

DISTRIBUTED AND DECENTRALIZED CONTROL OF THE POWER  
GRID

BY

ANGEL A. AQUINO-LUGO

DISSERTATION

Submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy in Electrical and Computer Engineering  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Doctoral Committee:

Professor Thomas J. Overbye, Chair  
Professor Peter W. Sauer  
Assistant Professor Alejandro D. Domínguez-García  
Professor David M. Nicol  
Associate Professor Raymond P. Klump

## **ABSTRACT**

The introduction of remotely controlled network devices is transforming the way the power system is operated and studied. The ability to provide real and reactive power support can be achieved at the end-user level. In this dissertation, a framework and algorithm to coordinate this type of end-user control are presented. The algorithm is based on a layered architecture that would follow a chain of command from the top layer (transmission grid) to the bottom layer (distribution grid). At the distribution grid layer, certain local problems can be solved without the intervention of the top layers. A reactive load control optimization algorithm to improve the voltage profile in the distribution grid is presented. The framework integrates agent-based technologies to manage the data and control actions required to operate this type of architecture.

In the distribution network, action can be initiated locally to find solutions to certain problems. That is the reason that in this dissertation decentralized optimization problems are studied to find a solution to control reactive power resources. Four decentralized optimization techniques are studied in two different distribution networks. From the analysis, the Lagrangian relaxation algorithms show the best results to implement a decentralized scheme to control reactive resources. Since capacitors are another reactive power resource to be controlled, the dissertation also presents a decentralized optimization algorithm to minimize losses in the distribution network. The decentralized algorithm results are found to be similar to those using a centralized algorithm.

Finally, because the decentralized optimization algorithm needs to iterate among regions to find a solution, another algorithm is introduced to find a local solution to reactive resource problems in the distribution network. The algorithm is based on sensitivities of voltages to reactive resources to estimate the top of a feeder bus voltage of a particular region inside the distribution network. The algorithm is shown to effectively find a solution to a local problem, and the results are similar to a centralized optimization problem.

The framework and the algorithms presented in this dissertation integrate agent-based technologies to manage the data and control actions required to operate this type of architecture.

To my parents

## **ACKNOWLEDGMENTS**

I would like to express my most sincere gratitude to my adviser, Professor Thomas J. Overbye. His support, guidance and comprehension during the entire Ph.D. process made it possible for me to finish this dissertation.

For their comments and help during the thesis process, I would like to thank the members of my committee: Professors Peter W. Sauer, Alejandro Domínguez-García, David Nicol and Ray Klump.

I would also like to thank the Power and Energy Systems group students and staff for their friendship and support these past four years

I would like to thank, from the bottom of my heart, my beloved friends Ricardo Sepúlveda, Hector Pulgar, Matias Negrete, Robert Lambert and Silvia Güajardo because their friendship and support are invaluable to me. Also I would like to thank my friends in Puerto Rico for their support during those difficult moments in the last few years. Without any of you guys I would not have made it to the end. You have no idea how much I owe you.

Finally, but not least, I would like to thank my parents and family for their love, support and understanding. They were always my biggest fans and always cheered me up when things were difficult for me. Thank you all.

## TABLE OF CONTENTS

1 INTRODUCTION .....	1
1.1 Motivation.....	1
1.2 Power Grid Operation and Control .....	2
1.3 Agent Applications .....	5
1.4 Thesis Overview .....	8
1.5 References.....	8
2 DISTRIBUTED CONTROL ALGORITHMS .....	11
2.1 Agents Interaction.....	11
2.1.1 Agent Management.....	12
2.1.2 FIPA-ACL Message Structure Specification.....	14
2.1.3 FIPA-ACL Communicative Act Library Specification .....	15
2.1.4 FIPA Contract Net Interaction Protocol Specification .....	17
2.2 Distributed Agents and Load-Control OPF .....	19
2.2.1 Load Control OPF Formulation.....	22
2.2.2 Agent Simulation in JADE and OPF Algorithm.....	31
2.2.3 Case Study for the OPF Algorithm.....	32
2.2.4 Second Case Study for the OPF Algorithm .....	34
2.3 Incident Command System .....	37
2.4 ICS Control Algorithm and Architecture.....	41
2.4.1 The Central Control Scheme.....	42
2.4.2 The Local Control Scheme .....	44
2.5 References.....	45
3 DISTRIBUTED CONTROL ALGORITHMS FOR REACTIVE RESOURCES .....	47
3.1 Distribution Power Flow.....	47
3.1.1 Distribution Feeder Line Models.....	48
3.1.2 Distribution Ladder Iterative Technique.....	52
3.2 The Voltage Problem Formulation .....	54
3.2.1 Newton's Method to Solve Optimization Problems .....	57
3.2.2 Ten and Thirty-Four-Bus Reactive Load Control Examples .....	59
3.3 Agent Simulations and Test-Bed Implementations .....	64
3.3.1 Agent Simulation Case Study .....	69
3.4 References.....	71
4 DECENTRALIZED CONTROL ALGORITHMS FOR REACTIVE RESOURCES .....	73
4.1 Local Reactive Problem in a Distribution Feeder.....	73
4.2 Decentralized Optimization in a Distribution Feeder .....	78
4.3 Auxiliary Problem Principle Algorithm.....	82
4.3.1 Auxiliary Problem Principle with Distribution Network Equations.....	86
4.4 Predictor-Corrector Proximal Multiplier Method.....	90

4.4.1 PCPM with Distribution Network Equations .....	94
4.5 Lagrangian Relaxation Decomposition Algorithm .....	95
4.5.1 Lagrangian Relaxation with Distribution Network Equations .....	98
4.6 Lagrangian Relaxation Based Decomposition Algorithm .....	103
4.6.1 Lagrangian Relaxation Based Decomposition Algorithm with Distribution Network Equations .....	106
4.7 Thirty-Four and Sixty-Nine Distribution Feeder Simulations .....	109
4.7.1 Thirty-Four Distribution Feeder Simulation and Results .....	109
4.7.2 Sixty-Nine Distribution Feeder Simulation and Results .....	114
4.8 Power Losses Minimization Problem .....	119
4.8.1 Thirty-Four Distribution Feeder Simulation and Results .....	122
4.8.2 Sixty-Nine Distribution Feeder Simulation and Results .....	125
4.9 Distribution Feeder Load Control Using Sensitivities .....	128
4.9.1 Distribution Feeder Sensitivity Analysis .....	129
4.9.2 Distribution Feeder Optimization with Sensitivities .....	133
4.9.3 Simulations and Results on Three Lateral Feeders on the Sixty-Nine Distribution Feeder .....	135
4.9.4 Distribution Feeder Optimization with Sensitivities .....	140
4.10 References .....	142
5 CONCLUSIONS .....	145
APPENDIX A SENSITIVITY CALCULATIONS .....	148
APPENDIX B TEN, THIRTY-FOUR AND SIXTY-NINE DISTRIBUTION FEEDERS DATA .....	155
AUTHOR'S BIOGRAPHY .....	165

# **1 INTRODUCTION**

## **1.1 Motivation**

Today, much conversation is being made about how the electric power grid will look in the future. The consensus is that it will incorporate new technologies that will let us control the grid in a “smart” way. The problem is that there are many different ideas about what “smart” grid means. A “smart” grid can be defined as the utilization of new digital and intelligent devices to replace the old analog devices in the power network. In this work, “smart” grid relates to using those new intelligent devices to allow for remote control, providing a new opportunity for decentralized control.

Many proponents of the smart grid think that controlling end-user devices, such as loads, will help and aid the power grid during stress and abnormal situations. This will be possible because of the improvements in monitoring and remotely controlled devices that are currently happening in the power grid. For example, the Grid Friendly Appliance controller developed at Pacific Northwest National Laboratory (PNNL) [1] will sense grid conditions by monitoring the frequency of the system and provide automatic load demand response in times of disruption to improve the frequency of the grid. This controller will be installed in certain appliances to turn them off or reduce the loading for a few minutes or even a few seconds, to allow the grid to stabilize. Projects like this will transform the way the power grid is operated and analyzed.

The challenges, whatever the definition, are enormous. The stimulus law of 2009 provides billions of dollars for smart grid funded projects and studies. Certainly the transformation of the grid will change the way it is operated and



analyzed. In this work, some new ideas on how to control the power grid in a decentralized but intelligent scheme are studied. Some examples are presented, as well as the challenges they bring to the electric power grid.

## **1.2 Power Grid Operation and Control**

Currently the grid is operated in a centralized manner. For example, the system protection against faults utilizes relays that are constantly monitoring the grid to detect abnormal conditions, and they initiate corrective action when needed. This protection implements local controls that are part of the Supervisory Control and Data Acquisition (SCADA) supervisory scheme, which is a centralized framework. Every time the power network fails, the central control center determines through the SCADA system which system elements and control actions should be implemented to either save the system from collapse or to reconfigure the system after an outage. The typical control actions include opening and closing breakers and switches, load shedding, connecting devices such as capacitors and reactors, among others.

With the proposed investment in smart grid technologies, new control schemes and frameworks are needed to take full advantage of the technologies currently being deployed in the power grid. For this reason the work extends the ideas presented in [2] and [3] for using real and reactive load as a resource to mitigate certain problems in the power grid. It would integrate the centralized structure of protective relays into the proposed control framework. In [2], a scheme that uses intelligent agents is implemented to relieve line overloads by

controlling certain loads in the grid. Also, a decentralized optimization algorithm was presented to minimize power losses in the distribution network. In [3], a scheme to control reactive power to maintain a healthy voltage profile is presented. The algorithm would be implemented using an intelligent control scheme following a chain-of-command structure called Incident Command System (ICS). The ICS is a “systematic tool used for the command, control, and coordination of an emergency response” [4]. It has a layer architecture that follows a chain of commands from top layers to bottom layers to help solve problems during emergency situations. The work presented in this thesis combines both intelligent frameworks into a more effective scheme that will allow control at the different levels of the power grid.

Distribution automation (DA) was the name used before the smart grid started to be used to indicate all of the improvements in the power grid. The DA concept is a generic term for the automation of the entire distribution system operation and covers the complete range of functions from protection to the Supervisory Control and Data Acquisition system (SCADA) and associated technology applications [5]. In other words, it is the ability to mix local automation, remote control of switching devices and central decision making into an effective operating architecture for the power distribution systems. DA has three main control functions. First, DA controls local automation in which the switch operation would be performed by the protection system or by a logic-based decision making operation. The second function is based on the SCADA control in which the switches can be operated by remote control while monitoring statuses, alarms, and

data measurements. The last function is the centralized automation in which the automatic switch operation is performed by the control center for centralized decision making for cases such as fault isolation, network reconfiguration and service restoration. Figure 1.1 shows a typical architecture for DA, which illustrates that DA starts from the loads that are connected to feeders with automation. Then these feeders get to the substation automation and the SCADA central control center. The FAGW (feeder automation gateway) manages the communication to multiple intelligent switches and acts as a data concentrator. In this work the described tasks that are performed in the DA will be the responsibility of agents connected to the substations and relays effectively localizing the system response in case of a failure or system disturbance. In the rest of the thesis, the term smart grid is used for this DA concept.

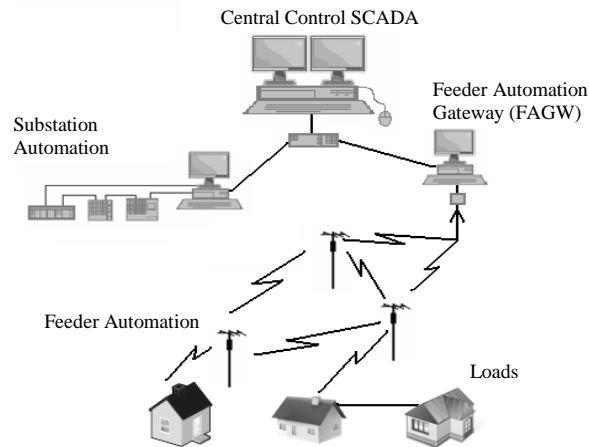


Figure 1.1: Distribution automation components [5]

The work presented in [2] showed that the decisions made in the distribution network can directly affect the transmission grid and vice versa. That is the reason

a good coordination among the power networks has to be implemented. For example, when a transmission line is out of service in a power system, it can create line-flow overloads in other lines that are in service. Line overloads in a transmission system may prevent power transfer. To solve this problem, the use of distributed agents can be implemented to coordinate a solution to relieve the line overloads performing a load optimal power flow (OPF). This work and results will be explained in detail in Chapter 2.

The DA, or smart grid, idea will require a lot of effort in developing new algorithms suitable for the new emerging control applications. This thesis will address and analyze some of those control problems.

### **1.3 Agent Applications**

One of the first applications of the agent concepts was the self-healing of power distribution networks in combat ships. During battles, the ships can suffer severe damage to the electrical system, and in a combat situation it is important to maintain the availability of energy to the loads to keep the ship operational [6-7]. People quickly realized that this concept could be applied to power system distribution networks. In [8], the authors present a multi-agent system (MAS) approach for a decentralized solution for the power system reconfiguration problem using Matlab Simulink S-functions as agents. Following the same approach as in [8], a restoration algorithm applying an expert system type of solution was presented using Matlab Simulink and the Stateflow toolbox was presented in [9]. In [10] an intelligent power routers (IPR) scheme was proposed

where control can be detached from the central control sites, and delegated to IPRs that would be distributed over the entire electric network to initialize and coordinate control actions.

These projects are just a few examples of agent type technologies for control applications. Another important work is the one conducted by the EPRI's IntelliGrid Consortium. Their work would try to implement the integration of data communications networks and energy equipment. The project incorporates Fast Simulation and Modeling (FSM), which is a high performance information technology (IT) infrastructure that combines software, hardware (computing, measurement and control), and communications [11-12].

In recent years, more attention has been given to multi-agent systems (MAS) applications as an alternative to implement decentralized control algorithms in real life. For this reason, the IEEE Power Engineering Society's MAS Working Group presented a two-paper series [13-14] about the MAS technologies applied to the power systems. The main conclusion was that with more experience and research in the matter, a better understanding of the different standards, methodologies, and agent models needed could be achieved. With that in mind, the work presented in this thesis addresses the challenges of studying these MAS technologies, and their possible application in the smart grid.

In the work presented in [15], a decentralized approach to mitigate cascading failures in the power grid is presented. The method implements reciprocal altruistic agents. These agents would consider the goals of neighbor or surrounding agents when achieving the solution of their own individual goals.

This concept is very important because, in the decentralized algorithm presented in this dissertation, the optimization algorithms require cooperation and data exchange from different regions to obtain a solution for any of the individual regions in which the system was separated. Using the concept of reciprocal altruistic agents, the agents would exchange data with the neighbor agents, which would enable the neighbor agents to get a solution to their own personal goal. In [15], the agents exchange data with a certain number of surrounding agents, and then each agent will perform a global optimization algorithm based on the information obtained from their neighbor agents. If data is missing from the neighbor agents, the local agent sets the border information to a predetermined value, for example, the border bus voltage to 1 p.u. This solution would work as long as the information is exchanged without any problem, but if the data is lost the agents would not be able to perform an optimization algorithm with the correct data. That is the reason that a perfect reciprocal altruistic algorithm cannot be used. In the work presented in Chapters 2 and 3, the agents would only consider the directly connected agents' information and data to solve their own personal goals. Also, the optimization algorithms presented in Chapter 4 are suitable for this type of implementation and can be tested using agent type technologies.

Before finishing this section it is important to present one of the most common simulation environments to create agent type simulations. The platform that will be used in the implementation of the decentralized and distributed algorithms presented in Chapters 2 and 3 is JADE, a JAVA framework for

developing FIPA (foundation for intelligent physical agents) compliant agent applications. JADE is one of the most widespread agent-oriented and completely distributed middleware systems to create agents. The framework provides a flexible infrastructure that allows easy extension with add-on modules and is one of the platforms proposed by the IEEE Power Engineering Society's MAS Working Group [13-14].

## **1.4 Thesis Overview**

In Chapter 2 an introduction to distributed control algorithms using agent technologies and an introduction of the incident command system are presented. The details of the proposed control algorithm are presented in Section 2.3. In Chapter 3 an algorithm to control reactive resources using the ICS framework is presented. In Chapter 4 an analysis of the decentralized optimization algorithm is presented. In Section 4.9 an algorithm suitable to follow the ICS framework is presented to control reactive resources locally in the distribution network. Finally the conclusions and future work are presented in Chapter 5.

## **1.5 References**

- [1] Pacific Northwest National Laboratory, Department of Energy GridWise Program, "Grid Friendly Appliance Controller," January 8, 2008, [Online]. Available: [http://gridwise.pnl.gov/technologies/transactive\\_controls.stm](http://gridwise.pnl.gov/technologies/transactive_controls.stm)
- [2] A. Aquino-Lugo and T. J. Overbye, "Agent Technologies for Control Application in the Power Grid," in *43<sup>rd</sup> Hawaii International Conference on System Sciences*, Jan. 2009, pp. 1-10.
- [3] K. M. Rogers, R. Klump, H. Khurana, A. Aquino-Lugo, and T. J. Overbye, "An Authenticated Control Framework for Distributed Voltage Support on

the Smart Grid,” *IEEE Transactions on Smart Grid*, vol. 1, no.1, pp. 40-47, June 2010.

- [4] U.S. Department of Transportation, Federal Highway Administration, Office of Operations, “Simplified Guide to the Incident Command System for Transportation Professionals,” August 2010, [Online]. Available: [http://ops.fhwa.dot.gov/publications/ics\\_guide/index.htm#ics2](http://ops.fhwa.dot.gov/publications/ics_guide/index.htm#ics2)
- [5] J. Northcote-Green and R. Wilson, *Control and Automation of Electrical Power Distribution Systems*, Boca Raton, FL: CRC Press, 2007.
- [6] S. Curcic, C. S. Ozveren, L. Crowe, and P. K. L. Lo, “Electric Power Distribution Network Restoration: A Survey of Papers and a Review of the Restoration Problem,” *Electric Power Systems Research*, vol. 35, no. 2, 73-86, 1995.
- [7] K. L. Butler, N. D. R. Sarma, and V.R. Prasad, “A New Method of Network Reconfiguration for Service Restoration in Shipboard Power Systems,” in *IEEE Transmission and Distribution Conference*, vol. 2, April 1999, pp. 658 - 662.
- [8] J. M Solanki, N. Schultz, and W. Gao, “Reconfiguration for Restoration of Power Systems using Multi-Agent System,” in *37<sup>th</sup> Annual North American Power Symposium*, Oct. 2005, pp. 390-395.
- [9] A. A. Aquino-Lugo and T. J. Overbye, “Distributed Intelligent Agents for Service Restoration and Control Applications,” in *40<sup>th</sup> Annual North American Power Symposium*, Sep. 2008, pp.1-7.
- [10] A. Irizarry-Rivera, M. Rodriguez-Martinez, B. Velez, M. Velez-Reyes, A. R. Ramirez-Orquin, E. O'Neill-Carrillo, and J. R. Cedeño, “Intelligent Power Routers: A Distributed Coordinated Approach for Electric Energy Processing Networks,” *International Journal of Critical Infrastructures*, vol. 3, pp. 20-57, Dec. 2006.
- [11] “Transmission Fast Simulation and Modeling(T-FSM)-Functional Requirements Document,” EPRI, Palo Alto, CA, Tech. Rep. 1011666, March 2005.
- [12] “Transmission Fast Simulation and Modeling(T-FSM)-Functional Requirements Document,” EPRI, Palo Alto, CA, Tech. Rep. 1011667, March 2005
- [13] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, “Multi-Agent Systems for Power Engineering Applications-Part I: Concepts, Approaches, and Technical



Challenges,” *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1743-1752, Nov. 2007.

- [14] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, “Multi-Agent Systems for Power Engineering Applications-Part II: Technologies, Standards, and Tools for Building Multi-agent Systems,” *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1753 - 1759, Nov 2007.
- [15] P. Hines and S. Talukdar, “Reciprocally Altruistic Agents for the Mitigation of Cascading Failures in Electrical Power Networks,” in *Proceeding of the International Conference of Infrastructure Systems*, Rotterdam, 2008, pp. 340-356.

## **2 DISTRIBUTED CONTROL ALGORITHMS**

The work done in decentralized algorithms has focused mostly on parallelizing the solution of the OPF. The concept can be easily applied to perform real time control of power grid devices using distributed and decentralized algorithms. The idea is to implement these algorithms with the help of agents that are distributed in the grid to obtain solutions without much human intervention. The work presented here is going to be expanded in Chapter 4 to incorporate the algorithms in the distribution network for decentralized optimization.

### **2.1 Agents Interaction**

Agents will play an important role in the implementation of the algorithms presented in this thesis. Before continuing, it is important to present certain aspects and details that will help in understanding how exactly the agent's simulation is done.

It was mentioned in Chapter 1 that the agents will be communicating by the implementation of the foundation for intelligent physical agents (FIPA) Contract Net Interaction protocol [1]. This particular protocol is one of many FIPA agents communication languages (ACL) protocols available for agents. The FIPA-ACL is based in speech theory in which messages represent communication acts similar to the way humans communicate. The FIPA ACL used many similar human interactions acts such as inform, request, agree, not understand and refuse. By

implementing and coordinating with these actions, the agents can coordinate and react to situations as a user programmed them to do.

### 2.1.1 Agent Management

In order to understand how the agents interact, it is important to explain how the agents are managed. The FIPA specifications state the need of a logical model reference for the creation, registration, location, communication, migration and operation of agents. Figure 2.1 presents the reference model for the agent's management.

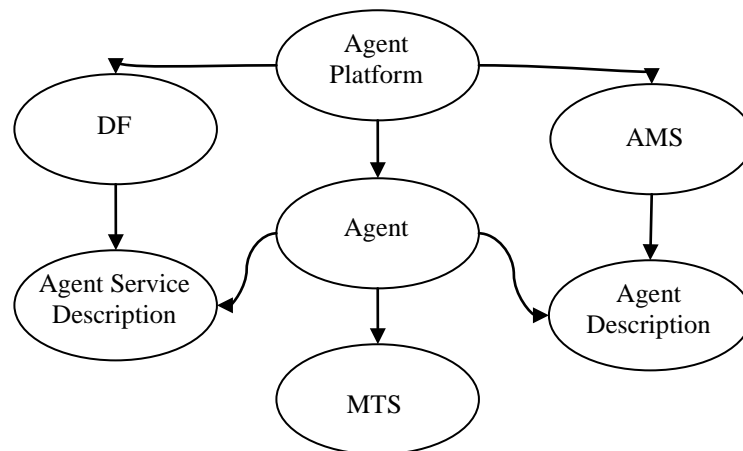


Figure 2.1: Agent management reference model ontology [1]

The details of the agent management reference model are explained here:

**Agent Platform:** It provides the physical infrastructure in which an agent “lives.”

The AP consists of the machines, operating systems, FIPA and additional software needed for the agents to run. It is the job of the developer to design the specific details of the AP, and this is not a subject for the FIPA standard beyond the components explained below.

**Agent:** An agent is a computational software that “lives” in an AP and act on behalf of a user. Typically an agent offers one or more computational services that can be published in a service description service. Then other agents can look for those services in a director facilitator to which agents are subscribed.

**Director Facilitator:** The DF provides a yellow page service to other agents. It keeps detailed information and a complete list of agents and the services they provide. An agent that wishes to publish its services needs to register to an appropriate DF. Typically an AP has its own DF and usually this is enough, but there can be another DF to which an agent can register. Other agents can look for a specific DF to search for an agent that provides a specific required service.

**Agent Management System:** The AMS is a mandatory component of an AP and is responsible for managing the operation of an AP, such as creation and deletion of agents [1]. The agents need to register to an AMS to get an agent identifier (AID) and keep a record of the agents living in a particular AP.

**Message Transport Service (MTS):** The MTS is a service that is provided by the AP to transport ACL messages between agents of an AP and between agents of different APs. The messages must provide a set of parameters such as to whom the message is sent in order to exchange messages.

Now that the model reference and the tools needed to create FIPA complaints agents are explained, some details about the FIPA-ACL message structure are going to be presented.

### 2.1.2 FIPA-ACL Message Structure Specification

The FIPA-ACL messages consist in a set of one or more parameters that will provide an effective communication between agents. The parameters needed will depend on the situation, but the performative parameter is mandatory for an ACL message. The performative parameter provides the type of communicative act for the message. Without this parameter the agents will not know how to interpret the messages. Other important parameters are the sender, receiver and the content of the message. A summary of the ACL message parameters is presented in Table 2.1 [1]. These parameters are used to provide an effective communication to the desired agents.

Table 2.1: ACL message parameters used for communication

Parameter	Description
Performative	Type of the communicative act of the message
sender	Identity of the sender of the message
receiver	Identity of the intended recipients of the message
reply-to	Which agent to direct subsequent messages to within a conversation thread
content	Content of the message
language	Language in which the content parameter is expressed
encoding	Specific encoding of the message content
ontology	Reference to an ontology to give meaning to symbols in the message content
protocol	Interaction protocol used to structure a conversation
conversation-id	Unique identity of a conversation thread
reply-with	An expression to be used by a responding agent to identify the message
in-reply-to	Reference to an earlier action to which the message is a reply
reply-by	A time/date indicating by when a reply should be received

### 2.1.3 FIPA-ACL Communicative Act Library Specification

The FIPA-ACL is based on speech act theory [1], which defines the functions of simply specified actions. Thus the FIPA-ACL defines communication in terms of communicative acts (CA) performed by the act of communicating [1]. These functions are defined in the FIPA CA Library specifications that include all the communicative acts that it allows for communication. Using these FIPA CA, the agents can carry complex communications similar to human interactions and also provide an effective way to exchange messages among agents. At the same time, they provide ways (depending on the situation) to terminate communications when an agreement is or is not reached. The FIPA CA Library can be used to create complex interaction protocols that will be different depending on the actions and situations in which the agents are going to be implemented. A summary of the FIPA CA is presented in Table 2.2 [1].

Table 2.2: FIPA CA descriptions

<b>FIPA Communicative Act</b>	<b>Description</b>
Accept Proposal	The action of accepting a previously submitted proposal to perform an action
Agree	The action of agreeing to perform some action, possibly in the future
Cancel	The action of one agent informing another agent that the first agent no longer has the intention that the second agent performs some action
Call for Proposal	The action of calling for proposals to perform a given action

Table 2.2: Continued

Confirm	The sender informs the receiver that a given proposition is true, where the receiver is known to be uncertain about the proposition
Disconfirm	The sender informs the receiver that a given proposition is false, where the receiver is known to believe, or believe it likely, that the proposition is true
Failure	The action of telling another agent that an action was attempted but the attempt failed
Inform	The sender informs the receiver that a given proposition is true
Inform If	A macro action for the agent of the action to inform the recipient whether or not a proposition is true
Inform Ref	A macro action allowing the sender to inform the receiver of some object believed by the sender to correspond to a specific descriptor, for example a name
Not Understood	The sender of the act (for example, <i>i</i> ) informs the receiver (for example, <i>j</i> ) that it perceived that <i>j</i> performed some action, but that <i>i</i> did not understand what <i>j</i> just did. A particular common case is that <i>i</i> tells <i>j</i> that <i>i</i> did not understand the message that <i>j</i> just sent to <i>i</i> .
Prerogative	The sender intends that the receiver treat the embedded message as sent directly to the receiver, and wants the receiver to identify the agents denoted by the given descriptor and send the received <i>propagate</i> message to them
Propose	The action of submitting a proposal to perform a certain action, given certain preconditions
Proxy	The sender wants the receiver to select target agents denoted by a given description and to send an embedded message to them
Query If	The action of asking another agent whether or not a given proposition is true
Query Ref	The action of asking another agent for the object referred to by a referential expression
Refuse	The action of refusing to perform a given action, and explaining the reason for the refusal

Table 2.2: Continued

Reject Proposal	The action of rejecting a proposal to perform some action during a negotiation
Request	The sender requests the receiver to perform some action. One important class of uses of the request act is to request the receiver to perform another communicative act
Request When	The sender wants the receiver to perform some action when some given proposition becomes true
Request Whenever	The sender wants the receiver to perform some action as soon as some proposition becomes true and thereafter each time the proposition becomes true again
Subscribe	The act of requesting a persistent intention to notify the sender of the value of a reference, and to notify again whenever the object identified by the reference changes

#### **2.1.4 FIPA Contract Net Interaction Protocol Specification**

The FIPA Contract Net Interaction Protocol (IP) describes the interaction between one agent (the Initiator) that wishes to have a task performed by one or more agents (the Participants) [1]. Typically any number of participants may respond to the specified task and the rest will be refused. This particular protocol was chosen because of the nature of the ICS control algorithm implemented in this research. The ICS scheme is based on a hierarchical structure in which a certain agent is responsible for other low level agents. Thus the leader agent of the particular realm must initiate or call for proposals to the low level agents in a lower realm. Then the leader agent will decide which agent will participate and which not, depending on the problem to be solved. For example in the case of controlling reactive loads, there are going to be certain agents that, at a particular moment in time, cannot control reactive load. Thus, after calling for proposals, the distribution agent at the top of the feeder will refuse any help from that particular



agent with no reactive load and will only accept the help from agents that have reactive load available to control in the rest of the distribution network. In this manner the amount of messages exchanged for communication will be minimized and only the necessary messages to particular agents are sent.

This protocol implements the following communication algorithm. This algorithm was modified to fit the type of control algorithm studied in this research work.

Step1) An initiator agent requests a task to be performed by other agents. Then the initiator calls for proposals (CFP) to the participant agents.

Step 2) The participant agents received the CFP and can either tell the initiator that they can perform the proposal or refuse it.

Step 3) If the participant agent indicates that the proposal can be satisfied, it sends that message to the initiator.

Step 4) The initiator confirms the participation proposal.

Step 5) If the participant agent receives the confirmation of the proposal, then the participant agent performs the task.

Step 6) After the task is performed, the participant agent informs the initiator that the task was performed.

Step 7) The initiator confirms that it has received the message that the task was performed. After this, no further requests to perform that task are sent to any other agent.

Every time the agents communicate, this type of interaction will be implemented. To illustrate the concept better, the next sections will illustrate how the agents and the proposed framework interaction will work.

## **2.2 Distributed Agents and Load-Control OPF**

In Section 1.3, various algorithms that use agents to find solutions to certain problems were presented. Also it was assumed that communication was to be implemented in order to exchange information between different agents. In this section two simple examples are presented to show the necessary requirements to implement agents technology for control applications in the power systems. The analysis shows the requirements necessary to implement that kind of solution and which simulation environments are suitable for this application.

The first presented case is when a transmission line is out of service in a power system, it can create line-flow overloads in other lines that are in service. Line overloads in a transmission system may prevent power transfer. To solve this problem, the use of distributed agents can be implemented to coordinate a solution to relieve the line overloads. In order to perform this task the power system is divided to regions as is shown in Figure 2.2.

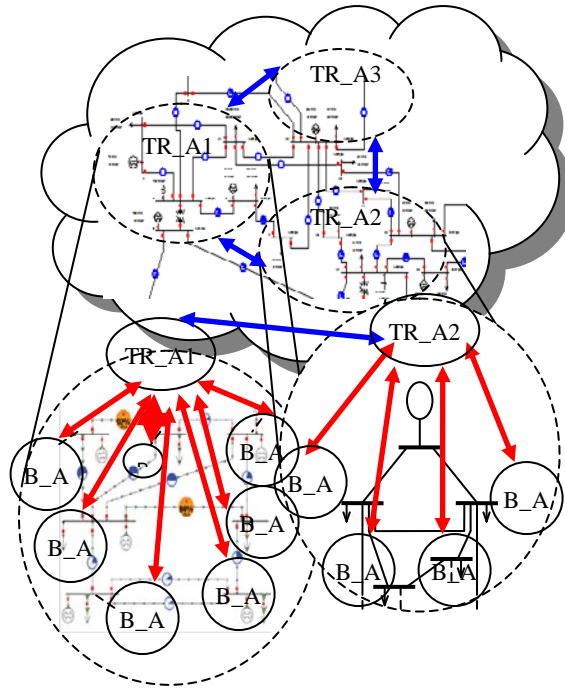


Figure 2.2: Power system divided in regions

Each region (TR\_A) is divided into transmission and distribution regions as shown in Figure 2.3. The transmission region (TR\_A) is responsible for network devices that can be controlled. In this case, it was assumed that some of the loads can be controlled. The agents at each load bus (B\_A) would know at a specific time the amount of load that is connected to that bus, as well as the load amount that can be controlled. These bus agents (B\_A) communicate with many of the distribution network devices and obtain and exchange data among the distribution network agents. One of these distribution network devices is the smart meter that is part of the advance meter infrastructure (AMI). Using data obtained from the AMI smart meter, the bus agent knows the amount of load that can be controlled.

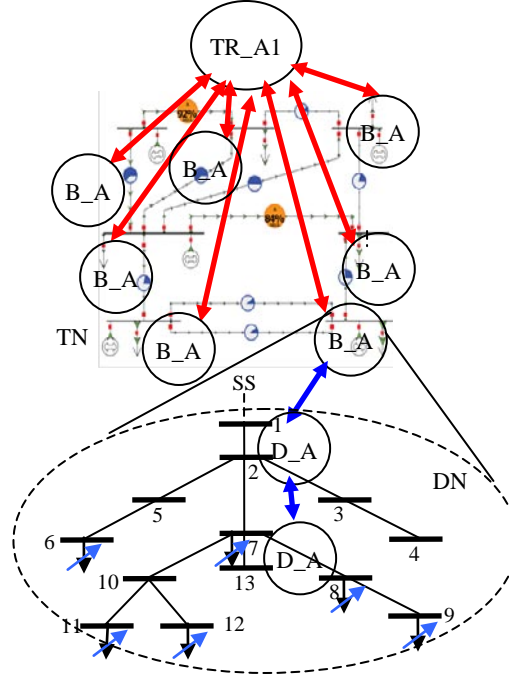


Figure 2.3: Transmission region and distribution region (TN is transmission network and DN is distribution network)

The AMI data would be collected by distribution agents (D\_A) that exchange information among themselves and the B\_A. Two types of load control can be performed. One is the pluggable hybrid connected to the grid to inject power and the other is the disconnection of loads for shedding purposes, but only for specific situations such as the line-overload case presented in Section 2.2.1. Based on the information collected at each bus agent, the region agent performs a local load OPF to determine the amount of load that needs to be connected (in case of pluggable hybrids) or disconnected (for load shedding) to relieve the overloaded lines. Also the regional agent negotiates with other regional agents in the transmission grid if a solution is not obtained. For this last case a decentralized OPF looks for an optimal solution. This decentralized OPF is going to be part of the work presented in Chapter 4.

### 2.2.1 Load Control OPF Formulation

To show how the proposed control algorithm would be implemented in real life, a small case example is presented. A load control OPF was implemented. The optimization problem can be defined as follows:

$$\begin{aligned} \min \quad & \sum_{n=1}^N L^n \\ \text{s.t.} \quad & PF \text{ constraints} \end{aligned} \quad (2.1)$$

$$\begin{aligned} Pflow_{ij_{\min}} &\leq Pflow_{ij} \leq Pflow_{ij_{\max}} \\ L_{\min}^n &\leq L^n \leq L_{\max}^n \end{aligned} \quad ,$$

where the PF constraints are the equality constraints of the OPF and are the power balance equations, which are obtained by imposing the conservation of active and reactive power to each bus of a power system network. First we define the active and reactive power injection at bus  $k$  as

$$P_k(V, \delta) = \sum_{j \in N} V_k V_j [G_{kj} \cos(\delta_k - \delta_j) + B_{kj} \sin(\delta_k - \delta_j)], \quad k \in N, k \neq 1 \quad (2.2)$$

$$Q_k(V, \delta) = \sum_{j \in N} V_k V_j [G_{kj} \sin(\delta_k - \delta_j) - B_{kj} \cos(\delta_k - \delta_j)], \quad k \in NPQ \quad (2.3)$$

where  $N$  is the set of all buses,  $P_k$  is the active power injection at bus  $i$ ,  $Q_k$  is the reactive power injection at bus  $i$ ,  $\delta_i$  is the voltage angle at bus  $i$ ,  $V_i$  is the voltage magnitude at bus  $i$ ,  $G_{ki}$  and  $B_{ki}$  are the real and imaginary elements of the bus admittance matrix at position  $(k, i)$ . In this load OPF, the amount of load ( $L^n$ ) to be controlled is minimized while satisfying the power flow constraints ( $PF$ ) and the transmission line flow ( $Pflow_{ij}$ ) limit. To satisfy the line-flow limit the sensitivity of the power flow in a line  $l$  after a change in power at a bus  $j$  was calculated. The

controllable loads can be controlled with a minimum and a maximum, meaning that this information would be available at the moment of optimization. An overloaded line case implementing the local OPF while interacting at the same time with the distributed intelligent agents is presented. The agents gather information about the amount of load that can be controlled at a certain moment in time. Remember that this information is collected from the smart meters and the distribution agents that are currently distributed in the distribution power network. These agents were simulated using JADE (Java Agent Development Framework).

### 2.2.1.1 Linear Programming Formulation

In this case, the OPF was solved using linear programming. The first step for the LP OPF formulation is to linearize the objective function. The objective function is:

$$f(x, u) = C_t = \sum_{z \in L} C_z(L^n) \quad (2.4)$$

where

$$C_z(L^n) = a_z + b_z L^n + c_z (L^n)^2 \quad (2.5)$$

For this particular example a cost was associated to connecting the loads. All of the controllable loads will have the same cost; thus, the optimization would only minimize the amount of controllable loads based on the equality and inequality constraints of the problem. Also note from (2.4) and (2.5) that the cost load function to be minimized is a quadratic function. The quadratic function can be approximated by dividing it into segments, each having a designated slope. Three

segment were chosen for this case. Figure 2.4 shows the segments represented by  $L_1^n, L_2^n, L_3^n$  and the slopes for each segment  $s_{i1}, s_{i2}, s_{i3}$ .

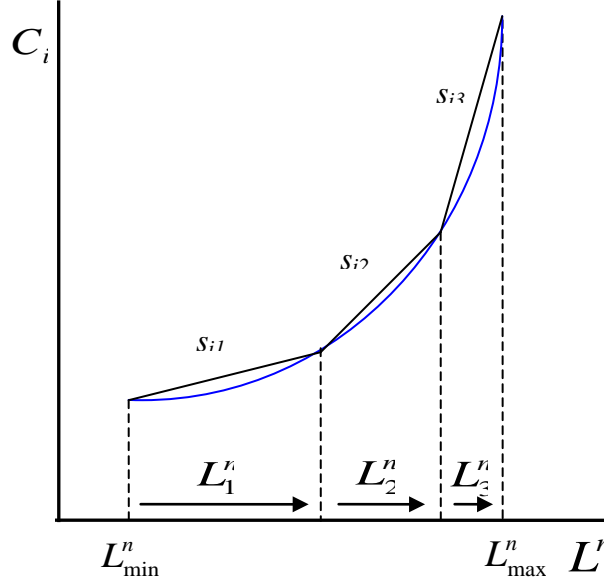


Figure 2.4: Linearization of the cost generation function

Then, the cost function is

$$C_z(L^n) = C_z(L_{\min}^n) + s_{i1}L_1^n + s_{i2}L_2^n + s_{i3}L_3^n, \quad i \in G \quad (2.6)$$

with constraint

$$0 \leq L_k^n \leq L_k^{n+}, \quad \text{for } n \in G, \quad k=1,2,3 \quad (2.7)$$

where  $L_k^{n+}$  is the difference between the starting and ending points of segment  $k$ .

The cost function is now made up of a linear expression in the  $L_k^n$  values. The active power output for load bus  $i$  is re-defined as

$$L^n = L_{\min}^n + L_1^n + L_2^n + L_3^n \quad (2.8)$$

Using linear programming, the solutions of the optimization problem are the control variables of the problem. The system's state variables (voltages and angles) and power conservation equality constraints are not directly included in

the LP optimization. Rather, constraints are set up in the LP that reflect the influence of changes in the control variables only.

Included in the LP OPF formulation is the constraint representing the power balance between active power generated and active power consumed by loads in the system, expressed as

$$P_{gen} - P_{load} = 0 \quad (2.9)$$

It is desired to express this power balance constraint as a linear function of the control variables. Thus, we take the derivative with respect to the vector of control variables  $\mathbf{u}$ , that in our case are the loads, and obtain the following expression:

$$\sum_u \left( \frac{\partial P_{gen}}{\partial u} \right) \Delta u - \sum_u \left( \frac{\partial P_{load}}{\partial u} \right) \Delta u = 0 \quad (2.10)$$

where  $\sum_u \left( \frac{\partial P_{gen}}{\partial u} \right)$  and  $\sum_u \left( \frac{\partial P_{load}}{\partial u} \right)$  are the sensitivity factors of the generated active power and consumed active power with respect to control variables  $\mathbf{u}$ , respectively. The change in control variables  $\Delta \mathbf{u} = \mathbf{u} - \mathbf{u}^0$ , where  $\mathbf{u}$  is the vector over which the objective function is minimized and  $\mathbf{u}^0$  is the result for the active power generation from the basic power flow. We proceed to substitute  $\Delta \mathbf{u}$  in (2.10), resulting in

$$\sum_u \left( \frac{\partial P_{gen}}{\partial u} \right) \mathbf{u} - \sum_u \left( \frac{\partial P_{load}}{\partial u} \right) \mathbf{u} = K_p \quad (2.11)$$

where



$$K_p = \sum_u \left( \frac{\partial P_{gen}}{\partial u} \right) u^0 - \sum_u \left( \frac{\partial P_{load}}{\partial u} \right) u^0 \quad (2.12)$$

Please note that the sensitivity factors and  $K_p$  are known constants; thus, (2.11) is a linear equality constraint depending only on vector  $u$ .

Taking advantage of the fact that inequality constraints are easier to include in the LP formulation, the OPF problem is extended to consider the maximum active power transfer of the transmission lines in the system that need to be relieved from overloads. The active power transfer or flow passing through a line connecting buses  $(i,j)$  can be computed with the results of the basic power flow state variables as

$$Pflow_{ij} = -G_{ij}V_i^2 + V_iV_j[G_{ij}\cos(\delta_i - \delta_j) + B_{ij}\sin(\delta_i - \delta_j)] \quad (2.13)$$

The inequality constraint on the lines active power transfer can be express as

$$Pflow_{ij} \leq Pflow_{ij}^{\max} \quad (2.14)$$

This constraint is modeled by forming a Taylor's series expansion of the active power transfer and only retaining the linear terms:

$$Pflow_{ij} = Pflow_{ij}^0 + \sum_u \left( \frac{\partial Pflow_{ij}}{\partial u} \right) \Delta u \leq Pflow_{ij}^{\max} \quad (2.15)$$

Substituting again  $\Delta u = u - u^0$  into (2.15), we obtain

$$\sum_u \left( \frac{\partial Pflow_{ij}}{\partial u} \right) u \leq Pflow_{ij}^{\max} - K_f \quad (2.16)$$

where

$$K_f = Pflow_{ij}^0 + \sum_u \left( \frac{\partial Pflow_{ij}}{\partial u} \right) u^0 \quad (2.17)$$

Section 2.2.1.2 provides further detail on the computation of sensitivity factor  $\sum_u \left( \frac{\partial Pflow_{ij}}{\partial u} \right)$ . In the LP algorithm the transmission line inequality constraints are not added at every iteration. The only inequalities that the algorithm has at every iteration are the ones presented in (2.7). Instead, the transmission line inequalities are verified at each iteration, and when they are not satisfied the inequality is added. In this way the algorithm is simplified and converges faster.

The LP OPF is solved as follows:

Step (1): Set the initial power flow conditions.

Step (2): Solve a power flow. This give us the initial generation for each generator.

Step (3): Obtain linearized constraints using equations (2.11) and (2.12)

Step (4): Set up and solve LP for the new control variables in each bus:

$$u_n = L_1^n + L_2^n + L_3^n \quad , n \in L \quad (2.18)$$

(Note that the LP problem can be easily placed in equality and inequality matrices and solved using Matlab® function *linprog*.)

Step (5): Check the feasibility of new vector  $u$  (loads's active power) by solving a new basic power flow.

Step (6): Check if load inequality constraints are still satisfied to verify that LP solution is feasible in the nonlinear original problem (recall that LP linearizes nonlinear functions and it is important to double check the

result with the original system). If the constraints are not satisfied, return to Step (4) to adjust the control variables.

Step (7): Check line MW flow limits. If they are not satisfied, we have to add a new inequality constraint using equation (2.16) and return to Step (4). If the line MW flow constraints are satisfied, the algorithm is stopped.

### 2.2.1.2 Sensitivity of Line Flows with Respect to Changes in Load

Linear sensitivity coefficients give an indication of the change in one system quantity (e.g., MW flow in a line) as another quantity is varied (e.g., generator MW output). These linear relationships are essential for the application of linear programming. Note that as the adjustable variable is changed, it is assumed that the power system reacts so as to keep all the power flow equations satisfied. As such, linear sensitivity coefficients can be expressed as partial derivatives, take for example

$$\frac{\partial P_{flow_{ij}}}{\partial L^n} \quad (2.19)$$

Equation (2.19) shows the sensitivity of the active power flow (MW) between buses  $(i,j)$  with respect to the active load at bus  $n$ . In this case the only sensitivities considered are the active power flow limits.

The following procedure is used to linearize the AC transmission system model for a power system to calculate the sensitivity coefficients. Equations (2.2)

and (2.3) described the bus power injection. At each bus the following equality has to be satisfied:

$$P_i(V_i, \delta_i) = P_i^{gen} - P_i^{load} \quad (2.20)$$

The set of equations that represents the first order approximation of the AC network around the initial point is the same as that generally used in the Newton power flow algorithm. That is

$$\sum \frac{\partial P_i}{\partial V_j} \Delta V_j + \sum \frac{\partial P_i}{\partial \delta_j} \Delta \delta_j = \Delta P_i^{load} \quad (2.21)$$

where  $i$  is the index of generators other than the slack bus and  $j$  is the index of the voltages and angles other than the reference bus. This is true because in the AC power flow the slack bus is dependent of the rest of the system. Note that this equation can be placed in matrix form for easier manipulation as follows:

$$\begin{bmatrix} \frac{\partial P_2}{\partial \delta_j} & \frac{\partial P_2}{\partial V_k} \\ \vdots & \vdots \\ \frac{\partial P_i}{\partial \delta_j} & \frac{\partial P_i}{\partial V_k} \end{bmatrix} \begin{bmatrix} \Delta \delta_j \\ \Delta V_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \cdots 0 \\ \vdots & \ddots \\ 0 \cdots 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta P_2^{load} \\ \vdots \\ \Delta P_k^{load} \end{bmatrix}, i \in NPV, j \in N, j \neq 1, k \in NPQ \quad (2.22)$$

where  $NPV$  is the set of buses where the power injection and voltage magnitude are specified (PV buses),  $NPQ$  is the set of buses where the active and reactive power are specified (PQ buses),  $\delta_i$  is the voltage angle at bus  $i$ , and  $V_i$  is the voltage magnitude at bus  $i$ .

This equation can be placed into a more compact format as follows:

$$[J_{px}] \Delta x = [J_{pu}] \Delta u \quad (2.23)$$

where the vector  $\mathbf{x}$  is the state vector of voltages and phase angles other than the reference and  $\mathbf{u}$  is the vector of control variables. The slack bus cannot be placed in this formulation because it would make the matrix singular and the inverse of  $J_{px}$  could not be calculated. Now, assuming that there are several transmission system dependent variables,  $h$ , the sensitivity with respect to changes in the control variables can be computed. This quantity can be expressed as a function of the state and control variables as follows:

$$h = [Pflow_{ij}(V, \delta)] \quad (2.24)$$

As before, a linear version of these variables around the operating point can be express as follows:

$$\Delta h = \begin{bmatrix} \frac{\partial h_1}{\partial \delta_j} & \frac{\partial h_1}{\partial V_k} \\ \vdots & \vdots \\ \frac{\partial h_r}{\partial \delta_j} & \frac{\partial h_r}{\partial V_k} \end{bmatrix} \begin{bmatrix} \Delta \delta_j \\ \Delta V_k \end{bmatrix} + \begin{bmatrix} \frac{\partial h_1}{\partial P_2^{load}} & \dots & \frac{\partial h_1}{\partial P_i^{load}} \\ \vdots & & \vdots \\ \frac{\partial h_r}{\partial P_2^{load}} & \dots & \frac{\partial h_r}{\partial P_i^{load}} \end{bmatrix} \begin{bmatrix} \Delta P_2^{load} \\ \vdots \\ \Delta P_k^{load} \end{bmatrix}, i \in NPV, j \in N, j \neq 1, k \in NPQ \quad (2.25)$$

where  $h_i$  is the function of the line  $kj$  MW flow and goes from the first line to the total number of lines  $r$ . Rearranging them into a compact format using the vectors  $\mathbf{x}$  and  $\mathbf{u}$  as before

$$\Delta h = [J_{hx}] \Delta \mathbf{x} + [J_{hu}] \Delta \mathbf{u} \quad (2.26)$$

Eliminating the  $\Delta \mathbf{x}$  variables using the equation (2.23) and rearranging,

$$\Delta \mathbf{x} = [J_{px}]^{-1} [J_{pu}] \Delta \mathbf{u} \quad (2.27)$$

Then substituting into equation (2.26) the following is obtained:

$$\Delta h = [J_{hx}] [J_{px}]^{-1} [J_{pu}] \Delta u + [J_{hu}] \Delta u \quad (2.28)$$

This last equation computes the linear sensitivity coefficients between the transmission lines MW flow and the active load power.

### 2.2.2 Agent Simulation in JADE and OPF Algorithm

JADE is a JAVA framework for developing FIPA (foundation for intelligent physical agents) compliant agent applications and was the platform used in the agent simulation presented in this section. Agents can be created and simulated using the JADE platform, and the power distribution system can be modeled using Matlab. Thus a connection with Matlab can be established to obtain power-flow and OPF optimization results. The JADE agents used the FIPA Contract Net Interaction protocol presented in Section 2.1.4.

The following OPF algorithm was implemented with the simulated JADE agents:

Step 1) Each bus agent (B\_A) gets the bus voltage magnitude, angle, and line power flows of directly connected lines to the bus from Matlab. This information is used in the Load OPF.

Step 2) Each B\_A sends data to the transmission region agent (TR\_A) every time there is a change in the data obtained from Matlab. Part of the data includes the information about load that can be controlled. The agent-to-agent communication protocol was explained in Section 2.1.4.

Step 3) If a line outage or a line overflow is detected, then the regional agent performs the Load OPF with the most recent available data. Once a solution is obtained, the result is sent to each B\_A.

Step 4) After the B\_A verifies that the amount of load requested by TR\_A can be controlled, the B\_A performs the control. A new power flow is obtained and the data is collected by the B\_A and sent again to the TR\_A.

Step 5) Once all of the B\_A agents perform the requested load control, the algorithm stops.

### **2.2.3 Case Study for the OPF Algorithm**

The example power system used in the OPF case is presented in Figure 2.5. This case results from the disconnection of line 2-5 because of an outage. The affected system has one overloaded line (2-6) at 92%; but the desired value is to be below 84%. The agents are controlling three loads that are also identified in Figure 2.5. The amount of load that can be controlled at each of the buses, as well as the results, is shown in Table 2.3.

After the load OPF was calculated, the most severe line 2-6 overload was reduced from 92% to 84%. Table 2.3 shows the results of the OPF, as well as the original and controllable loads. In order to achieve this goal, the net load at bus 6 was reduced from 110 MW to 83.48 MW. This result was obtained by connecting some pluggable hybrid cars that were simulated as a generator. The net effect on the bus is represented as a load shedding because this injection of pluggable hybrids would be in the distribution network and would be seen in the

transmission network as a change in the bus injection. But by coordinating the load response and the resulting bus injection change, the desired line flow can be obtained. Note that the load in bus 7 also has a net load change. This is a consequence of satisfying the line 3-7 loading constraint of 85%. It is important to mention that these results shown in Table 2.3 were obtained using the agent scheme presented in Figures 2.3.

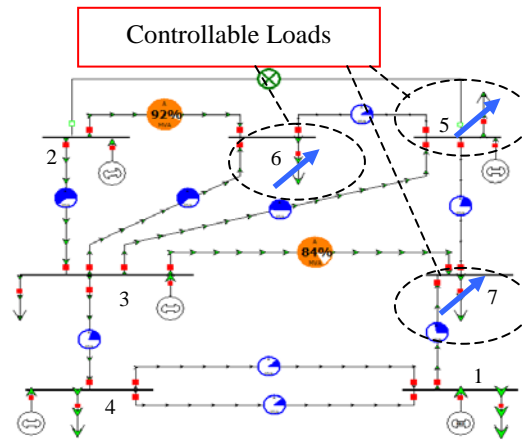


Figure 2.5: Case study and controllable loads

Table 2.3: Results for the OPF algorithm

Bus	Original Load (MW)	Amount of Load Controllable (MW)	Net Load after OPF (MW)
5	80	30	80
6	110	30	83.48
7	130	50	80

The bus agents (B\_A) gather the data measurements from the connected bus. The regional agents (TR\_A) receive the data measurements from the bus agents and use this information to run the load OPF of the entire region. For a future



implementation, a larger and more extensive power network will be used to incorporate a decentralized optimization algorithm among the different transmissions regions (TR\_A). In this case a solution will be obtained by coordinating the cooperation from the agents in the different transmission regions. This type of analysis can help decide which solution is more suitable, the decentralized or the centralized approach.

#### **2.2.4 Second Case Study for the OPF Algorithm**

The example presented in Section 2.2.3 is a worst case scenario as the amount of load being controlled is significant. To illustrate a more realistic case in which control of the loads would be reasonable, the following example is presented.

Line 3-7 has a real power limit of 82.3 MW and at the moment is just above that limit with a power flow of 82.35 MW. There are some penalties to the utility if the power exceeds that real power limit constraint. As it is just a small violation of the limit, this is a problem that can be solved easily by controlling the loads of the system.

The same algorithm using the agents that was presented in Section 2.2.3 is performed. Now it will show how the agents would have to interact with the distribution agents (D\_A). After the load control OPF was performed, the algorithm determined that the load at bus 7 had to be reduced by 140 kW. TR\_A sent the request to B\_A and the B\_A agreed to control that amount of load; the B\_A also performed a load OPF to determine which loads at the distribution network have to be controlled. For this case, the same 13-bus feeder used in the

loss minimization case was implemented as the distribution network. There were two agents, one controlling buses 1 - 6 and the other buses 7-13. The loads to be controlled are located at buses 6, 7, 8, 9, 11 and 12 as shown in Figure 2.6. The first agent controls the load at bus 6 and the rest of the loads are controlled by the second agent. The results are presented in Table 2.4. Again it is assumed that this result was obtained by connecting some pluggable hybrid cars that were simulated as a generator.

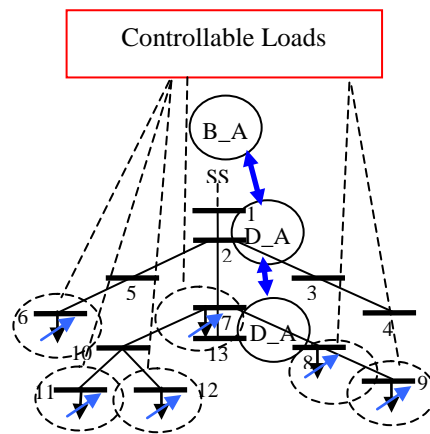


Figure 2.6: Distribution network with controllable loads

Table 2.4: Results for the distribution load control OPF case 1

Bus	Original Load (kW)	Amount of Load Controllable (kW)	Net Load after OPF (kW)
6	2.30	0.50	2.15
7	1.925	0.40	1.725
8	1.70	0.40	1.5
9	0.68	0.30	0.48
11	1.7	0.50	1.35
12	1.28	0.30	0.98

Once the solution is obtained by the B\_A, the results are sent to the D\_A in order for them to control the load. Then each D\_A sends a confirmation to the B\_A after the load control is performed. The B\_A sends a confirmation to the TR\_A to stop the algorithm, only after all of the D\_As have confirmed that loads were controlled.

The last simulation was performed using the simulated agents in JADE integrated with a real power system simulation run in Matlab. It is important to show the implications of these algorithms on both power networks. This type of analysis was not considered in the past but certainly is going to be in the future because detailed load data will be available.

The simulation took about 25 seconds to run for two reasons. Two different networks were simulated. Thus two different connections to Matlab are needed and there are agents in the distribution and the transmission network. Every time a B\_A communicates with the TR\_A, the TR\_A responds to each B\_A message one at a time. The messages are in a query, and once the message is addressed, another one is addressed. Each agent communication has its own message ID and each agent has its own name, so the agents can keep track of whom they are communicating with. All of this takes time to verify. For this type of algorithm each B\_A communicates with the TR\_A at least five times. This also is the case between the B\_A and the D\_A inside the distribution network. This analysis was implemented in a relatively small case; thus, if these results are extrapolated to larger systems, the simulation time increases as well as the complexity of the problem. For these larger cases, it is better to simulate different regions of the grid

in different computers so that each computer can parallelize the solution of the algorithm.

The size of the messages is about 1500 bytes. The typical time it takes to send a message and have it received by another agent is around 1 to 200 milliseconds. The time depends on the network conditions and the process running inside the algorithm. For example, messages will take longer if the agents need to compute an optimization problem and then send the results to other agents.

In the next section a more detailed distributed control architecture scheme is presented. The new algorithm will allow for the integration of a communication network into the analysis. Also, it will separate the different power networks in regions, making it easy to integrate the simulation with real controllable hardware devices.

## **2.3 Incident Command System**

Members of a chain of command structure such as the Incident Command System (ICS) follow a line of authority and responsibility. The ICS is a “systematic tool used for the command, control, and coordination of an emergency response” [2]. This system is used by firefighters and other emergency personnel for efficiently handling the emergency scenarios they face daily. From the widespread successful uses of this system, it has proven to be effective for dealing with emergencies and with large numbers of responders who may not all work together normally but have the same goals for the incident. Interestingly, a similar framework is needed for the intelligent control of power system devices to

respond efficiently when the power system is in crisis. In the ICS, each individual reports to only one supervisor. The individuals work in groups, and the group members report to a particular supervisor or officer who in turn reports to another specific officer. The functional unit with the highest authority is called command. Below command may be different sections, branches, functional groups, and geographical divisions [2]. The resources which actually perform the task are at the lowest level in the chain of command.

Initially the ICS was developed in the 1970s by fire services in California and Arizona as a management method to clarify command relationships for large-scale incidents [2]. Then it was applied to other emergency incidents as an effective way to manage the operation and cooperation during these incidents. One big application was to implement the ICS in the transportation sector to manage highway incidents. It provides an effective division of responsibilities among the many individuals that respond to an emergency by clearly establishing the chain of command of the management staff and the lower level chiefs and individuals.

For the power system events of interest in this thesis, the individual end-user real and reactive-power-controllable devices are the resources. Similar to the personnel resources in the ICS, end-user devices do not normally work together, but they have the same goal in a crisis.

Figure 2.7 shows the power grid as it is currently configured. A central EMS supervises conditions over the bulk transmission system. The transmission system meets the distribution system at the feeder relays, each of which serves a set of

downstream relays (Control Relays). The downstream relays control the delivery of power to various loads, which, as the smart grid continues to grow, will increasingly be regulated by a controller.

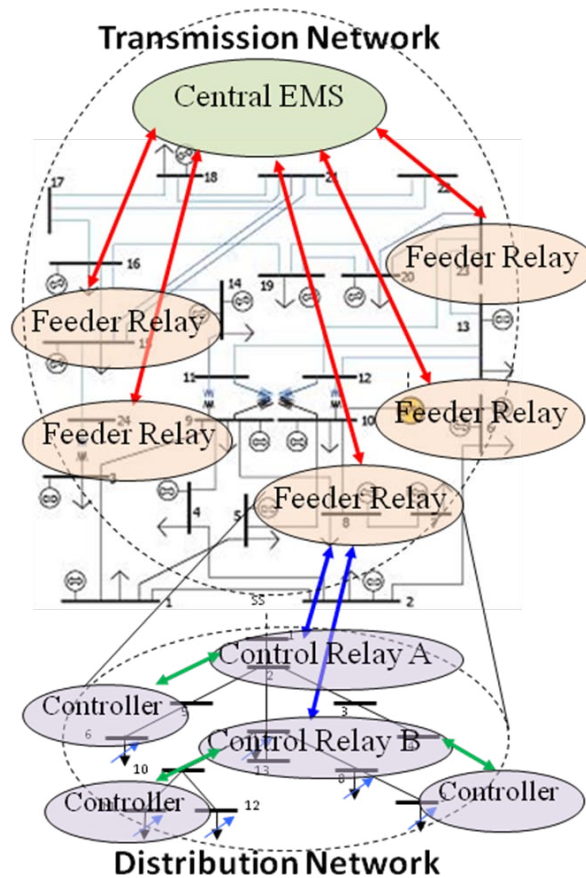


Figure 2.7: Transmission-distribution block diagram

In keeping with the ICS model, let us divide the nodes shown in Figure 2.7 into distinct supervisor-employee groupings called realms. Each realm consists of a top layer and a bottom layer. Each device in the top layer of a realm can supervise and control the activity of a set of devices in the bottom layer of the realm. The top-level devices in each realm do not communicate directly with any devices lower in the hierarchy than the bottom-level devices in their realm.

Instead, if control actions need to be taken further down in the hierarchy, the bottom-level devices of the realm, which are also the top-level devices of the next lower realm, will send the appropriate control signals downstream. This pattern of delegation is at the heart of the ICS model, and it provides a convenient way to segregate and secure communications on the smart grid.

In order to manage the information and control commands, the ICS command structure can be implemented using a multi-agent system architecture. The feeder relay will have an agent that manages the data and the control actions needed in the corresponding layer of the framework. The layered architecture can be implemented to allow two-way communications. In this type of vertical layer behaviors architecture, the flow of information comes from the bottom layers (get data) and from the top layers (control commands). Thus, the information goes in two different directions [1]. One way to coordinate this system is to implement a centralized multi-agent planning technique. In this technique, there is usually one coordinating agent that receives the information of other agents and plans/coordinates the individual actions of the bottom layer agents [1]. Then, since all the agents would have a single or specific task, the coordination of the system is rather straightforward. Another technique for coordinating this system involves a competitive negotiation in which each agent has a specific goal, and the degree of cooperation of individual agents is not known in advance. An example of this type of competitive negotiation is presented in [3-7] in which a set of agents is formed to coordinate a response to a problem while other agents coordinate a response to the same problem.

One of the main problems with the layered architecture is that if a direct communication link is lost from the central coordinating agent to the bottom agents, then the task cannot be performed. In order to solve this issue, the control algorithm would need to have a contingency response to this type of problem. In the ICS command structure presented in this thesis, the coordinating agent could be the Central EMS and the bottom layer agents could be the feeder relays and other relays connected to the Central EMS as shown in Figure 2.7. In Chapter 3, an algorithm that addresses these issues is presented. The algorithm complements the ICS model presented in [8] and is able to handle different control situations.

Note that this organization is flexible enough to handle problems in a decentralized way instead of always going through a central top-level controller. For example, if a top-level device on any of the lower realms detects a local problem, and if that device is suitably equipped to formulate a response, it can initiate correction of the problem by coordinating the devices beneath it. Such a situation would not need to rely on the Central EMS to send the control messages. Thus, potential applications of the framework extend beyond voltage control and could also benefit from the use of intelligent agents as in [9]. In general, such a scheme can be used to enact any corrective and preventive controls.

## **2.4 ICS Control Algorithm and Architecture**

The control algorithms that will be implemented using the proposed hierarchical arrangement of realms would have to be flexible enough to handle problems in a decentralized way instead of always in a centralized top-down



manner. In order to do this, the algorithm should be robust enough to handle situations locally while ensuring that the local actions do not affect other areas of the power grid.

#### **2.4.1 The Central Control Scheme**

To understand better this type of control involving realms and layers, consider the following algorithm.

- 1) The Central EMS detects a problem somewhere on the system. Based on the information and data received from the relays, it computes an aggregate response that would mitigate the problem. It formulates action requests and sends them through the hierarchy, where they are received by the feeder relays.
- 2) Once the request is received by the feeder relays, they must verify that the aggregated request can be performed. The feeder relays would verify the request by communicating to the downstream controllers and verifying that the aggregate response requested by the Central EMS can be performed.

Thus, to verify the request, the feeder relay agent computes a set of response actions that would allow it to fulfill the aggregated request. This is because the relay now needs to coordinate locally how the aggregate power requested from the Central EMS would be controlled at the specified moment in time in the distribution network.

- 3) After the feeder relay verifies that the control action can be performed and computes a set of responses for the controllers within its purview, it sends

a message to the Central EMS agreeing to do the requested control action.

If a local controller cannot perform the requested command, then the feeder relay should formulate a new local response. If a local solution within the distribution network cannot be found, then the Central EMS should be notified by the feeder relay and a new set of responses should be computed by the Central EMS.

- 4) At this point, if all of the feeder relays agree on the requested control, the Central EMS sends a command to the feeder relay confirming that the control action is going to be performed. Even if all of the feeder relays agree, the Central EMS will still have one last opportunity to cancel the control action, if, for example, it will affect other areas of the power network.
- 5) Once the confirmation from the Central EMS is received by the feeder relay, it will perform the control action by sending the commands to the controllers and relays to which they are connected.
- 6) Each controller then controls the loads under its supervision to meet the requests.
- 7) Once the control action is performed, the feeder relay will send a message to the Central EMS indicating that the control actions were completed. By doing this, the Central EMS will have a log of the control actions that are being performed in the power grid, which will allow various steps to be retraced if necessary.

With this type of control algorithm, the actions are not performed until after

there is verification from the top to the bottom layers that the algorithm can be performed without major consequences. There must be verification that the devices can be controlled and that the control of the devices is not going to create more problems for the power grid.

#### **2.4.2 The Local Control Scheme**

In the previous section, the algorithm was initiated from the very top of the command system (i.e. the Central EMS). However, there are going to be cases where the action would be initiated locally, say, from the feeder relays. For this type of scenario, the following algorithm will be implemented.

- 1) The feeder relay detects a problem for which it has the authority to initiate a local response.
- 2) After verifying that the control is not going to have a negative impact in the rest of the power grid (a task to be considered in future work), it will formulate control action requests and send them through the hierarchy where they are received by the load controllers.
- 3) Once the control action is performed, the feeder relay will send a message to the Central EMS indicating that the control actions were performed. By doing this, the Central EMS will have a log of the control actions that are being performed in the power grid.
- 4) At this point the Central EMS can determine if the control action will affect other regions of the power grid. If there is a negative effect in other regions of the grid, then a solution involving coordination with other

regions needs to be formulated and computed.

It is again important to notice that the Central EMS would have a log of all of the control actions that are being performed by the feeder relays. The purpose of this log is to have a record of what is happening in the grid. For this matter, the operators would know at any time what is being done in the grid and, if one of the control actions creates a problem or cannot be performed at a certain moment, they would consult the log and reverse the offending control actions.

The algorithms that will be presented in Chapters 3 and 4 of this thesis will follow the ICS architecture presented in this section.

## 2.5 References

- [1] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*, Chichester, England: John Wiley & Sons Ltd., 2007
- [2] “Simplified Guide to the Incident Command System for Transportation Professionals,” (2006, Feb.). U.S. Department of Transportation, Federal Highway Administration, Office of Operations, Tech. Rep. FWA-HOP-06-0004.
- [3] S. Curcic, C. S. Ozveren, L. Crowe, and P. K. L. Lo, “Electric Power Distribution Network Restoration: A Survey of Papers and a Review of the Restoration Problem,” *Electric Power Systems Research*, vol. 35, no. 2, 73-86, 1995.
- [4] K. L. Butler, N. D. R. Sarma, and V.R. Prasad, “A New Method of Network Reconfiguration for Service Restoration in Shipboard Power Systems,” in *IEEE Transmission and Distribution Conference*, vol. 2, April 1999, pp. 658 - 662.
- [5] J. M Solanki, N. Schultz, and W. Gao, “Reconfiguration for Restoration of Power Systems using Multi-Agent System,” in *37<sup>th</sup> Annual North American Power Symposium*, Oct. 2005, pp. 390-395.

- [6] A. A. Aquino-Lugo and T. J. Overbye, "Distributed Intelligent Agents for Service Restoration and Control Applications," in *40<sup>th</sup> Annual North American Power Symposium*, Sep. 2008, pp.1-7.
- [7] A. Irizarry-Rivera, M. Rodriguez-Martinez, B. Velez, M. Velez-Reyes, A. R. Ramirez-Orquin, E. O'Neill-Carrillo, and J. R. Cedeño, "Intelligent Power Routers: A Distributed Coordinated Approach for Electric Energy Processing Networks," *International Journal of Critical Infrastructures*, vol. 3, pp. 20-57, Dec. 2006.
- [8] K. M. Rogers, R. Klump, H. Khurana, A. Aquino-Lugo, and T. J. Overbye, "An Authenticated Control Framework for Distributed Voltage Support on the Smart Grid," *IEEE Transactions on Smart Grid*, vol. 1, no.1, pp. 40-47, June 2010.
- [9] A. Aquino-Lugo and T. J. Overbye, "Agent Technologies for Control Application in the Power Grid," in *43<sup>rd</sup> Hawaii International Conference on System Sciences*, Jan. 2009, pp. 1-10.

### **3 DISTRIBUTED CONTROL ALGORITHMS FOR REACTIVE RESOURCES**

The work presented in [1] investigates the integration of end-user reactive-power-controllable devices, such as solar panels and pluggable hybrid electric vehicles (PHEVs), to provide voltage support to the grid. In the previous work [1], it was demonstrated that, by controlling the reactive power of certain buses in the transmission network, the voltage profile through the grid can be maintained within the desired magnitude. However, in order to be able to control the reactive loads in the transmission network, the same analysis has to be performed in the distribution network. In the distribution network, the loads are served by different feeders and circuits. Therefore, the analysis is different from that of the transmission network, because the system is primarily radial. In this section, a strategy for identifying optimal control strategies on the distribution network for maintaining suitable voltage profiles is described.

#### **3.1 Distribution Power Flow**

Before presenting the voltage resources problem it is important to explain the basic idea behind the distribution power flow and its equations. Typical distribution feeders are radial and the common implemented iterative techniques for the transmission network power-flow studies are not used in distribution network feeder analysis because of poor convergence characteristics [1].

In order to solve the distribution power flow, an iterative ladder technique is used. This iterative technique is based on the ladder network theory of linear systems [2]. The technique is modified to incorporate the nonlinear characteristics of the distribution feeder [3]. For example, loads are modeled as constant PQ, and other devices such as transformers and shunt capacitors are modeled to represent the nonlinear characteristics of their behavior.

### 3.1.1 Distribution Feeder Line Models

Before continuing with the ladder iterative technique is important to explain the models and equations to represent a distribution system feeder. The following equations follow the same derivations developed and presented in [3]. Figure 3.1 represents an exact three-phase line segment model [3]. If Kirchhoff's current law (KCL) is applied for the line currents at node  $m$ , the following equations will be obtained:

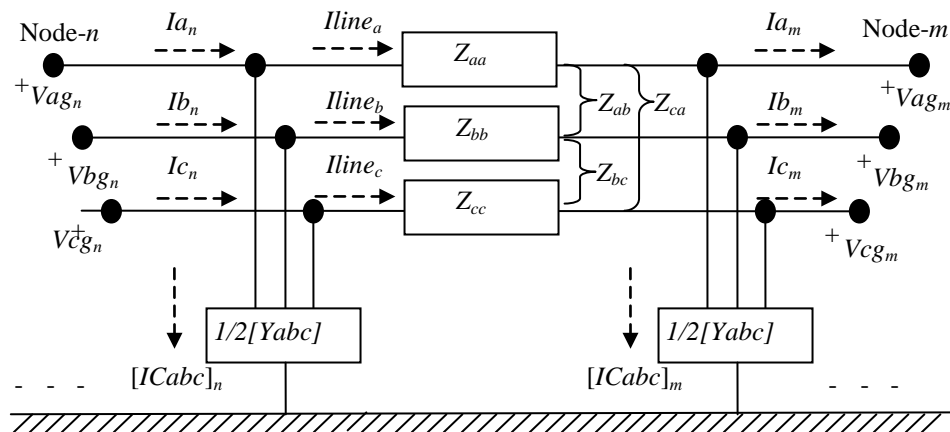


Figure 3.1: Three-phase exact line distribution feeder segment

$$\begin{bmatrix} Iline_a \\ Iline_b \\ Iline_c \end{bmatrix}_n = \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix}_m + \frac{1}{2} \begin{bmatrix} Y_{aa} & Y_{ab} & Y_{ac} \\ Y_{ba} & Y_{bb} & Y_{bc} \\ Y_{ca} & Y_{cb} & Y_{cc} \end{bmatrix} \cdot \begin{bmatrix} V_{ag} \\ V_{bg} \\ V_{cg} \end{bmatrix}_m \quad (3.1)$$

Equation (3.1) can also be expressed in a simplified form:

$$[Iline_{abc}]_n = [I_{abc}]_m + \frac{1}{2} [Y_{abc}] \cdot [VLG_{abc}]_m \quad (3.2)$$

Now applying Kirchoff's voltage law, the following equation is obtained:

$$\begin{bmatrix} V_{ag} \\ V_{bg} \\ V_{cg} \end{bmatrix}_n = \begin{bmatrix} V_{ag} \\ V_{bg} \\ V_{cg} \end{bmatrix}_m + \begin{bmatrix} Z_{aa} & Z_{ab} & Z_{ac} \\ Z_{ba} & Z_{bb} & Z_{bc} \\ Z_{ca} & Z_{cb} & Z_{cc} \end{bmatrix} \cdot \begin{bmatrix} Iline_a \\ Iline_b \\ Iline_c \end{bmatrix}_m \quad (3.3)$$

Equation (3.3) can also be expressed in a simplified form:

$$[VLG_{abc}]_n = [VLG_{abc}]_m + [Z_{abc}] \cdot [Iline_{abc}]_m \quad (3.4)$$

Substituting equation (3.2) into (3.4), and after some algebra, the following is obtained:

$$[VLG_{abc}]_n = [a] \cdot [VLG_{abc}]_m + [b] \cdot [I_{abc}]_m \quad (3.5)$$

where:

$$[a] = [U] + \frac{1}{2} \cdot [Z_{abc}] \cdot [Y_{abc}] \quad (3.6)$$

$$[b] = [Z_{abc}] \quad (3.7)$$

$$[U] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

Computing the currents entering node  $n$ ,



$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix}_n = \begin{bmatrix} Iline_a \\ Iline_b \\ Iline_c \end{bmatrix}_m + \frac{1}{2} \begin{bmatrix} Y_{aa} & Y_{ab} & Y_{ac} \\ Y_{ba} & Y_{bb} & Y_{bc} \\ Y_{ca} & Y_{cb} & Y_{cc} \end{bmatrix} \cdot \begin{bmatrix} V_{ag} \\ V_{bg} \\ V_{cg} \end{bmatrix}_n \quad (3.9)$$

Substituting equation (3.5) into (3.9), and after some algebra, the following is obtained:

$$[I_{abc}]_n = [c][VLG_{abc}]_m + [d] \cdot [I_{abc}]_m \quad (3.10)$$

where

$$[c] = [Y_{abc}] + \frac{1}{4} \cdot [Y_{abc}] \cdot [Z_{abc}] \cdot [Y_{abc}] \quad (3.11)$$

$$[d] = [U] + \frac{1}{2} \cdot [Z_{abc}] \cdot [Y_{abc}] \quad (3.12)$$

Now putting equation (3.5) and (3.10) into a partitioned matrix form

$$\begin{bmatrix} [VLG_{abc}]_n \\ [I_{abc}]_n \end{bmatrix} = \begin{bmatrix} [a] & [b] \\ [c] & [d] \end{bmatrix} \cdot \begin{bmatrix} [VLG_{abc}]_m \\ [I_{abc}]_m \end{bmatrix} \quad (3.13)$$

Now solving for voltages and currents at node  $m$ ,

$$\begin{bmatrix} [VLG_{abc}]_m \\ [I_{abc}]_m \end{bmatrix} = \begin{bmatrix} [a] & [b] \\ [c] & [d] \end{bmatrix}^{-1} \cdot \begin{bmatrix} [VLG_{abc}]_n \\ [I_{abc}]_n \end{bmatrix} \quad (3.14)$$

Since the determinant of the  $abcd$  matrix is

$$[a] \cdot [d] - [b] \cdot [c] = [U] \quad (3.15)$$

the inverse of the  $abcd$  matrix can easily be computed and equation (3.14)

becomes

$$\begin{bmatrix} [VLG_{abc}]_m \\ [I_{abc}]_m \end{bmatrix} = \begin{bmatrix} [d] & -[b] \\ -[c] & [a] \end{bmatrix} \cdot \begin{bmatrix} [VLG_{abc}]_n \\ [I_{abc}]_n \end{bmatrix} \quad (3.16)$$

When using the iterative ladder technique, at times it is necessary to get the voltages at node  $m$  as a function of voltages at node  $n$  and the currents at node  $m$ , thus solving equation (3.5) for the bus voltage at node  $m$ :

$$[VLG_{abc}]_m = [A] \cdot [VLG_{abc}]_n - [B] \cdot [I_{abc}]_m \quad (3.17)$$

where

$$[A] = [a]^{-1} \quad (3.18)$$

$$[B] = [a]^{-1} \cdot [b] \quad (3.19)$$

Finally, to obtain the line-to-line voltages the following equation is used:

$$\begin{bmatrix} V_{ab} \\ V_{bc} \\ V_{ca} \end{bmatrix}_n = [D] \cdot \begin{bmatrix} V_{ag} \\ V_{bg} \\ V_{cg} \end{bmatrix}_m \quad (3.20)$$

where

$$[D] = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.21)$$

Typically in the distribution network the shunt admittance of an overhead line is small and it can be neglected. Thus this simplified the previous equations. In Figure 3.2 the modified line segment of a distribution feeder is presented without the shunt admittances.

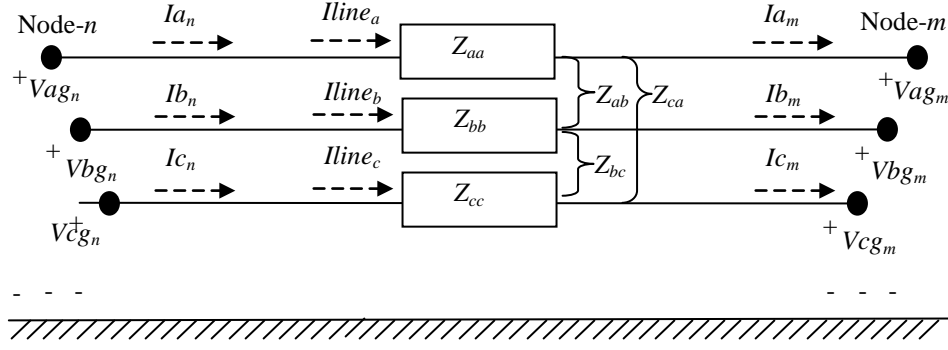


Figure 3.2: Three-phase exact line distribution feeder segment

Then the matrices in equations (3.5), (3.10), (3.16) and (3.17) become

$$[a] = [U] \quad (3.22)$$

$$[b] = [Z_{abc}] \quad (3.23)$$

$$[c] = [0] \quad (3.24)$$

$$[d] = [U] \quad (3.25)$$

$$[A] = [U] \quad (3.26)$$

$$[B] = [Z_{abc}] \quad (3.27)$$

Now all the necessary equations needed to develop the ladder iterative techniques have been explained. The following explanation of the ladder technique will be based on these equations.

### 3.1.2 Distribution Ladder Iterative Technique

The ladder iterative technique consists of two basics steps: the forward and the backward sweeps. These sweeps iterate until a solution of the source voltage is obtained within a specified tolerance. In this analysis the series and shunt components of the distribution feeder need to be modeled and the corresponding

equations need to be incorporated into the equations presented in Section 3.1.1. With these models the corresponding voltages and currents can be calculated. The details about the different distribution devices are beyond the scope of this thesis. Later in this chapter some of these models are going to be explained.

In order to understand the forward and backward sweep, a very simple example is going to be presented. The example is taken from [3]. Figure 3.3 shows a four-bus distribution feeder.

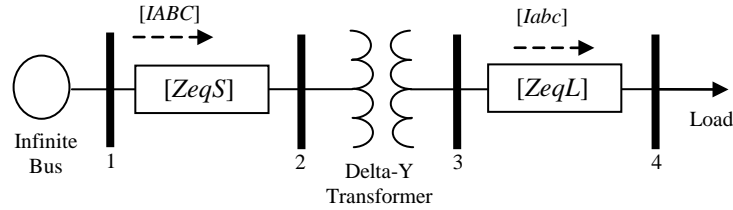


Figure 3.3: Four-bus distribution feeder

In this feeder there are four buses and three line segments. Each line segment has its own three-by-three impedance matrix ( $[ZeqS]$  and  $[ZeqL]$ ), and the Delta-Y transformer also has its own three-by-three impedance matrix. This is important because those are the impedance matrices that will be used in the voltage and current equations, just as explained in the previous section.

The ladder iterative technique starts with the forward sweep. For this, the voltages at bus 2, 3 and 4 are computed assuming a no load condition. Then the backward sweep starts by computing the currents from the bus 4 ( $[Iabc]$ ). These currents are the load current at bus 4. For this case the load was modeled as constant PQ. Thus this current is computed per phase using an equation similar to this:

$$I_{abc\_phase} = \left( \frac{S_{phase}}{V_{4\_phase}} \right)^* \quad (3.28)$$

where  $S_{phase}$  is the apparent power per-phase and  $V_{4\_phase}$  is the bus voltage at bus 4 per-phase.

Then the voltage at bus 3 is computed using equation (3.5), and the voltage and currents ( $[IABC]$ ) at bus 2 are computed using equations (3.5) and (3.10). Finally the voltage at node 1 is computed using equation (3.5). Now the voltage at bus 1 is compared to the specified value and if the error is bigger than the specified tolerance, then another forward sweep is started using the voltages and currents obtained from the previous iteration.

Now for the second forward sweep, the voltages at bus 2, 3 and 4 are computed using equation (3.17). Then another backward sweep is performed just as explained before. These iterations stop when the estimated voltage source falls below the specified tolerance.

With the previous example and equations in Section 3.1.1, the required knowledge to understand the equality constraints of the optimization problems was presented.

### 3.2 The Voltage Problem Formulation

The voltage control problem studied here has the following mathematical formulation:

$$\begin{aligned}
& \min \quad \sum_{k=1}^N \left( V_k(x, u) - V_{k_{spec}} \right)^2 \\
& s.t. \quad \text{DPF}(x, u) \\
& \quad \quad 0 \leq u \leq (u)^{\max}
\end{aligned} \tag{3.29}$$

where  $DPF$  are the distribution power-flow constraints. The power flow constraints are equality constraints describing the voltage and current relationship at each branch and node. In this case and in the rest of this thesis the distribution networks that are going to be analyzed are going to be single-phase. The relationships follow the same formulations and analysis presented in the Section 3.1.1. The single-phase representation of the network equations simplifies the analysis, but the same representation can easily be extended to three-phase systems by adding two more equations for the voltages and the same number for currents at each node. Also it is important to notice that to express these equations, even for the single-phase representation, each voltage and current equation needs to be expressed in its real and imaginary terms. Thus, for each voltage and current equation there is a real and imaginary equation describing the governing behavior of voltages and currents in a distribution network. Figure 3.4 presents a distribution feeder with the voltages and currents of interest.

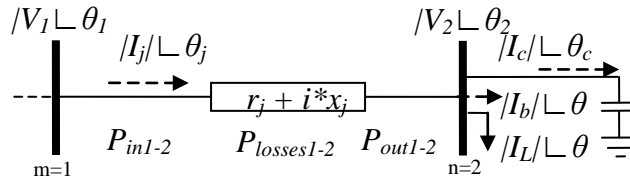


Figure 3.4: Branch diagram in a distribution feeder

The DPF equality constraints for the bus voltages are

$$\text{Re} \left[ \left( |V_m| \angle \theta_m \right) = \left( |V_n| \angle \theta_n \right) + \left( (r + i * x) * \left( |I_j| \angle \theta_j \right) \right) \right] \quad (3.30)$$

$$\text{Im} \left[ \left( |V_m| \angle \theta_m \right) = \left( |V_n| \angle \theta_n \right) + \left( (r + i * x) * \left( |I_j| \angle \theta_j \right) \right) \right] \quad (3.31)$$

Equations (3.30) and (3.31) may be rewritten as

$$\begin{aligned} & \left( |V_n| \cdot \cos \theta_n \right) = \\ & \left( |V_m| \cdot \cos \theta_m \right) - \left( |I_j| \cdot \cos \theta_j \right) \cdot (r) + \left( |I_j| \cdot \sin \theta_j \right) \cdot (x) \end{aligned} \quad (3.32)$$

$$\begin{aligned} & \left( |V_n| \cdot \sin \theta_n \right) = \\ & \left( |V_m| \cdot \sin \theta_m \right) - \left( |I_j| \cdot \cos \theta_j \right) \cdot (x) - \left( |I_j| \cdot \sin \theta_j \right) \cdot (r) \end{aligned} \quad (3.33)$$

The DPF equality constraints for the line currents are

$$\text{Re} \left[ |I_j| \angle \theta_j = |I_L| \angle \theta_L + |I_b| \angle \theta_b \right] \quad (3.34)$$

$$\text{Im} \left[ |I_j| \angle \theta_j = |I_L| \angle \theta_L + |I_b| \angle \theta_b \right] \quad (3.35)$$

where

$$\begin{aligned} |I_L| \angle \theta_L &= \frac{|S_{Load}|}{|V_n|} \angle (\theta_{Load} - \theta_n) = \\ & \left( \frac{S_{Load}}{V_n} \right)^* = \left( \frac{P_{Load} + i \cdot Q_{Load}}{|V_n| \cos \theta_n + i \cdot |V_n| \sin \theta_n} \right)^* \end{aligned} \quad (3.36)$$

In this problem, the reactive load will be controlled and can be represented as

$$Q_{Load} = Q_{LoadOld} - \Delta Q_{Load} \quad (3.37)$$

$\Delta Q_{Load}$  will be calculated in the optimization problem.

Equations (3.34) and (3.35) may be rewritten as

$$|I_j| \cdot \cos \theta_j = |I_L| \cdot \cos \theta_L + |I_b| \cdot \cos \theta_b \quad (3.38)$$

$$|I_j| \cdot \sin \theta_j = |I_L| \cdot \sin \theta_L + |I_b| \cdot \sin \theta_b \quad (3.39)$$

After some algebra, if equations (3.36) and (3.37) are substituted into (3.38) and (3.39), the resulting load current components with corresponding real and imaginary parts will be

$$\begin{aligned} & |I_L| \cdot \cos \theta_L + i \cdot |I_L| \cdot \sin \theta_L = \\ & \frac{P_{Load} \cdot |V_n| \cos \theta_n + Q_{LoadOld} \cdot |V_n| \sin \theta_n - \Delta Q_{Load} \cdot |V_n| \sin \theta_n}{(|V_n|)^2} + \\ & i \cdot \left( \frac{P_{Load} \cdot |V_n| \sin \theta_n - Q_{LoadOld} \cdot |V_n| \cos \theta_n + \Delta Q_{Load} \cdot |V_n| \cos \theta_n}{(|V_n|)^2} \right) \end{aligned} \quad (3.40)$$

For this problem, the inequality constraints are simply the maximum and minimum values the bus voltages can have and the maximum branch currents passing through the feeder. In this case, the cost function of the equations penalized the voltage inequality constraint for the buses in which we want the voltage to be above a certain value, typically 0.9 p.u. The proposed optimization problem can be solved using many optimization techniques including, for example, the Lagrangian approach. The details about this formulation and the solution used to solve it are going to be presented in the next section.

### 3.2.1 Newton's Method to Solve Optimization Problems

The optimization problem presented in the previous section can be formulated as a constrained nonlinear optimization problem as follows:

$$\begin{aligned} & \text{minimize} && f(z) \\ & \text{subject to} && g(z) = 0 \\ & && h(z) \leq 0 \end{aligned} \quad (3.41)$$



where  $z$  is the vector that consists of the  $x$  unknown variables and  $u$  control parameters; thus,  $z = [x \ u]^T$ ,  $f(z)$  is the objective function, and  $g(z)$  and  $h(z)$  represent the equality and inequality constraints, respectively. Also  $z \in \mathfrak{D}_1^n$  and it is assumed that  $f$ ,  $g$  and  $h$  are continuously differentiable functions from  $\mathfrak{D}_1^n$  to  $\mathfrak{D}_1$ ,  $\mathfrak{D}_1^m$  to  $\mathfrak{D}_1^n$  and  $\mathfrak{D}_1^q$  to  $\mathfrak{D}_1^r$  respectively.

Next, the problem is converted to an unconstrained minimization problem using the following Lagrange function [4] for this problem as

$$L(z, \lambda, \mu) = f(z) + \lambda^T g(z) + \mu^T h(z) \quad (3.42)$$

where  $\lambda$  is the vector of Lagrange multipliers related to the equality constraints and  $\mu$  is the vector of penalties that are applied when the inequality constraints are not satisfied.

To minimize the unconstrained problem, subject to the constraints, applying the necessary optimality conditions for an optimum  $z = [x^* \ u^*]^T$  we set the gradient of the Lagrange function to zero ( $\nabla L(y) = 0$ ), where  $y = [z \ \lambda \ \mu]^T$ . The gradient vector is divided into three parts corresponding to the partial derivatives of the Lagrange function with respect to each of its variables,  $x$ ,  $u$  and  $\lambda$ .

$$\nabla L_z = \frac{\partial L}{\partial z} = \frac{\partial f}{\partial z} + \left[ \frac{\partial g}{\partial z} \right]^T \lambda + \left[ \frac{\partial h}{\partial z} \right]^T \mu = 0 \quad (3.43)$$

$$\nabla L_\lambda = \frac{\partial L}{\partial \lambda} = g(z) = 0 \quad (3.44)$$

$$\nabla L_\mu = \frac{\partial L}{\partial \mu} = h(z) = 0 \quad (3.45)$$

Then applying the Newton methods to find the optimum, the following iteration is performed:

$$y^{(k+1)} = y^{(k)} + \alpha \Delta y^{(k)} \quad (3.46)$$

and  $\Delta y$  is obtained by solving

$$H(y^{(k)}) \cdot \Delta y^{(k)} = - \left[ L(y^{(k)}) \right]^T \quad (3.47)$$

where

$$H(y^{(k)}) = \nabla_y^2 L(y^{(k)}) \quad (3.48)$$

Thus using equations (3.43) to (3.45), the Hessian  $H(y^{(k)})$  can be evaluated and will have the following structure:

$$\nabla_y^2 L = \begin{bmatrix} \nabla_z^2 L & \nabla_{z\lambda}^2 L & \nabla_{z\mu}^2 L \\ \nabla_{\lambda z}^2 L & \nabla_{\lambda}^2 L & \nabla_{\lambda\mu}^2 L \\ \nabla_{\mu z}^2 L & \nabla_{\mu\lambda}^2 L & \nabla_{\mu}^2 L \end{bmatrix} = \begin{bmatrix} \nabla_z^2 L & \nabla_{z\lambda}^2 L & \nabla_{z\mu}^2 L \\ \nabla_{\lambda z}^2 L & 0 & 0 \\ \nabla_{\mu z}^2 L & 0 & 0 \end{bmatrix} \quad (3.49)$$

where

$$\nabla_{z\lambda}^2 L = \left[ \frac{\partial g}{\partial z} \right]^T = \left[ \nabla_{\lambda z}^2 L \right]^T \quad (3.50)$$

$$\nabla_{\mu z}^2 L = \left[ \frac{\partial h}{\partial z} \right]^T = \left[ \nabla_{z\mu}^2 L \right]^T \quad (3.51)$$

### 3.2.2 Ten and Thirty-Four-Bus Reactive Load Control Examples

The proposed optimization problem can be solved using many optimization techniques including, for example, the Lagrangian approach and Newton iteration scheme presented in Section 3.2.1. Thus the problem was formulated as a constrained nonlinear optimization problem as follows:

$$\begin{aligned}
& \text{minimize} && f(x,u) \\
& \text{subject to} && g(x,u) = 0 \\
& && h(x,u) \leq 0
\end{aligned} \tag{3.52}$$

where  $\mathbf{x}$  is the vector of unknown variables,  $\mathbf{u}$  is the vector of control parameters,  $f(x,u)$  is the objective function,  $g(x,u)$  and  $h(x,u)$  represent the equality and inequality constraints, respectively.

The problem is converted to an unconstrained minimization problem using the following Lagrange function for this problem:

$$L(x,u,\lambda,\beta) = f(x,u) + \lambda^T g(x,u) + \beta^T (u)^2 \tag{3.53}$$

or equivalently,

$$L(x,u,\lambda,\beta) = f(x,u) + \lambda^T g(x,u) + \beta_{\min}^T (0-u)^2 + \beta_{\max}^T \left(u - (u)^{\max}\right)^2 \tag{3.54}$$

where  $\lambda$  is the vector of Lagrange multipliers related to the equality constraints and  $\beta$  is the vector of penalties that are applied when the control variables are outside their minimum and maximum values. In this case, the controllable parameter is the  $\Delta Q_{Load}$  (controllable reactive load).

Thus using the Newton scheme, the optimization problem was tested on a 10-bus feeder and a 34-bus feeder. In both cases, the loads were modeled as constant PQ devices. The 10-bus system is shown in Figure 3.5. In the 10-bus system, the reactive loads to be controlled are located at buses 4, 5, 7, 8, 9 and 10. The results are presented in Tables 3.1 and 3.2.

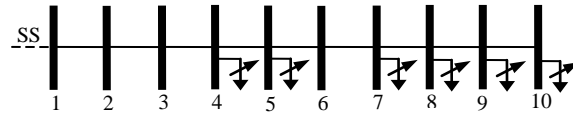


Figure 3.5: Ten-bus feeder at substation SS

Table 3.1: Reactive load kVARs for 9-bus system

Case	Controllable Load (kVARr)	Initial Load (kVARs)	Final Load (kVARs)
CV1: Bus 4	646	446	-200
CV1: Bus 5	1,940	1840	-100
CV1: Bus 7	210	110	-100
CV1: Bus 8	160	60	-100
CV1: Bus 9	330	130	-200
CV1: Bus 10	400	200	-200

Table 3.2: Bus voltage for 10-bus system

Case	Initial Volts (pu)	Final Volts (pu)
CV1: Bus 5	0.889	0.91879
CV1: Bus 7	0.85875	0.89384
CV1: Bus 8	0.83756	0.87613

From the results, it can be seen that the voltages in the system were increased through the control actions, and that all controllable reactive loads participated in the action. There are still voltages below the voltage target of 0.9 p.u. because it is the best solution that can be achieved for the 10-bus system.

In the second scenario, the algorithm was tested on a modified IEEE 34-bus distribution system presented in Figure 3.6. In this case (CV2 in Tables 3.3 and

3.4) the loads to be controlled are located at buses 17, 20, 22, 23, 25, 27, 29 and 30 respectively. In this case the voltages were initially below 0.9 p.u. from bus 17 to bus 34.

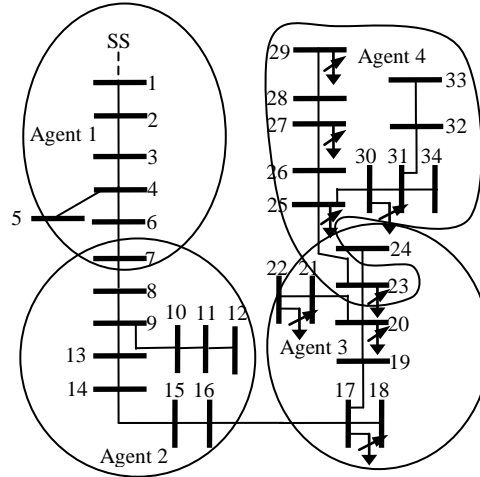


Figure 3.6: IEEE modified 34 bus feeder connected to substation SS

The results from this algorithm (Tables 3.3 and 3.4) show that using the algorithm can effectively find the amount of reactive load to be controlled. This fact is more visible in Table 3.4 for the case when only the voltage profile target was set to be at 0.9 p.u. For this case the total amount of controllable reactive load was 519 kVARs but only 415.9 kVARs was used to satisfy the desired set voltage. When the voltage profile target was 0.91 p.u., all of the controllable 519 kVARs of reactive load was used.

Table 3.3: Reactive load kVARs for 34-bus system

Case	Control Load (kVARs)	Initial Load (kVARs)	Final Load (kVARs) 0.91 pu case	Final Load (kVARs) 0.9 pu case
CV2: Bus 17	7	2	-5	-5
CV2: Bus 20	13	3	-10	3
CV2: Bus 22	95	75	-20	-20
CV2: Bus 23	31	1	-30	1
CV2: Bus 25	27	7	-20	-20
CV2: Bus 27	205	105	-100	-40.912
CV2: Bus 29	46	16	-30	-30
CV2: Bus 30	95	55	-40	-40

Table 3.4: Bus voltage for 34-bus system

Case	Initial Volts (pu)	Final Volts (pu) 0.91 pu case	Final Volts (pu) 0.9 pu case
CV2: Bus 19	0.85946	0.91088	0.90151
CV2: Bus 20	0.85945	0.91088	0.9015
CV2: Bus 21	0.85945	0.91087	0.9015
CV2: Bus 22	0.85829	0.9102	0.90082
CV2: Bus 23	0.85783	0.91017	0.90061
CV2: Bus 24	0.85783	0.91017	0.9006
CV2: Bus 25	0.85592	0.90928	0.89956
CV2: Bus 26	0.85588	0.90927	0.89954
CV2: Bus 27	0.85567	0.90921	0.89945
CV2: Bus 28	0.85554	0.90916	0.8994
CV2: Bus 29	0.85554	0.90916	0.8994
CV2: Bus 30	0.8556	0.90905	0.89933
CV2: Bus 31	0.85539	0.90885	0.89913
CV2: Bus 32	0.85537	0.90884	0.89912
CV2: Bus 33	0.85527	0.90875	0.89902
CV2: Bus 34	0.85537	0.90884	0.89911

### 3.3 Agent Simulations and Test-Bed Implementations

In order to test the algorithm in a realistic setting, a simulation test bed was created. The test bed includes agents simulated using JADE (Java Agent Development Framework) [5 - 7]. Remember that JADE is a JAVA framework for developing FIPA (foundation for intelligent physical agents) compliant agent

applications and is one of the most widespread agent-oriented and completely distributed middleware systems to create agents [5]. In the work presented in [8], a similar framework was created but worked only on a single computer. The testbed created in this thesis can have agents running in different computers and can communicate between them as long as they are connected to the same computer network. This will allow for a distributed implementation of the agents just as will occur in a real power system. In the real system, an agent would be housed in a control device, such as an intelligent relay. Therefore, each JADE-based agent models an agent that would be placed in an actual system control device.

The agents were modeled using the JADE platform, while the power network itself, including its loads, generators, and transmission lines, was modeled using Matlab. A connection between the agents and Matlab was established so that the agents could be informed of the state of the power system. For example, changes to the power system were modeled and solved in Matlab, and the resulting changes in system parameters, such as voltages and currents, were communicated to the JADE-based agents. Furthermore, the optimization algorithms were run and solved in Matlab. Agents could request an optimization from Matlab to evaluate control alternatives.

Now the implementation of the algorithm on this testbed is described. The system was designed to emulate the ICS architecture described in Chapter 2. Figure 3.7 shows the testbed in detail. Computer A simulates the transmission network and associated transmission-level agents. Similarly, computer B



simulates the distribution network and its associated agents. The ICS control algorithm presented in Section 2.4 was coded and executed on these computers to solve the reactive resources optimization problem described in Section 3.2.2.

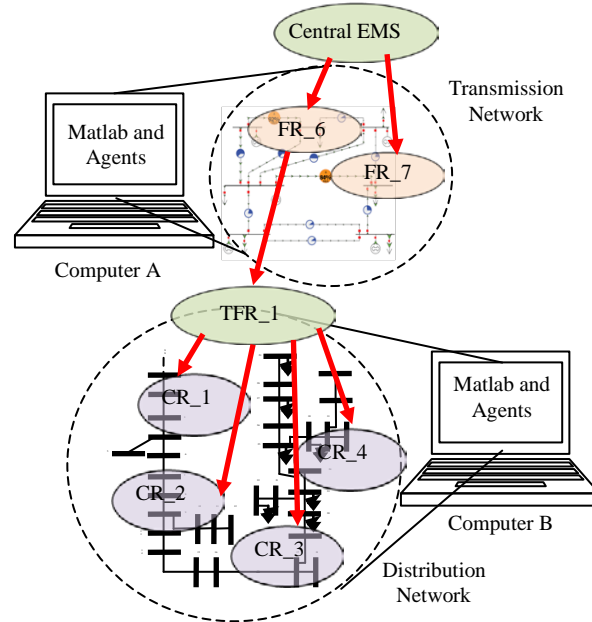


Figure 3.7: Agent and simulation test bed

The control algorithm was implemented as follows:

- 1) The Matlab-based power system model is recalculated based on new system conditions. These power model changes are then communicated to the agents housed in the Central EMS and downstream feeder relays and controllers, which are modeled using JADE.
- 2) The Central EMS detects a low voltage problem somewhere on the system. Based on the information and data received from the feeder relays (FR), it computes an aggregate response that would mitigate the problem by performing a transmission-level optimization problem that is similar to

the one presented in Section 3.2.2 but applied to a transmission network. In [9] there are more details on this algorithm. For this case, the equality constraints are the transmission network equations. After a solution of the optimization is obtained, the Central EMS agent will send the resulting requested load to be controlled through the hierarchy, where they are received by the feeder relays.

- 3) Once the request is received by the FRs, each FR will verify that the aggregated request can be performed. It does this by surveying the equipment downstream from it. Keep in mind that a transmission network bus usually consists of many substations, each of which aggregates many distribution networks. Therefore, the FR sees a multi-level network beneath it. We identify the top of these levels by the top feeder relay, or TFR. Thus, each distribution network feeder will have a TFR that will coordinate the controller relays (CR) of the feeder as shown in Figure 3.8. Note here that there could be multiple TFRs per FR.

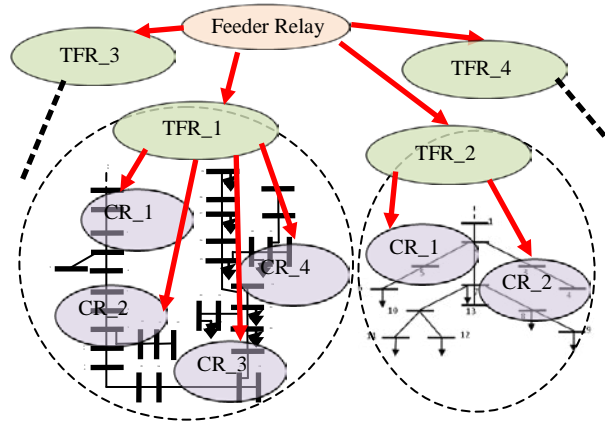


Figure 3.8: Feeder relays and top feeder relay agents in the distribution network

- 4) The FR relays would verify the request by communicating to the TFR.

The FR relay at this point knows how much load the TFR can control, but that does not mean that all of the reactive load can or should be controlled, because changes to one set of devices could have a harmful impact on other devices. Then, after this, the TFR will perform the voltage problem optimization to determine the amount of reactive load it can control. If a CR cannot perform the requested command, then the TFR should formulate a new local response. After a solution is obtained, a command is sent to the FR indicating the amount of load that the particular TFR can control.

- 5) Once the FR receives the controllable loads from all of the TFRs, it verifies that the aggregated request by the Central EMS agent can be performed.
- 6) After the FR verifies that the control action can be performed because all of the requested aggregated load can be controlled, it will send a message to the Central EMS agreeing to do the requested control action. If the FR cannot provide the control requested by the Central EMS, the algorithm proceeds to step 11.
- 7) At this point, the Central EMS sends a command to the FR confirming that the control action is going to be performed.
- 8) Once the confirmation from the Central EMS is received by the FR, it will send the control command to the TFRs, which will then send the specific commands to the connected CRs.

- 9) Each controller then sends a command to the load controllers under its supervision to meet the requests.
- 10) Once the control action is performed, the TFRs send a message to the FR indicating the control has been performed. Subsequently, the FR then sends a message to the Central EMS indicating that the control actions were performed. The algorithm then stops.
- 11) For the case identified in step 6 when the FR cannot control the requested aggregated load, the FR reports the amount of available load that is controllable. Then, the Central EMS will send a cancel command to the FR, and the FR will send a cancel command to the TFRs. At this point the Central EMS will try to find a new solution, and the algorithm will start again.

### **3.3.1 Agent Simulation Case Study**

In this section, an example to test the simulation is presented. The example presented in this case is a 7-bus transmission network with a low voltage of 0.94 p.u. at bus 6 and 0.98 p.u. at bus 7. The set-point values of their respective buses are 0.95 p.u. and 0.99 p.u.. Thus, a reactive support optimization algorithm was performed by the Central EMS agent, and the following loads were identified to be controlled: 3.52 MVARs for bus 6 and 17.083 MVARs for bus 7.

For this simulation, only the integration between the feeder relay at bus 6 (FR\_6) and the top feeder relay 1 (TFR\_1) was performed. The TFR\_1 is responsible for the same 34 bus system presented in Section 3.2.2. Again the same

simulation was performed for the 34-bus feeder and the results were the same. For this case, the voltage of the buses has to be improved to 0.9 p.u. In this distribution network, there are four agents (Figure 3.6). The first is responsible for buses 1 through 7 but no load is controlled. The second is responsible for buses 7 through 16 but no load is controlled. The third is controlling loads at buses 17, 20, 22 and 23 connected to those same buses and is responsible for buses 16 through 24. The fourth is controlling loads at buses 25, 27, 29 and 30 connected to those same buses and is responsible for buses 23 through 34, excluding 24. Once a solution for the reactive loads was obtained, the TFR sent the amount of load to be controlled to those relay controller agents. The results in Table 3.3 show that not every one of the available controllable loads needed to be used to satisfy the constraint. In other words, because of operational constraint in the distribution grid, not all of the available load can be used as requested by the FR. This is because there are cases in which setting voltages above certain values could affect the behavior of other devices such as tap-transformers or capacitors. Thus, using the algorithm to effectively determine the amount of load to be controlled is important because it ensures that operating constraints are obeyed.

The other TFR agents that are interacting with FR\_6 in Figure 3.8 were assumed to be connected to a distribution network feeder, but in this case only one TFR was connected to a simulated distribution network. However, this test bed can easily be extended to include other TFR agents if desired.

In the transmission network simulation, other feeder relays could have the same setup as the one presented in Figure 3.7. Thus the applications could be extended

to test or interact with bigger simulations. An interesting point is that the computers used to model the agents could be connected to real load control devices and thus should be able to model the effect of control strategies devised by the optimization algorithms. It is in the plan to perform this integration with actual load devices in future work.

### 3.4 References

- [1] C. Trevino, "Cases of difficult convergence in load-flow problems," in *IEEE Summer Power Meeting*, Los Angeles, 1970
- [2] W. H. Kersting and D. L. Medive, "An application of ladder network theory to the solution of three-phase radial load-flow problems," in *IEEE Winter Power Meeting*, New York, 1976.
- [3] W. H. Kersting, *Distribution System Modeling and Analysis*, 2<sup>nd</sup> ed. Boca Raton, FL: CRC Press, 2007.
- [4] D. P. Bertsekas, *Nonlinear Programming*, 2<sup>nd</sup> ed. Belmont, MA: Athena Scientific, 2003.
- [5] A. A. Aquino-Lugo and T. J. Overbye, "Distributed Intelligent Agents for Service Restoration and Control Applications," in *40<sup>th</sup> Annual North American Power Symposium*, Sep. 2008, pp.1-7.
- [6] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*, Chichester, England: John Wiley & Sons Ltd., 2007
- [7] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications-Part I: Concepts, Approaches, and Technical Challenges," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1743-1752, Nov. 2007.
- [8] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications-Part II: Technologies, Standards, and Tools for Building Multi-agent Systems," *IEEE Transactions on Power Systems*, vol. 22, pp. 1753 - 1759, Nov 2007.

- [9] K. M. Rogers, R. Klump, H. Khurana, A. Aquino-Lugo, and T. J. Overbye, "An Authenticated Control Framework for Distributed Voltage Support on the Smart Grid," *IEEE Transactions on Smart Grid*, vol. 1, no.1, pp. 40-47, June 2010.

## **4 DECENTRALIZED CONTROL ALGORITHMS FOR REACTIVE RESOURCES**

With the introduction of remotely controlled power network devices, new possibilities for control strategies are starting to emerge and some are relying on local control but with decentralized algorithms. Thus, agent based technologies and decentralized optimization techniques are studied in order to get local and faster responses for applications that formerly were coordinated by a centralized control.

In the previous chapter an algorithm to control reactive sources was presented. That algorithm, based on the incident command system, divides the power network into different realms, each with the authority to send commands and requests to lower level, but not to higher level, realms. In this chapter a particular kind of problem is studied: how to coordinate reactive resources (capacitors and reactive load) of a section of a distribution feeder locally and with the minimum information from other regions. To perform this type of coordination, decentralized optimization techniques are presented to study what requirements are needed and also to find the challenges to this local coordination.

### **4.1 Local Reactive Problem in a Distribution Feeder**

Before implementing decentralized optimization algorithms, the problem of interest, to which these techniques are going to be applied is presented. The problem is based on the fact that distribution feeder voltage can cause problems for end user devices and equipment. Typically voltage regulation problems can



cause improper or less-efficient equipment operation, for example, lights providing incorrect illumination [1] and tripping sensitive loads such as uninterruptible power supply by reverting to battery storage during high or low voltage situations [1]. Undervoltages can cause induction motors to overheat because they will draw more current, thus increasing their temperature. For this reason the American National Standards Institute (ANSI) standard ANSI C84.1-2006 [2] specifies two voltage ranges to provide a metric of normal steady-state voltages that must be supplied to users to minimize problems to them. The two voltage ranges for a normal three-wire 120/240 V service to a user can be simplified as follows [2]:

- 1) Range A: Electric supply systems shall be so designated and operated that most service voltages will be within the limits specified for Range A. The occurrence of voltages outside of these limits should be infrequent.
  - a. Maximum and Minimum Service Voltage: 126 V (+5%) and 114 V (-5%)
  - b. Maximum and Minimum Utilization Voltage: 125 V (+5%) and 110 V (-8.3%)
- 2) Range B: Voltages above and below Range A. When they occur, corrective measures shall be undertaken within a reasonable time to improve voltages to meet Range A requirements.
  - a. Maximum and Minimum Service Voltage: 127 V (+5.8%) and 110 V (-8.3%)

- b. Maximum and Minimum Utilization Voltage: 127 V (+5.8%) and 106 V (-11.7%)

For other voltage levels the standard provides different nominal voltage ranges for operation. The important thing is that these values will vary depending on the loading of the system. Figure 4.1 presents the effect on voltages in a distribution feeder of loading conditions from the first loaded user to the last user at the end of the feeder. Figure 4.1 also shows the ranges required by the ANSI C84 standard.

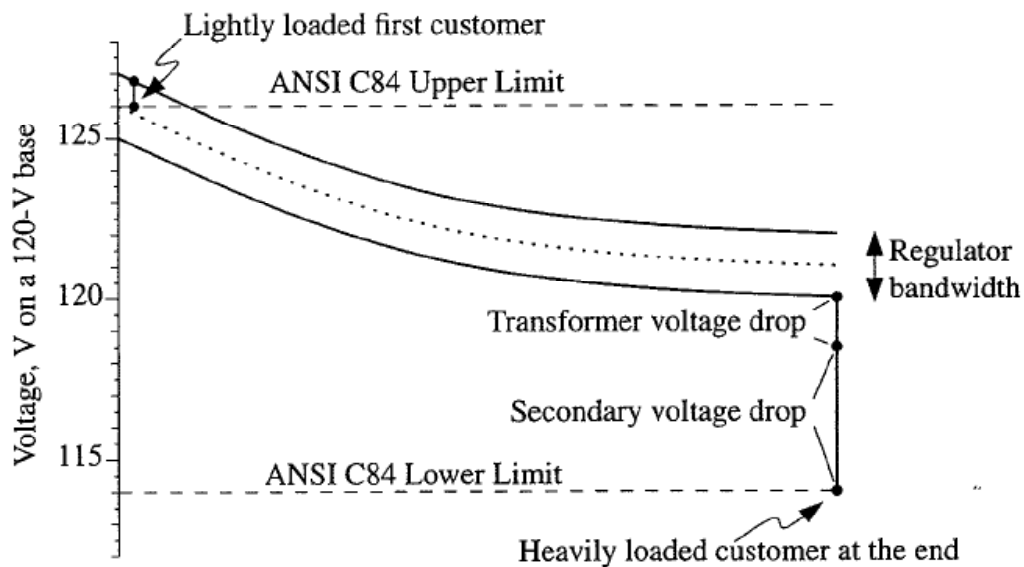


Figure 4.1: Voltage drop along a radial circuit with no capacitors or line regulators [1]

In distribution networks the voltage regulation problem is a constant and recurrent problem. The problem is a direct result of the line and its impedance. From Figure 4.2 the voltage drop across any distribution line can be approximated as follows:

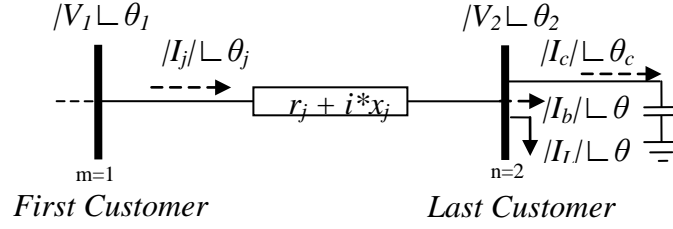


Figure 4.2: Branch diagram in a distribution feeder

$$V_{drop} = |V_2| - |V_1| \approx I_r \cdot r + I_x \cdot x \quad (4.1)$$

where  $V_{drop}$  is the voltage drop along the feeder,  $r$  is the line resistance,  $x$  is the line reactance,  $I_r$  is the line current due to real power flow (in phase with the voltage) and  $I_x$  is the line current due to reactive power flow ( $90^\circ$  out of phase with the voltage)[1, 3, 4]. If the real and reactive line currents are represented in terms of the load power factor ( $pf$ ), we get

$$I_r = I_j \cdot \cos(\theta_L) \quad (4.2)$$

$$I_x = I_j \cdot \sin(\theta_L) \quad (4.3)$$

$$\theta_L = \cos^{-1}(pf) \quad (4.4)$$

This is just an approximation of the voltage drop, but in [5] it was shown that it is accurate for most cases and the biggest errors occur under heavy currents and leading power factors. Most common load flow programs use a complete phasor calculation, but from this approximation many important conclusions about the type of the load can be obtained:

- For resistive loads: High power factor means that the voltage drop would be dependent on the line resistance.

- For Reactive loads: A low power factor of about 0.8 means that the voltage drop would be dependent on the line reactance. Thus poor power factor could increase voltage drop.

Load power factor plays an important role in the voltage profile of a distribution network, as is the case in transmission power networks. Many voltage regulation techniques are available to reduce the voltage drop. Those techniques are:

- Increase power factor (add capacitor or control reactive loads).
- Change line conductor size.
- Balance the circuits and convert single-phase sections to three-phase sections.

Typically to regulate the substation bus voltage the utilities use load tap-changing transformers (LTCs). This regulator is controlling the voltage at the source in the substation. The transformer taps on these LTCs can be changed between a bandwidth of  $\pm 2.5$  and  $\pm 5\%$  of the rated voltage. But because of the voltage drop at longer feeders other controls and regulators are needed. Some of these controls are:

- Extra line regulators: This would help get a better voltage profile by flattening the voltage along the feeder. Typically they are autotransformers with automatically adjusting remotely controlled taps [1, 6].
- Feeder capacitor: This would inject VARs to increase the voltage in the feeders, resulting in a lower setting for the LTCs at the substation [1].

- Line drop compensation: This technique would try to maintain the voltage at the last bus if the feeder was at an acceptable value. Also, it must increase the voltage when heavy load conditions are present, and during light load the boost in voltage should be minimal [4].
- Tighter bandwidth: This means reducing the bandwidth of regulators, thus reducing the voltage drop across the feeders. The problem is that more regulators would be needed, increasing the complexity of the communication infrastructure.

As was introduced in the previous chapter, in this work the reactive load is also added as an alternative to increase poor power factors and low voltage profile. In this chapter many decentralized optimization techniques are going to be studied and analyzed to discover if they could help in solving the voltage regulation and low voltage profile issue in a particular distribution feeder. The idea is to find a suitable algorithm that will allow solving this problem locally without minimum information from other regions of a distribution network.

## **4.2 Decentralized Optimization in a Distribution Feeder**

In recent years, research has focused on studying centralized problems in a decentralized scheme. In particular, the optimal power flow (OPF) can be solved by using decentralized algorithms to parallelize the solution and get faster results.

The main purpose of an optimal power flow (OPF) study is to optimize the static operating conditions of an electric power system. An OPF adjusts the controllable quantities in the system to optimize an objective function, while

satisfying a set of physical and operational constraints. In general, the OPF is formulated as a constrained nonlinear optimization problem as follows:

$$\begin{aligned}
 \min \quad & f(x) \\
 \text{s.t.} \quad & g(x) = \mathbb{C}, \\
 & h(x) \leq 0
 \end{aligned} \tag{4.5}$$

where  $g(x)$  are the power flow constraints  $h(x) \leq 0$  are the inequality constraints and  $x$  is the vector of unknown variables and input parameters, such as the voltages, angles, controllable shunts, capacitors, real and reactive power among others.

In [7] a parallelized OPF suitable for a coarse-grained distributed implementation is presented. The authors implemented the algorithm with three different mathematical decompositions in order to coordinate the distributed OPF. The work provides a simple explanation of the concept of decentralization and decomposition of a system in regions. The authors decompose the power system into two overlapping regions,  $a$  and  $b$ , as shown in Figure 4.3.

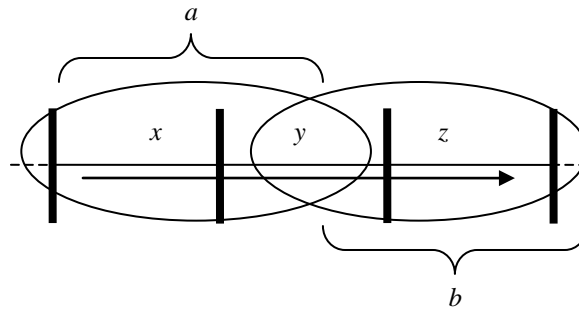


Figure 4.3: Decomposition of a power system in two regions,  $a$  and  $b$

Between the two regions there is an overlap region, with a vector of variables denoted by  $y$ . For example, this overlap region could be a tie line between the regions  $a$  and  $b$ . For each tie line there is a bus in the border of the region, and if the system does not have a bus in the border, a dummy bus is created. The border region bus will have some associated variables which are the voltage and angle, and the real and reactive power flows flowing through the bus. These variables are represented as the  $y$  vector in Figure 4.3. The  $y$  vector is also known as the complicated variables. The vector  $x$  consists of all the OPF variables that are relevant to region  $a$  but not already included in  $y$ . Similarly,  $z$  includes the region  $b$  variables not included in  $y$ . With these assumptions the OPF formulation can be written as follows:

$$\begin{aligned} \min \quad & f_a(x, y) + f_b(y, z) \\ \text{s.t.} \quad & \left. \begin{aligned} g_a(x, y) &= 0 \\ h_a(x, y) &\leq 0 \end{aligned} \right\} A, \quad (4.6) \\ & \left. \begin{aligned} g_b(y, z) &= 0 \\ h_b(y, z) &\leq 0 \end{aligned} \right\} B \end{aligned}$$

where  $g_a(x, y)$  are the equality power flow constraints and  $h_a(x, y) \leq 0$  are the inequality constraints for region  $a$ . Also, the  $f_a(x, y)$  and  $f_b(x, y)$  are the objective functions of the OPF for regions  $a$  and  $b$  respectively. Moreover, these constraints can further be expressed as follows:

$$\begin{aligned}
& \min \quad f_a(x, y) + f_b(y, z) \\
& s.t. \quad \left. \begin{aligned} g_{aa}(x) &= 0 \\ g_{ab}(x, y) &= 0 \\ h_{aa}(x) &\leq 0 \\ h_{ab}(x, y) &\leq 0 \end{aligned} \right\} A \\
& \quad \quad \quad \left. \begin{aligned} g_{bb}(z) &= 0 \\ g_{ba}(y, z) &= 0 \\ h_{bb}(z) &\leq 0 \\ h_{ba}(y, z) &\leq 0 \end{aligned} \right\} B
\end{aligned} \tag{4.7}$$

where  $g_{aa}(x)$  are the equality power flow constraints that do not contain the complicated variables in vector  $y$ , and  $g_{ab}(x, y)$  are also known as the complicated constraints because they involve variables in the border region from other regions. Similarly the same analogy is for the inequality constraint and the constraint in other regions. The authors in [7] used the auxiliary problem principle, which is a linearization of the augmented Lagrangian, resulting in a problem that can be separated into a sequence of subproblems. This formulation will be explained in Section 4.3. Once a problem is formulated as previously presented, it can be decomposed into subproblems by different techniques. Some of these techniques are going to be explained and analyzed in detail later in the chapter.

Once the optimization problem is formulated as in (4.6) the technique to solve it should be suitable to solve a sequence of subproblems. For example, the distributed OPF presented in [8] incorporates discrete control variables. The algorithm follows the same approach as in [7]. There are some boundary variables



among the region's border buses, thus creating a problem that would need to be separated into subproblems; then the problem is solved using a combination of a successive quadratic programming (SQP) method with a pseudo quasi Newton (DQPN) method. As part of the solution, the authors presented an algorithm to exchange data among different subsystems that were created during the decentralization implementation. The details on how this algorithm would work in a real life operation are missing. In [9], the concept of multi-area control was presented again. The authors assumed that areas are determined independently but influenced by flexible AC transmission systems (FACTS) devices. The idea is that one can solve a local optimization algorithm while assuming that the influences of other areas are constant. This time, the solution to the optimization algorithm was based on the technique of approximate Newton direction. The importance of these recent works is that the same decentralized approach as in [7] is presented in this thesis to divide a problem into subproblems that require the exchange of data among the interconnected regions.

This thesis will follow the same approach as the work presented in the previous paragraphs. In Sections 4.3 to 4.6, a description of the decomposition algorithms that are going to be tested to solve the problem presented in Section 4.1 is presented.

### **4.3 Auxiliary Problem Principle Algorithm**

Following the work presented in [7], the first decentralized algorithm to be explained is the auxiliary problem principle (APP). Specifically the case with

explicit coupling constraints [10] and not-strongly convex problems is explained. Since it cannot be guaranteed that the objective function and the entire problem are convex, a suitable algorithm that takes advantage of the augmented Lagrangians is presented. The augmented Lagrangian method penalizes constraint violations by adding a high penalty to the cost function of the problem. As a consequence, the resulting unconstrained problem approximates the original constrained problem. For example, taking the augmented Lagrangian for problem (4.5) by penalizing the equality constraints gives the following:

$$L(x, \lambda) = f(x) + \lambda^T g(x) + c^T \|g(x)\|^2 \quad (4.8)$$

where  $c$  is the penalty parameter for violating the equality constraint  $g(x)$ . By making  $c$  very large, there is high cost associated with the infeasibility of the constraint; thus the unconstrained minimum of  $L(x, \lambda)$  will be close to the minimum in  $f(x)$ . The following example, taken from [12], illustrates the concepts of the augmented Lagrangian. Consider the following problem:

$$\begin{aligned} \min \quad & f(x) = \frac{1}{2}(x_1^2 + x_2^2) \\ \text{s.t.} \quad & x_1 = 1 \end{aligned} \quad (4.9)$$

This problem has an optimal solution  $x^* = (1, 0)$  and corresponding Lagrange multiplier  $\lambda^* = -1$ . The augmented Lagrangian is

$$L(x, \lambda) = \frac{1}{2}(x_1^2 + x_2^2) + \lambda(x_1 - 1) + \frac{c}{2}(x_1 - 1)^2 \quad (4.10)$$

Thus by setting the gradient to zero it can be seen that its minimum is given by

$$x_1(\lambda, c) = \frac{c - \lambda}{c + 1}, \quad x_2(\lambda, c) = 0 \quad (4.11)$$

Thus, as  $\lambda \rightarrow \lambda^* = -1$ , then

$$\lim_{c \rightarrow \infty} x_1(\lambda, c) = 1 = x_1^*, \quad \lim_{c \rightarrow \infty} x_2(\lambda, c) = 0 = x_2^* \quad (4.12)$$

This result can be presented better in Figure 4.4 as  $c \rightarrow \infty$ ,  $L(x, \lambda)$  approaches the constraint minimum  $x^* = (1, 0)$ .

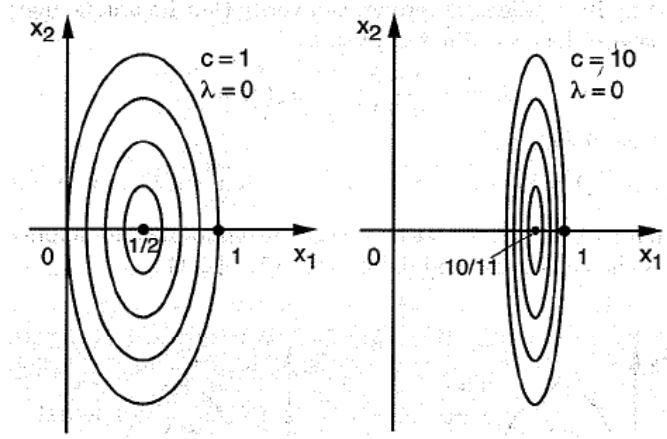


Figure 4.4: Equal cost surfaces of the augmented Lagrangian as  $c \rightarrow \infty$  [12]

The basic idea behind the augmented Lagrangian has been explained; thus, now the details of APP using augmented Lagrangian will be presented.

The augmented Lagrangian is used to decompose the problem. The problem presented in (4.6) and (4.7) has the following formulation as taken from [11] and [13]:

$$\begin{aligned} \min \quad & f_1(u) + f_2(u) \\ \text{s.t.} \quad & \Theta(u) = 0 \end{aligned} \quad (4.13)$$

where  $f_2$  and  $\Theta$  are assumed to be additive functions and  $f_1$  is assumed to be differentiable. In [11] they solve this kind of problem using the following augmented Lagrangian APP formulation:

$$u^{k+1} = \arg \min \left\{ \beta K(u) + f_2(u) + \langle f_1(u) - \beta \nabla K(u^k), u \rangle + \langle \lambda^k + c\Theta(u^k), \Theta(u) \rangle \right\} \quad (4.14)$$

$$\lambda^{k+1} = \lambda^k + \alpha \Theta(u^{k+1}) \quad (4.15)$$

where  $K$  is an auxiliary strongly convex differentiable function. But since in this thesis the cost function  $f$  is considered additive and the equality constraints of interest are additive, then the problem can be decomposed using the comment *iii* presented in Algorithm 14 of [11]. If the equality constraint is assumed to be equal to zero and  $f_2=f, f_1=0$  and  $K(u) = \|u\|^2/2$ , then the following subproblems for  $i = 1, \dots, N$  can be obtained:

$$u_i^{k+1} = \arg \min \left\{ f_i(u_i) + \frac{\beta}{2} \|u_i - u_i^k\|^2 + (\lambda^k)^T \Theta_i(u_i) + c \left( \sum_{j=1}^N \Theta_j(u_j^k)^T \Theta_i(u_i) \right) \right\} \quad (4.16)$$

$$\lambda^{k+1} = \lambda^k + \alpha \Theta(u^{k+1}) \quad (4.17)$$

Now this algorithm can be applied to the problem presented in (4.6) and (4.7). In (4.6) the  $y$  variable is duplicated in both regions. Then, if subproblems of the problem presented in (4.16) and (4.17) are going to be solved, the equality constraint  $\Theta(u) = 0$  becomes

$$(y_a - y_b) = 0 \quad (4.18)$$

where  $y_a$  and  $y_b$  are the complicated variables in regions  $a$  and  $b$ , respectively. Then, using the formulation presented in (4.6), (4.7) and (4.18), (4.16) and (4.17) becomes

$$(x^{k+1}, y_a^{k+1}, y_b^{k+1}, z^{k+1}) = \arg \min_{\substack{(x, y_a) \in A \\ (y_b, z) \in B}} \left\{ \begin{aligned} & f_a(x) + f_b(z) + \\ & \frac{\beta}{2} \|y_a - y_a^k\|^2 + \frac{\beta}{2} \|y_b - y_b^k\|^2 + \\ & c(y_a - y_b)^T (y_a^k - y_b^k) + (\lambda^k)^T (y_a - y_b) \end{aligned} \right\} \quad (4.19)$$

$$\lambda^{k+1} = \lambda^k + \alpha (y_a^{k+1} - y_b^{k+1}) \quad (4.20)$$

where  $\lambda$  is an estimate cost to maintain the constraint  $(y_a - y_b) = 0$  at each iteration.

The convergence of this problem was discussed in [13]. It has to be assumed that the sets  $\{y_a : \exists x \text{ such that } (x, y_a) \in A\}$  and  $\{y_b : \exists z \text{ such that } (x, y_b) \in B\}$  are

closed and convex, and also the following functions has to be convex and differentiable:

$$c_a(y_a) = \min_x \{f_a(x) : (x, y_a) \in A\} \quad (4.21)$$

$$c_b(y_b) = \min_z \{f_b(z) : (z, y_b) \in B\} \quad (4.22)$$

Assuming that  $\alpha < 2c < \beta$  and using Theorem 15 of [11], the problem of (4.19) and (4.20) converges to (4.13). This condition is too strict to be directly applicable to the problem in (4.6) because in general it cannot be proved that the cost of the solution is a convex function of the border or complicated constraints [7]. At least convergence is obtained, but it cannot be proved that the solution is going to be optimal because it cannot be guaranteed that problem (4.6) is strongly convex.

#### 4.3.1 Auxiliary Problem Principle with Distribution Network Equations

The distribution network equations are going to be applied to the APP problem explained in Section 4.3. First, it is important to determine the complicated variables  $y$  that are going to be duplicated between regions. Figure 4.5 provides a representation of a distribution network feeder.

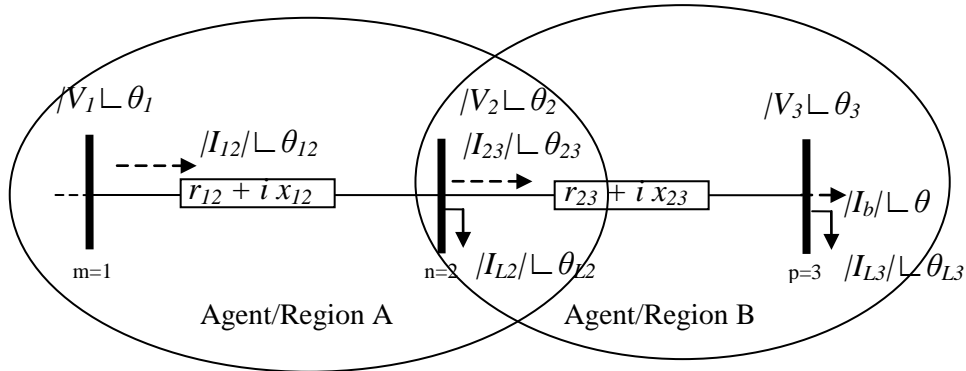


Figure 4.5: Distribution feeder with two agents or regions interconnected

In Figure 4.5 the regions are defined in a way that only a border bus is between each region. This means that the complicated variables are the quantities directly interacting with the bus. If the distribution power flow (DPF) network equations are revised, it will be clear which are the complicated variables for the interconnected regions and corresponding border bus. The complicated variables are the voltage ( $V_2$ ) and angle ( $\theta_2$ ) of the border bus 2 and the current ( $I_{23}$ ) and its angle ( $\theta_{23}$ ) leaving the border bus 2. The equations that involve only variables that are at the border buses are as follows:

$$\text{Re} \left[ \left( V_2 \angle \theta_2 \right) = \left( V_3 \angle \theta_3 \right) + \left( (r + i \cdot x) * \left( I_{23} \angle \theta_{23} \right) \right) \right] \quad (4.23)$$

$$\text{Im} \left[ \left( V_2 \angle \theta_2 \right) = \left( V_3 \angle \theta_3 \right) + \left( (r + i \cdot x) * \left( I_{23} \angle \theta_{23} \right) \right) \right] \quad (4.24)$$

Equations (4.23) and (4.24) may be rewritten to get the voltage at bus 3 as

$$\begin{aligned} \left( V_3 \angle \theta_3 \right) = \\ \left( V_2 \angle \theta_2 \right) - \left( I_{23} \angle \theta_{23} \right) \cdot (r_{23}) + \left( I_{23} \angle \theta_{23} \right) \cdot (x_{23}) \end{aligned} \quad (4.25)$$

$$\begin{aligned} \left( V_3 \angle \theta_3 \right) = \\ \left( V_2 \angle \theta_2 \right) - \left( I_{23} \angle \theta_{23} \right) \cdot (x_{23}) - \left( I_{23} \angle \theta_{23} \right) \cdot (r_{23}) \end{aligned} \quad (4.26)$$

The DPF equality constraints for the line currents 1-2 are

$$\text{Re} \left[ \left| I_{12} \angle \theta_{12} \right| = \left| I_{L2} \angle \theta_{L2} \right| + \left| I_{23} \angle \theta_{23} \right| \right] \quad (4.27)$$

$$\text{Im} \left[ \left| I_{12} \angle \theta_{12} \right| = \left| I_{L2} \angle \theta_{L2} \right| + \left| I_{23} \angle \theta_{23} \right| \right] \quad (4.28)$$

where

$$\begin{aligned}
|I_{L2}| \angle \theta_{L2} &= \frac{|S_{Load2}|}{|V_2|} \angle (\theta_{Load2} - \theta_2) = \\
\left( \frac{S_{Load2}}{V_2} \right)^* &= \left( \frac{P_{Load2} + i \cdot Q_{Load2}}{|V_2| \cos \theta_2 + i \cdot |V_2| \sin \theta_2} \right)^*
\end{aligned} \tag{4.29}$$

Equations (4.27) and (4.28) may be rewritten as

$$|I_{12}| \cdot \cos \theta_{12} = |I_{L2}| \cdot \cos \theta_{L2} + |I_{23}| \cdot \cos \theta_{23} \tag{4.30}$$

$$|I_{12}| \cdot \sin \theta_{12} = |I_{L2}| \cdot \sin \theta_{L2} + |I_{23}| \cdot \sin \theta_{23} \tag{4.31}$$

Note that in the optimization problem presented in (4.7) the  $y$  vector is

$$y = \begin{bmatrix} |V_2| \\ \theta_2 \\ |I_{23}| \\ \theta_{23} \end{bmatrix} \tag{4.32}$$

Now, applying these equations to the problem formulation presented in (4.7) and only presenting the equality constraints, the following is obtained:

$$\begin{aligned}
&\min \quad f_a(x, y) + f_b(y, z) \\
&s.t. \quad \left. \begin{aligned} g_{aa}(x) &= DPF(x) \\ g1_{ab}(x, y) &= |I_{12}| \cdot \cos \theta_{12} - |I_{L2}| \cdot \cos \theta_{L2} - |I_{23}| \cdot \cos \theta_{23} = 0 \\ g2_{ab}(x, y) &= |I_{12}| \cdot \sin \theta_{12} - |I_{L2}| \cdot \sin \theta_{L2} - |I_{23}| \cdot \sin \theta_{23} = 0 \end{aligned} \right\} A \\
&\quad \left. \begin{aligned} g_{bb}(z) &= DPF(z) \\ g1_{ba}(y, z) &= (|V_3| \cdot \cos \theta_3) - (|V_2| \cdot \cos \theta_2) \\ &\quad + (|I_{23}| \cdot \cos \theta_{23}) \cdot (r_{23}) - (|I_{23}| \cdot \sin \theta_{23}) \cdot (x_{23}) = 0 \\ g2_{ba}(y, z) &= (|V_3| \cdot \sin \theta_3) - (|V_2| \cdot \sin \theta_2) \\ &\quad + (|I_{23}| \cdot \cos \theta_{23}) \cdot (x_{23}) + (|I_{23}| \cdot \sin \theta_{23}) \cdot (r_{23}) = 0 \end{aligned} \right\} B
\end{aligned} \tag{4.33}$$

In equation (4.33) it can be seen that the problem can be decomposed easily by the APP formulation presented in Section 4.3. Is important to say that the complicated constraints  $g_{ab}(x,y)$  and  $g_{ba}(y,z)$  are different. For region  $A$  the complicated constraints  $g_{ab}(x,y)$  are the equations for currents  $I_{12}$ . The constraints to compute  $I_{23}$  are not added in the complicated constraints of region  $B$  because this current is computed in region  $B$  without the current  $I_{12}$ , but  $I_{12}$  in region  $A$  needs the complicated constraint of  $I_{23}$  to be calculated. The same happens with the complicated constraints  $g_{ba}(y,z)$  in region  $B$ . The complicated constraints for region  $B$  are the equations for voltage at bus 3. Notice that to get this voltage at bus 3, the voltage at bus 2 is needed and this voltage is one of the complicated variables. The reason that the voltage 3 used the equations (4.25) and (4.26) to get calculated is that at the end of the feeder the only way to compute the voltage at that bus is by using those equations.

With these equation formulations, the problem can be easily formulated as in equations, (4.19) and (4.20). Then the problem can be decomposed into subproblems. The subproblem for region  $A$  can be formulated as follows:

$$(x^{k+1}, y_a^{k+1}) = \arg \min_{(x, y_a) \in A} \left\{ f_a(x) + \frac{\beta}{2} \|y_a - y_a^n\|^2 + c(y_a)^T (y_a^k - y_b^k) + (\lambda^k)^T (y_a) \right\} \quad (4.34)$$

The same formulation can be applied for region  $B$ . It is important to mention that if a region  $X$  is connected to two other regions, then the region  $X$  complicated variables are the same but duplicated, meaning that it will have the same complicated variables for the bus where the current is entering the region and for the bus where the current is leaving the region. This means that the complicated



constraints for region  $X$  will be equations (4.25) and (4.26) for the bus voltage at the first border bus, and (4.30) and (4.31) for the current at the last border bus.

Then the algorithm to perform the decentralized optimization problem using the APP is as follows [7]:

- 1) Initialize all of the variables:  $x^0, y_a^0, y_b^0, \lambda^0$ .
- 2) In parallel, solve the subproblem for a particular region using (4.34).
- 3) Exchange the border bus complicated variables values ( $y_a^k$  and  $y_b^k$ ).
- 4) Update  $\lambda^{k+1}$  in region b with equation (4.20) and exchange the new update to region b.
- 5) Check that convergence of  $y_a^k$  and  $y_b^k$  is less than tolerance  $\tau$ .
- 6) If convergence  $> \tau$  go to step 2.

This kind of algorithm can be implemented with the agents scheme presented in Chapter 3, and the information that they need to communicate is the complicated constraints and the check for convergence between regions. This can be performed by any of the regions and it will stop sending updates to adjacent regions if the convergence is less than a tolerance.

#### 4.4 Predictor-Corrector Proximal Multiplier Method

The predictor-corrector proximal multiplier (PCPM) method is another decomposition algorithm for solving convex programming problems with separable structure. The algorithm is based on the Rockafellar proximal methods of multipliers [14]. This method is a realization of the proximal point algorithm

that converts the problem to a strongly convex one, making the problem have good convergence characteristics [14].

Since the PCPM algorithm is based on the proximal method of multipliers, it is important to give some insight into this method before continuing the discussion. Suppose the following convex problem with separable structure exists [15]:

$$\begin{aligned} \min \quad & f_1(x) + f_2(z) = F(u) \\ \text{s.t.} \quad & Ax = z \end{aligned} \tag{4.35}$$

where  $f$  and  $g$  are closed convex functions and  $A$  is a given  $m$  by  $n$  matrix. The Lagrangian formulation for this problem is

$$L(x, z, \lambda) = f_1(x) + f_2(z) + \lambda^T \cdot [Ax - z] \tag{4.36}$$

Since  $L(x, z, \lambda)$  is a closed convex-concave function, the sub-differential  $S$  on  $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$  is given by

$$S(x, z, \lambda) = \partial_{x,z} L(x, z, \lambda) \times \partial_{\lambda} (-L(x, z, \lambda)) \tag{4.37}$$

Then Rockafellar in [14] implemented the proximal point algorithm and is applied to  $S$  to get the proximal method by generating a sequence of  $x^k$  by letting  $x^{k+1}$  be an approximate solution to the modified version of (4.35) in which  $F(u)$  is replaced by [15]

$$F_0^k(u) = F_o(u) + \frac{1}{2} \beta^k \left( \|u - u^k\|^2 \right) \tag{4.38}$$

Thus applying the previous idea at each iteration of this proximal method of multipliers, given  $\hat{x}, \hat{z}, \hat{\lambda}$ , a minimization of the augmented Lagrangian with respect to  $x$  and  $z$  has to be done [15],

$$L_{\lambda}(x, z, \hat{\lambda}) = L_{\lambda}(x, z, \hat{\lambda}) + \frac{1}{2} \lambda \|Ax - z\| + \frac{1}{2} \beta (\|x - \hat{x}\|^2 + \|z - \hat{z}\|^2) \quad (4.39)$$

to obtain the next estimates of  $(x^+, z^+)$ , and then update the Lagrangian multipliers at each iteration by

$$\lambda^+ = \hat{\lambda} + \frac{1}{\beta} (Ax^+ - z^+) \quad (4.40)$$

Thus the resulting problem point algorithm in (4.38) has the following iterative scheme [15]:

$$u^{k+1} = \arg \min \left\{ F(u) + \frac{1}{2} \beta^k (\|u - u^k\|^2) \right\} \quad (4.41)$$

$$\frac{u^{k+1} - u^k}{(1/\beta^k)} \in \partial F(u^{k+1}) \quad (4.42)$$

where  $\partial F$  is the sub-differential of  $F$ . Thus, by using (4.42) in (4.39) and (4.40), the following iterative scheme is obtained:

$$\frac{x^{k+1} - x^k}{(1/\beta^k)} \in \partial f_1(x^{k+1}) + A^T \left( \lambda^k + \frac{1}{\beta^k} (Ax^{k+1} - z^{k+1}) \right) \quad (4.43)$$

$$\frac{z^{k+1} - z^k}{(1/\beta^k)} \in \partial f_2(z^{k+1}) - \left( \lambda^k + \frac{1}{\beta^k} (Ax^{k+1} - z^{k+1}) \right) \quad (4.44)$$

Thus the problem in the previous iterative scheme is that the iteration values of  $(x^{k+1}, z^{k+1})$  are coupled in the constraints by  $(Ax^{k+1} - z^{k+1})$ , so the algorithm cannot compute  $x^{k+1}$  and  $z^{k+1}$  separately. By performing an explicit computation of the constraints, the iterative scheme (4.43) and (4.44) becomes

$$\frac{x^{k+1} - x^k}{(1/\beta^k)} \in \partial f_1(x^{k+1}) + A^T \left( \lambda^k + \frac{1}{\beta^k} (Ax^k - z^k) \right) \quad (4.45)$$

$$\frac{z^{k+1} - z^k}{(1/\beta^k)} \in \partial f_2(z^{k+1}) - \left( \lambda^k + \frac{1}{\beta^k} (Ax^k - z^k) \right) \quad (4.46)$$

Thus by defining

$$\tilde{\lambda}^{k+1} = \lambda^k + \frac{1}{\beta^k} (Ax^k - z^k) \quad (4.47)$$

and by using (4.41), (4.45) and (4.46) become

$$x^{k+1} = \arg \min \left\{ f_1(x) + \tilde{\lambda}^{k+1} [Ax] + \frac{\beta^k}{2} (\|x - x^k\|^2) \right\} \quad (4.48)$$

$$z^{k+1} = \arg \min \left\{ f_2(z) - \tilde{\lambda}^{k+1} [z] + \frac{\beta^k}{2} (\|z - z^k\|^2) \right\} \quad (4.49)$$

To complete the iteration of the proximal multipliers, they are updated by

$$\lambda^{k+1} = \lambda^k + \frac{1}{\beta^k} (Ax^{k+1} - z^{k+1}) \quad (4.50)$$

This Lagrangian multiplier update rule is nothing else than computing a proximal (maximization) iterate of the dual variable  $\lambda$ :

$$\lambda^{k+1} = \arg \min \left\{ \lambda^{k+1} [Ax^{k+1} - z^{k+1}] - \frac{\beta^k}{2} (\|\lambda - \lambda^k\|^2) \right\} \quad (4.51)$$

Thus it will produce  $\tilde{\lambda}^{k+1}$  by replacing  $Ax^{k+1} - z^{k+1}$  with  $Ax^k - z^k$  in (4.51) [15]. Then this algorithm performs two proximal steps to update the Lagrangian multipliers, one with the predictor  $\tilde{\lambda}^{k+1}$  and a second with the corrector  $\lambda^{k+1}$ . This is the reason it is called the PCPM algorithm.

#### 4.4.1 PCPM with Distribution Network Equations

In this case the problem is formulated as the same problem presented in (4.7). The separable regions of the distribution feeder will follow the same representation as in Figure 4.5. Thus the same assumptions are made with the complicated constraints and variables. For these reasons the resulting decentralized problem formulations will be the same as in (4.33).

The PCPM problem can then easily be formulated and decomposed as in equations (4.48) and (4.49) as follows:

$$(x^{k+1}, y_a^{k+1}) = \arg \min_{(x, y_a) \in A} \left\{ f_a(x) + \frac{\beta^k}{2} \|y_a - y_a^n\|^2 + (\hat{\lambda}^{k+1})^T (y_a) \right\} \quad (4.52)$$

$$(z^{k+1}, y_b^{k+1}) = \arg \min_{(z, y_b) \in B} \left\{ f_b(z) + \frac{\beta^k}{2} \|y_b - y_b^n\|^2 + (\hat{\lambda}^{k+1})^T (y_b) \right\} \quad (4.53)$$

Also, equations (4.47) and (4.50) are modified and the resulting equations are

$$\tilde{\lambda}^{k+1} = \lambda^k + \frac{1}{\beta^k} (y_a^k - y_b^k) \quad (4.54)$$

$$\lambda^{k+1} = \lambda^k + \frac{1}{\beta^k} (y_a^{k+1} - y_b^{k+1}) \quad (4.55)$$

Then the algorithm to perform the decentralized optimization problem using the PCPM is as follows:

- 1) Initialize all of the variables:  $x^0, y_a^0, y_b^0, \lambda^0$ .
- 2) Solve the predictor  $\tilde{\lambda}^{k+1}$  in region  $b$  computation using (4.54) and exchange the new predictor update to region  $a$ .
- 3) In parallel, solve the subproblem for each region using (4.52) and (4.53).
- 4) Exchange the border bus complicated variables values ( $y_a^k$  and  $y_b^k$ ).

- 5) Update corrector  $\lambda^{k+1}$  in region  $b$  with equation (4.55) and exchange the new corrector update to region  $a$ .
- 6) Check that convergence of  $y_a^k$  and  $y_b^k$  is less than tolerance  $\tau$ .
- 7) If convergence  $> \tau$  go to step 2.

This kind of algorithm can be implemented with the agents scheme presented in Chapter 3, and the information that they need to communicate is the complicated constraints and the check for convergence between regions. The  $\beta^k$  is a series of positive scalars that, by Theorem 3.1 in [14], will satisfy

$$\varepsilon \leq \frac{1}{\beta^k} \leq \min \left( \frac{1-\varepsilon}{2}, \frac{1-\varepsilon}{2\|y_a - y_a^k\|}, \frac{1-\varepsilon}{2\|y_b - y_b^k\|} \right) \quad (4.56)$$

for some

$$0 < \varepsilon \leq \min \left( \frac{1}{3}, \frac{1}{2\|y_a - y_a^k\| + 1}, \frac{1}{2\|y_b - y_b^k\| + 1} \right) \quad (4.57)$$

This algorithm can be performed by any of the regions and it will stop sending updates to adjacent regions if the convergence is less than a tolerance.

## 4.5 Lagrangian Relaxation Decomposition Algorithm

The Lagrangian relaxation technique has been used extensively in power systems to solve the unit commitment problem [16-19] and the optimal power flow of multiple interconnected areas [20-21]. In this thesis the same technique is applied to the distribution network as a decentralized optimization algorithm

technique. To understand the concept suppose the same optimization problem as in (4.6) exists but without the inequality constraints:

$$\begin{aligned}
& \min \quad f_a(x, y) + f_b(y, z) \\
& \text{s.t.} \quad \left. \begin{aligned} g_{aa}(x) &= 0 \\ g_{ab}(x, y) &= 0 \end{aligned} \right\} A \\
& \quad \quad \quad \left. \begin{aligned} g_{bb}(z) &= 0 \\ g_{ba}(y, z) &= 0 \end{aligned} \right\} B
\end{aligned} \tag{4.58}$$

In problem (4.58) the constraints  $g_{ab}(x, y)$  and  $g_{ba}(y, z)$  are the complicated constraints in this problem. If they are removed, the resulting problem could be separated into small subproblems. For this case the Lagrangian relaxation techniques is applied. In the Lagrangian relaxation technique, the complicated techniques are eliminated by forming a Lagrangian function in which those constraints are added to the objective function using a multiplier  $\bar{\lambda}$  [12].

Thus the Lagrangian function will take the following form:

$$L(x, y, \bar{\lambda}) = f_a(x, y) + f_b(y, z) + \bar{\lambda}_a^T g_{ab}(x, y) + \bar{\lambda}_b^T g_{ba}(y, z) \tag{4.59}$$

In order to solve this type of problem, the dual function is formulated as

$$\begin{aligned}
q(\bar{\lambda}) &= \min_{x, y, z} L(x, y, \bar{\lambda}) \\
& \text{s.t.} \quad \left. \begin{aligned} g_{aa}(x) &= 0 \end{aligned} \right\} A \\
& \quad \quad \quad \left. \begin{aligned} g_{bb}(z) &= 0 \end{aligned} \right\} B
\end{aligned} \tag{4.60}$$

Then the dual problem is defined as follows:

$$\max \quad q(\bar{\lambda}) \tag{4.61}$$

The key aspect to notice is that for any  $\bar{\lambda}$ , the dual value of  $q(\bar{\lambda})$  provides a lower bound to the optimal primal value of  $f_a^*$  and  $f_b^*$ , and so does the optimal dual value  $q^*$  [12].

The minimization of the Lagrangian function (4.59) is facilitated if the problem has a structure that can be separated using a decomposition. In our case the problem can be separated into subproblems if the border variables  $y$  between the regions are kept fixed ( $\bar{y}$ ) in each of the resulting subproblems [22]. Thus the resulting subproblems, given a value of the multipliers  $\bar{\lambda}$ , will take the following form:

$$\begin{aligned} \min_x \quad & f_a(x, \bar{y}) + \bar{\lambda}_a^T g_{ab}(x, \bar{y}) \\ \text{s.t.} \quad & g_{aa}(x) = 0 \end{aligned} \Bigg\} A \quad (4.62)$$

and

$$\begin{aligned} \min_z \quad & f_b(\bar{y}, z) + \bar{\lambda}_b^T g_{ba}(\bar{y}, z) \\ \text{s.t.} \quad & g_{bb}(z) = 0 \end{aligned} \Bigg\} B \quad (4.63)$$

These subproblems (4.62) and (4.63) can be solved independently at each of the corresponding regions. But before continuing with the development of the algorithm, it is important to mention how the maximization of the dual function  $q(\bar{\lambda})$  is going to be performed. Typically the dual function  $q(\bar{\lambda})$  is not explicitly known, so an iterative procedure is implemented to optimize it.

In order to maximize the function  $q(\bar{\lambda})$  an  $x_\lambda$  is needed that minimizes the Lagrangian in subproblem (4.62). Thus, since  $x_\lambda$  is required to evaluate the dual



function at  $\lambda$ , a subgradient  $g_{ab}(x_\lambda, \bar{y})$  of the complicated constraint is obtained. Since the dual function is also nondifferentiable, a subgradient method can be used to generate a sequence of dual feasible points according to the iteration

$$\lambda^{k+1} = \lambda^k - s^k \nabla q(\lambda) \quad (4.64)$$

But since  $\nabla q(\lambda) = g_{ab}(x_\lambda, \bar{y})$ , (4.64) becomes

$$\lambda^{k+1} = \lambda^k - s^k g_{ab}(x_\lambda, \bar{y}) \quad (4.65)$$

where  $s^k$  is the step size of the multiplier update. In this thesis it will follow the same formulation as in [21]:

$$s^k = \frac{1}{\omega + \zeta \cdot k} \quad (4.66)$$

where  $\omega$  and  $\zeta$  are some positive scalars.

Now all the required equations to perform the Lagrangian relaxation method are described. In the next sections, the required assumption to implement this method in the distribution network as well as the complete algorithm are presented.

#### 4.5.1 Lagrangian Relaxation with Distribution Network Equations

The distribution network equations are going to be applied to the Lagrangian relaxation problem presented in the previous section. First, it is important to determine the complicated variables  $y$  that are going to be fixed during the iterations of the algorithm. Figure 4.6 provides a representation of a distribution network feeder for the Lagrangian relaxation problem. The regions are defined such that each region is clearly separated by a tie line. In this case, buses 2 and 3

are the border buses for regions  $A$  and  $B$  respectively. This means that the complicated variables are the quantities directly relating these two buses. If the distribution power flow (DPF) network equations are revised, it will be clear which are the complicated variables that need to be fixed between the interconnected regions in the complicated constraints of the Lagrangian in (4.59). As in the APP and PCPM problems, the complicated variables are the voltage ( $V_2$ ) and angle ( $\theta_2$ ) of the border bus 2 and the current ( $I_{23}$ ) and its angle ( $\theta_{23}$ ) leaving the border bus 2. The difference is that now the region equations and their complicated variables are different.

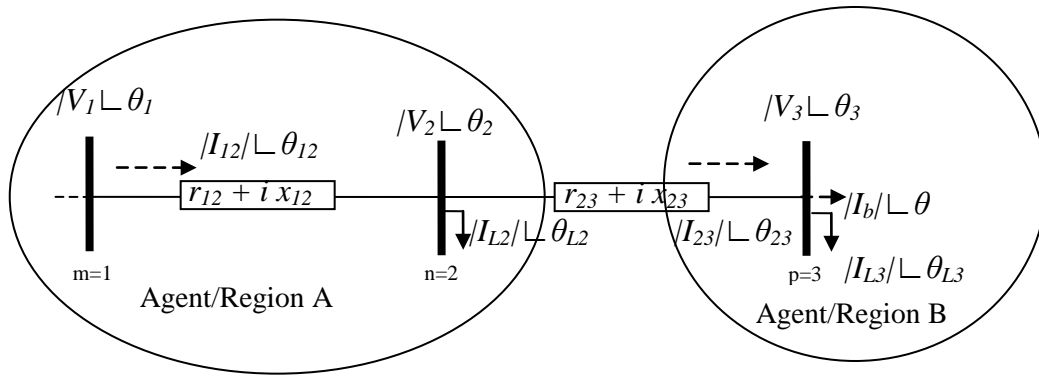


Figure 4.6: Distribution Feeder with two regions interconnected

Before continuing with the derivation of the equations, it is better to formulate the subproblems presented in (4.62) and (4.63) as follows:

$$\begin{aligned} \min_x \quad & f_a(x, \bar{y}_{ab}) + \bar{\lambda}_a^T g_{ab}(x, \bar{y}_{ab}) \\ \text{s.t.} \quad & g_{aa}(x) = 0 \end{aligned} \quad (4.67)$$

and

$$\begin{aligned} \min_z \quad & f_b(\bar{y}_{ba}, z) + \bar{\lambda}_b^T g_{ba}(\bar{y}_{ba}, z) \\ \text{s.t.} \quad & g_m(z) = 0 \end{aligned} \Bigg\} B \quad (4.68)$$

For region  $A$ , the complicated constraints are DPF equality constraints for the line currents 1-2

$$\text{Re} \left[ |I_{12}| \angle \theta_{12} = |I_{L2}| \angle \theta_{L2} + |I_{23}| \angle \theta_{23} \right] \quad (4.69)$$

$$\text{Im} \left[ |I_{12}| \angle \theta_{12} = |I_{L2}| \angle \theta_{L2} + |I_{23}| \angle \theta_{23} \right] \quad (4.70)$$

where

$$\begin{aligned} |I_{L2}| \angle \theta_{L2} &= \frac{|S_{Load2}|}{|V_2|} \angle (\theta_{Load2} - \theta_2) = \\ \left( \frac{S_{Load2}}{V_2} \right)^* &= \left( \frac{P_{Load2} + i \cdot Q_{Load2}}{|V_2| \cos \theta_2 + i \cdot |V_2| \sin \theta_2} \right)^* \end{aligned} \quad (4.71)$$

In equations (4.69) and (4.70) the current  $|I_{23}|$  and the angle  $\theta_{23}$  are going to be fixed values because these quantities are the complicated variables related to region  $B$ . The reason is that these values are computed in the region  $B$ . Then these equations can be rewritten as

$$|I_{12}| \cdot \cos \theta_{12} = |I_{L2}| \cdot \cos \theta_{L2} + |\bar{I}_{23}| \cdot \cos \bar{\theta}_{23} \quad (4.72)$$

$$|I_{12}| \cdot \sin \theta_{12} = |I_{L2}| \cdot \sin \theta_{L2} + |\bar{I}_{23}| \cdot \sin \bar{\theta}_{23} \quad (4.73)$$

Equations (4.72) and (4.73) will correspond to the  $g_{ab}(x, \bar{y}_{ab})$  equations in (4.67), where  $\bar{y}_{ab}$  correspond to the fixed values of variables that are dependent on interconnected region  $B$ .

For region  $B$ , the complicated constraints are DPF equality constraints that involve the voltage and angle at bus 2:

$$\text{Re} \left[ \left( |V_2| \angle \theta_2 \right) = \left( |V_3| \angle \theta_3 \right) + \left( (r + i * x) * \left( |I_{23}| \angle \theta_{23} \right) \right) \right] \quad (4.74)$$

$$\text{Im} \left[ \left( |V_2| \angle \theta_2 \right) = \left( |V_3| \angle \theta_3 \right) + \left( (r + i * x) * \left( |I_{23}| \angle \theta_{23} \right) \right) \right] \quad (4.75)$$

In equations (4.74) and (4.75), the voltage  $|V_2|$  and the angle  $\theta_2$  are the complicated constraints that are going to be fixed in the subproblem of region  $B$ . The bus voltage  $|V_2|$  and the angle  $\theta_2$  are computed in the region A subproblem. Thus these equations can be rewritten as

$$\begin{aligned} \left( |V_3| \cdot \cos \theta_3 \right) = \\ \left( |\bar{V}_2| \cdot \cos \bar{\theta}_2 \right) - \left( |I_{23}| \cdot \cos \theta_{23} \right) \cdot (r_{23}) + \left( |I_{23}| \cdot \sin \theta_{23} \right) \cdot (x_{23}) \end{aligned} \quad (4.76)$$

$$\begin{aligned} \left( |V_3| \cdot \sin \theta_3 \right) = \\ \left( |\bar{V}_2| \cdot \sin \bar{\theta}_2 \right) - \left( |I_{23}| \cdot \cos \theta_{23} \right) \cdot (x_{23}) - \left( |I_{23}| \cdot \sin \theta_{23} \right) \cdot (r_{23}) \end{aligned} \quad (4.77)$$

Equations (4.76) and (4.77) will correspond to the  $g_{ba}(x, \bar{y}_{ba})$  equations in (4.68), where  $\bar{y}_{ba}$  correspond to the fixed values of variables that are dependent on interconnected region A.

Note that in the optimization problem presented in (4.67) and (4.68),  $\bar{y}_{at}$  and  $\bar{y}_{ba}$  are

$$\bar{y}_{ab} = \begin{bmatrix} |\bar{I}_{23}| \\ \bar{\theta}_{23} \end{bmatrix} \quad (4.78)$$

$$\bar{y}_{ba} = \begin{bmatrix} |\bar{V}_2| \\ \bar{\theta}_2 \end{bmatrix} \quad (4.79)$$

Then  $\bar{y}_{at} \in z$  and is calculated in the subproblem (4.68), and  $\bar{y}_{bc} \in x$  and is calculated in the subproblem (4.67).

Now the algorithm to perform the decentralized optimization problem using the Lagrangian relaxation method can be presented as follows:

- 1) Initialize all of the variables:  $x^0$ ,  $\bar{y}_{at}^0$ ,  $\bar{y}_{bc}^0$ ,  $\bar{\lambda}_a^c$  and  $\bar{\lambda}_b^c$ .
- 2) In parallel, solve the subproblem for a particular region using (4.67) and (4.68).
- 3) Exchange the border bus complicated variables values ( $\bar{y}_{at}$  and  $\bar{y}_{bc}$ ).
- 4) Update the multipliers  $\bar{\lambda}$  in regions  $a$  and  $b$  by solving the subgradient iterative scheme of equations (4.65) and (4.66).
- 5) Check that convergence, if  $\bar{y}_{at}$  and  $\bar{y}_{bc}$  change in two consecutive iterations, is less than tolerance  $\tau$  [21].
- 6) If convergence  $> \tau$ , go to step 2.

Like the previous decentralized algorithms, this kind of algorithm can be implemented with the agent's scheme presented in Chapter 3, and the information that they need to communicate is the complicated constraints and the check for convergence between regions. This can be performed by any of the regions and it will stop sending updates to adjacent regions if the convergence is less than a tolerance.

## 4.6 Lagrangian Relaxation Based Decomposition Algorithm

In Section 4.5, a Lagrangian relaxation method was presented. That method as well as the APP and PCPM require the computation of the multipliers by an auxiliary method. Also, the problem is decomposed in subproblems that need to be solved independently, followed by interchange of information about variables that are shared between adjacent regions. In [22] and [23] a new decomposition method was proposed in which only a single iteration at each subproblem is required and no auxiliary method is required to update the Lagrangian multipliers of complicated constraints.

Suppose the problem presented in (4.58) is reformulated as

$$\begin{aligned}
 \min \quad & f_a(x_a) + f_b(z_b) \\
 \text{s.t.} \quad & \left. \begin{aligned} g_{aa}(x_a) &= 0 \\ g_{ab}(x_a, z_{ab}) &= 0 \end{aligned} \right\} A \\
 & \left. \begin{aligned} g_{bb}(z_b) &= 0 \\ g_{ba}(x_{ba}, z_b) &= 0 \end{aligned} \right\} B
 \end{aligned} \tag{4.80}$$

where  $x_a$  correspond to variables in area  $a$  only, and  $z_b$  correspond to variables in area  $b$  only. Also  $z_{ab}$  correspond to variables that are computed in region  $b$ , but  $g_{ab}$  is a function that is dependent on it. The same relationship holds for  $x_{ba}$ , so  $z_{ab} \in z$  and  $x_{ba} \in x$ . Instead of solving this problem with the Lagrangian relaxation technique presented the previous section, the complicated constraints of adjacent regions are moved into the other region subproblems along with its corresponding multipliers  $\bar{\lambda}$ . Then the problem will take the following formulation:

$$\begin{aligned}
& \min_x \quad f_a(x_a) + \bar{\lambda}_b^T g_{ba}(x_{ba}, \bar{z}_b) \\
& \text{s.t.} \quad \left. \begin{aligned} g_{aa}(x_a) &= 0 \\ g_{ab}(x_a, \bar{z}_{ab}) &= 0 \end{aligned} \right\} A
\end{aligned} \tag{4.81}$$

and

$$\begin{aligned}
& \min_z \quad f_b(z_b) + \bar{\lambda}_a^T g_{ab}(\bar{x}_a, z_{ab}) \\
& \text{s.t.} \quad \left. \begin{aligned} g_{bb}(z_b) &= 0 \\ g_{ba}(\bar{x}_{ba}, z_b) &= 0 \end{aligned} \right\} B
\end{aligned} \tag{4.82}$$

Note that in (4.81) the multipliers  $\bar{z}_b$  and  $\bar{\lambda}_b$  are kept fixed but are computed in region  $b$ , and for (4.82)  $\bar{x}_a$  and  $\bar{\lambda}_a$  are also kept fixed in (4.82) but are computed in region  $a$ .

Before continuing with the analysis of this method, the decomposition algorithm is going to be presented [23]:

- 1) Initialize all of the variables:  $x_a^c, z_b^c, x_{bc}^0, z_{at}^0, \bar{\lambda}_a^c$  and  $\bar{\lambda}_b^c$ .
- 2) In parallel, solve one iteration of each subproblem for a particular region using (4.81) and (4.82).
- 3) Obtain  $\lambda_a, \bar{x}_{bc}$  and  $\bar{x}_a$  in the region  $a$  subproblem and exchange it to region  $b$ . Obtain  $\lambda_b, \bar{z}_{at}$  and  $\bar{z}_b$  in the region  $b$  subproblem and exchange it to region  $a$ .
- 4) Check that convergence, if  $\bar{x}_{bc}, \bar{x}_a, \bar{z}_{at}$  and  $\bar{z}_b$  change in two consecutive iterations, is less than tolerance  $\tau$  [21].
- 5) If convergence  $> \tau$ , go to step 2.

To discuss the convergence of this algorithm, let us find the search direction  $(\Delta_a^N, \Delta_b^N)$  for areas  $a$  and  $b$  for a centralized approach to the problem. By using the Karush-Kuhn Tucker (KKT) conditions, the search direction can be computed by solving at each iteration a system of linear equations of the form [23]

$$KKT = \begin{bmatrix} KKT_a & KKT_{ba} \\ KKT_{ab} & KKT_b \end{bmatrix} \begin{bmatrix} \Delta_a^N \\ \Delta_b^N \end{bmatrix} = - \begin{bmatrix} \nabla_{x_a, \lambda_a} L \\ \nabla_{x_b, \lambda_b} L \end{bmatrix} \quad (4.83)$$

where  $L$  is the Lagrangian function of the problem in (4.80) and the superscripts  $N$  indicate the Newton directions. Also  $KKT_a$ ,  $KKT_b$ , and  $KKT_{ab}$  are the Newton matrices for regions  $a$  and  $b$ :

$$KKT_a = \begin{bmatrix} \nabla_{x_a, x_a}^2 L & \nabla_{x_a} h_a^T \\ \nabla_{x_a} h_a & 0 \end{bmatrix} \quad (4.84)$$

$$KKT_b = \begin{bmatrix} \nabla_{x_b, x_b}^2 L & \nabla_{x_b} h_b^T \\ \nabla_{x_b} h_b & 0 \end{bmatrix} \quad (4.85)$$

$$KKT_{ab} = \begin{bmatrix} \nabla_{x_a, x_a}^2 L & \nabla_{x_b} h_a^T \\ \nabla_{x_a} h_b & 0 \end{bmatrix} \quad (4.86)$$

Note that  $KKT_{ab} = KKT_{ba}$ . Then the corresponding moving directions for areas  $a$  and  $b$ ,  $(\Delta_a, \Delta_b)$ , computed in step 2 of the algorithm when the local optimization of the subproblem is solved, can be obtained by solving the decomposable and approximate linear system of equations:



$$\overline{KKT} = \begin{bmatrix} KKT_a & 0 \\ 0 & KKT_b \end{bmatrix} \begin{bmatrix} \Delta_a \\ \Delta_b \end{bmatrix} = - \begin{bmatrix} \nabla_{x_a, \lambda_a} L \\ \nabla_{x_b, \lambda_b} L \end{bmatrix} \quad (4.87)$$

Thus solving these equations at each region to get the solution to the local optimization problem, the sufficient condition of convergence of this decomposition algorithm is given below [23]. At the optimal solution of (4.80), it holds that

$$\rho \left( I - \overline{KKT}^{-1} KKT \right) < 1 \quad (4.88)$$

Then the proposed algorithm converges locally to the solution at a linear rate. In (4.88),  $I$  is the identity matrix and  $\rho(A)$  is the spectral radius of a matrix  $A$ . Also it is assumed that the problem in (4.80) has functions twice differentiable. It is important to mention that by using Newton's method, the local rate of convergence for a centralized approach can be quadratic [23].

Now all the required equations to perform this decomposition algorithm based on the Lagrangian relaxation method are described. In the next section, the required assumption to implement this method in the distribution network as well as the complete algorithm are presented.

#### 4.6.1 Lagrangian Relaxation Based Decomposition Algorithm with Distribution Network Equations

The distribution network equations are going to be applied to the problem presented in the previous section. First, it is important to determine the complicated variables ( $\overline{x}_{bc}$ ,  $\overline{x}_c$ ,  $\overline{z}_{at}$  and  $\overline{z}_l$ ) and multipliers ( $\overline{\lambda}_c$  and  $\overline{\lambda}_l$ ) that are

going to be fixed during the iterations of the algorithm. For this case, Figure 4.6 provides a representation of a distribution network feeder for the Lagrangian relaxation problem.

For region  $a$  the complicated constraints are the currents equations:

$$|I_{12}| \cdot \cos \theta_{12} = |I_{L2}| \cdot \cos \theta_{L2} + |I_{23}| \cdot \cos \theta_{23} \quad (4.89)$$

$$|I_{12}| \cdot \sin \theta_{12} = |I_{L2}| \cdot \sin \theta_{L2} + |I_{23}| \cdot \sin \theta_{23} \quad (4.90)$$

For region  $b$  the complicated constraints are the voltage equations:

$$\begin{aligned} (|V_3| \cdot \cos \theta_3) = \\ (|V_2| \cdot \cos \theta_2) - (|I_{23}| \cdot \cos \theta_{23}) \cdot (r_{23}) + (|I_{23}| \cdot \sin \theta_{23}) \cdot (x_{23}) \end{aligned} \quad (4.91)$$

$$\begin{aligned} (|V_3| \cdot \sin \theta_3) = \\ (|V_2| \cdot \sin \theta_2) - (|I_{23}| \cdot \cos \theta_{23}) \cdot (x_{23}) - (|I_{23}| \cdot \sin \theta_{23}) \cdot (r_{23}) \end{aligned} \quad (4.92)$$

Note that in subproblem (4.81),  $g_{ab}$  equations are (4.89) and (4.90) and in subproblem (4.82),  $g_{ba}$  equations are (4.91) and (4.92).

Then the complicated variables for region  $a$  subproblem (4.81) are

$$\bar{z}_b = \begin{bmatrix} |V_3| \\ \theta_3 \\ |I_{23}| \\ \theta_{23} \end{bmatrix} \quad (4.93)$$

in complicated constraints (4.91) and (4.92). The complicated variables in equations (4.89) and (4.90) are

$$\bar{z}_{ab} = \begin{bmatrix} |I_{23}| \\ \theta_{23} \end{bmatrix} \quad (4.94)$$

The complicated variables for the region  $b$  subproblem (4.82) are

$$\bar{x}_a = \begin{bmatrix} |I_{23}| \\ \theta_{23} \\ |V_2| \\ \theta_2 \end{bmatrix} \quad (4.95)$$

in complicated constraints (4.88) and (4.89). The complicated variables in equations (4.90) and (4.91) are

$$\bar{x}_{ab} = \begin{bmatrix} |V_2| \\ \theta_2 \end{bmatrix} \quad (4.96)$$

The last important components that need to be exchanged and kept fixed at each iteration of the algorithm are the Lagrangian multipliers associated with the complicated constraint of adjacent regions. In this algorithm these multipliers are not updated by an auxiliary method because, instead, the complicated constraints are also exchanged into the other regions along with the Lagrangian associated with it. For example the Lagrangian functions of subproblem (4.81) and (4.82) will be

$$L(x_a, x_{ba}, \lambda_{aa}, \lambda_a) = f_a(x_a) + \bar{\lambda}_b^T g_{ba}(x_{ab}, \bar{z}_b) + \lambda_a^T g_{ab}(x_a, \bar{z}_b) + \lambda_{aa}^T g_{aa}(x_a) \quad (4.97)$$

$$L(z_b, z_{ab}, \lambda_{bb}, \lambda_b) = f_b(z_b) + \bar{\lambda}_a^T g_{ab}(\bar{x}_a, z_{ab}) + \lambda_b^T g_{ba}(\bar{x}_{ba}, z_b) + \lambda_{bb}^T g_{bb}(z_b) \quad (4.98)$$

Notice that at each iteration, each of these Lagrangian functions are going to be solved, so the multipliers  $\bar{\lambda}_c$  and  $\bar{\lambda}_c$  are going to be calculated. Then in step 3 presented in Section 4.6, the multiplier  $\bar{\lambda}_c$  is sent to region  $b$  and  $\bar{\lambda}_c$  is sent to region  $a$ .

Like the previous decentralized algorithms, this kind of algorithm can be implemented with the agents scheme presented in Chapter 3, and the information

that they need to communicate is the complicated constraints and the check for convergence between regions.

## **4.7 Thirty-Four and Sixty-Nine Distribution Feeder Simulations**

The decentralized optimization algorithms presented in Sections 4.3-4.6 were tested on a 34-bus feeder and a 69-bus feeder. In both cases, the loads were modeled as constant PQ devices.

### **4.7.1 Thirty-Four Distribution Feeder Simulation and Results**

In the first scenario the algorithm was tested on a modified IEEE 34-bus distribution system presented in Figure 4.7. For this case, bus 25 is connected to bus 24. In Section 3.2.2 the 34-bus system has bus 25 connected to bus 23. The data, line impedances and loads for this system are presented in Appendix B. This modification was implemented to simplify the derivation of the equations and regions in the decentralized algorithms. The loads to be controlled are located at buses 17, 20, 22, 23, 25, 27, 29 and 30 respectively. In this case the voltages were initially below 0.9 p.u. from bus 17 to bus 34. Since most of the loads are located from bus 17 and higher, the system is divided in two regions. For the auxiliary problem principle (APP) and the predictor-corrector proximal multiplier algorithm (PCPM), the two different regions are as shown in Figure 4.7 One region is controlled by agent 1 and control load buses 1 to 24. The second region is controlled by agent 2 and control load buses 24 to 34.

For the Lagrangian relaxation (LR) algorithm presented in Section 4.5 and the Lagrangian relaxation based decomposition (LRBD) algorithm presented in

Section 4.6, the region decompositions are shown in Figure 4.8. One region is controlled by agent 1 and control load buses 1 to 24. The second region is controlled by agent 2 and control load buses 25 to 34.

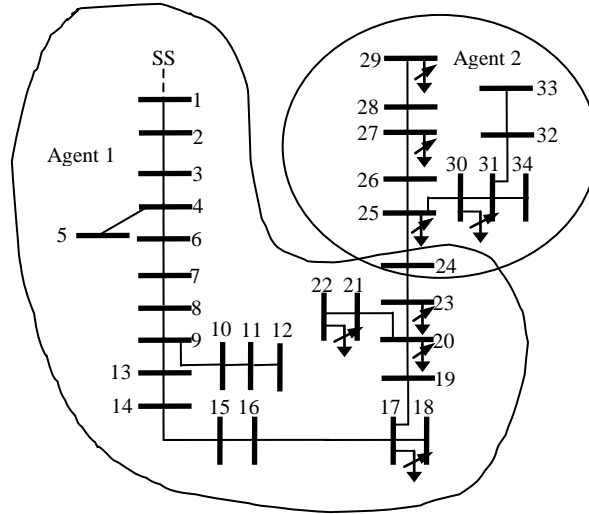


Figure 4.7: IEEE modified 34-bus feeder connected to substation SS for APP and PCPM algorithm

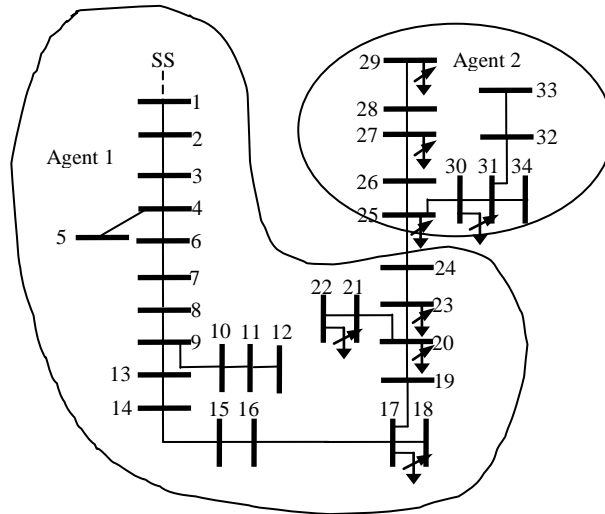


Figure 4.8: IEEE modified 34-bus feeder connected to substation SS for LR and LRBD algorithm

The results from the four algorithms are presented in Tables 4.1 and 4.2, which show that using the algorithm can effectively find the amount of reactive load to be controlled. In Table 4.1 the result on the final controllable loads is presented. There it can be seen that the total of controllable reactive load in the central case was 412.28 kVARs. The two best algorithms at estimating the controllable load were the Lagrangian relaxation (LR) and the Lagrangian relaxation based decomposition (LRBD) algorithm. The amount of controllable load was 403.8 kVARs and 410 kVARs respectively. These results are close to the centralized controllable load. The auxiliary principle problem and the predictor-corrector proximal multiplier (PCPM) estimate 242.17 kVARs and 357 kVARs respectively. Also, in estimating the voltages, the LR and LRBD methods estimate a better voltage profile than the APP and PCPM methods. The APP seems to underestimate the voltages in region 1 but not in region 2.

Table 4.1: Reactive load kVARs for 34-bus system

Case	Control Load (kVARs)	Initial Load (kVARs)	Central Opt.	APP	PCPM	LR	LRBD
Bus 17	7	2	-5	-5.874	-5.0021	2.0003	2
Bus 20	13	3	1.7186	-11.392	-10.002	3.0004	-10.002
Bus 22	95	75	75	-21.405	-20.008	44.253	50.936
Bus 23	31	1	-30	-31.401	1.0097	1.0033	1
Bus 25	27	7	-20	-20.005	-20.004	-20.018	-20
Bus 27	205	105	-100	105	-100	-100.02	-100
Bus 29	46	16	-30	-30.003	16.002	-30.022	-30
Bus 30	95	55	-40	36.909	45.009	-40.021	-40
Total Controlled:			412.2814	242.171	356.9954	403.824	410.066

Table 4.2: Bus voltage for 34-bus system

Case	Initial Volts (pu)	Central Opt.	Final Volts (pu) 0.90 pu APP	Final Volts (pu) 0.9 pu PCPM	Final Volts (pu) 0.9 pu LR	Final Volts (pu) 0.9 pu LRBD
Bus 19	0.85946	0.90149	0.88646	0.89415	0.90047	0.90113
Bus 20	0.85945	0.90149	0.88646	0.89415	0.90047	0.90113
Bus 21	0.85945	0.90148	0.88646	0.89414	0.90047	0.90112
Bus 22	0.85829	0.90038	0.88577	0.89346	0.8995	0.90013
Bus 23	0.85783	0.90078	0.88517	0.89291	0.89968	0.90035
Bus 24	0.85783	0.90053	0.88472	0.8925	0.89943	0.90035
Bus 25	0.85592	0.89962	0.89595	0.8977	0.89659	0.89946
Bus 26	0.85588	0.89961	0.89592	0.89768	0.89658	0.89945
Bus 27	0.85567	0.89956	0.89575	0.8976	0.89652	0.8994
Bus 28	0.85554	0.89951	0.8957	0.89748	0.89647	0.89934
Bus 29	0.85554	0.89951	0.8957	0.89747	0.89647	0.89934
Bus 30	0.8556	0.8994	0.89566	0.8974	0.89636	0.89923
Bus 31	0.85539	0.8992	0.89546	0.89719	0.89616	0.89903
Bus 32	0.85537	0.89918	0.89545	0.89718	0.89615	0.89902
Bus 33	0.85527	0.89909	0.89535	0.89709	0.89605	0.89892
Bus 34	0.85537	0.89918	0.89544	0.89718	0.89614	0.89902

In terms of efficiency, the LR method only requires three iterations of the entire method. These results can be seen in Table 4.3. The LRBD iterates 145 times and takes 9.7 seconds, which gives a solution very close to the centralized case of 150 iterations and 10.78 seconds. It is not surprising to have similar results, because the decomposition is based on performing single iterations of each region. This algorithm will be impractical to implement in a real system

because it will require too much communication between regions in order to find a solution. The second best algorithm was the LR method. It iterates three times among the regions before obtaining a solution. It takes 166.87 seconds in total to find a solution. It is important to mention that each region performs a local optimization problem that iterates, but only needs information from other regions three times. The other two algorithms (APP and PCPM) iterate more than three times and take longer to find a solution.

Table 4.3 Iterations and CPU time comparison for 34-bus feeder

Case	Iterations	Seconds
Central	150	10.78
APP	6	232
PCPM	8	195.95
LR	3	166.87
LRBD	145	9.7501

In overall the Lagrangian relaxation algorithms seems to perform better than the augmented Lagrangian based algorithms. The Lagrangian relaxation algorithms fixed the border variables in order to find a solution. The augmented Lagrangian ones simply duplicate the variables, and only in the optimum where these variables are supposed to be equal is the solution found. Also, because of the topology of the distribution network and the way a distribution power flow is performed, the Lagrangian relaxation algorithms obtain better solutions. In order to perform a power flow, the voltage at the top of the feeder needs to be fixed and the rest of the analysis is performed based on the fact that this voltage was fixed.



Thus the LR methods fixed the voltage at the top of the region, thus basically doing the same procedure as the distribution power flow, while the augmented Lagrangian methods simply estimate this voltages at the top of the region. This makes the augmented Lagrangian algorithms iterate more to find a solution that is the same among regions.

#### **4.7.2 Sixty-Nine Distribution Feeder Simulation and Results**

In the first scenario the algorithm was tested on a modified 69-bus distribution system [24] presented in Figures 4.9 and 4.10. In [24] there were two bus 3's and a line with no impedance between them, so it was eliminated and put together in the same node. Also some of the loads were modified to have low voltage conditions in many parts of the system. The data, line impedances and loads for this system are presented in Appendix B.

The loads to be controlled are located at buses 19, 21, 22, 43, 46, 47, 52, 58, 63, 66 and 69 respectively. In this case, the voltages were initially below 0.94 p.u. from bus 18 to 22, from bus 40 to 47 and from bus 61 to 69. In this system the distribution feeder was divided in three different regions. For the auxiliary problem principle (APP), the three different regions are as shown in Figure 4.9. One region is controlled by agent 1 and control load buses 1 to 28. The second region is controlled by agent 2 and control load buses 28 to 48. The third region is controlled by agent 3 and control load buses 48 to 69.

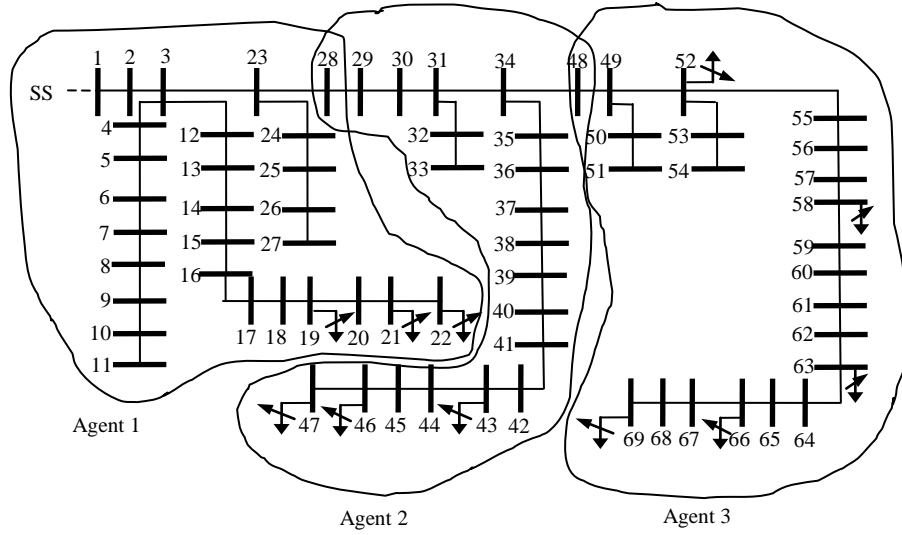


Figure 4.9: 69-bus feeder connected to substation SS for APP algorithm

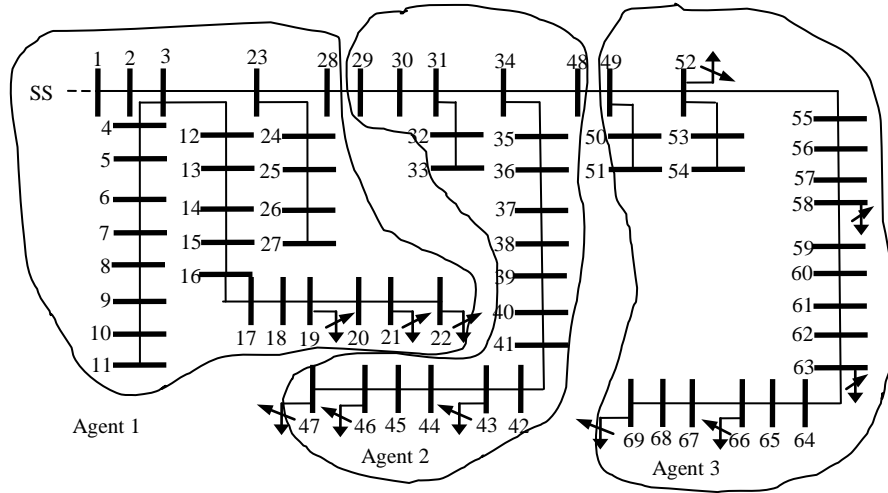


Figure 4.10: 69-bus feeder connected to substation SS for LR and LRBD algorithm

For the Lagrangian relaxation (LR) algorithm presented in Section 4.5 and the Lagrangian relaxation based decomposition (LRBD) algorithm presented in Section 4.6, the region decompositions are shown in Figure 4.10. One region is controlled by agent 1 and control load buses 1 to 28. The second region is

controlled by agent 2 and control load buses 29 to 48. The third region is controlled by agent 3 and control load buses 49 to 69.

The results from the four algorithms are presented in Tables 4.4, 4.5 and 4.6, which show that using the algorithm can effectively find the amount of reactive load to be controlled. For these results the PCPM algorithm was not tested in the 69-bus feeder because from the previous case it was shown that it gives the less efficient result in terms of time and iteration.

From the results it can be seen that the same pattern as the 34-bus feeder is found. The LR and LRBD algorithm provide the best results in terms of controllable reactive load. The centralized case estimates 3270.75 kVARs of controllable load. The LR and LRBD estimate 3938.33 kVARs and 3277.46 kVARs. Again the LRBD gives the best result for controllable loads, but iterates 490 times (Table 4.6). For practical applications the LR will provide the best results, although it iterates more than the APP case, but provides better results for the controllable reactive loads and voltage estimation.

The LR and LRBD estimate the voltage very similar to the centralized case. The APP algorithm underestimates the voltages, but this is not surprising because it also estimates the controllable loads differently. Even though the controllable load is almost the same to that found by the LR algorithm, the individual controllable loads are different from the centralized case, in particular, the bus 43 controllable load. In the APP algorithm the amount of controllable load was 1765.96 kVARs and with the LR it was 1315.68 kVARs. The LR amount was closer to the 1100.96 kVARs amount of the centralized optimization. That is the

reason the APP almost controls the same amount of reactive load, because most of the reactive load comes from bus 43 load and not from the other reactive loads in bus 66 and 69 in which the amount estimated by the APP is less than the other two algorithms.

Table 4.4: Reactive load kVARs for 34-bus system

Case	Control Load (kVARs)	Initial Load (kVARs)	Central Opt.	APP	LR	LRBD
Bus 18	100	114.3	114.3	114.3	114.3	114.31
Bus 21	700	1126.3	520.04	537.45	519.65	608.45
Bus 22	700	1126.3	426.3	426.01	426.3	426.27
Bus 43	1776	888	-212.96	-877.96	-427.68	-156.81
Bus 46	324	162	-162	-162	-162.27	-162.16
Bus 47	84	42	-42.006	-42	-42.228	-42.272
Bus 52	208	104	104	-8.1087	177.06	-104.55
Bus 58	60	30	30.001	27.173	-92.166	30.259
Bus 63	362	181	181	48.879	-204.63	181.24
Bus 66	240	120	-115.53	13.262	-120.73	-56.927
Bus69	220	110	-110	-44	-122.04	-111.37
Total Controlled:			3270.755	3970.895	3938.334	3277.46

Table 4.5: Bus voltage for 34-bus system

Case	Initial Volts (pu)	Central Opt. 0.95 pu	Final Volts (pu) 0.95 pu APP	Final Volts (pu) 0.95 pu LR	Final Volts (pu) 0.95 pu LRBD
Bus 18	0.93866	0.95257	0.952397	0.95258	0.95165
Bus 19	0.93694	0.95129	0.951112	0.95131	0.95034
Bus 20	0.93657	0.95102	0.95084	0.95103	0.95006
Bus 21	0.93211	0.94781	0.947615	0.94783	0.94677
Bus 22	0.93209	0.9478	0.947599	0.94781	0.94676
Bus 40	0.93658	0.95684	0.963397	0.95651	0.95674
Bus 41	0.93314	0.95448	0.961511	0.95429	0.95434
Bus 42	0.92916	0.95177	0.959366	0.95174	0.95159
Bus 43	0.92319	0.94852	0.957288	0.94885	0.94824
Bus 44	0.9229	0.94837	0.957144	0.9487	0.94809
Bus 45	0.92251	0.94818	0.956971	0.94851	0.94791
Bus 46	0.92063	0.94733	0.956125	0.94766	0.94705
Bus 47	0.92006	0.94707	0.955869	0.94741	0.9468
Bus 61	0.93902	0.9519	0.947389	0.95023	0.95205
Bus 62	0.93815	0.95125	0.946674	0.94977	0.95137
Bus 63	0.93675	0.95021	0.945509	0.94902	0.95029
Bus 64	0.93671	0.95019	0.945466	0.949	0.95027
Bus 65	0.93636	0.95	0.945137	0.94882	0.95005
Bus 66	0.93558	0.94957	0.944467	0.94841	0.94959
Bus 67	0.93475	0.94911	0.943767	0.94796	0.94912
Bus 68	0.9344	0.94891	0.943434	0.94777	0.94893
Bus 69	0.93423	0.94882	0.943311	0.94769	0.94884

In terms of efficiency, the same pattern as in the 34-bus case feeder was found. The LR and LRBD give better results and behave better, taking 644.95 and 108.34 seconds. The LR locally in each of the three regions iterates to find a solution, but only needs information from other regions eight times while the LRBD iterates 490 times but takes less time. Also, it is important to notice than in the 34-bus case the LR only iterates three times but here iterates eight. This is due to the fact that in the 69-bus feeder there are three regions. Thus the decentralized algorithm is bigger and requires more iterations to find solutions among the regions.

Table 4.6: Iterations and CPU time comparison for 69-bus feeder

Case	Iterations	Seconds
Central	161	63.13
APP	6	819.38
LR	8	644.95
LRBD	490	108.34

## 4.8 Power Losses Minimization Problem

Another reactive resource application in which control actions can be initiated locally by the distribution system relays is the power losses minimization problem. In a power system, most of the power losses are located in the distribution network. These losses increase demand during peak load conditions [25]. To reduce power losses, capacitors are used in the distribution system to provide reactive power support and voltage regulation [26]. By minimizing losses the voltage profile of the system also improves.

To formulate the problem, the equation used for power losses in this optimization problem will be defined. Figure 4.11 shows a diagram of a distribution feeder.

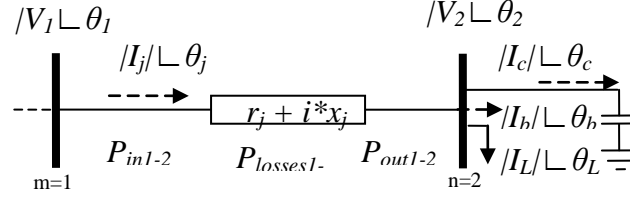


Figure 4.11: Branch diagram in a distribution feeder

The equation for the losses incurred in this feeder is

$$P_{losses}^j(c, x) = r_j * |I_j|^2 = \frac{|V_m|^2 |I_j|^2 \cos(\theta_m - \theta_j) - |V_n|^2 |I_j|^2 \cos(\theta_n - \theta_j)}{|V_m|^2 - |V_n|^2} \quad (4.99)$$

where  $c$  is the  $i^{th}$  capacitor in the power network,  $x$  are the state variables (voltages and currents),  $r_j$  is the resistance of the  $j^{th}$  branch,  $I^j$  is the current of the  $j^{th}$  branch, and  $V_m$  and  $V_n$  are the voltages of the  $j^{th}$  branch. In this analysis the current magnitude  $I^j$  is defined as

$$|I^j|^2 = (real(I_L^j) + real(I_b^j) + real(I_c^j))^2 + (imag(I_L^j) + imag(I_b^j) + imag(I_c^j))^2 \quad (4.100)$$

where  $I_L^j$  is the load current connected at the  $j^{th}$  branch,  $I_b^j$  is the current coming from other branches connected at the  $j^{th}$  branch and  $I_c^j$  is the current coming from the  $i^{th}$  capacitor connected to the  $j^{th}$  branch. Notice that these currents are at the end point of a branch. Also, for this analysis, the capacitors were assumed to be constant power devices. This assumption is based on the fact that, for a small time period, these capacitors could be seen as a constant power source because the local region responds in a matter of seconds in order to minimize system losses.

The resulting optimization problem is a mixed-integer programming problem because the capacitors are a discrete variable that is either on or off. However, for the analysis of this problem, the discrete variables are relaxed and are assumed to take constant values such that the mixed-integer programming problem can be reduced to a nonlinear programming problem.

Now the optimization problem can be defined as follows:

$$\begin{aligned} \min \quad & \sum_{j=1}^N P_{losses}^j(c, x) \\ \text{s.t.} \quad & DPF(c, x) \text{ constraints} \\ & 0 \leq c \leq (c)^{\max} \end{aligned} \quad (4.101)$$

where  $DPF$  are the distribution power-flow constraints that are going to be presented next. The power flow constraints are equality constraints describing the voltage and current relationship at each branch and node. For each voltage and current equation, there is a real and imaginary equation describing the governing behavior of voltages and currents in a distribution network.

The current equations for the power flow using Figure 4.11 can be written as:

$$R \left[ |I_j| \angle \theta_j = |I_L| \angle \theta_L + |I_c| \angle \theta_c + |I_b| \angle \theta_b \right] \quad (4.102)$$

$$I \left[ |I_j| \angle \theta_j = |I_L| \angle \theta_L + |I_c| \angle \theta_c + |I_b| \angle \theta_b \right] \quad (4.103)$$

where

$$|I_L| \angle \theta_L = \frac{|S_{Load}|}{|V_n|} \angle (\theta_{Load} - \theta_n) \quad (4.104)$$

$$|I_c| \angle \theta_c = \left( \frac{-i * CapMVar}{|V_n| \angle \theta_n} \right)^* \quad (4.105)$$



The inequality constraints are simply the maximum and minimum values the bus voltages can attain and the maximum branch currents passing through the feeder. In equation (4.105) the current is calculated assuming that the PQ load is constant. In equation (4.105), the capacitor current is calculated assuming that the capacitors are constant power devices and  $CapMVar$  is the capacitor MVar rating. Also if equation (4.105) is decomposed into its real and imaginary components, the following expression for the capacitor current is obtained:

$$\begin{aligned} |I_L| \cdot \cos \theta_L + i \cdot |I_L| \cdot \sin \theta_L = & \frac{-CapMVar|V_n| \sin \theta_n}{(|V_n|)^2} \\ & + i \cdot \left( \frac{CapMVar|V_n| \cos \theta_n}{(|V_n|)^2} \right) \end{aligned} \quad (4.106)$$

Equation (4.106) has a similar expression as the  $\Delta Q_{Load}$  of reactive load that was presented in Section 3.2. This means that the presented algorithms can be easily applied to the losses minimization problem.

#### 4.8.1 Thirty-Four Distribution Feeder Simulation and Results

In the first scenario, the algorithm was again tested on a modified IEEE 34-bus distribution system presented in Figure 4.12. The same modification as the ones presented in Section 4.7.1 are made. The capacitors to be controlled are located at buses 5, 12, 22, 27 and 29 respectively. For simplicity of the algorithms, this system was divided in two regions. For the auxiliary problem principle (APP) and the predictor-corrector proximal multiplier algorithm (PCPM), the two different regions are as shown in Figure 4.12. One region is

controlled by agent 1 and control load buses 1 to 24. The second region is controlled by agent 2 and control load buses 24 to 34.

For the Lagrangian relaxation (LR) algorithm and the Lagrangian relaxation based decomposition (LRBD) algorithm presented, the region decompositions are shown in Figure 4.13. One region is controlled by agent 1 and control load buses 1 to 24. The second region is controlled by agent 2 and control load buses 25 to 34.

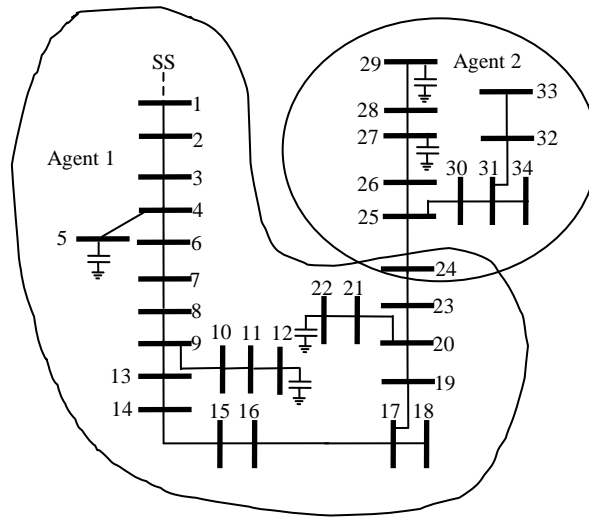


Figure 4.12: IEEE modified 34-bus feeder with capacitors connected to substation SS for APP and PCPM algorithm

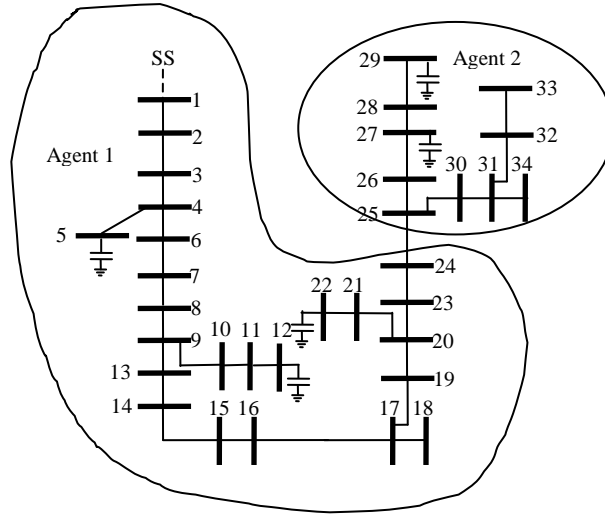


Figure 4.13: IEEE modified 34 bus feeder with capacitors connected to substation SS for LR and LRBD algorithm

In this case the algorithms seem to reduce the losses almost by the same amount. The four algorithms minimize the losses to a range between 83.21 kW and 83.42 kW (Tables 4.7 and 4.8). The best way to differentiate between algorithms is by comparing their performance, and again the two Lagrangian relaxation algorithms seem to outperform the augmented Lagrangian algorithms. The LR and LRBD algorithms take 65.15 seconds and 33.76 seconds (Table 4.9), respectively, to find a solution. In contrast the APP and the PCPM take 221.25 seconds and 558.97 seconds respectively. In terms of iterations, the LR again finds a solution in three iterations, followed by the APP with four iterations. The LRBD gives the best solution in losses but iterates 585 times, which makes it virtually impossible to implement with an agent based scheme.

Table 4.7: Capacitors kVARs comparison between centralized and decentralized algorithms in 34-bus feeder

Case	Central Opt. (kVARs)	APP (kVARs)	PCPM (kVARs)	LR (kVARs)	LRBD (kVARs)
Bus 5	90.037	81.803	81.585	76.682	88.783
Bus 12	110.94	107.3	107.25	106.46	109.58
Bus 22	124.14	117	117.17	139.99	119.21
Bus 27	251.8	227.73	227.85	201.9	245.38
Bus 29	26.477	26.478	26.478	26.478	26.385
Total	603.394	560.311	560.333	551.51	589.338

Table 4.8: Loss comparison between centralized and decentralized algorithms in 34-bus feeder

Case	Central Opt. (kW)	APP (kW)	PCPM (kW)	LR (kW)	LRBD (kW)	Losses No Caps (kW)
Losses	83.18	83.36	83.36	83.42	83.21	119.98

Table 4.9: Iterations and CPU time comparison for 34-bus feeder with capacitors

Case	Iterations	Seconds
Central	105	7.35
APP	4	221.25
PCPM	14	558.97
LR	3	65.146
LRBD	585	33.76

#### 4.8.2 Sixty-Nine Distribution Feeder Simulation and Results

In the first scenario the algorithm was again tested on a modified 69-bus distribution system [24] presented in Figures 4.14 and 4.15. The same modifications presented in Section 4.7.2 were made to the system. The capacitors to be controlled are located at buses 21, 22, 40, 45, 47, 61 and 69 respectively. The capacitors have the following limits: 100, 100, 400, 900, 100 and 100 kVARs respectively. In this system, the distribution feeder was divided in three different

regions. For the auxiliary problem principle (APP) the three different regions are as shown in Figure 4.14. One region is controlled by agent 1 and control load buses 1 to 28. The second region is controlled by agent 2 and control load buses 28 to 48. The third region is controlled by agent 3 and control load buses 48 to 69.

For the Lagrangian relaxation (LR) algorithm and the Lagrangian relaxation based decomposition (LRBD) algorithm, the region decompositions are shown in Figure 4.15. The first region is controlled by agent 1 and control load buses 1 to 28. The second region is controlled by agent 2 and control load buses 29 to 48. The third region is controlled by agent 3 and control load buses 49 to 69.

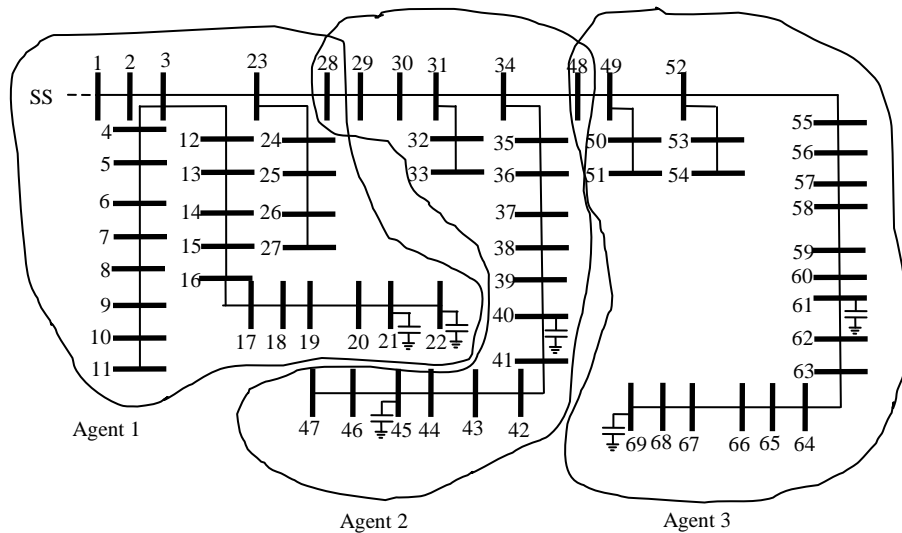


Figure 4.14: 69-bus feeder connected to substation SS for APP algorithm

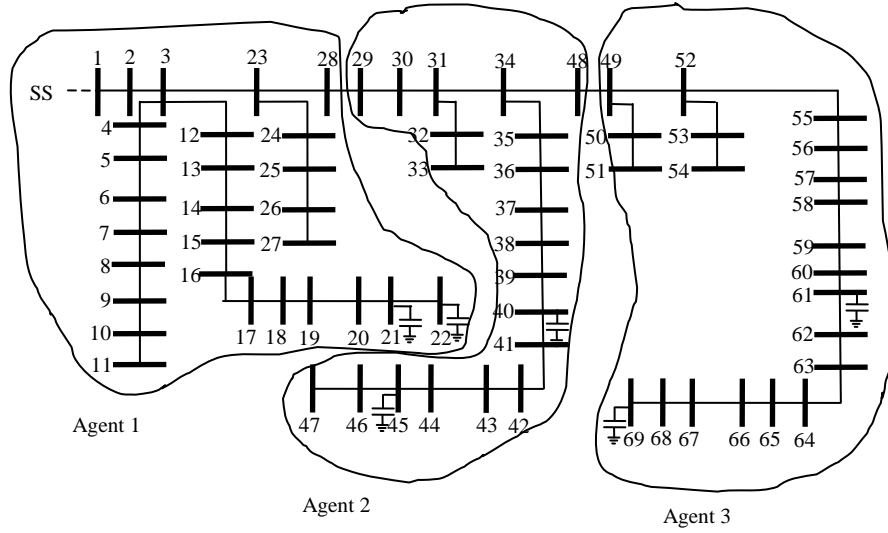


Figure 4.15: 69-bus feeder connected to substation SS for LR and LRBD algorithm

For this case the same tendencies as in the previous three scenarios are found. The LR and the LRBD decomposition algorithms estimate the losses as close as the centralized 345.2 kW solution (Tables 4.10 and 4.11). The LRBD solution in the sizing of the capacitors is almost the same as the centralized, but it iterates 218 times. The LR gives the best overall solution because it iterates five times and takes 365.7 seconds to find a solution (Table 4.12). The APP takes nine iterations and find a solution after 1067.66 seconds.

Table 4.10: Capacitors kVARs comparison between centralized and decentralized algorithms in 69-bus feeder

Case	Central Opt. (kVARs)	APP (kVARs)	LR (kVARs)	LRBD (kVARs)
Bus 21	106.06	105.39	105.39	101.39
Bus 22	105.86	105.39	105.39	109.62
Bus 40	400.81	310.0	221.79	405.77
Bus 45	910.64	880.0	911.42	914.31
Bus 61	104.12	83.0	100.35	102.09
Bus 68	104.56	83.5	100.32	116.62
Total	1732.05	1567.28	1544.66	1749.8

Table 4.11: Loss comparison between centralized and decentralized algorithms in 69-bus feeder

Case	Central Opt. (kW)	APP (kW)	LR (kW)	LRBD (kW)	Losses No Caps (kW)
Losses	345.2	348.18	347.27	344.57	442.18

Table 4.12 Iterations and CPU time comparison

Case	Iterations	Seconds
Central	439	163.85
APP	9	1067.66
LR	5	365.73
LRBD	218	48.84

## 4.9 Distribution Feeder Load Control Using Sensitivities

In the previous section, different decentralized optimization algorithms were presented. The results showed that the algorithm could be implemented to control reactive resources in a distribution network feeder. The only problem is that these algorithms require many iterations and exchanging data among the different regions in order to get to a solution. If any of the data is lost, the final result is not going to be the best possible solution.

In this section an algorithm that will follow the incident command system structure presented in Section 2.3 is presented. This algorithm still requires the exchange of data in the distribution system, but the amount of messages is less than in the decentralized optimization algorithms presented in the previous sections.

Before continuing with the proposed algorithm, the sensitivity analysis derivation is presented in the next section.

#### 4.9.1 Distribution Feeder Sensitivity Analysis

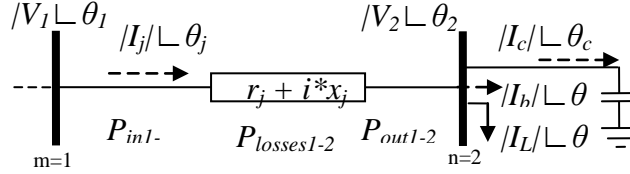


Figure 4.16: Branch diagram in a distribution feeder

Recalling from Section 3.2, and referring to Figure 4.16, the distribution power flow (DPF) equality constraints for the bus voltages are

$$\text{Re} \left[ \left( |V_m| \angle \theta_m \right) = \left( |V_n| \angle \theta_n \right) + \left( (r + i * x) * \left( |I_j| \angle \theta_j \right) \right) \right] \quad (4.107)$$

$$\text{Im} \left[ \left( |V_m| \angle \theta_m \right) = \left( |V_n| \angle \theta_n \right) + \left( (r + i * x) * \left( |I_j| \angle \theta_j \right) \right) \right] \quad (4.108)$$

Equations (4.107) and (4.108) may be rewritten as

$$\begin{aligned} \left( |V_n| \cdot \cos \theta_n \right) = \\ \left( |V_m| \cdot \cos \theta_m \right) - \left( |I_j| \cdot \cos \theta_j \right) \cdot (r) + \left( |I_j| \cdot \sin \theta_j \right) \cdot (x) \end{aligned} \quad (4.109)$$

$$\begin{aligned} \left( |V_n| \cdot \sin \theta_n \right) = \\ \left( |V_m| \cdot \sin \theta_m \right) - \left( |I_j| \cdot \cos \theta_j \right) \cdot (x) - \left( |I_j| \cdot \sin \theta_j \right) \cdot (r) \end{aligned} \quad (4.110)$$

From Figure 4.16, the DPF equality constraints for the line currents are

$$\text{Re} \left[ |I_j| \angle \theta_j = |I_L| \angle \theta_L + |I_b| \angle \theta_b \right] \quad (4.111)$$



$$\text{Im} \left[ |I_j| \angle \theta_j = |I_L| \angle \theta_L + |I_b| \angle \theta_b \right] \quad (4.112)$$

where

$$\begin{aligned} |I_L| \angle \theta_L &= \frac{|S_{Load}|}{|V_n|} \angle (\theta_{Load} - \theta_n) = \\ \left( \frac{S_{Load}}{V_n} \right)^* &= \left( \frac{P_{Load} + i \cdot Q_{Load}}{|V_n| \cos \theta_n + i \cdot |V_n| \sin \theta_n} \right)^* \end{aligned} \quad (4.113)$$

In this problem, the reactive load will be controlled and can be represented as

$$Q_{Load} = Q_{LoadOld} - \Delta Q_{Load} \quad (4.114)$$

$\Delta Q_{Load}$  will be calculated in the optimization problem.

Then equations (4.111) and (4.112) may be rewritten as

$$|I_j| \cdot \cos \theta_j = |I_L| \cdot \cos \theta_L + |I_b| \cdot \cos \theta_b \quad (4.115)$$

$$|I_j| \cdot \sin \theta_j = |I_L| \cdot \sin \theta_L + |I_b| \cdot \sin \theta_b \quad (4.116)$$

where  $I_L$  is the load current magnitude and  $I_b$  is the branch current of the next feeder connected at bus 2. After some algebra, if equations (4.113) and (4.114) are substituted into (4.115) and (4.116), the resulting load current components with their corresponding real and imaginary parts are

$$\begin{aligned} |I_L| \cdot \cos \theta_L + i \cdot |I_L| \cdot \sin \theta_L &= \\ \frac{P_{Load} \cdot |V_n| \cos \theta_n + Q_{LoadOld} \cdot |V_n| \sin \theta_n - \Delta Q_{Load} \cdot |V_n| \sin \theta_n}{(|V_n|)^2} + \\ i \cdot \left( \frac{P_{Load} \cdot |V_n| \sin \theta_n - Q_{LoadOld} \cdot |V_n| \cos \theta_n + \Delta Q_{Load} \cdot |V_n| \cos \theta_n}{(|V_n|)^2} \right) \end{aligned} \quad (4.117)$$

where  $Q_{LoadOld}$  is the original load magnitude at the load bus.

Now that all of the required distribution network equations have been discussed, the sensitivity computations are presented. In this derivation is assumed that a function  $f$  representing the bus voltage ( $V$ ) magnitude exists:

$$f(x,u) = V(x,u) \quad (4.118)$$

where  $x$  is a vector of all of the state variables that correspond to bus voltages and angles, and to the line currents and angles

$$x = \begin{bmatrix} V \\ \theta_V \\ I \\ \theta_I \end{bmatrix} \quad (4.119)$$

and  $u$  is the controllable reactive load ( $\Delta Q_{Load}$ ):

$$u = \Delta Q_{Load} \quad (4.120)$$

Then the sensitivity of  $f$  with respect to  $u$  can be computed as the partial derivatives as

$$\frac{\partial f}{\partial u} = Sensf_u = \left[ \frac{\partial V}{\partial x} \frac{\partial x}{\partial u} + \frac{\partial V}{\partial u} \right] \quad (4.121)$$

In order to obtain the sensitivities of state variables to change in reactive power, additional relationships between the state variables and the power flow equations are needed. Using the first order approximation of the DPF around an initial point, the following relationship is obtained:

$$\begin{bmatrix} \Delta V \\ \Delta \theta_V \\ \Delta I \\ \Delta \theta_I \end{bmatrix} = - \left[ \frac{\partial g_{DPF}}{\partial x} \right]^{-1} [\Delta g_{DPF}] \quad (4.122)$$

where  $g_{DPF}$  are the power flow equations presented in (4.106) to (4.115) used to solve the distribution power flow.

It is important to mention that equation (4.121) comes from the fact that the state variables can be obtained using the Newton power flow applied to the distribution power equations. This approach converges if a good starting point is applied and as long as the changes between one starting point and another are not big. For the cases when the reactive load is changing, the algorithm converges without any problems. The other relationship needed is between the bus voltages and line currents and the changes in reactive load power. Thus from this relationship the sensitivities of bus voltages and line currents to reactive load changes ( $\Delta u = \Delta Q_{Load}$ ) are obtained in the following matrix:

$$[\Delta g_{DPF}] = \left[ \frac{\partial g_{DPF}}{\partial u} \right] [\Delta u] \quad (4.123)$$

Now the relationship between the state variables and the changes in reactive power can be determined. The resulting matrix is the sensitivity of state variables to changes in reactive power.

$$\begin{bmatrix} \Delta V \\ \Delta \theta_v \\ \Delta I \\ \Delta \theta_l \end{bmatrix} = \frac{\partial x}{\partial u} \begin{bmatrix} \Delta V \\ \Delta \theta_v \\ \Delta I \\ \Delta \theta_l \end{bmatrix} \quad (4.124)$$

$$\frac{\partial x}{\partial u} = - \left[ \frac{\partial g_{DPF}}{\partial x} \right]^{-1} \frac{\partial g_{DPF}}{\partial u} \quad (4.125)$$

Thus now:

$$Sensf_{-u} = \left[ \frac{\partial V}{\partial x} \frac{\partial x}{\partial u} + \frac{\partial V}{\partial u} \right] = \left[ - \frac{\partial V}{\partial x} \left[ \frac{\partial g_{DPF}}{\partial x} \right]^{-1} \frac{\partial g_{DPF}}{\partial u} + \frac{\partial V}{\partial u} \right] \quad (4.126)$$

In order to get the partial derivative  $\partial V / \partial x$  and  $\partial V / \partial u$ , the equations, (4.109), (4.110), (4.115), (4.116) and (4.117) are combined resulting in the following voltage magnitude equation:

$$|V_n| = \sqrt{\left(|V_n| \cdot \cos \theta_n\right)^2 + \left(|V_n| \cdot \sin \theta_n\right)^2} = \sqrt{\left(\left(|V_m| \cdot \cos \theta_m\right) - \left(|I_j| \cdot \cos \theta_j\right) \cdot (r) + \left(|I_j| \cdot \sin \theta_j\right) \cdot (x)\right)^2 + \left(\left(|V_m| \cdot \sin \theta_m\right) - \left(|I_j| \cdot \cos \theta_j\right) \cdot (x) - \left(|I_j| \cdot \sin \theta_j\right) \cdot (r)\right)^2} \quad (4.127)$$

where

$$|I_j| \cdot \cos \theta_j = \frac{P_{Load} \cdot |V_n| \cos \theta_n + Q_{LoadOld} \cdot |V_n| \sin \theta_n - \Delta Q_{Load} \cdot |V_n| \sin \theta_n}{\left(|V_n|\right)^2} + |I_b| \cdot \cos \theta_b \quad (4.128)$$

$$|I_j| \cdot \sin \theta_j = \frac{P_{Load} \cdot |V_n| \sin \theta_n - Q_{LoadOld} \cdot |V_n| \cos \theta_n + \Delta Q_{Load} \cdot |V_n| \cos \theta_n}{\left(|V_n|\right)^2} + |I_b| \cdot \sin \theta_b \quad (4.129)$$

To illustrate these derivations, the Figure 4.16 is assumed to be used; thus only one feeder is connected at bus 2. The derivation of these equations is presented in Appendix A.

#### 4.9.2 Distribution Feeder Optimization with Sensitivities

Once the sensitivities are calculated, the optimization problem equality constraints are modified to include the reactive load sensitivities, in particular the equations concerning the top of the feeder. If the feeder in Figure 4.16 is used to develop the distribution power flow (DPF) equations in an optimization problem the voltage at bus  $m$  will need to be fixed in order to solve the optimization problem. The idea is to estimate this voltage change by using the sensitivities

computed in equation 4.118. As was seen with the results in Section 4.7 and 4.8, the decentralized optimization algorithms need to iterate to find a suitable solution. This is because a local optimization problem cannot find a solution on its own without taking into the analysis the effect of surrounding and local regions.

In this section a simpler solution is developed to solve the reactive load optimization problem without the need for iterating to find a local solution. This algorithm is only intended for use in particular feeders when low voltage or low power factor conditions are detected.

If equations (4.109) and (4.110) are rewritten as an equality, then the voltage magnitude of bus 1 is going to be estimated by using sensitivities. The resulting equations are:

$$\begin{aligned} (|V_n| \cdot \cos \theta_n) = \\ (|Vsens_m| \cdot \cos \theta_m) - (|I_j| \cdot \cos \theta_j) \cdot (r) + (|I_j| \cdot \sin \theta_j) \cdot (x) \end{aligned} \quad (4.130)$$

$$\begin{aligned} (|V_n| \cdot \sin \theta_n) = \\ (|Vsens_m| \cdot \sin \theta_m) - (|I_j| \cdot \cos \theta_j) \cdot (x) - (|I_j| \cdot \sin \theta_j) \cdot (r) \end{aligned} \quad (4.131)$$

where

$$Vsens_m = Vold_m + Sensf_{\Delta Q1} \cdot \Delta Q1 + \dots + Sensf_{\Delta QN} \cdot \Delta QN \quad (4.132)$$

$Vold_m$  is the initial voltage before performing the optimization problem and  $Vsens_m$  is the new update at each iteration of the local optimization problem.  $N$  is the number of buses. Thus  $Vsens_m$  estimates the bus voltage based on the change in reactive power at a certain bus. The local optimization problem can be formulated as follows:

$$\begin{aligned}
\min \quad & \sum_{k=1}^N (V_k(x,u) - V_{k\_spec})^2 \\
s.t. \quad & \text{DPF}(x,u) \\
& 0 \leq u \leq (u)^{\max}
\end{aligned} \tag{4.133}$$

In the previous problem the only modified DPF equations are the first two equations concerning the bus voltage at the top of the feeder.

This modification comes from the fact that in any particular distribution feeder, in order to obtain a power flow solution, the bus voltage at the top of the feeder needs to be fixed in order to get a solution. If an optimization is performed assuming a fixed value, the results will be dependent on the magnitude of the fixed bus voltage and angle at the top of the feeder. By using sensitivities, the effect of changing reactive power in the system can be estimated.

#### **4.9.3 Simulations and Results on Three Lateral Feeders on the Sixty-Nine Distribution Feeder**

The optimization was tested in the 69-bus distribution feeder that was presented in Figure 4.9 on page 115. For this case the algorithm was run in three of the laterals. The first lateral is the lateral from bus 3 to bus 22. The second is the lateral from bus 34 to bus 47. The final lateral is the one from bus 48 to bus 69.

The results for lateral 3-22 are presented in Tables 4.13 - 4.15. The voltage results in Table 4.13 show that the modified optimization using sensitivities computes the voltage almost the same as the centralized optimization. If the modified voltages are compared to the power flow solution using the calculated

reactive load (Table 4.14) obtained from the modified optimization, it can be seen that the voltages are very similar. The sensitivities computed almost the same amount of reactive load, 1312.72 kVARs and 1307.34 kVARs, for the centralized and modified optimization, respectively.

Also, when comparing the calculated power factor of the power input at bus 12, it can be seen that they are almost identical. It was 0.83163 for the centralized case and 0.83122 for the modified case.

Table 4.13: Lateral voltages from bus 3 to bus 22

Case	Centralized (p.u.)	Modified Sensitivities (p.u.)	Power Flow Solution (p.u.)
Bus 12	0.99929	0.99928	0.99929
Bus 13	0.99319	0.99318	0.99319
Bus 14	0.98791	0.98789	0.9879
Bus 15	0.98638	0.98636	0.98638
Bus 16	0.98632	0.9863	0.98631
Bus 17	0.96228	0.96224	0.96226
Bus 18	0.9526	0.95254	0.95256
Bus 19	0.95132	0.95126	0.95128
Bus 20	0.95105	0.95099	0.95101
Bus 21	0.94775	0.94778	0.94781
Bus 22	0.94774	0.94777	0.94779

Table 4.14: Lateral reactive loads from bus 3 to bus 22

Case	Control Load (kVARs)	Initial Load (kVARs)	Central Opt.	Modified Sensitivities
Bus 18	100	114.3	14.29	114.3
Bus 21	700	1126.3	613.59	518.96
Bus 22	700	1126.3	426.3	426.3
Total:	1500		1312.72	1307.34

Table 4.15: Initial top feeders bus voltages, currents, angles and power factors of lateral 3-22

	Central Op	Modified Sensitivities
Voltage Bus12	0.99929	0.99928
Angle Voltage Bus 12	-0.027182	-0.026725
Current 3-22	8.7461	8.7471
Angle Current 3-22	-33.761	-33.802
Pf	0.83163	0.83122
Initial pf	0.76132	0.76132

The results for the laterals 34-47 are presented in Tables 4.16 - 4.18. The voltage results in Table 4.16 show that the modified optimization using sensitivities computes the voltage almost the same as the centralized optimization. But if the modified voltages are compared to the power flow solution using the calculated reactive load (Table 4.17) obtained from the modified optimization, it can be seen that the voltages are very similar. In this case the modified case computed almost the same amount of reactive load, 1622.93 kVARs and 1476.04 kVARs, for the centralized and modified optimization, respectively. The amount is different but the results are accurate. This is because the sensitivities are computed a high voltage value at bus 34, but this is not much different from the actual case.

Also, when comparing the calculated power factor of the power input at bus 35, it can be seen that are almost identical. It was 0.96701 for the centralized case and 0.98741 for the modified case. The modified case overestimates the actual power factor but it is correct for the amount of reactive load computed.



Table 4.16: Lateral voltages from bus 34 to bus 47

Case	Centralized (p.u.)	Modified Sensitivities (p.u.)	Power Flow Solution (p.u.)
Bus 35	0.9805	0.98084	0.97999
Bus 36	0.97892	0.97917	0.97831
Bus 37	0.97677	0.97689	0.97603
Bus 38	0.97471	0.9747	0.97384
Bus 39	0.96247	0.96197	0.9611
Bus 40	0.95651	0.95577	0.9549
Bus 41	0.95422	0.95338	0.95251
Bus 42	0.9516	0.95065	0.94977
Bus 43	0.94854	0.94733	0.94646
Bus 44	0.94839	0.94718	0.94631
Bus 45	0.94821	0.947	0.94612
Bus 46	0.94735	0.94614	0.94527
Bus 47	0.94709	0.94588	0.94501

Table 4.17: Lateral reactive loads from bus 34 to bus 47

Case	Control Load (kVARs)	Initial Load (kVARs)	Central Opt.	Modified Sensitivities
Bus 43	1776	888	-326.93	-180.04
Bus 46	324	162	-162	-162
Bus 47	84	42	-42	-42
Total:	2184		1622.93	1476.04

Table 4.18: Initial top feeders bus voltages, currents, angles and power factors of lateral 34-47

	Central Op	Modified Sensitivities
Voltage Bus 35	0.9805	0.98084
Angle Voltage Bus 35	-0.36848	0.18246
Current 34-35	1.5007	1.4653
Angle Current 34-35	14.389	9.2832
Pf	0.96701	0.98741
Initial pf	0.75672	0.75672

The results for the last lateral 48-69 show the same tendencies as the previous two laterals. In this case the reactive load calculated for the centralized case was 892.78 kVARs but only 560.38 kVARs for the modified case (Table 4.19). But the voltages, calculated for the modified case were correct for that amount of

reactive load, as can be seen in Table 4.20 when they are compared to the power flow solution. That is the reason the modified case calculates an input power factor at bus 49 of 0.96592, while the centralized case computes a power factor of 0.99873 (Table 4.21).

Overall, the results of the modified case are very similar to those of the centralized case and the accuracy is not much different from that obtained for the decentralized cases. This is the reason this type of solution can be implemented in an algorithm: it can find the amount of reactive load to be controlled without the need to constantly iterate between regions to obtain a solution.

Table 4.19: Lateral reactive loads from bus 48 to bus 69

Case	Control Load (kVARs)	Initial Load (kVARs)	Central Opt.	Modified Sensitivities
Bus 52	208	104	-104	104
Bus 58	60	30	30	30
Bus 63	362	181	-43.78	80.623
Bus 66	240	120	-120	-120
Bus69	220	110	-110	-110
Total:	1090		892.78	560.377

Table 4.20: Lateral voltages from bus 48 to bus 69

Case	Centralized (p.u.)	Modified Sensitivities (p.u.)	Power Flow Solution (p.u.)
Bus 49	0.97214	0.97244	0.97038
Bus 50	0.97209	0.97238	0.97033
Bus 51	0.97209	0.97238	0.97032
Bus 52	0.96822	0.96802	0.96595
Bus 53	0.96789	0.96769	0.96562
Bus 54	0.96789	0.96769	0.96562
Bus 55	0.96374	0.96329	0.96119
Bus 56	0.95927	0.95855	0.95644
Bus 57	0.95481	0.95381	0.95169
Bus 58	0.95399	0.95293	0.95081
Bus 59	0.95255	0.95139	0.94927
Bus 60	0.95253	0.95137	0.94925
Bus 61	0.95169	0.95044	0.94832
Bus 62	0.95115	0.94984	0.94772
Bus 63	0.95027	0.94887	0.94675
Bus 64	0.95026	0.94885	0.94673
Bus 65	0.95006	0.94866	0.94654
Bus 66	0.94965	0.94824	0.94612
Bus 67	0.94918	0.94776	0.94565
Bus 68	0.94898	0.94757	0.94546
Bus 69	0.9489	0.94748	0.94537

Table 4.21: Initial top feeders bus voltages, currents, angles and power factors lateral 48-69

	Central Op	Modified Sensitivities
Voltage Bus 49	0.97214	0.97244
Angle Voltage Bus 49	-0.21893	0.37887
Current 48-49	1.1716	1.155
Angle Current 48-49	2.6636	-14.623
pf	0.99873	0.96592
Initial pf	0.80722	0.80722

#### 4.9.4 Distribution Feeder Optimization with Sensitivities

In Section 2.4.2 a local control scheme was presented to initiate control actions locally by the distribution feeder. The optimization of the feeder using sensitivities can be used for this local control scheme by implementing the following algorithm. In Figure 4.17 the feeder relays architecture is presented. A distribution network can be divided in regions and each of the regions will have a

top feeder relay (TFR) that will control devices. The three lateral feeders of the 69-bus distribution network presented in the last section can be controlled by one TFR. The regions will implement the following algorithm:

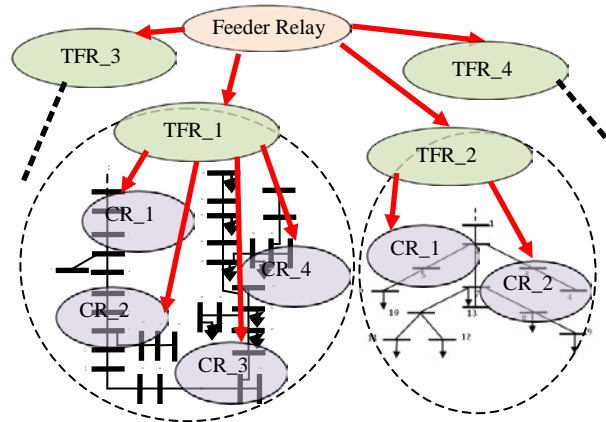


Figure 4.17: Feeder relays and top feeder relay agents in the distribution network

- 1) The TFR detects a low voltage situation inside its region for which it has the authority to initiate a local response.
- 2) The TFR relay will send the latest data to the feeder relay (FR). Since the FR is receiving data from all of the distribution network it can use that information to compute the sensitivities of voltage to reactive power.
- 3) After the sensitivities are computed they are sent to the TFR relays. Then the TFR will use the latest sensitivities to run the optimization problem presented in Section 4.9.2.
- 4) After the solution is found, the TFR sends control commands to the controllers (CR)
- 5) Once the control action is performed, the TFR will send a message to the

FR indicating that the control actions were performed. By doing this, the FR will have a log of the control actions that are being performed in the power grid.

This type of algorithm can be easily implemented with the ICS algorithm presented in Chapter 2. This algorithm will be like another function of the agent. This algorithm follows the same command structure of the ICS commands and does not require iterations to find a solution to a problem. That is the reason the decentralized optimization algorithms are not implemented with the presented agent's architecture. The sensitivity analysis was incorporated in the ICS architecture presented in Chapter 3.

#### **4.10 References**

- [1] T. A. Short, *Electric Power Distribution Equipment and Systems*. Boca Raton, FL: Taylor & Francis Group CRC Press, 2006.
- [2] *Electric Power Systems and Equipment-Voltage Ratings (60 Hertz)*, ANSI Standard C.84.1:2006,2006.
- [3] P. N. Vovos, A. E. Kiprakis, A. R. Wallace and G. P. Harrison, "Centralized and Distributed Voltage Control: Impact on Distributed Generation Penetration," *IEEE Transaction on power Systems*, vol. 22, no. 1, pp. 476-483, Feb. 2007.
- [4] A. Uchida, S. Watanabe and S. Iwamoto, "A Voltage Control Strategy for Distribution Networks with Dispersed Generations," in *IEEE Power Engineering Society General Meeting*, pp. 1-6, 2007
- [5] C. W. Brice, "Comparison of Approximate and Exact Voltage Drop Calculations for Distribution Lines," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, no. 11, pp. 4428-44231, Nov. 1982.

- [6] F. Katiraei, C. Abbey and R. Bahry, "Analysis of Voltage Regulation Problem for a 25-kV Distribution Network with Distributed Generation," in *IEEE Power Engineering Society General Meeting*, pp. 1-8, 2006.
- [7] B. H. Kim and R. Baldick, "Coarse-Grained Distributed Power Flow," *IEEE Transactions on Power Systems*, vol. 12, no. 2, pp. 932-939, May 1997.
- [8] C. H. Lin and S. Y. Lin, "Distributed Optimal Power Flow With Discrete Control Variables of Large Distributed Power Systems," *IEEE Transactions on Power Systems*, vol. 23, pp. 1383-1392, Aug. 2008.
- [9] G. Hug-Glanzmann and G. Andersson, "Decentralized Optimal Power Flow Control for Overlapping Areas in Power Systems," *IEEE Transactions on Power Systems*, vol. 24, pp. 327-336, Feb. 2009.
- [10] G. Cohen, "Optimization by Decomposition and Coordination: A Unified Approach," *IEEE Transaction on Automatic Control*, vol. AC-23, no. 2, pp. 222-232, April 1978.
- [11] G. Cohen and D. L. Zhu, "Decomposition Coordination Methods in Large Scale Optimization Problems: The Non-Differentiable Case and the Use of Augmented Lagrangians," *Advances in Large Scale Systems*, J. B. Cruz, ed. Greenwich, CT: JAI Press, pp. 203-266, 1984.
- [12] D. P. Bertsekas, *Nonlinear Programming*, 2<sup>nd</sup> ed. Belmont, MA: Athena Scientific, 2003.
- [13] B. H. Kim and R. Baldick, "A Comparison of Distributed Distributed Power Flow," *IEEE Transactions on Power Systems*, vol. 15, no. 2, pp. 599-604, May 2000.
- [14] R. T. Rockafellar, "Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming," *Mathematics of Operations Research*, vol. 1, no. 2, pp. 97-116, May 1976.
- [15] G. Chen and M. Teboulle, "A Proximal-Based Decomposition Method for Convex Minimization Problems," *Mathematical Programming*, vol. 64, no. 1, pp. 81-101, 1994.
- [16] A. Merlin and P. Sandrin, "A New Method for Unit Commitment at Electricité de France," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-102, no. 5, pp. 1218-1225, May 1983.
- [17] F. Zhuang and F. D. Galiana, "Toward a more Rigorous and Practical Unit Commitment by Lagrangian Relaxation," *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 763-773, May 1988.

- [18] H. Yan, P. B. Luh, X. Guan and P. M. Rogan, "Scheduling of Hydrothermal Power Systems," *IEEE Transactions on power Systems*, vol. 8, no. 3, pp. 1135-1365, Aug. 1993.
- [19] S. J. Wang, S. M. Shahidehpour, D. S. Kirschen, S. Mokhtari and G. S. Irisarri, "Short-Term Generation Scheduling with Transmission and Environmental Constraints using an Augmented Lagrangian Relaxation," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1333-1340, Aug. 1993.
- [20] J. A. Aguado, V. H. Quintana and A. J. Conejo, "Optimal Power Flows of Interconnected Power Systems," in *Proceedings of the IEEE Power Engineering Society Summer Meeting*, vol. 2, pp. 814-819, 1999.
- [21] A. J. Conejo and J. A. Aguado, "Multi-Area Coordinated Decentralized DC Optimal Power Flow," *IEEE Transactions on Power Systems*, vol. 13, no. 4, pp. 1272-1278, Nov. 1998.
- [22] A. J. Conejo, F. J. Nogales and F. J. Prieto, "A Decomposition Procedure Based on Approximate Newton Directions," *Mathematical Programming*, vol. 93, no. 3, pp. 495-515, Sep. 2002.
- [23] A. J. Conejo, F. J. Nogales and F. J. Prieto, "A Decomposition Methodology Applied to Multi-Area Optimal Power Flow Problem," *Annals of Operations Research*, no. 120, pp. 99-116, 2003.
- [24] M. E. Baran and F. F. Wu, "Optimal Capacitor Placement on Radial Distribution Systems," *IEEE Transactions on Power Delivery*, vol. 4, no. 1, pp. 725-734, Jan. 1989.
- [25] J. J. Grainger and S. H. Lee, "Capacity Release by Shunt Capacitor Placement on Distribution Feeders: A new voltage-dependent model," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, no. 5, May 1982, pp. 1236-1244.
- [26] K. M. Rogers, R. Klump, H. Khurana, T. J. Overbye, "Smart-Grid –Enabled Load and Distributed Generation as a Reactive Resource," in *2010 IEEE PES Conference on Innovative Smart Grid Technologies*, 2010, pp. 1-8.

## 5 CONCLUSIONS

This thesis presented a control algorithm framework that could be implemented in the smart grid. The framework is based on a hierarchical structure in which each action follows a chain of command from the top layer (control center) to the bottom layers (distribution network and loads).

In Chapter 2 the control framework was introduced in an algorithm that controls real power to relieve certain line overloads. The algorithm shows the interactions needed between the transmission and the distribution network in order to achieve an accurate solution. Then the incident command system was introduced. The framework is based on a chain of command in which high level individuals are responsible for low level individuals. For the power system events of interest in this thesis, the individual end-user real and reactive-power-controllable devices are the resources to be controlled. Similar to the personnel resources in the ICS, end-user devices do not normally work together, but they have the same goal in a crisis. Thus, a key component of the control algorithm framework is a reactive load control optimization algorithm to improve the voltage profile in distribution networks. The unified control algorithm framework, encompassing both the transmission and distribution networks and their associated agents, was implemented in software and tested on two different systems, where it was found to provide effective voltage control.

In the distribution network, control actions can be initiated locally to find solutions to certain problems. That is the reason that in Chapter 4 decentralized optimization problems were studied to find a solution to control reactive power



resources. The idea behind the analysis was to find the necessary requirements to implement local control in the distribution network. Four decentralized optimization techniques were studied in two different distribution networks. From the analysis, the Lagrangian relaxation algorithms show the best results to implement a decentralized scheme to control reactive resources. The only problem with the decentralized algorithms is that they still need to iterate between regions in order to find a solution. This will make the process very dependent on communication and constant interchange of information and data among regions in the distribution network.

Since capacitors are another reactive power resource to be controlled, Chapter 4 of the thesis also presented a decentralized optimization algorithm to minimize losses in the distribution network. The decentralized algorithm results were found to be similar to those using a centralized algorithm. The results corroborate those results found when controlling reactive loads. Again, the Lagrangian relaxation algorithms perform better in achieving a global solution to a certain problem.

Because the decentralized optimization algorithm needs to iterate among regions to find a solution, another algorithm was introduced in Chapter 4 to find a local solution to reactive resource problems in the distribution network. The algorithm is based on computing sensitivities of voltages to reactive resources to estimate the top of a feeder bus voltage of a particular region inside the distribution network. The algorithm showed that it can effectively find a solution to a local problem, and the results were similar to a centralized optimization problem. The accuracy of the algorithm is not that different when it is compared

to the decentralized optimization algorithm's results. This type of algorithm could be implemented locally to improve the voltage profile in a distribution feeder lateral, and it is very simple to implement. Another important fact is that it is based on the way a distribution power flow finds solutions of state variables in a system.

For future work, the algorithms will be tested with real load hardware. Instead of only simulating the effect of a controller, a battery can be used to provide reactive power to the grid. Also, the distribution optimization will be formulated to consider a three-phase model of the distribution network. With this formulation, a three-phase reactive sensitivity will be computed to see if this type of approach can help in finding local solutions to real power distribution networks. With a three-phase problem formulation, the effect of many distribution network devices can be incorporated into the analysis.

## APPENDIX A SENSITIVITY CALCULATIONS

Sensitivity derivation equations can be computed with the partial derivatives of the voltage equation with respect to the state variables. Starting with the partial derivative with respect to current, then

$$\frac{\partial V}{\partial I_b} = \frac{(Iec1 + Iec2)}{(2 \cdot V_n^2 \cdot Iec3)} \quad (A.1)$$

where

$$Iec1 = 2 \cdot (-V_n \cdot r \cdot \cos(\theta_b) + V_n \cdot x \cdot \sin(\theta_b)) \cdot \left( \begin{aligned} &(-V_n \cdot I_b \cdot r \cdot \cos(\theta_b) + (\Delta Q_{Load} \cdot x - x \cdot Q_{Load} - P_{Load} \cdot r) \cdot \cos(\theta_n) + \dots) \\ &V_n \cdot V_m \cdot \cos(\theta_m) + V_n \cdot I_b \cdot x \cdot \sin(\theta_b) + \dots \\ &(x \cdot P_{Load} + (\Delta Q_{Load} - Q_{Load}) \cdot r) \cdot \sin(\theta_n) \end{aligned} \right) \quad (A.2)$$

$$Iec2 = 2 \cdot (V_n \cdot x \cdot \cos(\theta_b) + V_n \cdot r \cdot \sin(\theta_b)) \cdot \left( \begin{aligned} &(V_n \cdot I_b \cdot x \cdot \cos(\theta_b) + (\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot \cos(\theta_n) + \dots) \\ &+ V_n \cdot I_b \cdot r \cdot \sin(\theta_b) + x \cdot (-\Delta Q_{Load} + Q_{Load}) \cdot \sin(\theta_n) + \dots \\ &r \cdot P_{Load} \cdot \sin(\theta_n) - V_n \cdot V_m \cdot \sin(\theta_m) \end{aligned} \right) \quad (A.3)$$

$$Iec3 = \sqrt{1/V_n^2 \cdot (A^2 + B^2)} \quad (A.4)$$

$$A = -V_n \cdot I_b \cdot r \cdot \cos(\theta_b) + (\Delta Q_{Load} \cdot x - x \cdot Q_{Load} - P_{Load} \cdot r) \cdot \cos(\theta_n) + \dots \\ + V_n \cdot I_b \cdot x \cdot \sin(\theta_b) + (x \cdot P_{Load} + (\Delta Q_{Load} - Q_{Load}) \cdot r) \cdot \sin(\theta_n) + \dots \\ + V_n \cdot V_m \cdot \cos(\theta_m) \quad (A.5)$$

$$B = V_n \cdot I_b \cdot x \cdot \cos(\theta_b) + (\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot \cos(\theta_n) + \dots \\ + V_n \cdot I_b \cdot r \cdot \sin(\theta_b) + x \cdot (-\Delta Q_{Load} + Q_{Load}) \cdot \sin(\theta_n) + \dots \\ r \cdot P_{Load} \cdot \sin(\theta_n) - V_n \cdot V_m \cdot \sin(\theta_m) \quad (A.6)$$

Also:

$$\frac{\partial V}{\partial \theta_b} = \frac{(\theta bec1 + \theta bec2)}{(2 \cdot V_n^2 \cdot \theta bec3)} \quad (A.7)$$

where

$$\begin{aligned} \theta_{bec1} = & 2 \cdot (V_n \cdot I_b \cdot x \cdot \cos(\theta_b) + V_n \cdot I_b \cdot r \cdot \sin(\theta_b)) \cdot \\ & \left( -V_n \cdot I_b \cdot r \cdot \cos(\theta_b) + (\Delta Q_{Load} \cdot x - x \cdot Q_{Load} - P_{Load} \cdot r) \cdot \cos(\theta_n) + \dots \right. \\ & \left. V_n \cdot V_m \cdot \cos(\theta_m) + V_n \cdot I_b \cdot x \cdot \sin(\theta_b) + \dots \right. \\ & \left. (x \cdot P_{Load} + (\Delta Q_{Load} - Q_{Load}) \cdot r) \cdot \sin(\theta_n) \right) \end{aligned} \quad (A.8)$$

$$\begin{aligned} \theta_{bec2} = & 2 \cdot (V_n \cdot I_b \cdot r \cdot \cos(\theta_b) - V_n \cdot I_b \cdot x \cdot \sin(\theta_b)) \cdot \\ & \left( V_n \cdot I_b \cdot x \cdot \cos(\theta_b) + (\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot \cos(\theta_n) + \dots \right. \\ & \left. + V_n \cdot I_b \cdot r \cdot \sin(\theta_b) + x \cdot (-\Delta Q_{Load} + Q_{Load}) \cdot \sin(\theta_n) + \dots \right. \\ & \left. r \cdot P_{Load} \cdot \sin(\theta_n) - V_n \cdot V_m \cdot \sin(\theta_m) \right) \end{aligned} \quad (A.9)$$

$$\theta_{bec3} = \sqrt{1/V_n^2 \cdot (A^2 + B^2)} \quad (A.10)$$

$$\begin{aligned} A = & -V_n \cdot I_b \cdot r \cdot \cos(\theta_b) + (\Delta Q_{Load} \cdot x - x \cdot Q_{Load} - P_{Load} \cdot r) \cdot \cos(\theta_n) + \dots \\ & + V_n \cdot I_b \cdot x \cdot \sin(\theta_b) + (x \cdot P_{Load} + (\Delta Q_{Load} - Q_{Load}) \cdot r) \cdot \sin(\theta_n) + \dots \\ & + V_n \cdot V_m \cdot \cos(\theta_m) \end{aligned} \quad (A.11)$$

$$\begin{aligned} B = & V_n \cdot I_b \cdot x \cdot \cos(\theta_b) + (\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot \cos(\theta_n) + \dots \\ & + V_n \cdot I_b \cdot r \cdot \sin(\theta_b) + x \cdot (-\Delta Q_{Load} + Q_{Load}) \cdot \sin(\theta_n) + \dots \\ & r \cdot P_{Load} \cdot \sin(\theta_n) - V_n \cdot V_m \cdot \sin(\theta_m) \end{aligned} \quad (A.12)$$

Also:

$$\frac{\partial V}{\partial V_n} = \frac{(V_{nec1} + V_{nec2})}{(V_{nec3})} \quad (A.13)$$

where

$$\begin{aligned}
Vnec1 = & \left(1/V_n^4\right) \cdot \left(2 \cdot \left(-2V_n I_b r \cos(\theta_b) + \dots\right.\right. \\
& \left.\left.2 \cdot V_n \cdot (V_m \cdot \cos(\theta_m) + I_b \cdot x \cdot \sin(\theta_b))\right)\right) \cdot \\
& \left( \begin{aligned}
& -\left(V_n^2\right) \cdot (I_b \cdot r \cdot \cos(\theta_b)) + \left(\frac{\Delta Q_{Load} \cdot x - x \cdot Q_{Load}}{P_{Load} \cdot r} - \right) \cdot V_m \cdot \cos(\theta_n) + \dots \\
& \left(V_n^2\right) \cdot (V_m \cdot \cos(\theta_m) + I_b \cdot x \cdot \sin(\theta_b)) + \dots \\
& (\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot V_m \cdot \sin(\theta_n) + \dots \\
& 2 \cdot (2 \cdot V_n \cdot I_b \cdot x \cdot \cos(I_b) + 2 \cdot V_n \cdot I_b \cdot r \cdot \sin(I_b) - 2 \cdot V_n \cdot V_m \cdot \sin(\theta_m)) \cdot \dots \\
& \left(\left(V_n^2\right) \cdot (I_b \cdot x \cdot \cos(\theta_b)) + (\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot V_m \cdot \cos(\theta_n) + \right. \\
& \left.\left(V_n^2\right) \cdot (I_b \cdot r \cdot \sin(\theta_b)) + x(-\Delta Q_{Load} + Q_{Load}) \cdot V_m \cdot \sin(\theta_n) + \dots\right. \\
& \left.P \cdot r \cdot V_m \cdot \sin(\theta_n) - \left(V_n^2\right) \cdot (V_m \cdot r \cdot \sin(\theta_m))\right)
\end{aligned} \right) \quad (A.14)
\end{aligned}$$

$$\begin{aligned}
Vnec2 = & \left(1/V_n^5\right) \cdot (4) \cdot \\
& \left( \begin{aligned}
& \left(-\left(V_n^2\right) \cdot (I_b \cdot r \cdot \cos(\theta_b)) + \dots\right. \\
& \left.\left(\frac{\Delta Q_{Load} \cdot x - x \cdot Q_{Load}}{P_{Load} \cdot r} - \right) \cdot V_m \cdot \cos(\theta_n) + \dots\right. \\
& \left.\left(V_n^2\right) \cdot (V_m \cdot \cos(\theta_m) + I_b \cdot x \cdot \sin(\theta_b)) + \dots\right. \\
& \left.(\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot V_m \cdot \sin(\theta_n)\right) + \dots \\
& \left(\left(V_n^2\right) \cdot (I_b \cdot x \cdot \cos(\theta_b)) + \left(\frac{\Delta Q_{Load} \cdot r - r \cdot Q_{Load}}{P_{Load} \cdot x} + \right) \cdot V_m \cdot \cos(\theta_n) + \right. \\
& \left.\left(V_n^2\right) \cdot (I_b \cdot r \cdot \sin(\theta_b)) + x(-\Delta Q_{Load} + Q_{Load}) \cdot V_m \cdot \sin(\theta_n) + \dots\right. \\
& \left.P \cdot r \cdot V_m \cdot \sin(\theta_n) - \left(V_n^2\right) \cdot (V_m \cdot r \cdot \sin(\theta_m))\right)
\end{aligned} \right)^2 \quad (A.15)
\end{aligned}$$

$$\begin{aligned}
V_{nec3} = & \left( \left( 1/V_n^4 \cdot \right. \right. \\
& \left. \left( \left( -\left(V_n^2\right) \cdot \left(I_b \cdot r \cdot \cos(\theta_b)\right) + \dots \right. \right. \right. \\
& \left. \left( \frac{\Delta Q_{Load} \cdot x - x \cdot Q_{Load}}{P_{Load} \cdot r} - \right) \cdot V_m \cdot \cos(\theta_n) + \dots \right. \\
& \left. \left( V_n^2 \right) \cdot \left( V_m \cdot \cos(\theta_m) + I_b \cdot x \cdot \sin(\theta_b) \right) + \dots \right. \\
& \left. \left( \Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x \right) \cdot V_m \cdot \sin(\theta_n) \right) \\
& + \dots \left. \right)^2 \left. \right)^{1/2} \\
2 \cdot & \left( \left( \left( V_n^2 \right) \cdot \left( I_b \cdot x \cdot \cos(\theta_b) \right) + \dots \right. \right. \\
& \left( \frac{\Delta Q_{Load} \cdot r - r \cdot Q_{Load}}{P_{Load} \cdot x} + \right) \cdot V_m \cdot \cos(\theta_n) + \\
& \left( V_n^2 \right) \cdot \left( I_b \cdot r \cdot \sin(\theta_b) \right) + \dots \\
& x \left( -\Delta Q_{Load} + Q_{Load} \right) \cdot V_m \cdot \sin(\theta_n) + \dots \\
& P \cdot r \cdot V_m \cdot \sin(\theta_n) - \left( V_n^2 \right) \cdot \left( V_m \cdot r \cdot \sin(\theta_m) \right) \left. \right)^2 \left. \right)^{1/2}
\end{aligned} \tag{A.16}$$

Also:

$$\frac{\partial V}{\partial \theta_n} = \frac{(\theta_{nec1} + \theta_{nec2})}{(\theta_{nec3})} \tag{A.17}$$

where

$$\theta_{nec1} = \left( 2 \cdot \left( \left( \frac{\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x}{\Delta Q_{Load} \cdot x - x \cdot Q_{Load} - P_{Load} \cdot r} \right) \cdot V_m \cdot \cos(\theta_n) - \dots \right) \right) \tag{A.18}$$

$$\theta_{nec2} = \left( \begin{array}{l} 2 \cdot \left( \begin{array}{l} x \cdot (-\Delta Q_{Load} + Q_{Load}) \cdot V_m \cdot \cos(\theta_n) + \dots \\ P_{Load} \cdot r \cdot V_m \cos(\theta_n) - \dots \\ (\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot V_m \cdot \sin(\theta_n) \end{array} \right) \cdot \\ \left( \begin{array}{l} (V_n^2) \cdot (I_b \cdot x \cdot \cos(\theta_b)) + \dots \\ (\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot V_m \cdot \cos(\theta_n) + \dots \\ (V_n^2) \cdot (I_b \cdot r \cdot \sin(\theta_b)) + \dots \\ x \cdot (-\Delta Q_{Load} + Q_{Load}) \cdot V_m \cdot \sin(\theta_n) + \dots \\ P_{Load} \cdot r \cdot V_m \sin(\theta_n) - \dots \\ (V_n^2) \cdot V_m \cdot \sin(\theta_m) \end{array} \right) \end{array} \right) \quad (A.19)$$

$$\theta_{nec3} = \left( \begin{array}{l} (1/(V_n^4)) \cdot \\ \left( \begin{array}{l} - (V_n^2) \cdot (I_b \cdot r \cdot \cos(\theta_b)) + \dots \\ \left( \begin{array}{l} \Delta Q_{Load} \cdot x - x \cdot Q_{Load} - \\ P_{Load} \cdot r \end{array} \right) \cdot V_m \cdot \cos(\theta_n) + \dots \\ (V_n^2) \cdot (V_m \cdot \cos(\theta_m) + I_b \cdot x \cdot \sin(\theta_b)) + \dots \\ (\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot V_m \cdot \sin(\theta_n) \end{array} \right)^2 + \dots \\ 2 \cdot (V_n^4) \cdot \left( \begin{array}{l} (V_n^2) \cdot (I_b \cdot x \cdot \cos(\theta_b)) + \dots \\ \left( \begin{array}{l} \Delta Q_{Load} \cdot r - r \cdot Q_{Load} + \\ P_{Load} \cdot x \end{array} \right) \cdot V_m \cdot \cos(\theta_n) + \\ (V_n^2) \cdot (I_b \cdot r \cdot \sin(\theta_b)) + \dots \\ x \cdot (-\Delta Q_{Load} + Q_{Load}) \cdot V_m \cdot \sin(\theta_n) + \dots \\ P \cdot r \cdot V_m \cdot \sin(\theta_n) - (V_n^2) \cdot (V_m \cdot r \cdot \sin(\theta_m)) \end{array} \right)^2 \end{array} \right)^{1/2} \quad (A.20)$$

Also:

$$\frac{\partial V}{\partial V_m} = \frac{(Vmec1)}{(Vmec2)} \quad (A.21)$$

where

$$V_{mec1} = \left( 2 \cdot V_n \cdot \cos(\theta_m) \cdot \begin{pmatrix} -V_n \cdot I_b \cdot r \cdot \cos(\theta_b) + \dots \\ \left( \Delta Q_{Load} \cdot x - x \cdot Q_{Load} - \right. \\ \left. P_{Load} \cdot r \right) \cdot \cos(\theta_n) + \dots \\ V_n \cdot V_m \cdot \cos(\theta_m) + V_n \cdot I_b \cdot x \cdot \sin(\theta_b) + \dots \\ \left( (\Delta Q_{Load} - Q_{Load}) \cdot r + \right. \\ \left. P_{Load} \cdot x \right) \cdot \sin(\theta_n) \end{pmatrix} - \dots \right) \quad (A.22)$$

$$\left( 2 \cdot V_n \cdot \sin(\theta_m) \cdot \begin{pmatrix} -V_n \cdot I_b \cdot x \cdot \cos(\theta_b) + \dots \\ \left( \Delta Q_{Load} \cdot r - r \cdot Q_{Load} + \right. \\ \left. P_{Load} \cdot x \right) \cdot \cos(\theta_n) + \dots \\ V_n \cdot I_b \cdot r \cdot \sin(\theta_b) + x \cdot (-\Delta Q_{Load} + Q_{Load}) \sin(\theta_n) + \dots \\ (P_{Load} \cdot r) \cdot \sin(\theta_n) - V_n \cdot V_m \cdot \sin(\theta_m) \end{pmatrix} \right)$$

$$V_{mec2} =$$

$$2 \cdot (V_n^2) \cdot \left( \left( 1/(V_n^2) \right) \cdot \left( \begin{pmatrix} - (V_n) \cdot (I_b \cdot r \cdot \cos(\theta_b)) + \dots \\ \left( \Delta Q_{Load} \cdot x - x \cdot Q_{Load} - \right. \\ \left. P_{Load} \cdot r \right) \cdot \cos(\theta_n) + \dots \\ (V_n) \cdot (V_m \cdot \cos(\theta_m) + I_b \cdot x \cdot \sin(\theta_b)) + \dots \\ (\Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x) \cdot \sin(\theta_n) \end{pmatrix}^2 + \dots \right) \right)^{1/2} \quad (A.23)$$

Also:

$$\frac{\partial V}{\partial \theta_m} = \frac{(\theta_{mec1})}{(\theta_{mec2})} \quad (A.24)$$

where



$$\begin{aligned}
\theta_{mec1} = & \left( -2 \cdot V_n \cdot V_m \cdot \sin(\theta_m) \cdot \left( \begin{aligned} & \left( -V_n \cdot I_b \cdot r \cdot \cos(\theta_b) + \dots \right. \\ & \left. \left( \left( \Delta Q_{Load} - Q_{Load} \right) \cdot x - \right. \right. \\ & \left. \left. P_{Load} \cdot r \right) \cos(\theta_n) + \dots \right. \\ & V_n \cdot V_m \cdot \cos(\theta_m) + \dots \\ & V_n \cdot I_b \cdot x \cdot \sin(\theta_b) + \dots \\ & \left. \left( \left( \Delta Q_{Load} - Q_{Load} \right) \cdot r + \right. \right. \\ & \left. \left. P_{Load} \cdot x \right) \sin(\theta_n) \right) \right) - \dots \\ & \left( \begin{aligned} & \left( V_n \cdot I_b \cdot x \cdot \cos(\theta_b) + \dots \right. \\ & \left( \Delta Q_{Load} \cdot r - r \cdot Q_{Load} + \right. \\ & \left. P_{Load} \cdot x \right) \cos(\theta_n) + \dots \\ & V_n \cdot I_b \cdot r \cdot \sin(\theta_b) + \dots \\ & x \cdot (-\Delta Q_{Load} + Q_{Load}) \sin(\theta_n) + \dots \\ & (P_{Load} \cdot r) \sin(\theta_n) - V_n \cdot V_m \cdot \sin(\theta_m) \end{aligned} \right) \right) \end{aligned} \quad (A.25)
\end{aligned}$$

$$\begin{aligned}
\theta_{mec2} = & \left( \left( 1/(V_n^2) \right) \cdot \left( \begin{aligned} & \left( -\left( V_n \right) \cdot \left( I_b \cdot r \cdot \cos(\theta_b) \right) + \dots \right. \\ & \left( \Delta Q_{Load} \cdot x - x \cdot Q_{Load} - \right. \\ & \left. P_{Load} \cdot r \right) \cos(\theta_n) + \dots \\ & \left( V_n \right) \cdot \left( V_m \cdot \cos(\theta_m) + I_b \cdot x \cdot \sin(\theta_b) \right) + \dots \\ & \left( \Delta Q_{Load} \cdot r - r \cdot Q_{Load} + P_{Load} \cdot x \right) \sin(\theta_n) \end{aligned} \right)^2 + \dots \right)^{1/2} \\
& 2 \cdot (V_n^2) \cdot \left( \begin{aligned} & \left( \left( V_n \right) \cdot \left( I_b \cdot x \cdot \cos(\theta_b) \right) + \dots \right. \\ & \left( \Delta Q_{Load} \cdot r - r \cdot Q_{Load} + \right. \\ & \left. P_{Load} \cdot x \right) \cos(\theta_n) + \\ & \left( V_n \right) \cdot \left( I_b \cdot r \cdot \sin(\theta_b) \right) + \dots \\ & x \cdot (-\Delta Q_{Load} + Q_{Load}) \sin(\theta_n) + \dots \\ & P \cdot r \cdot \sin(\theta_n) - \left( V_n \right) \cdot \left( V_m \cdot \sin(\theta_m) \right) \end{aligned} \right)^2 \end{aligned} \quad (A.26)
\end{aligned}$$

## APPENDIX B TEN, THIRTY-FOUR AND SIXTY-NINE DISTRIBUTION FEEDERS DATA

Figure B.1 and Tables B.1 and B.2 show Chapter 3 ten-bus feeder data.

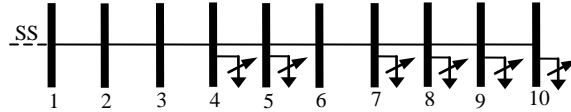


Figure B.1: Ten-bus feeder in Chapter 3

Table B.1: Resistance and reactance for 10-bus system

Bus From	Bus To	Resistance in Ohms ( $\Omega$ )	Reactance in Ohms ( $\Omega$ )
1	2	0.1233	0.4127
2	3	0.014	0.6051
3	4	0.7463	1.205
4	5	0.6984	0.6084
5	6	1.9831	1.7276
6	7	0.9053	0.7886
7	8	2.0552	1.164
8	9	4.7953	2.716
9	10	5.3434	3.0264

Table B.2: 10-bus feeder bus load

Bus	Real Power (P) in kW	Reactive Power (Q) In kVARs
1	0	0
2	1840	460
3	980	340
4	1790	446
5	1598	1840
6	1610	600
7	780	110
8	1150	60
9	980	130
10	1640	200

Figure B.2 and Tables B.3 and B.4 show Chapter 3 34-bus feeder data.

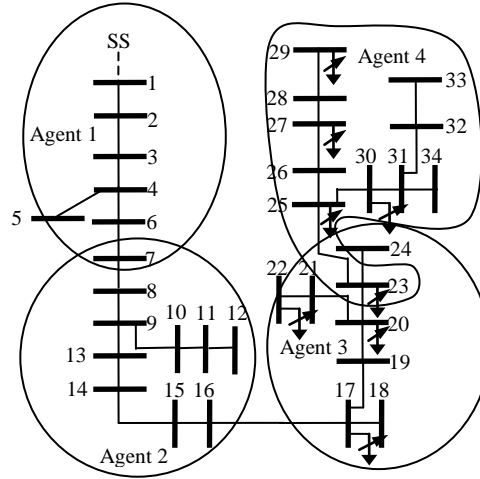


Figure B.2: IEEE modified 34 bus feeder in Chapter 3

Table B.3: Resistance and reactance for 34-bus system for Chapter 3

Bus From	Bus To	Resistance in Ohms ( $\Omega$ )	Reactance in Ohms ( $\Omega$ )
1	2	0.23406	0.20205
2	3	0.15695	0.13548
3	4	2.9239	2.5241
4	5	1.8543	0.99404
4	6	3.402	2.9368
6	7	2.6971	2.3283
7	8	0.000907	0.000783
8	9	0.028123	0.024277
9	10	0.15513	0.13392
10	11	15.383	8.2466
11	12	4.3898	2.3532
9	13	0.92625	0.79959
13	14	0.27488	0.23729
14	15	0.076205	0.065784
15	16	1.8543	1.6007
16	17	0.047174	0.040723
17	18	7.4537	3.9957
17	19	11.767	6.3078
19	20	0.000907	0.000783
20	21	0.001361	0.001175

Table B.3: Continued

21	22	0.958	0.827
20	23	0.44453	0.38374
23	24	0.14697	0.12687
23	25	0.5289	0.45657
25	26	0.025402	0.021928
26	27	0.12247	0.10572
27	28	0.33022	0.28506
28	29	0.048081	0.041507
25	30	0.18325	0.1582
30	31	0.24313	0.20988
31	32	0.025402	0.021928
32	33	0.4409	0.38061
31	34	0.078019	0.06735

Table B.4: 34-bus feeder bus load for Chapter 3

Bus	Real Power (P) in kW	Reactive Power (Q) In kVARs
1	0	0
2	30	15
3	30	15
4	16	8
5	16	8
6	10	5
7	10	5
8	4	2
9	5	2
10	34	17
11	34	17
12	135	70
13	5	2
14	40	20
15	7	3
16	10	5
17	4	2
18	4	2
19	7	3
20	7	3
21	17	12
22	150	75
23	2	1
24	2	1
25	13	7
26	9	5
27	135	105
28	25	12
29	20	16
30	110	55
31	30	15
32	28	14
33	28	14
34	22	11

Figure B.3 and Tables B.5 and B.6 show Chapter 4 34-bus feeder data.

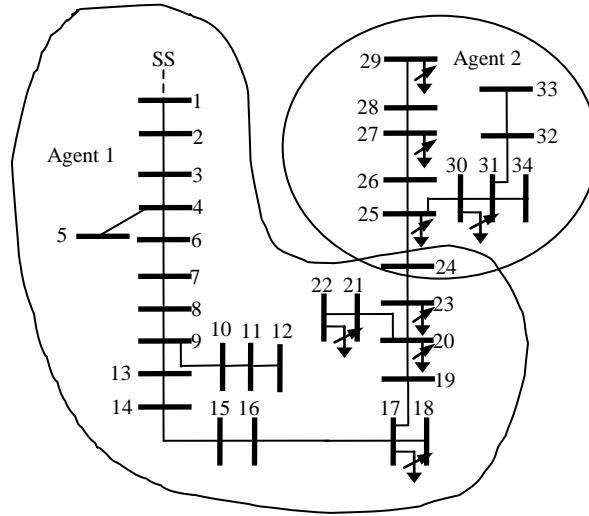


Figure B.3: IEEE modified 34 bus feeder in Chapter 4

Table B.5: Resistance and reactance for 34-bus system for Chapter 4

Bus From	Bus To	Resistance in Ohms ( $\Omega$ )	Reactance in Ohms ( $\Omega$ )
1	2	0.23406	0.20205
2	3	0.15695	0.13548
3	4	2.9239	2.5241
4	5	1.8543	0.99404
4	6	3.402	2.9368
6	7	2.6971	2.3283
7	8	0.000907	0.000783
8	9	0.028123	0.024277
9	10	0.15513	0.13392
10	11	15.383	8.2466
11	12	4.3898	2.3532
9	13	0.92625	0.79959
13	14	0.27488	0.23729
14	15	0.076205	0.065784
15	16	1.8543	1.6007
16	17	0.047174	0.040723
17	18	7.4537	3.9957
17	19	11.767	6.3078
19	20	0.000907	0.000783
20	21	0.001361	0.001175
21	22	0.958	0.827
20	23	0.44453	0.38374

Table B.5: Continued

23	24	0.14697	0.12687
24	25	0.5289	0.45657
25	26	0.025402	0.021928
26	27	0.12247	0.10572
27	28	0.33022	0.28506
28	29	0.048081	0.041507
25	30	0.18325	0.1582
30	31	0.24313	0.20988
31	32	0.025402	0.021928
32	33	0.4409	0.38061
31	34	0.078019	0.06735

Table B.6: 34-bus feeder bus load for Chapter 4

Bus	Real Power (P) in kW	Reactive Power (Q) In kVARs
1	0	0
2	30	15
3	30	15
4	16	8
5	16	8
6	10	5
7	10	5
8	4	2
9	5	2
10	34	17
11	34	17
12	135	70
13	5	2
14	40	20
15	7	3
16	10	5
17	4	2
18	4	2
19	7	3
20	7	3
21	17	12
22	150	75
23	2	1
24	0	0
25	13	7
26	9	5
27	135	105
28	25	12
29	20	16
30	110	55
31	30	15
32	28	14
33	28	14
34	22	11

Figure B.4 and Tables B.7 and B.8 show Chapter 5 34-bus feeder data.

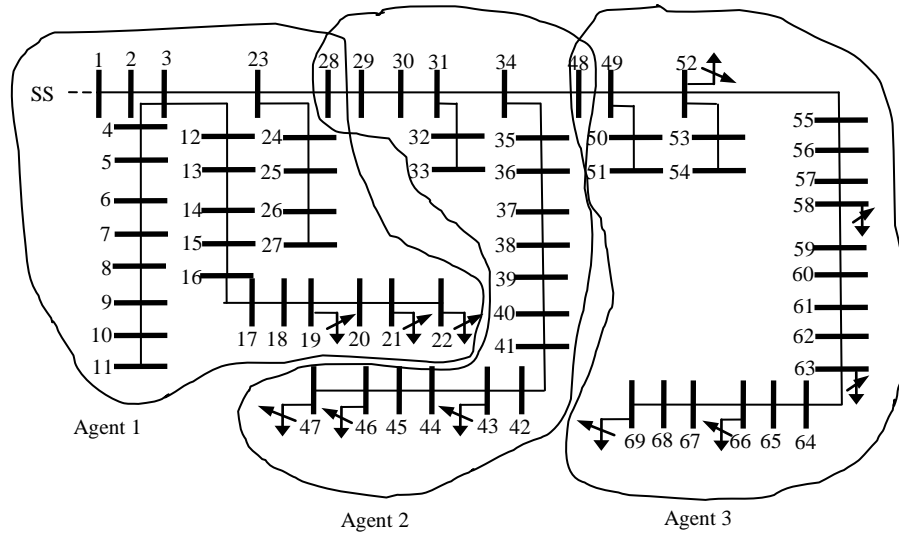


Figure B.4: 69-Bus feeder in Chapter 4

Table B.7: Resistance and reactance for 69-bus system for Chapter 4

Bus From	Bus To	Resistance in Ohms ( $\Omega$ )	Reactance in Ohms ( $\Omega$ )
1	2	0.0005	0.0012
2	3	0.0005	0.0012
3	4	0.0044	0.0108
4	5	0.064	0.1565
5	6	0.3978	0.1315
6	7	0.0702	0.0232
7	8	0.351	0.116
8	9	0.839	0.2816
9	10	1.708	0.5646
10	11	1.474	0.4873
3	12	0.0044	0.0108
12	13	0.064	0.1565
13	14	0.1053	0.123
14	15	0.0304	0.0355
15	16	0.0018	0.0021
16	17	0.7283	0.8509
17	18	0.31	0.3623



Table B.7: Continued

18	19	0.041	0.0478
19	20	0.0092	0.0116
20	21	0.1089	0.1373
21	22	0.0009	0.0012
3	23	0.0015	0.0036
23	24	0.0034	0.0084
24	25	0.0851	0.2083
25	26	0.2898	0.7091
26	27	0.0822	0.2011
23	28	0.0251	0.0294
28	29	0.366	0.1864
29	30	0.3811	0.1941
30	31	0.0922	0.047
31	32	0.0928	0.0473
32	33	0.3319	0.1114
31	34	0.0493	0.0251
34	35	0.174	0.0886
35	36	0.203	0.1034
36	37	0.2842	0.1447
37	38	0.2813	0.1433
38	39	1.59	0.5337
39	40	0.7837	0.263
40	41	0.3042	0.1006
41	42	0.3861	0.1172
42	43	0.5075	0.2585
43	44	0.0974	0.0496
44	45	0.145	0.0738
45	46	0.7105	0.3619
46	47	1.041	0.5302
34	48	0.819	0.2707
48	49	0.1872	0.0619
49	50	0.2012	0.0611
50	51	0.0047	0.0014
49	52	0.7114	0.2351
52	53	0.7394	0.2444
53	54	0.0047	0.0016
52	55	1.03	0.34
55	56	1.044	0.345
56	57	1.058	0.3496
57	58	0.1966	0.065
58	59	0.3744	0.1238
59	60	0.0047	0.0016
60	61	0.3276	0.1083
61	62	0.2106	0.0696
62	63	0.3416	0.1129
63	64	0.014	0.0046
64	65	0.1591	0.0526
65	66	0.3463	0.1145
66	67	0.7488	0.2475
67	68	0.3089	0.1021
68	69	0.1732	0.0572

Table B.8: 69-bus feeder bus load for Chapter 4

Bus	Real Power (P) in kW	Reactive Power (Q) In kVARs
1	0	0
2	0	0
3	0	0
4	26	18.6
5	26	18.6
6	0	0
7	0	0
8	0	0
9	14	10
10	19.5	14
11	6	4
12	1126	1118.6
13	1126	1118.6
14	0	0
15	1124	1117
16	124	117
17	111.2	111
18	0	0
19	216	114.3
20	0	0
21	1139.2	1126.3
22	2139.2	1126.3
23	0	0
24	0	0
25	79	56.4
26	384.7	274.5
27	384.7	274.5
28	0	0
29	2.6	2.2
30	40.4	30
31	75	54
32	40.5	28.3
33	3.6	2.7
34	30	22
35	4.35	3.5
36	26.4	19
37	24	17.2
38	0	0
39	0	0
40	0	0
41	100	72
42	0	0
43	844	888
44	32	23
45	0	0
46	227	162
47	59	42

Table B.8: Continued

48	0	0
49	173	123
50	18	13
51	18	13
52	145	104
53	28	20
54	28	20
55	8	5.5
56	8	5.5
57	0	0
58	45.5	30
59	60	35
60	110	35
61	0	0
62	1	0.6
63	214	181
64	5.3	3.5
65	0	0
66	128	120
67	0	0
68	14	10
69	114	110

## **AUTHOR'S BIOGRAPHY**

Angel A. Aquino-Lugo was born in Bayamón, Puerto Rico. He received the B.S. and M.S. degrees in electrical engineering from the University of Puerto Rico at Mayagüez in 2004 and 2006 respectively. He joined the Power and Energy Systems group at the University of Illinois at Urbana-Champaign in August 2006. His research interests include power system analysis, decentralized optimization and smart grid control.