\ \vdots\end{matrix}\right\}^T + \left\{\begin{matrix}\vdots\\ F_{ux}^{i,k}\,\mathbf{x}^{\nu}\\ \vdots\end{matrix}\right\}^T \\
\frac{\partial g^i\left(\mathbf{x}^{i,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{x}^i} &= Y_x^i + \left\{\begin{matrix}\vdots\\ \left(\mathbf{x}^{i,\nu}\right)^T F_{xx}^{i,k}\\ \vdots\end{matrix}\right\} + \left\{\begin{matrix}\vdots\\ F_{xx}^{i,k}\,\mathbf{x}^{i,\nu}\\ \vdots\end{matrix}\right\}^T + \left\{\begin{matrix}\vdots\\ \left(\mathbf{u}^{\nu}\right)^T F_{ux}^{i,k}\\ \vdots\end{matrix}\right\} \\
\frac{\partial h_m^i\left(\mathbf{x}^{i,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}} &= Y_{fu}^i + \left\{\begin{matrix}\vdots\\ \left(\mathbf{u}^{\nu}\right)^T F_{fuu}^{i,k}\\ \vdots\end{matrix}\right\} + \left\{\begin{matrix}\vdots\\ F_{fuu}^{i,k}\,\mathbf{u}^{\nu}\\ \vdots\end{matrix}\right\}^T + \left\{\begin{matrix}\vdots\\ F_{fux}^{i,k}\,\mathbf{x}^{\nu}\\ \vdots\end{matrix}\right\}^T \\
\frac{\partial h_m^i\left(\mathbf{x}^{i,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{x}^i} &= Y_{fx}^i + \left\{\begin{matrix}\vdots\\ \left(\mathbf{x}^{i,\nu}\right)^T F_{fxx}^{i,k}\\ \vdots\end{matrix}\right\} + \left\{\begin{matrix}\vdots\\ F_{fxx}^{i,k}\,\mathbf{x}^{i,\nu}\\ \vdots\end{matrix}\right\}^T + \left\{\begin{matrix}\vdots\\ \left(\mathbf{u}^{\nu}\right)^T F_{fux}^{i,k}\\ \vdots\end{matrix}\right\}
\end{aligned}
\tag{64}
$$

Note that $\partial g^i\left(\mathbf{x}^{i,\nu},\mathbf{u}^{\nu}\right)\big/\partial \mathbf{x}^i$ is the Jacobian matrix of network model i. Therefore, the linearized model constraints are expressed as

$$
a^i\mathbf{d} + b^i \le 0 \tag{65}
$$

where a^i and b^i are

$$
\begin{aligned}
a^i &= \frac{\partial h_m^i\left(\mathbf{x}^{i,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}} - \hat{\mathbf{y}}^i\,\frac{\partial g^i\left(\mathbf{x}^{i,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}} \\
b^i &= h_m^i\left(\mathbf{x}^{i,\nu},\mathbf{u}^{\nu}\right)
\end{aligned}
\tag{66}
$$

For the linearization of the objective function in (62), only the base case network states $\mathbf{x}^{0,\nu}$ are needed. Hence, similar to (45) and (46), the LSCOPFM has

$$
\begin{aligned}
J &\cong J\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)+\left(\frac{\partial J\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}}-\hat{\mathbf{x}}^{0}\frac{\partial g^{0}\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}}\right)\mathbf{d}\\
\hat{\mathbf{x}}^{0} &= \frac{\partial J\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{x}}\left(\frac{\partial g^{0}\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{x}}\right)^{-1}
\end{aligned}
\tag{67}
$$

where $\partial J\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)/\partial \mathbf{u}$ and $\partial J\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)/\partial \mathbf{x}^{0}$ are obtained as follows based on (41).

$$
\begin{aligned}
\frac{\partial J\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}} &= Y_{ou}+\left(\mathbf{u}^{\nu}\right)^{T}F_{ouu}+\left(F_{ouu}\mathbf{u}^{\nu}\right)^{T}+\left(F_{oux}\mathbf{x}^{0,\nu}\right)^{T}\\
\frac{\partial J\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{x}^{0}} &= Y_{ox}+\left(\mathbf{x}^{0,\nu}\right)^{T}F_{oxx}+\left(F_{oxx}\mathbf{x}^{0,\nu}\right)^{T}+\left(\mathbf{u}^{\nu}\right)^{T}F_{oux}
\end{aligned}
\tag{68}
$$

As a result, the objective function after linearization has the exact same expression as that the objective function in (53), but this time c^{T} and e are

$$
\begin{aligned}
c^{T} &= \frac{\partial J\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}}-\hat{\mathbf{x}}^{0}\frac{\partial g\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)}{\partial \mathbf{u}}\\
e &= J\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)
\end{aligned}
\tag{69}
$$

Since all networks share the same set of controls $\mathbf{u}$ as well as control movements $\mathbf{d}$, the minimum and maximum limit of control movement j are computed as

$$
\begin{aligned}
d_{\min,j} &= \max\left(u_{\min,j}-u_{j}^{\nu},-u_{\mathrm{llim},j}\right)\\
d_{\max,j} &= \min\left(u_{\max,j}-u_{j}^{\nu},u_{\mathrm{llim},j}\right)
\end{aligned}
\tag{70}
$$

With $N+1$ networks in total, the LSCOPFM is constructed by (66), (69) and (70) as

$$
\begin{aligned}
\min \quad & c^T \mathbf{d} + e \\
\text{s.t.} \quad & a^0 \mathbf{d} + b^0 \le 0 \\
& a^1 \mathbf{d} + b^1 \le 0 \\
& \vdots \\
& a^N \mathbf{d} + b^N \le 0 \\
& \mathbf{d}_{\min} \le \mathbf{d} \le \mathbf{d}_{\max}
\end{aligned}
\tag{71}
$$

The following steps are solving (71), updating the operating point in every network, adjusting the maximum control excursion limits, and checking for constraint violations as well as the algorithm convergence in the next iteration.

7.3.2 Acquisition of Optimal Solution

Since the SLP algorithms utilized to solve the MQFOPFM and SCQOPFM are basically the same, tolerances ε_0, ε_1 and ε_2 are also defined here. As for the binary convergence indicators, NonModelConstraints and ModelConstraints are defined and used the same way. Indicator LFOPFSolution is renamed as LSCOPFSolution in solving the SCQOPF problem.

After the LSCOPFM is obtained, it is solved in an LP solver with non-negative surplus variables introduced to guarantee feasibility, producing solution $\mathbf{d}$ and surplus variable values $\mathbf{s}$. Indicator LSCOPFSolution is set to 1 only when $\left\|\mathbf{s}^{\nu}\right\|_{\infty} < \varepsilon_2$. With controls $\mathbf{u}^{\nu+1} = \mathbf{u}^{\nu} + \mathbf{d}$, the SLP algorithm sweeps through all network models and computes the updated states via Newton's method described in Section 6.3.2. As a result, the new operating point for each network i is acquired as $\left(\mathbf{x}^{i,\nu}, \mathbf{u}^{\nu}\right)$. Then, the objective function value is computed using the base case operating point. Again, same as the solution

method for the MQFOPFM, for every continuous control j, its maximum control excursion limit $u_{\text{llim},j}$ is multiplied by ρ if $J\left(\mathbf{x}^{0,\nu+1},\mathbf{u}^{\nu+1}\right) \geq J\left(\mathbf{x}^{0,\nu},\mathbf{u}^{\nu}\right)$.

After the $\mathbf{u}_{\text{llim}}$ values are adjusted, the SLP algorithm proceeds to the next iteration. The inequality constraints of each network are checked for violations with respect to the corresponding operating point. NonModelConstraints = 1 if there is no new model constraint in all networks, or else NonModelConstraints = 0. Similarly, ModelConstraints is set to 1 when the model constraints in all networks are satisfied. If any model constraint in any network model is violated, ModelConstraints is 0. The SLP algorithm converges when LSCOPFSolution, NonModelConstraints and ModelConstraints are all 1.

The controls obtained upon algorithm convergence are the optimal control actions to be implemented into the system, which is able to operate safely and reliability. The occurrence of any contingency within the N contingencies considered in the SCQOPFM should not violate any constraint in the network.

7.4 Summary

In addition to the MQFOPFM, another OPF related application of the network SCAQCF model is to construct the SCQOPFM in the frequency domain, in which the device and network SCAQCF models only model their operations for one single time instant. The SCQOPF problem formulated contains both the base case and contingency network models. All devices in the network are operational in the base case, while outage devices exist in the contingencies. The imported contingency networks are the same as the base case network except that the outage devices are removed. Hence, the device outage

model is introduced with an outage control factor and it is also presented in the frequency domain SCAQCF standard syntax. By stacking up the base case and contingency network models, the SCQOPFM is constructed together with a user-defined objective function. Note that each network model has its own states, controls and constraints, while the same set of controls are shared among all networks. Compared to the MQFOPFM, the SCQOPFM has only one stage but many contingencies.

A similar SLP algorithm is used to solve the SCQOPF problem. Model constraints are defined in all networks. The costate method is applied to the objective function and each network model. Again, the use of model constraints and the costate method significantly reduces the size of the optimization problem to be solved. Combining the linearized network models with the linearized objective function yields the LSCOPFM. Solved by an LP solver, the LSCOPFM solution is the control movements, which are used to compute the controls shared by all networks. With the new control values, the states of every network model are updated via Newton's method. After the adjustment of the maximum control excursion limits, the SLP algorithm moves on to the next iteration. Convergence is checked after the constraint violations are tested. This time, the algorithm converges if the LSCOPFM has a feasible solution in the previous iteration and no constraint is violated in all network models.

CHAPTER 8. MULTI-STAGE QUADRATIC FLEXIBLE OPTIMAL POWER FLOW NUMERICAL CASE

A program has been written in Visual C++ to run the two OPF applications described in the above chapters. As shown in Figure 12, users need to select from the dropdown menu an OPF application and import the corresponding device models to start the process. The three tolerance values used in the SLP algorithms are also on the user interface for users to define.

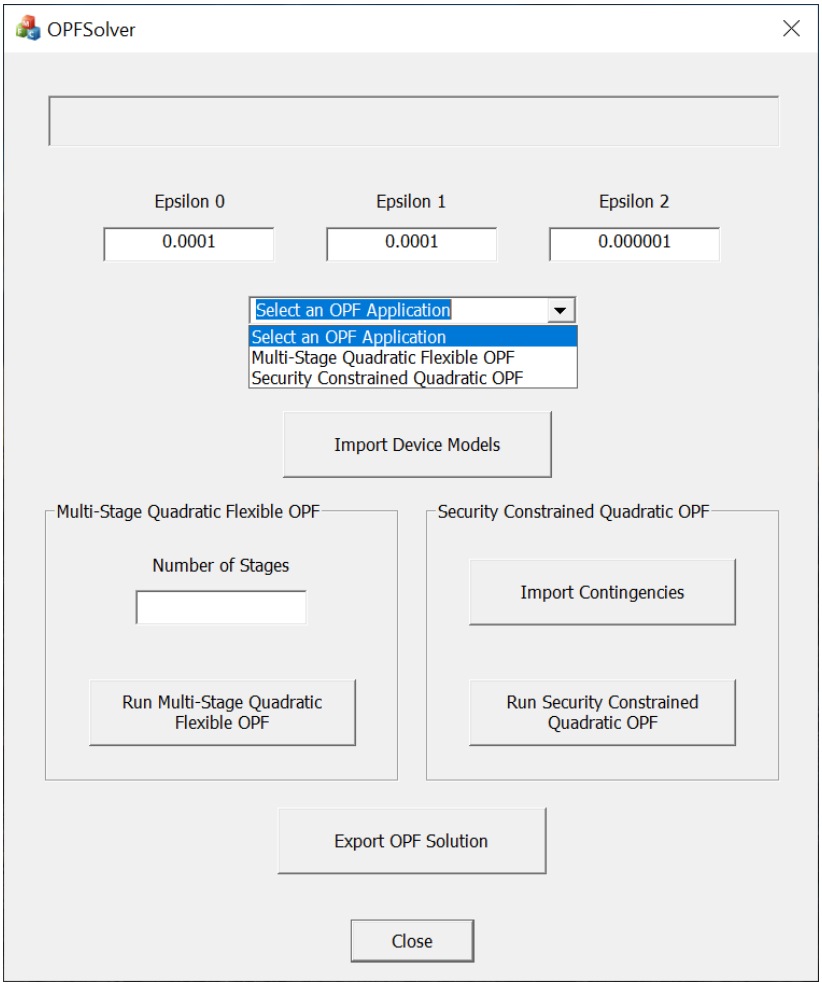

Figure 12 – Selection of OPF application on the user interface.

This chapter presents a numerical case to demonstrate the performance of the proposed method on the multi-stage quadratic flexible OPF application, which is run on a slightly modified real distribution feeder model provided by Public Service Company of New Mexico (PNM). The MQFOPFM of the feeder model is formulated in the quasi-dynamic domain, containing the device SCAQCF models in (12) and the voltage profile optimization objective in (24). Before running the algorithm, the number of stages need to be entered. Once the selected MQFOPF problem is solved, the user is able to export the final OPF solution.

8.1 Distribution Feeder Model Description

The PNM distribution feeder model shown in Figure 13 has 187 buses, including 177 three-phase buses and 10 single-phase buses. The distribution voltage is 12.47 kV which is stepped down from 115 kV at the transmission level. At the residential level, both 480 V at three-phase buses and 120/240 V at single-phase buses with split phases exist. For the sake of describing the system in more details, it is divided into 4 sections as illustrated in Figure 13.

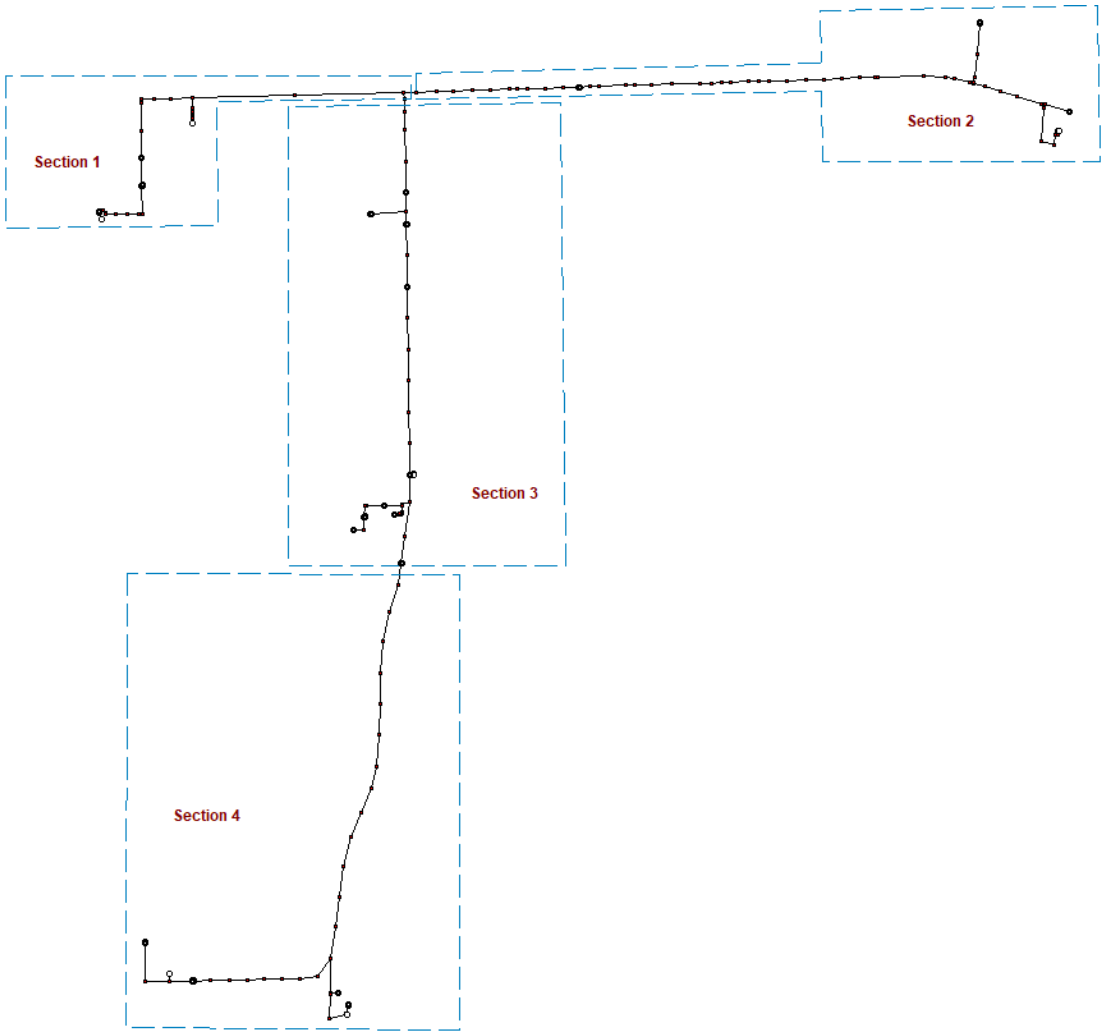

Figure 13 – Real PNM distribution feeder model.

The 10 single-phase buses are supplied by 10 single-phase transformers with a secondary center tap, which converts 12.47 kV at the distribution level to split-phase 120/240 V at the residential level. A 120 V load is connected to each split phase, so there are 20 single-phase loads in total. The number of three-phase loads is 10, including both 12.47 kV and 480 V aggregated loads. The system also has 4 PV sources, 4 energy storage units (ESUs), 1 three-phase equivalent source, 2 three-phase capacitor banks and 10 three-phase two-winding transformers. The PV sources and ESUs are distributed in section 1 and

section 4. The distribution lines include 159 three-phase lines and 7 single-phase lines modeled as equivalent Π circuits without neutral conductors. For voltage profile optimization, a fictious voltage magnitude device is added to each node selected for levelization, which adds up to 43 in total. The system is ground with a 1 Ω impedance at every bus with a neutral phase, resulting in 46 grounding impedances.

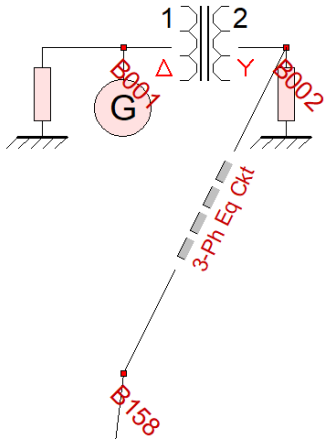

Figure 14 – Zoomed-in view of the slack bus.

The three-phase equivalent source is connected to slack bus B001 in section 2, simulating the 115 kV transmission system, as shown in Figure 14 a zoomed-in view. Both three-phase capacitor banks are located at B007 and B103 in section 1 of the system with a rated reactive power output of 1800 kVar. The PNM feeder loads are listed in Table 1. Note that the single-phase loads are connected to 120/240 V split-phase buses, with L1 and L2 representing the two 120 V nodes at those buses. The available controls are given in Table 2 with their initial values and maximum control excursion limits. For the control types, *tap* denotes the unitless transformer tap setting (ratio between primary and secondary voltages), P and Q are the real and reactive power outputs respectively, while V_{mag} represents the terminal voltage magnitude. All the controls are continuous variables.

Table 1 – PNM feeder loads.

	Bus		Voltage	Real Power	Reactive Power
Three-Phase Loads	B099		12.47 kV	1885 kW	1292 kVar
	B199		480.0 V	23.092 kW	3.290 kVar
	B200		480.0 V	92.367 kW	13.162 kVar
	B206		480.0 V	23.092 kW	3.290 kVar
	B208		480.0 V	46.184 kW	6.581 kVar
	B209		480.0 V	92.367 kW	13.162 kVar
	B211		480.0 V	230.918 kW	32.904 kVar
	B212		480.0 V	9.237 kW	1.316 kVar
	B213		480.0 V	89.1 kW	12.696 kVar
	B217		480.0 V	25.5 kW	19.2 kVar
Single-Phase Loads	B201	L1	120.0 V	1.539 kW	0.219 kVar
		L2	120.0 V	1.539 kW	0.219 kVar
	B202	L1	120.0 V	3.849 kW	0.548 kVar
		L2	120.0 V	3.849 kW	0.548 kVar
	B203	L1	120.0 V	3.849 kW	0.548 kVar
		L2	120.0 V	3.849 kW	0.548 kVar
	B204	L1	120.0 V	1.539 kW	0.219 kVar
		L2	120.0 V	1.539 kW	0.219 kVar
	B205	L1	120.0 V	3.849 kW	0.548 kVar
		L2	120.0 V	3.849 kW	0.548 kVar
	B207	L1	120.0 V	1.539 kW	0.219 kVar
		L2	120.0 V	1.539 kW	0.219 kVar
	B210	L1	120.0 V	1.539 kW	0.219 kVar
		L2	120.0 V	1.539 kW	0.219 kVar
	B214	L1	120.0 V	3.849 kW	0.548 kVar
		L2	120.0 V	3.849 kW	0.548 kVar
	B215	L1	120.0 V	1.539 kW	0.219 kVar
		L2	120.0 V	1.539 kW	0.219 kVar
	B216	L1	120.0 V	3.849 kW	0.548 kVar
		L2	120.0 V	3.849 kW	0.548 kVar

Table 2 – Available control variables in PNM feeder.

Device	Bus	Control	Bounds	Initial Value	Initial Maximum Excursion Limit
Transformer	B093 – B199	*tap*	0.9 – 1.1	1.0	0.02
Transformer	B089 – B200	*tap*	0.9 – 1.1	1.0	0.02
Transformer	B082 – B206	*tap*	0.9 – 1.1	1.0	0.02
Transformer	B092 – B208	*tap*	0.9 – 1.1	1.0	0.02
Transformer	B031 – B209	*tap*	0.9 – 1.1	1.0	0.02
Transformer	B098 – B211	*tap*	0.9 – 1.1	1.0	0.02
Transformer	B157 – B212	*tap*	0.9 – 1.1	1.0	0.02
Transformer	B133 – B213	*tap*	0.9 – 1.1	1.0	0.02
Transformer	B198 – B217	*tap*	0.9 – 1.1	1.0	0.02
Transformer	B001 – B002	*tap*	0.9 – 1.1	1.0	0.02
PV Source	B170	Q (MVar)	-1.0 – 1.0	0.0	0.25
PV Source	B196	Q (MVar)	-5.0 – 5.0	0.0	1.0
PV Source	B217	Q (MVar)	-0.5 – 0.5	0.0	0.05
PV Source	B025	Q (MVar)	-3.0 – 3.0	0.0	0.5
Energy Storage Unit	B170	P (MW)	-3.0 – 3.0	1.0	0.25
		V_{mag} (pu)	0.95 – 1.05	1.0	0.01
Energy Storage Unit	B196	P (MW)	-5.0 – 5.0	5.0	0.4
		V_{mag} (pu)	0.95 – 1.05	1.0	0.01
Energy Storage Unit	B217	P (MW)	-1.0 – 1.0	1.0	0.1
		V_{mag} (pu)	0.95 – 1.05	1.0	0.01
Energy Storage Unit	B025	P (MW)	-5.0 – 5.0	3.0	0.4
		V_{mag} (pu)	0.95 – 1.05	1.0	0.01

Within the horizon of 3 hours, the real power outputs P of the PV sources at B170, B196, B217 and B025 are fixed at 2.5 MW, 8 MW, 1 MW and 9 MW, respectively. It is assumed that the PV sources are able to control their reactive power outputs Q. Besides the control bounds, the PV sources and ESUs also have some other operational constraints

that are presented in Table 3, where S is the apparent power output computed by $S = |P + jQ|$ and E is the energy stored. Subscript max shows the maximum value allowed. The nominal voltage is denoted by V_{nom}. Table 3 also presents the parameters used in the constraints and their corresponding units, which are all in actual units. The initial energy stored in the ESUs at B170, B196, B217 and B025 are around 0.5, 0.8, 0.3 and 0.5 of their corresponding storage capacities $E_{\max}$, respectively.

Table 3 – Constraints in photovoltaic sources and energy storage units.

Device	Bus	Constraint	Parameter	Unit
PV Source	B170	$S \le S_{\max}$	$S_{\max} = 5.0$	MVA
		$0.95V_{nom} \le V_{mag} \le 1.05V_{nom}$	$V_{nom} = 12.47$	kV
PV Source	B196	$S \le S_{\max}$	$S_{\max} = 20.0$	MVA
		$0.95V_{nom} \le V_{mag} \le 1.05V_{nom}$	$V_{nom} = 12.47$	kV
PV Source	B217	$S \le S_{\max}$	$S_{\max} = 1.0$	MVA
		$0.95V_{nom} \le V_{mag} \le 1.05V_{nom}$	$V_{nom} = 480.0$	V
PV Source	B025	$S \le S_{\max}$	$S_{\max} = 10.0$	MVA
		$0.95V_{nom} \le V_{mag} \le 1.05V_{nom}$	$V_{nom} = 12.47$	kV
Energy Storage Unit	B170	$S \le S_{\max}$	$S_{\max} = 5.0$	MVA
		$0.1E_{\max} \le E \le 0.9E_{\max}$	$E_{\max} = 2.0$	MWh
Energy Storage Unit	B196	$S \le S_{\max}$	$S_{\max} = 8.0$	MVA
		$0.1E_{\max} \le E \le 0.9E_{\max}$	$E_{\max} = 5.0$	MWh
Energy Storage Unit	B217	$S \le S_{\max}$	$S_{\max} = 2.0$	MVA
		$0.1E_{\max} \le E \le 0.9E_{\max}$	$E_{\max} = 1.0$	MWh
Energy Storage Unit	B025	$S \le S_{\max}$	$S_{\max} = 8.0$	MVA
		$0.1E_{\max} \le E \le 0.9E_{\max}$	$E_{\max} = 4.0$	MWh

Every fictitious voltage magnitude device described by (25) also has a constraint on its selected voltage, which is $0.95 \le V_{mag} \le 1.05$ in per unit. Nodes with capacitor banks and loads are selected for voltage levelization because a certain range of voltage is required for the devices at those nodes to be operational. All the selected voltages in the PNM feeder are listed in Table 4, grouped together based on the feeder sections. As shown in (24), each selected voltage is presented with its target value V_{tar} and tolerance η. Although all the listed buses contain three phases, some of the buses only have one phase levelized when the phase is used to supply 2 split-phase loads at a split-phase bus. The steady-state values are the voltages obtained without optimization under the initial controls. Buses B010, B068, B076, B079, B106, B118, B130, B150, B155 and B169 are respectively connected to split-phase buses B201, B207, B202, B215, B205, B214, B216, B203, B204 and B210 through transformers. As an example, a zoomed-in view of the connections at B010 and B201 in section 2 are given in Figure 15, which includes 2 single-phase loads, 2 grounding impedances, 1 single-phase transformer and 1 fictitious voltage magnitude device.

Table 4 – Selected voltages to be levelized.

Section	Bus	Phase	Target Value	Tolerance	Steady-State Value
Section 1	B007	A	7.2 kV	5 %	7.202 kV
		B	7.2 kV	5 %	7.211 kV
		C	7.2 kV	5 %	7.199 kV
	B099	A	7.2 kV	5 %	7.197 kV
		B	7.2 kV	5 %	7.204 kV
		C	7.2 kV	5 %	7.194 kV
	B103	A	7.2 kV	5 %	7.201 kV
		B	7.2 kV	5 %	7.210 kV
		C	7.2 kV	5 %	7.198 kV

Section 2	B010	B	7.2 kV	5 %	6.884 kV
	B068	B	7.2 kV	5 %	6.890 kV
	B155	C	7.2 kV	5 %	6.871 kV
Section 3	B199	A	0.277 kV	5 %	0.2640 kV
		B	0.277 kV	5 %	0.2641 kV
		C	0.277 kV	5 %	0.2630 kV
	B200	A	0.277 kV	5 %	0.2638 kV
		B	0.277 kV	5 %	0.2639 kV
		C	0.277 kV	5 %	0.2628 kV
	B206	A	0.277 kV	5 %	0.2617 kV
		B	0.277 kV	5 %	0.2618 kV
		C	0.277 kV	5 %	0.2599 kV
	B213	A	0.277 kV	5 %	0.2617 kV
		B	0.277 kV	5 %	0.2619 kV
		C	0.277 kV	5 %	0.2601 kV
	B076	C	7.2 kV	5 %	6.857 kV
	B079	C	7.2 kV	5 %	6.773 kV
	B106	C	7.2 kV	5 %	6.840 kV
	B118	B	7.2 kV	5 %	6.835 kV
	B150	C	7.2 kV	5 %	6.803 kV
	B169	A	7.2 kV	5 %	6.836 kV
Section 4	B208	A	0.277 kV	5 %	0.2756 kV
		B	0.277 kV	5 %	0.2760 kV
		C	0.277 kV	5 %	0.2750 kV
	B209	A	0.277 kV	5 %	0.2742 kV
		B	0.277 kV	5 %	0.2747 kV
		C	0.277 kV	5 %	0.2737 kV
	B211	A	0.277 kV	5 %	0.2757 kV
		B	0.277 kV	5 %	0.2762 kV
		C	0.277 kV	5 %	0.2753 kV
	B212	A	0.277 kV	5 %	0.2753 kV
		B	0.277 kV	5 %	0.2757 kV
		C	0.277 kV	5 %	0.2747 kV
	B130	C	7.2 kV	5 %	7.167 kV

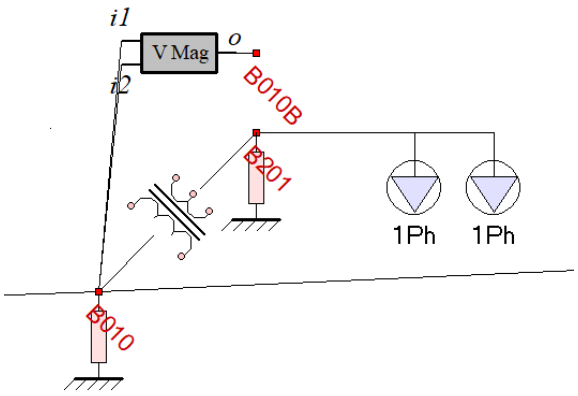

Figure 15 – Zoomed-in view of the connections at B010 and B201.

With the feeder model defined, the voltage profile optimization for the PNM network is performed. The number of stages is set to be 12 in this test case with time step size t_h being 15 minutes, so the optimization horizon is 3 hours. Since each stage contains both time t and t_m variables, with these setups, the MQFOPFM formulated has 67944 states and 528 controls. The network voltage profile at time t of all stages are optimized.

8.2 Multi-Stage Quadratic Flexible Optimal Power Flow Results

The three pre-defined small positive tolerance values used in the SLP algorithm are $\varepsilon_0 = 0.0001$, $\varepsilon_1 = 0.0001$ and $\varepsilon_0 = 0.000001$. Parameter ρ used to adjust the maximum control excursion limits in each iteration is set to be 0.8. After the device models are imported through the user interface in Figure 12 and 12 is entered as the number of stages, the MQFOPF is run on a personal laptop (i5-6300U, 8GB RAM). The whole process takes about 5.5 minutes, including constructing and solving the MQFOPFM. The optimal solution is obtained in 17 SLP iterations.

The optimal control values obtained upon convergence are given in Table 5. For every control in each stage, the time t and t_m values are the same as listed. The optimal voltage magnitudes at the levelized nodes at time t in all stages are listed in Table 6. Obviously, the selected voltages are successfully driven to their corresponding target values, which are very different from the steady-state values in Table 4. All the voltage magnitudes are well within the range of 0.95 – 1.05 pu.

Table 5 – Multi-stage quadratic flexible OPF optimal control values.

Section 1							
B170 PV Source	Stage	1	2	3	4	5	6
	Q (MVar)	-0.2446	-0.4084	0.3856	-0.4084	-0.4084	-0.8084
	Stage	7	8	9	10	11	12
	Q (MVar)	0.4430	-0.2403	0.0836	-0.6729	-0.8204	-0.9024
B196 PV Source	Stage	1	2	3	4	5	6
	Q (MVar)	0.4483	0.4483	0.4483	0.4483	0.4483	0.6122
	Stage	7	8	9	10	11	12
	Q (MVar)	0.6122	0.6122	0.4483	0.4483	0.4483	0.8170
B170 Energy Storage Unit	Stage	1	2	3	4	5	6
	P (MW)	0.3042	0.3042	0.3042	0.3042	0.1626	-0.2473
	V_{mag} (pu)	1.0022	1.0022	1.0022	1.0022	1.0022	1.0039
	Stage	7	8	9	10	11	12
	P (MW)	-0.2726	-0.2769	-0.2200	-0.3253	-0.3253	-0.2509
	V_{mag} (pu)	1.0039	1.0039	1.0022	1.0022	0. 9957	0.9994
B196 Energy Storage Unit	Stage	1	2	3	4	5	6
	P (MW)	0.9354	0.9354	0.9354	0.9354	0.9354	0.9354
	V_{mag} (pu)	1.0022	1.0022	1.0022	1.0022	1.0022	1.0005
	Stage	7	8	9	10	11	12
	P (MW)	0.9354	0.9354	0.9354	0.9354	1.1975	1.1975
	V_{mag} (pu)	1.0005	1.0005	1.0022	1.0022	1.0022	1.0047

Section 2							
B001 – B002 Transformer	Stage	1	2	3	4	5	6
	tap	1.0557	1.0557	1.0557	1.0557	1.0589	1.0516
	Stage	7	8	9	10	11	12
	tap	1.0516	1.0516	1.0557	1.0557	1.0426	1.0499
Section 3							
B093 – B199 Transformer	Stage	1	2	3	4	5	6
	tap	1.0159	1.0159	1.0159	1.0159	1.0159	1.0151
	Stage	7	8	9	10	11	12
	tap	1.0151	1.0151	1.0250	1.0250	1.0118	1.0118
B089 – B200 Transformer	Stage	1	2	3	4	5	6
	tap	1.0159	1.0159	1.0159	1.0159	1.0159	1.0151
	Stage	7	8	9	10	11	12
	tap	1.0151	1.0151	1.0250	1.0250	1.0118	1.0118
B082 – B206 Transformer	Stage	1	2	3	4	5	6
	tap	1.0250	1.0250	1.0250	1.0250	1.0118	1.0151
	Stage	7	8	9	10	11	12
	tap	1.0151	1.0151	1.0250	1.0250	1.0118	1.0118
B133 – B213 Transformer	Stage	1	2	3	4	5	6
	tap	1.0250	1.0250	1.0250	1.0250	1.0118	1.0250
	Stage	7	8	9	10	11	12
	tap	1.0250	1.0250	1.0118	1.0118	1.0118	1.0118
Section 4							
B092 – B208 Transformer	Stage	1	2	3	4	5	6
	tap	0.9779	0.9779	0.9779	0.9779	0.9779	0.9779
	Stage	7	8	9	10	11	12
	tap	0.9779	0.9779	0.9779	0.9779	0.9757	0.9757
B031 – B209 Transformer	Stage	1	2	3	4	5	6
	tap	0.9770	0.9770	0.9770	0.9770	0.9770	0.9770
	Stage	7	8	9	10	11	12
	tap	0.9770	0.9770	0.9770	0.9770	0.9770	0.9770

B098 – B211 Transformer	Stage	1	2	3	4	5	6
	tap	0.9738	0.9738	0.9738	0.9738	0.9738	0.9738
	Stage	7	8	9	10	11	12
	tap	0.9738	0.9738	0.9738	0.9738	0.9738	0.9738
B157 – B212 Transformer	Stage	1	2	3	4	5	6
	tap	0.9757	0.9757	0.9757	0.9757	0.9790	0.9790
	Stage	7	8	9	10	11	12
	tap	0.9790	0.9790	0.9757	0.9757	0.9757	0.9770
B198 – B217 Transformer	Stage	1	2	3	4	5	6
	tap	1.0112	1.0112	1.0112	1.0186	1.0054	1.0186
	Stage	7	8	9	10	11	12
	tap	1.0186	1.0186	1.0186	1.0186	1.0186	1.0061
B217 PV Source	Stage	1	2	3	4	5	6
	Q (MVar)	-0.0037	-0.0037	-0.0037	-0.0034	-0.0037	-0.0037
	Stage	7	8	9	10	11	12
	Q (MVar)	-0.0034	-0.0037	-0.0037	-0.0034	-0.0035	-0.0015
B025 PV Source	Stage	1	2	3	4	5	6
	Q (MVar)	-3.0000	-3.0000	-3.0000	-3.0000	-3.0000	-3.0000
	Stage	7	8	9	10	11	12
	Q (MVar)	-3.0000	-3.0000	-3.0000	-3.0000	-3.0000	-3.0000
B217 Energy Storage Unit	Stage	1	2	3	4	5	6
	P (MW)	-0.0162	-0.0162	-0.0162	0.0316	-0.0162	-0.0162
	V_{mag} (pu)	1.0210	1.0210	1.0210	1.0235	1.0301	1.0235
	Stage	7	8	9	10	11	12
	P (MW)	-0.0162	-0.0162	-0.0162	-0.0162	-0.0162	-0.0162
	V_{mag} (pu)	1.0235	1.0235	1.0235	1.0235	1.0235	0.9970
B025 Energy Storage Unit	Stage	1	2	3	4	5	6
	P (MW)	-0.0652	-0.0652	-0.0652	-0.0652	0.0167	0.0167
	V_{mag} (pu)	1.0310	1.0310	1.0310	1.0310	1.0392	1.0392
	Stage	7	8	9	10	11	12
	P (MW)	0.0167	0.0167	-0.0652	-0.0652	-0.2296	-0.5404
	V_{mag} (pu)	1.0392	1.0392	1.0310	1.0310	1.0310	1.0310

Table 6 – Optimal voltage magnitudes at levelized nodes.

Section 1							
Stage		1	2	3	4	5	6
B007	A	7.221 kV	7.221 kV	7.221 kV	7.221 kV	7.222 kV	7.225 kV
	B	7.230 kV	7.230 kV	7.230 kV	7.230 kV	7.230 kV	7.234 kV
	C	7.218 kV	7.218 kV	7.218 kV	7.218 kV	7.218 kV	7.222 kV
Stage		7	8	9	10	11	12
B007	A	7.225 kV	7.225 kV	7.222 kV	7.222 kV	7.190 kV	7.214 kV
	B	7.234 kV	7.234 kV	7.230 kV	7.230 kV	7.199 kV	7.223 kV
	C	7.222 kV	7.222 kV	7.218 kV	7.218 kV	7.187 kV	7.210 kV
Stage		1	2	3	4	5	6
B099	A	7.212 kV	7.212 kV	7.212 kV	7.212 kV	7.212 kV	7.224 kV
	B	7.221 kV	7.221 kV	7.221 kV	7.221 kV	7.221 kV	7.233 kV
	C	7.210 kV	7.210 kV	7.210 kV	7.210 kV	7.210 kV	7.221 kV
Stage		7	8	9	10	11	12
B099	A	7.224 kV	7.224 kV	7.212 kV	7.212 kV	7.166 kV	7.192 kV
	B	7.233 kV	7.233 kV	7.221 kV	7.221 kV	7.174 kV	7.201 kV
	C	7.221 kV	7.221 kV	7.210 kV	7.210 kV	7.163 kV	7.190 kV
Stage		1	2	3	4	5	6
B103	A	7.222 kV	7.222 kV	7.222 kV	7.222 kV	7.222 kV	7.223 kV
	B	7.230 kV	7.230 kV	7.230 kV	7.230 kV	7.231 kV	7.232 kV
	C	7.217 kV	7.217 kV	7.217 kV	7.217 kV	7.218 kV	7.219 kV
Stage		7	8	9	10	11	12
B103	A	7.223 kV	7.223 kV	7.222 kV	7.222 kV	7.195 kV	7.218 kV
	B	7.232 kV	7.232 kV	7.230 kV	7.230 kV	7.203 kV	7.227 kV
	C	7.219 kV	7.219 kV	7.217 kV	7.217 kV	7.191 kV	7.214 kV
Section 2							
Stage		1	2	3	4	5	6
B010	B	7.156 kV	7.156 kV	7.156 kV	7.155 kV	7.172 kV	7.155 kV
Stage		7	8	9	10	11	12
B010	B	7.155 kV	7.155 kV	7.160 kV	7.161 kV	7.136 kV	7.166 kV
Stage		1	2	3	4	5	6
B068	B	7.279 kV	7.279 kV	7.279 kV	7.279 kV	7.298 kV	7.268 kV
Stage		7	8	9	10	11	12
B068	B	7.268 kV	7.268 kV	7.285 kV	7.287 kV	7.227 kV	7.269 kV
Stage		1	2	3	4	5	6
B155	C	7.246 kV	7.246 kV	7.246 kV	7.245 kV	7.264 kV	7.236 kV

Stage		7	8	9	10	11	12
B155	C	7.236 kV	7.236 kV	7.252 kV	7.253 kV	7.199 kV	7.239 kV

Section 3

Stage		1	2	3	4	5	6
B199	A	277.02 V	277.02 V	277.02 V	276.98 V	277.75 V	277.10 V
	B	277.21 V	277.21 V	277.21 V	277.17 V	277.94 V	277.29 V
	C	276.03 V	276.03 V	276.03 V	276.00 V	276.76 V	276.11 V
Stage		7	8	9	10	11	12
B199	A	277.10 V	277.10 V	279.54 V	279.55 V	275.60 V	276.52 V
	B	277.29 V	277.29 V	279.73 V	279.75 V	275.79 V	276.71 V
	C	276.11 V	276.11 V	278.54 V	278.56 V	274.62 V	275.54 V
Stage		1	2	3	4	5	6
B200	A	276.97 V	276.97 V	276.97 V	276.94 V	277.74 V	277.09 V
	B	277.16 V	277.16 V	277.16 V	277.12 V	277.93 V	277.28 V
	C	275.94 V	275.94 V	275.94 V	275.90 V	276.70 V	276.06 V
Stage		7	8	9	10	11	12
B200	A	277.10 V	277.10 V	279.49 V	279.50 V	275.57 V	276.48 V
	B	277.28 V	277.29 V	279.67 V	279.69 V	275.75 V	276.66 V
	C	276.06 V	276.06 V	278.44 V	278.46 V	274.54 V	275.45 V
Stage		1	2	3	4	5	6
B206	A	279.38 V	279.38 V	279.38 V	279.34 V	277.13 V	277.73 V
	B	279.52 V	279.52 V	279.52 V	279.48 V	277.27 V	277.87 V
	C	277.47 V	277.47 V	277.47 V	277.42 V	275.23 V	275.83 V
Stage		7	8	9	10	11	12
B206	A	277.73 V	277.73 V	279.43 V	279.44 V	275.72 V	276.46 V
	B	277.87 V	277.87 V	279.57 V	279.58 V	275.86 V	276.60 V
	C	275.83 V	275.83 V	277.52 V	277.53 V	273.83 V	274.57 V
Stage		1	2	3	4	5	6
B213	A	279.42 V	279.42 V	279.42 V	279.37 V	277.17 V	280.43 V
	B	279.58 V	279.58 V	279.58 V	279.54 V	277.33 V	280.60 V
	C	277.65 V	277.65 V	277.65 V	277.60 V	275.41 V	278.65 V
Stage		7	8	9	10	11	12
B213	A	280.43 V	280.43 V	275.92 V	275.93 V	275.76 V	276.50 V
	B	280.60 V	280.60 V	276.08 V	276.09 V	275.92 V	276.66 V
	C	278.66 V	278.66 V	274.17 V	274.18 V	274.01 V	274.74 V
Stage		1	2	3	4	5	6
B076	C	7.092 kV	7.092 kV	7.092 kV	7.091 kV	7.112 kV	7.102 kV
Stage		7	8	9	10	11	12

B076	C	7.102 kV	7.102 kV	7.094 kV	7.094 kV	7.085 kV	7.108 kV
Stage		1	2	3	4	5	6
B079	C	7.070 kV	7.070 kV	7.070 kV	7.069 kV	7.104 kV	7.096 kV
Stage		7	8	9	10	11	12
B079	C	7.096 kV	7.096 kV	7.072 kV	7.072 kV	7.068 kV	7.086 kV
Stage		1	2	3	4	5	6
B106	C	7.087 kV	7.087 kV	7.087 kV	7.086 kV	7.110 kV	7.101 kV
Stage		7	8	9	10	11	12
B106	C	7.101 kV	7.101 kV	7.089 kV	7.089 kV	7.081 kV	7.103 kV
Stage		1	2	3	4	5	6
B118	B	7.123 kV	7.123 kV	7.123 kV	7.122 kV	7.156 kV	7.149 kV
Stage		7	8	9	10	11	12
B118	B	7.149 kV	7.149 kV	7.124 kV	7.124 kV	7.120 kV	7.139 kV
Stage		1	2	3	4	5	6
B150	C	7.093 kV	7.093 kV	7.093 kV	7.092 kV	7.129 kV	7.122 kV
Stage		7	8	9	10	11	12
B150	C	7.122 kV	7.122 kV	7.094 kV	7.094 kV	7.090 kV	7.108 kV
Stage		1	2	3	4	5	6
B169	A	7.118 kV	7.118 kV	7.118 kV	7.117 kV	7.151 kV	7.143 kV
Stage		7	8	9	10	11	12
B169	A	7.143 kV	7.143 kV	7.120 kV	7.120 kV	7.115 kV	7.135 kV

Section 4

Stage		1	2	3	4	5	6
B208	A	277.13 V	277.13 V	277.13 V	277.15 V	279.25 V	279.07 V
	B	277.53 V	277.53 V	277.53 V	277.54 V	279.65 V	279.47 V
	C	276.46 V	276.46 V	276.46 V	276.48 V	278.57 V	278.39 V
Stage		7	8	9	10	11	12
B208	A	279.08 V	279.08 V	277.10 V	277.11 V	276.46 V	276.48 V
	B	279.47 V	279.47 V	277.50 V	277.50 V	276.85 V	276.88 V
	C	278.39 V	278.39 V	276.43 V	276.43 V	275.78 V	275.80 V
Stage		1	2	3	4	5	6
B209	A	276.60 V	276.60 V	276.60 V	276.60 V	278.73 V	278.69 V
	B	277.05 V	277.05 V	277.05 V	277.04 V	279.18 V	279.13 V
	C	276.04 V	276.04 V	276.04 V	276.04 V	278.16 V	278.12 V
Stage		7	8	9	10	11	12
B209	A	278.69 V	278.69 V	276.60 V	276.60 V	276.59 V	276.66 V
	B	279.13 V	279.13 V	277.04 V	277.05 V	277.04 V	277.10 V
	C	278.12 V	278.12 V	276.04 V	276.04 V	276.03 V	276.09 V

Stage		1	2	3	4	5	6
B211	A	276.85 V	276.85 V	276.85 V	276.85 V	279.05 V	279.05 V
	B	277.33 V	277.33 V	277.33 V	277.33 V	279.53 V	279.53 V
	C	276.42 V	276.42 V	276.42 V	276.42 V	278.61 V	278.61 V
Stage		7	8	9	10	11	12
B211	A	279.05 V	279.05 V	276.85 V	276.85 V	276.86 V	276.86 V
	B	279.53 V	279.53 V	277.33 V	277.33 V	277.33 V	277.33 V
	C	278.61 V	278.61 V	276.42 V	276.42 V	276.42 V	276.42 V
Stage		1	2	3	4	5	6
B212	A	276.38 V	276.38 V	276.38 V	276.40 V	279.42 V	279.26 V
	B	276.78 V	276.78 V	276.78 V	276.80 V	279.82 V	279.66 V
	C	275.70 V	275.70 V	275.70 V	275.72 V	278.74 V	278.58 V
Stage		7	8	9	10	11	12
B212	A	279.26 V	279.26 V	276.36 V	276.36 V	276.35 V	276.74 V
	B	279.66 V	279.66 V	276.75 V	276.76 V	276.74 V	277.14 V
	C	278.58 V	278.58 V	275.68 V	275.68 V	275.67 V	276.06 V
Stage		1	2	3	4	5	6
B130	C	7.374 kV	7.374 kV	7.374 kV	7.375 kV	7.431 kV	7.426 kV
Stage		7	8	9	10	11	12
B130	C	7.426 kV	7.426 kV	7.374 kV	7.374 kV	7.373 kV	7.374 kV

The state of charge (SOC) of all ESUs across the optimization horizon of 3 hours (12 stages) are plotted in Figure 16. Note that the SOC of an ESU is defined as the ratio between its stored energy E and its storage capacity $E_{\max}$, so SOC is unitless between 0 and 1. Figure 17 shows how the MQFOPFM objective function value varies over the 17 iterations, with the initial value (without optimization) being 271.8253. The optimal value obtained upon convergence is 13.5207.

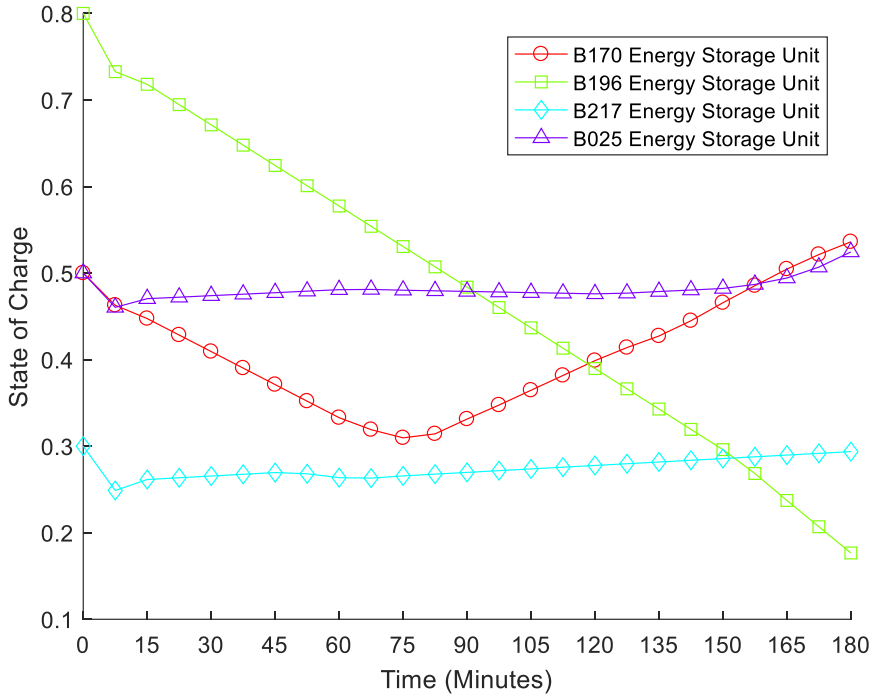

Figure 16 – State of charge of energy storage units across optimization horizon.

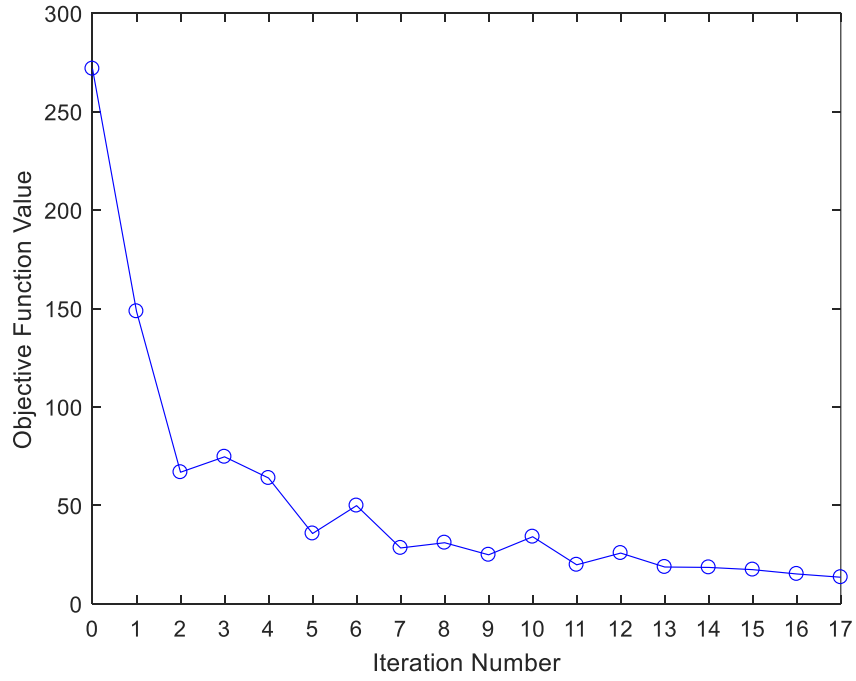

Figure 17 – MQFOPFM objective function value over iterations.

Since the optimization horizon is 3 hours but the algorithm only takes 5.5 minutes to run, the voltage profile of the PNM feeder can be optimized online. The algorithm performance with different optimization horizons is given in Table 7, which includes the problem dimensions, number of iterations and runtime in each case.

Table 7 – Algorithm performance with different optimization horizons.

Optimization Horizon	Number of Stages	Number of States	Number of Controls	Number of Iterations	Algorithm Runtime
1 hour	4	22648	176	24	33.687 s
2 hours	8	45296	352	19	126.553 s
3 hours	12	67944	528	17	322.880 s
4 hours	16	90592	704	24	906.689 s
5 hours	20	113240	880	30	2018.744 s
6 hours	24	135888	1056	30	3723.605 s

Since the PV source real power outputs may vary due to weather changes, the MQFOPF is able to incorporate this feature by optimizing the network voltage profile for a shorter period of time and using the model predictive control with moving optimization horizons.

For comparison, the problem is also solved with commercial nonlinear optimization software Knitro [80]. The MQFOPFM in (32) is imported to the Knitro solver as a quadratic constrained quadratic program. Since running Knitro C++ interface consumes a lot of memory, only a single-stage problem of the PNM feeder with 5662 states, 44 controls as well as 5882 equality and inequality constraints can be run on the personal laptop. However, even with this small problem that represents a horizon of 15 minutes, Knitro takes nearly 6.5 hours to solve, while SLP needs less than 1 second. A much smaller case

with the reduced-order model of the PNM feeder [81] that contains 15 buses is also tested using Knitro. The sing-stage problem of the reduced-order PNM feeder has 820 states, 18 controls and 922 optimization constraints, which takes 14.5 seconds for the Knitro embedded program to reach an optimal solution. When the number of stages is extended to 3 for this 15-bus system, Knitro returns an infeasible solution after 50 minutes of solving, while the SLP integrated MQFOPF algorithm takes only 0.5 seconds to reach an optimum.

8.3 Summary

This chapter demonstrates how the proposed method is used for the MQFOPF application, which is tested on an actual PNM feeder model. The algorithm is written in Visual C++.

The MQFOPF is used to optimize the voltage profile in the PNM feeder model. The MQFOPFM constructed contains 12 stages with an interval of 15 minutes, so the optimization horizon is 3 hours. The OPF problem is solved within 18 SLP iterations. The optimal controls obtained upon convergence are able to drive the PNM system to the optimal operating point that minimizes the objective function value and at the same time satisfies all constraints in the system. The whole process starting from forming the network model based on the object-oriented device models to obtaining the optimal solution takes around 5.5 minutes on a personal laptop. Tests are also performed to measure the time spent for cases with different number of stages, up to 6 hours of optimization horizon. The results show that the algorithm runtime is much less than the horizon, meaning that the optimization can be run online.

Commercial optimization software Knitro is used to optimize the voltage profile in the PNM feeder considering only 1 stage, but the solution time is way over 6 hours. Even for a much smaller case, the proposed method greatly outperforms Knitro in terms of speed. These comparisons indicate that the SLP solution algorithm developed is very efficient.

As a result, the proposed method for the MQFOPF application has been proven to be successful and promising. Moreover, the algorithm is able to exceed commercial nonlinear optimization software Knitro in both speed and solution quality.

CHAPTER 9. SECURITY CONSTRAINED QUADRATIC OPTIMAL POWER FLOW NUMERICAL CASE

In this chapter, the security constrained quadratic OPF is chosen from the dropdown menu in Figure 12. The frequency domain device models, including the device outage models, are imported after the three tolerance values used in the SLP algorithms are defined.

The SCQOPFM is constructed using (57) with the minimization of total base case generation cost in (28) as its objective. In this application, the contingencies to be considered are required before the algorithm can be run. The proposed method for the SCQOPF is tested on the IEEE 57-bus system. Since the original test case provided in [82] does not include any voltage or flow limits, some modifications have been made to the system to make it suitable for the proposed SCQOPF. Once the selected SCQOPF problem is solved, the user is able to export the final OPF solution.

9.1 IEEE 57-Bus System Description

The IEEE 57-bus system is shown in Figure 18, containing 42 loads, 65 lines, 15 transformers, 3 capacitor banks and 7 generators. Note that it is a 138 kV three-phase system assumed to be balanced, so in this example only the positive sequence system is used for simplicity, but all the data given are in their three-phase quantities.

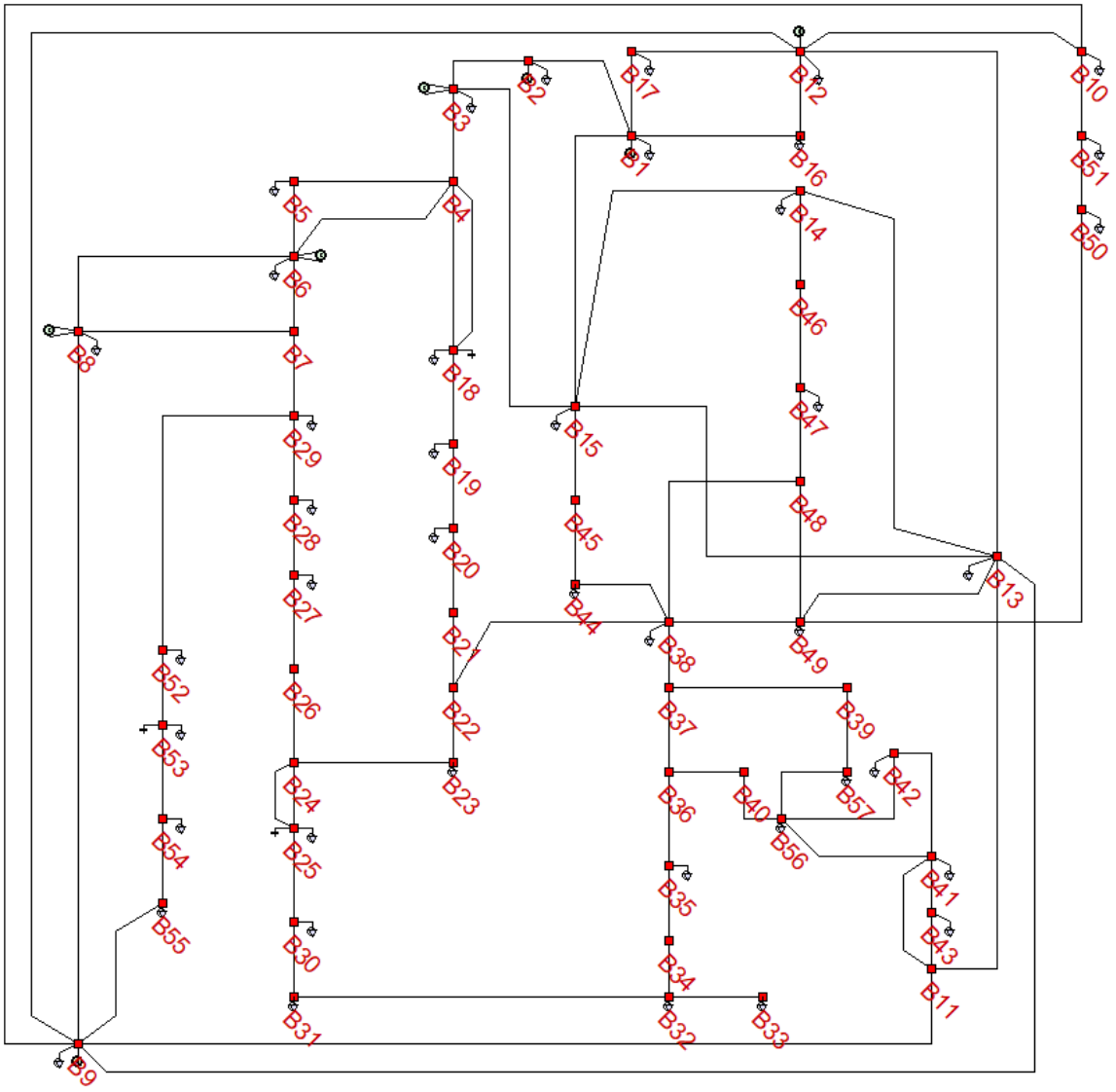

Figure 18 – IEEE 57-bus system.

B1 is the slack bus with the generator at B1 being the slack generator, so its voltage setpoint is controllable. The other 6 generators are controlled through their real and reactive power outputs. The transformer taps are also controllable. Hence, there are 28 continuous controls in the system and they are given in Table 8 together with their bounds, initial values and initial maximum excursion limits. The voltage setpoint is denoted by V_{mag} and the unitless transformer tap setting is represented by tap, while the real and reactive power

outputs are respectively P and Q. Note that there are two transformers connected in parallel between B4 and B18. Similarly, B24 and B25 are connected via two parallel lines.

Table 8 – Controls in IEEE 57-bus system.

Device	Bus	Control	Bounds	Initial Value	Initial Maximum Excursion Limit
Generator	B1	V_{mag}	0.95 – 1.05	1.04	0.01
Generator	B2	P (MW)	0.0 – 100.0	0.0	5.0
		Q (MVar)	-17.0 – 50.0	-0.755	2.5
Generator	B3	P (MW)	0.0 – 140.0	40.0	7.0
		Q (MVar)	-10.0 – 60.0	-0.905	3.0
Generator	B6	P (MW)	0.0 – 100.0	0.0	5.0
		Q (MVar)	-8.0 – 25.0	0.871	1.25
Generator	B8	P (MW)	0.0 – 550.0	450.0	27.5
		Q (MVar)	-140.0 – 200.0	62.1	10.0
Generator	B9	P (MW)	0.0 – 100.0	0.0	5.0
		Q (MVar)	-3.0 – 9.0	2.288	0.45
Generator	B12	P (MW)	100.0 – 410.0	310.0	20.5
		Q (MVar)	-150.0 – 155.0	128.631	7.75
Transformer	B4 – B18	*tap*	0.9 – 1.1	0.978	0.02
Transformer	B4 – B18	*tap*	0.9 – 1.1	0.97	0.02
Transformer	B7 – B29	*tap*	0.9 – 1.1	0.967	0.02
Transformer	B9– B55	*tap*	0.9 – 1.1	0.94	0.02
Transformer	B10 – B51	*tap*	0.9 – 1.1	0.93	0.02
Transformer	B11 – B41	*tap*	0.9 – 1.1	0.955	0.02
Transformer	B11 – B43	*tap*	0.9 – 1.1	0.958	0.02
Transformer	B13 – B49	*tap*	0.9 – 1.1	0.9	0.02
Transformer	B14 – B46	*tap*	0.9 – 1.1	0.9	0.02
Transformer	B15 – B45	*tap*	0.9 – 1.1	0.955	0.02
Transformer	B21 – B20	*tap*	0.9 – 1.1	1.043	0.02

Transformer	B24 – B26	*tap*	0.9 – 1.1	1.043	0.02
Transformer	B34 – B32	*tap*	0.9 – 1.1	0.975	0.02
Transformer	B39 – B57	*tap*	0.9 – 1.1	0.98	0.02
Transformer	B40 – B56	*tap*	0.9 – 1.1	0.958	0.02

Table 9 shows the loading condition of the test system, including the location as well as the real and reactive power consumptions of every load. The 3 capacitor banks are located at B18, B25 and B53 with respective ratings of 10 MVar, 5.9 MVar and 6.3 MVar.

Table 9 – IEEE 57-bus system load information.

Bus	B1	B2	B3	B5	B6	B8
Load	55.0 MW 17.0 MVar	3.0 MW 88.0 MVar	41.0 MW 21.0 MVar	13.0 MW 4.0 MVar	75.0 MW 2.0 MVar	150.0 MW 22.0 MVar
Bus	B9	B10	B12	B13	B14	B15
Load	121.0 MW 26.0 MVar	5.0 MW 2.0 MVar	377.0 MW 24.0 MVar	18.0 MW 2.3 MVar	10.5 MW 5.3 MVar	22.0 MW 5.0 MVar
Bus	B16	B17	B18	B19	B20	B23
Load	43.0 MW 3.0 MVar	42.0 MW 8.0 MVar	27.2 MW 9.8 MVar	3.3 MW 0.6 MVar	2.3 MW 1.0 MVar	6.3 MW 2.1 MVar
Bus	B25	B27	B28	B29	B30	B31
Load	6.3 MW 3.2 MVar	9.3 MW 0.5 MVar	4.6 MW 2.3 MVar	17.0 MW 2.6 MVar	3.6 MW 1.8 MVar	5.8MW 2.9 MVar
Bus	B32	B33	B35	B38	B41	B42
Load	1.6 MW 0.8 MVar	3.8 MW 1.9 MVar	6.0 MW 3.0 MVar	14.0 MW 7.0 MVar	6.3 MW 3.0 MVar	7.1 MW 4.4 MVar
Bus	B43	B44	B47	B49	B50	B51
Load	2.0 MW 1.0 MVar	12.0 MW 1.8 MVar	29.7 MW 11.6 MVar	18.0 MW 8.5 MVar	21.0 MW 10.5 MVar	18.0 MW 5.3 MVar
Bus	B52	B53	B54	B55	B56	B57
Load	4.9 MW 2.2 MVar	20.0 MW 10.0 MVar	4.1 MW 1.4 MVar	6.8 MW 3.4 MVar	7.6 MW 2.2 MVar	6.7 MW 2.0 MVar

The SCQOPFM is subject to a number of constraints, including bus voltage limits and branch flow limits. The voltage at each bus with either a generator or a load ranges from 0.95 pu to 1.05 pu. As for the branch flow limits defined in Table 10, the lines have current limits while the transformers have power limits. The current or power flowing into a line or a transformer from either side of the device has to satisfy the limits. The normal and contingency limits are the maximum amounts of flows allowed in the base case and contingencies, respectively. In this example, the branch flows have contingency limits that are 25 % higher than their normal limits. The terminal voltages of generators and loads must be between 0.95 pu and 1.05 pu in both base case and contingencies. For the slack generator at B1, its real and reactive power outputs are not controllable, but they are constrained within ranges 0 – 575.88 MW and -140 – 200 MVar, respectively.

Table 10 – Branch flow limits.

Line					
Location	Normal Limit	Contingency Limit	Location	Normal Limit	Contingency Limit
B1 – B2	915.81 A	1144.77 A	B1 – B15	826.40 A	1033.00 A
B1 – B16	825.75 A	1032.19 A	B1 – B17	610.03 A	762.54 A
B2 – B3	835.17 A	1043.96 A	B3 – B4	344.81 A	431.01 A
B3 – B15	357.04 A	446.30 A	B4 – B5	94.61 A	118.26 A
B4 – B6	147.72 A	184.65 A	B5 – B6	51.33 A	64.16 A
B6 – B7	87.57 A	109.47 A	B6 – B8	268.26 A	335.33 A
B7 – B8	955.42 A	1194.27 A	B8 – B9	1980.80 A	2476.00 A
B9 – B10	174.44 A	218.05 A	B9 – B11	170.56 A	213.20 A
B9 – B12	79.79 A	99.74 A	B9 – B13	51.74 A	64.68 A
B10 – B12	345.01 A	431.26 A	B11 – B13	130.04 A	162.55 A
B12 – B13	319.10 A	398.88 A	B12 – B16	260.94 A	326.17 A
B12 – B17	314.59 A	393.23 A	B13 – B14	291.22 A	364.02 A

B13 – B15	389.33 A	486.66 A	B14 – B15	405.02 A	506.27 A
B18 – B19	26.71 A	33.39 A	B19 – B20	19.17 A	23.96 A
B21 – B22	8.40 A	10.50 A	B22 – B23	52.89 A	66.11 A
B22 – B38	121.77 A	152.22 A	B23 – B24	33.66 A	42.08 A
B24 – B25	51.06 A	63.82 A	B24 – B25	62.82 A	78.53 A
B25 – B30	44.93 A	56.16 A	B26 – B27	72.51 A	90.64 A
B27 – B28	110.95 A	138.69 A	B28 – B29	149.09 A	186.36 A
B29 – B52	110.99 A	138.74 A	B30 – B31	38.53 A	48.16 A
B31 – B32	9.89 A	12.37 A	B32 – B33	53.30 A	66.63 A
B34 – B35	71.34 A	89.18 A	B35 – B36	128.60 A	160.75 A
B36 – B37	142.41 A	178.02 A	B36 – B40	58.80 A	73.50 A
B37 – B38	182.54 A	228.18A	B37 – B39	24.45 A	30.56 A
B38 – B44	276.53 A	345.67 A	B38 – B48	183.29 A	229.11 A
B38 – B49	62.30 A	77.88 A	B41 – B42	51.82 A	64.78 A
B41 – B43	152.86 A	191.07 A	B41 – B56	72.81 A	91.01 A
B42 – B56	21.50 A	26.88 A	B44 – B45	174.65 A	218.31 A
B46 – B47	317.42 A	396.78 A	B47 – B48	150.16 A	187.69 A
B48 – B49	68.70 A	85.88 A	B49 – B50	83.17 A	103.96 A
B50 – B51	64.08 A	80.10 A	B52 – B53	72.25 A	90.31 A
B53 – B54	87.33 A	109.17 A	B54 – B55	137.95 A	172.44 A
B56 – B57	29.84 A	37.30 A			

Transformer

Location	Normal Limit	Contingency Limit	Location	Normal Limit	Contingency Limit
B4 – B18	49.95 MVA	62.44 MVA	B4 – B18	37.44 MVA	46.81 MVA
B7 – B29	82.20 MVA	102.75 MVA	B9 – B55	25.48 MVA	31.85 MVA
B10 – B51	33.31 MVA	41.63 MVA	B11 – B41	31.62 MVA	39.52 MVA
B11 – B43	41.06 MVA	51.33 MVA	B13 – B49	118.49 MVA	148.12 MVA
B14 – B46	87.07 MVA	108.84 MVA	B15 – B45	50.35 MVA	62.94 MVA
B21 – B20	6.48 MVA	8.10 MVA	B24 – B26	21.64 MVA	27.05 MVA
B34 – B32	26.01 MVA	32.51 MVA	B39 – B57	10.85 MVA	13.56 MVA
B40 – B56	7.16 MVA	8.95 MVA			

The objective function of the example SCQOPFM is the minimization of the total generation cost given by (28). Different generators have different quadratic cost functions, which are defined by cost coefficients a_g, b_g and c_g in (27). The cost coefficients chosen for each generator that come from the MATPOWER case [83] are listed in Table 11.

Table 11 – Generator cost coefficients.

Generator	a_g (\$/hr)	b_g (\$/MW/hr)	c_g (\$/MW^2/hr)
Generator B1	0.0	20.0	0.07758
Generator B2	0.0	40.0	0.01
Generator B3	0.0	20.0	0.25
Generator B6	0.0	40.0	0.01
Generator B8	0.0	20.0	0.02222
Generator B9	0.0	40.0	0.01
Generator B12	0.0	20.0	0.03226

In this test case, 5 contingencies are considered. Every contingency has a device outage, which is given in Table 12. The outage control factor μ_c is chosen to be 0.000001 for constructing the device outage models.

Table 12 – Contingencies considered in security constrained quadratic OPF.

Contingency	1	2	3	4	5
Outage Device	Line B8 – B9	Line B9 – B12	Line B12 – B13	Generator B3	Transformer B15 – B45

As a result, the example SCQOPFM constructed contains 6 networks, including 1 base case and 5 contingencies. Each network model has 446 states. The 28 controls are shared among the 6 networks. The optimization is performed for only one single time

instant in the frequency domain. The objective is minimizing the total generation cost in the base case while satisfying the constraints in both the base case and the selected contingencies.

9.2 Security Constrained Quadratic Optimal Power Flow Results

The tolerance values defined for the SLP algorithm are $\varepsilon_0 = 0.0001$, $\varepsilon_1 = 0.0001$ and $\varepsilon_0 = 0.000001$. Parameter ρ used to adjust the maximum control excursion limits in each iteration is 0.8. With the frequency domain device models imported through the user interface in Figure 12, information of the 5 contingencies is also imported before the security constrained quadratic OPF is run. The whole SCQOPFM formulation and solution process takes about 1 second on a personal laptop (i5-6300U, 8GB RAM). The optimal solution is obtained at iteration 37, with the optimal control values listed in Table 13. Each one of them is within its allowable control range. It is noticeable from Table 8 that 6 generator controls and 1 transformer tap setting are at their upper bounds.

Table 13 – Optimal control values for IEEE 57-bus system.

Device	Location	Control	Optimal Value
Generator	B1	Voltage Setpoint	1.05 pu
Generator	B2	Real Power Output	100.0 MW
		Reactive Power Output	50.0 MVar
Generator	B3	Real Power Output	74.0102 MW
		Reactive Power Output	38.6257 MVar
Generator	B6	Real Power Output	22.6886 MW
		Reactive Power Output	22.6227 MVar
Generator	B8	Real Power Output	310.2826 MW
		Reactive Power Output	23.9020 MVar

Generator	B9	Real Power Output	100.0 MW
		Reactive Power Output	9.0 MVar
Generator	B12	Real Power Output	410.0 MW
		Reactive Power Output	65.7106 MVar
Transformer	B4 – B18	Tap Setting	0.9765
Transformer	B4 – B18	Tap Setting	0.9688
Transformer	B7 – B29	Tap Setting	0.9824
Transformer	B9– B55	Tap Setting	0.9501
Transformer	B10 – B51	Tap Setting	0.9894
Transformer	B11 – B41	Tap Setting	0.9097
Transformer	B11 – B43	Tap Setting	0.9484
Transformer	B13 – B49	Tap Setting	0.9029
Transformer	B14 – B46	Tap Setting	0.9314
Transformer	B15 – B45	Tap Setting	0.9245
Transformer	B21 – B20	Tap Setting	1.0615
Transformer	B24 – B26	Tap Setting	1.1000
Transformer	B34 – B32	Tap Setting	0.9449
Transformer	B39 – B57	Tap Setting	0.9589
Transformer	B40 – B56	Tap Setting	0.9823

The bus voltage magnitudes across the example test system under optimal operating conditions in all networks are given in Table 14. The system voltage profile is different in different networks. The voltages posted in Table 14 are also plotted in Figure 19 to better show the voltage profile differences between networks.

Table 14 – Optimal bus voltage magnitudes in base case and contingencies.

Bus	Base Case	Contingency				
		1	2	3	4	5
B1	1.0500 pu	1.0500 pu	1.0500 pu	1.0500 pu	1.0500 pu	1.0500 pu

B2	1.0465 pu	1.0463 pu	1.0455 pu	1.0437 pu	1.0368 pu	1.0456 pu
B3	1.0327 pu	1.0320 pu	1.0289 pu	1.0215 pu	0.9942 pu	1.0291 pu
B4	1.0266 pu	1.0303 pu	1.0217 pu	1.0149 pu	0.9896 pu	1.0208 pu
B5	1.0196 pu	1.0316 pu	1.0130 pu	1.0076 pu	0.9854 pu	1.0114 pu
B6	1.0215 pu	1.0373 pu	1.0142 pu	1.0094 pu	0.9889 pu	1.0122 pu
B7	1.0090 pu	1.0276 pu	1.0002 pu	0.9963 pu	0.9784 pu	0.9974 pu
B8	1.0197 pu	1.0453 pu	1.0101 pu	1.0077 pu	0.9913 pu	1.0087 pu
B9	1.0088 pu	0.9779 pu	0.9978 pu	0.9970 pu	0.9828 pu	0.9976 pu
B10	1.0158 pu	0.9931 pu	1.0091 pu	1.0184 pu	0.9928 pu	1.0044 pu
B11	1.0008 pu	0.9779 pu	0.9927 pu	0.9811 pu	0.9758 pu	0.9889 pu
B12	1.0307 pu	1.0144 pu	1.0272 pu	1.0500 pu	1.0117 pu	1.0220 pu
B13	1.0058 pu	0.9894 pu	1.0002 pu	0.9796 pu	0.9823 pu	0.9954 pu
B14	1.0021 pu	0.9892 pu	0.9971 pu	0.9808 pu	0.9778 pu	0.9919 pu
B15	1.0177 pu	1.0098 pu	1.0137 pu	1.0023 pu	0.9935 pu	1.0156 pu
B16	1.0302 pu	1.0179 pu	1.0276 pu	1.0444 pu	1.0161 pu	1.0238 pu
B17	1.0327 pu	1.0260 pu	1.0314 pu	1.0404 pu	1.0251 pu	1.0293 pu
B18	1.0468 pu	1.0499 pu	1.0413 pu	1.0333 pu	1.0074 pu	1.0381 pu
B19	1.0027 pu	0.9976 pu	0.9969 pu	0.9864 pu	0.9681 pu	0.9802 pu
B20	0.9873 pu	0.9774 pu	0.9813 pu	0.9694 pu	0.9560 pu	0.9567 pu
B21	1.0377 pu	1.0266 pu	1.0312 pu	1.0172 pu	1.0070 pu	1.0014 pu
B22	1.0366 pu	1.0243 pu	1.0300 pu	1.0157 pu	1.0069 pu	0.9983 pu
B23	1.0360 pu	1.0242 pu	1.0293 pu	1.0151 pu	1.0060 pu	0.9980 pu
B24	1.0401 pu	1.0370 pu	1.0320 pu	1.0199 pu	1.0071 pu	1.0076 pu
B25	1.0302 pu	1.0258 pu	1.0215 pu	1.0078 pu	0.9944 pu	0.9940 pu
B26	0.9478 pu	0.9443 pu	0.9404 pu	0.9298 pu	0.9178 pu	0.9186 pu
B27	0.9812 pu	0.9912 pu	0.9726 pu	0.9657 pu	0.9500 pu	0.9615 pu
B28	1.0002 pu	1.0157 pu	0.9913 pu	0.9859 pu	0.9688 pu	0.9845 pu
B29	1.0160 pu	1.0355 pu	1.0070 pu	1.0025 pu	0.9846 pu	1.0030 pu
B30	1.0124 pu	1.0064 pu	1.0036 pu	0.9893 pu	0.9760 pu	0.9749 pu

B31	0.9895 pu	0.9800 pu	0.9807 pu	0.9655 pu	0.9528 pu	0.9500 pu
B32	1.0063 pu	0.9919 pu	0.9980 pu	0.9822 pu	0.9711 pu	0.9657 pu
B33	1.0042 pu	0.9897 pu	0.9958 pu	0.9800 pu	0.9689 pu	0.9635 pu
B34	0.9884 pu	0.9744 pu	0.9813 pu	0.9663 pu	0.9575 pu	0.9513 pu
B35	0.9957 pu	0.9810 pu	0.9886 pu	0.9737 pu	0.9651 pu	0.9587 pu
B36	1.0053 pu	0.9903 pu	0.9983 pu	0.9835 pu	0.9752 pu	0.9686 pu
B37	1.0130 pu	0.9982 pu	1.0061 pu	0.9914 pu	0.9831 pu	0.9758 pu
B38	1.0375 pu	1.0239 pu	1.0311 pu	1.0165 pu	1.0084 pu	0.9982 pu
B39	1.0109 pu	0.9958 pu	1.0040 pu	0.9892 pu	0.9810 pu	0.9741 pu
B40	1.0040 pu	0.9884 pu	0.9969 pu	0.9821 pu	0.9738 pu	0.9679 pu
B41	1.0430 pu	1.0192 pu	1.0345 pu	1.0209 pu	1.0144 pu	1.0244 pu
B42	1.0098 pu	0.9882 pu	1.0012 pu	0.9870 pu	0.9797 pu	0.9853 pu
B43	1.0500 pu	1.0260 pu	1.0414 pu	1.0287 pu	1.0228 pu	1.0354 pu
B44	1.0499 pu	1.0374 pu	1.0439 pu	1.0297 pu	1.0213 pu	0.9939 pu
B45	1.0863 pu	1.0761 pu	1.0812 pu	1.0680 pu	1.0591 pu	0.9941 pu
B46	1.0644 pu	1.0498 pu	1.0586 pu	1.0419 pu	1.0374 pu	1.0473 pu
B47	1.0463 pu	1.0313 pu	1.0401 pu	1.0242 pu	1.0182 pu	1.0210 pu
B48	1.0435 pu	1.0287 pu	1.0372 pu	1.0219 pu	1.0152 pu	1.0140 pu
B49	1.0465 pu	1.0289 pu	1.0401 pu	1.0256 pu	1.0190 pu	1.0230 pu
B50	1.0199 pu	0.9998 pu	1.0132 pu	1.0063 pu	0.9932 pu	1.0000 pu
B51	1.0257 pu	1.0022 pu	1.0189 pu	1.0256 pu	1.0019 pu	1.0135 pu
B52	0.9902 pu	0.9949 pu	0.9800 pu	0.9762 pu	0.9579 pu	0.9768 pu
B53	0.9826 pu	0.9803 pu	0.9719 pu	0.9684 pu	0.9502 pu	0.9691 pu
B54	1.0119 pu	0.9924 pu	1.0008 pu	0.9984 pu	0.9815 pu	0.9991 pu
B55	1.0500 pu	1.0147 pu	1.0386 pu	1.0374 pu	1.0220 pu	1.0381 pu
B56	1.0066 pu	0.9879 pu	0.9983 pu	0.9838 pu	0.9761 pu	0.9771 pu
B57	1.0040 pu	0.9868 pu	0.9958 pu	0.9810 pu	0.9730 pu	0.9713 pu

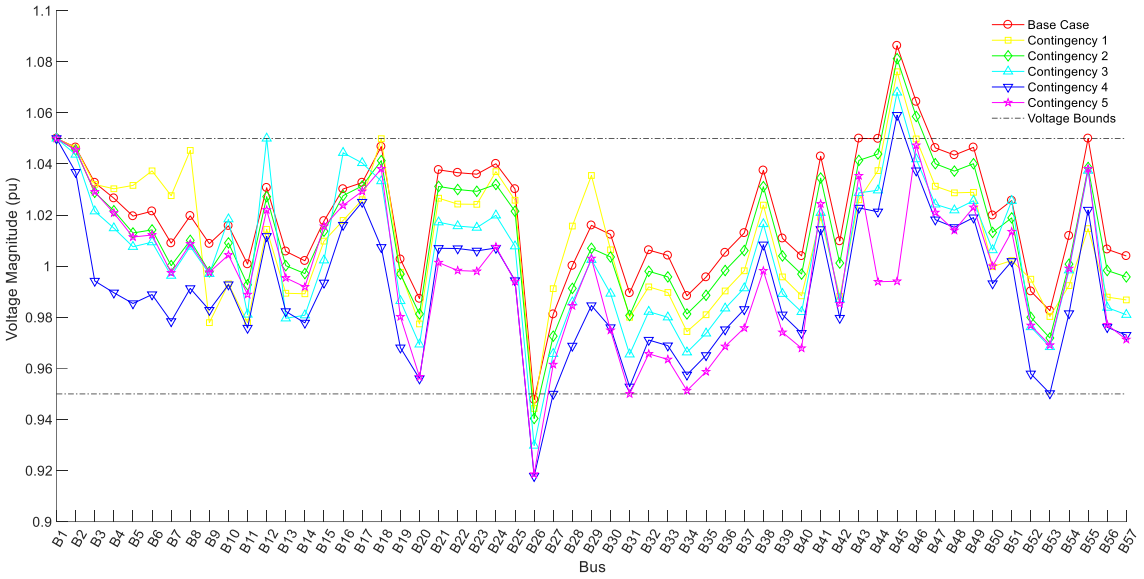

Figure 19 – Voltage profiles across IEEE 57-bus system in different networks.

It can be noticed that voltages at B26, B45 and B46 in some networks are outside the range of 0.95 – 1.05 pu. This is because there is no generator or load at those buses according to Figure 18, so no voltage constraint is defined for the three buses and the abnormal voltages will not cause any equipment damage. Since the SCQOPFM is constructed based on device models and the system constraints are introduced by device models, no voltage bounds can be directly added to a bus unless the devices connected to that bus contain voltage constraints. This problem can be solved by adding the voltage limits to both sides of a line model, but this approach will give many repetitive constraints. The proposed algorithm treats every device model the same way due to object orientation, which means that there are N sets of voltage constraints for the same bus when N lines are connected to it. Hence, to refrain from increasing the size of the example SCQOPFM due to redundant constraints, currently the bus voltage constraints are only defined in generator and load models. Besides B26, B45 and B46, the voltages at other buses are in the acceptable range as shown in both Table 14 and Figure 19.

When the security constrained quadratic OPF problem is solved, the optimal objective function value (minimum total generation cost) is 43698.15 $/hr. It has been reduced iteratively from an initial value of 51364.67 $/hr as shown in Figure 20.

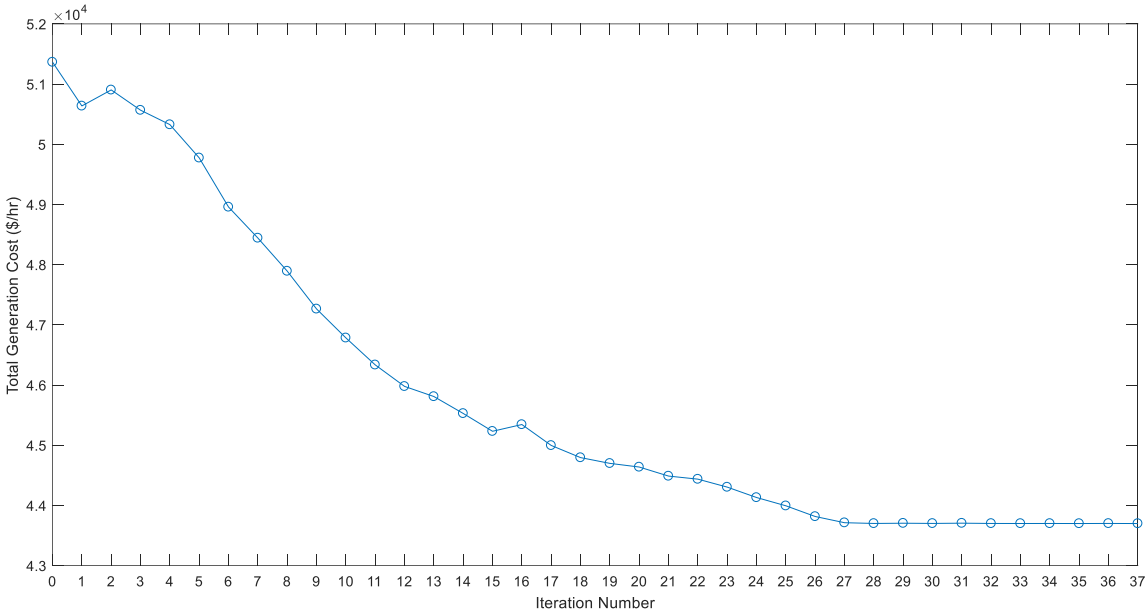

Figure 20 – Variation of total generation cost across iterations.

For comparison, another security constrained quadratic OPF case is run without including any contingency, forming and solving the benchmark SCQOPFM. Hence, the optimization is performed on only 1 network, which is the base case with all devices operational. This time, the SCQOPF problem is solved at iteration 10 in 0.06 seconds. The optimal objective function value is 42631.03 $/hr, which is less than that of the example SCQOPFM with 5 contingencies considered. It makes sense as there are less constraints in the benchmark SCQOPFM with no contingency, so the solution space is larger.

Table 15 gives the optimal controls of the benchmark case, which is quite different from those in Table 13. With no contingency involved, the benchmark problem does not

need to worry about possible device outages, so the controls are able to move more freely. The generator at B8 sees a large increase in its real power generation when Table 15 is compared with Table 13. This is because the outage of line B8 – B9 in contingency 1 of the example SCQOPFM causes a big problem in transferring the power from B8 to other parts of the system. Therefore, the generator at B8 cannot generate as much real power as in the benchmark case even though it helps reducing the total generation cost in the base case of the system.

Table 15 – Solved controls in benchmark SCQOPFM.

Device	Location	Control	Optimal Value
Generator	B1	Voltage Setpoint	1.05 pu
Generator	B2	Real Power Output	50.0 MW
		Reactive Power Output	24.2450 MVar
Generator	B3	Real Power Output	79.4436 MW
		Reactive Power Output	29.0950 MVar
Generator	B6	Real Power Output	48.0713 MW
		Reactive Power Output	0.8710 MVar
Generator	B8	Real Power Output	444.5734 MW
		Reactive Power Output	58.4041 MVar
Generator	B9	Real Power Output	20.1719 MW
		Reactive Power Output	4.0880 MVar
Generator	B12	Real Power Output	410.0 MW
		Reactive Power Output	85.1336 MVar
Transformer	B4 – B18	Tap Setting	0.9827
Transformer	B4 – B18	Tap Setting	0.9824
Transformer	B7 – B29	Tap Setting	0.9712
Transformer	B9– B55	Tap Setting	0.9660
Transformer	B10 – B51	Tap Setting	0.9729
Transformer	B11 – B41	Tap Setting	0.9000

Transformer	B11 – B43	Tap Setting	0.9578
Transformer	B13 – B49	Tap Setting	0.9067
Transformer	B14 – B46	Tap Setting	0.9672
Transformer	B15 – B45	Tap Setting	0.9556
Transformer	B21 – B20	Tap Setting	1.0030
Transformer	B24 – B26	Tap Setting	1.0687
Transformer	B34 – B32	Tap Setting	0.9750
Transformer	B39 – B57	Tap Setting	0.9939
Transformer	B40 – B56	Tap Setting	1.0380

The same example SCQOPF with the 5 contingencies in Table 12 considered is also solved with Knitro [80] for more comparisons. Since the SCQOPFM is sent to the Knitro solver in the form of (62), which is referred to be Knitro SCQOPFM, the number of variables is 2704 while the total number of equality and inequality constraints is 4236. The total runtime including SCQOPF formulation and solution by Knitro is 11.5 minutes. The optimal objective function value is 43697.63 $/hr, which is only 0.52 $/hr less than the value obtained through the SLP approach. Since the proposed SCQOPF algorithm only takes about 1 second to solve the same problem, its overall performance is way better than Knitro.

The base case optimal bus voltages of the example, benchmark and Knitro SCQOPFMs are plotted for comparison in Figure 21. In the benchmark results, only the voltage at B45 is above 1.05 pu because no voltage constraints are imposed on that bus. In general, the benchmark SCQOPFM has a better system voltage profile, which is reasonable. The existence of the 5 contingencies in the example case greatly affects the controls in the IEEE 57-bus system, so the voltages solved from the power flow based on

the controls have already taken the possible device outages into account. Since the Knitro SCQOPFM also includes the 5 contingencies, it gives a voltage profile that closely matches the one generated by the example SCQOPFM. This is reasonable considering the small difference in their objective function values.

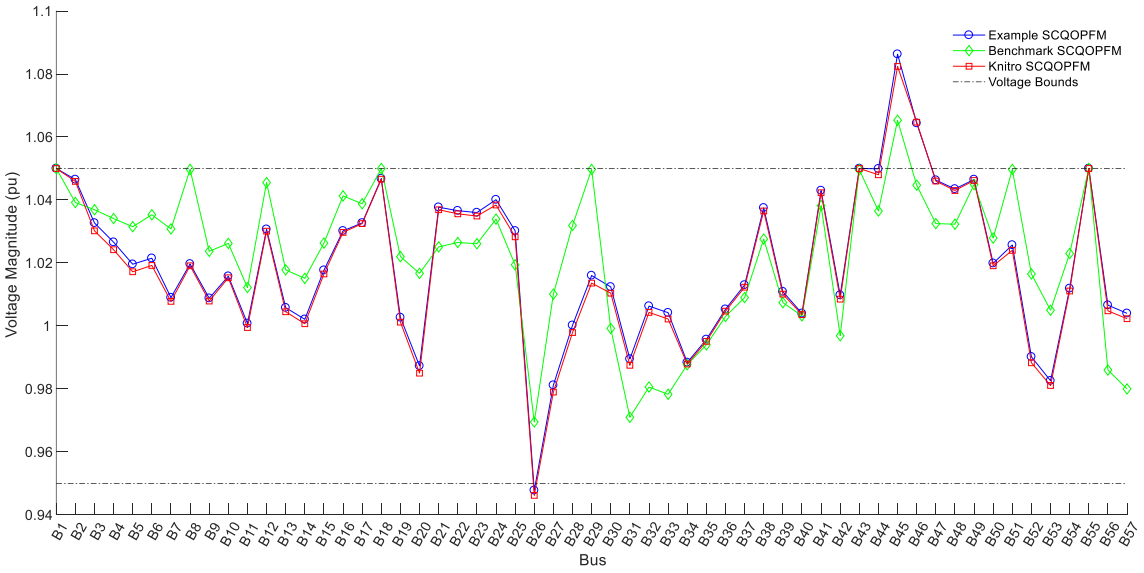

Figure 21 – Base case voltages in three different SCQOPFMs.

9.3 Summary

This chapter demonstrates how the proposed method is used for the SCQOPF application, which is tested on the IEEE 57-bus system with addition of selected contingencies. The whole SCQOPF problem construction and solution process is coded in Visual C++.

The example SCQOPFM of the test system has 6 networks, 1 base case and 5 contingencies, with different device types taken into consideration for possible device outages. The objective function is minimizing the total generation cost in the base case.

The process of forming and solving the SCQOPF problem takes about 1 second. It takes the SLP 38 iterations before the optimal controls shared among different networks are acquired. All the constraints in all the networks are satisfied under the optimal operating condition, which successfully minimizes the total generation cost of the 57-bus system. A benchmark case is introduced by only optimizing the base case without incorporating any contingency. The runtime for constructing and solving the benchmark SCQOPFM is 0.06 seconds. The optimal objective function value is less than the example case as the optimal solution is less restricted, resulting in a different set of control values. The benchmark SCQOPFM also gives a better system voltage profile than the example SCQOPFM that has to accommodate 5 different contingencies.

The numerical case with the same contingencies is also solved using commercial solver Knitro. The outcomes are very close to the results produced by the proposed SCQOPF algorithm. When the program uses Knitro instead of SLP, the final objective function value has a 0.52 $/hr decrease, but the program runtime is 11.5 minutes, which is much worse. The comparison demonstrates an outstanding performance of the developed approach, especially in efficiency.

In general, the proposed method has generated promising results for the SCQOPF application. The performance of the algorithm developed is highly desirable as it outperforms commercial software Knitro.

CHAPTER 10. CONCLUSIONS AND FUTURE WORK DIRECTIONS

This chapter concludes the dissertation and discusses future work directions.

10.1 Conclusions

The following work has been completed for this thesis.

1) A new formulation of OPF problems based on KCL equations and detailed object-oriented device models has been proposed.
2) An algorithm that is able to autonomously formulate and solve multi-stage quadratic flexible OPF and security constrained quadratic OPF problems has been developed.
3) Sequential linear programming algorithms tailored to the two different OPF applications have been developed with the costate method incorporated for linearization.
4) A program with a user-interface has been written in Visual C++ to formulate and solve the OPF problems autonomously.
5) Numerical cases have been used to demonstrate the performance of the proposed method on both OPF applications, giving promising results.

Unlike the conventional OPF formulations that are based on power mismatch equations at each node in a power system, the proposed OPF formulation uses KCL equations instead and individual device models form the basis of the OPF problem. Standard modeling syntax SCAQCF is utilized for object orientation. With all the devices

modeled in the same form, the network model is constructed also in the SCAQCF syntax. Given an objective function by the user, an OPF problem is directly obtained.

Two different types of OPF problems can be formed based on the proposed formulation method depending on the user's selection, including the multi-stage quadratic flexible OPF and the security constrained quadratic OPF, whose objectives are the optimization of system voltage profile and the minimization of total generation cost, respectively. The MQFOPFM is formed by stacking the objective function and the network model over a number of stages defined, while the SCQOPFM construction involves the combination of different networks, including base case and contingencies.

The OPF problems are solved via tailored SLP algorithms, in which the model constraints are added gradually. The costate method is used to linearize the objective function and the model constraints, giving a linearized problem with respect to controls only. Hence, the size of the problems to be solved is as small as possible in each iteration. The solution algorithms are extremely efficient due to the above features. The optimal solutions obtained are the control actions to be implemented to drive the systems to the optimal operating conditions.

The whole process from network formulation to constructing and solving either OPF problem is autonomous, thus reducing possible human errors. The multi-stage quadratic flexible OPF is tested on a PNM real feeder model and the security constrained quadratic OPF is run on the IEEE 57-bus system. In both numerical cases, the algorithm developed gives promising results and outperforms commercial nonlinear optimization software Knitro. Therefore, the performance of the proposed method is highly desirable.

The main contributions of this dissertation are:

1) Detailed physically based device models form the basis of OPF formulation, giving more accurate information of the system.
2) Object-oriented modeling makes it easy and efficient to integrate various DERs into OPF problems.
3) Formulated OPF problems consist of mostly linear KCL equations and the rest are quadratic equations, thus reducing the complexity of the problem.
4) Incorporating the costate method in tailored SLP algorithms with gradually added violated constraints greatly reduces problem size and increases efficiency.
5) The whole OPF formulation and solution process is fully autonomous, reducing the possibility of introducing human errors.

10.2 Future Work Directions

The future work includes the following possible directions.

First, since the OPF problems are formulated based on detailed device models, there are scalability issues, especially when the number of stages for the MQFOPFM or the number of contingencies for the SCQOPFM is too large. Hence, the algorithm efficiency needs to be improved, possibly by using parallel computing techniques.

Second, the test systems are still not big enough to demonstrate the performance of the proposed method on practical systems. Larger real systems need to be simulated and used for testing the OPF algorithms. In addition, more devices such as converters, wind turbines and plug-in electric vehicles are to be modeled and incorporated into the process.

Third, the current SLP solution method for solving the OPF problems still has much space for improvements. For example, the parameter ρ used to adjust the maximum control excursion limits in each iteration is based on experience. A better method to adjust the control movements between consecutive iterations could be incorporated into the SLP algorithm to increase the speed and quality of convergence, like a mechanism that adjusts the movements based on the amount of existing constraint violations.

Fourth, discrete variables such as binary controls are to be included in the systems in the future. The SLP algorithm requires some modifications to accommodate the addition of discrete variables, possibly by introducing some pre-defined rules to set the discrete values in the solution to appropriate integers.

Last but not least, the two OPF applications can be combined into one, giving a security constrained multi-stage quadratic flexible OPF problem. When there are forecasts available, the model predictive control can be used to achieve optimizations on moving horizons with changing OPF models.

REFERENCES

[1] J. Carpentier, "Contribution to the economic dispatch problem", *Bulletin de la Societe Francoise des Electriciens*, vol. 3, no. 8, pp. 431-447, 1962.

[2] W. Zhang, F. Li and L. M. Tolbert, "Review of reactive power planning: objectives, constraints, and algorithms," in *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2177-2186, Nov. 2007.

[3] J. Peschon, D. S. Piercy, W. F. Tinney, O. J. Tveit and M. Cuenod, "Optimum control of reactive power flow," in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-87, no. 1, pp. 40-48, Jan. 1968.

[4] H. W. Dommel and W. F. Tinney, "Optimal power flow solutions," in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-87, no. 10, pp. 1866-1876, Oct. 1968.

[5] O. Alsac and B. Stott, "Optimal load flow with steady-state security," in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-93, no. 3, pp. 745-751, May 1974.

[6] G. C. Contaxis, C. Delkis and G. Korres, "Decoupled optimal load flow using linear or quadratic programming," in *IEEE Transactions on Power Systems*, vol. 1, no. 2, pp. 1-7, May 1986.

[7] S. Iwamoto and Y. Tamura, "A fast load flow method retaining nonlinearity," in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-97, no. 5, pp. 1586-1599, Sept. 1978.

[8] B. S. Babic, "Decoupled load flow with variables in rectangular form," in *IEE Proceedings C - Generation, Transmission and Distribution*, vol. 130, no. 3, pp. 98-102, May 1983.

[9] Q. Y. Jiang, H. -D. Chiang, C. X. Guo and Y. J. Cao, "Power-current hybrid rectangular formulation for interior-point optimal power flow," in *IET Generation, Transmission & Distribution*, vol. 3, no. 8, pp. 748-756, August 2009.

[10] Y. Xu, J. Hu, W. Gu, W. Su and W. Liu, “Real-time distributed control of battery energy storage systems for security constrained DC-OPF,” in *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1580-1589, May 2018.

[11] D. W. Wells, “Method for economic secure loading of a power system,” in *Proceedings of the Institution of Electrical Engineers*, vol. 115, no. 8, pp. 1190-1194, August 1968.

[12] H. Nicholson and M. J. H. Sterling, “Optimum dispatch of active and reactive generation by quadratic programming,” in *IEEE Transactions on Power Apparatus and Systems,* vol. PAS-92, no. 2, pp. 644-654, March 1973.

[13] H. Glavitsch and M. Spoerry, “Quadratic loss formula for reactive dispatch,” in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-102, no. 12, pp. 3850-3858, Dec. 1983.

[14] I. Hano, Y. Tamura, S. Narita and K. Matsumoto, “Real time control of system voltage and reactive power,” in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-88, no. 10, pp. 1544-1559, Oct. 1969.

[15] J. C. Kaltenbach and L. P. Hajdu, “Optimal corrective rescheduling for power system security,” in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-90, no. 2, pp. 843-851, March 1971.

[16] K. Iba, “Reactive power optimization by genetic algorithm,” in *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 685-692, May 1994.

[17] A. Rabiee, A. Soroudi and A. Keane, “Information gap decision theory based OPF with HVDC connected wind farms,” in IEEE Transactions on Power Systems, vol. 30, no. 6, pp. 3396-3406, Nov. 2015.

[18] R. A. Jabr, S. Karaki and J. A. Korbane, “Robust multi-period OPF with storage and renewables,” in IEEE Transactions on Power Systems, vol. 30, no. 5, pp. 2790-2799, Sept. 2015.

[19] B. Stott and O. Alsac, “Fast decoupled load flow,” in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-93, no. 3, pp. 859-869, May 1974.

[20] B. Stott and E. Hobson, “Power system security control calculations using linear programming, Part I,” in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-97, no. 5, pp. 1713-1720, Sept. 1978.

[21] Y. Tao and A. P. S. Meliopoulos, “An sequential linear programming algorithm for security-constrained optimal power flow,” *41st North American Power Symposium*, Starkville, MS, 2009.

[22] A. Castillo, P. Lipka, J. Watson, S. S. Oren and R. P. O'Neill, “A successive linear programming approach to solving the IV-ACOPF,” in IEEE Transactions on Power Systems, vol. 31, no. 4, pp. 2752-2763, July 2016.

[23] J. Nanda, D. P. Kothari and S. C. Srivastava, “New optimal power-dispatch algorithm using Fletcher's quadratic programming method,” in *IEE Proceedings C - Generation, Transmission and Distribution*, vol. 136, no. 3, pp. 153-161, May 1989.

[24] Y. Tao and A. P. S. Meliopoulos, “Optimal power flow via quadratic power flow,” *2011 IEEE/PES Power Systems Conference and Exposition*, Phoenix, AZ, 2011.

[25] T. Liu, B. Sun and D. H. K. Tsang, “Rank-one solutions for SDP relaxation of QCQPs in power systems,” in *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 5-15, Jan. 2019.

[26] T. Gomez, I. J. Perez-Arriaga, J. Lumbreras and V. M. Parra, “A security-constrained decomposition approach to optimal reactive power planning,” in *IEEE Transactions on Power Systems*, vol. 6, no. 3, pp. 1069-1076, Aug. 1991.

[27] X. Bai and H. Wei, “Semi-definite programming-based method for security-constrained unit commitment with operational and optimal power flow constraints,” in *IET Generation, Transmission & Distribution*, vol. 3, no. 2, pp. 182-197, February 2009.

[28] J. Peschon, D. W. Bree and L. P. Hajdu, “Optimal power-flow solutions for power system planning,” in *Proceedings of the IEEE*, vol. 60, no. 1, pp. 64-70, Jan. 1972.

[29] T. C. Giras and S. N. Talukdar, “Quasi-Newton method for optimal power flows”, in *International Journal of Electrical Power & Energy Systems*, vol. 3, no. 2, pp. 59-64, 1981.

[30] W. D. Rosehart, C. A. Canizares and A. Vannelli, “Sequential methods in solving economic power flow problems,” *IEEE 1997 Canadian Conference on Electrical and Computer Engineering,* St. Johns, Nfld., 1997.

[31] K. R. C. Mamandur and R. D. Chenoweth, “Optimal control of reactive power flow for improvements in voltage profiles and for real power loss minimization,” in *IEEE Transactions on Power Apparatus and Systems,* vol. PAS-100, no. 7, pp. 3185-3194, July 1981.

[32] C. N. Lu and M. R. Unum, “Network constrained security control using an interior point algorithm,” in *IEEE Transactions on Power Systems*, vol. 8, no. 3, pp. 1068-1076, Aug. 1993.

[33] J. A. Momoh, S. X. Guo, E. C. Ogbuobiri and R. Adapa, “The quadratic interior point method solving power system optimization problems,” in *IEEE Transactions on Power Systems*, vol. 9, no. 3, pp. 1327-1336, Aug. 1994.

[34] E. D. Castronuovo, J. M. Campagnolo and R. Salgado, “On the application of high performance computation techniques to nonlinear interior point methods,” in *IEEE Transactions on Power Systems*, vol. 16, no. 3, pp. 325-331, Aug. 2001.

[35] J. Lavaei and S. H. Low, “Zero duality gap in optimal power flow problem,” in IEEE Transactions on Power Systems, vol. 27, no. 1, pp. 92-107, Feb. 2012.

[36] C. Coffrin, H. L. Hijazi and P. Van Hentenryck, “The QC relaxation: a theoretical and computational study on optimal power flow,” in *IEEE Transactions on Power Systems,* vol. 31, no. 4, pp. 3008-3018, July 2016.

[37] J. Yuryevich and Kit Po Wong, “Evolutionary programming based optimal power flow algorithm,” in *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1245-1250, Nov. 1999.

[38] H. R. Cai, C. Y. Chung and K. P. Wong, “Application of differential evolution algorithm for transient stability constrained optimal power flow,” in *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 719-728, May 2008.

[39] C. Sumpavakup, I. Srikun and S. Chusanapiputt, “A solution to the optimal power flow using artificial bee colony algorithm,” *2010 International Conference on Power System Technology,* Hangzhou, 2010.

[40] M. R. AlRashidi and M. E. El-Hawary, “Hybrid particle swarm optimization approach for solving the discrete OPF problem considering the valve loading effects,” in *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2030-2038, Nov. 2007.

[41] B. Allaoua and A. Laoufi, “Collective intelligence for optimal power flow solution using ant colony optimization,” in *Leonardo Electronic Journal of Practices and Technologies*, vol. 7, no. 13, pp. 88-105, Jan. 2008.

[42] V. J. Gutierrez-Martinez, C. A. Canizares, C. R. Fuerte-Esquivel, A. Pizano-Martinez and X. Gu, “Neural-network security-boundary constrained optimal power flow," in *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 63-72, Feb. 2011.

[43] Y. T. Hsiao, C. C. Liu, H. D. Chiang and Y. L. Chen, “A new approach for optimal VAr sources planning in large scale electric power systems,” in *IEEE Transactions on Power Systems*, vol. 8, no. 3, pp. 988-996, Aug. 1993.

[44] C. Mishra, S. P. Singh and J. Rokadia, “Optimal power flow in the presence of wind power using modified cuckoo search,” in *IET Generation, Transmission & Distribution*, vol. 9, no. 7, pp. 615-626, 2015.

[45] T. Wang, M. Meskin and I. Grinberg, “Comparison between particle swarm optimization and cuckoo search method for optimization in unbalanced active distribution system,” *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, Oshawa, ON, 2017.

[46] S. Sivasubramani and K. S. Swarup, “Sequential quadratic programming based differential evolution algorithm for optimal power flow problem,” in *IET Generation, Transmission & Distribution*, vol. 5, no. 11, pp. 1149-1154, Nov. 2011.

[47] S. Gill, I. Kockar and G. W. Ault, “Dynamic optimal power flow for active distribution networks,” in *IEEE Transactions on Power Systems*, vol. 29, no. 1, pp. 121-131, Jan. 2014.

[48] C. Zhong, A. P. Sakis Meliopoulos, J. Sun, M. Saeedifard and B. Xie, “Modeling of converter losses with high fidelity in a physically based object-oriented way,” *2018 IEEE Power & Energy Society General Meeting (PESGM)*, Portland, OR, 2018.

[49] G. De Carne, M. Liserre, B. Xie, C. Zhong, S. A. P. Meliopoulos and C. Vournas, “Multiphysics modelling of asynchronously-connected grids,” *2018 Power Systems Computation Conference (PSCC)*, Dublin, 2018.

[50] A. P. Meliopoulos, G. J. Cokkinides, George K. Stefopoulos, "Quadratic integration method", in *Proceedings of the International Power System Transients Conference (IPST)*, pp. 19-23, June, 2005.

[51] C. Zhong, A. P. S. Meliopoulos, B. Xie, J. Xie, K. Liu and H. Shao, "Detailed Multiphysics Modeling of Air-Conditioned House," *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, USA, 2019.

[52] C. Zhong and J. L. Mathieu, "Relation between overheating of distribution transformers and switching frequency of electric loads used for demand response," *2015 North American Power Symposium (NAPS)*, Charlotte, NC, 2015.

[53] K. Ma, C. Yuan, X. Xu, J. Yang and Z. Liu, "Optimising regulation of aggregated thermostatically controlled loads based on multi-swarm PSO," in *IET Generation, Transmission & Distribution*, vol. 12, no. 10, pp. 2340-2346, May 2018.

[54] A. Radaideh, U. Vaidya and V. Ajjarapu, "Sequential set-point control for heterogeneous thermostatically controlled loads through an extended markov chain abstraction," in *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 116-127, Jan. 2019.

[55] S. Meliopoulos, C. Zhong, G. Cokkinides, B. Xie, C. Dalton, P. Myrda and E. Farantatos, "Autonomous multi-stage flexible OPF for active distribution systems with DERs," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, pp. 3455-3464, Jan. 2019.

[56] B. Xie, A. P. S. Meliopoulos, C. Zhong, Y. Liu, L. Sun and J. Xie, "Distributed quasi-dynamic state estimation incorporating distributed energy resources," *2018 North American Power Symposium (NAPS)*, Fargo, ND, 2018.

[57] B. Xie, A. P. S. Meliopoulos, Y. Liu and L. Sun, "Distributed quasi-dynamic state estimation with both GPS-synchronized and non-synchronized data," *2017 North American Power Symposium (NAPS)*, Morgantown, WV, 2017.

[58] Y. Liu, S. Choi, A. P. S. Meliopoulos, R. Fan, L. Sun and Z. Tan, "Dynamic state estimation enabled preditive inverter control," *2016 IEEE Power and Energy Society General Meeting (PESGM)*, Boston, MA, 2016.

[59] L. Sun, A. P. S. Meliopoulos, Y. Liu and B. Xie, "Dynamic state estimation based synchronous generator model calibration using PMU data," *2017 IEEE Power & Energy Society General Meeting (PESGM)*, Chicago, IL, 2017.

[60] J. Xie, A. P. S. Meliopoulos, B. Xie, C. Zhong and K. Liu, "Geoelectric field estimation during geomagnetic disturbances," *2019 IEEE Power & Energy Society General Meeting (PESGM)*, Atlanta, GA, 2019.

[61] A. P. S. Meliopoulos, G. J. Cokkinides, P. Myrda, Y. Liu, R. Fan, L. Sun, R. Huang and Z. Tan, "Dynamic state estimation-based protection: status and promise," in *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 320-330, Feb. 2017.

[62] R. Fan, A. P. S. Meliopoulos, L. Sun, Z. Tan and Y. Liu, "Transformer inter-turn faults detection by dynamic state estimation method," *2016 North American Power Symposium (NAPS)*, Denver, CO, 2016.

[63] O. Vasios, B. Xie and A. P. S. Meliopoulos, "Estimation based protection of three-phase saturable core transformer for cross-country fault detection," *2019 IEEE Power & Energy Society General Meeting (PESGM)*, Atlanta, GA, 2019.

[64] Y. Liu, S. Meliopoulos, N. Tai, L. Sun and B. Xie, "Protection and fault locating method of series compensated lines by wavelet based energy traveling wave," *2017 IEEE Power & Energy Society General Meeting (PESGM)*, Chicago, IL, 2017.

[65] Y. Liu, Y. An and B. Xie, "State estimation based transmission line fault locating with sequence distributed parameter models," *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, Beijing, 2017.

[66] Y. Liu, Z. Tan and J. Xie, "Phasor domain transmission line fault locating with three phase distributed parameter modeling," *2018 IEEE Power & Energy Society General Meeting (PESGM)*, Portland, OR, 2018.

[67] J. Xie, A. P. S. Meliopoulos and B. Xie, "Transmission line fault classification based on dynamic state estimation and support vector machine," *2018 North American Power Symposium (NAPS)*, Fargo, ND, 2018.

[68] B. Xie, A. P. S. Meliopoulos, G. Cokkinides, J. Xie, C. Zhong, Y. Liu and T. Prevost, "Dynamic state estimation based unit protection," *2019 IEEE Power & Energy Society General Meeting (PESGM)*, Atlanta, GA, 2019.

[69] J. Xie, A.P. Meliopoulos, and Y. Liu, "Low broadband transmission line model for geomagnetically induced current analysis," in *2018 IEEE Power & Energy Society General Meeting*, Portland, OR, 2018.

[70] A. P. S. Meliopoulos, J. Xie, and G. Cokkinides, "Power system harmonic analysis under geomagnetic disturbances," *2018 18th International Conference on Harmonics and Quality of Power (ICHQP)*, Ljubljana, 2018.

[71] K. Liu, A. P. S. Meliopoulos, B. Xie, C. Zhong and J. Xie, "Quasi-dynamic domain modeling of line-commutated converters with the analytical approach," *2019 North American Power Symposium (NAPS)*, Wichita, KS, 2019.

[72] S. Meliopoulos, G. Cokkinides, R. Beyah, S. Walters and P. Myrda, "Cyber Security and Operational Reliability," *2015 48th Hawaii International Conference on System Sciences*, Kauai, HI, 2015.

[73] S. Meliopoulos, G. Cokkinides, R. Fan, L. Sun and B. Cui, "Command authentication via faster than real time simulation," *2016 IEEE Power and Energy Society General Meeting (PESGM)*, Boston, MA, 2016.

[74] A. P. S. Meliopoulos, G. Cokkinides, R. Fan and L. Sun, "Data attack detection and command authentication via cyber-physical comodeling," in *IEEE Design & Test*, vol. 34, no. 4, pp. 34-43, Aug. 2017.

[75] C. Zhong, A. P. Sakis Meliopoulos, G. J. Cokkinides and B. Xie, "Object-oriented voltage control for AC-DC hybrid distribution systems," *2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, Charlotte, NC, 2018.

[76] H. Shao, S. Li, C. Zhong and T. Habetler, "Multi-objective design optimization of synchronous reluctance machines based on the analytical model and the evolutionary algorithms," *2019 North American Power Symposium (NAPS)*, Wichita, KS, 2019.

[77] F. Capitanescu and L. Wehenkel, "Optimal power flow computations with a limited number of controls allowed to move," in *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 586-587, Feb. 2010.

[78] A. P. Meliopoulos, G. Cokkinides, B. Xie, C. Zhong and J. Johnson, "Full state feedback control for virtual power plants," *Sandia Technical Report*, SAND2017-10178, Sep. 2017.

[79] C. Zhong, J. Sun, J. Xie, S. Grijalva and A. P. S. Meliopoulos, “Real-time human activity-based energy management system using model predictive control,” *2018 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2018.

[80] R. H. Byrd, J. Nocedal, and R.A. Waltz, “KNITRO: An integrated package for nonlinear optimization,” in G. di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, Springer, pp. 35-59, 2006.

[81] C. Zhong, B. Xie and A. P. S. Meliopoulos, “Distribution network voltage profile optimization via multi-stage flexible optimal power flow,” *2019 North American Power Symposium (NAPS)*, Wichita, KS, 2019.

[82] K. Carr, “IEEE 57-bus system,” *Illinois Center for a Smarter Electric Grid (ICSEG)*, Oct. 2013. [Online]. Available: https://icseg.iti.illinois.edu/ieee-57-bus-system/.

[83] DR Power, “IEEE 57-bus case,” *MATPOWER Test Cases*, Jan. 2018. [Online]. Available: https://egriddata.org/dataset/ieee-57-bus-case.

VITA

Chiyang Zhong was born in Nanning, China in 1991. He received a Bachelor's degree in Electrical Engineering and Automation from Huazhong University of Science and Technology in Wuhan, China in 2013. He also received a Bachelor's degree in Electronic and Electrical Engineering from the University of Birmingham in the UK the same year. He later received a Master's degree in Electrical Engineering and Computer Science from the University of Michigan at Ann Arbor, Michigan in 2015. He joined Georgia Institute of Technology in Fall 2015 to pursue his Ph.D. degree.

Chiyang is a student member of IEEE and a student member of IEEE Power & Energy Society. He has interned at Mott MacDonald in Summer 2019. Currently, he is actively looking for a job in the industry.