**Final Report:**
**Design and Implementation of a Power Distribution System Simulator**

and Tra Ngoc Nguyen, Hoang Thai Duong, Nguyen The Viet, Nguyen Hai Duc
`cs2015_{nguyen.tn, duong.ht, viet.nt, duc.nh}@student.vgu.edu.vn`

*Vietnamese German University, Vietnam*

# Abstract[1]

Due to the massive expansion of renewable energy, many European countries launched a conversion to the next generation power grid. The large-scale use of distributed power generators such as wind and photovoltaic energy in low-voltage networks places new challenging demands on electricity grids: While low-voltage networks in traditional power grids are conventionally unmonitored, the decentralized nature of volatile and renewable energies results in a strong need to control and monitor actors in order to react timely to the variable energy demands:

The electricity grid is an unstable balance in which power generation and power consumption must balance each other at every time. When the mains frequency decreases, consumers slow down energy production more than electricity generators can generate. As the mains frequency increases, there is more power generated than needed. Thus, the mains frequency is the reference value of the grid stability. For longer-term frequency decreases, additional energy needs to be fed into the grid and, conversely, frequency increases require power plants to rapidly reduce their capacity. A loss of stability occurs when the grid is no longer able to return to a stable operating point following the occurrence of a disturbance that results in a large imbalance between power generation and load. This can then cause persistent oscillations of the grid frequency with automatic shut-downs of generating units and hence blackouts or in the worst case it can damage critical infrastructure. In order to keep the grid frequency stable at 50 Hz, it requires an intelligent supply-demand mechanism and, in the case of over-frequency or under-frequency, of a functioning control energy system, which is summarized as 'Demand Side Management' (DSM).

Simulations can be used to analyse and evaluate various control mechanisms

[1]Author: Tra Ngoc Nguyen.

to improve DSM. They allow to generate unorthodox consumer profiles or abnormal scenarios, that are not easy to reproduce in real world environments. The objective of this project is to develop a mains frequency simulator for a small metropolitan power grids. The resulting application should allow the user to simulate and visualize the stability of a power grid with various parameters and show the simulation result as a graph.

# Contents

# Chapter 1

# Introduction[1]

*This chapter aims to provide the objective and structure of this thesis. The following sections summarize our research goals and provide an overview of the organization and structure of this work.*

## 1.1 Motivation

This project is required by "Vietnam-German University" (VGU) in Project Module as a mandatory course in "Computer Science" (CS) major Bachelor Program.

## 1.2 Objective and Requirements

The project will create a simulator to show how the power grid react to different simulated situations through various control mechanisms. The purpose is to replicate as closely as possible to how existing power grids react to similar conditions. This simulator will receive initial information like: number of consumers, their maximum and minimum usage in Walt, number of generators, their maximum generated power, etc... These information will be used to initalize the simulator and the output will be a graph to show how much electricity is produced and how much is used in a particular time in a day. The Objective of this project is to design a simulator do the work above. In the course of this project, we must meet the following requirements:

[1] Author: Tra Ngoc Nguyen.

### 1.2.1 Generator Model Requirements

MUST be registered at the Control Model

MUST provide an interface to measure the momentary power

MUST provide the minimum and maximum supply

MUST provide the maximum supply-change per iteration

MUST provide an interface for the Control Model to request a supply change

MAY change supply due to internal or external conditions (e.g. time, weather, price ..)

MAY turn automatically off after a nr. of iterations (e.g. battery, pump power plant)

### 1.2.2 Consumer Model Requirements

MUST be registered at the Control Model

MUST have an On and Off state, each with defined power

MAY be registered as cluster with other consumers (e.g. households)

MUST provide an interface to measure the power

MUST provide an interface for remote changes

MAY change state due to internal or external conditions (e.g. time, weather, price ..)

### 1.2.3 Control Model Requirements

MUST be able register an arbitary number of generators and consumers

MUST be able to un-register each component

MUST compute the total demand every iteration

MUST compute the total cost every iteration

MUST compute the mains frequency every iteration (50Hz minus the difference of demand-supply, whereas 10

MUST unregister 10

MUST unregister 15

MUST unregister components if mains frequency ¿ 51Hz ¡ 49 Hz for 3 Iterations (defect)

MAY request state changes in consumers for demand side management

MAY request supply and state changes in generators for demand side management

MAY request to start or shutdown generators for demand side management

MAY change the electricity price

### 1.2.4 Graphical User Interface Requirements

MUST be capable to control current the iteration

MUST be capable to show current demand, supply and frequency

MUST be capable to show the number of consumer and generators

MUST be capable to show current weather and electricity price

MUST be capable to show name and power of individual consumer or generator

## 1.3 Outline

This report consist of 2 major part - First, a detail analysis of our software. This include software architecture, use-case analysis, class diagram. The seccond portion will be Code explanation base on the analysis in the previous chapter. As said. The rest of the report will be organized as follow:

**Chapter 2** : Software analysis

**Chapter 3** : Code explanation

**Chapter 4** : Conclusion