

ANALYSIS OF VOLTAGE REGULATION AND NETWORK SUPPORT TECHNOLOGIES

Frans Jacobus Rossouw



Thesis presented in partial fulfilment of the requirements for the degree of Master of Science of Engineering at the University of Stellenbosch.

Supervisor: Dr H.J. Beukes

December 2000

DECLARATION

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

SUMMARY

Recent advances in semiconductor device development pushed a large number of network devices onto the market. These devices can solve network problems more effectively and economically than ever before. Network planners need tools to analyse and implement such devices to help solve the largest network problem in South Africa: voltage regulation. Rural networks experience the majority of voltage-regulation problems in South Africa. The networks are long sub-transmission and reticulation networks and are modelled by two generic networks, namely a radial network and a two-source ring network. The equations describing voltage regulation for the generic networks are developed and implemented in PSAT, a software analysis tool. The voltage regulation for two case studies that represent the two generic networks are analysed. Four generic network devices are defined and various control methods for these devices are developed to solve the network problem. The aim of PSAT is to help the network planner to quickly evaluate a number of possible solutions and to choose the best solution for further studies. This is demonstrated with the aid of the case studies. PSAT provides a sturdy platform on which future developments, such as stability analyses, can be built. However, PSAT can already function as a stand-alone analysis tool to solve voltage regulation as a network problem.

OPSOMMING

Onlangse vooruitgang in halfgeleier ontwikkeling het 'n groot aantal netwerktoestelle op die mark geplaas. Hierdie toestelle kan netwerk probleme doeltreffender en meer ekonomies oplos as ooit vantevore. 'n Behoefte aan 'n pakket wat netwerkbeplanners in staat stel om die netwerktoestelle te analiseer, is geïdentifiseer. So 'n pakket sal hulle help om die vernaamste netwerkprobleem in Suid-Afrika, nl. spanningsregulasie, op te los. Die oorgrote meerderheid spanningsregulasie probleme word op die platteland ondervind. Plattelandse netwerke word gekenmerk deur lang sub-transmissie en retikulasie netwerke. Hierdie netwerke word met behulp van twee generiese netwerke gemodelleer. 'n Radiale netwerk en 'n dubbelbron ring-netwerk word aangewend om enige plattelandse netwerk te analiseer. Vergelykings is vir spanningsanalise ontwikkel en in PSAT, 'n analitiese sagteware pakket, geïmplementeer. Twee gevallestudies is gedoen om die twee netwerke afsonderlik voor te stel en die vergelykings van PSAT te evalueer. Alle netwerktoestelle is in een van vier generiese kategorieë geklassifiseer. Modelle is vir elk van die kategorieë ontwikkel vir spanningsregulasie analise. Die doel van PSAT is om die netwerk beplanner te help om vinnig en effektief soveel moontlik opsies te ondersoek as oplossings vir 'n spesifieke netwerk probleem. PSAT is reeds 'n alleenstaande pakket wat in die toekoms uitgebrei sal word om na die analise van stabiliteitsprobleme te kyk.

TABLE OF CONTENTS

LIST OF FIGURES	VI
LIST OF TABLES	XI
LIST OF ABBREVIATIONS	XII
ACKNOWLEDGEMENTS	XIII
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 METHOD OF ANALYSIS	9
2.1 INTRODUCTION	10
2.2 NETWORK CONFIGURATION AND PARAMETERS.....	12
2.2.1 <i>Network configuration</i>	12
2.2.2 <i>Network parameters</i>	14
2.3 NETWORK PROBLEMS	16
2.4 FIRST ASSESSMENT AND SOLUTIONS TO NETWORK PROBLEMS	17
2.4.1 <i>Network device configuration</i>	17
2.4.2 <i>Network device technology</i>	18
2.4.3 <i>Solution</i>	19
CHAPTER 3 VOLTAGE REGULATION AND VOLTAGE REGULATORS	26
3.1 INTRODUCTION	27
3.2 VOLTAGE REGULATION	28
3.2.1 <i>Simple circuit diagram</i>	28
3.2.2 <i>Obtaining network parameters</i>	29
3.2.3 <i>Simplified network</i>	31
3.2.4 <i>Cathedral Peak Hotel case study</i>	35
3.3 SHUNT VOLTAGE REGULATORS	42
3.3.1 <i>Shunt compensation with active power specified</i>	44
3.3.2 <i>Shunt compensation with reactive power specified</i>	50
3.3.3 <i>Minimum shunt compensator rating</i>	54
3.3.4 <i>Summary of shunt compensation</i>	58
3.4 SERIES VOLTAGE REGULATORS	60
3.4.1 <i>Series compensator with active power specified</i>	63
3.4.2 <i>Series compensator with reactive power specified</i>	67

3.4.3	<i>Minimum rating series compensator</i>	68
3.4.4	<i>Summary of series compensation</i>	72
3.5	SERIES-SHUNT COMPENSATION	73
3.6	IN-LINE COMPENSATION	81
3.7	NEW LINE	89
3.8	SUMMARY	94
3.8.1	<i>Equations</i>	94
3.8.2	<i>Cathedral Peak case study</i>	95
3.8.3	<i>Analysis tool</i>	97
CHAPTER 4	ZIMBANE CASE STUDY	98
4.1	INTRODUCTION	99
4.2	UNCOMPENSATED NETWORK	101
4.3	SHUNT COMPENSATORS	106
4.3.1	<i>Voltage regulation with an SVC</i>	106
4.3.2	<i>Minimum-rating shunt device</i>	109
4.4	SERIES COMPENSATORS	112
4.4.1	<i>Purely reactive series compensator</i>	112
4.4.2	<i>Minimum-rating series compensator</i>	113
4.5	SERIES-SHUNT COMPENSATION	114
4.6	IN-LINE COMPENSATION	117
4.7	SUMMARY OF THE ZIMBANE CASE STUDY	120
CHAPTER 5	FUTURE DEVELOPMENTS	122
CHAPTER 6	CONCLUSION	129
CHAPTER 7	REFERENCES	133
ADDENDUM A	CALCULATION OF THE IMPEDANCE OF 25 KM HARE LINE	137
ADDENDUM B	PROGRAM LISTINGS	140

LIST OF FIGURES

Fig. 1.1: PSAT internal layout	6
Fig. 2.1: Flow chart of quality of supply analysis	11
Fig. 2.2: Circuit diagram of extensive networks	12
Fig. 2.3: Equivalent circuit diagram, without approximations, for above networks.....	13
Fig. 2.4: Shunt compensator	20
Fig. 2.5: Series compensator	21
Fig. 2.6: Series-shunt compensator	21
Fig. 2.7: In-line compensator	21
Fig. 2.8: Input interface of PSAT	22
Fig. 2.9: Output interface of PSAT	22
Fig. 3.1: Circuit diagram for analysing the voltage regulation of V_{PCCR}	28
Fig. 3.2: Modelling of a distribution network.....	30
Fig. 3.3: Simplification of Fig. 3.1 through Thévenin and Norton transformations	32
Fig. 3.4: Equivalent circuit diagram for analysing the voltage regulation of V_{PCCR}	33
Fig. 3.5: Projected network feeding Cathedral Peak Hotel in 2005 and its equivalent circuit diagram	36
Fig. 3.6: Circuit diagram for the Cathedral Peak network.....	36
Fig. 3.7: Voltage regulation as a function of network position and voltage angle for the uncompensated Cathedral Peak network.....	38
Fig. 3.8: Voltage regulation as a function of network position for the uncompensated Cathedral Peak network at $\delta = 0$ rad	38
Fig. 3.9: Active and reactive power in the uncompensated Cathedral Peak network at $\delta = 0$ rad.....	39
Fig. 3.10: Current magnitude and angle in the uncompensated Cathedral Peak network.....	40
Fig. 3.11: Voltage and current vector diagrams for the uncompensated Cathedral Peak network.....	41
Fig. 3.12: Shunt voltage regulator	42
Fig. 3.13: Simplified shunt-compensated network.....	43
Fig. 3.14: Cathedral Peak network with a shunt compensator.....	46
Fig. 3.15: Voltage plots for the Cathedral Peak network under shunt compensation with no energy storage at $\delta = 0$ rad.....	47

Fig. 3.16: Voltage and current vector diagrams for Cathedral Peak network with shunt compensation without energy storage at $\delta = 0$ rad.....	47
Fig. 3.17: Current magnitude and angle for shunt compensation with $P_{Comp2} = 0$ p.u. at $\delta = 0$ rad.....	48
Fig. 3.18: Shunt compensator voltage and current for $P_{Comp2} = 0$	48
Fig. 3.19: Power transfer under shunt compensation with $P_{Comp2} = 0$	49
Fig. 3.20: Cathedral Peak network with a shunt compensator.....	52
Fig. 3.21: Voltage magnitude and angle under shunt compensation with $Q_{Comp2} = 0$	52
Fig. 3.22: Power transfer under shunt compensation with $Q_{Comp2} = 0$	53
Fig. 3.23: Current magnitude and angle in the shunt-compensated Cathedral Peak network.....	54
Fig. 3.24: Cathedral Peak network with a shunt compensator.....	55
Fig. 3.25: Voltage graphs for the Cathedral Peak network with a minimum shunt compensator.....	56
Fig. 3.26: Power transfer for minimum shunt compensator.....	56
Fig. 3.27: Current graphs for the Cathedral Peak network with minimum shunt compensator.....	57
Fig. 3.28: Current graphs for the minimum shunt compensator.....	57
Fig. 3.29: Power transfer for minimum shunt compensator without additional capacitor banks.....	58
Fig. 3.30: Series voltage regulator.....	60
Fig. 3.31: Simplified series-compensated network.....	61
Fig. 3.32: Cathedral Peak network with series compensation.....	64
Fig. 3.33: Voltage plots and voltage vector diagram at $\delta = 0$ rad for the Cathedral Peak network under series compensation with no energy storage.....	65
Fig. 3.34: V_{Comp1} for the Cathedral Peak network under series compensation with no energy storage.....	66
Fig. 3.35: Current magnitude and angle for series compensation with $P_{Comp1} = 0$	66
Fig. 3.36: Power transfer under series compensation with $P_{Comp1} = 0$	67
Fig. 3.37: Cathedral Peak network with a series compensator.....	69
Fig. 3.38: Series-compensator rating as a function of θ_{PCCR} and δ	70
Fig. 3.39: Voltage plots for the Cathedral Peak network with minimum series compensation.....	70

Fig. 3.40: Voltage and power plots for the series compensator of minimum rating	71
Fig. 3.41: Current magnitude and angle for minimum series compensation	71
Fig. 3.42: Power transfer under minimum series compensation	72
Fig. 3.43: Series-shunt voltage regulator	73
Fig. 3.44: Simplified network with a series-shunt-compensator	74
Fig. 3.45: Cathedral Peak network with a series-shunt compensator	76
Fig. 3.46: Voltage plots for the Cathedral Peak network under series-shunt compensation ...	77
Fig. 3.47: Voltage diagrams for the series compensator in the series-shunt compensated Cathedral Peak network	78
Fig. 3.48: Current magnitude and angle for series-shunt compensation	78
Fig. 3.49: Power transfer under series-shunt compensation	79
Fig. 3.50: Power transfer across the Cathedral Peak network under series-shunt compensation at $\delta = 0$ rad	79
Fig. 3.51: Power delivery by the separate series and shunt compensators in the Cathedral Peak	80
Fig. 3.52: In-line device as voltage regulator	81
Fig. 3.53: Simplified network with an in-line compensator	82
Fig. 3.54: Cathedral Peak network with an in-line compensator	84
Fig. 3.55: Voltage plots across the Cathedral Peak network with an in-line compensator for $\delta = 0$ rad	85
Fig. 3.56: Voltage plots across the Cathedral Peak network with an in-line compensator for $\delta = 0$ rad and reactive power injection of 2.51×10^{-3} p.u. at PCCS	86
Fig. 3.57: Voltage plots across the Cathedral Peak network with an in-line compensator and reactive power injection of 100×10^{-3} p.u. at PCCS	86
Fig. 3.58: Current magnitude and angle for in-line compensation	87
Fig. 3.59: Power transfer in the in-line compensated network	87
Fig. 3.60: Circuit diagram for the Cathedral Peak network with 33 kV injection from Driel	89
Fig. 3.61: Equivalent circuit diagram for analysing the voltage regulation of the Cathedral Peak network with 33 kV injection from Driel	90
Fig. 3.62: Voltage diagrams for the Cathedral Peak network with a new 33 kV line added from Driel	91
Fig. 3.63: $ V_{PCCR} $ for 33 kV and 11 kV transmission from Driel	91

Fig. 3.64: Current diagrams for the Cathedral Peak network with a new line added from Driel.....	92
Fig. 3.65: Power transfer across the Winterton-Driel network.....	92
Fig. 3.66: Power transfer across the Winterton-Driel network at $\delta=0.616$ rad and $\delta=2.38$ rad.....	93
Table 3.1: Summary of voltage regulator performance in the Cathedral Peak network.....	96
Fig. 4.1: Pembroke-Zimbane-Eros network.....	99
Fig. 4.2: Diagram of the simplified Pembroke-Zimbane network.....	102
Fig. 4.3: Voltage magnitude and angle across the Zimbane network.....	103
Fig. 4.4: Voltage profile at the PCC.....	103
Fig. 4.5: Active and reactive power dissipated across the Zimbane network.....	104
Fig. 4.6: Power vector diagram at $\delta=1.41$ rad.....	104
Fig. 4.7: Power vector diagram at $\delta=0$ rad.....	105
Fig. 4.8: Zimbane power system model with a shunt compensator.....	106
Fig. 4.9: Voltage magnitude and angle across the SVC-compensated Zimbane network.....	107
Fig. 4.10: Power requirements for the SVC.....	107
Fig. 4.11: Active and reactive power dissipated across the Zimbane network.....	108
Fig. 4.12: Load power for the Zimbane network equipped with an SVC.....	108
Fig. 4.13: Voltage graphs for minimum shunt compensation.....	109
Fig. 4.14: Power graphs for a minimum-rating shunt device.....	110
Fig. 4.15: Power graphs for the Zimbane network with a minimum-rating shunt device.....	110
Fig. 4.16: Sending-end power graphs for the Zimbane network.....	111
Fig. 4.17: Series-compensated Zimbane network.....	112
Fig. 4.18: Voltage graphs of the two solutions for a purely reactive series compensator.....	112
Fig. 4.19: Voltage graphs of the four solutions for a minimum rating series compensator.....	113
Fig. 4.20: Series-shunt-compensated Zimbane network.....	114
Fig. 4.21: Voltage graphs for the Zimbane network with a series-shunt compensator.....	114
Fig. 4.22: Voltage plots of the two solutions for a purely reactive series compensator.....	115
Fig. 4.23: Power transfer across the Zimbane network equipped with a series-shunt device.....	115
Fig. 4.24: Power required by the series-shunt device.....	116
Fig. 4.25: Power required by the series-shunt device vs. power required by the SVC.....	116
Fig. 4.26: Zimbane network with an in-line compensator.....	117

Fig. 4.27: Voltage graphs for the Zimbane network with an in-line compensator	118
Fig. 4.28: Voltage graphs at $\delta=0$ rad for the Zimbane network with an in-line compensator	118
Fig. 4.29: Power transfer across the Zimbane network equipped with an in-line compensator	119
Table 4.1: Summary of voltage regulator performance in the Zimbane network at $\delta=0$ rad.....	120
Fig. 5.1: Voltages across the network for a shunt compensator implemented at the receiving-end	124
Fig. 5.2: Power transfer for a shunt compensator implemented at the receiving end	124
Fig. 5.3: Power and power factor for a shunt compensator implemented at the receiving end	125
Fig. 5.4: Equivalent circuit diagram, without approximations, for radial networks	126
Fig. 5.5: Voltage harmonics at Refinery substation	127
Fig. 6.1: Inputs and outputs of PSAT voltage-regulation engine.....	130
Fig. A.1: Pole structure of a typical 33 kV Hare line	138

LIST OF TABLES

Table 3.1: Summary of voltage regulator performance in the Cathedral Peak network.....	96
Table 4.1: Summary of voltage regulator performance in the Zimbane network at $\delta = 0 \text{ rad}$	120

LIST OF ABBREVIATIONS

δ	phase angle of the sending-end voltage
ac	alternating current
ACSR	Aluminium Conductor Steel Reinforced
dc	direct current
DVR	Dynamic Voltage Restorer
FACTS	Flexible ac Transmission Systems
GMR	geometric mean radius
GUI	graphical user interface
HVDC	High Voltage dc (line)
NER	National Electricity Regulator
PCC	point of common coupling
PCCL	point of common coupling at load
PCCR	point of common coupling at receiving end
PCCS	point of common coupling at sending end
PSAT	Power System Analysis Tool
SVC	Static Var Compensator
UPFC	Unified Power-Flow Controller
USE	Universal Semiconductor Electrification (device)

ACKNOWLEDGEMENTS

I would like to convey my gratitude to the people and institutions that made this project and thesis possible:

Dr Johan Beukes for his invaluable support and guidance;
ESKOM and PMC for their financial support.

CHAPTER 1 INTRODUCTION

Traditionally new lines or capacitors for reactive power compensation were the only methods available to solve network problems. However, many commercial network devices are available with which network problems, including voltage regulation, can be addressed more effectively, both in technical and financial terms. In South Africa voltage regulation is the most widely encountered network problem due to the extensive rural networks.

Many new options arose for semiconductor applications on power systems. The advent of new technologies such as high-power thyristors and transistors made network devices a reality. Hingorani called such devices on transmission level FACTS (Flexible AC Transmission System) devices [A1], [B1], [B2]. At distribution level, they are referred to as custom power devices [A2]. The concept of SETS was introduced for electrification applications [A3].

Utility planners are frequently reluctant to integrate network support technologies with the planning process, mainly due to a lack of adequate tools. A need was therefore identified for an analysis tool that models networks and support technologies in a manner known to power system engineers, i.e. with power transfer, voltage regulation and current graphs as well as vector diagrams [A4]. The two most important characteristics of this tool must be:

- To assist in the understanding of network problems, network technologies and their interaction;
- To provide an analysis tool for first assessment in the network problem-solving process, i.e. to give an indication of the required device configuration and rating.

Such an analysis tool was developed under the name PSAT (Power System Analysis Tool). With PSAT, network planners can be trained in the aspects of new technology and, with the necessary knowledge, implement innovative network technologies even on a proactive basis. Network problems in new networks can be identified and solved in the design phase, resulting in the construction of an already compensated network. This can result in extensive cost benefits as post-construction alterations to the network are eliminated.

In practice networks are classified into three categories according to the voltage level, namely transmission, sub-transmission and reticulation networks. Transmission networks operate at voltage levels from 220 kV to 765 kV. Sub-transmission voltage levels are 66 kV and 132 kV, while 11 kV, 22 kV and 33 kV networks are classified as reticulation networks. A further classification is made in terms of network configurations. Radial networks feed one or

more loads from a single, stiff voltage source. Ring or mesh networks feed one or more loads from more than one source, which are not necessarily stiff compared to the rest of the network.

In South Africa all transmission networks are mesh networks. The line impedance is mainly determined by the line reactance, as the line resistance is very low relative to the reactance. A widely used approximation is to ignore losses. Based on the small line resistance, this valid approximation greatly simplifies calculations. Due to a stiff transmission network, voltage regulation is generally not a problem on transmission level.

Sub-transmission networks can be either radial or mesh networks (ring feeders). Mesh networks are used in well-developed areas on sub-transmission level. These stiff networks do not experience voltage regulation problems and natural upgrades are determined by thermal current limits. In rural areas, radial sub-transmission networks are used to feed isolated loads over long distances; 66 kV and 132 kV radial lines can transport up to 80 MVA. Voltage regulation is often a problem at the end of these lines due to load growth, both along and at the end of the line. Long ring-fed lines experience similar problems.

A special case is a distributed generator at the end of a radial line. This situation is encountered increasingly with utilities tending to move away from large-scale energy production towards dispersed generation such as small-scale hydro-generation, solar dishes, gas turbines, diesel generators and fuel cells. The benefits to the utility are:

- Reduced voltage drops due to reduced peak currents
- Reactive power support
- Network reliability enhancement
- Deferral of system upgrades
- Reduced capital risk.

Reticulation networks are mainly radial networks, but in densely populated areas ring networks are used. The ring networks stretch over short distances and are often heavily loaded. Voltage regulation is, however, not a problem as the thermal current limit restricts power transfer levels before voltage regulation is affected. In rural areas long radial reticulation networks are stretched to supply ever-increasing loads. Voltage regulation often

is a problem on these networks. Smaller rating distributed generators are increasingly found on radial reticulation networks.

Voltage regulation is therefore mainly a problem in two generic types of networks:

- Long radial lines that feed rural networks
- Mesh networks with two stiff voltage sources.

The latter includes ring-fed rural networks and distributed generators connected to the end of radial lines. The two-source mesh networks are therefore network-to-network or network-to-distributed generation systems. Radial networks are a special case of mesh networks where there is only one voltage source and are therefore network-to-customer systems. It can thus be said that the three main parties in modern power systems are the network (the utility), the customer and distributed generators. The voltage-regulation equations for PSAT are developed to analyse the above two network types.

It is important to note the difference between PSAT and other simulation packages for power systems. The South African utility company, Eskom, uses mostly two power system simulation packages, namely PSSE and Reticmaster. PSSE can be used to simulate any network, whether it is radial or meshed. It requires detail models of all network components and can only simulate one specific network situation at a time. Reticmaster requires less detailed models, but are restricted to radial networks. Limited support is provided for ring networks. Both packages solve the networks iteratively. Quick evaluation of a host of different solutions to a specific network problem is not possible with either package.

PSAT is used for the first assessment of a network problem by evaluating all possible solutions. One or two best solutions are identified and the above packages can then be used to apply these solutions. This work is set apart from other analyses in that no approximations are made in the development of the analytical equations. Sub-transmission and reticulation networks are lossy and cannot adequately be modelled by traditional methods based on line reactance only. The resistive and reactive components of sub-transmission line impedances are of the same order. No approximations can therefore be made in terms of line impedances. Reticulation networks are mainly resistive, but the reactances cannot be ignored for accurate analyses. PSAT solves all networks analytically and does not employ iterative procedures. For PSAT to be able to handle all network situations and support technologies, the equations must be accurate for all voltage angles and lossy networks and loads. As PSAT is used for

first assessment of voltage regulation, it is assumed that the network voltages and currents are balanced. Single-line diagrams and voltage, current and power graphs portray therefore all three (balanced) phases of the network.

The aim of PSAT is, for a specific network situation, to assist in the selection of:

1. The configuration of the network device
2. The control method of the network device
3. The technology of the network device
4. The rating of the network device and sub-components.

The technology of the network device is determined by the need for active power compensation, which is directly linked to the control method of the network device. A combination of the device configuration and control method determines the device rating for a specific network situation.

After determining the three aspects above with the aid of PSAT, a detailed study should be conducted in traditional simulation packages such as PSSE or Reticmaster. This study should investigate issues such as unbalance, protection, and other system aspects.

Network devices aimed at sub-transmission and reticulation networks are on the forefront of network technologies. Distributed generators are included as network devices under the banner of support technologies because they are normally applied to support existing networks. Network devices can increase the stability margin to such an extent that the network can be operated at much higher voltage angles than before. Uncompensated networks are typically operated at voltage angles less than 0.7 rad to maintain an adequate first-swing stability margin. PSAT models all networks with a voltage angle ranging from 0 to π rad. Emphasizing the training capability of PSAT, this also aids comprehension in network behaviour for larger voltage angles. These characteristics and abilities of PSAT can play a crucial role in extending the South African network. Approximately 50% of the country is still not electrified and cost effective electrification are therefore one of Eskom's main concerns.

PSAT is aimed at voltage regulation, but future developments will include the full quality of supply spectrum, e.g. harmonics, dips and interruptions. These quality of supply issues can

be modelled by applying the principles for voltage regulation successively for sub-cycle time steps.

This thesis aims to document the development and practical application of the voltage regulation engine of PSAT. The internal layout of PSAT is shown in Fig. 1.1. The network parameters are obtained from either practical measurements or data from other simulation packages, e.g. Reticmaster or PSSE.

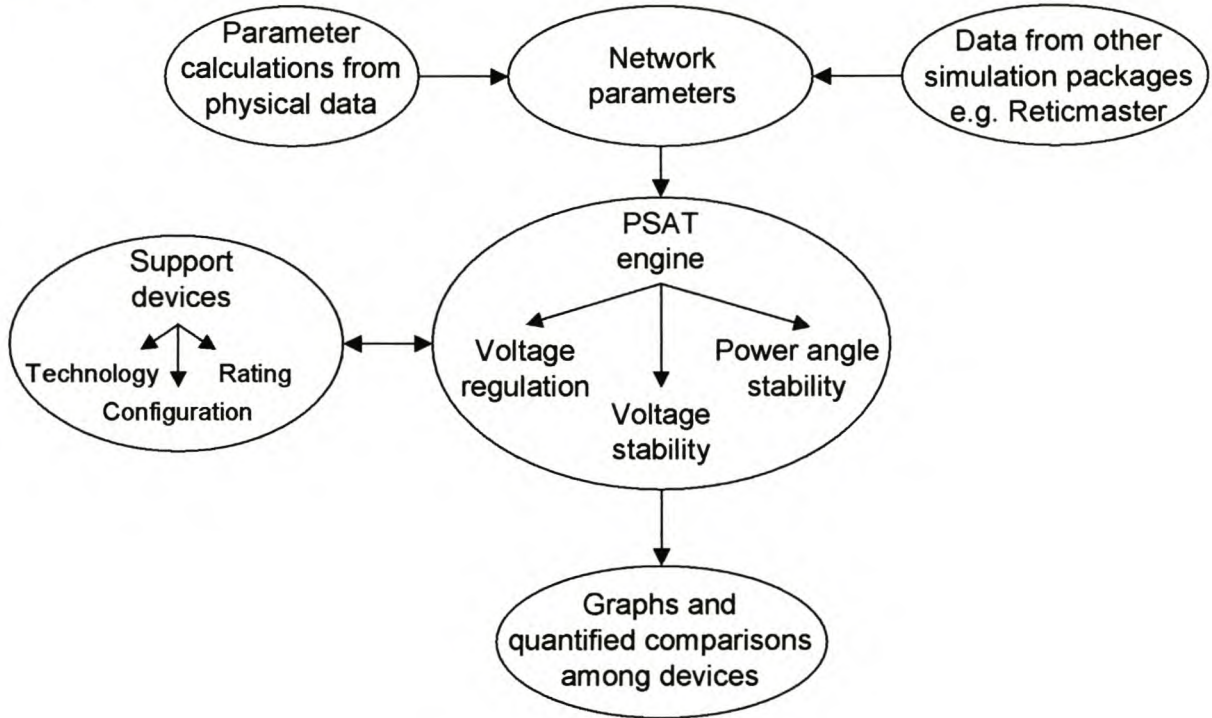


Fig. 1.1: PSAT internal layout

PSAT will eventually perform voltage regulation, voltage stability and basic power angle stability analyses. The scope of this thesis is restricted to the voltage regulation engine. The stability engines will be included in future versions of PSAT and will employ most of the work already developed for the voltage regulation part. The voltage regulation engine allows voltage regulation analyses to be performed quickly for all available support technologies. The equations derived in this thesis can easily be adapted for the stability engines. The method of determining the required device rating can be specified according to the desired control algorithm. A device of minimum rating can be implemented or a device with limited active or reactive power ratings can be used. This enables the network planner to conserve limited energy reserves or implement a device that delivers purely reactive power, e.g. capacitor banks.

In the development of PSAT, the following contributions were made:

1. The process of determining the best option of network compensation was documented extensively on a high level;
2. A standard network to represent any two-source mesh network without any approximations was developed;
3. Equations were developed for solving the standard network compensated by one of the four generic devices without any approximations;
4. Three-dimensional graphs were created to represent voltage, current and power across the network for all voltage angles.

An integrated approach is followed in the presentation of this thesis to emphasise the practical application of the theory developed. This deviates from the traditional presentation of a thesis where the theory is developed in one chapter, the implementation of the theory is dealt with in a following chapter and the final chapter presents the practical results. An integrated approach links the theory more effectively with the practical world and endless pages of tedious theoretical work are avoided. The reader is also gradually trained to interpret and apply the developed three-dimensional graphs of voltage, current and power. It must, however, be emphasised that the theoretical work is still the fundamental academic contribution of this thesis. Two case studies are presented to show the application of PSAT for the two network configurations mentioned earlier. The Cathedral Peak case study is a radial network and presented as an integral part of the development of the equations. With only one voltage source in the network, the voltage, current and power graphs are easier to understand. The developed equations are also easier to verify with the radial network results. The second case study deals with the Zimbane network and involves Zimbane substation fed from two networks. A recently installed static Var Compensator (SVC) and a hydro-generator near the substation can be used for voltage regulation. The graphs of this two-source network are more complicated and portray the abilities of PSAT more extensively.

Chapter 2 discusses the process of determining the best option for network compensation on a high level. Quality of supply and the influence of network configuration and parameters receive special attention. The focus falls mainly on voltage regulation. Network compensators are discussed and categorised as four generic devices, namely shunt, series, series-shunt and in-line devices. Chapter 2 is concluded with a discussion of the PSAT interface, its features and its development.

Chapter 3 contains the most important benefits of this work. The chapter starts with the development of an equivalent network, without any approximations, to model voltage regulation on any two-source mesh network. The equations for voltage regulation of in an uncompensated network are developed and then the Cathedral Peak case study is introduced. All developed equations are implemented in PSAT. Voltage regulation on the Cathedral Peak network is subsequently analysed with PSAT to demonstrate both the validity and the practical application of the developed models and equations. Critical evaluation of the resulting voltage, current and power graphs also demonstrates the tutoring ability of PSAT. Chapter 3 continues the integrated approach by developing the equations and analysing the Cathedral Peak case study for each of the four generic devices. The chapter is concluded with the analysis of a traditional method of solving a voltage-regulation problem, i.e. building a new line.

The Zimbane case study is presented in Chapter 4. Firstly, the uncompensated network is analysed to determine the extent of the voltage-regulation problem. Subsequently the respective compensators are implemented to determine the best solution to the network problem. In this process both the practical application and tutoring abilities of PSAT are demonstrated.

Chapter 5 discusses future developments in PSAT. Methods to deal with harmonics, flicker, dips, interruptions and unbalance are suggested. Further control strategies are also discussed. Mentioning the incorporation of voltage stability and financial modelling in PSAT concludes the chapter.

The thesis is concluded in Chapter 6 with a summary of the theoretical and practical results obtained with PSAT.

CHAPTER 2 METHOD OF ANALYSIS

2.1 INTRODUCTION

2.2 NETWORK CONFIGURATION AND PARAMETERS

2.2.1 Network configuration

2.2.2 Network parameters

2.3 NETWORK PROBLEMS

2.4 FIRST ASSESSMENT AND SOLUTIONS TO NETWORK PROBLEM

2.4.1 Network device configuration

2.4.2 Network device technology

2.4.3 Solution

2.1 INTRODUCTION

This chapter outlines the process of solving a network problem. The network configuration and parameters are discussed first after which section 2.3 deals with various network problems. In section 2.4 network device configurations and technologies are discussed. The chapter is concluded with a discussion of PSAT.

An ideal ac power network has constant voltage and frequency levels at every point of interconnection between the network and one or more loads. Such a point is referred to as the point of common coupling (PCC). Loads can be consumer loads, other power networks or distributed generators. Furthermore, the power factors at all PCCs are unity and no harmonic interference is present. All these parameters are independent of the load size and characteristics, with no interference from one load to another [B3].

A non-ideal network can be characterised in terms of the supply voltage waveform at a particular PCC. *Quality of supply* can therefore be defined as the ability of the supply voltage and frequency to remain within specified parameters, while the power factor strives towards unity [B3]. These parameters quantify the quality of supply and include voltage regulation, dips, interruptions, harmonics, flicker and unbalance. In South Africa NRS 048 [B4] documents the above parameters and was published in 1996 by the South African Bureau of Standards. Parameters to define acceptable quality of supply are given. NRS 048 is the authority on which the National Electricity Regulator (NER) regulates the interests of the national utility company, Eskom, and the electricity consumers.

For a number of years quality of supply has been a well-known and well-defined concept in academic circles. In practice utilities often treat quality of supply and network planning as separate issues, mainly because quality of supply only recently became a general issue. Quality of supply is therefore often seen as a symptom of existing networks and treated by installing additional infrastructure (e.g. extra lines or compensators) to compensate for a specific problem. As this normally goes along with high capital investment, it is only dealt with upon consumer complaints.

Fig. 2.1 outlines a proposed approach to assess the quality of supply at a particular PCC. The aim is to derive analytical models to predict the quality of the supply problems in terms of

voltage and power factor. Models of network technologies that improve the quality of the supply is also developed.

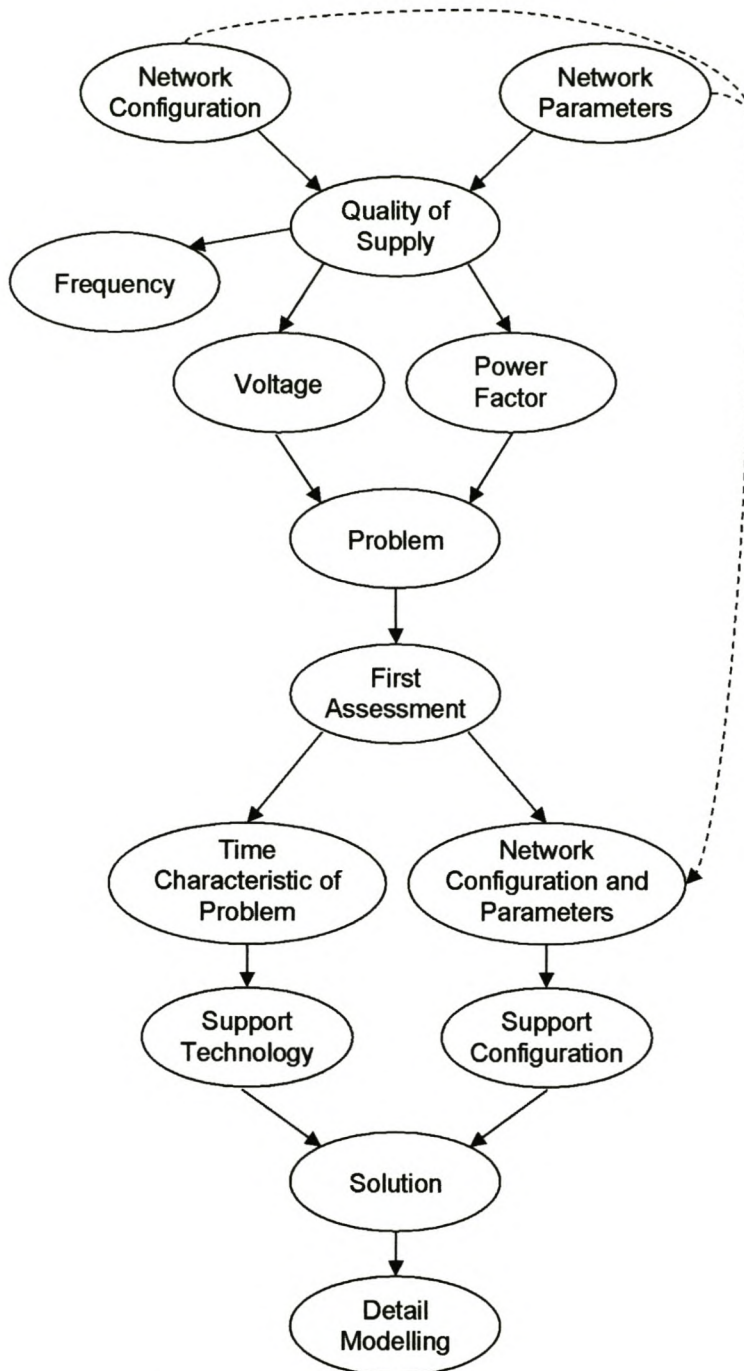


Fig. 2.1: Flow chart of quality of supply analysis

2.2 NETWORK CONFIGURATION AND PARAMETERS

The supply voltage and frequency as well as the power factor are functions of the specific network configuration and the network parameters, e.g. voltage sources, impedances and load power. The quality of supply is therefore a function of the network configuration and of the network parameters, as shown in Fig. 2.1. This section examines these two aspects further.

2.2.1 Network configuration

The network configuration can be described in terms of a sending and receiving end connected at the PCC. It is shown in Chapter 3 that complicated networks, as in Fig. 2.2, could be simplified to the equivalent Thévenin network in Fig. 2.3 [A4]. The network is simplified by successive Thévenin-Norton transformations until a single voltage source and impedance remains at the respective sending and receiving end. The only condition is that parallel connections may only exist between adjacent nodes and therefore mesh networks with more than two sources are excluded. All impedances are treated as complex numbers and for that reason resistive components (losses) are included.

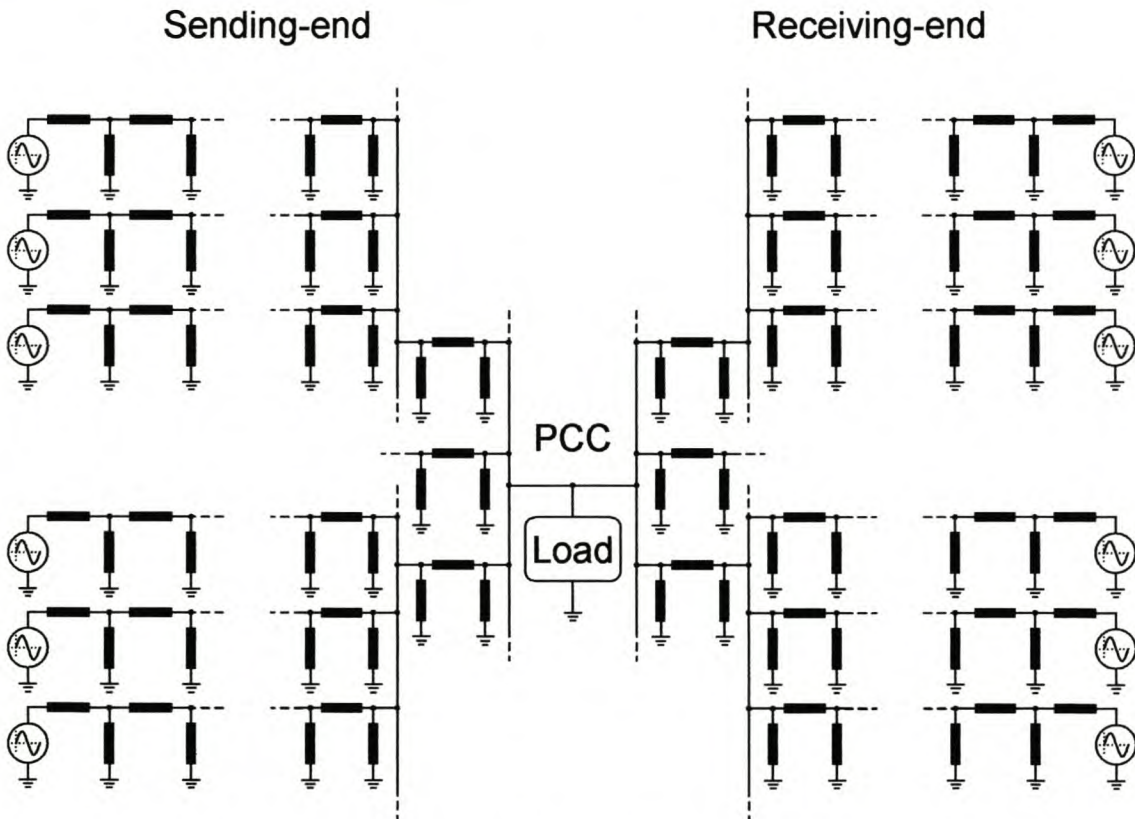


Fig. 2.2: Circuit diagram of extensive networks

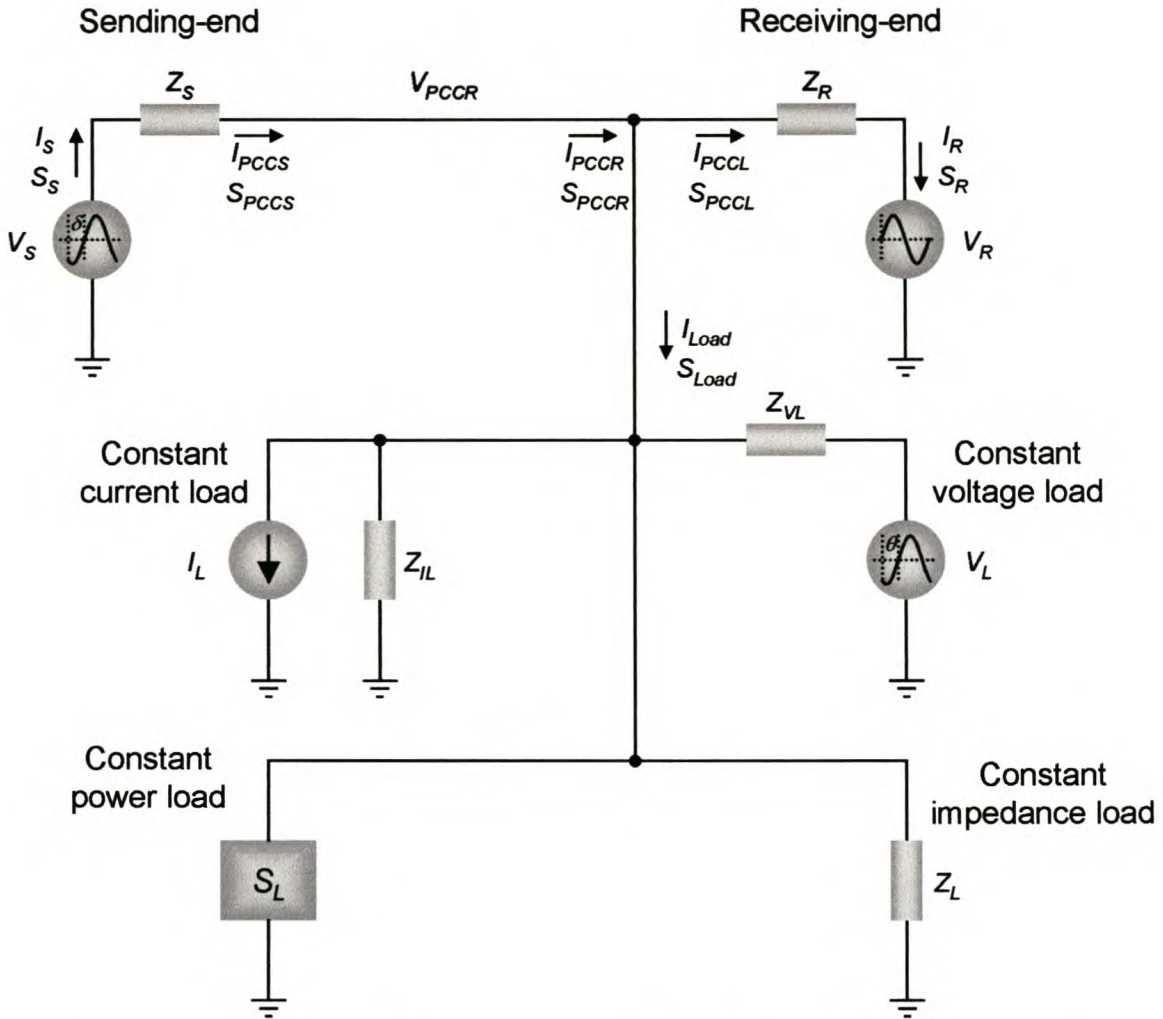


Fig. 2.3: Equivalent circuit diagram, without approximations, for above networks

The system in Fig. 2.3 allows the sending and receiving ends to model two networks connected to the load at the PCC. Four possibilities exist for the load, i.e.:

- A constant voltage load
- A constant impedance load
- A constant power load
- A constant current load.

A constant voltage load is typically encountered where the load is another network or a distributed generator. The back-emf of large synchronous machines can also be modelled in this manner. Rural or residential loads are typically constant impedance loads. These include heating elements of kettles, heaters and geysers as well as incandescent lamps. Most industrial loads fall into the category of constant power loads, which are typically several and large rotating machines. A thyristor rectifier with a significant dc inductance is a typical

constant current load [A5]. A particular load can comprise any combination of these generic loads.

2.2.2 Network parameters

The network parameters in Fig. 2.3, namely V_S , V_R , Z_S , Z_R , I_L , Z_{IL} , Z_{VL} , V_L , S_L and Z_L , are functions solely of the original network parameters, i.e. voltages delivered by voltage sources, currents delivered by current sources, impedances and load power. All the parameters are treated as complex values to include losses. This is a necessity to be able to model transmission, sub-transmission and reticulation networks. In practice, network designers traditionally assumed transmission impedances to be purely reactive and reticulation impedances to be purely resistive [C1]. This rule of thumb works well in most cases, but gives poor results when there is not a significant difference between the resistive and reactive network impedance values. Sub-transmission networks are a typical example of such a situation. The Zimbabwe case study in Chapter 4 is such a sub-transmission network.

V_{PCCR} are determined solely by the network parameters of Fig. 2.3. The quality of supply at the PCC is evaluated by applying the NRS 048 criteria to V_{PCCR} . The parameters of Fig. 2.3 can be determined through Thévenin-Norton transformations from the original network parameters. For large distribution systems with many loads, this process can be very tedious and time consuming. Baran used voltage and power data measured at the substation to calculate voltage drops across radial feeders to loads [A6]. This method is adapted to calculate the equivalent voltage source and impedance for either the sending or receiving end. V_S and Z_S or V_R and Z_R can be calculated from the voltage and power values at the substation and the voltage and power values at the PCC. Section 3.2.2 gives a full explanation of this method as well as the developed equations. The voltages and powers can be obtained through practical field measurements or from previous analyses of the network. The above method provides a simple but accurate bridge between PSAT and any other simulation software, such as Reticmaster. From the voltage and power data at any two nodes in Reticmaster, an equivalent network can be constructed for the network between the nodes. This method is restricted to constant impedance loads between the nodes concerned. With most voltage regulation problems occurring in rural areas, where loads have constant impedances, this restriction is acceptable.

It can thus be concluded that the quality of supply is solely a function of network configuration and network parameters. The network parameters, and therefore the quality of supply, are influenced by contingencies such as line faults, load variations and operating conditions.

2.3 NETWORK PROBLEMS

As discussed at the beginning of this chapter, voltage, power factor and frequency characterise the quality of supply. The quality of supply problems defined by NRS 048 can all be modelled with the circuit in Fig. 2.3, as discussed in Chapter 5.

The scope of this thesis is restricted to voltage regulation, which is only one of the parameters determining the voltage quality. Voltage regulation is the most common quality of supply problem encountered. There are two main reasons for this: the majority of loads are linear and the power network in South Africa is generally secure. NRS 048 defines voltage regulation as *the ability of the steady-state rms voltage to remain between the upper and lower limits* [B4]. These limits are specified as $\pm 5\%$ of the nominal system voltage for ≥ 500 V networks and $\pm 10\%$ of the nominal system voltage for < 500 V networks. A quality of supply problem occurs when these limits are exceeded. The specific network problem is modelled from the practical network configuration and parameters. Voltage regulation is modelled by adjusting the line impedances Z_S and Z_R . If generator regulation is a problem, it can be modelled by adjusting V_S or V_R . Even a complicated radial network can be simplified to these four parameters. PSAT is then used to determine the best solution to this problem. Since PSAT portrays voltage, current and power graphs, power factor can be included in the analysis.

In many cases a new line is still the only solution considered to strengthen the network and improve the voltage regulation. This goes along with high capital investment, long lead times, servitude establishment and can present other problems, such as high fault levels and stability issues. Sometimes it is impossible to construct a new line due to the topography, ecology or legislation regarding the site.

When economic growth is predicted inaccurately, planners are often faced with the problem that load growth is under- or over-estimated. By using turnkey solutions, risk can be reduced or needs can be temporarily addressed until new infrastructure is in place. Furthermore, the benefit of deferring capital investment can subsidise other projects.

2.4 FIRST ASSESSMENT AND SOLUTIONS TO NETWORK PROBLEMS

2.4.1 Network device configuration

Network technologies present an alternative in terms of network devices, which can be used to solve network problems at the PCC. All network support devices can be classified in one of four categories according to their topology [A5], [A7], [A8], [A9], [A10], [A11], [A12], [A13], [A14], [C2]. The four categories are:

- Shunt devices
- Series devices
- Series-shunt devices
- In-line devices.

A shunt capacitor, a synchronous condenser, a static Var compensator, a static compensator and a distributed generator (not to be confused with a distributed generator as power source at the receiving end) are examples of shunt devices. Series devices include a series capacitor, a thyristor-controlled series capacitor (TCSC) and a dynamic voltage restorer (DVR). The unified power-flow controller (UPFC) is a combination of a series and shunt device with a common dc bus and is therefore classified as a series-shunt device. A back-to-back converter (e.g. a universal semiconductor electrification (USE) device) and an HVDC line are network technologies that connect two separate ac networks by a dc link. These devices are not aimed at quality of supply compensation, but have the ability to do so, and are classified as in-line devices.

An energy-storage device can be connected to the dc bus of any transistor-controlled device [C2]. A device with energy storage or a distributed generator can supply both active and reactive power, enabling control of both the voltage magnitude at the PCC ($|V_{PCCR}|$) and the voltage angle at the PCC (θ_{PCCR}). Three strategies to control this extra degree of freedom are based on specifying one of the following requirements:

- Minimum compensator rating
- The active power supplied by the compensator
- The reactive power supplied by the compensator.

If θ_{PCCR} is controlled to achieve the minimum compensator rating, the active and reactive power supplied by the compensator are minimised, while the required voltage regulation is

maintained. If the compensator can supply none or a limited amount of active power (e.g. to stretch stored energy reserves), the active power supplied by the compensator can be specified. A small synchronous generator with no external control over its field current can only supply a fixed amount of reactive power [B5]. Fixed capacitor banks also supply a fixed amount of reactive power. θ_{PCCR} can be controlled to maintain the specified reactive power supplied by the compensator. Further control strategies, which can be developed in future, are suggested in Chapter 5.

General guidelines to determine the best support configuration exist. A shunt compensator, for instance, will have a smaller rating than a series compensator when no load is connected at the PCC and Z_S is large compared to Z_R . On the other hand, a series device will have a smaller rating when there is no receiving end network and Z_S is small compared to the load. PSAT is, however, invaluable to determine the best solution from all the above-mentioned configurations and control strategies. The best support configuration and control strategy is determined by the specific network configuration and parameters.

2.4.2 Network device technology

With the support configuration decided, the support technology must be determined. The network device technology will determine the reaction time and variation of the compensating power. A fixed capacitor compensates continuously, but the compensation level cannot be adjusted. Switched capacitors provide some control over the amount of compensation, but can have a reaction time of up to 10 minutes if the controller does not monitor the instantaneous voltage waveform. This period allows switched-out capacitor banks to discharge to near-zero voltage levels. The capacitor banks can now be switched in at any time in the network cycle and the maximum voltage across the capacitor banks will not exceed the instantaneous network voltage peak. Should a capacitor bank be switched out at the positive peak of the network voltage waveform, it will be positively charged. If this capacitor bank is switched back when the instantaneous network voltage is negative, double the network voltage is present across the capacitor banks. This can lead to excessive transients and even equipment failure. Intelligent controllers can minimise this time delay by monitoring the network waveform and capacitor voltage to achieve synchronised switching.

A thyristor-controlled device like an SVC can react within a half cycle but a transistor-controlled device (e.g. a static compensator) can react in less than a millisecond. A generator,

in combination with an energy-storage device, can be connected to the dc bus of a transistor-controlled compensator [C3]. The favourable dynamics of the compensator are then complemented with the continuous variable power supplied by the generator. Although the dynamic response would be in the order of a few cycles, a generator can also interface directly with the ac network. The advantage of continuous variable compensation is maintained. Switched capacitors are therefore adequate for voltage regulation and power factor correction in general, but large, rapidly varying loads would require at least a thyristor-controlled device. A transistor-controlled device is typically used for harmonic filtering and dip compensation for sensitive plants.

The technology can only be selected once the information about the customer and his plant has been obtained. Device cost increases substantially with speed of response. The textile industry (5 ms) and the uranium-enrichment industry (2 ms) are typical customers requiring very short response time and will thus use transistor-based technology. Other plants, such as stone crushers, are less sensitive to time response and can cope with responses of approximately 20 ms. In these cases thyristor-controlled technology can be used. Independent processes such as water pumping for irrigation do not require any dynamic response; therefore, fixed devices (e.g. capacitor banks) can be implemented to condition the voltage and improve plant efficiency.

For a specific situation the energy-storage technology (e.g. batteries, a flywheel, a super-conductive coil or a fuel cell), together with the amount of stored energy, determines the maximum period of active power delivery by a compensator. Batteries, a flywheel or a super-conductive coil should be adequate for dip compensation, provided the periods between severe dips are long enough to restore the energy used for compensation. Voltage regulation implies periods longer than three seconds. Distributed generators such as fuel cells, or flow batteries [A15] can be used to supply active power over long periods or even continuously.

2.4.3 Solution

One or two optimal solutions based on the discussed factors may exist for a specific network problem. PSAT is used to evaluate each support technology and configuration to determine the optimum solution. The device rating is determined by the specific network configuration and parameters, as well as the device configuration and control method. Equations to analyse every device configuration and control method are therefore needed. The requirements for

these equations are very stringent. No approximations are allowed because the conditions to validate these approximations are violated by the requirements of PSAT. Generally the voltage angle (δ) of a network is limited due to stability concerns, thereby validating the approximations $\sin \delta \approx \delta$ and $\cos \delta \approx 1$. Support technologies can change the power transfer characteristic of a network to such an extent that first-swing stability is maintained for $0 \leq \delta \leq \pi$ rad [B2], rendering these approximations invalid. To maintain applicability to all network configurations, losses cannot be ignored, as many simulation tools do. Lines on sub-transmission networks have almost equal resistances and reactances. The calculation of effective line impedances (including loads) between two nodes on a network, also results in significant resistive components. All resistive components of impedances are accounted for in the developed equations. Chapter 3 deals with the development and implementation of these equations.

The various device configurations are modelled with the circuit of Fig. 2.3. The models of the four generic network devices are shown in Fig. 2.4 to Fig. 2.7. In these figures a single load represents any combination of the various loads of Fig. 2.3. The equations are developed from these models. The aim of the models is to be simple enough (transients are ignored) to aid in the understanding of network problems and solutions, but powerful enough to compare various solutions to a network problem.

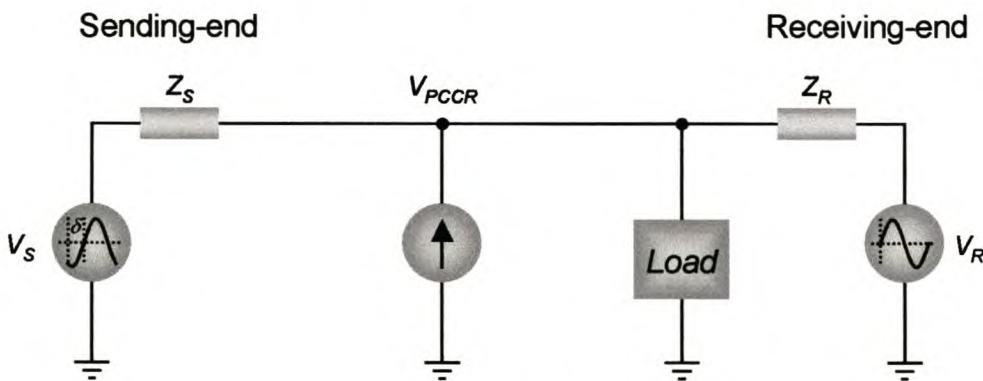


Fig. 2.4: Shunt compensator

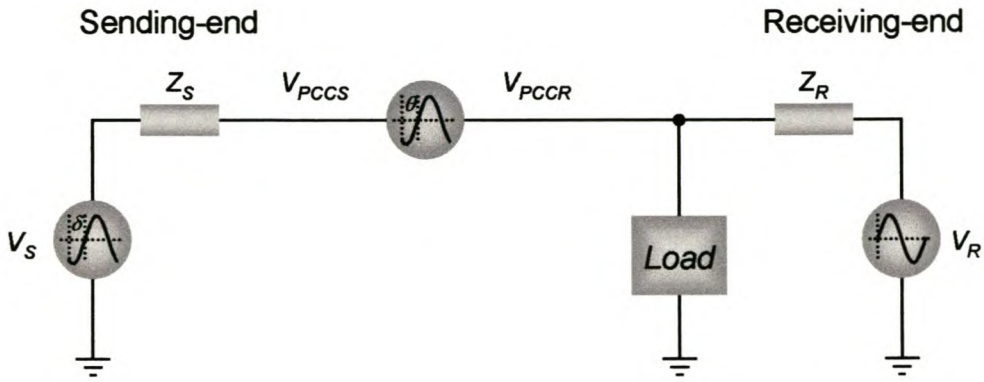


Fig. 2.5: Series compensator

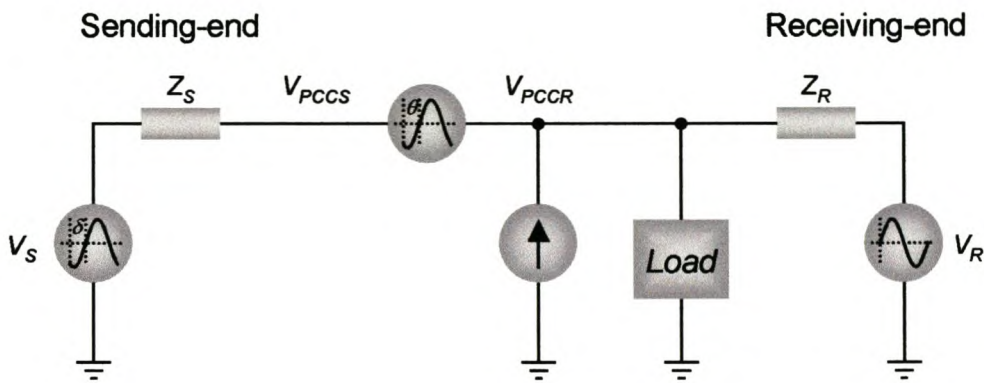


Fig. 2.6: Series-shunt compensator

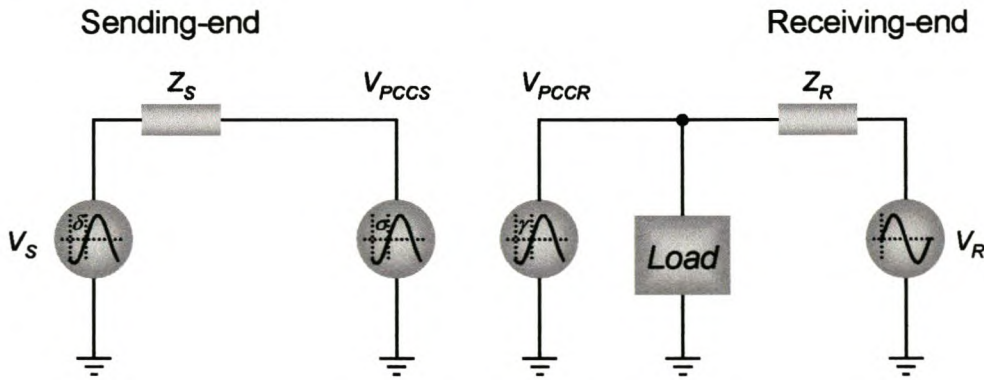


Fig. 2.7: In-line compensator

The above models and developed equations are implemented in PSAT, of which the input interface is shown in Fig. 2.8 and the output interface in Fig. 2.9. Keeping the simulation circuits as simple as possible, a basic understanding of the fundamental network quantities (namely voltage, current and power) is developed. Any network problem can be simulated with this circuit, as discussed in section 2.3, but currently only voltage regulation is implemented.

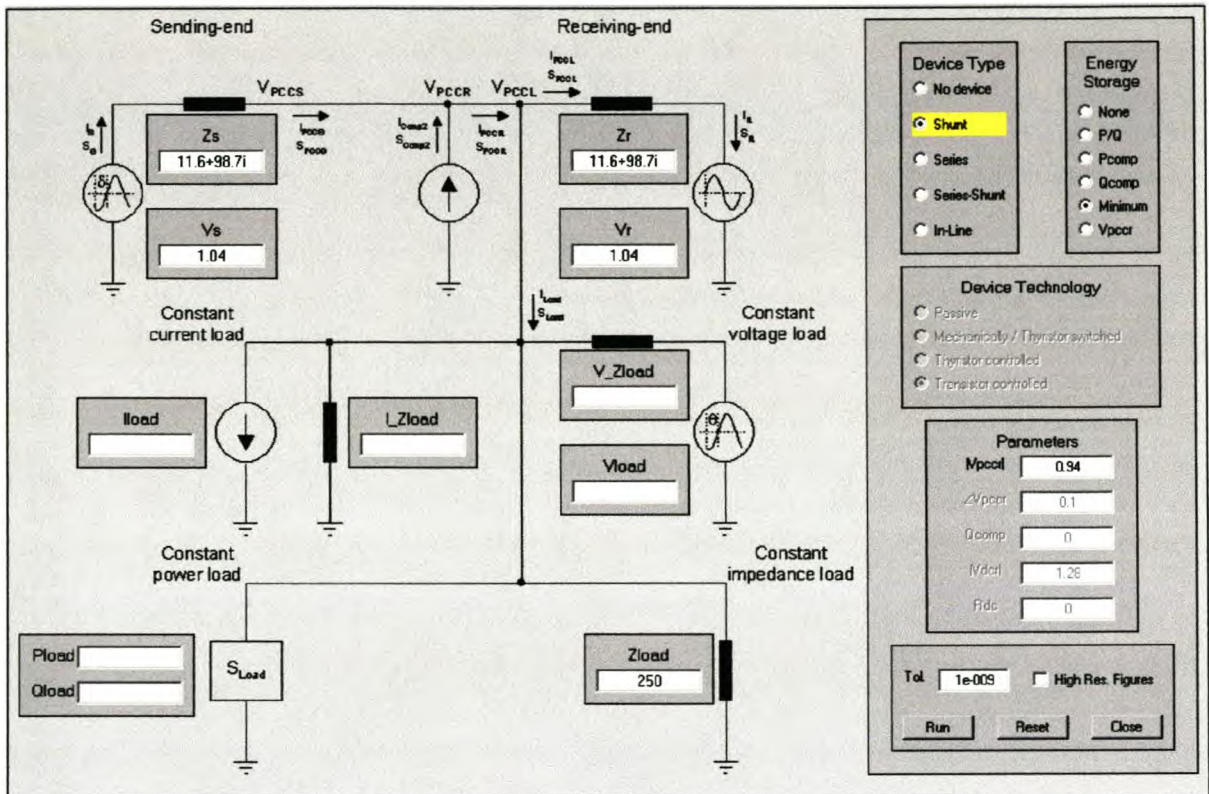


Fig. 2.8: Input interface of PSAT

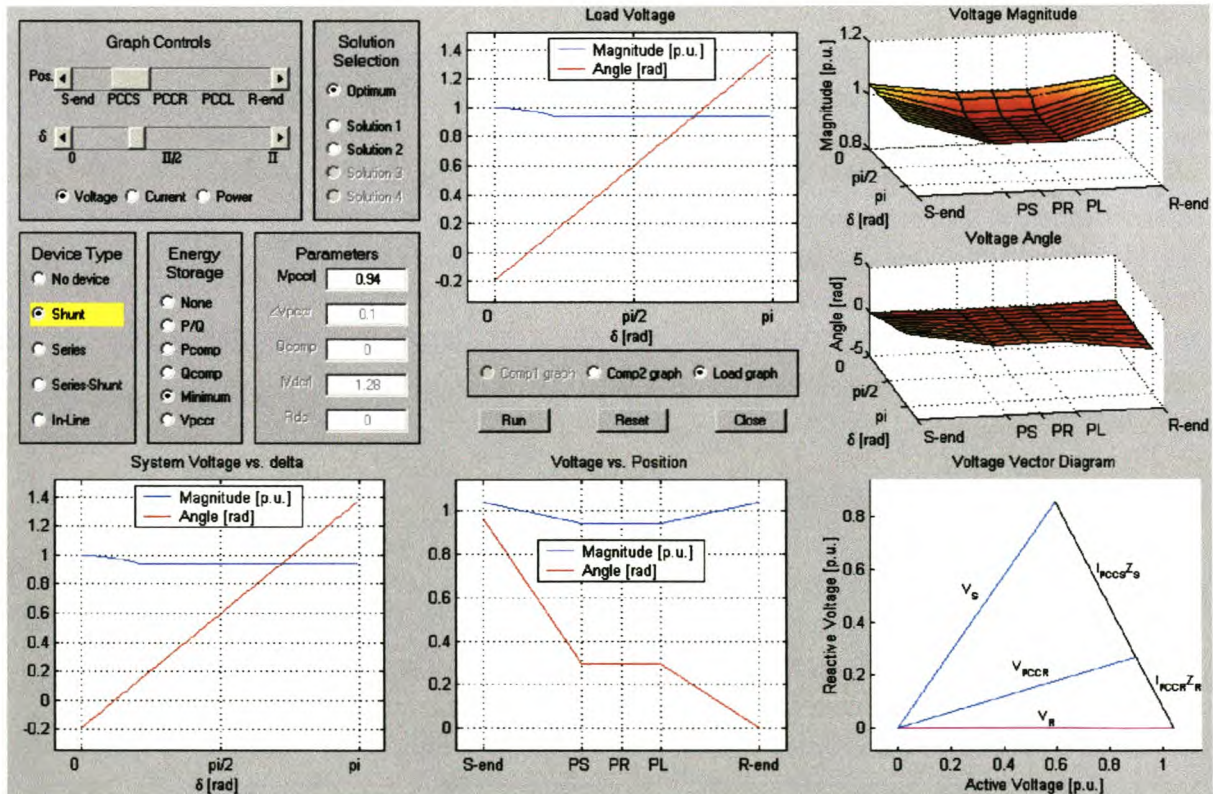


Fig. 2.9: Output interface of PSAT

Three-dimensional graphs are used to portray voltage, current and power across the network. The voltage angle across the network is plotted on the X-axis and the physical position on the

network is plotted on the Y-axis. For voltage and current graphs, magnitude and angle are plotted on the Z-axis. Active and reactive power are plotted on the Z-axis for power graphs. In addition to the three-dimensional graphs, four two-dimensional graphs are also displayed. Two of them provide cross-sections through the three-dimensional graphs at user-selected points (one cross-section is parallel to the XZ-plane and the other parallel to the YZ-plane). A third can display the voltage, current or power of either the compensator or the load. A vector diagram completes the package.

The voltage angle at the receiving end is kept constant at 0 rad and the sending-end voltage angle δ is varied from 0 to π rad. Hereby the power angle is also equal to δ for purely active or reactive network impedances [B6], which is often a valid approximation for practical networks. For complex impedances, plots against δ still clearly show how network voltage, current and power vary with the voltage phase difference across the network.

Implementation

PSAT is implemented using Matlab, developed by Mathworks Inc. Matlab was chosen as it is fundamentally a mathematical package but it also supports the development of object-oriented graphical user interfaces (GUIs). Equations and formulae implemented in Matlab can easily be exported to other simulation packages that allow the user to create custom devices. It also provides excellent graphing tools that are not always available in simulation packages.

Matlab GUIs are built around figures. A function is linked to each object in the figure and is executed according to the selections made or inputs given by the user. Matlab provides excellent printed and on-line documentation [B7] regarding GUIs, and therefore the functions pertaining to the GUI and other graphics are not discussed here. The full program listing is included in Addendum B. Some critical equations are altered to prevent unauthorised copying of the software.

The heart of PSAT is the voltage regulation engine implemented in the file *Voltreg.m*, which is also listed in Addendum B. Although the development of the equations is discussed in Chapter 3, the software implementation is discussed here along with the development of PSAT.

Voltreg.m starts by importing the network parameters specified by the user. Selective parameters used only for certain devices are read throughout the function (*Voltreg.m*). Impedances not specified are set to infinity. Voltage and current source values not specified are set to zero. These parameters are therefore of no consequence in following formulae. In lines 41 and 42 the receiving-end and load parameters are transformed to represent the simpler circuit diagrams presented in Chapter 3.

The uncompensated circuit is solved next irrespective of the compensator chosen. This enables *Voltreg.m* to revert to the uncompensated voltage levels if no valid solution is found for a specific combination of device type and control method. The same method is followed for solving every network scenario:

1. Constants that occur in the voltage regulation equations are specified.
2. The voltage regulation equations are solved in terms of $|V_{PCCR}|$ or θ_{PCCR} .
3. The results are tested by back-substitution into the equations from which the voltage regulation equations were derived.
4. (Only for compensated networks) If no valid solution is found for a specific δ the required $|V_{PCCR}|$ is lowered until at least one valid solution is found or until the uncompensated value for $|V_{PCCR}|$ is reached.
5. Invalid solutions are eliminated to prevent the sorting algorithm from choosing an invalid solution as the most effective one.
6. If no valid solution is found for one or more δ a yellow bar lights up in the PSAT GUI. If no valid solution is found for all δ a red bar is shown. This indicates to the user that the chosen device configuration and control strategy is unable to perform the required voltage regulation for some or all δ .
7. With V_{PCCR} known the rest of the circuit is solved with Ohm's Law and Kirchoff's Voltage and Current Laws.
8. The parameters for the simplified circuit of the load and the receiving end are transformed back to the original circuit parameters.

Duplicate solutions are removed and erratic phase jumps are eliminated when the compensator voltage or current is zero. If more than one solution exist for a specific compensator and control method, an optimum solution is determined for every δ . The optimum solution is determined by evaluating various parameters, in the following order:

1. Minimum difference between specified and regulated magnitude of V_{PCCR}
2. Minimum total apparent compensator rating
3. Smallest angle of total compensator rating (a compensator requiring less active power is therefore favoured)
4. Smallest angle magnitude of regulated V_{PCCR}
5. Positive angles of V_{PCCR} are favoured above negative angles
6. Minimum magnitude of V_{PCCS} .

Finally all valid solutions, together with the optimum solution, are assigned to a data matrix that is used by the GUI to plot the various graphs.

PSAT is used to determine the best solution to a specific network problem and enables the network planner to quickly evaluate a large number of options without the time and costs involved in a detailed simulation of every option considered. Therefore, using PSAT can achieve large savings in terms of time and cost.

The detailed development of the analysis equations and their implementation are discussed in Chapter 3. The Cathedral Peak case study is presented as an integral part of the theory to emphasise the practical application and tutoring abilities of PSAT. This case study is the first of two case studies that are typical examples of the two generic network types discussed in Chapter 1. The Cathedral Peak case study involves a radial reticulation network. The results of a radial network are simpler than that of a twin-source mesh network. This case study was chosen to verify the developed equations and introduce the voltage, current and power graphs generated by PSAT. The Zimbane case study in Chapter 4 involves a two-source mesh network and gives a more extensive view of the abilities of PSAT.

CHAPTER 3 VOLTAGE REGULATION AND VOLTAGE REGULATORS

3.1 INTRODUCTION

3.2 VOLTAGE REGULATION

- 3.2.1 Simple circuit diagram
- 3.2.2 Obtaining network parameters
- 3.2.3 Simplified network
- 3.2.4 Cathedral Peak Hotel case study

3.3 SHUNT VOLTAGE REGULATORS

- 3.3.1 Shunt compensation with active power specified
- 3.3.2 Shunt compensation with reactive power specified
- 3.3.3 Minimum shunt compensator rating
- 3.3.4 Summary of shunt compensation

3.4 SERIES VOLTAGE REGULATORS

- 3.4.1 Series compensator with active power specified
- 3.4.2 Series compensator with reactive power specified
- 3.4.3 Minimum rating series compensator
- 3.4.4 Summary of series compensation

3.5 SERIES-SHUNT COMPENSATION

3.6 IN-LINE COMPENSATION

3.7 NEW LINE

3.8 SUMMARY

- 3.8.1 Equations
- 3.8.2 Cathedral Peak case study
- 3.8.3 Analysis tool

3.1 INTRODUCTION

Chapter 3 is the main chapter of the thesis because the voltage regulation equations for PSAT are developed in this chapter. The aspects of voltage regulation discussed in Chapter 2 are implemented in this chapter. The chapter is started with the development of a generic circuit diagram that can be used to analyse voltage regulation for two-source mesh networks as well as radial networks. Two methods to obtain the parameters for this generic network are presented in section 3.2.2.

The radial network case study, namely the Cathedral Peak network, is introduced in section 3.2.4. The generic network for this system is constructed and the voltage regulation at the PCC is analysed. In the rest of the chapter the equations for shunt, series, series-shunt and in-line compensators are developed. The equations for shunt and series compensation are developed for various control strategies. The compensators are then analysed as voltage regulators on the Cathedral Peak network.

The chapter is concluded with the analysis of a traditional method to solve voltage regulation problems, i.e. by constructing a new line.

3.2 VOLTAGE REGULATION

3.2.1 Simple circuit diagram

Taking into account all the factors discussed in Chapter 2, the voltage regulation at the PCC must now be determined. Fig. 3.1 is the generic circuit diagram for any radial network. This network depicts a typical situation where power is transferred across a long line between two independent stiff busbars, with one or more loads along the line. The sending end, receiving end and the load are connected at the PCC. The load can be any combination between the four generic load types (constant current, constant voltage, constant power and constant impedance loads). The voltage regulation at the PCC is given by the magnitude of V_{PCCR} . V_{PCCS} , V_{PCCR} and V_{PCCL} refer currently to the same node, and are therefore equivalent. However, this nomenclature is required when compensators are added to the circuit in later sections.

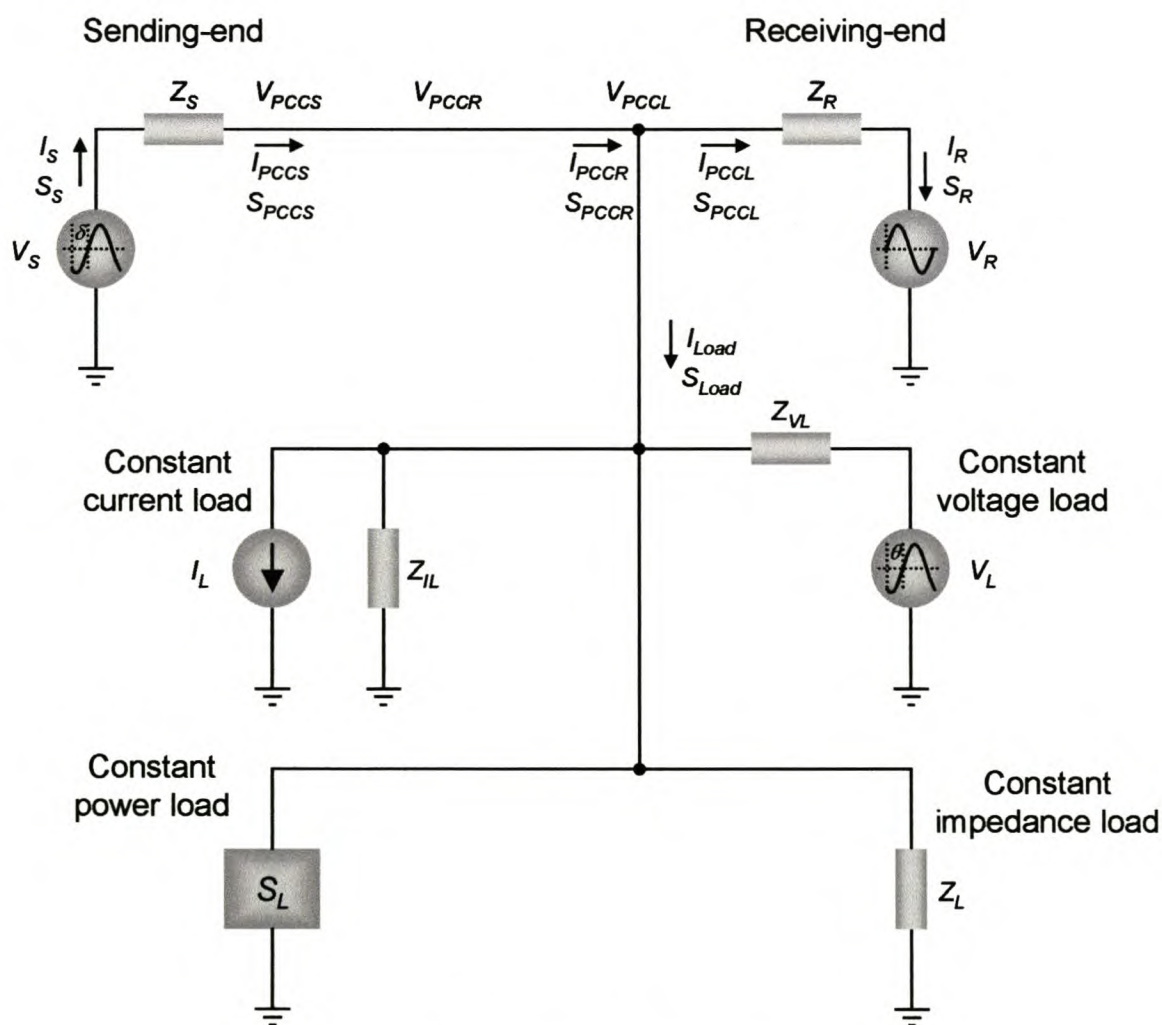


Fig. 3.1: Circuit diagram for analysing the voltage regulation of V_{PCCR}

3.2.2 Obtaining network parameters

From a modelling viewpoint, there are two distinct types of radial networks. Reticulation or distribution networks (below 66 kV) generally have a number of loads connected along the radial lines. Sub-transmission networks (66 kV and 132 kV) may also have loads connected along the line, but generally consist of long lines connecting substations with other substations or loads.

The latter type of network can be modelled by calculating the line impedances from the line data (conductor type, number of conductors, length of line, etc.), using ABCD parameters or RXB parameters. Transformer impedances and load characteristics complete the network model. This model can then be simplified by one or two Thévenin-Norton source transformations until the network model corresponds to Fig. 3.1.

The many loads connected along distribution lines necessitate many Thévenin-Norton source transformations (at least two per load) if the above method is used. A simpler method, which can also be used on transmission and sub-transmission lines, requires only a small number of calculations and can use data either from practical measurements or from other models of the same network.

Fig. 3.2 portrays the sending end of the network, with many loads connected to the distribution line. V_{SS} , S_{SS} , V_{PCCS} and S_{PCCS} (all complex) can be obtained from field measurements or from other network simulations (e.g. *Reticmaster*). With these parameters known, the line and loads connected to it can be represented by a single series and shunt impedance, as shown in (b).

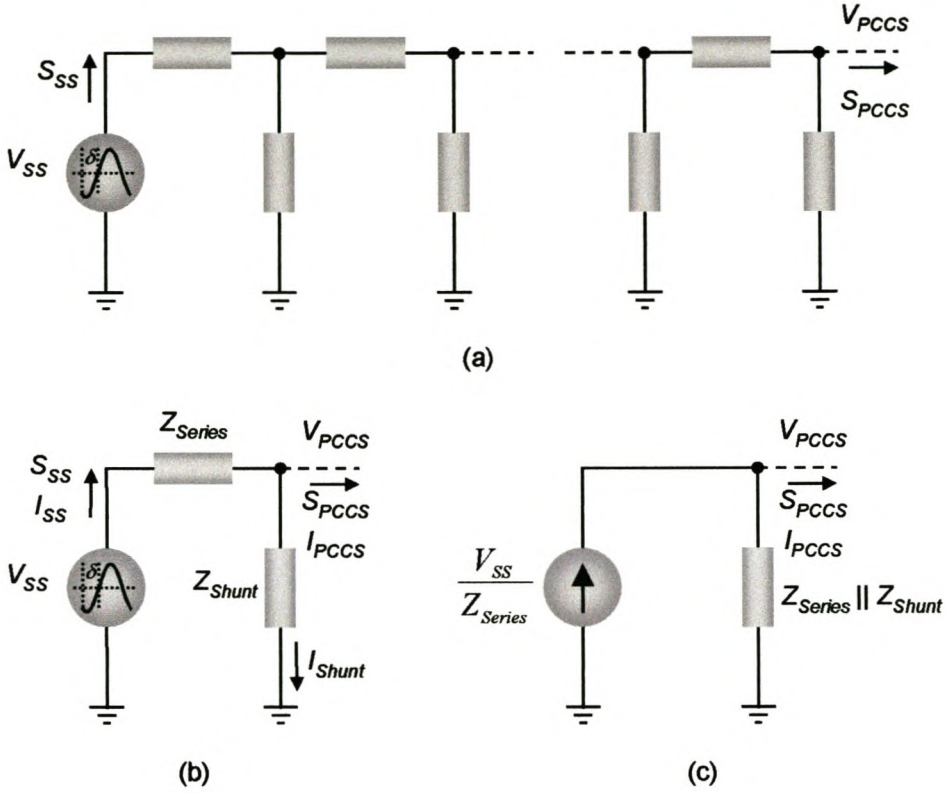


Fig. 3.2: Modelling of a distribution network

From (b), Z_{Series} and Z_{Shunt} are calculated as follows:

$$\begin{aligned}
 V_{Z_{Series}} &= V_{SS} - V_{PCCS} \\
 I_{SS} &= \frac{S_{SS}^*}{V_{SS}^*} \\
 Z_{Series} &= \frac{V_{Z_{Series}}}{I_{SS}} = \frac{|V_{SS}|^2 - V_{PCCS} V_{SS}^*}{S_{SS}^*}
 \end{aligned} \tag{3.1}$$

$$\begin{aligned}
 I_{PCCS} &= \frac{S_{PCCS}^*}{V_{PCCS}^*} \\
 I_{Shunt} &= I_{SS} - I_{PCCS} \\
 Z_{Shunt} &= \frac{V_{PCCS}}{I_{Shunt}} = \frac{|V_{PCCS}|^2 V_{SS}^*}{S_{SS}^* V_{PCCS}^* - S_{PCCS}^* V_{SS}^*}
 \end{aligned} \tag{3.2}$$

With V_{SS} , Z_{Series} and Z_{Shunt} known, a Thévenin-Norton source transformation results in (c). A Norton-Thévenin source transformation finally takes (c) to the sending end of Fig. 3.1, where V_S and Z_S are given by equations (3.3) and (3.4).

$$\begin{aligned}
 V_S &= \frac{V_{SS}}{Z_{Series}} \left(\frac{Z_{Series} Z_{Shunt}}{Z_{Series} + Z_{Shunt}} \right) \\
 &= \frac{V_{SS} Z_{Shunt}}{Z_{Series} + Z_{Shunt}}
 \end{aligned} \tag{3.3}$$

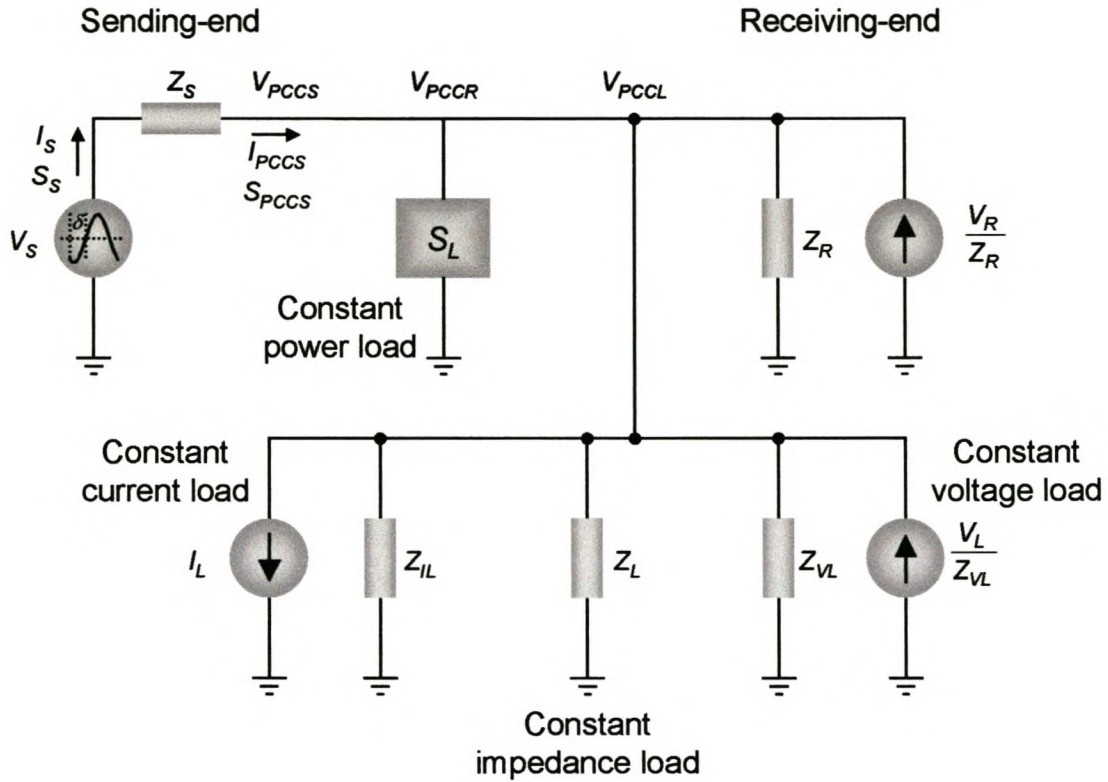
$$\begin{aligned}
 Z_S &= Z_{Series} \parallel Z_{Shunt} \\
 &= \frac{Z_{Series} Z_{Shunt}}{Z_{Series} + Z_{Shunt}}
 \end{aligned} \tag{3.4}$$

Any radial network with numerous loads connected to it can be simplified to a simple Thévenin equivalent circuit with equations (3.1) to (3.4). This method is not restricted to the sending-end of the network, but can also be applied to the receiving end. The sending-end parameters of the Zimbabwe case study in Chapter 4 is calculated with this method of network reduction.

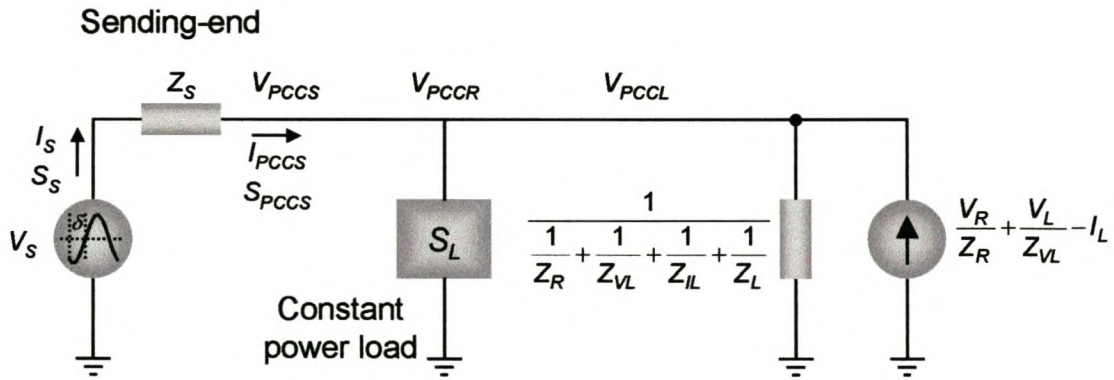
3.2.3 Simplified network

Without any approximations, Fig. 3.1 can be simplified to a less complicated equivalent circuit diagram. An equivalent circuit is derived for the load and the receiving end. The sending end remains unchanged; therefore, the voltage, current and power at the PCC are the same in both circuits. The constant power load S_L is also maintained in its present format, because it is not possible to calculate the current drawn by S_L before V_{PCCR} is known. The constant current, constant voltage and constant impedance loads are combined with the receiving end through Thévenin and Norton transformations, as shown in Fig. 3.3:

1. All Thévenin sources, i.e. the constant voltage load and the receiving end are transformed into their equivalent Norton sources (Fig. 3.3 (a)).
2. The shunt impedances of the receiving end (Z_R), the constant voltage source (Z_{VL}), the constant current source (Z_{IL}) and the constant impedance source (Z_L) are parallel and can be combined into a single impedance. The shunt current sources of the receiving end, the constant current load and the constant impedance load are summed (Fig. 3.3 (b)).
3. Finally, the resulting Norton source is transformed into its Thévenin equivalent (Fig. 3.4).



(a)



(b)

Fig. 3.3: Simplification of Fig. 3.1 through Thévenin and Norton transformations

The resulting circuit diagram is depicted in Fig. 3.4. The constant impedance, constant current and constant voltage loads are included in V_R' and Z_R' at the receiving end, which are given by (3.5) and (3.6), respectively.

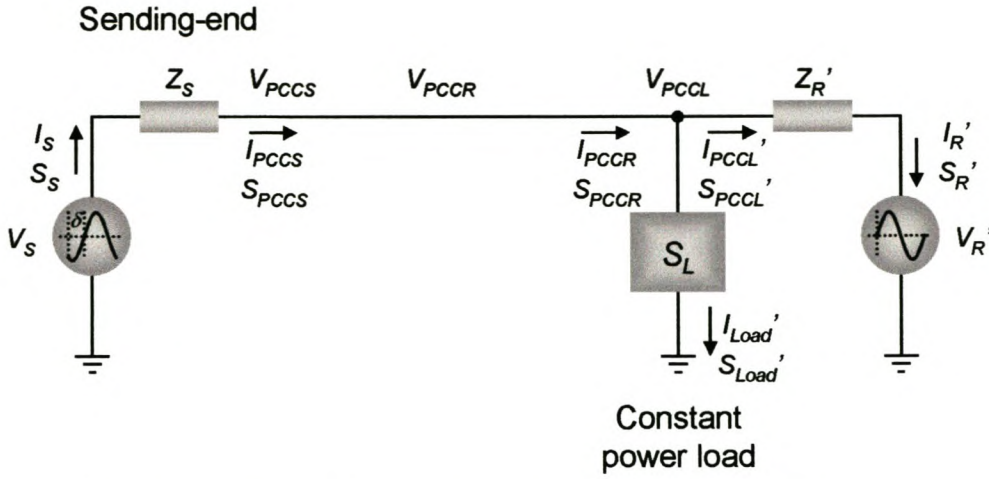


Fig. 3.4: Equivalent circuit diagram for analysing the voltage regulation of V_{PCCR}

$$V_R' = \frac{\frac{V_R}{Z_R} + \frac{V_L}{Z_{VL}} - I_L}{\frac{1}{Z_R} + \frac{1}{Z_{VL}} + \frac{1}{Z_{IL}} + \frac{1}{Z_L}} = \left(\frac{V_R}{Z_R} + \frac{V_L}{Z_{VL}} - I_L \right) \left(\frac{1}{Z_R} + \frac{1}{Z_{VL}} + \frac{1}{Z_{IL}} + \frac{1}{Z_L} \right) \quad (3.5)$$

$$Z_R' = \frac{1}{\frac{1}{Z_R} + \frac{1}{Z_{VL}} + \frac{1}{Z_{IL}} + \frac{1}{Z_L}} \quad (3.6)$$

The voltage regulation at the PCC is subsequently derived from Ohm's Law and Kirchoff's Voltage and Current Laws. From Kirchoff's Current Law, the sum of the power received at and delivered from the PCC is zero:

$$S_{PCCS} = S_{PCCL}' + S_{Load}' \quad (3.7)$$

From Fig. 3.4, $V_{PCCS} = V_{PCCR} = V_{PCCL}$; therefore S_{PCCS} can be derived from V_{PCCR} and I_{PCCS} . The magnitude and angle of V_{PCCR} are given by $|V_{PCCR}|$ and θ_{PCCR} , respectively.

$$S_{PCCS} = V_{PCCR} I_{PCCS}^* \quad (3.8)$$

$$\begin{aligned} S_{PCCS} &= V_{PCCR} \left(\frac{V_S - V_{PCCR}}{Z_S} \right)^* \\ &= |V_{PCCR}| (\cos \theta_{PCCR} + j \sin \theta_{PCCR}) (c_a + jc_b) + |V_{PCCR}|^2 (c_c + jc_d) \end{aligned} \quad (3.9)$$

$$\text{where } c_a = \operatorname{Re} \left(\frac{V_S^*}{Z_S^*} \right) \quad c_b = \operatorname{Im} \left(\frac{V_S^*}{Z_S^*} \right) \quad c_c = \operatorname{Re} \left(-\frac{1}{Z_S^*} \right) \quad c_d = \operatorname{Im} \left(-\frac{1}{Z_S^*} \right)$$

Similarly, S_{PCCL} can be derived from V_{PCCR} and I_{PCCL} :

$$S_{PCCL}' = V_{PCCR} I_{PCCL}^* \quad (3.10)$$

$$\begin{aligned} S_{PCCL}' &= V_{PCCR} \left(\frac{V_{PCCR} - V_R}{Z_R} \right)^* \\ &= |V_{PCCR}| (\cos \theta_{PCCR} + j \sin \theta_{PCCR}) (c_e + jc_f) + |V_{PCCR}|^2 (c_g + jc_h) \end{aligned} \quad (3.11)$$

$$\text{where } c_e = \operatorname{Re} \left(-\frac{V_R^*}{Z_R^*} \right) \quad c_f = \operatorname{Im} \left(-\frac{V_R^*}{Z_R^*} \right) \quad c_g = \operatorname{Re} \left(\frac{1}{Z_R^*} \right) \quad c_h = \operatorname{Im} \left(\frac{1}{Z_R^*} \right)$$

The active and reactive power of S_{Load}' are given by P_{Load}' and Q_{Load}' , respectively. Equations (3.9) and (3.11) are substituted into (3.7), which results in a complex equation in two unknowns, namely $|V_{PCCR}|$ and θ_{PCCR} . The active and reactive parts of this complex equation is given by (3.12) and (3.13). These equations both have terms in $\cos \theta_{PCCR}$ and $\sin \theta_{PCCR}$, therefore both are quadratic functions of θ_{PCCR} . In addition, both equations are quadratic functions of $|V_{PCCR}|$. With all the constants known, $|V_{PCCR}|$ and θ_{PCCR} can be solved from these two equations.

$$|V_{PCCR}|^2 (c_c - c_g) + |V_{PCCR}| \cos \theta_{PCCR} (c_a - c_e) - |V_{PCCR}| \sin \theta_{PCCR} (c_b - c_f) - P_{Load}' = 0 \quad (3.12)$$

$$|V_{PCCR}|^2 (c_d - c_h) + |V_{PCCR}| \cos \theta_{PCCR} (c_b - c_f) + |V_{PCCR}| \sin \theta_{PCCR} (c_a - c_e) - Q_{Load}' = 0 \quad (3.13)$$

The solutions to $|V_{PCCR}|$ and θ_{PCCR} are fourth-order solutions. Explicit solutions exist for fourth-order equations but the formulae are very long and tedious to work with. Maple, a symbolic mathematical software package from Waterloo, was used to solve the above equations. Addendum B contains the listing of *SolPQ.mws*, the file written to solve $|V_{PCCR}|$ and θ_{PCCR} .

To model the practical network accurately, the correct solution must be chosen from the four possibilities. It is assumed that the loads are gradually added to the network. The solution closest to V_{PCCR} with no load connected to the network is therefore chosen. If all the loads are added at once, it is very difficult to say which theoretical solution is the correct one. Negative solutions to $|V_{PCCR}|$ are ignored as the phase angle is already given by θ_{PCCR} .

With V_{PCCR} known, the circuit in Fig. 3.1 is solved. All network voltages, currents and power can be calculated from V_S , V_R , Z_S , Z_R , the load parameters and V_{PCCR} . It must be remembered that the circuit in Fig. 3.4 is only a tool to solve the original circuit in Fig. 3.1.

The voltage regulation problem is influenced by the network parameters. It is clear from the above equations that the network impedances play a vital role in the voltage regulation at the PCC. In practice failure of equipment over a long period and overloading of the network cause undesirable impedances. When the constant power load currents are high, P_{Load} and Q_{Load} are high. High load currents caused by the constant voltage, constant impedance and constant current loads, result in a high I_{PCC} . This can result in poor voltage regulation due to the voltage drops across Z_S and Z_R . Generator regulation can be modelled by adjusting V_S to reflect the generator output voltage.

3.2.4 Cathedral Peak Hotel case study

For a better understanding of the practical relevance of this work, a case study is presented as an integral part of the thesis. Due to the rugged Drakensberg terrain, a 45 km long 11 kV radial line feeds the Cathedral Peak Hotel, which is right at the end of the line. Various other customers are also connected along the line. Currently the voltage regulation at the hotel is just within limits, but loads in the area are increasing rapidly. Load-growth studies of this region indicate a voltage regulation problem in 2005. The projected network for 2005 is shown in Fig. 3.5 [C4].

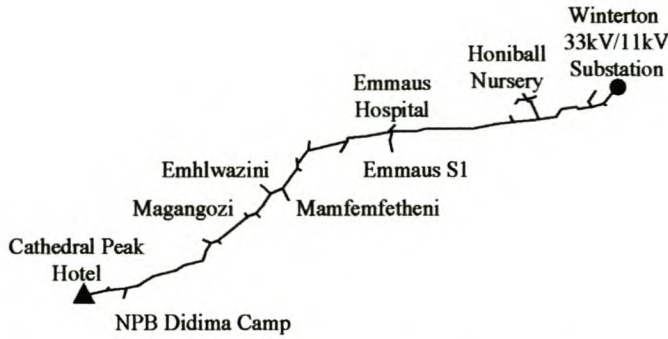


Fig. 3.5: Projected network feeding Cathedral Peak Hotel in 2005 and its equivalent circuit diagram

Fig. 3.6 shows the circuit diagram for this network. The load, which is the Cathedral Peak Hotel, is at the end of the line. This is a radial network and has only one voltage source, namely Winterton substation. Consequently, the receiving end of Fig. 3.1 is omitted. The projected voltage-regulation problem will occur in the winter when the heating demands of Honiball Nursery and Emmaus hospital are the highest. The hotel load can be modelled as 90% constant impedance and 10% constant power in the winter because incandescent lamps and heating elements (e.g. ovens, kettles, boilers and heaters) are the major loads at the hotel. These are constant impedance loads, which consume more power under higher voltage levels. The 10% constant power load models electric machines (e.g. refrigeration and ventilation fans). V_S and Z_S are obtained from the Eskom report on the Cathedral Peak network [C4].

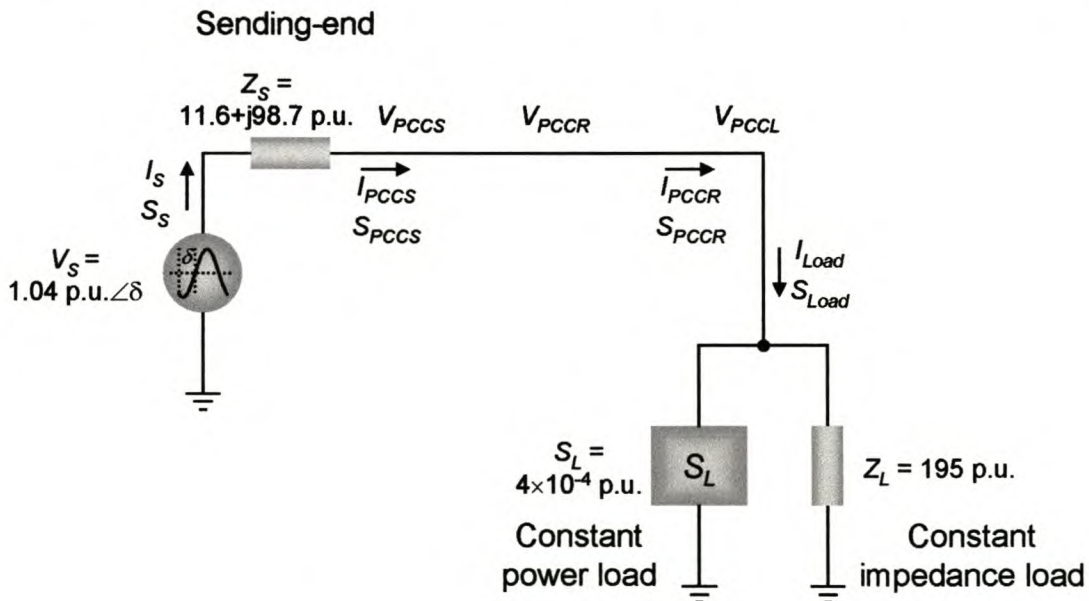


Fig. 3.6: Circuit diagram for the Cathedral Peak network

Field measurements indicate that the power factor of the hotel is very good. For analysis purposes, it is taken as one. All values are per unit, with base values $S_{base} = 100$ MVA and

$V_{base} = 11$ kV, except for the hotel load itself, where $V_{base} = 230$ V. Equations (3.12) and (3.13) is used in PSAT to compute the voltage regulation at the PCC.

The phase angle of V_S (δ , as indicated Fig. 3.6) was given a fixed value when the network parameters were determined. All other parameters are kept constant while δ is varied from 0 to π rad to simulate various voltage angles across the network. A higher δ generally results in a higher power transfer across the network, but can also influence the voltage regulation at the PCC. In practice uncompensated networks are seldom operated at phase (power) angles larger than 0.7 rad, but a better understanding is developed by analysing the network voltage and power for $0 \leq \delta \leq \pi$. The possibility exists that a compensated network can be operated at larger voltage angles because the compensator can increase the first swing stability margin [B2].

The voltage plots indicating the voltage regulation for this network are shown in Fig. 3.7. Fig. 3.7 represents the voltage regulation across the network of Fig. 3.5. The physical location on the network is plotted on the Y-axis and δ is plotted on the X-axis. Therefore, the Z-axis represents the respective voltage magnitude and angle at a specific location in the network, with a certain phase angle across the network. The graphs are plotted for $0 \leq \delta \leq \pi$.

For the Cathedral Peak network, where there is no receiving-end source, the three-dimensional plots seem superfluous. The voltage magnitude remains constant across δ . The only influence δ has on the voltage angle is an offset equal to δ . These plots do not seem very innovative as the added third dimension carries very little information. This scenario changes as soon as a voltage source is added at the receiving end, as shown in section 3.7 and in Chapter 4 for the Zimbane case study. The Cathedral Peak case study was specifically selected for the relatively simple graphs it produces. The correctness of the developed equations cannot be clearly verified with more complex graphs, although PSAT is able to produce the more involved graphs required by more complicated networks.

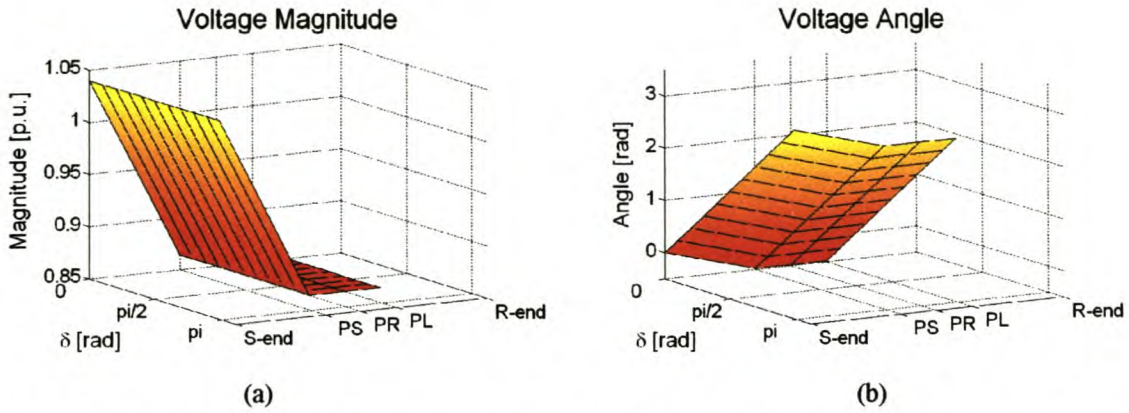


Fig. 3.7: Voltage regulation as a function of network position and voltage angle for the uncompensated Cathedral Peak network

The voltage-regulation problem at the PCC is clearly visible in the magnitude plot. In all plots, the PCC is labelled as follows: PS for PCCS, PR for PCCR and PL for PCCL. Due to the constant load of the hotel and only one voltage source in the network, the voltage magnitude at a specific location in the network remains constant for all δ . With no receiving-end source or impedance, the voltage graphs stop at the PCC. The voltage drop across Z_S is visible between the sending end and PCCS in both the voltage magnitude (a) and angle (b) graphs. For a radial network the relative voltage angle across the network stays constant for all δ . δ provides only an offset in (b) that is equal to δ . The third dimension therefore contains no information for a radial network. From this point forward all graphs will thus be presented in two dimensions, but it must be remembered that the three-dimensional graphs remain available in PSAT. A cut parallel to the YZ-plane at $\delta=0$ rad reveals the true extent of the voltage regulation problem.

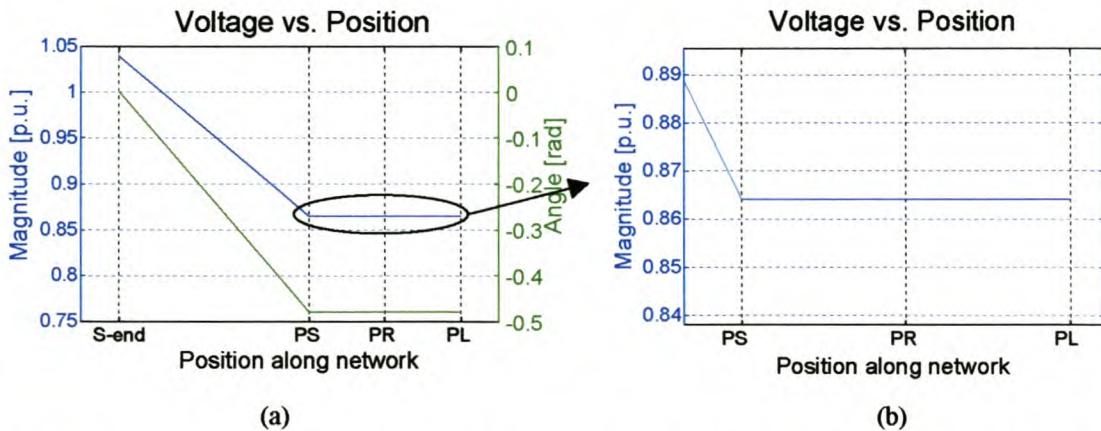


Fig. 3.8: Voltage regulation as a function of network position for the uncompensated Cathedral Peak network at $\delta = 0$ rad

In Fig. 3.8 $|V_{PCCR}|$ equals 0.864 p.u., which is well below the NRS 048 standard of 0.9 p.u. for <500 V networks [B4]. A higher load demand will worsen the voltage regulation at the PCC. The equivalent sending end voltage magnitude is 1.04 p.u., as stipulated above. The voltage drop across Z_S is:

$$1.04\angle 0 - 0.864\angle -0.483 = 0.486 \text{ p.u. } \angle 0.970 \text{ rad}$$

The negative voltage angle at PCCR is due to the voltage drop across Z_S while the sending-end voltage angle, δ , is zero.

The power transfer across the network of Fig. 3.6 is plotted in Fig. 3.9. With the constant load there is no variation in the network power as a function of δ . The load is situated at PCCL. The zero reactive power demanded by the load corresponds to the unity power factor of the hotel. The drop in active and reactive power between the sending end and the PCC is the power consumed by Z_S . The active power consumed by Z_S is 0.278×10^{-3} p.u. (27.8 kW), while the reactive power consumption is 2.36×10^{-3} p.u. (236 kVar). The equations and analysis software can be checked by correlating the voltage drop across Z_S , the power consumed by Z_S and the impedance of Z_S .

$$\begin{aligned} S_{Z_S} &= \frac{|V_{Z_S}|^2}{Z_S^*} \\ &= \frac{0.486^2}{11.6 - j98.7} \\ &= (0.277 + j2.36) \times 10^{-3} \text{ p.u.} \\ &= 27.7 + j236 \text{ kVA} \end{aligned}$$

The calculated and measured values of the power dissipated in Z_S correspond, proving implementation of the equations in PSAT to be accurate. The slight difference can be attributed to rounding errors.

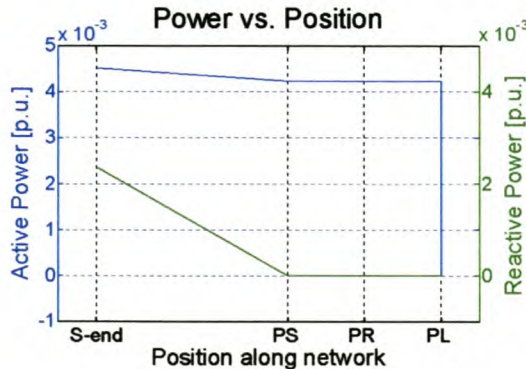


Fig. 3.9: Active and reactive power in the uncompensated Cathedral Peak network at $\delta = 0$ rad

The power dissipated in the load is represented by the step in power at PCCL. This step is purely active and equal to 0.00423 p.u., which corresponds to the specified value of S_{PCCS} . This translates into a load power rating of 423 kW.

Fig. 3.10 portrays the current magnitude and angle across the network. With constant power delivery across δ , the current magnitude and angle display a similar characteristic as the voltage magnitude and angle. The current magnitude is constant across δ and the current angle offset varies with δ . Only the current angle at $\delta=0$ rad is shown here. There is no receiving end and therefore all the current passes through the load, as indicated by the step in both the current magnitude and angle to zero at PCCL. At $\delta=0$ rad, the current demand of the load is $I_{Load} = 4.89 \times 10^{-3}$ p.u. $\angle -0.048$ rad.

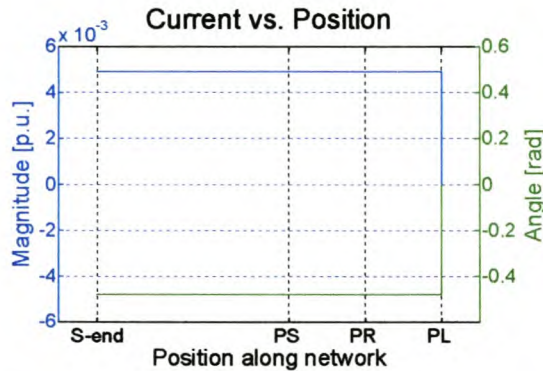


Fig. 3.10: Current magnitude and angle in the uncompensated Cathedral Peak network

From the voltage plot V_{PCCL} is 0.864 p.u. $\angle -0.048$ rad at this point. Using the load voltage and current, the load power is calculated.

$$\begin{aligned}
 S_{Load} &= V_{PCCL} I_{Load}^* \\
 &= 4.23 \times 10^{-3} \text{ p.u. } \angle 0 \text{ rad} \\
 &= 423 \text{ kW}
 \end{aligned}$$

This result corresponds to the measured load power of 0.00423 p.u. and the load power given by S_{PCCS} above.

The negative current angle at $\delta=0$ rad can be explained with the aid of the voltage and current vector diagrams in Fig. 3.11 that is plotted at $\delta=0$ rad. The scaling of the Y-axis and X-axis differs significantly to create larger vector diagrams.

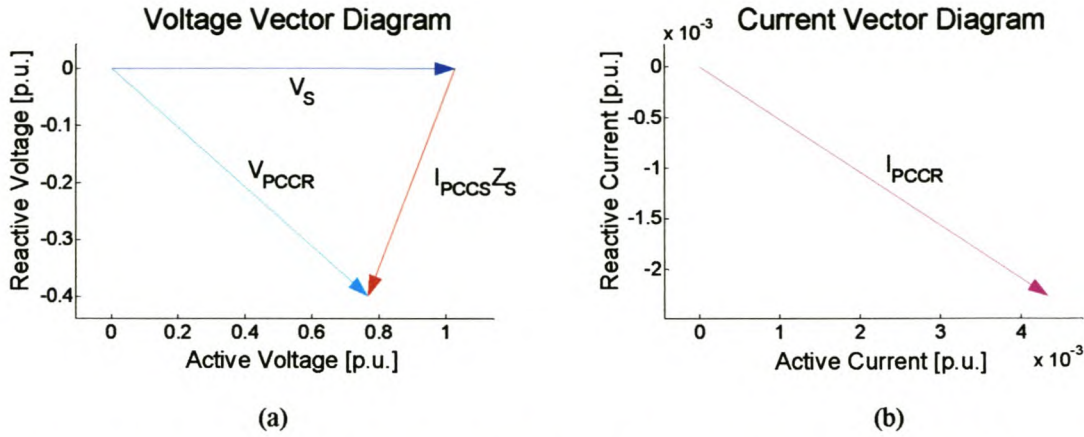


Fig. 3.11: Voltage and current vector diagrams for the uncompensated Cathedral Peak network

For small values of δ (less than 0.483 rad), the angle of V_{PCCR} is negative. This is due to the constant load current that results in a constant voltage drop across Z_S , as shown in the voltage vector diagram in Fig. 3.11. The phase angles of V_{PCCR} and I_{Load} are identical because the power factor of the load is unity. Therefore, just like the angle of V_{PCCR} , the current angle is negative for $\delta < 0.483$ rad and maintains this offset with respect to δ (the voltage angle of V_S) for all δ .

In the next section the compensator topologies and control strategies discussed in section 2.4.1 will be investigated and analysed as solutions to the voltage-regulation problem.

3.3 SHUNT VOLTAGE REGULATORS

A shunt compensator is very well suited to compensate for poor voltage regulation when Z_S is large compared to the load and Z_R , because V_{PCCR} is regulated by controlling the current through Z_S . With Z_S large, little compensation current is necessary to regulate V_{PCCR} . For radial lines, the line impedance is generally much smaller than the typical load to enhance power transfer and minimise network losses. The exception is for lines with many loads connected to them. V_S and Z_S are then equivalent parameters that represent the radial line and all loads connected to it. In this case, Z_S can be relatively large, compared to the load impedance. In a mesh network with two voltage sources and no load at the PCC, a shunt compensator can be an effective voltage regulator that increases power transfer across the network. The equivalent circuit diagram for a network with a shunt compensator at the PCC is shown in Fig. 3.12.

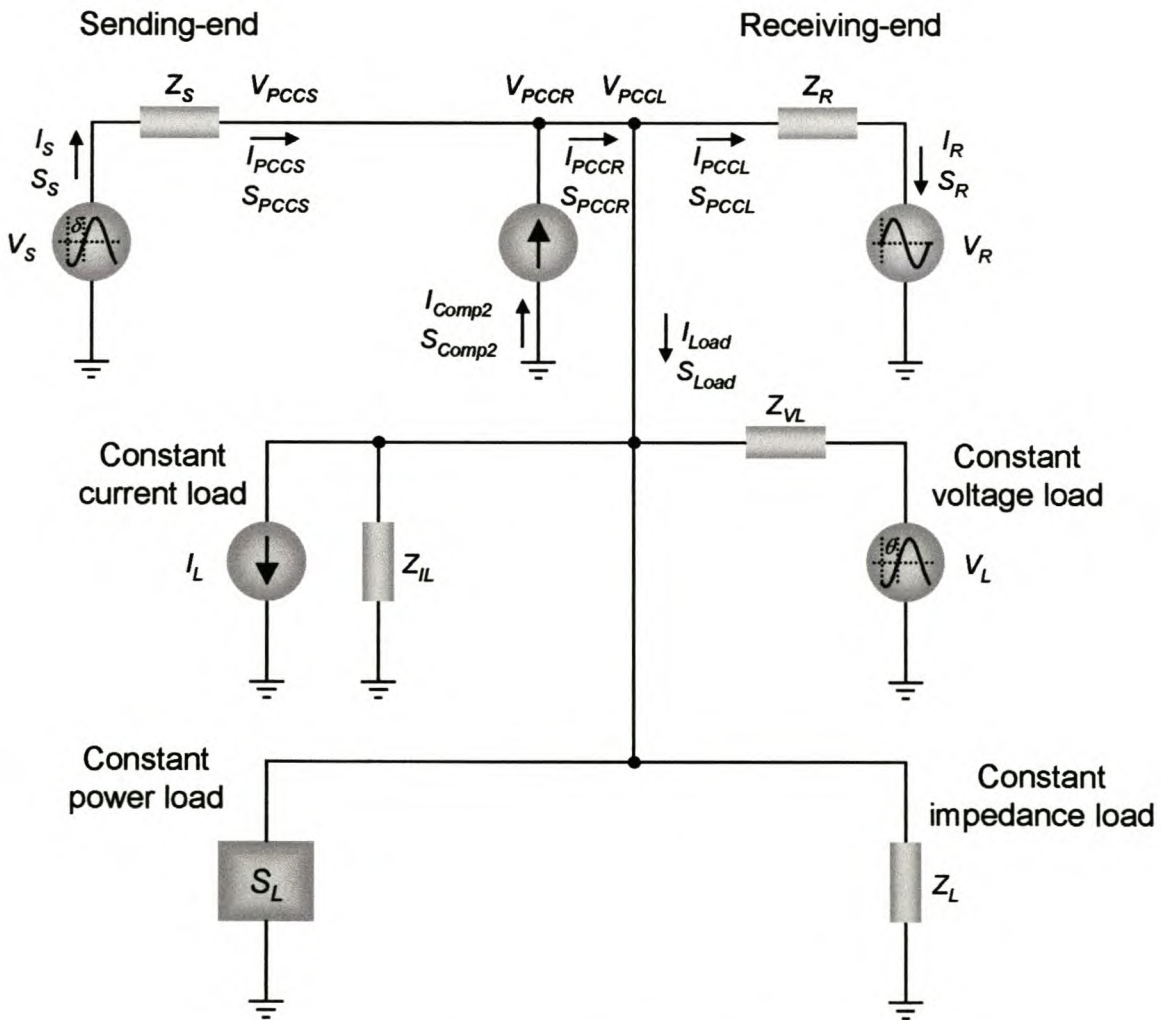


Fig. 3.12: Shunt voltage regulator

As in section 3.2.3, the above network can be simplified, without any approximations, to the circuit in Fig. 3.13, which renders the calculation of V_{PCCR} less difficult.

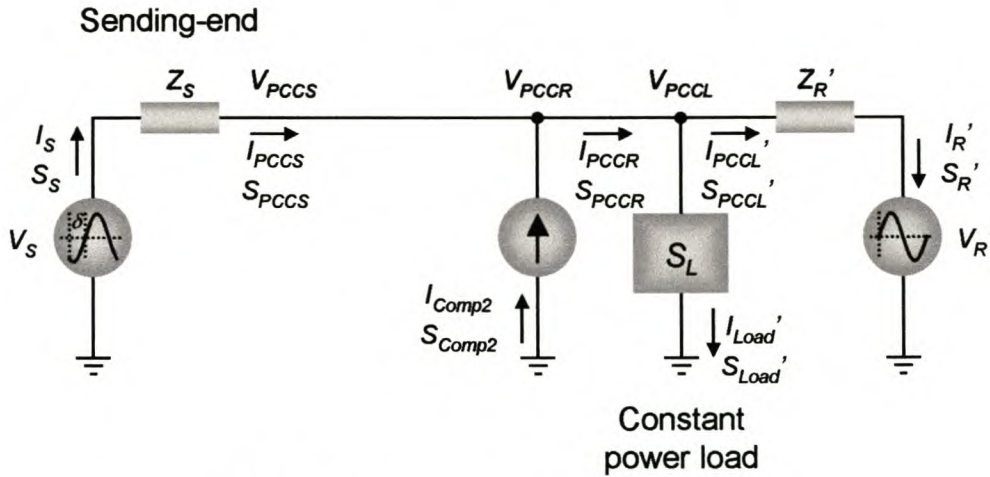


Fig. 3.13: Simplified shunt-compensated network

$|V_{PCCR}|$ is specified by the network planner, while V_s , V_R , Z_s , and Z_R are known network parameters. The equations for these parameters are derived in section 3.2.2. As in the uncompensated case, V_{PCCS} , V_{PCCR} and V_{PCCL} are equivalent. If the phase angle of V_{PCCR} (namely θ_{PCCR}) can be solved, all other unknowns can be calculated from Ohm's Law and Kirchoff's Current Law. The derivation of the equations for θ_{PCCR} is started at Kirchoff's Current Law: the sum of the currents and power received at and delivered from the PCC is zero:

$$\begin{aligned} S_{PCCR} &= S_{PCCS} + S_{Comp2} \\ &= S_{PCCL}' + S_{Load}' \end{aligned} \tag{3.14}$$

For the uncompensated case, S_{PCCS} and S_{PCCL}' are given by equations (3.9) and (3.11) in section 3.2.3. These equations are repeated here and are valid for all compensated networks because the compensators are inserted between PCCS and PCCR. Equation (3.15) is valid for power transferred from the sending-end to PCCS. Equation (3.16) describes power transfer from PCCL to the receiving-end. These two equations are therefore valid for any compensator situated between PCCS and PCCR.

$$S_{PCCS} = |V_{PCCR}|(\cos\theta_{PCCR} + j\sin\theta_{PCCR})(c_a + jc_b) + |V_{PCCR}|^2(c_c + jc_d) \quad (3.15)$$

$$S_{PCCCL}' = |V_{PCCR}|(\cos\theta_{PCCR} + j\sin\theta_{PCCR})(c_e + jc_f) + |V_{PCCR}|^2(c_g + jc_h) \quad (3.16)$$

where $c_a = \text{Re}\left(\frac{V_S^*}{Z_S}\right)$ $c_b = \text{Im}\left(\frac{V_S^*}{Z_S}\right)$ $c_c = \text{Re}\left(-\frac{1}{Z_S^*}\right)$ $c_d = \text{Im}\left(-\frac{1}{Z_S^*}\right)$

and $c_e = \text{Re}\left(-\frac{V_R^*}{Z_R}\right)$ $c_f = \text{Im}\left(-\frac{V_R^*}{Z_R}\right)$ $c_g = \text{Re}\left(\frac{1}{Z_R^*}\right)$ $c_h = \text{Im}\left(\frac{1}{Z_R^*}\right)$

The active and reactive power of S_{Load}' are given by P_{Load}' and Q_{Load}' , respectively, and P_{Comp2} and Q_{Comp2} represent the active and reactive power of the shunt compensator. S_{Comp2} is the apparent power delivered by the shunt compensator. Substituting (3.15) and (3.16) in (3.14) and setting the result to zero produces an equation in complex variables. Equation (3.17) is the active part of this equation and (3.18) is the reactive part.

$$\begin{aligned} &|V_{PCCR}|^2(c_c - c_g) + |V_{PCCR}|\cos\theta_{PCCR}(c_a - c_e) \\ &- |V_{PCCR}|\sin\theta_{PCCR}(c_b - c_f) - P_{Load}' + P_{Comp2} = 0 \end{aligned} \quad (3.17)$$

$$\begin{aligned} &|V_{PCCR}|^2(c_d - c_h) + |V_{PCCR}|\cos\theta_{PCCR}(c_b - c_f) \\ &+ |V_{PCCR}|\sin\theta_{PCCR}(c_a - c_e) - Q_{Load}' + Q_{Comp2} = 0 \end{aligned} \quad (3.18)$$

The two equations above have three unknowns, namely θ_{PCCR} , P_{Comp2} and Q_{Comp2} . A third equation is therefore required to solve the three unknowns. The third equation is determined by the control strategy and availability of energy storage, as discussed in section 2.4.1. In the following sections θ_{PCCR} , P_{Comp2} and Q_{Comp2} will be solved for the following cases:

- P_{Comp2} (active power delivered by compensator) specified by the network planner
- Q_{Comp2} (reactive power delivered by compensator) specified by the network planner
- Minimum S_{Comp2} (minimum apparent power delivered by the compensator).

3.3.1 Shunt compensation with active power specified

The Cathedral Peak Hotel has a 500 kVA standby generator that is derated to 450 kVA due to the high altitude. The generator only turns in during outages [C4]. Using the generator as a synchronous condenser may be a solution to the voltage-regulation problem. Used in this manner, the generator injects no active power into the network and therefore does not expend any fuel (no energy storage required). With the generator connected directly to the ac

network, reaction time is in the order of cycles and continuously variable compensation is achieved.

An alternative solution is to install shunt capacitors, which also deliver purely reactive power, but at 11 kV switching-cycles can be as long as ten minutes if the controller does not monitor the network voltage waveform. The long switching time enables the capacitors to discharge, preventing a possible voltage of double the peak network voltage across a capacitor. An intelligent controller, which monitors the voltage waveform, can be used to switch the capacitors when the network voltage equals the capacitor voltage. Such controller can reduce switching cycles to milliseconds. Switching transients introduced by the capacitor switching are also a major drawback of this method.

The exceptionally good power factor of the hotel raises suspicions about the effectiveness of shunt compensation without energy storage. PSAT will be used to investigate this solution.

Analysis equations

With no active power injected, $P_{Comp2} = 0$. P_{Comp2} can also be specified as a non-zero constant when, for example, the amount of active power needs to be limited to conserve stored energy reserves. With P_{Comp2} known in (3.17), θ_{PCCR} is the only unknown in this equation and can be solved accordingly. The following substitutions are made in (3.17) to convert the trigonometric terms into algebraic terms:

$$\cos \theta_{PCCR} = \frac{1 - T^2}{1 + T^2} \quad (3.19)$$

$$\sin \theta_{PCCR} = \frac{2T}{1 + T^2} \quad (3.20)$$

$$\text{where } T = \tan\left(\frac{\theta_{PCCR}}{2}\right) \quad (3.21)$$

The resulting equation is quadratic in T :

$$T^2 c_i + T c_j + c_k = 0 \quad (3.22)$$

$$\begin{aligned} \text{where } c_i &= |V_{PCCR}|^2 (c_c - c_g) - |V_{PCCR}| (c_a - c_e) - P_{Load} + P_{Comp2} \\ c_j &= -2|V_{PCCR}| (c_b - c_f) \\ c_k &= |V_{PCCR}|^2 (c_c - c_g) + |V_{PCCR}| (c_a - c_e) - P_{Load} + P_{Comp2} \end{aligned} \quad (3.23)$$

Solving T from (3.22) and substituting (3.21) in the result, solves θ_{PCCR} .

$$\theta_{PCCR} = 2 \arctan \left(\frac{-c_j \pm \sqrt{c_j^2 - 4c_i c_k}}{2c_i} \right) \quad (3.24)$$

For every δ the two solutions for θ_{PCCR} are back-substituted in equation (3.17) as a validity test. If both solutions are valid, the solution resulting in the lowest compensator rating (S_{comp2}) is chosen by PSAT. The user still has the option to override the software and choose the required solution manually.

Practical application

Fig. 3.14 is the circuit diagram of the compensated network. $|V_{PCCR}|$ is specified as 0.94 p.u. to limit the load on the generator and to adhere to the NRS 048 minimum specification [B4]. The network parameters are known from Fig. 3.6.

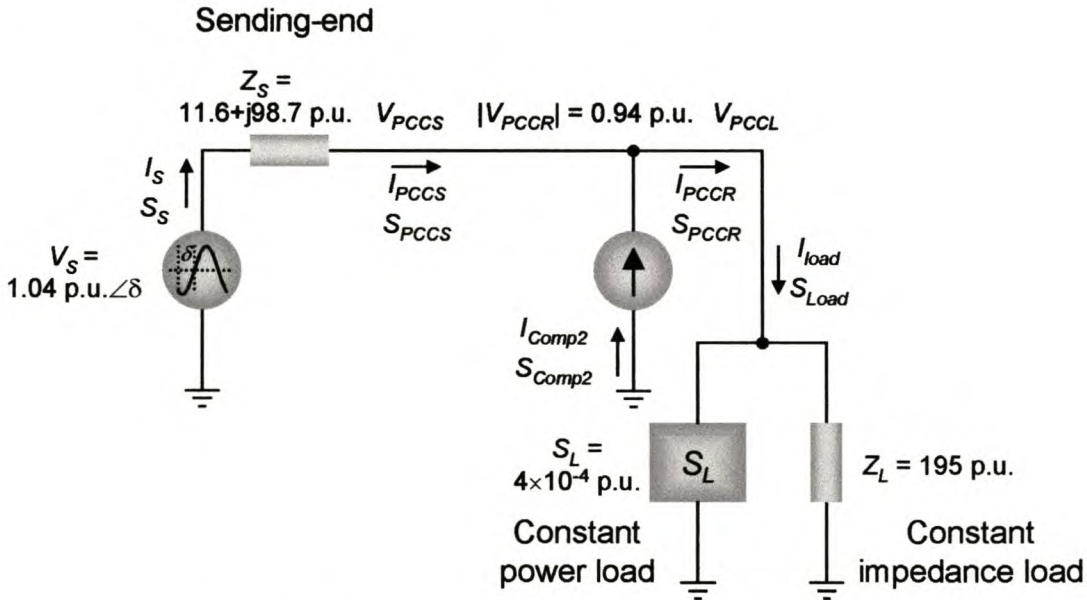


Fig. 3.14: Cathedral Peak network with a shunt compensator

The voltage diagrams generated with PSAT at $\delta = 0 \text{ rad}$ are shown in Fig. 3.15. As with the uncompensated case, two-dimensional graphs are used because the voltage, current and power across the network do not change significantly with δ . $|V_{PCCR}|$ is raised to 0.94 p.u., which is how the shunt compensator is expected to work. The sending-end voltage magnitude is the expected 1.04 p.u.

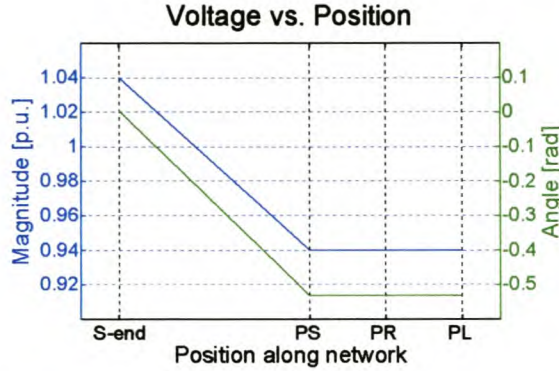


Fig. 3.15: Voltage plots for the Cathedral Peak network under shunt compensation with no energy storage at $\delta = 0$ rad

For small values of δ the voltage angle of V_{PCCR} , as well as the load current angle, is negative, as explained in Fig. 3.11. For the compensated network, the voltage angle at the PCC is even more negative, as seen in Fig. 3.15 and Fig. 3.16, both drawn at $\delta = 0$ rad. If the voltage vector diagram of Fig. 3.16 is compared to Fig. 3.11, the larger magnitude of V_{PCCR} (0.94 p.u. vs. 0.86 p.u.) is clearly visible. This is due to the voltage regulation by the shunt compensator. 90% of the load is a constant impedance load, therefore the larger load voltage necessitates a larger $|I_{PCCR}|$, as seen when the current vector diagrams in Fig. 3.16 (b) and Fig. 3.11 (b) are compared (5.25×10^{-3} p.u. vs. 4.89×10^{-3} p.u.). The larger I_{PCCR} combined with I_{Comp2} result in a larger $|I_{PCCS}|$ (5.36×10^{-3} p.u. vs. 4.89×10^{-3} p.u.). For the uncompensated case $I_{PCCS} = I_{PCCR}$. The larger I_{PCCS} flow through Z_S , resulting in a larger voltage drop across Z_S ($I_{PCCS}Z_S$), thereby increasing the negative phase angle of V_{PCCR} for small values of δ . This is evident in Fig. 3.16 (a).

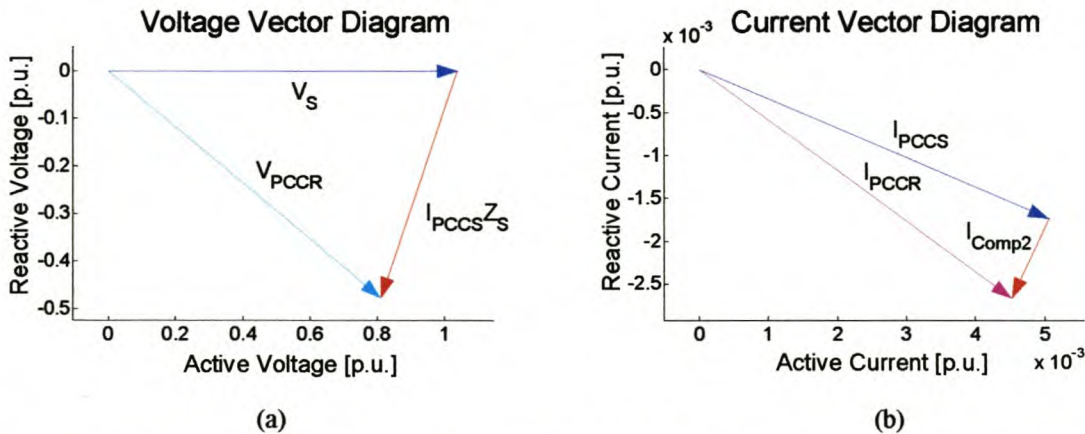


Fig. 3.16: Voltage and current vector diagrams for Cathedral Peak network with shunt compensation without energy storage at $\delta = 0$ rad

The current graphs for this network are shown in Fig. 3.17. The step in the current magnitude and angle at PCCR is due to the current injected by the compensator. It must be remembered that the step at PCCR in the magnitude and angle graphs is not directly an indication of the angle and magnitude of I_{Comp2} . The true magnitude and angle of I_{Comp2} are given by (3.25), which shows that $\theta_{I_{Comp2}} \neq \theta_{I_{PCCR}} - \theta_{I_{PCCS}}$ and $|I_{Comp2}| \neq |I_{PCCR}| - |I_{PCCS}|$, except when $I_{PCCR} = 0$, $I_{PCCS} = 0$ or $I_{PCCR} = I_{PCCS}$.

$$\begin{aligned}
 I_{Comp2} &= I_{PCCR} - I_{PCCS} \\
 &= |I_{PCCR}| \angle \theta_{I_{PCCR}} - |I_{PCCS}| \angle \theta_{I_{PCCS}} \\
 \therefore |I_{Comp2}| &= \sqrt{|I_{PCCR}|^2 + |I_{PCCS}|^2 - 2|I_{PCCR}||I_{PCCS}|\cos(\theta_{I_{PCCR}} - \theta_{I_{PCCS}})} \\
 \theta_{I_{Comp2}} &= \arctan\left(\frac{|I_{PCCR}|\sin\theta_{I_{PCCR}} - |I_{PCCS}|\sin\theta_{I_{PCCS}}}{|I_{PCCR}|\cos\theta_{I_{PCCR}} - |I_{PCCS}|\cos\theta_{I_{PCCS}}}\right)
 \end{aligned} \tag{3.25}$$

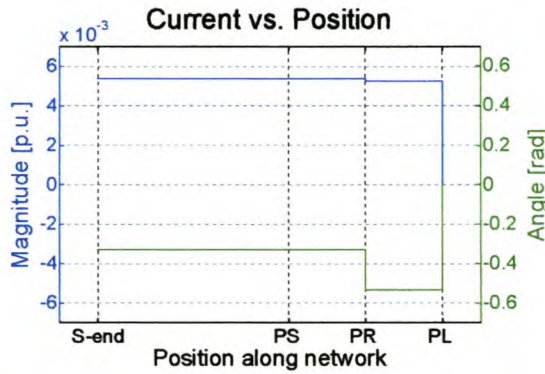


Fig. 3.17: Current magnitude and angle for shunt compensation with $P_{Comp2} = 0$ p.u. at $\delta = 0$ rad
 PSAT provides for this need by supplying extra two-dimensional graphs for the compensator and the load. These graphs portray the true values of the voltages and currents represented by steps in the magnitude and angle graphs. The true magnitudes and angles of V_{Comp2} (V_{PCCR}) and I_{Comp2} are shown in Fig. 3.18.

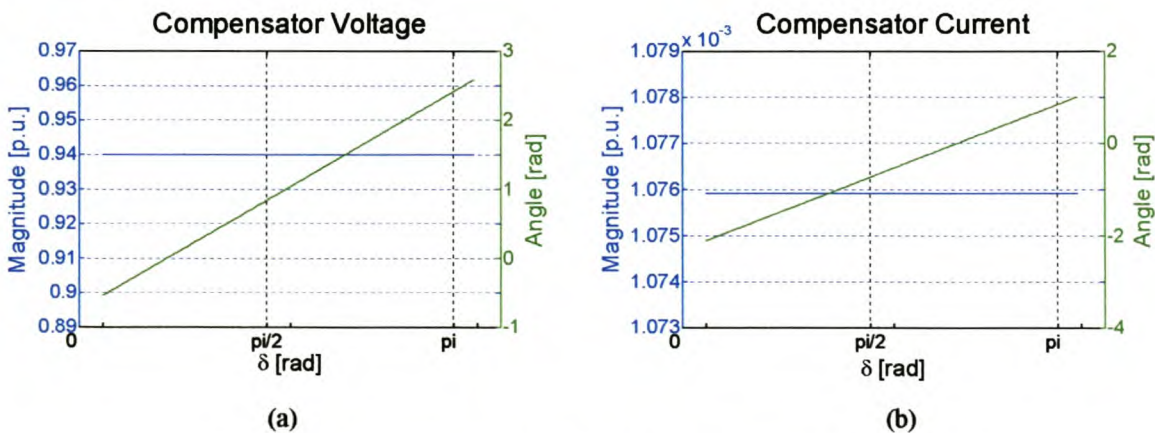


Fig. 3.18: Shunt compensator voltage and current for $P_{Comp2} = 0$

The current angle lags behind the voltage angle with $\pi/2$ rad for all δ . This corresponds to the requirement that the compensator can only deliver reactive power. With the phase relationship known, the product of the current and voltage magnitudes will give the power injected by the compensator:

$$\begin{aligned}
 S_{Comp2} &= V_{Comp2} \times I_{Comp2}^* \\
 &= |V_{Comp2}| |I_{Comp2}| \angle (\theta_{V_{Comp2}} - \theta_{I_{Comp2}}) \\
 &= (0.94 \times 1.076 \times 10^{-3}) \angle \frac{\pi}{2} \\
 &= j1.01 \times 10^{-3} \text{ p.u.} \\
 &= 101 \text{ kVar}
 \end{aligned} \tag{3.26}$$

Since I_{PCC_L} is zero in Fig. 3.12, the load current can be read directly from Fig. 3.17. At $\delta = 0$ rad, $I_{Load} = 5.25 \times 10^{-3}$ p.u. $\angle -0.336$ rad. The negative angle is due to the negative angle of I_{PCCR} , which is explained in Fig. 3.16. From Fig. 3.14, $I_{Load} = I_{PCCR}$.

The compensator injects no active power, as the absence of a step at PCCR in the active power graph in Fig. 3.19 indicates. The compensator injects 1.01×10^{-3} p.u. reactive power, which translates to 101 kVar and corresponds to the calculated value in (3.26). This is well within the capability of the 450-kVA generator of the hotel. Operated as a synchronous condenser, the generator can rapidly (in less than a second) supply the required reactive power.

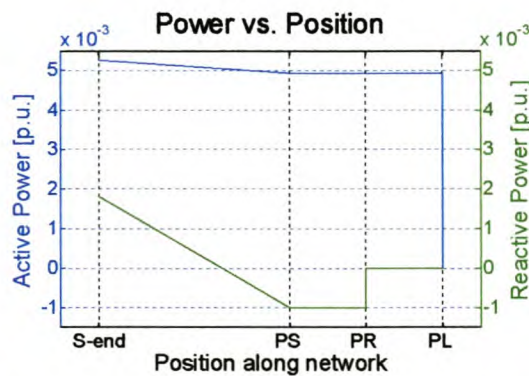


Fig. 3.19: Power transfer under shunt compensation with $P_{Comp2} = 0$

The power transfer capability of the network increased, as expected, with the injection of reactive power. Due to the increased $|V_{PCCR}|$, the largely constant impedance load draws more power (4.93×10^{-3} p.u. vs. 4.23×10^{-3} p.u. for the uncompensated load). The power absorbed by Z_S increased from $(0.279 + j2.37) \times 10^{-3}$ p.u. to $(0.333 + j2.83) \times 10^{-3}$ p.u. due to the larger

current demanded by the load and the compensator ($I_{PCCS} = 5.36 \times 10^{-3}$ p.u. vs. $I_{PCCS} = 4.89 \times 10^{-3}$ p.u. for the uncompensated load). It seems that a higher current through Z_S must worsen the voltage regulation at PCCR, but the current angle is such that the voltage drop across Z_S is reduced, as seen in Fig. 3.15. The result is a higher power transfer capability across Z_S .

At 101 kVar compensating power, this is a very effective solution, which denies the reservations expressed at the start of this section. PSAT can therefore serve a great purpose in the quick and effective analysis of seemingly ineffective solutions. The cost involved to implement a purely reactive shunt compensator will be that of modifying the generator to function as a synchronous condenser. As with all solutions by network compensation, this option does not result in the versatility of feeding the Cathedral Peak network from two sources. Building a new line does provide this added security. Since the hotel still has the standby generator, it is not of great importance to the load.

3.3.2 Shunt compensation with reactive power specified

The exceptionally good power factor and large effective line resistance feeding the hotel suggest that a purely active compensator would probably be a very good choice. In this case Q_{Comp2} , the amount of reactive power injected by the compensator, is specified as a control parameter. θ_{PCCR} is then solved from (3.18) since Q_{Comp2} is known.

Eskom proposed using the hotel generator to solve the voltage-regulation problem, thereby strengthening the network through distributed generation. The voltage-regulation problem exists only during peak times. Peak times are typically in the winter months over weekends when a nursery and hospital along the line have the highest electrical heating demands. Although not currently a problem, tourist seasons may also present peak demands in the future. Under these circumstances, the life-cycle cost of the generator is approximately 50% lower than the cost of new infrastructure [C4].

Used in this manner, the distributed generator is analysed as a shunt compensator inserted at PCCR. Q_{Comp2} is chosen as zero to model the generator when it is working at unity power factor, i.e. the generator supplies only active power. With the generator connected directly to the ac network, reaction time is in the order of cycles but continuously variable compensation is achieved.

Analysis equations

With Q_{Comp2} known, θ_{PCCR} is the only unknown in (3.18) and can be solved accordingly. For easy reference, equation (3.18) is repeated in (3.27):

$$\begin{aligned} &|V_{PCCR}|^2(c_d - c_h) + |V_{PCCR}| \cos \theta_{PCCR} (c_b - c_f) \\ &+ |V_{PCCR}| \sin \theta_{PCCR} (c_a - c_e) - Q_{Load}' + Q_{Comp2} = 0 \end{aligned} \quad (3.27)$$

The substitutions made in the previous section (equations (3.19) to (3.21)) are repeated for equation (3.27), resulting in (3.28):

$$T^2 c_1 + T c_m + c_n = 0 \quad (3.28)$$

$$\begin{aligned} \text{where } c_1 &= |V_{PCCR}|^2(c_d - c_h) + |V_{PCCR}|(c_b - c_f) - Q_{Load}' \\ c_m &= 2|V_{PCCR}|(c_a - c_e) \\ c_n &= |V_{PCCR}|^2(c_d - c_h) + |V_{PCCR}|(c_b - c_f) - Q_{Load}' \end{aligned} \quad (3.29)$$

Solving T from (3.28) and substituting (3.21) in the result solves θ_{PCCR} .

$$\theta_{PCCR} = 2 \arctan \left(\frac{-c_m \pm \sqrt{c_m^2 - 4c_1 c_n}}{2c_1} \right) \quad (3.30)$$

For every δ PSAT tests the validity of the two solutions for θ_{PCCR} by substituting the solutions back into equation (3.27). The software chooses the solution that results in the lowest compensator rating (S_{comp2}) if both solutions are valid. The user still has the option to override the software and choose the required solution manually. With $|V_{PCCR}|$, θ_{PCCR} and all the network parameters known, the circuit can be solved with Ohm's Law and Kirchoff's Current and Voltage Laws.

Practical application

Fig. 3.14, the circuit diagram of the shunt-compensated network, is repeated in Fig. 3.20 for easy reference. $|V_{PCCR}|$ is specified as 0.94 p.u. to limit the load on the generator and to adhere to the NRS 048 minimum specification of 0.9 p.u. for <500 V networks [B4]. V_S , Z_S , Z_L , and S_L are known from Fig. 3.6.

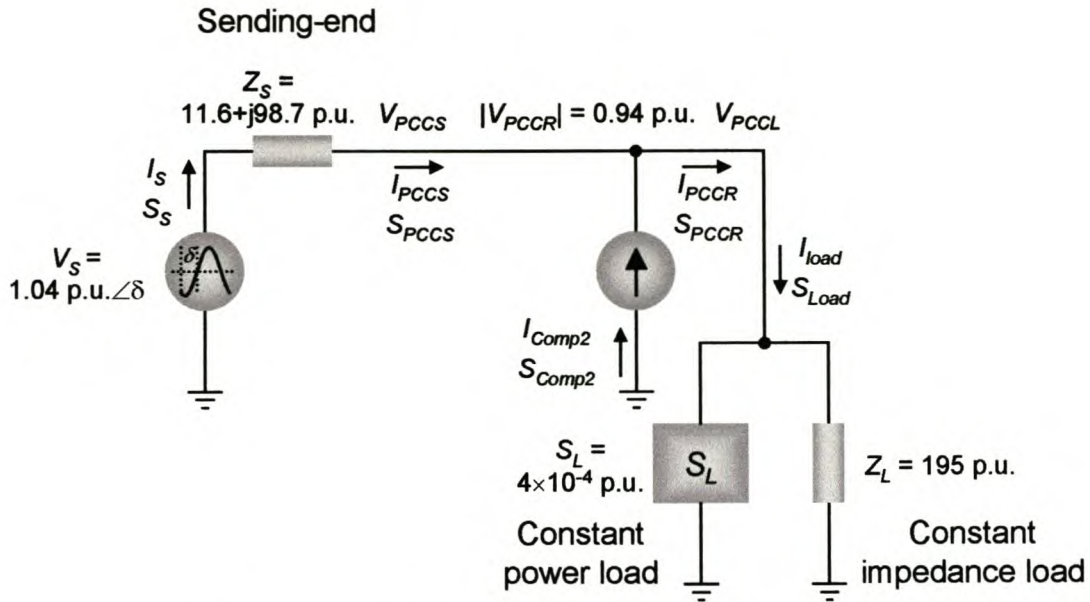


Fig. 3.20: Cathedral Peak network with a shunt compensator

The desired voltage regulation of 0.94 p.u. is realised, as shown in the voltage plots of Fig. 3.21. The change in voltage angle across Z_S is nearly imperceptible because the reactance of Z_S is less than 5% of the combined resistance of Z_S and Z_R .

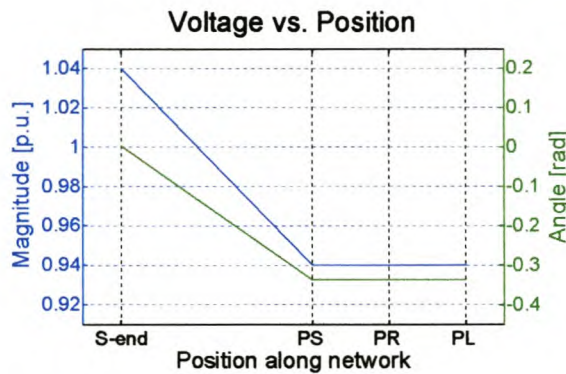


Fig. 3.21: Voltage magnitude and angle under shunt compensation with $Q_{Comp2} = 0$

The voltage drop across Z_S is therefore reduced from 0.486 p.u. $\angle 0.970$ rad to 0.153 p.u. $\angle 0.310$ rad. The negative voltage angle at PCCR for small δ is similar to the uncompensated case.

The shunt compensator is located at PCCR and the load at PCCL. The compensator injects no reactive power as the absence of a step in the reactive power graph indicates. 1.63×10^{-3} p.u. active power is injected at PCCR. A device rating of 163 kVA (163 kW) is therefore required for this control strategy. This corresponds to the device rating obtained in the Eskom studies [C4]. The 450 kVA generator of the hotel is more than capable of

supplying this power. With the unity power factor of the load and no reactive power compensation, Z_S is the only reactive power drain on the network. Z_S draws 1.21×10^{-3} p.u. reactive power and 0.143×10^{-3} p.u. active power, which results in Z_S requiring only 51.2% of the apparent power it required in the uncompensated network. The reduced voltage drop across Z_S accounts for the reduced power consumption.

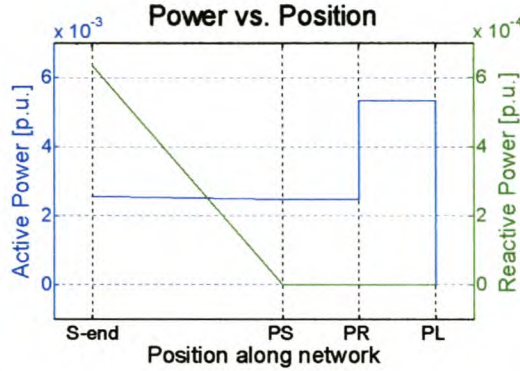


Fig. 3.22: Power transfer under shunt compensation with $Q_{Comp2} = 0$

The expected load power for the compensated case is calculated below as 493 kW. The step of 4.93×10^{-3} p.u. at PCCL in Fig. 3.22 corresponds to the calculated value. The above findings serve as further proof of the validity of the equations implemented in PSAT.

$$\begin{aligned}
 S_{Load} &= \frac{|V_{PCCR}|^2}{Z_{Load}} + S_{Const.Load} \\
 &= \frac{|0.94|^2}{195} + 0.0004 \\
 &= 4.93 \times 10^{-3} \text{ p.u. } \angle 0 \text{ rad} \\
 &= 493 \text{ kW}
 \end{aligned}$$

With the power constant for all δ , the current angle behaviour is expected to be similar to that of the voltage angle. This is indeed the case in Fig. 3.23. The current magnitude is expected to display the compensator current as a step at PCCL and the load current as a step at PCCR. Again, Fig. 3.23 complies. The current angle shows no step at PCCR. From equation (3.25), this indicates that the compensator current angle is the same as the angles of I_{PCCS} and I_{PCCR} . This angle must equal the angle of V_{PCCR} for the compensator to inject purely active power. This is indeed the case when Fig. 3.21 is compared to Fig. 3.23. As with the voltage angle offset, the current angle offset corresponds to the uncompensated case.

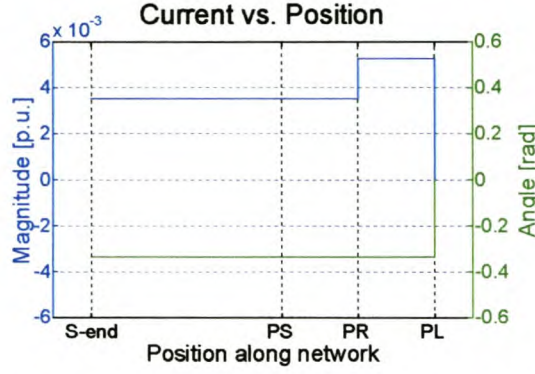


Fig. 3.23: Current magnitude and angle in the shunt-compensated Cathedral Peak network

The purely active compensator requires only 163 kW compensation power. The 450 kVA standby generator of the hotel can easily supply this demand. Costs are therefore limited to diesel used by the generator under peak network loads and the capital cost of adapting the generator to function as a distributed resource.

3.3.3 Minimum shunt compensator rating

The extra degree of freedom introduced by the shunt compensator can also be controlled by minimising the apparent power delivered by the compensator, i.e. $|S_{Comp2}|$. The result is a compensator of minimum rating.

Analysis equations

$|S_{Comp2}|$ is given by (3.31):

$$|S_{Comp2}| = \sqrt{P_{Comp2}^2 + Q_{Comp2}^2} \quad (3.31)$$

Differentiating (3.31) with respect to θ_{PCCR} and setting the result to zero produces an equation in P_{Comp2} and Q_{Comp2} .

$$\begin{aligned} \frac{\partial |S_{Comp2}|}{\partial \theta_{PCCR}} &= \frac{2P_{Comp2} \frac{\partial P_{Comp2}}{\partial \theta_{PCCR}} + 2Q_{Comp2} \frac{\partial Q_{Comp2}}{\partial \theta_{PCCR}}}{2\sqrt{P_{Comp2}^2 + Q_{Comp2}^2}} = 0 \\ \therefore P_{Comp2} \frac{\partial P_{Comp2}}{\partial \theta_{PCCR}} + Q_{Comp2} \frac{\partial Q_{Comp2}}{\partial \theta_{PCCR}} &= 0 \end{aligned} \quad (3.32)$$

Equations (3.17) and (3.18) are substituted in (3.32).

$$c_r \cos \theta_{PCCR} + c_s \sin \theta_{PCCR} = 0 \quad (3.33)$$

$$\begin{aligned} \text{where } c_r &= |V_{PCCR}|^2 [(c_a - c_e)(c_d - c_h) - (c_b - c_f)(c_c - c_g)] + P_{Load}(c_b - c_f) - Q_{Load}(c_a - c_e) \\ c_s &= |V_{PCCR}|^2 [(c_a - c_e)(c_c - c_g) + (c_b - c_f)(c_d - c_h)] + P_{Load}(c_a - c_e) + Q_{Load}(c_b - c_f) \end{aligned}$$

θ_{PCCR} is now the only unknown in the resulting equation and can therefore be solved:

$$\theta_{PCCR} = \arctan(\pm c_r, c_s) \quad (3.34)$$

Equation (3.34) uses the four-quadrant arc tangent, resulting in two solutions for θ_{PCCR} . The analysis software tests the solutions for validity and then chooses the solution that results in the lowest $|S_{Comp2}|$.

Practical application

Applying this control technique to the Cathedral Peak generator requires modifications to the generator to control its power factor and a controller to maintain minimum generator rating. The generator can operate down to a power factor of 0.85.

For easy reference the circuit diagram of the shunt-compensated network is repeated in Fig. 3.24. As in the previous cases, $|V_{PCCR}|$ is specified as 0.94 p.u. and V_S , Z_S , Z_L , and S_L are known from Fig. 3.6.

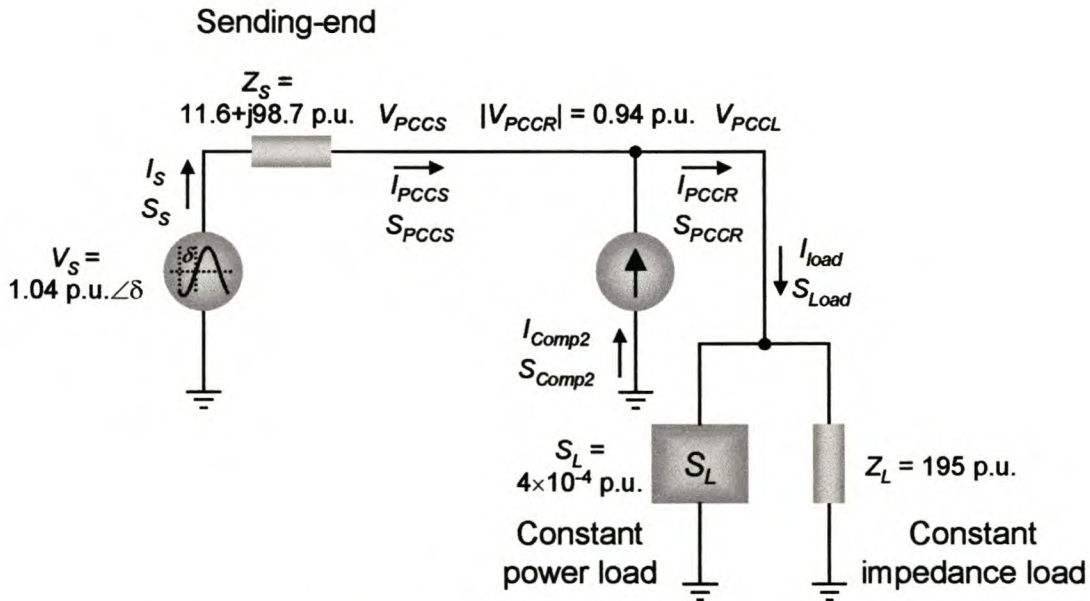


Fig. 3.24: Cathedral Peak network with a shunt compensator

The voltage regulation graphs in Fig. 3.25 correspond to the previous two cases for shunt compensation. $|V_{PCCR}|$ is maintained at 0.94 p.u., as desired, and the voltage drop across Z_S is reduced to 0.475 p.u. $\angle 1.13 \text{ rad}$, compared to 0.486 p.u. $\angle 0.970 \text{ rad}$ for the uncompensated case.

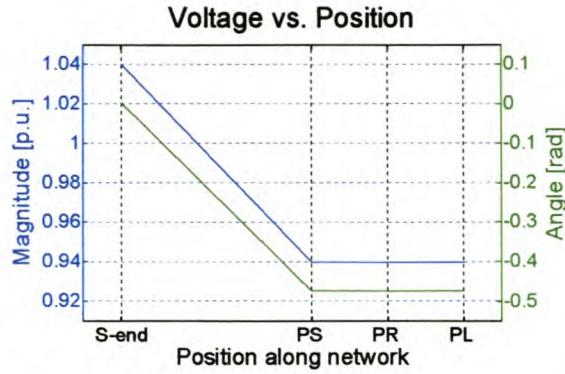


Fig. 3.25: Voltage graphs for the Cathedral Peak network with a minimum shunt compensator

The power transfer graphs in Fig. 3.26 indicate an active power injection of 0.460×10^{-3} p.u. and reactive power compensation of 0.681×10^{-3} p.u. This translates to a compensator rating of 0.822×10^{-3} p.u. The minimum-rating compensator requires a rating of 82.2 kVA and 46 kW, respectively. Compared to a purely active power compensator, this is a large reduction of 77.8 kVA and 114 kW (71%), but 68.1 kVar reactive power is required. Due to the increased $|V_{PCCR}|$, the load power increased to 4.93×10^{-3} p.u., as in the previous shunt-compensated cases.

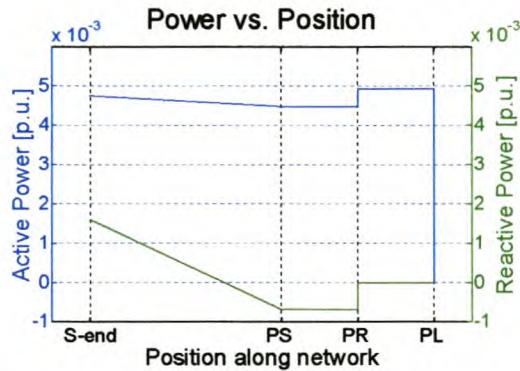


Fig. 3.26: Power transfer for minimum shunt compensator

The trade-off is the cost of the fuel saved by reducing the output power and the capital investment of providing the equipment to deliver the required reactive power for the minimum-rating compensator. The reactive power can be sourced from modifying the generator or installing adequate capacitor banks. The reaction time of the capacitor banks (if switchable) will be orders of magnitude slower than the generator reaction time. The generator can operate down to a power factor of 0.85. Minimum compensation requires an operating power factor of 0.56. A combination of capacitor banks and the distributed generator is therefore required for this solution. The capacitors can supply the bulk of the required reactive power, while the generator can rapidly compensate with active power and

the balance of the required reactive power. A further advantage of this scenario is that the generator will probably suffer less armature heating due to the lower power factor [B5].

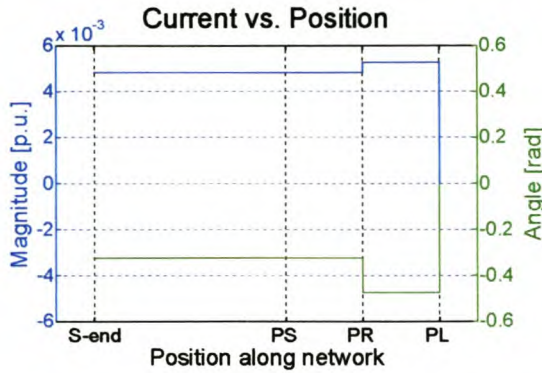


Fig. 3.27: Current graphs for the Cathedral Peak network with minimum shunt compensator

The current graphs in Fig. 3.27 are similar to the current graphs of the purely reactive shunt compensator in section 3.3.1. As discussed previously, the compensator current magnitude and angle cannot be read directly from the steps at PCCR in the above current graphs. The compensator current graphs in Fig. 3.28 portray the true values of the compensator current magnitude and angle for all δ .

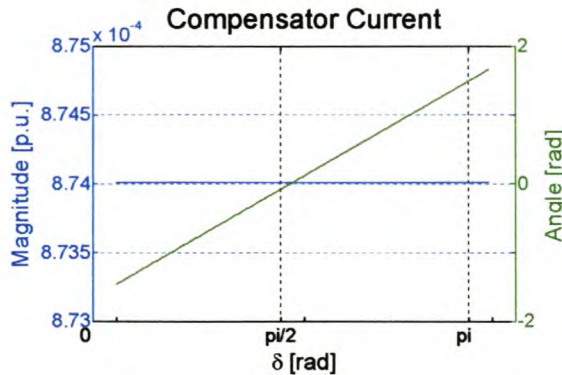


Fig. 3.28: Current graphs for the minimum shunt compensator

The constant load fed by a radial line requires a constant compensator current magnitude for all δ . The increasing voltage angle across the network (δ) necessitates θ_{Load2} to increase from -1.45 to 1.66 rad to maintain a constant offset with respect to δ .

As in the previous section, the reactive power supplied by the compensator, namely Q_{Comp2} , can be used as a control parameter. Q_{Comp2} is reduced from 0.681×10^{-3} p.u. (which resulted in a minimum compensator rating) to increase the power factor. By trial and error the value of Q_{Comp2} is determined that results in a power factor of 0.85 for the generator. This is the

operating point where the compensator rating is a minimum without needing additional capacitor banks. The power graphs for this scenario are displayed in Fig. 3.29.

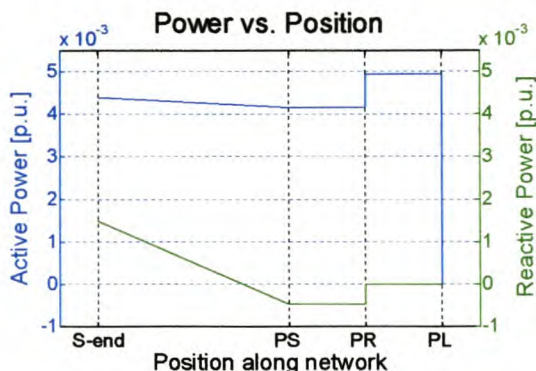


Fig. 3.29: Power transfer for minimum shunt compensator without additional capacitor banks

The value required for Q_{Comp2} is 0.48×10^{-3} p.u. to maintain a power factor of 0.85. This translates into a compensator rating of 48 kVar and 76.9 kW. The resulting apparent power rating is 90.7 kVA, which is a mere 10.3% higher than the minimum compensator rating. The active power rating is 67% higher than the minimum power compensator but is still 47% smaller than the purely active power compensator at 163 kW. The higher rating is a small price to pay when it is considered that no additional infrastructure in terms of capacitor banks is required. The reduced power factor will also benefit the generator as armature heating is reduced [B5].

3.3.4 Summary of shunt compensation

All the above control strategies regulated the voltage angle of V_{PCCR} by means of the shunt compensator. The voltage angle does not influence the load in any way but the compensator rating is a function of this angle, as demonstrated in this section. In all the voltage angle graphs in section 3.3 the phase angle of the sending-end voltage source, δ , only introduced an offset in the voltage angle graph. It did not influence the relative voltage angles across the network. The conclusion reached from this observation is that it is not the absolute voltage angle at PCCR that determines the compensator rating, but the relative voltage angle between the sending end and PCCR. As mentioned for the uncompensated case, the Cathedral Peak network is a special case with no receiving-end voltage source. This conclusion is therefore only valid for this special case. Chapter 5 explores this idea further as it is to be implemented in future versions of PSAT.

The minimum shunt compensator requires the lowest apparent compensation power, but requires additional capacitors as the generator cannot operate at a power factor as low as 0.56. Operating costs will therefore be lower than the purely active compensator, but higher than the purely reactive power compensator. On the other hand, capital costs for capacitor banks and a controller for the generator will be less than the purely reactive power compensator, which requires more capacitor banks. The purely active compensator requires no capacitors and is therefore the most cost effective concerning capital costs. The best solution is to operate the generator at its minimum power factor of 0.85. This keeps running costs down and the only capital expenditure required is for modifying the generator to control its power factor.

3.4 SERIES VOLTAGE REGULATORS

Series compensators are widely recognised as voltage regulators [B2]. A series compensator injects a voltage in series with the network and therefore has a small voltage rating relative to the network voltage. The full load current flows through the compensator, which can be problematic under fault conditions. Series compensators can effectively compensate for poor voltage regulation when the load impedance or Z_R is large compared to Z_S . This renders the series compensator ideal for voltage regulation for loads at the end of the line. Fig. 3.12 is the circuit diagram for any radial network with a series compensator at the PCC.

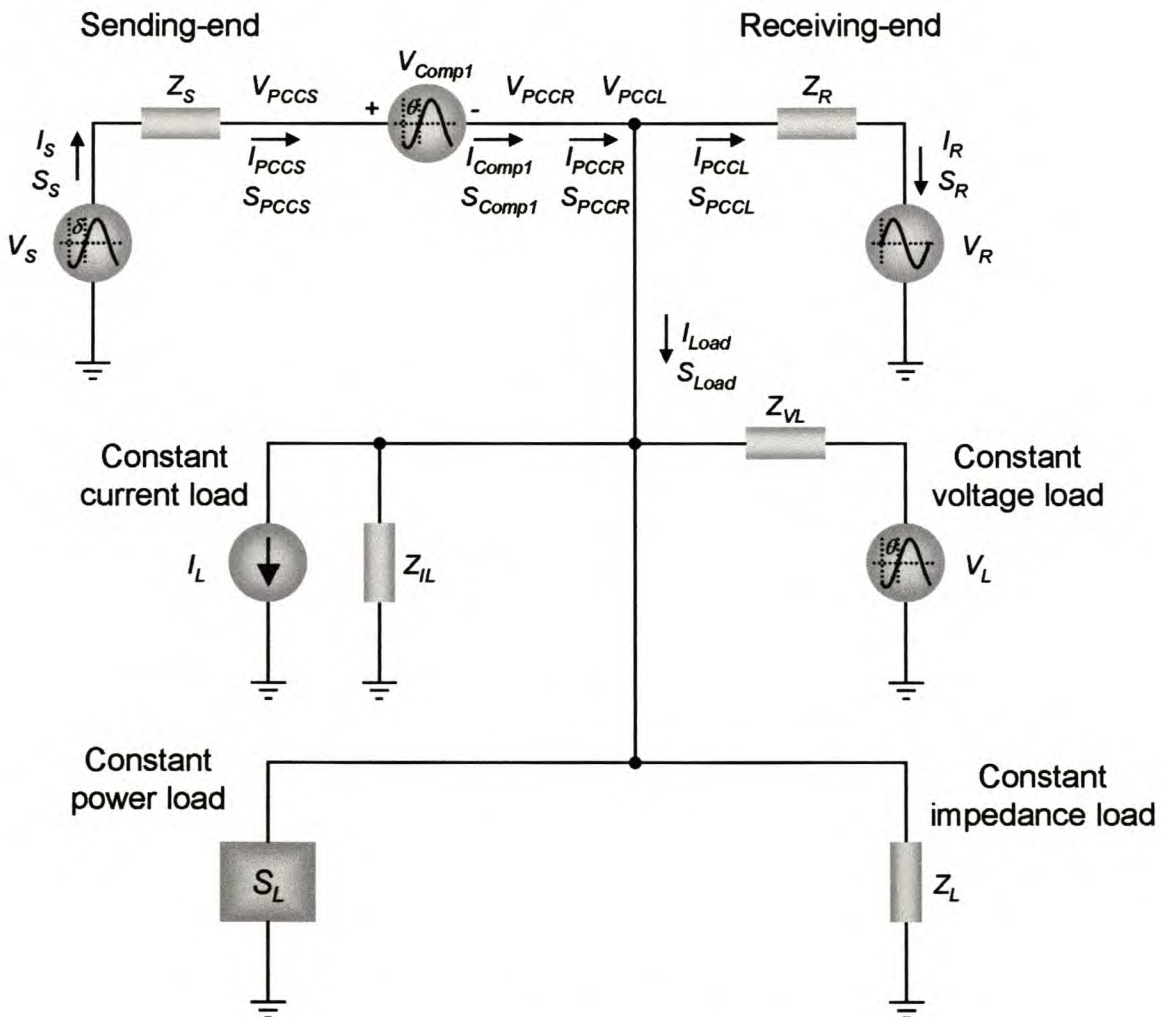


Fig. 3.30: Series voltage regulator

As in the previous cases, the above network is simplified, without any approximations, to the circuit in Fig. 3.31. Equations are now derived for PSAT to analyse voltage regulation under series compensation. The equations for Z_R' and V_R' are given by equations (3.5) and (3.6) in section 3.2.3. From section 3.2.2, V_S , Z_S and S_L are known network parameters. $|V_{PCCR}|$ is

specified by the network planner. As in the uncompensated and shunt-compensated cases, V_{PCCR} and V_{PCCL} are equivalent, but for the series-compensated network V_{PCCS} is given by (3.35).

$$V_{PCCS} = V_{PCCR} + V_{Comp1} \quad (3.35)$$

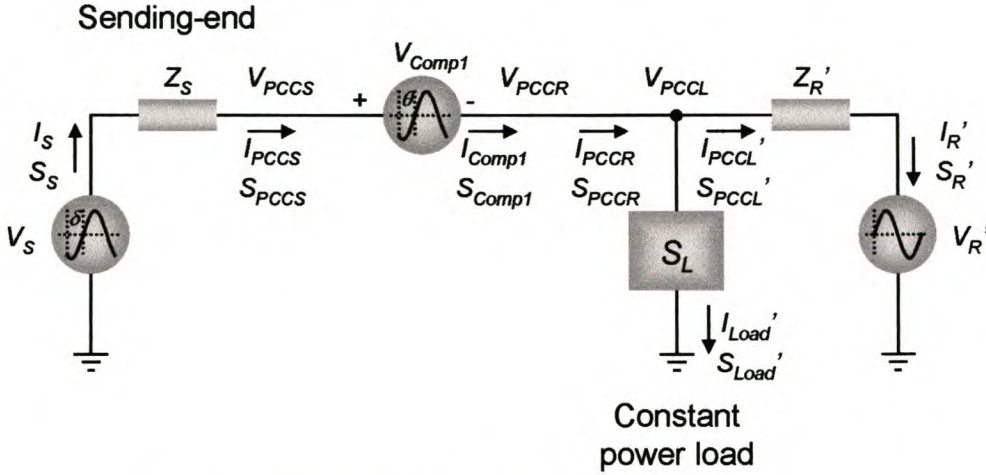


Fig. 3.31: Simplified series-compensated network

The objective is again to solve θ_{PCCR} ; thereafter all other unknowns can easily be calculated from Ohm's Law and Kirchoff's Current and Voltage Laws. I_{Comp1} is given by (3.36) and V_{Comp1} follows from (3.35).

$$\begin{aligned} I_{Comp1} &= I_{PCCL}' + I_{Load}' \\ &= \frac{V_{PCCR} - V_R}{Z_R} + \frac{S_{Load}^*}{V_{PCCR}^*} \end{aligned} \quad (3.36)$$

$$\begin{aligned} V_{Comp1} &= V_{PCCR} - V_{PCCS} \\ &= V_{PCCR} - (V_S - I_{Comp1} Z_S) \end{aligned} \quad (3.37)$$

S_{Comp1} can be calculated from the above two equations:

$$\begin{aligned} S_{Comp1} &= V_{Comp1} I_{Comp1}^* \\ &= (V_{PCCR} - V_S) \left(\frac{V_{PCCR} - V_R}{Z_R} + \frac{S_{Load}^*}{V_{PCCR}^*} \right)^* \\ &\quad + Z_S \left(\frac{V_{PCCR} - V_R}{Z_R} + \frac{S_{Load}^*}{V_{PCCR}^*} \right) \left(\frac{V_{PCCR} - V_R}{Z_R} + \frac{S_{Load}^*}{V_{PCCR}^*} \right)^* \end{aligned} \quad (3.38)$$

Equation (3.38) is multiplied by $|V_{PCCR}|^2$ and set to zero. Equations (3.39) and (3.40) are the respective real and imaginary parts of the resulting equation. There are three unknowns in these two equations, namely θ_{PCCR} , P_{Comp1} and Q_{Comp1} .

$$c_t \cos \theta_{PCCR} + c_u \sin \theta_{PCCR} + c_v P_{Comp1} + c_w = 0 \quad (3.39)$$

$$c_x \cos \theta_{PCCR} + c_y \sin \theta_{PCCR} + c_z Q_{Comp1} + c_{aa} = 0 \quad (3.40)$$

where

$$c_t = \left[\begin{array}{l} |V_{PCCR}|^3 \operatorname{Re} \left(-\frac{V_R^* Z_R + V_R^* Z_S + V_S Z_R + V_R Z_S}{|Z_R|^2} \right) \\ + |V_{PCCR}| \operatorname{Re} \left(-V_S S_{Load} - \frac{V_R^* Z_S Z_R S_{Load}^* + V_R Z_S Z_R^* S_{Load}}{|Z_R|^2} \right) \end{array} \right]$$

$$c_u = \left[\begin{array}{l} |V_{PCCR}|^3 \operatorname{Im} \left(\frac{V_R^* Z_R + V_R^* Z_S - V_S Z_R - V_R Z_S}{|Z_R|^2} \right) \\ + |V_{PCCR}| \operatorname{Im} \left(-V_S S_{Load} + \frac{V_R^* Z_S Z_R S_{Load}^* - V_R Z_S Z_R^* S_{Load}}{|Z_R|^2} \right) \end{array} \right]$$

$$c_v = -|V_{PCCR}|^2$$

$$c_w = |V_{PCCR}|^4 \operatorname{Re} \left(\frac{Z_R + Z_S}{|Z_R|^2} \right) + \operatorname{Re} \left(|S_{Load}|^2 Z_S \right) \\ + |V_{PCCR}|^2 \operatorname{Re} \left(S_{Load} + \frac{Z_S Z_R^* S_{Load} + V_S V_R^* Z_R + |V_R|^2 Z_S + Z_S Z_R S_{Load}^*}{|Z_R|^2} \right)$$

and

$$c_x = \left[\begin{array}{l} |V_{PCCR}|^3 \operatorname{Im} \left(-\frac{V_R^* Z_R + V_R^* Z_S + V_S Z_R + V_R Z_S}{|Z_R|^2} \right) \\ + |V_{PCCR}| \operatorname{Im} \left(-V_S S_{Load} - \frac{V_R^* Z_S Z_R S_{Load}^* + V_R Z_S Z_R^* S_{Load}}{|Z_R|^2} \right) \end{array} \right]$$

$$c_y = \left[\begin{array}{l} -|V_{PCCR}|^3 \operatorname{Re} \left(\frac{V_R^* Z_R + V_R^* Z_S - V_S Z_R - V_R Z_S}{|Z_R|^2} \right) \\ -|V_{PCCR}| \operatorname{Re} \left(-V_S S_{Load} + \frac{V_R^* Z_S Z_R S_{Load}^* - V_R Z_S Z_R^* S_{Load}}{|Z_R|^2} \right) \end{array} \right]$$

$$c_z = -|V_{PCCR}|^2$$

$$c_{aa} = |V_{PCCR}|^4 \operatorname{Im} \left(\frac{Z_R + Z_S}{|Z_R|^2} \right) + \operatorname{Im} \left(|S_{Load}|^2 Z_S \right) \\ + |V_{PCCR}|^2 \operatorname{Im} \left(S_{Load} + \frac{Z_S Z_R^* S_{Load} + V_S V_R^* Z_R + |V_R|^2 Z_S + Z_S Z_R S_{Load}^*}{|Z_R|^2} \right)$$

As with the shunt compensator, θ_{PCCR} , P_{Comp1} and Q_{Comp1} can only be solved if the two above equations are supplemented by an extra equation. This equation specifies the control strategy of the extra degree of freedom added by energy storage, as discussed in section 2.4.1. The same three control strategies used for the shunt compensator are dealt with here are once again: active power specified, reactive power specified and a device of minimum-rating.

The standby generator of the Cathedral Peak Hotel cannot be used as a series compensator by itself. It can, however, be used as an energy source connected through a regulator to the dc bus of a transistor-based series device. Depending on the rating of the device, this can be a very costly option. Series capacitors are a viable alternative to shunt compensation due to their relatively low cost. Of course, depending on the controller, series capacitors are also limited by their switching time, as discussed in section 3.3.1. Increasing in cost and complexity, a TCSC or a transistor-based device, namely a Static Synchronous Series Compensator (SSSC), can also be used as a voltage regulator [B2]. The TCSC can supply only reactive power with a reaction time of half a cycle, while the SSSC, if equipped with energy storage, can supply both active and reactive power in less than a millisecond.

In the previous sections the shunt compensator proved to be a very good solution to the Cathedral Peak voltage-regulation problem. With the aid of PSAT, the effectiveness of a series compensator is also evaluated.

3.4.1 Series compensator with active power specified

A purely reactive series compensator, like a series capacitor or a TCSC, injects no active power and is modelled by setting $P_{Comp1} = 0$. If the active power transfer for an SSSC is limited due to a limited stored energy, for instance, P_{Comp1} can be set to limit the active power transfer to a certain level.

Analysis equations

With P_{Comp1} known in (3.39), θ_{PCCR} is solved in the same manner as with the purely reactive shunt compensator. Substituting (3.19) to (3.21) into (3.39), θ_{PCCR} is solved as:

$$\theta_{PCCR} = 2 \arctan \left(\frac{c_u \mp \sqrt{c_u^2 - c_w^2 + c_t^2}}{c_u - c_t} \right) \quad (3.41)$$

PSAT tests the validity of the two solutions for every δ . If a solution does not satisfy (3.39) it is deemed invalid. Should both solutions be valid, the solution that results in the lowest compensator rating (S_{comp1}) is chosen by PSAT. As before, the user still has the option to choose the required solution manually.

Practical application

Fig. 3.14 is the now familiar circuit diagram of the compensated network. $|V_{PCCR}|$ is again specified as 0.94 p.u. to limit the load on the compensator and to adhere to the NRS 048 minimum specification [B4]. The network parameters are known from Fig. 3.6.

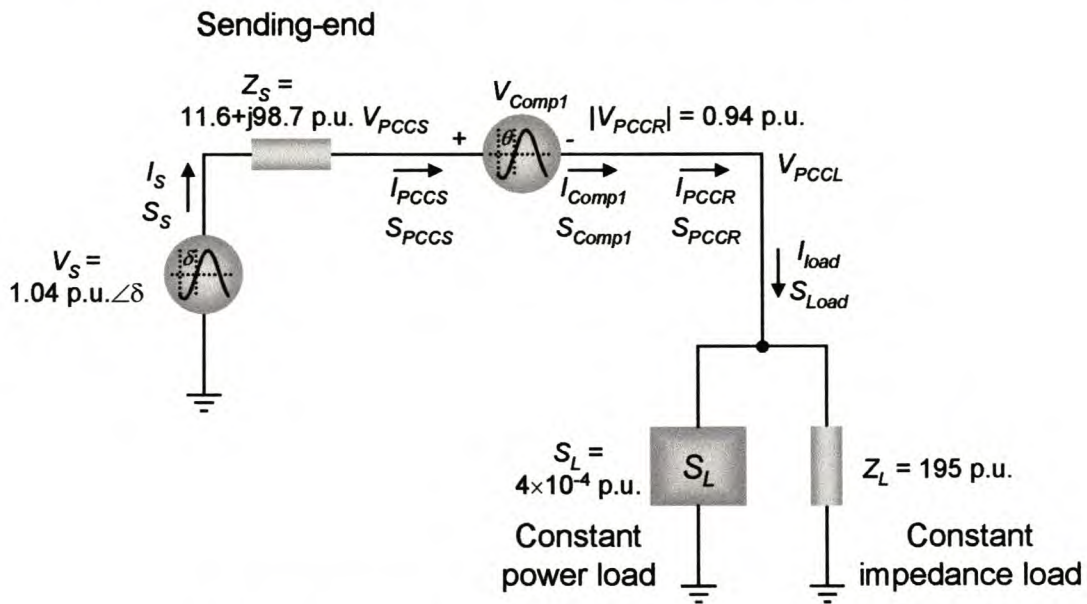


Fig. 3.32: Cathedral Peak network with series compensation

The voltage diagrams generated with PSAT are shown in Fig. 3.33 (a) at $\delta = 0$ rad. $|V_{PCCR}|$ is raised to 0.94 p.u., which is how the series compensator is expected to work. The sending-end voltage magnitude is the expected 1.04 p.u. The voltage magnitude falls across Z_s to 0.969 at PCCS. The voltage magnitude falls further across the series compensator to 0.94 p.u. at PCCR. The voltage drop across the compensator is necessary to maintain $|V_{PCCR}|$ at 0.94 p.u. with pure reactive power injection. This is the same reason why the voltage angle drops to -0.517 rad at PCCS before rising to -0.272 rad at PCCR. PCCR and PCCL are the same node in Fig. 3.32, which explains the constant values between these locations on the graph.

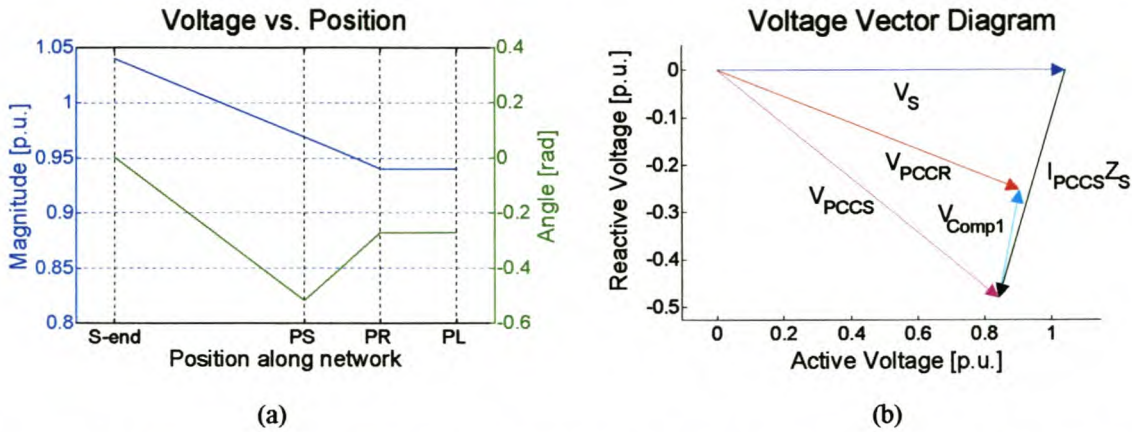


Fig. 3.33: Voltage plots and voltage vector diagram at $\delta = 0$ rad for the Cathedral Peak network under series compensation with no energy storage

The voltage vector diagram in (b) explains the above phenomena more clearly. The axes are of different scale to give a better view of the vectors. The load power factor is unity. The load current is therefore in phase with V_{PCCR} . This is a purely series network before the load; therefore I_{Comp2} is also in series with V_{PCCR} . For the compensator to inject purely reactive power, V_{Comp1} must be $\pi/2$ rad out of phase with I_{Comp1} and V_{PCCR} . Hence, V_{PCCR} and V_{Comp1} must be at a right angle to one another for the compensator to inject purely reactive power. Due to the different scaling of the axes in (b), this right angle looks skewed. Furthermore, for all δ the vectors maintain their relative magnitudes and angles to one another due to the constant load. The whole vector diagram pivots around the origin as δ , the angle of V_S , increases. With the given network parameters and $|V_{PCCR}|$ set to 0.94 p.u., the above phenomena are consequently inevitable.

As explained in section 3.3.1, the magnitude difference between two points on the magnitude plot cannot be read off directly. The true values for V_{Comp1} are shown in Fig. 3.34. The step in the voltage angle from π to $-\pi$ rad is a theoretical jump to keep the graph in the primary angle interval between π and $-\pi$ rad. It is therefore of no practical consequence.

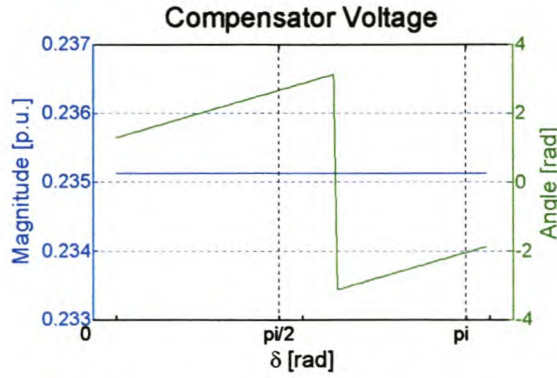


Fig. 3.34: V_{Comp1} for the Cathedral Peak network under series compensation with no energy storage

Although the series compensator must carry the full load current (5.25×10^{-3} p.u. in Fig. 3.35), the voltage across the compensator ($|V_{Comp1}|$) is quite small (0.235 p.u. in Fig. 3.34). This limits the rating of the compensator to 1.23×10^{-3} p.u., as shown in Fig. 3.36.

Fig. 3.35 gives the current graphs for the series-compensated Cathedral Peak network at $\delta = 0$ rad. Because this is a series network, the current graphs are very simple. The current magnitude and angle stays constant throughout the network and drops to zero after the load at PCCL. The negative current angle for small δ (less than 0.30 rad) can be explained with the voltage vector diagram in Fig. 3.33. As explained above, the current and V_{PCCR} must have the same angle due to the unity power factor of the load. As can be seen from Fig. 3.33, the angle of V_{PCCR} , and thus the current angle, will be negative for $\delta < 0.30$ rad.

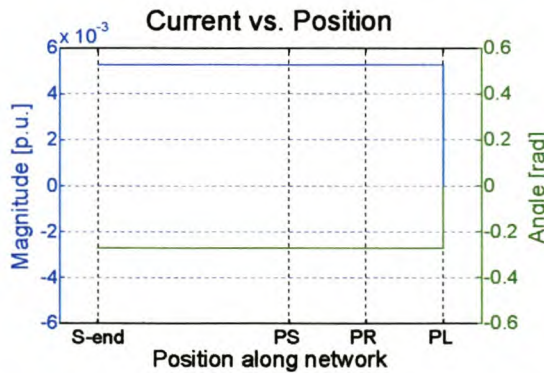


Fig. 3.35: Current magnitude and angle for series compensation with $P_{Comp1} = 0$

The absence of an active power slope between PCCS and PCCR in Fig. 3.36 implies that the series compensator injects no active power. However, the compensator injects 1.23×10^{-3} p.u. reactive power. This translates to a compensator rating of 123 kVar, which is 22 kVar more than the purely reactive shunt compensator. An advantage of the series

compensator is that 33.6 kVar (18%) less reactive power is demanded from the sending end, which is Winterton substation.

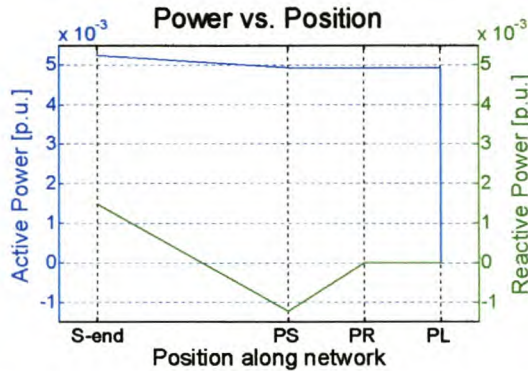


Fig. 3.36: Power transfer under series compensation with $P_{Comp1} = 0$

Should the reactive power load on Winterton substation be a problem, the purely reactive series compensator can be preferred above the purely shunt compensator. A 22% higher rating is required for the series compensator, as well as the necessary bypass switches to prevent network fault currents from flowing through the compensator. The purely reactive shunt compensator is therefore more cost effective than the series compensator. If the standby generator is configured as a synchronous condenser, the advantages of the shunt compensator increase even more.

3.4.2 Series compensator with reactive power specified

The development of the equations for a series compensator with a specific reactive power capability is discussed in this section. A purely active series compensator does not make much sense. An energy source like a generator cannot inject active power in series with the network without a transistor-based compensator like the SSSC. The energy source is connected to the dc bus of the SSSC. The ability to deliver reactive power is inherently part of a transistor-based compensator as full control is available over the voltage-current phase relationship. Although this may be a solution to the Cathedral Peak voltage-regulation problem, the analysis of this case study is postponed to the next section. The reason is that there are no specific practical limits to be set on the reactive power injected into the Cathedral Peak network. A minimum rating approach will make more sense for the Cathedral Peak scenario. The equations are still developed here and included in PSAT to make provision for other networks where this control strategy may be applicable.

Analysis equations

With Q_{Comp1} a constant in (3.40), θ_{PCCR} is the only unknown and can subsequently be solved.

$$c_x \cos \theta_{PCCR} + c_y \sin \theta_{PCCR} + c_{ab} = 0 \quad (3.42)$$

$$\text{where } c_{ab} = c_z Q_{Comp1} + c_{aa}$$

Substituting (3.19) to (3.21) into (3.42), θ_{PCCR} is solved as:

$$\theta_{PCCR} = 2 \arctan \left(\frac{c_y \mp \sqrt{c_y^2 - c_{ab}^2 + c_x^2}}{c_y - c_x} \right) \quad (3.43)$$

Equation (3.43) is a function of the four-quadrant arc tangent and therefore has two solutions. For every δ PSAT tests the validity of every solution. With both solutions valid, PSAT chooses the solution that gives the lowest compensator rating (S_{comp1}). As before, the user may still choose the required solution manually.

3.4.3 Minimum rating series compensator

One of the most important control strategies is to minimise the compensation power required while achieving the required voltage regulation. This will reduce the required compensator rating and therefore reduce costs. It must be kept in mind that a purely reactive solution may exist (e.g. capacitor banks) that could be much cheaper than the minimised solution (e.g. an SSSC), which requires both active and reactive power injection. Evaluation and comparison of the effectiveness of various solutions is one of the key purposes of PSAT. Once the ratings of the various types of compensators are established, cost comparisons can be made to determine the best solution, both in terms of cost and performance.

Analysis equations

Equation (3.32) from the minimum shunt compensator can also be used to derive the equations for a minimum series compensator:

$$P_{Comp1} \frac{\partial P_{Comp1}}{\partial \theta_{PCCR}} + Q_{Comp1} \frac{\partial Q_{Comp1}}{\partial \theta_{PCCR}} = 0 \quad (3.44)$$

P_{Comp1} and Q_{Comp1} are obtained from (3.39) and (3.40), respectively. Substituting these equations into (3.44) results in a single equation with one unknown, θ_{PCCR} .

$$c_{ac} \cos^2 \theta_{PCCR} - c_{ac} \sin^2 \theta_{PCCR} + c_{ad} \cos \theta_{PCCR} + c_{ae} \sin \theta_{PCCR} + c_{af} \cos \theta_{PCCR} \sin \theta_{PCCR} = 0 \quad (3.45)$$

$$\text{where } c_{ac} = \frac{c_t c_u}{c_v^2} + \frac{c_x c_y}{c_z^2} \quad c_{ad} = \frac{c_w c_u}{c_v^2} + \frac{c_{aa} c_y}{c_z^2}$$

$$c_{ae} = -\frac{c_w c_t}{c_v^2} - \frac{c_{aa} c_x}{c_z^2} \quad c_{af} = -\left(\frac{c_t}{c_v}\right)^2 + \left(\frac{c_u}{c_v}\right)^2 - \left(\frac{c_x}{c_z}\right)^2 + \left(\frac{c_y}{c_z}\right)^2$$

θ_{PCCR} is now the only unknown in (3.45) and can therefore be solved. With the squared trigonometric terms, (3.45) is a fourth-order equation in θ_{PCCR} . An explicit solution for fourth-order equations exists, but it is very lengthy and tedious to work with. *Maple* was therefore used again to calculate the four explicit solution formulas for θ_{PCCR} . The program listing of *Solto4thorder.mws*, the file written to solve θ_{PCCR} from (3.45), is included in Addendum B.

PSAT computes all four solutions to θ_{PCCR} and tests their validity by back-substitution into equation (3.45). As before, if more than one solution is valid the one resulting in the lowest compensator rating is chosen, unless the user overrides the software.

Practical application

For easy reference Fig. 3.14, the now familiar circuit diagram of the series-compensated network, is shown below. True to the previous analyses, $|V_{PCCR}|$ is again specified as 0.94 p.u. to limit compensation power and to adhere to the NRS 048 minimum specification [B4].

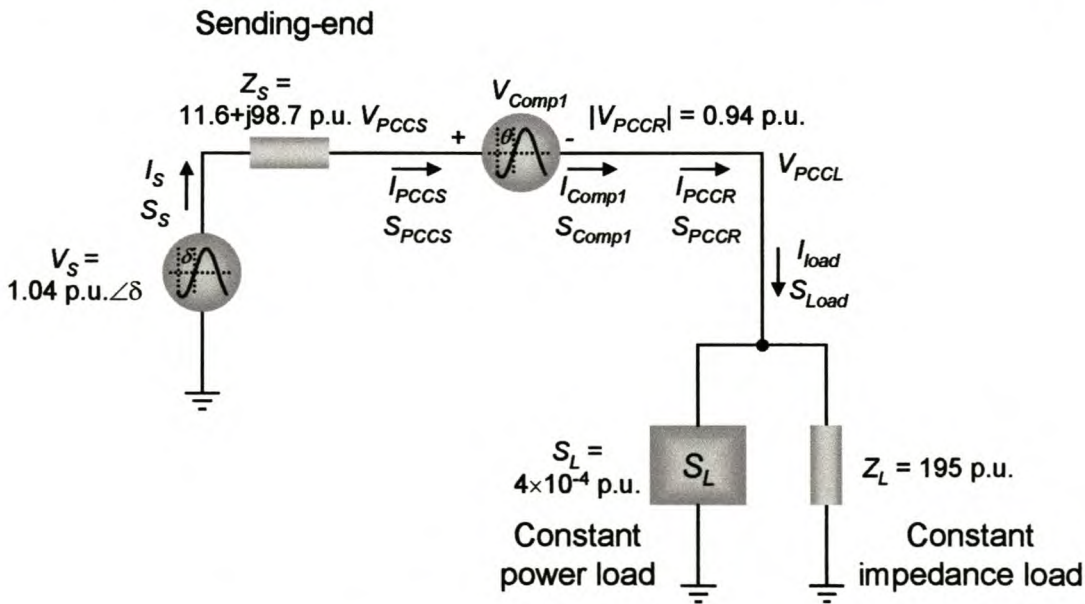


Fig. 3.37: Cathedral Peak network with a series compensator

Fig. 3.38 shows the series-compensator rating as a function of δ and θ_{PCCR} . It is clear that only one solution of θ_{PCCR} results in a minimum rating compensator for every δ in the Cathedral Peak network.

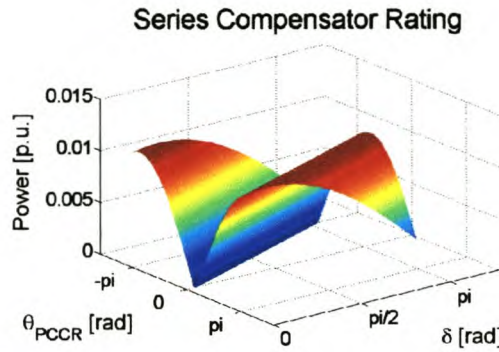


Fig. 3.38: Series-compensator rating as a function of θ_{PCCR} and δ

The voltage diagrams for the above solution, as generated by PSAT, are shown in Fig. 3.39 for $\delta = 0$ rad. $|V_{PCCR}|$ is at the specified 0.94 p.u., but $|V_{PCCS}|$ remains at the uncompensated level of 0.864 p.u. because the series compensator is situated between PCCS and PCCR.

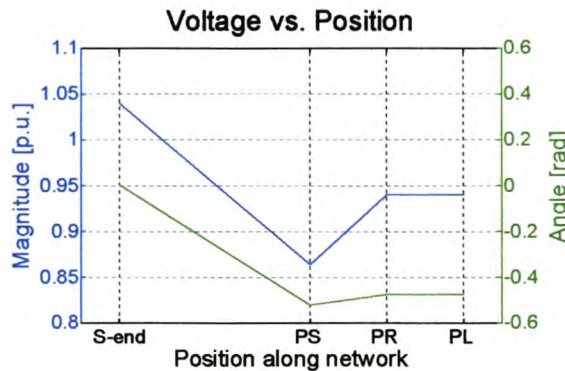


Fig. 3.39: Voltage plots for the Cathedral Peak network with minimum series compensation

The true voltage added by the series compensator, as well as the power injected by the compensator, is shown in Fig. 3.40. The compensator voltage magnitude is 0.0869 p.u. for all δ due to the constant load. This shows a typical advantage of the series compensator: only a small voltage needs to be injected by the compensator to achieve the desired voltage regulation. This limits the compensator rating regardless of the full load current that flows through the compensator. The minimum rating of a series compensator for the Cathedral Peak network is 0.457×10^{-3} p.u. or 45.7 kVA with a power factor of 88.6 (40.5 kW and 20.9 kVA).

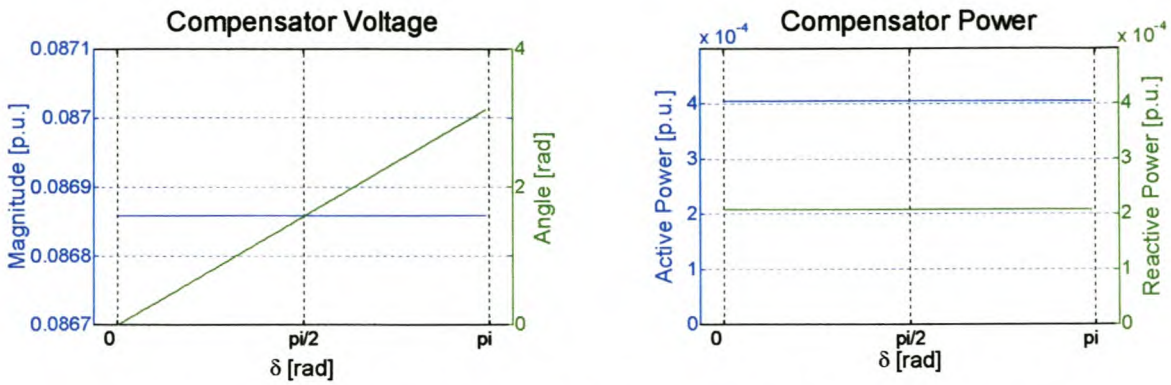


Fig. 3.40: Voltage and power plots for the series compensator of minimum rating

The current graphs are shown in Fig. 3.41. The current magnitude is the same as in the network compensated by a purely reactive series compensator. This is to be expected as the voltage magnitude is regulated to the same 0.94 p.u. and the load remains unchanged. From Ohm's Law the current magnitudes must therefore also correspond. The current angle of the minimum rating compensator is 0.2 rad less than the current angle of the purely reactive series compensator for all δ . For a minimum compensator rating, the voltage angle is 0.2 rad less than the voltage angle of the purely reactive compensator. Due to the constant load and series network, the current angle must also be 0.2 rad less to satisfy Ohm's Law.

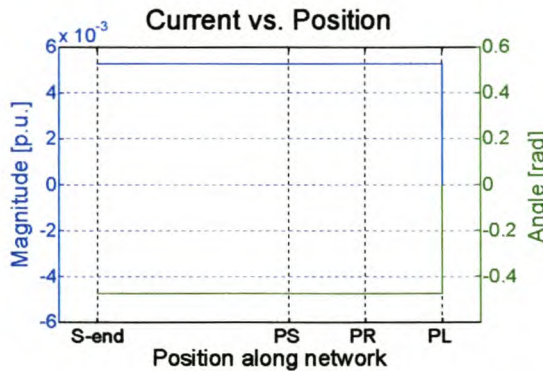


Fig. 3.41: Current magnitude and angle for minimum series compensation

The slopes between PCCS and PCCR in the active and reactive power plots correspond to the compensator power rating graph in Fig. 3.40. Compared to a minimum shunt voltage regulator, the series device rating is 44% less apparent power and 12% less active power.

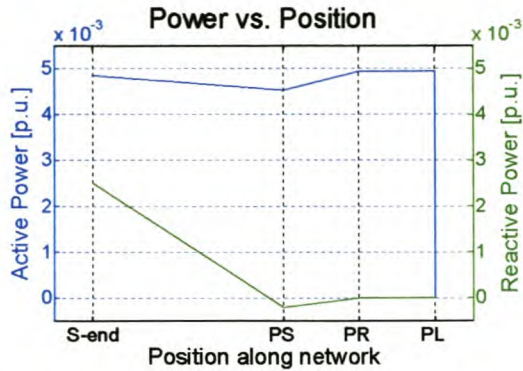


Fig. 3.42: Power transfer under minimum series compensation

Although the series device rating is much less than the shunt device rating for minimum compensation, the implementation costs are much higher. As discussed in section 3.4, a rectifier and a transistor-based compensator are necessary to connect the hotel generator in series with the network. A minimum shunt compensator requires only modifications to the generator to render its power factor controllable and capacitor banks to supply the bulk reactive power due to the low power factor required. The 11% saving in active power and therefore operating costs in terms of fuel is a small advantage of the series compensator, but it is insignificant when compared to the capital cost disadvantage.

Compared to the costs of building a new line, a series compensator of 45.7 kVA will be much more cost effective. The versatility of feeding the Cathedral Peak network from two sources is lost if a new line is not built, but research continues to operate series devices as Uninterruptible Power Supplies (UPSs).

3.4.4 Summary of series compensation

Only two series compensator options are viable for the Cathedral Peak network, namely purely reactive and minimum compensation. Purely reactive series compensation requires 123 kVar reactive power injection, while the device rating of a minimum-rating series compensator is 45.7 kVA. The latter requires an expensive transistor-based compensator with a continuous energy supply, while purely reactive compensation can be done with a series capacitor or even a Thyristor-Controlled Series Capacitor, which is more expensive.

3.5 SERIES-SHUNT COMPENSATION

The third option in terms of non-traditional network technologies is a series-shunt compensator, which is a combination of the above two devices. The series and shunt devices are fed from a common dc bus and therefore the devices can exchange active power between them. With two devices, which can do both active and reactive compensation, two complex network parameters can be fully controlled. The compensator therefore has four degrees of freedom. Both magnitude and angle of V_{PCCR} can be controlled, which accounts for two degrees of freedom. The series compensator does this. The shunt compensator controls the power factor at PCCS with reactive power injection, which accounts for another degree of freedom. The shunt compensator also absorbs active power and transfers it to the series device across the dc bus, which is the last degree of freedom.

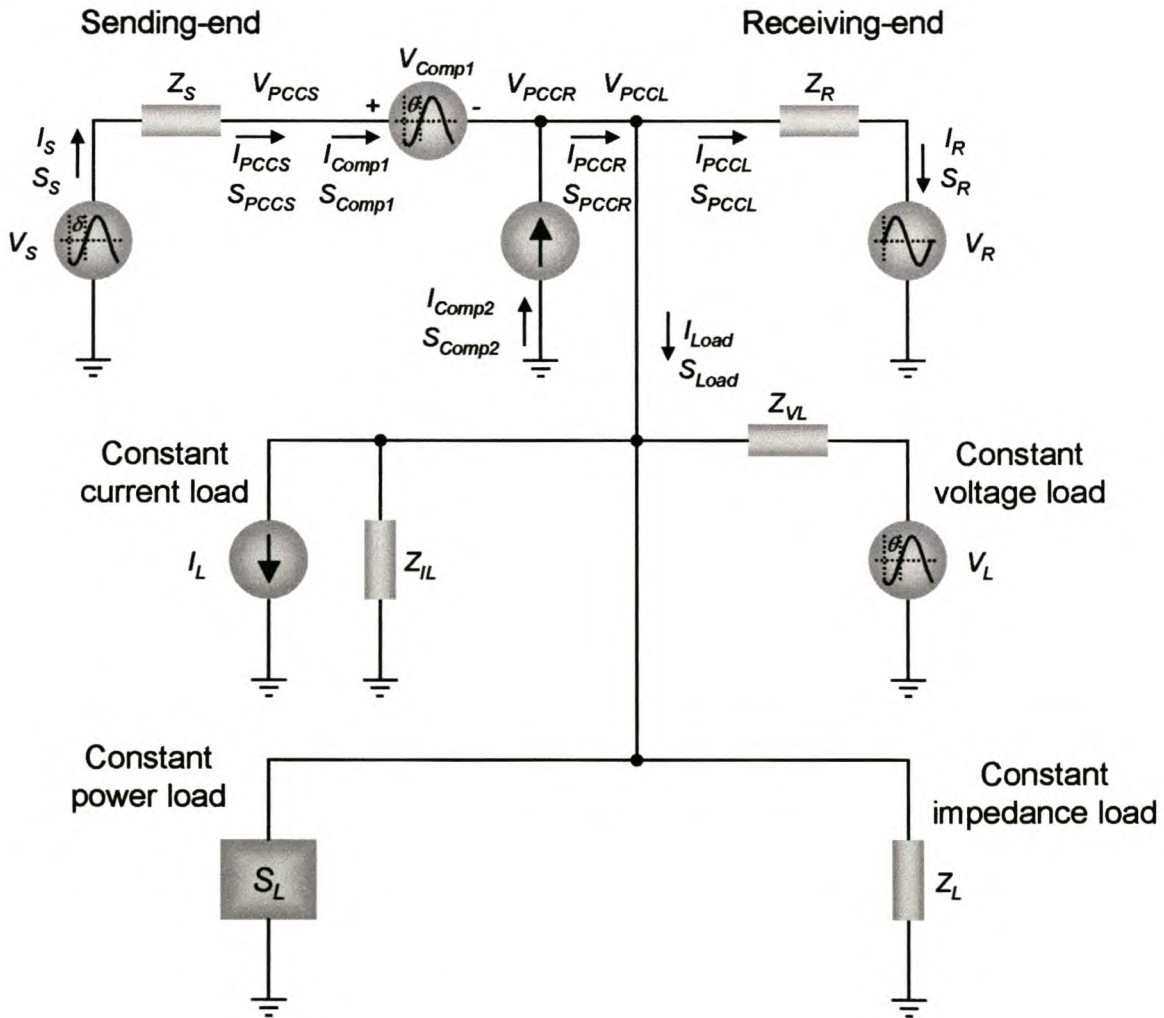


Fig. 3.43: Series-shunt voltage regulator

The series-shunt device therefore has no need for an energy-storage device, which eliminates the need for an extra control equation. The circuit diagram for a series-shunt compensated network is shown in Fig. 3.43.

Analysis equations

With the series-shunt compensator all possible help is needed to simplify the equations. The above network is therefore simplified, without any approximations, to the circuit in Fig. 3.44, which renders the solutions to the network voltages less difficult. This process is fully documented in section 3.2.3.

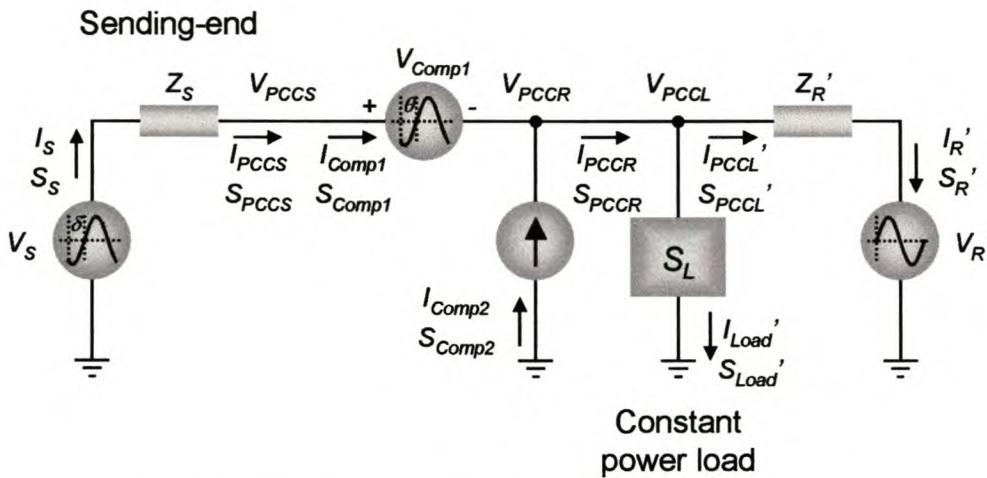


Fig. 3.44: Simplified network with a series-shunt-compensator

For the series-shunt compensator, the network planner specifies both the required magnitude and angle of V_{PCCR} . If the angle is left unspecified, the circuit can only be solved with fifth- and higher-order equations. He also specifies the required power factor at PCCS, the point of common coupling between the sending-end network and the compensator. V_R' and Z_R' follow from section 3.2.3 and V_S and Z_S are known network parameters. I_{PCCL}' and S_{PCCR} can therefore be solved:

$$I_{PCCL}' = \frac{V_{PCCR} - V_R'}{Z_R'} \quad (3.46)$$

$$\begin{aligned} S_{PCCR} &= S_{Load}' + V_{PCCR} I_{PCCL}'^* \\ &= S_{Load}' + \frac{|V_{PCCR}|^2 - V_{PCCR} V_R'^*}{Z_R'^*} \end{aligned} \quad (3.47)$$

The shunt compensator must supply all the active power required by the series compensator. If losses are ignored, the active power supplied by the series compensator is fully absorbed by

the shunt compensator. It can therefore be argued that the active power of S_{PCCR} is the only active power supplied by the sending end. The reactive power demand from the sending end is regulated by the specified power factor PF at PCCS. P_{PCCS} and Q_{PCCS} are therefore given by (3.48) and (3.49).

$$P_{PCCS} = P_{PCCR} = \text{Re}(S_{PCCR}) \quad (3.48)$$

$$Q_{PCCS} = \frac{\sqrt{1 - PF^2}}{PF} P_{PCCS} \quad (3.49)$$

The object is to solve $|V_{PCCS}|$ and θ_{PCCS} . Up to this point $S_{PCCS} = P_{PCCS} + jQ_{PCCS}$, V_S and Z_S are known. If S_{PCCS} can be written in terms of $|V_{PCCS}|$ and θ_{PCCS} , these two unknowns, and the whole circuit, can be solved. I_{PCCS} follows from Ohm's Law:

$$I_{PCCS} = \frac{V_S - V_{PCCS}}{Z_S} \quad (3.50)$$

$$\begin{aligned} S_{PCCS} &= V_{PCCS} I_{PCCS}^* \\ &= |V_{PCCS}|^2 (c_{ak} + jc_{al}) + |V_{PCCS}| (\cos \theta_{PCCS} + j \sin \theta_{PCCS}) (c_{ag} + jc_{ah}) \end{aligned} \quad (3.51)$$

$$\text{where } c_{ag} = \text{Re}\left(\frac{V_S^*}{Z_S^*}\right) \quad c_{ah} = \text{Im}\left(\frac{V_S^*}{Z_S^*}\right) \quad c_{ak} = \text{Re}\left(-\frac{1}{Z_S^*}\right) \quad c_{al} = \text{Im}\left(-\frac{1}{Z_S^*}\right)$$

Splitting (3.51) into real and imaginary parts, P_{PCCS} and Q_{PCCS} can be written as functions of $|V_{PCCS}|$ and θ_{PCCS} .

$$P_{PCCS} = |V_{PCCS}| c_{ag} \cos \theta_{PCCS} + |V_{PCCS}| c_{ah} \sin \theta_{PCCS} + |V_{PCCS}|^2 c_{ak} \quad (3.52)$$

$$Q_{PCCS} = |V_{PCCS}| c_{ah} \cos \theta_{PCCS} + |V_{PCCS}| c_{ag} \sin \theta_{PCCS} + |V_{PCCS}|^2 c_{al} \quad (3.53)$$

P_{PCCS} and Q_{PCCS} are known and $|V_{PCCS}|$ and θ_{PCCS} can therefore be solved from the two equations (3.52) and (3.53). Similar to (3.12) and (3.13), the above equations terms in $\cos \theta_{PCCS}$ and $\sin \theta_{PCCS}$ and are therefore quadratic functions of θ_{PCCS} . They are also quadratic functions of $|V_{PCCS}|$, which results in fourth-order solutions. The *Maple* file, *SolPQ.mws* (listed in Addendum B) are used again to solve $|V_{PCCS}|$ and θ_{PCCS} because the solutions to the fourth-order equations are very long and tedious to work with.

As always, the solutions for $|V_{PCCS}|$ and θ_{PCCS} are tested for validity and thereafter the solution resulting in the lowest compensator rating ($S_{comp1} + S_{comp2}$) is chosen by PSAT - that is, if the user does not override the software and choose the required solution manually.

Practical application

With the network parameters from Fig. 3.6, the circuit diagram of the series-shunt compensated network is drawn in Fig. 3.45. For consistency, $|V_{PCCR}|$ is specified to 0.94 p.u. to limit the compensator rating and to adhere to the NRS 048 minimum specification [B4]. For the series-shunt compensator, θ_{PCCR} is also needed and therefore specified as 0.1 rad. The load does not have a constant voltage or constant current characteristic, so θ_{PCCR} makes no difference to the load. It only determines the load current angle as the load has a fixed power factor of one. The network planner must also specify the input power factor to the compensator, i.e. the power factor at PCCS. It is taken as unity to limit reactive power demand from Winterton substation at the sending end.

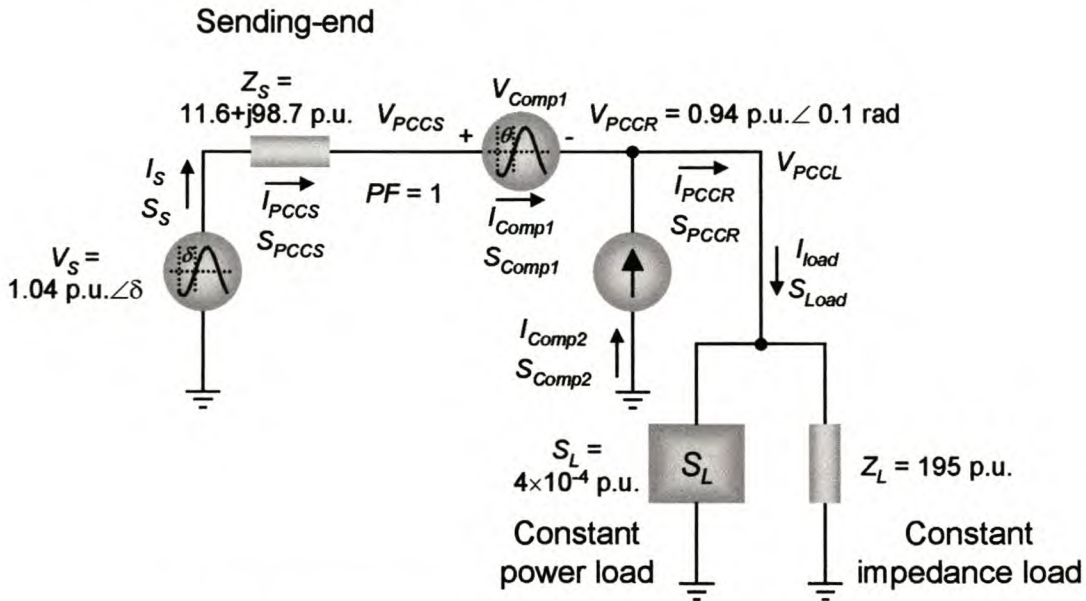


Fig. 3.45: Cathedral Peak network with a series-shunt compensator

PSAT generated the voltage diagrams in Fig. 3.46. $|V_{PCCR}|$ is raised to 0.931 p.u., which is 0.009 p.u. short of the required voltage regulation. The series-shunt compensator is not able to regulate the voltage magnitude at PCCR to the required 0.94 p.u. With no energy storage, the compensator draws the required compensating power from the sending-end before injecting it back into the network to regulate $|V_{PCCR}|$. In this network, the voltage drop across Z_S due to the load and compensating currents is too large to sustain 0.94 p.u. voltage regulation by the series-shunt compensator. As the compensator needs more power to push $|V_{PCCR}|$ up, it draws more current through Z_S , which increases the voltage drop across Z_S . $|V_{PCCS}|$ drops further and consequently the compensator needs even more power to compensate for the voltage difference between $|V_{PCCS}|$ and the required $|V_{PCCR}|$. One way to

break this loop is to add an energy-storage device to the dc bus of the compensator. The popular energy-storage devices like batteries, flywheels, and Superconducting Magnetic Energy Storage (SMES) devices cannot deliver the continuous power required for voltage regulation. A special power generator such as fuel cells, a diesel generator, a gas turbine, a solar dish or a flow battery [A15] is the only solution. This option adds to the already expensive series-shunt device and is therefore not covered here.

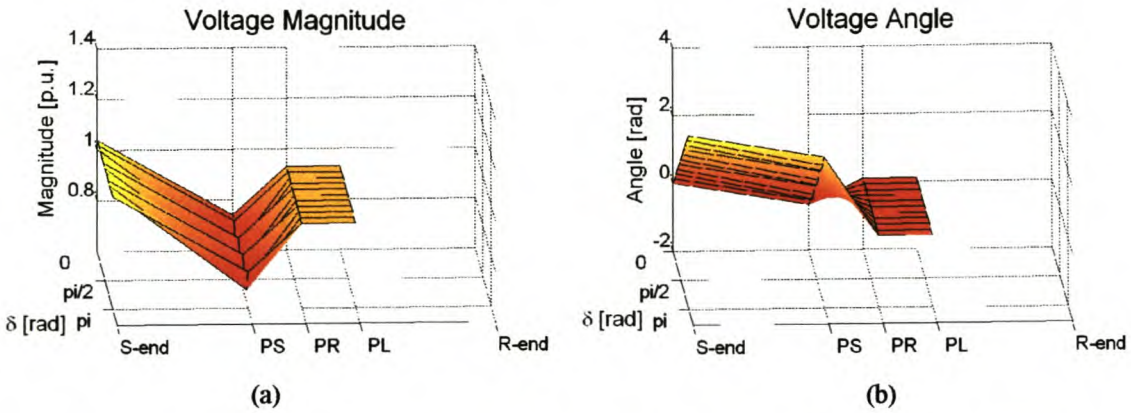


Fig. 3.46: Voltage plots for the Cathedral Peak network under series-shunt compensation

PSAT tried to calculate θ_{PCCS} and $|V_{PCCS}|$ for $|V_{PCCR}| = 0.94$ p.u., but obtained only invalid answers. It subsequently lowered $|V_{PCCR}|$ for each δ until a valid answer for θ_{PCCS} and $|V_{PCCS}|$ were obtained. This explains the ripple in V_{PCCS} across δ in the voltage magnitude and angle graphs. In Fig. 3.47 the voltage injected by the series compensator also shows this ripple, as expected. The large voltage magnitudes injected by the series compensator to regulate $|V_{PCCR}|$ confirm the problem of the increasing voltage drop across Z_S due to the compensator current. With such a large voltage injected, the series compensator is expected to have a relatively large power requirement.

The fact that the magnitude and angle of a voltage difference cannot be read directly from the network voltage plots as well as the negative voltage angle for small values of δ has been thoroughly explained in section 3.3.1.

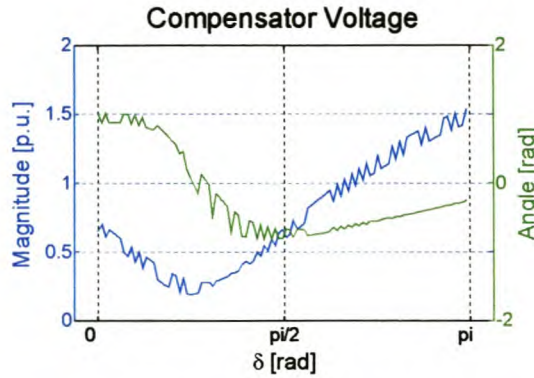


Fig. 3.47: Voltage diagrams for the series compensator in the series-shunt compensated Cathedral Peak network

The ripple in the current graphs in Fig. 3.48 also reflects the compensator's attempts to regulate $|V_{PCCR}|$ to 0.94 p.u. Both the magnitude and angle plots show steps at PCCR and PCCL, which corresponds to the respective shunt compensation and load currents. The series compensator regulates $|V_{PCCR}|$ and therefore the ripple is absent from PCCR onwards. The step in current angle corresponds to the regulated angle of V_{PCCR} (0.1 rad) across the unity power-factor load.

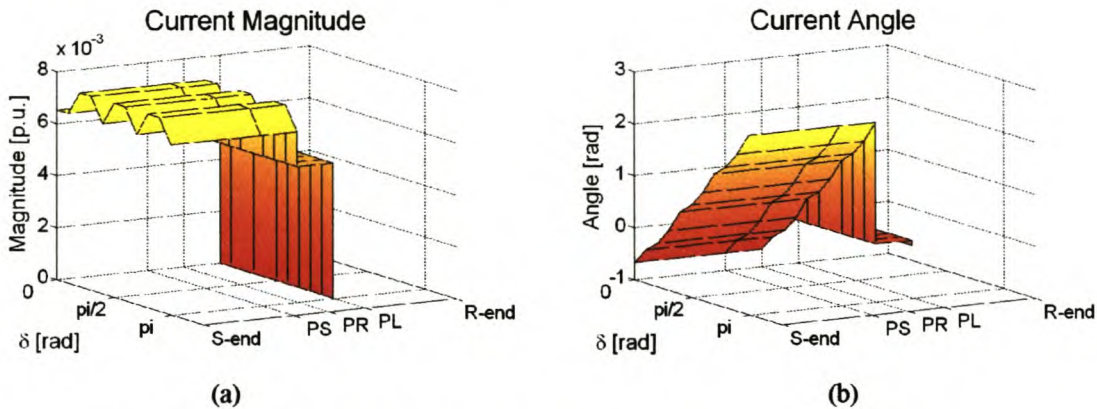


Fig. 3.48: Current magnitude and angle for series-shunt compensation

The same ripple is also visible in the power graphs in Fig. 3.49. Unlike the power graphs of the shunt and series compensators in the previous sections, the power graphs of the series-shunt compensator is not constant across δ . The reason is that θ_{PCCR} is kept constant by the series-shunt compensator, as seen in Fig. 3.46. As δ varies from 0 to π rad, the compensators must inject variable amounts of active and reactive power to keep θ_{PCCR} constant.

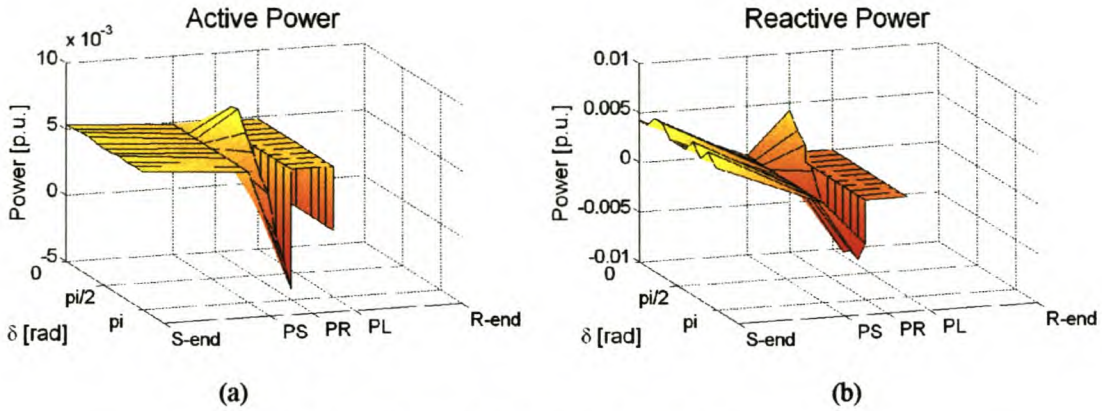


Fig. 3.49: Power transfer under series-shunt compensation

Since the compensator has no energy-storage device, the active power absorbed and injected must be equal because losses are ignored. This is true for all δ , but Fig. 3.50 clearly shows this for $\delta = 0$ rad, where the active power at PCCS equals the active power after the shunt compensator at PCCR. The slope between PCCS and PCCR represents the series-device power and the step at PCCR is the shunt-device power. The reactive power absorbed and injected also balances out but this is due to a different reason. The load power factor is unity and the load therefore absorbs no reactive power. Similarly, due to the unity power factor at PCCS, as specified for this network, the reactive power at PCCS is zero. With the input and output reactive power zero, the compensator must supply and absorb an equal amount of reactive power.

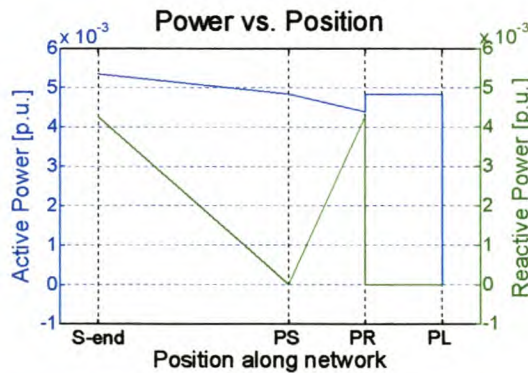


Fig. 3.50: Power transfer across the Cathedral Peak network under series-shunt compensation at $\delta = 0$ rad

The active and reactive power delivery for the separate series and shunt compensators are shown in Fig. 3.51. The fact that the power delivered by the series compensator is the negative of the power delivered by the shunt compensator for all δ , supports the above observations. Although not immediately visible in the power graphs, the maximum apparent power delivered or absorbed by the separate compensators is 10.5×10^3 p.u. (1.05 MVA) at

$\delta = 3.08$ rad. The active power delivered and absorbed per compensator is 9.08×10^{-3} p.u. (908 kW) at this point. It can therefore be reasoned that the maximum rating of the series-shunt compensator is 2.10 MVA (1.82 MW). This certainly confirms the suspicions of a high compensator rating due to the large voltages injected by the series compensator. This is however an unusually large voltage angle at which the network is operated. For $\delta < \pi/2$ rad the maximum total device rating is 1.04 MVA. This is also the rating at $\delta = 0$ rad.

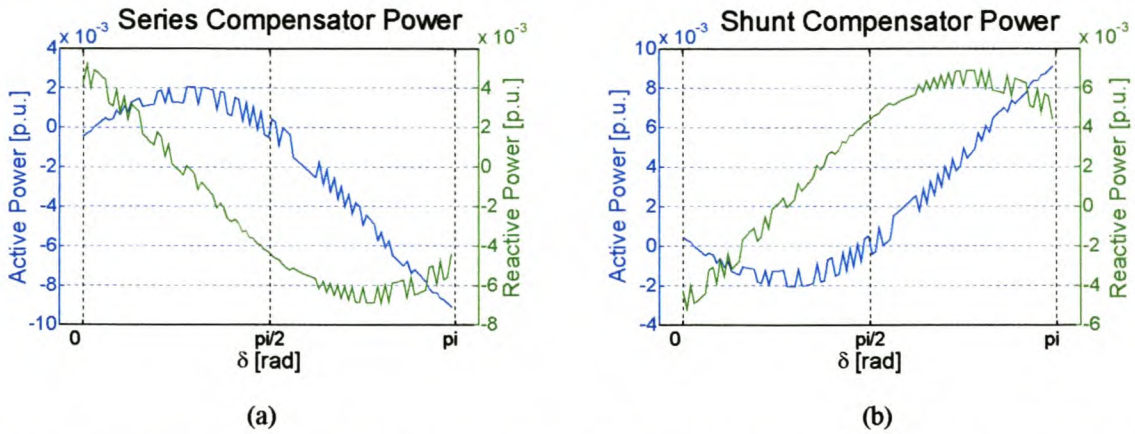


Fig. 3.51: Power delivery by the separate series and shunt compensators in the Cathedral Peak

For the Cathedral Peak network a series-shunt compensator is definitely not a viable option at its high rating. The most important advantage of a series-shunt compensator is that it does not require energy storage to perform active power compensation. In other networks, this can result in the series-shunt compensator costing less than a shunt or a series compensator with an energy-storage device.

3.6 IN-LINE COMPENSATION

Finally, in-line compensation is considered as a method of voltage regulation. Although the in-line compensator is generally not seen as a voltage regulator per se, it has the ability to perform voltage regulation [B2].

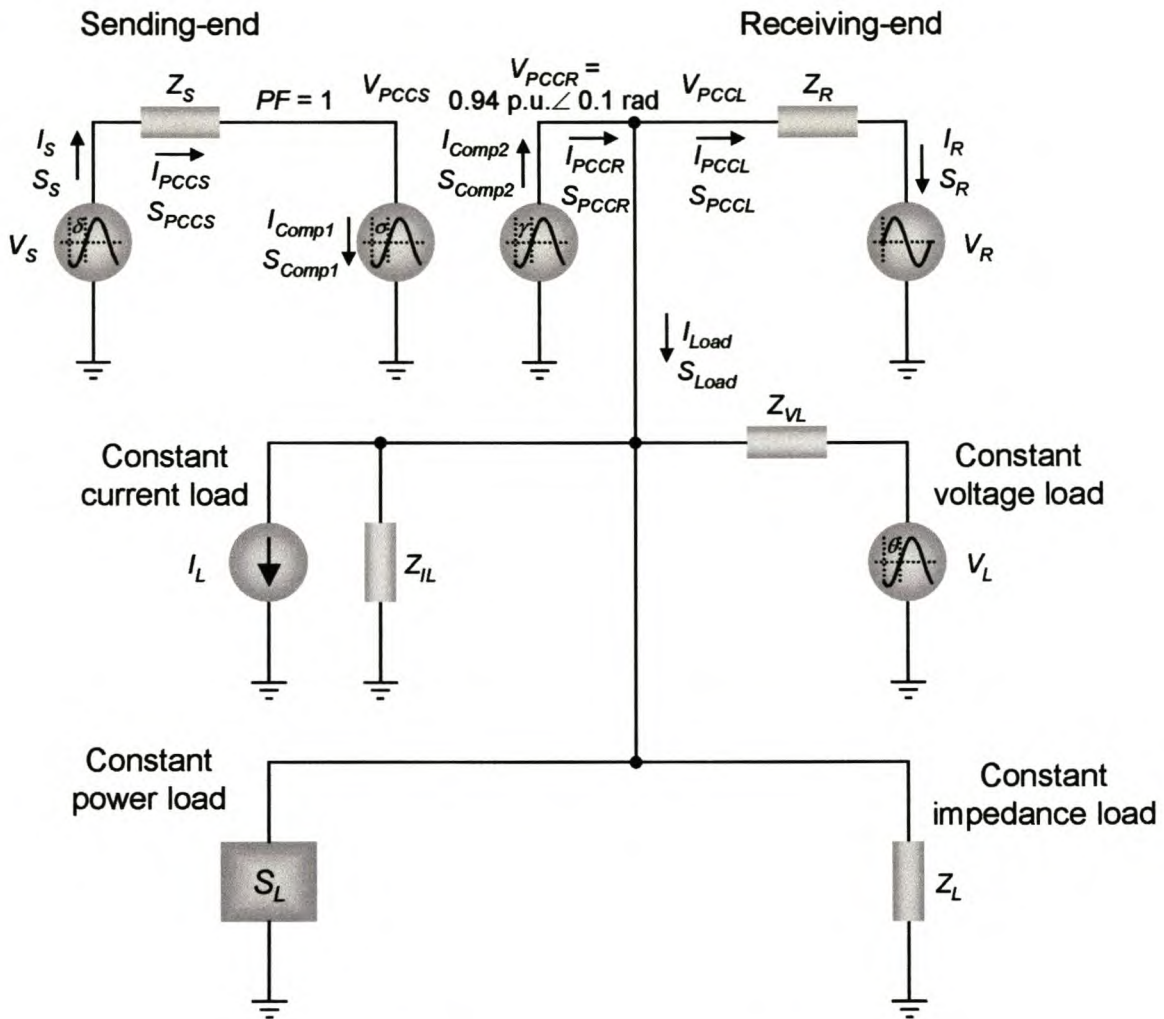


Fig. 3.52: In-line device as voltage regulator

As shown in Fig. 3.52, two compensators can be connected back-to-back or via a High-Voltage DC (HVDC) line. The compensator at the sending end absorbs active and reactive power and transfers the active power across the dc link to the receiving-end compensator. Generally the compensators are based on thyristor technology and require therefore reactive power to be able to transmit active power. This reactive power can be supplied by the ac network, but capacitor banks are often used to supply this demand [A8]. The input power factor at PCCS is therefore one of the parameters specified by the network planner. The receiving-end compensator transmits the active power received over the dc link to the

connected ac network. At this point, the compensator has the ability to regulate both the magnitude and angle of the voltage supplied to the ac network. One of the popular uses for in-line compensators is to connect ac networks with phase-angle differences [B2]. Similar to the series-shunt device, the in-line device has no need for an energy-storage device, which eliminates the need for an extra control equation.

Analysis equations

The process of reducing the above network, without any approximations, to Fig. 3.53 is fully documented in section 3.2.3.

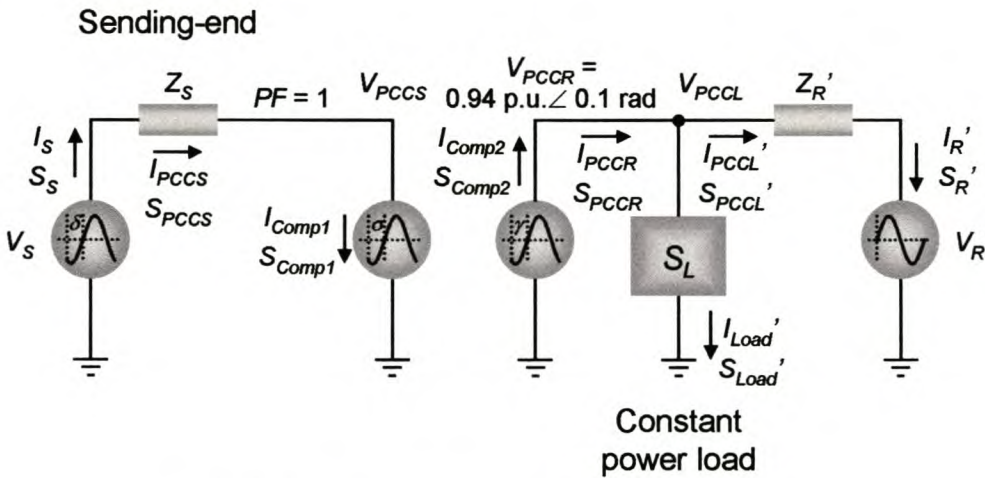


Fig. 3.53: Simplified network with an in-line compensator

As discussed above, the network planner specifies both the required magnitude and angle of V_{PCCR} . He also specifies the required power factor at PCCS, the point of common coupling between the sending-end network and the compensator. V_R' and Z_R' follow from section 3.2.3 and V_S and Z_S are known network parameters. As with the series-shunt device, I_{PCCL}' and S_{PCCR} can therefore be solved:

$$I_{PCCL}' = \frac{V_{PCCR} - V_R'}{Z_R'} \quad (3.54)$$

$$\begin{aligned} S_{PCCR} &= S_{Load}' + V_{PCCR} I_{PCCL}'^* \\ &= S_{Load}' + \frac{|V_{PCCR}|^2 - V_{PCCR} V_R'^*}{Z_R'^*} \end{aligned} \quad (3.55)$$

It is assumed that the compensator or additional capacitor banks supply the reactive power demanded by S_{PCCR} . The active power is supplied via the dc link. V_{DCR} is the operating voltage of the dc link and is specified by the network planner. The dc link voltage depends on

the ac network voltages and the design of the in-line compensator. Whether the compensator is a back-to-back converter or an HVDC line also influences the dc link voltage. With S_{PCCR} and V_{DCR} known, the current through the dc link (I_{DC}) is calculated.

$$I_{DC} = \frac{\text{Re}(S_{PCCR})}{V_{DCR}} \quad (3.56)$$

The dc resistance of an HVDC line can cause a significant voltage drop across the line. This is taken into account by PSAT and the network planner has to specify this parameter. The sending-end voltage of the dc link, V_{DCS} , can now be calculated.

$$V_{DCS} = V_{DCR} + I_{DC} R_{DC} \quad (3.57)$$

With the sending-end voltage and the current of the dc link known, the active power transferred by the sending-end compensator, P_{PCCS} , can be calculated. Q_{PCCS} , the reactive power required at PCCS, is determined by the power factor PF at PCCS, which is specified by the network planner.

$$P_{PCCS} = V_{DCS} I_{DC} \quad (3.58)$$

$$Q_{PCCS} = \frac{\sqrt{1 - PF^2}}{PF} P_{PCCS} \quad (3.59)$$

From here the same procedure as with the series-shunt compensator is used to solve $|V_{PCCS}|$ and θ_{PCCS} . Up to this point $S_{PCCS} = P_{PCCS} + jQ_{PCCS}$, V_S and Z_S are known. Writing S_{PCCS} in terms of $|V_{PCCS}|$ and θ_{PCCS} can solve these two unknowns. I_{PCCS} follows from Ohm's Law:

$$I_{PCCS} = \frac{V_S - V_{PCCS}}{Z_S} \quad (3.60)$$

$$\begin{aligned} S_{PCCS} &= V_{PCCS} I_{PCCS}^* \\ &= |V_{PCCS}|^2 (c_{ak} + jc_{al}) + |V_{PCCS}| (\cos \theta_{PCCS} + j \sin \theta_{PCCS}) (c_{ag} + jc_{ah}) \end{aligned} \quad (3.61)$$

$$\text{where } c_{ag} = \text{Re}\left(\frac{V_S^*}{Z_S}\right) \quad c_{ah} = \text{Im}\left(\frac{V_S^*}{Z_S}\right) \quad c_{ak} = \text{Re}\left(-\frac{1}{Z_S^*}\right) \quad c_{al} = \text{Im}\left(-\frac{1}{Z_S^*}\right)$$

Splitting (3.51) into real and imaginary parts, P_{PCCS} and Q_{PCCS} can be written as functions of $|V_{PCCS}|$ and θ_{PCCS} .

$$P_{PCCS} = |V_{PCCS}| c_{ag} \cos \theta_{PCCS} + |V_{PCCS}| c_{ah} \sin \theta_{PCCS} + |V_{PCCS}|^2 c_{ak} \quad (3.62)$$

$$Q_{PCCS} = |V_{PCCS}| c_{ah} \cos \theta_{PCCS} + |V_{PCCS}| c_{ag} \sin \theta_{PCCS} + |V_{PCCS}|^2 c_{al} \quad (3.63)$$

P_{PCCS} and Q_{PCCS} are known from (3.58) and (3.59). $|V_{PCCS}|$ and θ_{PCCS} can therefore be solved from the two equations above. As with the series-shunt compensator and the uncompensated

network, the above equations both have terms in $\cos\theta_{PCCS}$ and $\sin\theta_{PCCS}$ and are therefore quadratic functions of θ_{PCCS} . Along with second-order terms in $|V_{PCCS}|$, this results in fourth-order solutions and the *Maple* file, *SolPQ.mws* (listed in Addendum B) is used to solve $|V_{PCCS}|$ and θ_{PCCS} . This reduces the chance of human error because the solutions to the fourth-order equations are very long and tedious to work with. Once again PSAT tests the solutions for $|V_{PCCS}|$ and θ_{PCCS} for validity and thereafter the solution resulting in the lowest compensator rating ($S_{comp1} + S_{comp2}$) is chosen. The user still has the option to override the software and choose the required solution manually.

Practical application

The network parameters from Fig. 3.6 are used to draw the circuit diagram of the Cathedral Peak network, this time with an in-line compensator, as shown in Fig. 3.54. $|V_{PCCR}|$ is again specified as 0.94 p.u. As for the series-shunt compensator, θ_{PCCR} is also needed and therefore specified as 0.1 rad. As discussed with the series-shunt compensator, θ_{PCCR} only determines the load current angle as the load has a fixed power factor of unity. The network planner must also specify the power factor at PCCS, the dc-line resistance and the operating voltage of the dc link. The power factor is taken as unity to limit reactive power demand from Winterton substation at the sending end, the dc line resistance is zero (back-to-back converter) and the operating voltage is chosen as $1 \text{ p.u.}/0.78 = 1.28 \text{ p.u.}$ [B8].

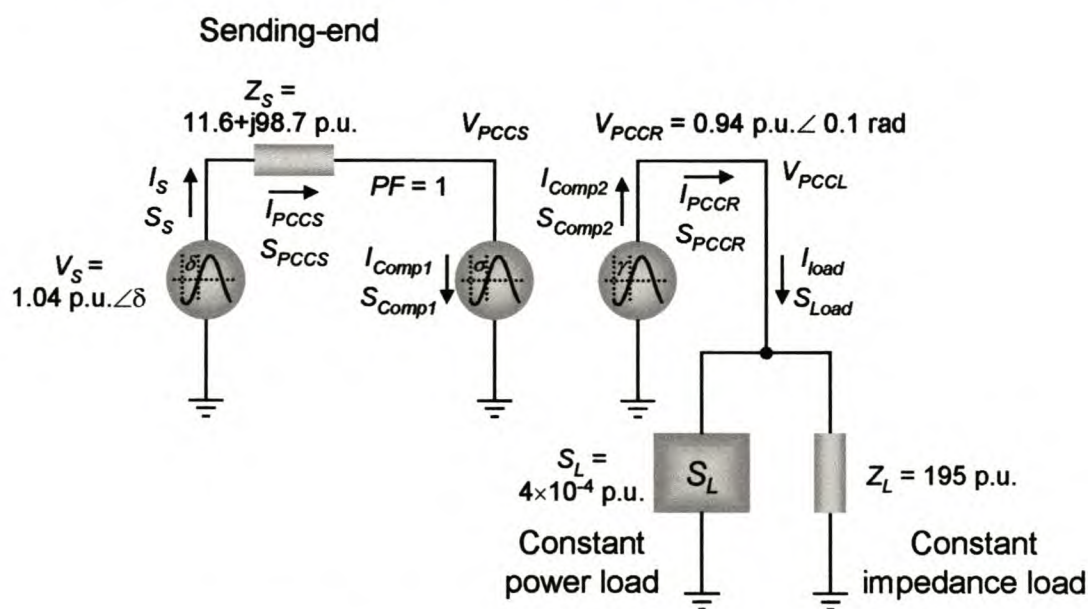


Fig. 3.54: Cathedral Peak network with an in-line compensator

The voltage diagrams across the in-line compensated network at $\delta = 0$ rad are shown in Fig. 3.55. The in-line compensator cannot raise $|V_{PCCR}|$ to the required 0.94 p.u. Only 0.931 p.u. is achieved, which is 0.009 p.u. short of the required voltage regulation. With the particular active and reactive power demand at PCCS, no solution could be found for V_{PCCS} and θ_{PCCS} with the specific values of V_S and Z_S of the Cathedral Peak network. PSAT therefore reduces the required $|V_{PCCR}|$ to the value closest to the specified 0.94 p.u. for which a solution to $|V_{PCCS}|$ and θ_{PCCS} can be found.

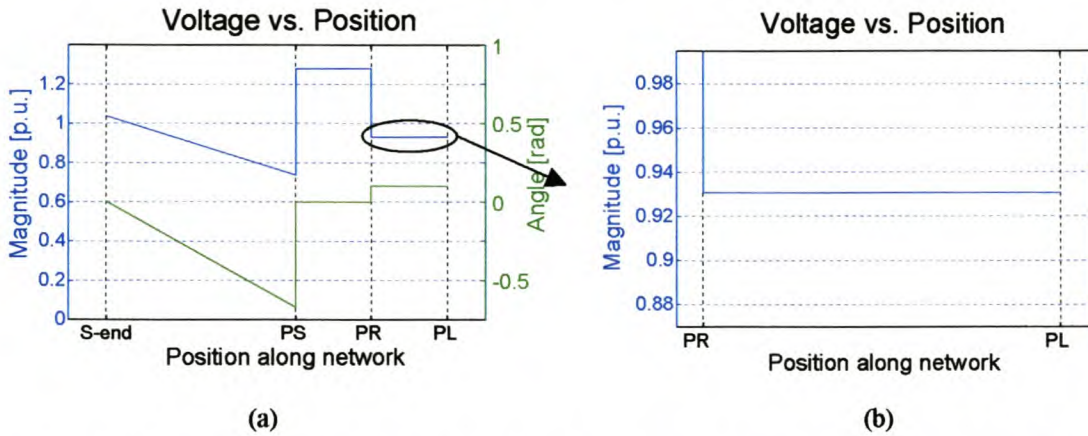


Fig. 3.55: Voltage plots across the Cathedral Peak network with an in-line compensator for $\delta = 0$ rad. By testing various reactive power specifications at PCCS with PSAT, it was found that the in-line compensator can regulate $|V_{PCCR}|$ to 0.94 p.u. if at least 0.07×10^{-3} p.u. reactive power is delivered at PCCS. This resulted in a power demand of $(5.49 + j4.66) \times 10^{-3}$ p.u. at the sending end. The minimum power demand at the sending end (Winterton substation) is achieved when 2.51×10^{-3} p.u. reactive power is delivered at PCCS. The power demand at Winterton substation is then purely active at 5.22×10^{-3} p.u. This situation requires a power factor of 0.89 at PCCS, but reactive power is delivered and not absorbed in this case. This is equivalent to a power angle of -0.47 rad at PCCS. If the sending-end compensator cannot deliver the required 251 kVar reactive power, shunt capacitors can solve the problem. The resulting voltage graphs at $\delta = 0$ rad are shown in Fig. 3.56.

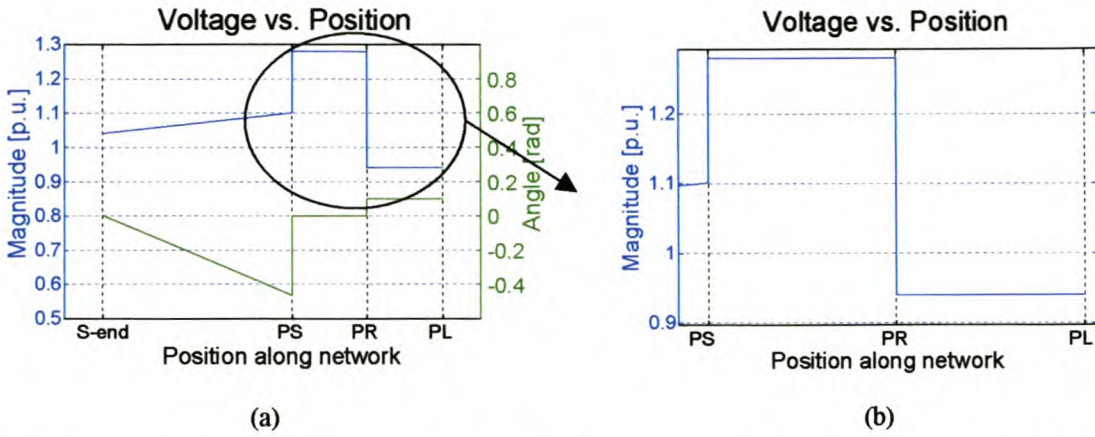


Fig. 3.56: Voltage plots across the Cathedral Peak network with an in-line compensator for $\delta = 0$ rad and reactive power injection of 2.51×10^{-3} p.u. at PCCS

As expected, the injected reactive power raised the voltage magnitude at PCCS. $|V_{PCCS}|$ is 1.1 p.u. for 2.51×10^{-3} p.u. reactive power injection. This falls barely within the NRS 048 voltage regulation specification of $\pm 10\%$ for LV networks [B4]. The injected reactive power is therefore adjusted until $|V_{PCCS}| = |V_{PCCR}| = 0.94$ p.u. This is true for a power angle of -0.20 rad, which results in reactive power delivery of 1.00×10^{-3} p.u. or 100 kVar. The voltage diagrams for this situation are shown in Fig. 3.57.

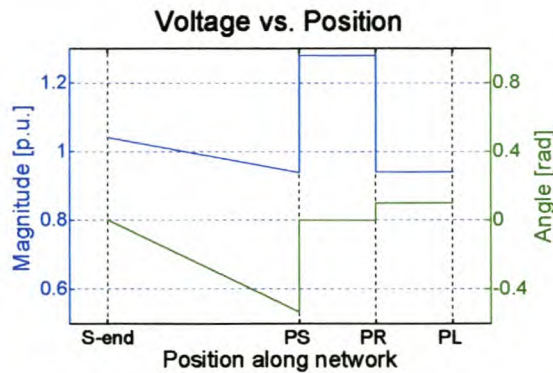


Fig. 3.57: Voltage plots across the Cathedral Peak network with an in-line compensator and reactive power injection of 100×10^{-3} p.u. at PCCS

The required voltage magnitude of 0.94 p.u. is realised at PCCS and PCCR. The raised voltage between PCCS and PCCR is the dc line voltage of 1.28 p.u. The voltage angle is zero between PCCS and PCCR, which is expected, as a dc voltage does not have a phase angle. The small step in the voltage angle at PCCL is due to the compensator at the receiving end that regulates the voltage angle at 0.1 rad, as specified.

The dc link current is visible in the current graphs in Fig. 3.58 between PCCS and PCCR. As expected, the current magnitude drops across the dc link due to the higher voltage. Similar to

the dc voltage angle, the current angle is also zero across the dc link, as expected. With a unity power factor load, the load current phase angle is expected to equal θ_{PCCR} , which is evident in the current angle plot. The current angle is 0.1 rad between PCCR and PCCL, the same as the voltage angle between PCCR and PCCL.

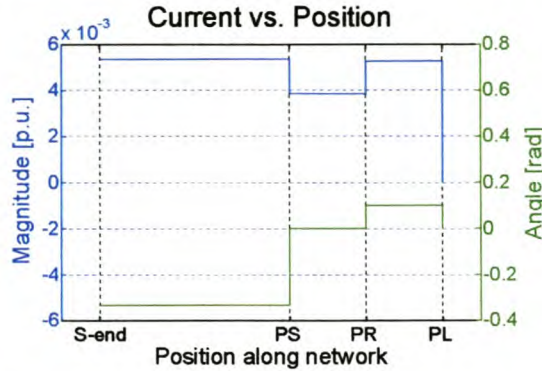


Fig. 3.58: Current magnitude and angle for in-line compensation

The power transfer across the network is represented by the power graphs in Fig. 3.59. As losses are ignored, there is no step at PCCS and PCCR in the active power plot, which is where the ac-dc converters are located. The step in the reactive power plot at PCCS shows the reactive power injection of 1.00×10^{-3} p.u. at the ac side of the sending-end converter. The dc line does not have any reactive power requirements and as the load has a unity power factor, the reactive power is zero all the way from PCCS to the load. As the dc link completely detaches the two ac networks, the power delivery does not vary with δ for the entire network. With δ varying from 0 to π rad and V_{PCCR} constant, this would not have been the case in a single ac network, as seen with the series-shunt compensator.

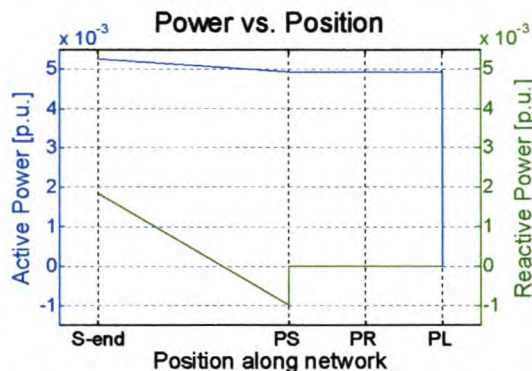


Fig. 3.59: Power transfer in the in-line compensated network

From Fig. 3.59 the sending-end compensator rating is 5.03×10^{-3} p.u., which translates into 503 kVA and 493 kW at a power factor of 0.98. A change in the power factor specified at PCCS does not alter the active compensator rating, as the power demanded by the load

remains constant with V_{PCCR} . The receiving-end compensator rating is 4.93×10^{-3} p.u., which translates into 493 kVA (493 kW).

These ratings are eleven times the rating of the minimum series compensator for the Cathedral Peak network. A possibility is to use an in-line device with a single-phase dc-to-ac converter at the receiving end. This will reduce costs but it is unlikely to match the costs of a series or shunt compensator. The USE device is such an in-line compensator but at 5 kVA it is far too small for this application [A16].

3.7 NEW LINE

The conventional approach to solve the voltage-regulation problem at Cathedral Peak Hotel is to strengthen the network. This can be achieved in many ways, as described in the Eskom report [C4]. Another option is to construct a new 33 kV line from Driel substation to the hotel, which is a distance of 25 km. A 33/11 kV substation will supply Cathedral Peak, while the 33 kV line can be extended to Cathkin substation to provide strengthening, as well as an alternative supply, for the 11 kV Cathkin network. As suggested in option 3 in the Eskom report [C4], Hare conductor will be used. The circuit diagram is shown in Fig. 3.60. As in the uncompensated case, all values are per unit, with $S_{base} = 100$ MVA, $V_{base} = 11$ kV for the sending end, $V_{base} = 230$ V for the single-phase hotel load on the LV side of the 11kV/400V transformer, and $V_{base} = 33$ kV for the receiving end. Winterton substation is included in the sending end voltage source and Driel is modelled by the receiving-end voltage source.

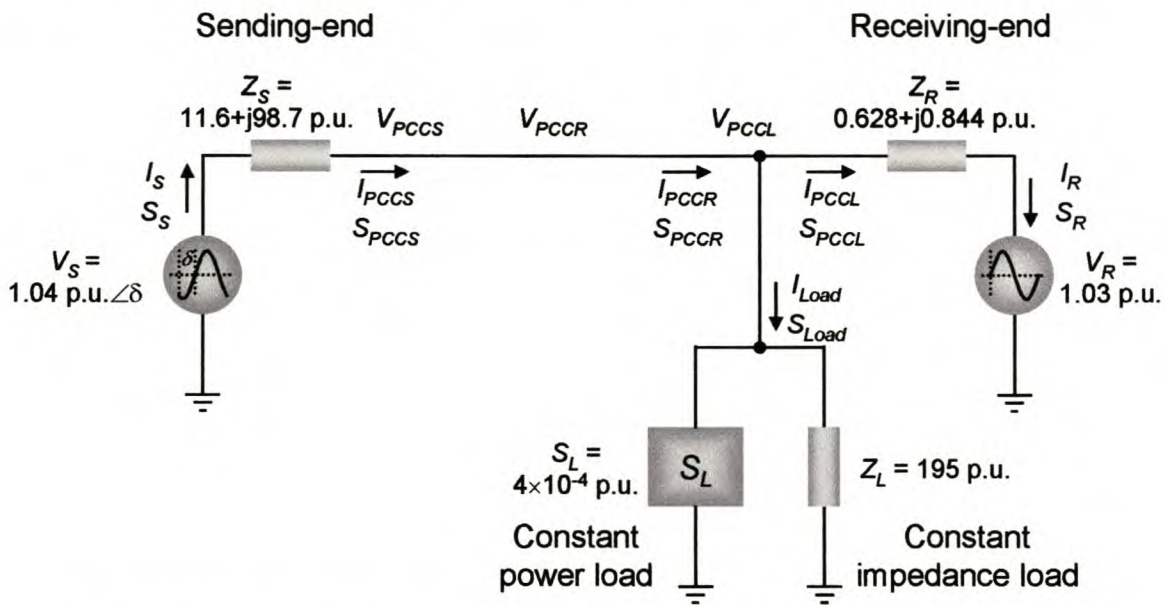


Fig. 3.60: Circuit diagram for the Cathedral Peak network with 33 kV injection from Driel

The above circuit is simplified to Fig. 3.61, as shown in section 3.2.3. V_R' and Z_R' is determined by (3.5) and (3.6), where $V_R = 1.03$ p.u. and $Z_R = 0.628 + j0.840$ p.u. V_R is assumed 1.03 p.u. as this is the operating voltage of the rest of the network. Z_R is the impedance of 25 km of Hare line at 33 kV, as calculated in Addendum A. From Fig. 3.60 the load parameters are: $V_L = 0$, $Z_{VL} = \infty$, $I_L = 0$, $Z_{IL} = \infty$ and $Z_L = 195$ p.u.

$$\begin{aligned}
 V_R' &= \left(\frac{V_R}{Z_R} + \frac{V_L}{Z_{VL}} - I_L \right) \left(\frac{1}{Z_R} + \frac{1}{Z_{VL}} + \frac{1}{Z_{IL}} + \frac{1}{Z_L} \right) \\
 &= \left(\frac{1.03}{0.628 + j0.844} \right) \left(\frac{1}{0.628 + j0.844} + \frac{1}{195} \right) \\
 &= 0.934 \text{ p.u.} \angle -1.86 \text{ rad}
 \end{aligned}
 \tag{3.64}$$

$$\begin{aligned}
 Z_R' &= \frac{1}{\frac{1}{0.628 + j0.840} + \frac{1}{195}} \\
 &= 0.630 + j0.839 \text{ p.u.}
 \end{aligned}
 \tag{3.65}$$

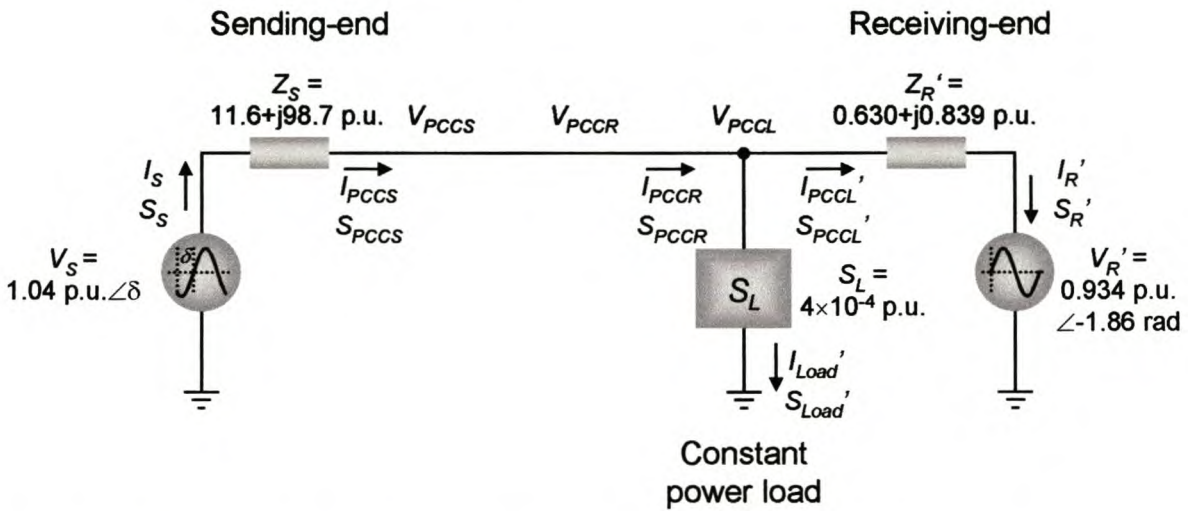


Fig. 3.61: Equivalent circuit diagram for analysing the voltage regulation of the Cathedral Peak network with 33 kV injection from Driel

The user-interface of PSAT allows the user to specify the parameters of Fig. 3.60 directly. PSAT does the above simplifications to be able to calculate V_{PCCR} by solving equations (3.12) and (3.13) in section 3.2.3.

The voltage diagrams for the above network are shown in Fig. 3.62. The benefit of plotting in three dimensions is evident as the voltage now varies with both δ and position along the network. The respective sending- and receiving-end voltage magnitudes are 1.04 p.u. and 1.03 p.u., as specified. As δ increases, the difference between the vectors V_S and V_R also increases, resulting in a current that rises with δ . This is evident in the current magnitude plot in Fig. 3.64. With I_{PCCS} and I_{PCCR} rising with δ , it is expected that the voltage drops across Z_S and Z_R will also rise with δ . This can be seen in the voltage magnitude plot. The reactance of Z_R is two orders of magnitude smaller than the reactance of Z_S . The change in voltage angle

across Z_R is therefore nearly imperceptible in Fig. 3.62. This is the same reason why the reactive power across Z_R appears to be constant in Fig. 3.65. $|V_{PCCR}|$ never drops below 1 p.u., and therefore adheres to NRS 048 [B4] for all δ , as shown for 33 kV transmission in Fig. 3.63. If 11kV transmission was used instead of 33 kV from Driel (Fig. 3.63), the maximum value of δ where the voltage regulation adheres to NRS 048 ($|V_{PCCR}| \geq 0.9$ p.u.) is 2.41 rad. This is due to a nine-fold increase in Z_R , attributed to the change in Z_{base} . For 11 kV, V_{base} for the receiving-end changes to 11 kV, therefore Z_{base} is 3^2 smaller, which increases Z_R with a factor of nine. Under normal circumstances, the network will never be operated at voltage angles (δ) larger than 0.7 rad, but 11 kV transmission may become inadequate for future load increases on the network.

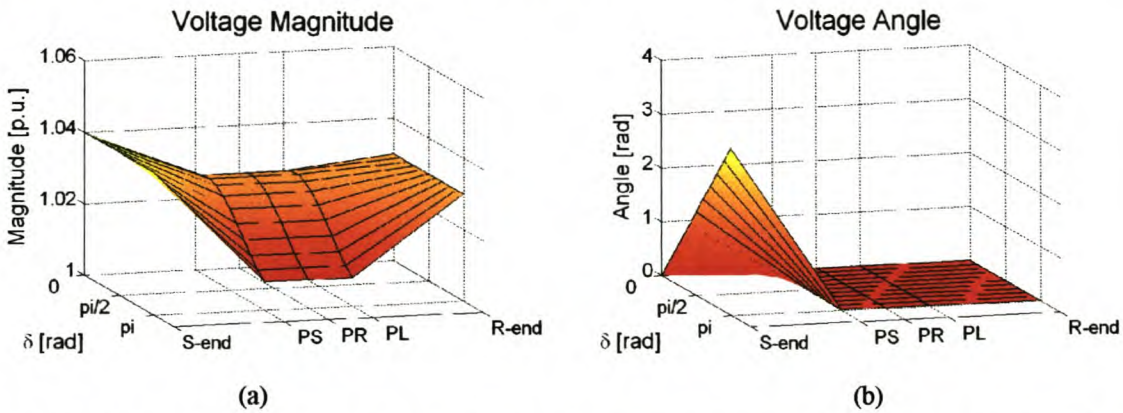


Fig. 3.62: Voltage diagrams for the Cathedral Peak network with a new 33 kV line added from Driel

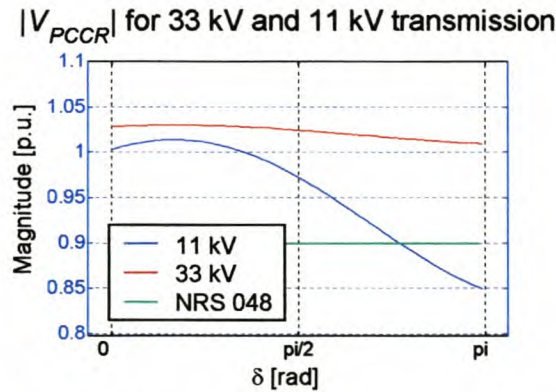


Fig. 3.63: $|V_{PCCR}|$ for 33 kV and 11 kV transmission from Driel

As specified in Fig. 3.60, the voltage angle at the sending end is δ and zero at the receiving end. θ_{PCCR} follows the voltage angle at the receiving end more closely than that at the sending end. This is due to the large difference between Z_R and Z_S ; Z_S is more than six times larger than Z_R . Compared to I_{PCCS} and I_{PCCL} , I_{Load} is very small. This is evident from the small step at PCCL in the current graphs of Fig. 3.64. If the relatively small load current is ignored, the

same current flows through Z_S and Z_R . The voltage drop across the larger Z_S is therefore also larger than the voltage drop across Z_R .

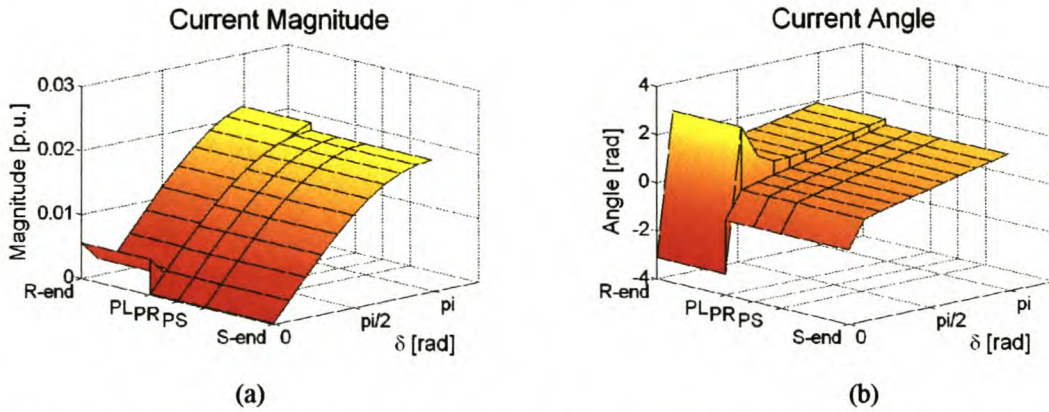


Fig. 3.64: Current diagrams for the Cathedral Peak network with a new line added from Driel

To properly show all facets of the current graphs, the orientation of these graphs differs from that of the other graphs in this section. The current graphs are rotated by 110° counter-clockwise around the Z-axis. The step in the angle of I_{PCCL} at $\delta = 0$ rad is a theoretical step from $-\pi$ to π to keep the current angle in the primary interval between $-\pi$ and π . At $\delta = 0$ rad, $I_{PCCS} = 0$ p.u., and therefore I_{PCCL} must be negative because I_{Load} is positive and $I_{PCCS} = I_{PCCL} + I_{Load}$.

The power transfer graphs in Fig. 3.65 show how power transfer is shared between Winterton and Driel substations. For a particular δ , the power plot indicates the power flowing at a specific position in the network. Positive power flows from Winterton to Driel.

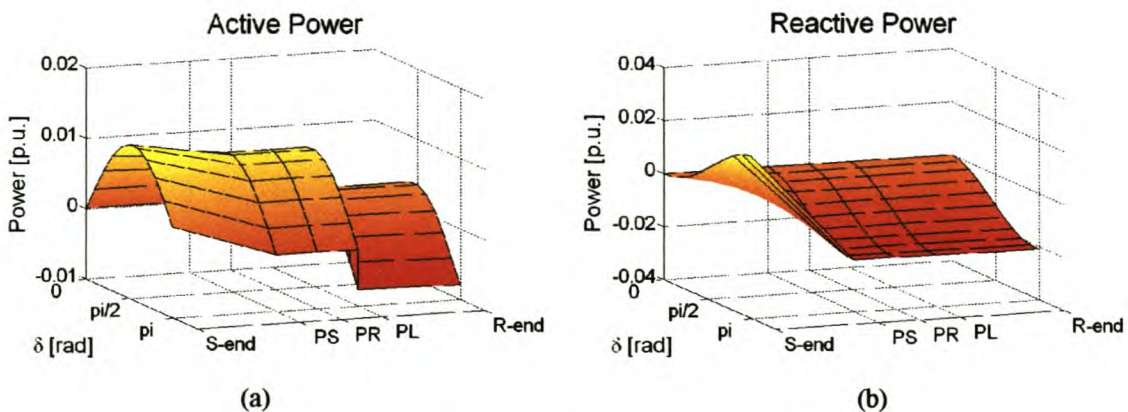


Fig. 3.65: Power transfer across the Winterton-Driel network

The two-dimensional cuts parallel to the network position axes at $\delta = 0.616$ rad and $\delta = 2.38$ rad are shown in Fig. 3.66. For both graphs the active power at PCCL is zero, therefore Winterton supplies all active power required by Z_S (0.428×10^{-3} p.u. and

4.17×10^{-3} p.u.) and the load (5.74×10^{-3} p.u. and 5.65×10^{-3} p.u.) at these specific voltage angles. Driel supplies the active power required by Z_R (3.37×10^{-6} p.u. and 210×10^{-6} p.u.). The negative power values at Driel indicate that power is supplied to the network. Nearly all reactive power values in Fig. 3.66 are negative, indicating that Driel supplies nearly all the reactive power requirements of this network, except for locations close to Winterton. Due to the relatively small Z_R , the change in active and reactive power from PCCR to the receiving-end is very small.

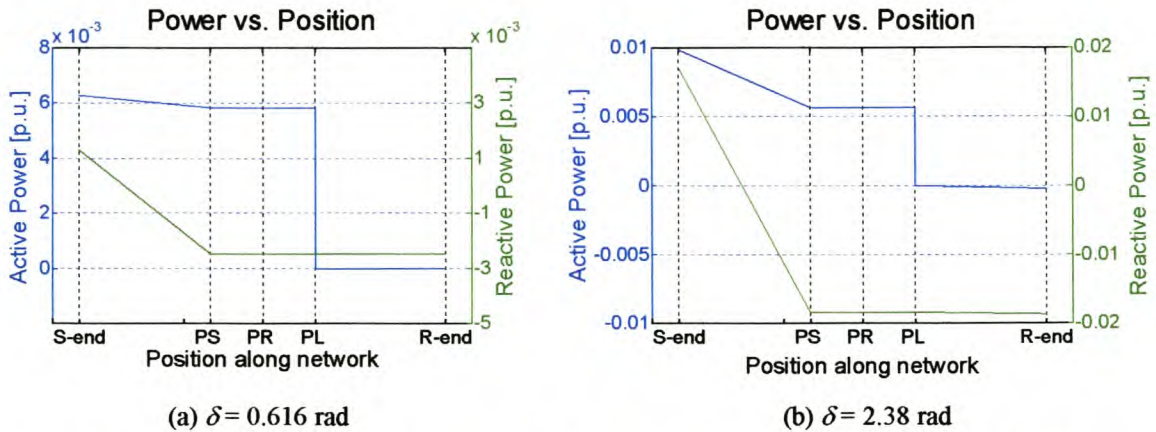


Fig. 3.66: Power transfer across the Winterton-Driel network at $\delta = 0.616$ rad and $\delta = 2.38$ rad

In Fig. 3.65 it can be seen that for $0.616 \text{ rad} < \delta < 2.38 \text{ rad}$, the active power of more points on the network will be positive and therefore supplied from Winterton. Similarly, values of δ smaller than 0.616 rad or larger than 2.38 rad will bias the active power delivery to Driel. As δ increases, both Winterton and Driel deliver more reactive power. δ can therefore be used to control the allocation and flow of power across the network. This control strategy is briefly discussed in Chapter 5.

From the above graphs it is clear that a new 33 kV line from Driel substation solves the voltage regulation problem at Cathedral Peak. A further advantage is an increase in network stability and security as Cathedral Peak is fed from more than one source. This is, however, a very expensive option that will cost more than R 2.5 million in 2000 prices [C4]. It is clear that the other network options previously analysed are much more cost-effective solutions to the voltage-regulation problem.

3.8 SUMMARY

Voltage regulation and voltage regulators were discussed in this chapter. This is already a well-covered subject in electrical engineering [A5], [A6], [B1], [B2], [B6], [C2], [C4] and [C5], but always with approximations in terms of losses and small network voltage angles. As proved by the Cathedral Peak case study and the case study in Chapter 4, the conditions for these approximations are not valid in lossy distribution networks and networks equipped with compensators.

The analyses in this chapter continued with the equivalent circuit in Fig. 3.1, which was developed without approximations in section 3.2. Equations were developed without any assumptions to represent the voltage regulation in this circuit. These equations were implemented in an analysis tool and the resulting three- and two-dimensional voltage, current and power graphs were presented.

Generic voltage regulators in terms of a new line as well as shunt, series, series-shunt and in-line compensators were thoroughly investigated. Equations to model these network options were developed and implemented in PSAT. The Cathedral Peak case study was presented as an integral part of the chapter both to demonstrate the practical application and to verify the developed equations.

3.8.1 Equations

To solve the network in Fig. 3.1 properly, a different approach had to be taken in the development of the equations for every compensator. The starting point for the shunt compensator was the sum of power injected at the PCC, which follows from Kirchoff's Current Law. The equations for the series compensator followed from the equation for the power injected by the compensator. Losses had to be ignored in the series-shunt compensator to obtain a solution for the circuit. This seems to contradict the aim of this study, but this step is valid because the compensator losses are insignificant compared to the losses in the network impedances. The losses for the in-line compensator were also ignored with the same reasoning. For these two compensators the power required at PCCS was the starting point for the development of their equations.

The equations are long and tedious to work with and some, as for the minimum series compensator, are fourth order, which is the highest order for which explicit solutions exist. Great care had to be taken in the implementation of these equations. Firstly, the valid and applicable solutions must be identified out of up to four solutions for every problem. From these solutions the best one must be chosen, which can also be difficult if more than one solution deliver nearly similar results.

Another possibility is to solve the above networks with iterative methods such as Newton-Raphson. Many of the above problems are then eliminated, but another set of problems such as local minima and singular matrices are then encountered. A big advantage of iterative procedures is that little, if any, network simplification is needed and mesh networks can be handled.

This study was specifically aimed at radial networks; therefore, a solution for mesh networks was not required. In an attempt to obtain interpretable equations, the solutions were developed analytically. However, the resulting equations are so long and complicated it is impossible to interpret them constructively. With some equations of the fourth degree, the possibilities of extending these methods to more complicated networks are very limited. Further dimensions will be added to the equations, forcing the use of iterative methods to solve the analytical equations.

It is therefore concluded that these analytical equations work very well for the current scope of PSAT in terms of network complexity, but that iterative solutions will be required for networks that are more complicated.

3.8.2 Cathedral Peak case study

The Cathedral Peak case study thoroughly validated the above equations and served as a showcase for their practical application. Table 3.1 is a summary of the findings for the voltage regulation at Cathedral Peak Hotel.

Table 3.1: Summary of voltage regulator performance in the Cathedral Peak network

Compensator	Control strategy	$ V_{PCCR} $ [p.u.]	Device rating	
			[kVA]	[kW]
None	None	0.86		
Shunt	Purely reactive	0.94	101	0
	Purely active	0.94	160	160
	Minimum rating	0.94	82	46
Series	Purely reactive	0.94	123	0
	Minimum rating	0.94	46	41
Series-shunt	No energy storage	0.93	1038	420
In-line	No energy storage	0.94	996	986

Without compensation, the voltage regulation at Cathedral Peak Hotel drops to 0.86 p.u., which is well below the level of 0.9 p.u. required by NRS 048 [B4]. The construction of a new 33 kV line from Driel substation solved the problem, but at a very large capital outlay. One advantage is that network security increases significantly as Cathedral Peak can then be fed from two sources.

All five applicable shunt and series compensators solve the voltage-regulation problem. Purely reactive shunt compensation requires 101 kVar reactive power injection. This is less than purely reactive series compensation (123 kVar). Should the standby generator be used as a synchronous condenser, reactive shunt compensation is more cost effective to implement, has a faster response time and requires less reactive power injection. Purely active compensation is probably the most cost-effective solution to implement, as the standby generator of the hotel can be converted into a distributed generator. The 450 kVA generator can easily supply the required 160 kW.

The minimum series compensator has the lowest rating of all options investigated thus far. At 45.7 kVA, it undercuts the second lowest rating option, which is the minimum shunt compensator, by 36.5 kVA. However, this is a costly option, as a transistor-based series device with a continuous energy supply is needed. The minimum-rating shunt device will probably require a smaller capital outlay as the standby generator can provide a dynamic

margin of reactive power. The bulk reactive power can be supplied with capacitor banks switched with a simple controller.

For this network a series-shunt or in-line device is definitely out of the question. The ratings of these devices are at least one order of magnitude larger than the series and shunt devices. Due to the excessive ratings for $\delta > \pi/2$ rad, the maximum series-shunt device ratings for $\delta \leq \pi/2$ rad are shown. For other networks these devices can be very effective. It is generally assumed that an in-line device is of higher rating than a series-shunt device. For this specific network the series-shunt device has a 3% higher rating than the in-line device. However, neither the series-shunt device, nor the in-line device, was able to attain the required 0.94 p.u. for $|V_{PCCR}|$ with a unity power factor at PCCS. The in-line device was only able to achieve this at a leading power factor of 0.98 and a device rating of 996 kVA. Under the same conditions the series-shunt device rating is 626 kVA for $\delta < 1.27$ rad. Normally, a network is never operated at voltage angles higher than 0.7 rad.

From the above discussion, the purely active or minimum-rating shunt device seems to be the most effective solution in terms of power rating and cost. Detailed studies should now be conducted to determine the best solution for Cathedral Peak.

3.8.3 Analysis tool

If a detailed study were conducted for every one of the above options, an enormous time and financial budget would be necessary. PSAT made it possible to compare the technical benefits and disadvantages of every option within a matter of hours. This is an inevitable tool for network planners of the near future, who will have to consider such options to limit capital costs and stretch the network. The innovative three-dimensional voltage, current and power graphs are a great aid in understanding network and compensator interaction. Vector diagrams and two-dimensional cuts enhance comprehension of network behaviour even further.

CHAPTER 4 ZIMBANE CASE STUDY

4.1 INTRODUCTION

4.2 UNCOMPENSATED NETWORK

4.3 SHUNT COMPENSATORS

4.3.1 Voltage regulation with an SVC

4.3.2 Minimum-rating shunt device

4.4 SERIES COMPENSATORS

4.4.1 Purely reactive series compensator

4.4.2 Minimum-rating series compensator

4.5 SERIES-SHUNT COMPENSATION

4.6 IN-LINE COMPENSATION

4.7 SUMMARY OF THE ZIMBANE CASE STUDY

4.1 INTRODUCTION

Chapter 1 identified two network configurations that often experience voltage-regulation problems. These were rural radial networks and rural mesh networks with two relatively stiff voltage sources. The Cathedral Peak network is a typical example of a rural radial network which, because of the single source, produced less complicated voltage, current and power graphs. This chapter presents the Zimbane case study as an example of a typical twin-source mesh network. First, the uncompensated network is analysed to determine the voltage regulation at the PCC, Zimbane substation. The rest of the chapter is devoted to analysing the four generic compensators to regulate the voltage at Zimbane substation. Various control strategies are also investigated.

The Pembroke-Zimbane-Eros system is a mesh network with two voltage sources and is situated in the Eastern Cape, South Africa. Pembroke substation is near East London and Zimbane substation lies 230 km further near Umtata, as shown in Fig. 4.1. Eros substation is 180 km from Umtata near Kokstad in Natal. Two pairs of small hydro-generators are located near Zimbane substation at First and Second Falls, respectively. With the PCC at Zimbane substation, Pembroke substation is modelled as the sending end and Eros substation as the receiving end.

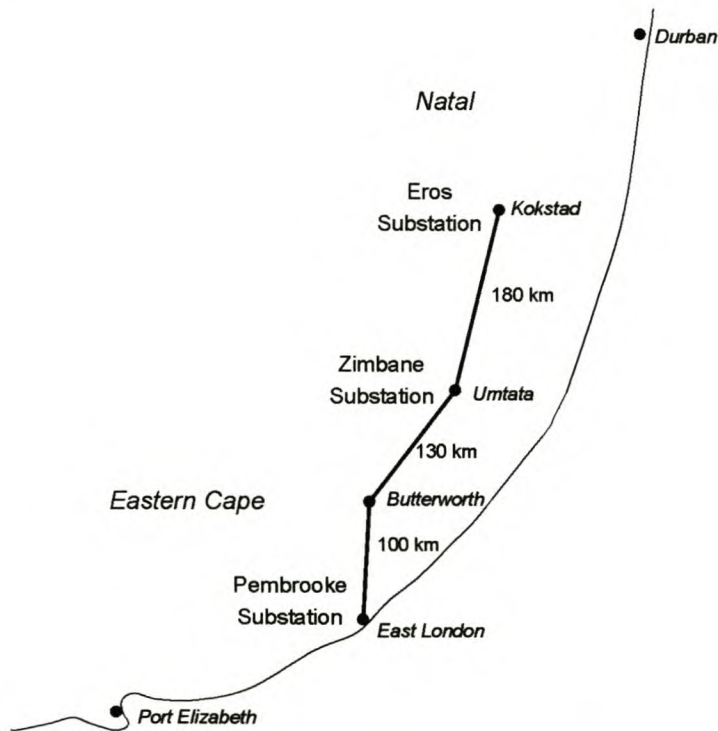


Fig. 4.1: Pembroke-Zimbane-Eros network

With the incorporation of the Transkei Electricity Corporation in 1994, Eskom inherited the Zimbane substation, which experienced voltage-regulation problems under full load. After many simulations and extensive studies, Eskom moved the Buffalo SVC to Zimbane substation for voltage-regulation purposes [A17]. Recently, the line between Zimbane and Eros substations was completed. This improved the voltage regulation at Zimbane significantly. However, from an operations viewpoint the Zimbane network is not very easy to operate [C6], [C7] and voltage regulation can still be a problem sometimes.

Using the developed simulation tools, this system is first simulated without the SVC. Thereafter a number of voltage-regulation solutions are investigated:

- A purely reactive shunt compensator, e.g. an SVC
- A minimum-rating shunt compensator, e.g. the hydro-generators combined with the SVC
- A purely reactive series compensator, e.g. a TCSC
- A minimum-rating series compensator, e.g. a DVR
- A series-shunt device, e.g. a UPFC
- An in-line device, e.g. a back-to-back converter.

The simulated results for the uncompensated and the SVC-compensated networks will be shown to agree with Eskom's simulated and practically measured data, proving the effectiveness of PSAT as a quick, accurate and easily comprehensible analysis tool. Practical data of the other solutions are not available, since the SVC is currently installed.

4.2 UNCOMPENSATED NETWORK

The network configuration is pictured in Fig. 4.2, with Zimbane substation feeding a peak load of 52 MW. Zimbane is fed from Pembroke with a 132 kV Wolf sub-transmission line over a distance of 230 km. Various loads are situated along the line. From simulations done by Eskom in PSSE, the voltage and power values at both Pembroke and Zimbane substations are available for this scenario [C1]:

- Pembroke voltage: $V_{SS} = 1.05$ p.u. $\angle 0.251$ rad
- Pembroke power: $S_{SS} = 0.826 + j0.563$ p.u.
- Zimbane voltage: $V_{PCCS} = 0.92$ p.u. $\angle 0$ rad
- Zimbane power: $S_{PCCS} = 0.565$ p.u. $\angle 0$ rad.

Due to local power factor correction, the power factor at Zimbane is very good and for all practical purposes taken as unity. PSSE takes the voltage angle at Zimbane as zero, with the voltage angle at Pembroke indicating the voltage angle difference across the network. It is emphasised that the receiving-end voltage angle is the reference voltage angle in PSAT. The above values are only used to calculate the equivalent network parameters. Using these values, the equivalent network parameters V_S and Z_S are calculated from equations (3.1) to (3.4). These equations assume that all loads along the Pembroke-Zimbane line are constant impedance loads. For the rural Eastern Cape, this is a valid assumption.

$$\begin{aligned} Z_{Series} &= \frac{|V_{SS}|^2 - V_{PCCS} V_{SS}^*}{S_{SS}^*} \\ &= 2.64 \times 10^{-3} + j0.293 \text{ p.u.} \end{aligned}$$

$$\begin{aligned} Z_{Shunt} &= \frac{|V_{PCCS}|^2 V_{SS}^*}{S_{SS}^* V_{PCCS}^* - S_{PCCS}^* V_{SS}^*} \\ &= 1.41 + j1.62 \text{ p.u.} \end{aligned}$$

$$\begin{aligned} V_S &= \frac{V_{SS} Z_{Shunt}}{Z_{Series} + Z_{Shunt}} \\ &= 0.948 \text{ p.u. } \angle 0.171 \text{ rad} \end{aligned}$$

$$\begin{aligned} Z_S &= \frac{Z_{Series} Z_{Shunt}}{Z_{Series} + Z_{Shunt}} \\ &= 0.0234 + j0.263 \text{ p.u.} \end{aligned}$$

The network data for the Zimbane-Eros line was obtained from Eskom [A17], [C6], [C7]. A number of different configurations and loads exist along this line and operating conditions vary significantly. V_R and Z_R are therefore approximate values that aim to present a worst-case scenario: $V_R = 1$ p.u. and $Z_R = 0.144 + j0.164$ p.u. The accuracy of these values is adequate for this study as the aim is not to do a detailed network study, but to show the practical usage of PSAT.

The above system is depicted in Fig. 4.2. The power dissipated in the distributed capacitance of the line is less than 1% of the total reactive power dissipated in the line and is therefore ignored. There seems to be a discrepancy between the voltage magnitudes of the sending end (0.948 p.u.) and that of Pembroke substation (1.05 p.u.). This is only a result of the Thévenin source transformations made in simplifying the network.

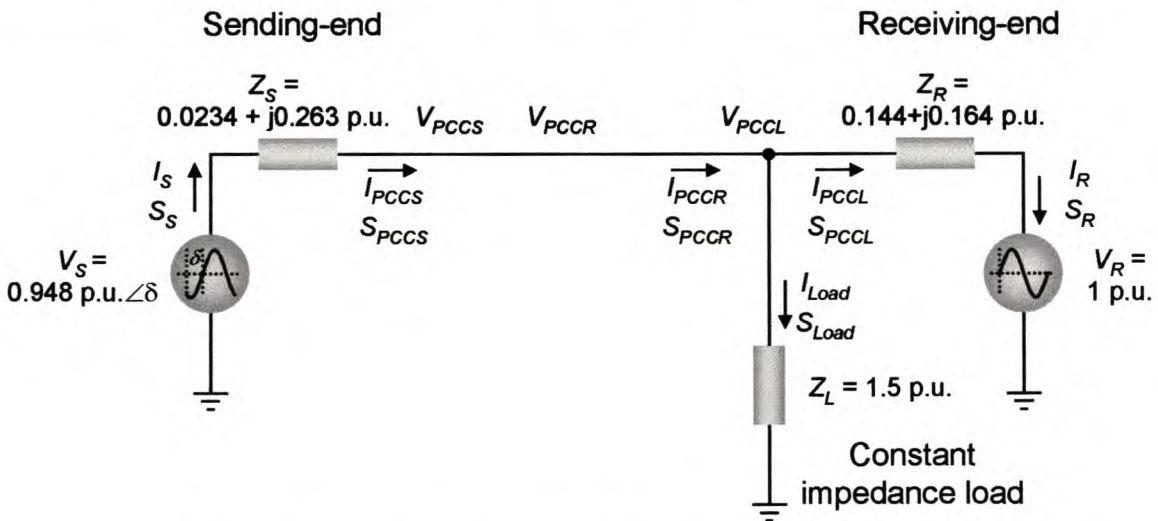


Fig. 4.2: Diagram of the simplified Pembroke-Zimbane network

Fig. 4.3 to Fig. 4.6 show the simulation results for the above network. The voltage magnitude at the PCC is clearly below 1 p.u. and drops as low as 0.341 p.u. for $\delta = \pi$ rad. The voltage magnitudes at the sending and receiving ends match the values specified for V_S and V_R . The voltage angles also correspond to the values specified in Fig. 4.2. The sending-end voltage angle is equal to δ and the receiving-end angle, which is the reference angle, is zero for all δ . For the dual-source network, the voltage drops across Z_S and Z_R are not constant for all δ , as with a radial network. The voltage drops depend on the network parameters, the load power and δ .

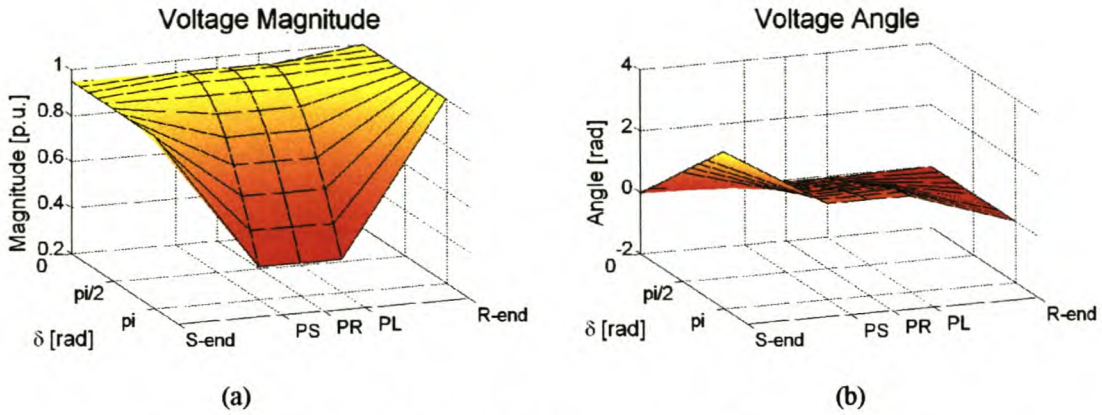


Fig. 4.3: Voltage magnitude and angle across the Zimbane network

The voltage profile at the PCC for δ ranging from 0 to π rad is shown in Fig. 4.4. This graph is a cut parallel to the XZ-plane at PCCR. Both the voltage magnitude and angle are shown. The voltage magnitude is less than the NRS 048 minimum of 0.95 p.u. for $\delta < 0.0636$ rad and $\delta > 1.20$ rad. Normal operating conditions normally falls within this interval; therefore, there should not be voltage-regulation problems under normal operating conditions. The problematic voltage regulation for small δ can worsen under other operating conditions and it is worthwhile investigating the influence of the SVC and hydro-generators further. The voltage-regulation problem is caused by the specific values of the network parameters V_S , V_R , Z_S and Z_R and remains undesirable, even without any load connected at PCCL. The load worsens the voltage-regulation problem, as expected. For small δ the PCC voltage angle is negative due to the voltage drop across Z_S and Z_R with a very small sending-end voltage angle. The load currents drawn through Z_S and Z_R result in a voltage drop across the impedances.

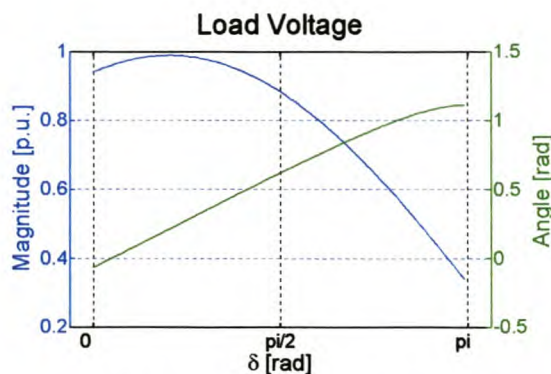


Fig. 4.4: Voltage profile at the PCC

Very little active power is dissipated across Z_S , as shown in Fig. 4.5. This corresponds to the resistive part of Z_S that is an order of magnitude smaller than the other impedances in the network. The step in active power at PCCL is the power dissipated by the load at Zimbane

substation. No step is present in the reactive power graph at PCCL, confirming the unity power factor load at Zimbane. Although distorted by the losses in the network, the reactive power demand at the sending and receiving ends are typical co-sinusoidal functions of δ , while the active power demands are sinusoidal functions of δ . These characteristics are typical of power transfer between two voltage sources via a lossless network [B6]. The chosen direction of positive power flow is shown in Fig. 4.2. This explains the seeming demand for reactive power at the receiving end.

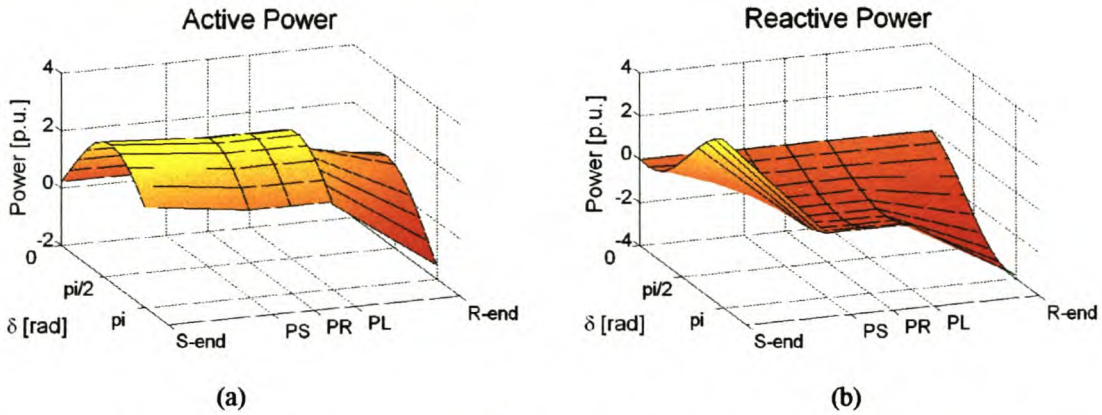


Fig. 4.5: Active and reactive power dissipated across the Zimbane network

The power vector diagram in Fig. 4.6 also shows the load power, S_{Load} , to be purely active. S_S delivers both active and reactive power. Due to its low resistance, Z_S consumes very little active power. The sum of these two vectors gives the power that flows through PCCR, namely S_{PCCR} . S_{Load} is subtracted from S_{PCCR} to result in the power level at PCCL, S_{PCCL} . The power drop across S_{ZR} and the power supplied by S_R completes the power vector diagram. S_R seemingly consumes power, but if the direction of power flow in Fig. 4.2 is taken into account, it can be seen that it delivers power.

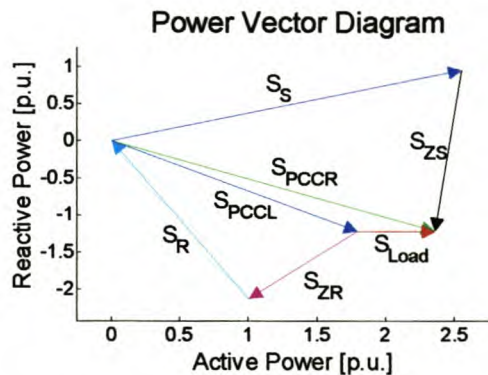


Fig. 4.6: Power vector diagram at $\delta = 1.41$ rad

PSAT shows the power consumed by the load as a separate two-dimensional graph, as shown in Fig. 4.7. This enables the user to determine the load power at a glance without any

calculations. The load at Zimbane is modelled as a constant impedance load to portray the rural loads, which are mostly incandescent lamps and heating elements. With Z_L constant, the load power follows the same profile as the voltage across the load, namely V_{PCCR} . This is clearly visible when Fig. 4.7 is compared to Fig. 4.4. With the load operating at unity power factor, no reactive power is drawn.

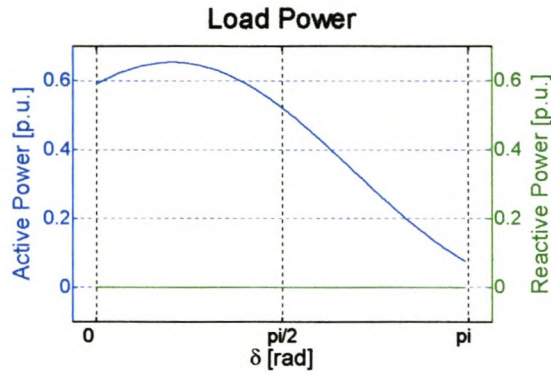


Fig. 4.7: Load power as a function of δ

Shunt, series, series-shunt and in-line devices are subsequently analysed as voltage regulators at Zimbane substation. Various control strategies are also investigated. The required voltage regulation is 1 p.u. in all cases.

4.3 SHUNT COMPENSATORS

With the uncompensated case simulated, the next step is to analyse the shunt-compensated Zimbane network, as shown in Fig. 4.8. The circuit diagrams for the purely reactive and minimum-rating shunt compensators are the same. From an analysis viewpoint, the only difference is the availability of stored energy and the control method.

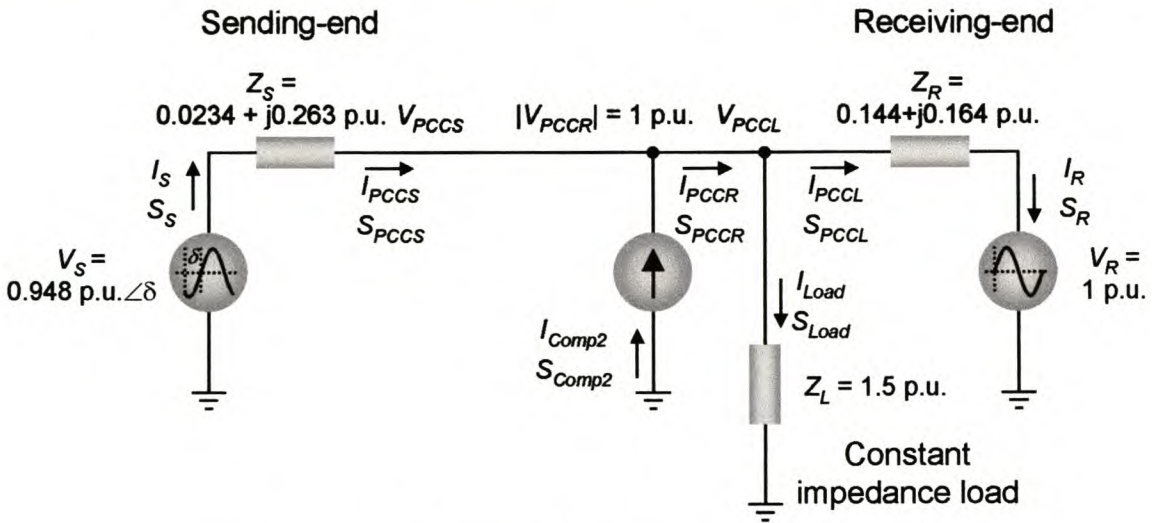


Fig. 4.8: Zimbane power system model with a shunt compensator

4.3.1 Voltage regulation with an SVC

The circuit diagram for the Zimbane network with an SVC is shown in Fig. 4.8. The SVC is classified as a shunt compensator with no energy-storage capabilities. The simulation results are shown in Fig. 4.9 and Fig. 4.11. As expected, the voltage magnitude at the PCC is now unity, verifying the voltage-regulating capability of the SVC. The SVC regulates $|V_{PCCR}|$ purely with reactive power injection. At $\delta = 2.82$ rad the voltage collapses, as an infinite rating is required to regulate $|V_{PCCR}|$ at this point with a purely reactive compensator. This is evident in Fig. 4.10, where the reactive and apparent power graphs lie on top of each other due to the zero active power.

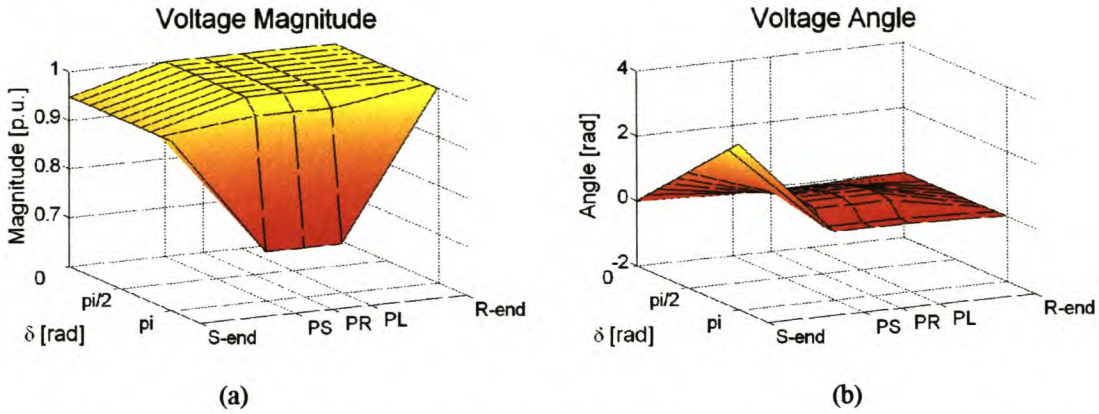


Fig. 4.9: Voltage magnitude and angle across the SVC-compensated Zimbane network

From the voltage angle graph it can be seen that the compensated voltage angle does not differ significantly from the uncompensated voltage angle, except at the point of voltage collapse. This is to be expected because, without energy storage, the SVC can regulate only one network parameter.

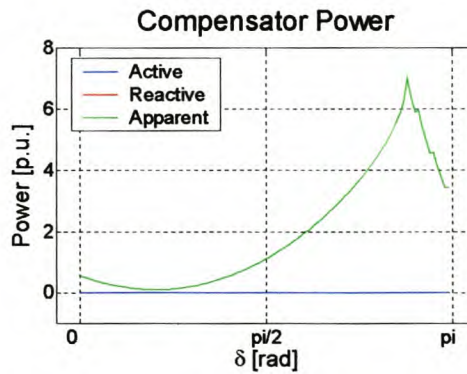


Fig. 4.10: Power requirements of the SVC

The SVC can only inject reactive power, which is clearly visible in Fig. 4.11. The step in reactive power at PCCR is created by the SVC adding its reactive power to the network. No active power is injected by the SVC, as expected. The step at PCCL in the active power graph is due to the unity power factor load at Zimbane. Comparing the active power in Fig. 4.11 with that of Fig. 4.5, one can see the increase in the active power transfer capability of the network. The SVC effectively provides reactive power compensation for the network in its attempt to regulate the magnitude of V_{PCCR} . It is therefore to be expected that the network must gain some active power transfer capability.

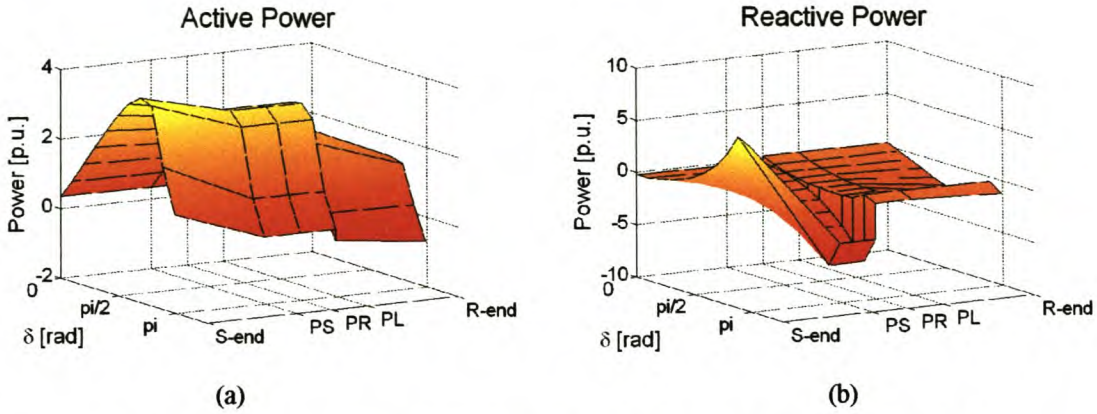


Fig. 4.11: Active and reactive power dissipated across the Zimbane network

With V_{PCCR} larger, the constant impedance load dissipates more power. This is evident when Fig. 4.12 is compared to Fig. 4.7. Due to the regulated V_{PCCR} , the power consumption is constant for all δ up to the point of voltage collapse.

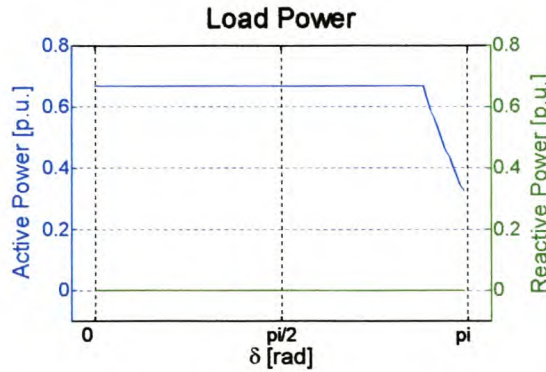


Fig. 4.12: Load power for the Zimbane network equipped with an SVC

The Zimbane SVC has a capacitive rating of 35 MVar or 0.380 p.u. This enables the SVC to successfully regulate $|V_{PCCR}|$ for $0.143 < \delta < 1.12$ rad. The SVC is currently in operation on this network and still performs very well, especially when the line from Eros is not operating [C7]. The results of simulations done by Eskom also lead to this conclusion [A17].

Other device options will now be investigated to show their advantages and disadvantages compared to the SVC. The whole idea of moving an existing SVC to Zimbane substation was motivated by, among other reasons, the cost savings of not buying a new device. If Eskom did not have an SVC available, a new device might have been considered. The choice of devices would then have been much wider.

4.3.2 Minimum-rating shunt device

A minimum-rating device injects both active and reactive power in such a manner that the device rating is a minimum. Shunt devices that can be equipped with energy-storage devices to supply the active power must be transistor-based. Although such devices with ratings higher than 10 MVA are not readily available yet, the hydro-generators at First and Second Falls can be used in conjunction with the SVC to achieve a more effective solution.

Fig. 4.8 is the circuit diagram for the shunt-compensated Zimbane network. The voltage diagrams for minimum shunt compensation are shown in Fig. 4.13.

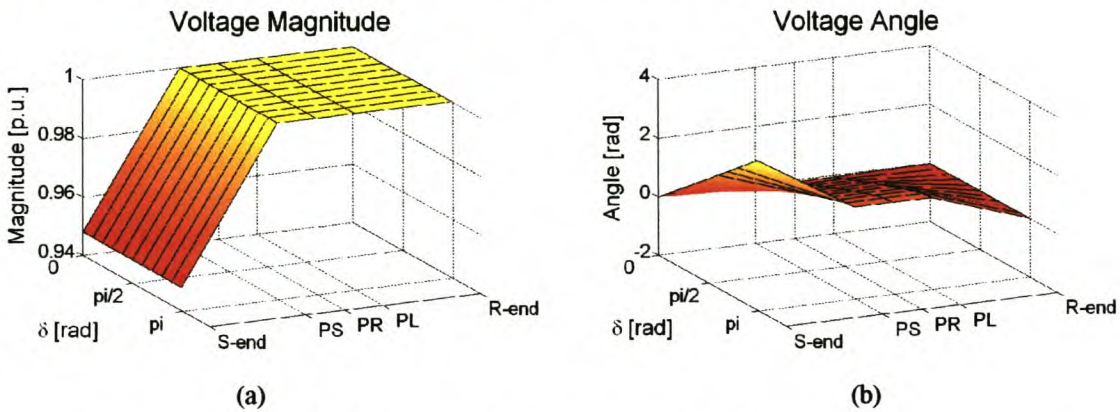


Fig. 4.13: Voltage graphs for minimum shunt compensation

From Fig. 4.13 it is evident that the addition of active power solved the problem of voltage collapse. The minimum-rating shunt compensator regulates $|V_{PCCR}|$ to the required 1 p.u. for all δ . From the power graphs in Fig. 4.14, the rating of the minimum shunt device is less than 0.330 p.u. (30 MVA) for $0.143 < \delta < 1.12$ rad. The respective ratings for the SVC and hydro-generators are 0.290 p.u. (27 MVar) and 0.160 p.u. (15 MW). For the same interval, the stand-alone SVC rating is less than 0.380 p.u. (35 MVA). The total amount of hydro-power available is 17 MW. Therefore, for 15 MW hydro-power, the SVC can be operating at 8 MVar less than when operating stand-alone. Alternatively, if 17 MW hydro-power is used, the voltage can be regulated for $0.0937 < \delta < 1.17$ rad.

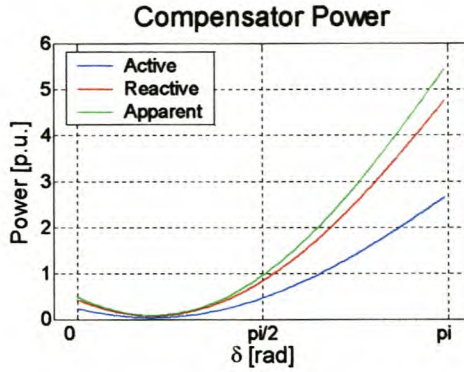


Fig. 4.14: Power graphs for a minimum-rating shunt device

The network power graphs are shown in Fig. 4.15. The effects of the shunt compensator and the Zimbane load are visible in the steps in (a) and (b). The steps at PCCR show the power injected by the shunt compensator. The active power step of the unity power factor load at Zimbane is visible at PCCL in (a). No such step occurs in the reactive power graph. It must be remembered that the constant levels of power between PCCS and PCCR, and PCCR and PCCL are artificial because PCCS, PCCR and PCCL are the same node in Fig. 4.8.

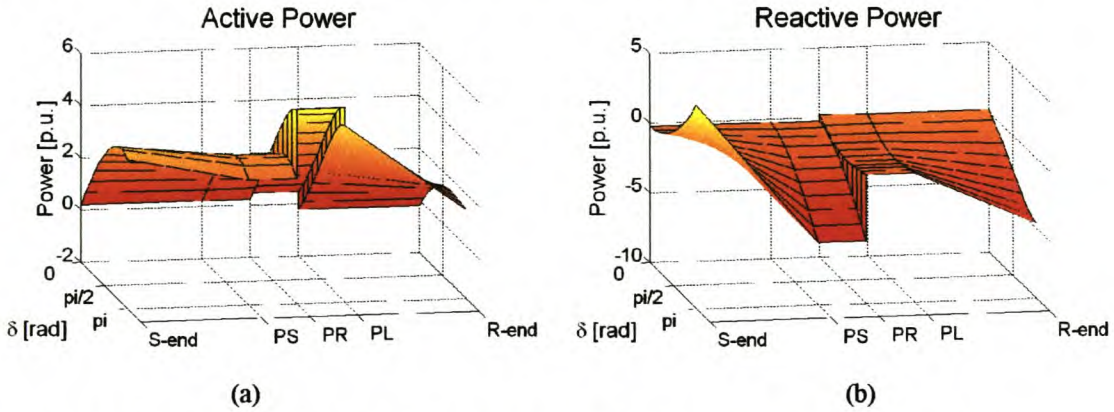


Fig. 4.15: Power graphs for the Zimbane network with a minimum-rating shunt device

A cut along the XZ-plane at the sending end shows the power requirements of the sending end more clearly in Fig. 4.16 (a). This standard plot is available in PSAT. Fig. 4.16 (b) shows a similar graph for the uncompensated network. Comparing (a) to (b), it is evident that, below $\delta = \pi/2$ rad, very little extra power demands are placed on the sending end for the compensated network. Above $\pi/2$ rad, the large increase in the active power graph in (a) indicates that the compensated network has a much larger first-swing stability margin. It can safely be operated at voltage angles as high as $\pi/2$ rad. This aspect of compensation could not be analysed if the PSAT equations relied on often-used approximations such as $\sin \delta \approx \delta$ and $\cos \delta \approx 1$.

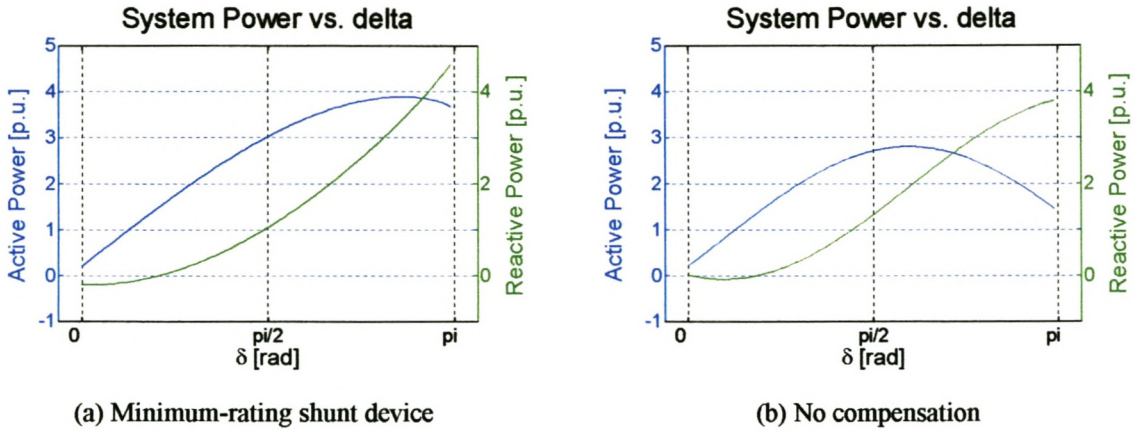


Fig. 4.16: Sending-end power graphs for the Zimbane network

4.4 SERIES COMPENSATORS

Series compensators connected to the Zimbane network, as shown in Fig. 4.17, are investigated in this section. The Eskom report doubted the effectiveness of series compensators due to the good power factor of the Zimbane network [A17]. This section will show their reservations to be justified.

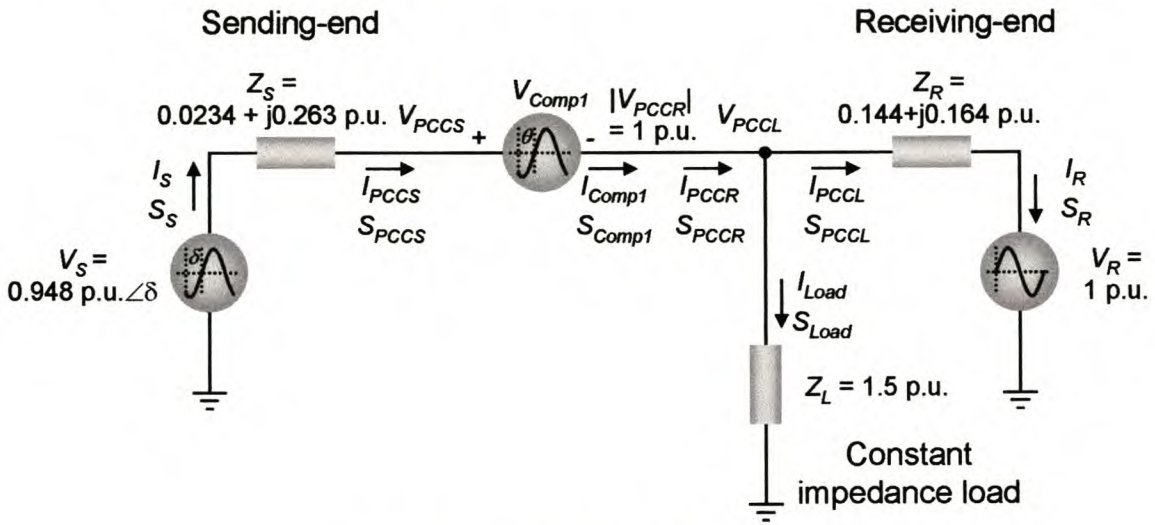


Fig. 4.17: Series-compensated Zimbane network

4.4.1 Purely reactive series compensator

A TCSC or series capacitor can provide purely reactive series compensation for the Zimbane network. The circuit diagram is shown in Fig. 4.17. The equations derived for a purely reactive series compensator in section 3.4.1 have two solutions. The voltage magnitudes of both solutions are shown in Fig. 4.18.

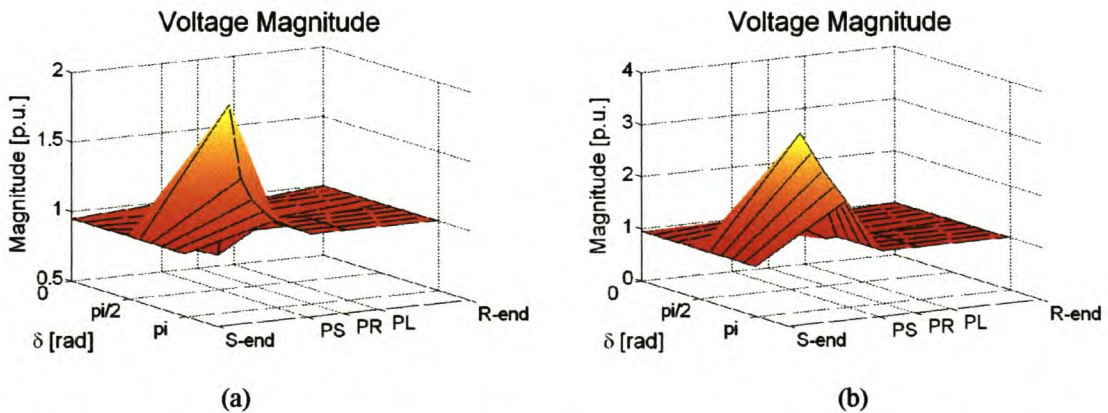


Fig. 4.18: Voltage graphs of the two solutions for a purely reactive series compensator

It is clear that a purely reactive series compensator cannot be used on the Zimbane network. The voltage magnitude just before the compensator at PCCS rises to 1.8 p.u. and 3 p.u. for the

respective solutions. This is due to the combination of a purely active load and Z_S and Z_R that are relatively small compared to the load impedance. This effect looks the same as resonance, but is in fact a voltage regulation problem.

4.4.2 Minimum-rating series compensator

A DVR is an example of a series compensator that can inject both active and reactive power when equipped with an energy resource. This device can therefore be controlled to use the minimum power to achieve the desired voltage regulation at PCCR. Fig. 4.17 shows how a DVR, just like any other series device, is connected to the Zimbane network. For the minimum series compensator, the derived equations in section 3.4.3 have four solutions. Fig. 4.19 shows the voltage magnitude graphs of all four solutions.

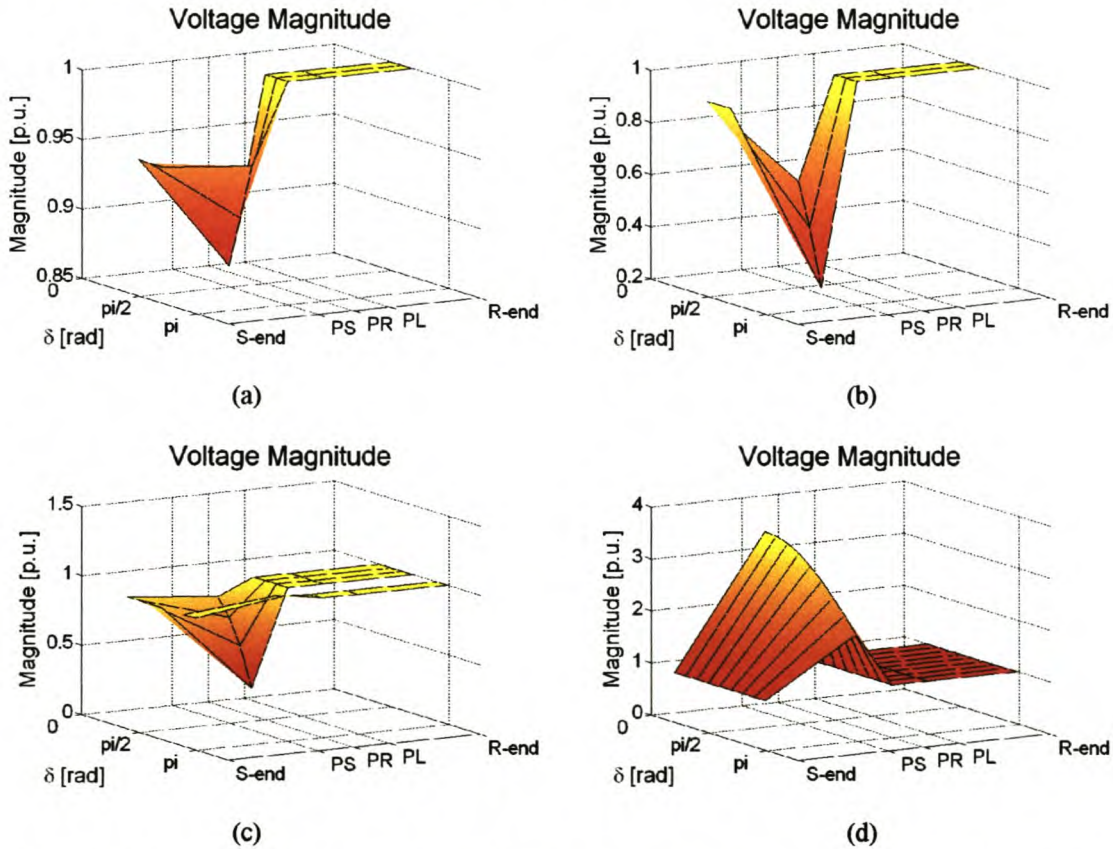


Fig. 4.19: Voltage graphs of the four solutions for a minimum rating series compensator

None of the solutions is valid for $\delta \leq 0.59$ rad; (a), (b) and (c) are only valid for limited values of δ , (d) is a valid solution for $\delta > 0.59$ rad, but causes $|V_{PCCS}|$ to rise up to 3.48 p.u. at $\delta = 0.68$ rad. Not even the addition of active power can render a series compensator viable for the Zimbane network. These findings agree with those of Eskom [A17].

4.5 SERIES-SHUNT COMPENSATION

Series-shunt compensators can be configured in a number of ways, with and without energy resources. One of the popular configurations is shown in Fig. 4.20. The shunt compensator regulates the power factor at PCCS and absorbs active power that is transferred via the dc bus to the series compensator. The series compensator regulates both the voltage magnitude and angle at PCCR. This configuration eliminates the need for an energy resource and has the advantage of power factor control combined with voltage regulation.

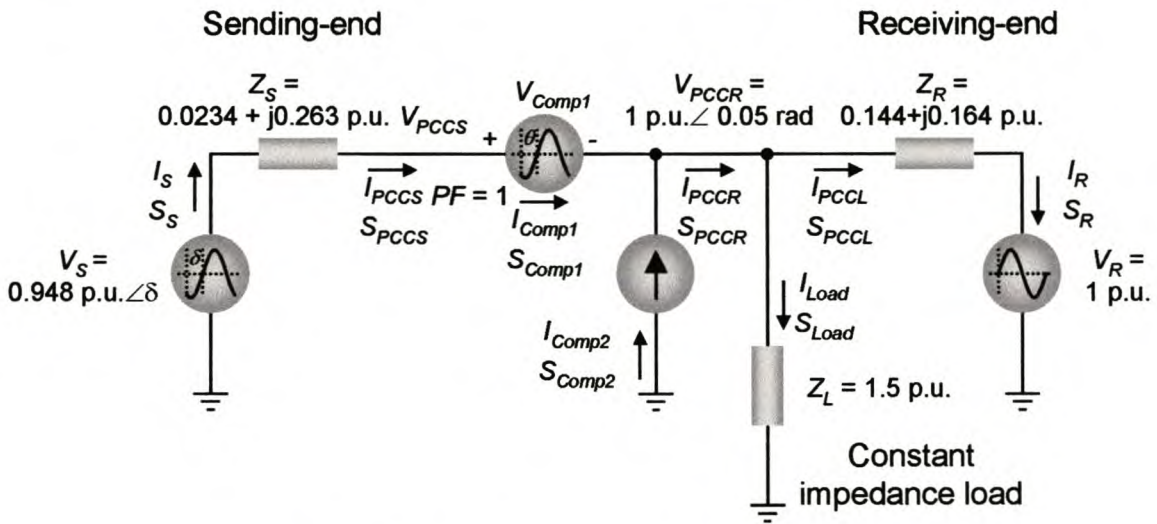


Fig. 4.20: Series-shunt-compensated Zimbane network

For the series-shunt compensator, the network planner must specify both the required magnitude and angle of V_{PCCR} . The power factor at PCCS must also be specified. As shown in Fig. 4.20, the required $V_{PCCR} = 1 \text{ p.u.} \angle 0.05 \text{ rad}$ and the power factor is unity. The resulting voltage graphs are shown in Fig. 4.21.

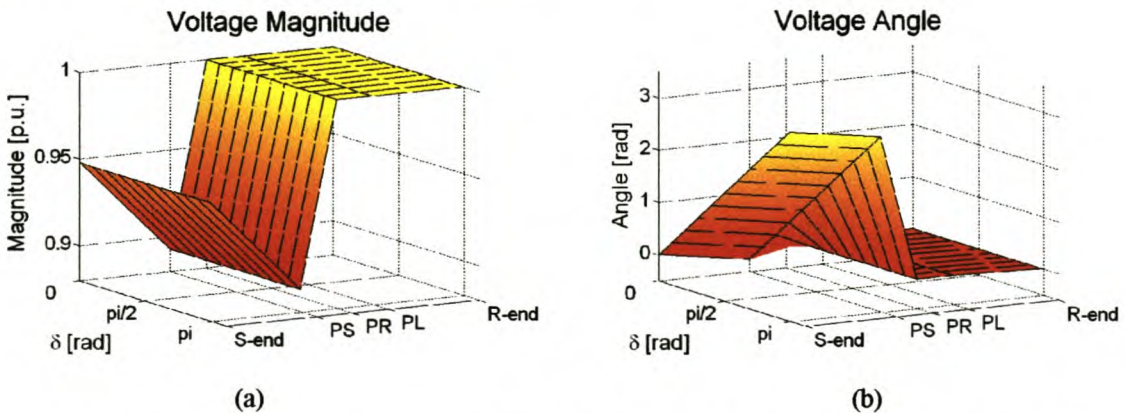


Fig. 4.21: Voltage graphs for the Zimbane network with a series-shunt compensator

In Fig. 4.22 two-dimensional graphs at $\delta = 0$ rad show that the required voltage regulation is attained for both the magnitude and angle at PCCR. These values are maintained for all δ .

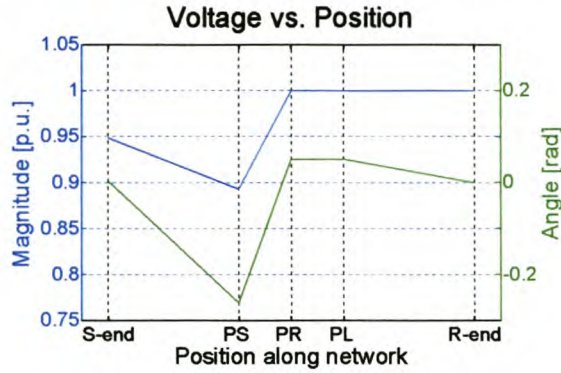


Fig. 4.22: Voltage graphs at $\delta = 0$ rad for the series-shunt compensated network

A significant problem with the series-shunt compensator is the large compensation current required. All active power injected by the device is sourced from the network. Large currents flowing through Z_S result in a large voltage drop across Z_S . Consequently, the voltage at PCCS (just before the compensator) drops to 0.893 p.u., which is lower than the uncompensated voltage level at PCCS for $\delta < 1.53$ rad.

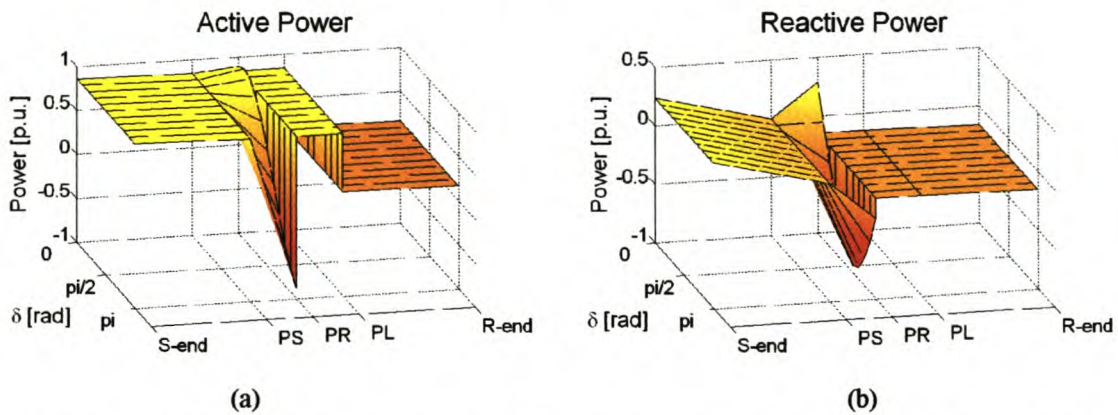


Fig. 4.23: Power transfer across the Zimbane network equipped with a series-shunt device

The power graphs are shown in Fig. 4.23. With V_{PCCR} regulated, the relative voltage angle between PCCS and the receiving end stays constant. The active and reactive power transfer between these two points is therefore constant for all δ . Due to the small angle of V_{PCCR} , this power transfer is only $(7.60 + j8.60) \times 10^{-3}$ p.u. and therefore difficult to see in Fig. 4.23.

The power factor is controlled to be unity at PCCS, as required. The zero reactive power for all δ at PCCS in (b) confirms this. Losses are ignored; therefore, the series device injects all active power absorbed by the shunt device back into the network. The result is a constant

power demand from the sending end for all δ . The required device ratings are depicted in Fig. 4.24. The series device power is shown in (a) and the shunt device power in (b).

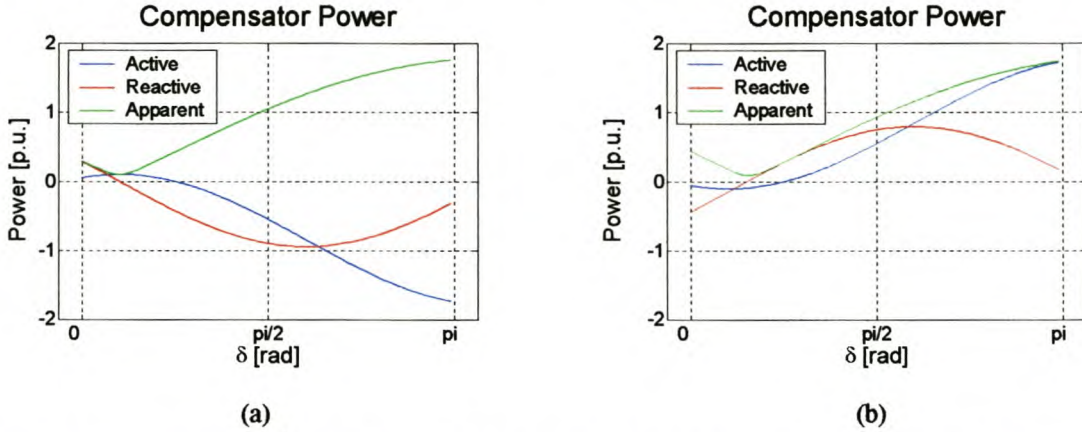


Fig. 4.24: Power required by the series-shunt device: (a) series compensator (b) shunt compensator

The active power ratings of the series and shunt devices are equal but opposite in sign. This corresponds to the shunt device absorbing the active power required by the series device. The reactive power ratings differ because the series device regulates V_{PCCR} while the shunt device regulates the power factor at PCCS.

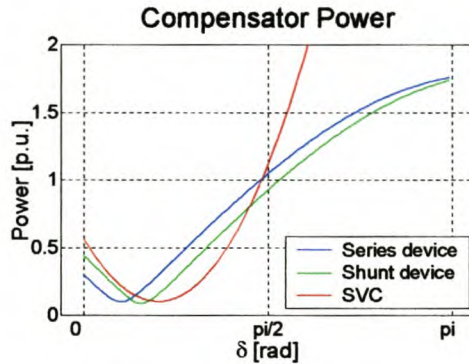


Fig. 4.25: Power required by the series-shunt device vs. power required by the SVC

The ratings of the apparent power required by the compensators of the series-shunt device and those of the SVC are compared in Fig. 4.25. Both the series and shunt device ratings are less than the rating of the SVC for $\delta < 0.436$ rad. For larger δ the shunt device requires a higher rating than the SVC. For $\delta > 0.560$ rad, the SVC has the lowest rating. However, the total rating of the series-shunt device is the sum of the series and shunt device ratings. The SVC therefore has a lower rating for all $\delta < 1.61$ rad. For higher δ the SVC rating strives towards infinity and the SVC-compensated network eventually ends in voltage collapse, as discussed in section 4.3.1.

4.6 IN-LINE COMPENSATION

Similar to the series compensator, the in-line compensator regulates both the voltage magnitude and angle at PCCR, as well as the power factor at PCCS. All required active power is sourced from the network at the sending end and transferred to the device at the receiving end via the dc bus. The need for an energy resource is therefore eliminated and power factor control is combined with voltage regulation in one compensator. The Zimbane network equipped with an in-line device is shown in Fig. 4.26.

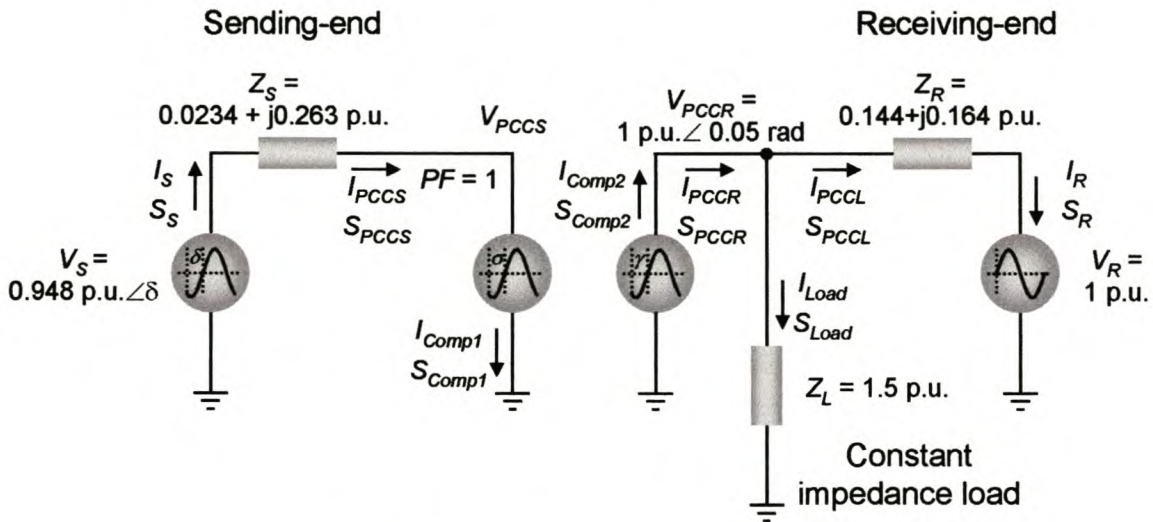


Fig. 4.26: Zimbane network with an in-line compensator

Both the required magnitude and angle of V_{PCCR} , as well as the power factor at PCCS must be specified. V_{PCCR} is specified as $1 \text{ p.u.} \angle 0.05 \text{ rad}$ and the power factor is unity. The dc bus voltage and resistance must also be specified. For a back-to-back converter these values are not important from a network analysis viewpoint. However, for HVDC lines the losses due to the dc line resistance are significant. For the Zimbane network a back-to-back converter is used. The dc bus voltage is specified as $1 \text{ p.u.}/0.78 = 1.28 \text{ p.u.}$ [B8] and the dc resistance is zero for the back-to-back converter. The voltage graphs of the above circuit are shown in Fig. 4.27.

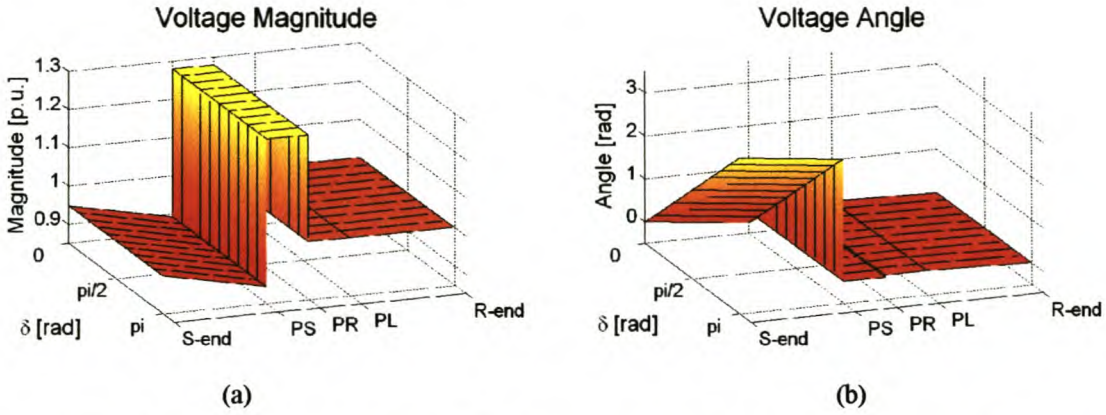


Fig. 4.27: Voltage graphs for the Zimbane network with an in-line compensator

With both the magnitude and angle of V_{PCCR} regulated to a constant value for all δ , the voltage magnitude graph stays constant for all δ . The voltage angle graph has an offset equal to δ at the sending end, but is constant at the receiving end due to the regulated V_{PCCR} . Fig. 4.28 shows the voltages across the network at $\delta=0$ rad. The voltage drop across Z_S is visible between the sending end and PCCS. Since all active compensating power is drawn from the sending end, the voltage drop across Z_S is large enough to pull $|V_{PCCS}|$ down to 0.893 p.u., which is far below the accepted 0.95 p.u. The compensator is situated between PCCS and PCCR. The voltage levels between these two points represent the voltage on the dc bus. The voltage angle is zero for the dc voltage and the dc voltage magnitude is 1.28, as specified. At PCCR the voltage is regulated to the specified 1 p.u. $\angle 0.05$ rad. There is no drop in voltage magnitude between V_{PCCR} and the receiving end as the voltage magnitude is unity at both locations. However, the drop in voltage angle permits power transfer across Z_R .

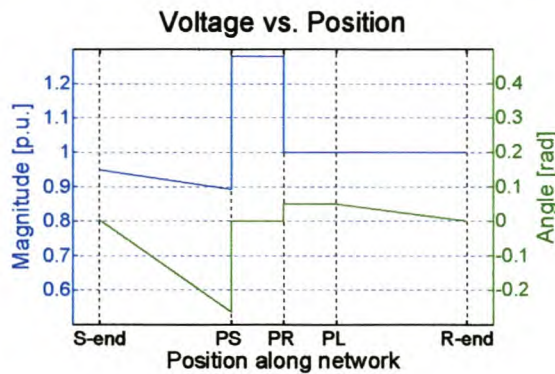


Fig. 4.28: Voltage graphs at $\delta=0$ rad for the Zimbane network with an in-line compensator

The power graphs for this scenario are depicted in Fig. 4.29. With the constant power factor at PCCS and the regulated V_{PCCR} for all δ , the power transfer across the network does not vary with δ . Fig. 4.29 is therefore valid for all δ .

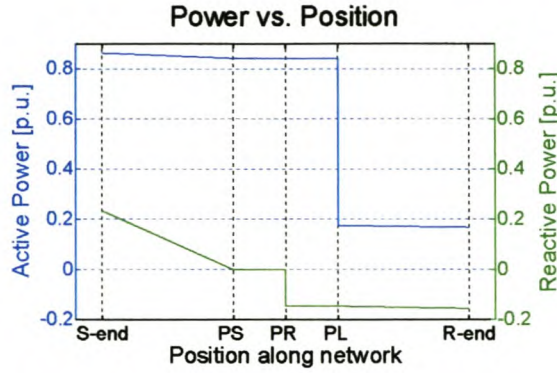


Fig. 4.29: Power transfer across the Zimbane network equipped with an in-line compensator

The power dissipated in Z_S is visible as a drop in active and reactive power between the sending end and PCCS. There is no step in reactive power at PCCS because the power factor is regulated to unity. No reactive power can be transferred across the dc bus. If the regulated power factor were not unity, reactive power would be transferred to the ac side of the sending-end device. With zero reactive power transfer across the dc bus, a step at PCCS would have indicated this change in reactive power from ac to dc.

The ratings of the devices at the sending and receiving ends of the in-line compensator are equal at 0.843 p.u. as equal power passes through them. The individual ratings are much higher than any other option considered. The total device rating of 1.686 p.u. therefore renders this device option not worth consideration.

4.7 SUMMARY OF THE ZIMBANE CASE STUDY

Summary of case study

The findings for the Zimbane case study are summarised in Table 4.1. All ratings are given at $\delta = 0$ rad to eliminate excessive ratings at high voltage angles.

Table 4.1: Summary of voltage regulator performance in the Zimbane network at $\delta = 0$ rad

Compensator	Control strategy	$ V_{PCCR} $ [p.u.]	Device rating	
			[MVA]	[MW]
None	None	0.94		
Shunt	Purely reactive	1	35	0
	Minimum rating	1	30	15
Series	Purely reactive	-	-	-
	Minimum rating	-	-	-
Series-shunt	No energy storage	1	81	18
In-line	No energy storage	1	155	155

From the viewpoint of the case study, there are two viable options for the Zimbane network, namely an SVC to perform voltage regulation at PCCR or the SVC combined with the hydro-generators. The standalone SVC rating is 35 MVA. With the hydro-generators, the SVC rating is reduced by 8 MVar (23%) at the expense of 15 MW hydro-power. The total compensator rating is therefore 30 MVA. True to the Eskom findings, the series devices resulted in severe over-voltages at PCCS. These devices are therefore not considered. Both the series-shunt and in-line devices draw their active compensating power from the sending-end network and therefore do not require an energy resource. For the Zimbane network, this increased power transfer across Z_S resulted in a voltage magnitude of less than 0.9 p.u. at PCCS. Besides the under-voltage problem, these compensators each require two devices, which result in very high total compensator ratings (81 MVA and 155 MVA, respectively).

Summary of PSAT

The results of this chapter show the abilities and benefits of PSAT more extensively. It is clear that the more complicated two-source mesh network produces results that are difficult to fully comprehend without the aid of the three-dimensional graphs.

The link between voltage regulation and first-swing stability was briefly discussed for the minimum-rating shunt compensator. The injection of active and reactive power extended the first-swing stability margin to more than $\pi/2$ rad, which emphasised the need for accurate analysis equations without approximations.

CHAPTER 5 FUTURE DEVELOPMENTS

FURTHER CONTROL STRATEGIES

**FURTHER DEVELOPMENTS IN THE REPRESENTATION OF
RADIAL LINES**

OTHER QUALITY OF SUPPLY PROBLEMS

PARALLEL DEVELOPMENTS

PSAT is a complete package for voltage regulation analysis. It is, however, purposeful to look sideways at parallel developments and forward to future developments in the same research project.

Possibilities for further control strategies are mentioned after which further developments in the presentation of voltage, current and power on radial lines are discussed. Other quality of supply problems and possible methods to analyse them are also presented. Finally a look at parallel developments is taken.

Further control strategies

Two other control strategies for the shunt and series compensators are briefly discussed in this section. For each of these strategies, a control equation can be developed to act as the third equation to solve P_{Comp} , Q_{Comp} and θ_{PCCR} , along with respective two equations developed for each of the shunt and series compensators.

The voltage angle at PCCR can be used to regulate the active power transfer across the network. The network planner specifies the required active power transfer if he chooses this option. Active and reactive power from the shunt or series compensator are then used to control $|V_{PCCR}|$ and the active power transfer across the network. The Muldersvlei SVC does not regulate the voltage magnitude, but regulates the voltage angle to control power transfer between the Western Cape region and the Highveld, where most of South Africa's power is generated. As the SVC only controls reactive power, only one degree of freedom exists and therefore only the voltage angle can be controlled.

Another possibility is to specify the apparent power rating of the device. The network planner will then be able to analyse the performance of available devices with a specific rating.

Further developments in the representation of radial lines

An alternative method to model compensators on radial lines is presented in this section. A shunt compensator is implemented at the receiving end of the Cathedral Peak network. Hereby the effect of the relative angle between the sending-end voltage source and the voltage angle of the compensator can be investigated. V_R is set to the required voltage magnitude at the PCC, namely 0.94 p.u. Z_R approaches zero and is therefore set to 1×10^{-6} p.u. The

resulting voltage graphs in Fig. 5.1 correspond to those in the rest of section 3.3, and serve as an indication that this implementation is valid.

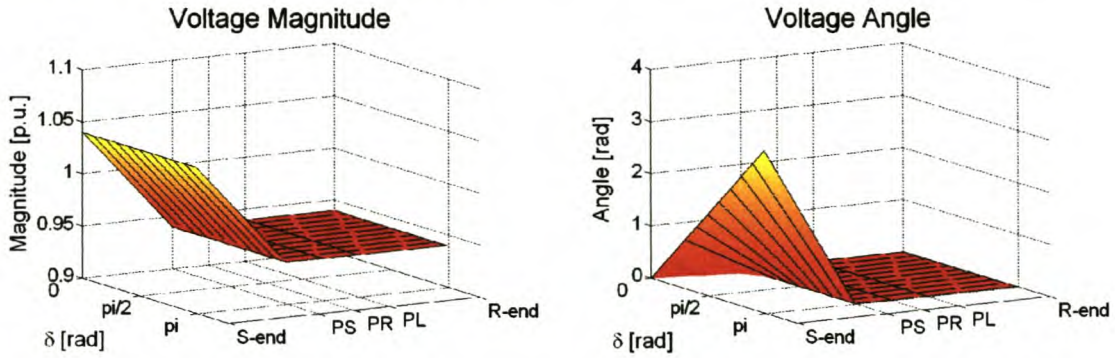


Fig. 5.1: Voltages across the network for a shunt compensator implemented at the receiving-end

The power transfer graphs in Fig. 5.2 shows the dependence of the compensator rating (now at the receiving end) on δ . δ is the voltage angle at the sending end, but also represents the relative voltage angle across the network as the receiving-end voltage angle is chosen as zero. With Z_R approaching zero, there is no significant voltage drop across or power dissipation in Z_R , as required.

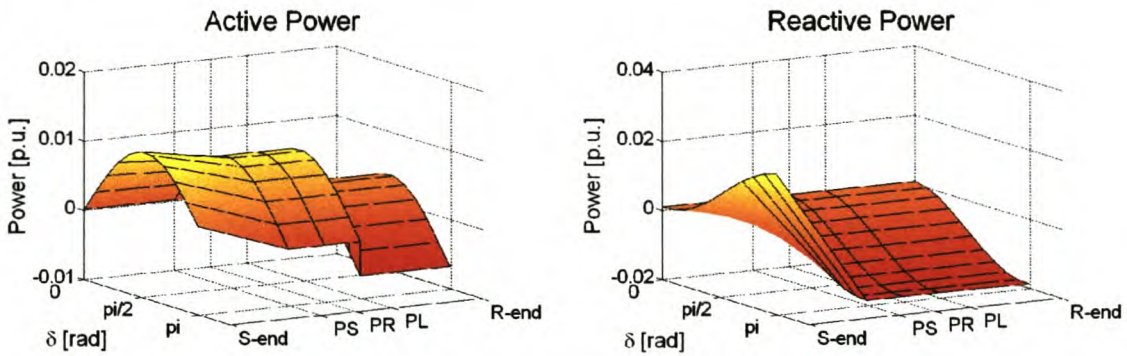


Fig. 5.2: Power transfer for a shunt compensator implemented at the receiving end

The active and reactive power, as well as the apparent power and power factor at the receiving end, are shown more clearly in Fig. 5.3. The operating points for the purely reactive compensator is at $\delta = 0.535$ rad and $\delta = 2.37$ rad, where the active power is zero. The operating point at $\delta = 0.535$ rad is chosen as it results in a smaller compensator rating. This angle is equal to the relative voltage angle across the purely active shunt-compensated network in section 3.3.1, which is the voltage angle across Z_S . To keep this relative voltage angle across the network constant, the shunt compensator has to regulate the voltage angle at PCCR to follow the sending-end voltage angle as it changes. This is evident in the rising voltage angle at PCCR in Fig. 3.15.

Similarly, for the purely active power compensator, the operating point is at $\delta = 0.339$ rad, where the reactive power is zero. This angle corresponds to the voltage angle across the network, and therefore across Z_S , in section 3.3.2.

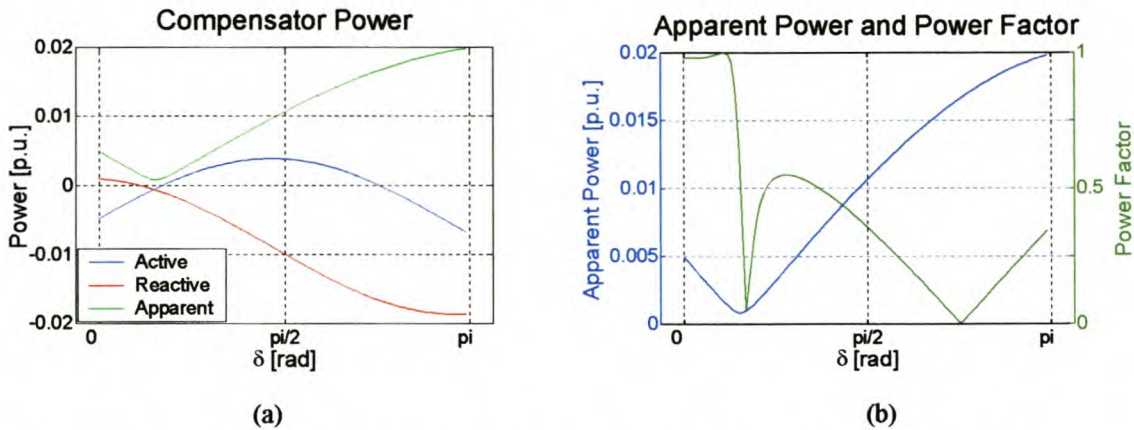


Fig. 5.3: Power and power factor for a shunt compensator implemented at the receiving end

From the power factor graph in Fig. 5.3 (b), it is evident that the minimum-rating shunt compensator requires a compensator power factor of ± 0.6 . This corresponds to the 0.56 power factor obtained in section 3.3.3. To adhere to the minimum power factor of the generator, namely 0.85, the compensator must operate at $\delta = 0.438$ rad, just before the point of minimum rating. This result in a shunt-compensator rating of 90.7 kVA and 76.9 kW, as obtained in section 3.3.3.

This method of portraying the compensator power as a function of the relative voltage angle across a radial network aids in the understanding of the compensator-network interaction. In future this method will be extended to series and other compensators. Mesh networks cannot be analysed in the same way as more than one source is present in the network itself.

Other quality of supply problems

PSAT will be extended in future to include the other quality of supply categories specified by NRS 048 [B4]0. In future work these quality of supply problems will all be modelled with the circuit in Fig. 2.3, which is repeated in Fig. 5.4 for easy reference.

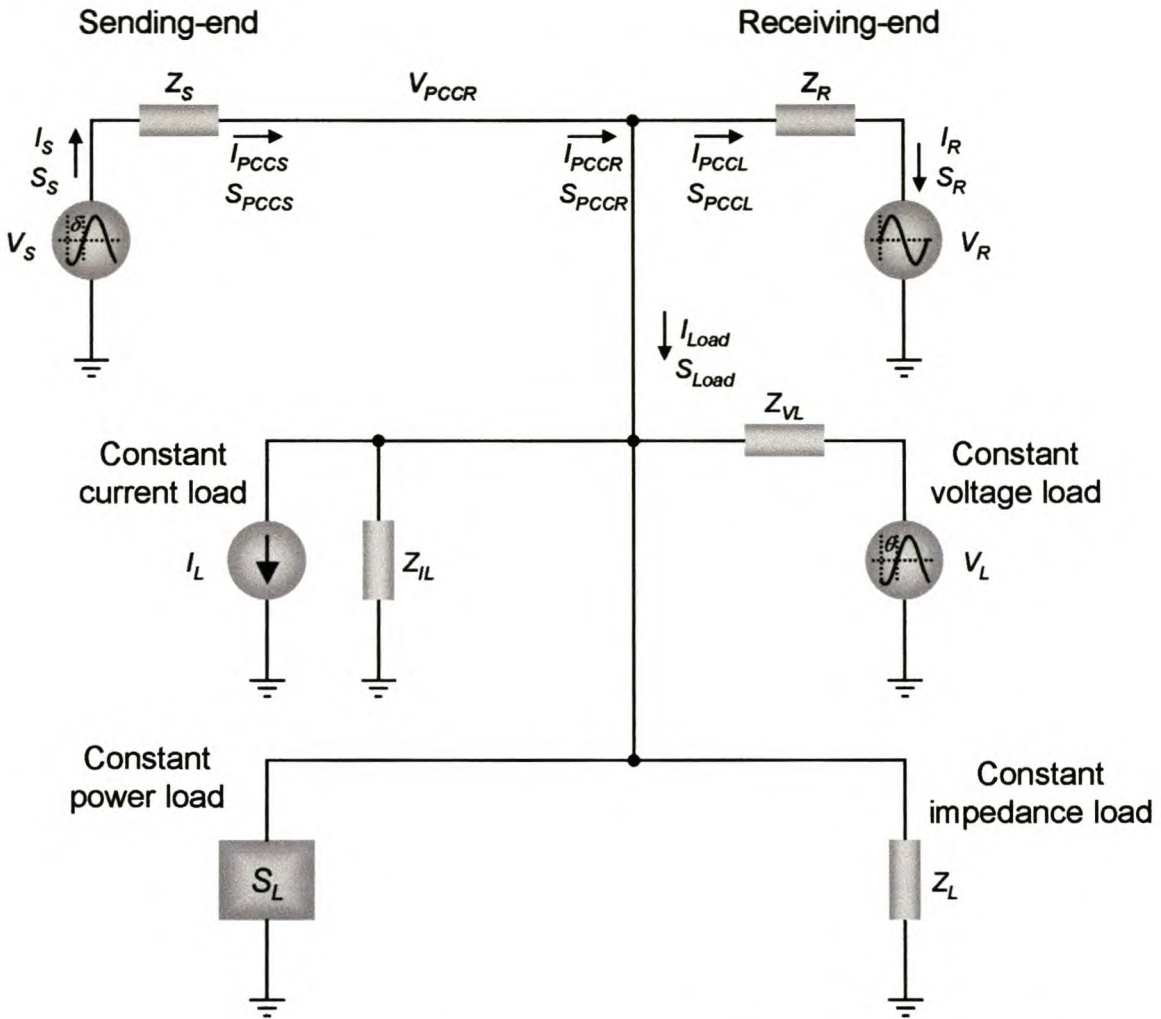


Fig. 5.4: Equivalent circuit diagram, without approximations, for radial networks

A synopsis of possible methods to model these quality of supply problems follows:

- Voltage dips from the network side are modelled by adjusting the magnitude and angle of V_S to reflect the dipped voltage. Dips from the customer side are modelled as a large load by lowering V_L , Z_{VL} , Z_{IL} and Z_L and increasing I_L and S_L to reflect the load.
- Voltage harmonics are dealt with by modelling each harmonic separately on a circuit equivalent to Fig. 5.4 (the reactances are adjusted to the relevant harmonic frequency). All the harmonic voltages and currents from the respective harmonic circuits are then superimposed to reflect the full extent of the harmonic problem. As in NRS 048, current harmonics are not dealt with separately, although this may be necessary in future, as discussed by Peng [A5].
- Flicker and voltage harmonics are essentially the same occurrence at different frequencies – flicker below the network frequency and harmonics above the network

frequency (in multiples of the network frequency). Flicker is therefore dealt with in a similar way as voltage harmonics.

- Currently only three-phase, balanced circuits are considered, but voltage unbalance can be modelled by constructing an equivalent circuit for the positive, negative and zero-phase sequences.
- Voltage interruptions are modelled by setting V_S or V_R to zero. Alternatively, Z_S or Z_R can be set to infinity to model a break in transmission due to an open circuit breaker or physical line failure.

During a quality of supply evaluation conducted in November 1999 at Palabora Mining Company, voltage regulation, voltage dips, unbalance and voltage harmonics were measured. All measured values were within the NRS 048 specifications for the respective quality of supply categories, except the voltage harmonics in Refinery substation, as shown in Fig. 5.5.

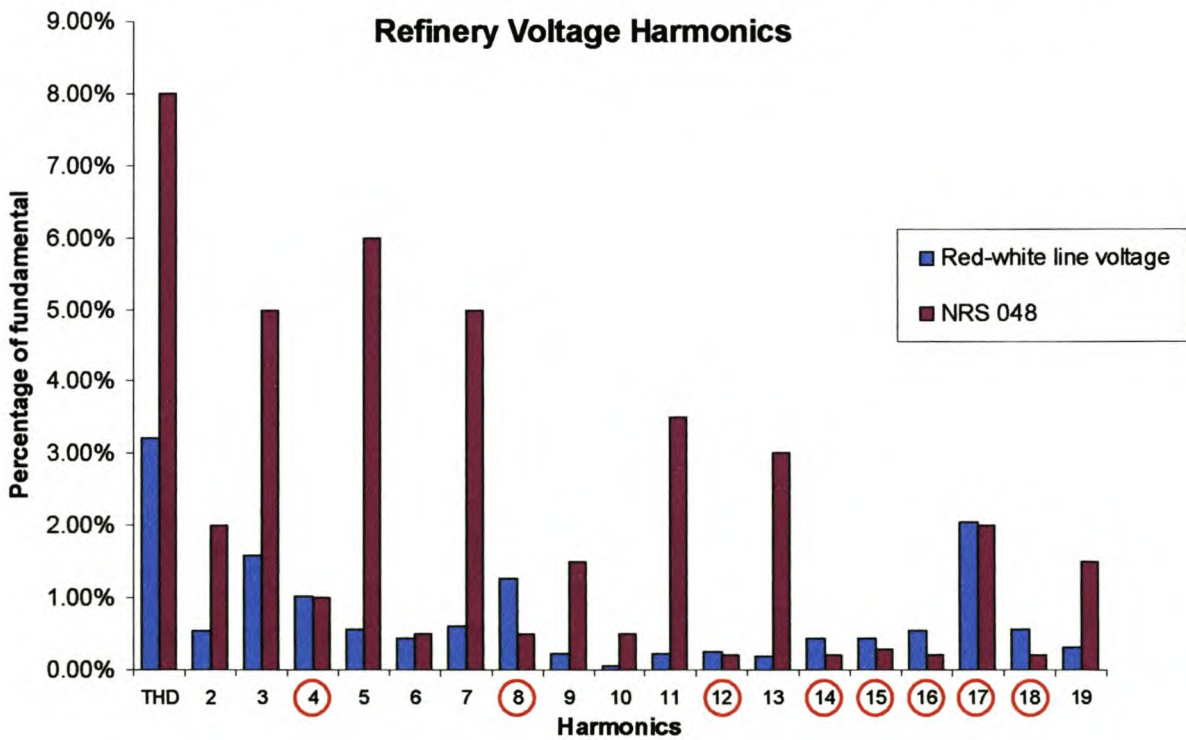


Fig. 5.5: Voltage harmonics at Refinery substation

The refinery is the final step in the copper purification process. Electrolysis is used to produce 99.8% pure copper. Thyristor rectifiers produce the necessary dc voltage for the electrolysis process. Due to the old control cards (circa late 1960s) the phase currents are not balanced. The voltage drop across the impedance of the line feeding the refinery results in the excessive even harmonics in Fig. 5.5. The fifteenth and seventeenth harmonics are

characteristic of the large induction machines used in the casting plant, which are fed from Refinery substation.

With all the harmonic voltages and currents known for the incomer at Refinery substation, as well as the voltage harmonics at PMC substation (from where Refinery substation is fed), a separate circuit similar to Fig. 5.4 is drawn for every harmonic. In this case a circuit is constructed for the following harmonics: 4, 8, 12, 14, 15, 16, 17 and 18. The other harmonics fall within the NRS 048 specifications and need not be modelled. The impedance and harmonic source values can be calculated with Ohm's law from the measured harmonic voltages and currents.

Re-investment of money saved on capital expenditure benefits future upgrades of a lower risk. A typical consequence of line harmonics that can be investigated with this tool, is the current stress voltage harmonics place on shunt capacitors. This is relevant for the large power factor correction capacitor banks installed at PMC substation.

Parallel developments

Mr Gert Fourie is working on a voltage stability analysis tool that will be incorporated into the voltage-regulation analysis tool presented in this thesis. This will ultimately enable the network planner to analyse both the voltage regulation and stability across a network. The effect of voltage-regulating network technologies on network stability can be analysed in the same analysis tool to distinguish even further between the available solutions for a specific network problem.

Mr Bernard Meyer is investigating the economic side of the various network technologies. His work will also be included in PSAT to enable the network planner to quickly judge the economic impact and results of the various solutions to a network problem.

Mr Frank Engelbrecht is investigating transmission line models and parameters. He is writing an extension to the input interface of PSAT that will calculate the equivalent impedances of a transmission line automatically. He is also investigating methods to share data between Reticmaster and PSAT. Using the methods developed in section 3.2.2, network parameters will be sourced directly from Reticmaster.

CHAPTER 6 CONCLUSION

Recent advances in semiconductor device development pushed a large number of network devices onto the market. These devices can solve network problems more effectively and economically than ever before. Network planners need tools to analyse and implement such devices. They also need to be trained in such analysis and implementation. This thesis aims to document the development of such tool, namely PSAT.

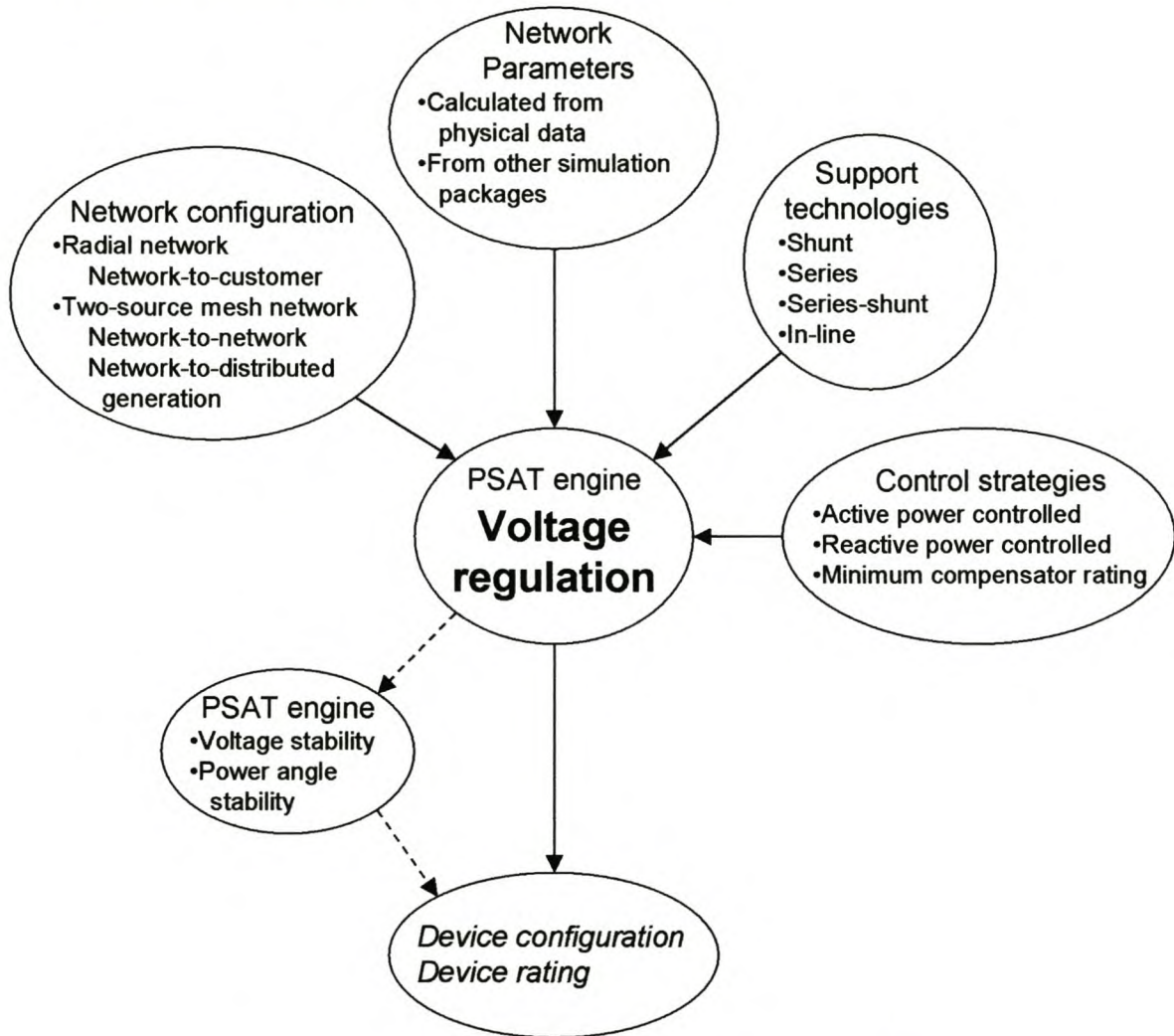


Fig. 6.1: Inputs and outputs of PSAT voltage-regulation engine

A diagram of the technical inputs and outputs of the PSAT voltage-regulation engine is shown in Fig. 6.1. The equations of the voltage regulation engine must consider the following inputs:

- Network configuration
- Network parameters
- Support technologies
- Control strategies.

The network configuration was discussed in Chapter 1 and Chapter 2. Rural networks experience the majority of voltage-regulation problems in South Africa. The networks are long sub-transmission and reticulation networks and modelled by two generic networks, namely a radial network and a two-source mesh network.

Most rural networks are radial, i.e. network-to-customer networks. Where a load is fed from two sources, whether it is a network-to-network or a network-to-distributed generation system, it can be modelled as a mesh network with two voltage sources.

The equations for voltage regulation were developed for a two-source mesh network, but since a radial network is a special case of this network, they can be applied for both generic network types.

As discussed in Chapter 1 and Chapter 2, the network parameters can be calculated from practical measurements or from simulation data of other software packages, e.g. Reticmaster. Section 3.2.2 discussed methods and equations to develop an equivalent circuit from the network data obtained.

Having constructed the equivalent network model, a wide range of support technologies and control strategies must be evaluated to find the optimum solution for the voltage regulation problem. Energy-storage devices can be incorporated with a shunt or series device to enable both active and reactive power compensation. A control strategy is needed to regulate this extra degree of freedom. The network planner can choose to regulate the active power injected, the reactive power injected or to control both to obtain a minimum-rating device that performs the required voltage regulation.

A generic network was developed in Chapter 3 on which all of the above scenarios can be analysed. Equations were developed to describe the voltage regulation for an uncompensated network as well as a network compensated by any combination of support technology and control strategy. PSAT is set apart from other power system simulation packages by not making any approximations in the development of the equations. Hereby lossy networks, which are typical for rural systems, can be accurately analysed. Support technologies open up the possibility to operate networks at voltage angles greater than the traditional 0.7 rad. With the no-approximations approach, PSAT is able to analyse network operation up to a voltage

angle of π rad. For each specific combination of these inputs, the best solution in terms of device configuration and rating is determined with the aid of PSAT. The equations developed for voltage regulation will be adapted to analyse voltage and power angle stability in future.

The Cathedral Peak case study was presented as an integral part of the equation development in Chapter 3. This showed the practical implementation of PSAT and introduced the voltage, current and power graphs that described the conditions across each network. Cathedral Peak is a radial network-to-customer network and results in less complicated graphs due to there being only one voltage source in the system.

The Zimbane case study in Chapter 4 is an example of the second generic network, namely a mesh network with two voltage sources. The analysis results of this network are more complicated and are a further indication of the benefits of PSAT. The three-dimensional graphs of voltage, current and power showed the network conditions for all voltage angles across the network at any physical location. These graphs aid the network planner in understanding the influence of every support technology and control method on the network. Dedicated graphs for the device voltage, current and power enable the planner to select the best solution to a network problem in terms of device configuration and rating.

PSAT provides a sturdy platform on which the future developments discussed in Chapter 5 can be built. However, PSAT can already function as an analysis aid to solve the network problem encountered most often in South Africa: voltage regulation.

CHAPTER 7 REFERENCES

Journals and conference papers

- [A1] N.G. Hingorani, *Flexible AC Transmission*, IEEE Spectrum, April 1993, pp. 40-45.
- [A2] N.G. Hingorani, *Introducing Custom Power*, IEEE Spectrum, June 1995, pp. 41-48.
- [A3] N.G. Hingorani, R.G. Stephen, N. Sudja, C. Nietsch and M. Weinhold, *Use of Power Electronics for Rural Electrification*, Cigré, Paris, 1998.
- [A4] F.J. Rossouw, H.J. Beukes and J.H.R. Enslin, *Modelling Network Problems with Power Flow Equations and Diagrams*, IEEE AFRICON, Cape Town, 1999.
- [A5] F.Z. Peng, *Application Issues of Active Power Filters*, IEEE Industry Applications Magazine, pp. 21-29, September/October 1998.
- [A6] M.E. Baran and M. Hsu, *Volt/Var Control at Distribution Substations*, IEEE Trans. on Power Systems, Vol. 14, No. 1, February 1999.
- [A7] L. Gyugi, *Dynamic Compensation of AC Transmission Lines by Solid-State Synchronous Voltage Sources*, IEEE/PES 1993 Summer Meeting, Vancouver, 1993.
- [A8] A.E. Hammad and W.F. Long, *Performance and Economic Comparisons between Point-to-Point HVDC Transmission and Hybrid Back-to-Back HVDC/AC Transmission*, IEEE/PES 1989 Transmission and Distribution Conference, New Orleans, 1989.
- [A9] B. Singh, K. Al-Haddad and A. Chandra, *A Review of Active Filters for Power Quality Improvement*, IEEE Transactions on Industrial Electronics, Vol. 45, No. 5, pp 960-971, October 1999.
- [A10] S.G. Helbing and G.G. Karaday, *Investigations of an Advanced Form of Series Compensation*, IEEE/PES 1993 Summer Meeting, Vancouver, 1993.
- [A11] L. Gyugi, *Unified Power-flow Control Concept for Flexible AC Transmission Systems*, IEE Proceedings-C, Vol. 139, No.4, 1992.
- [A12] H. Fujita and H. Akagi, *The Unified Power Quality Conditioner: The Integration of Series- and Shunt-Active Filters*, IEEE Transactions on Power Electronics, Vol. 13, No. 2, March 1998.
- [A13] J.R. de Silva, C.P. Arnold and J. Arrillaga, *Capability Chart for an HVDC link*, IEE Proceedings, Vol. 134, Part C, No. 3, May 1987.
- [A14] J.J. Paserba, N.W. Miller, E.V. Larsen, and R.J. Piwko, *A Thyristor Controlled Series Compensation Model for Power System Stability Analysis*, IEEE/PES 1994 Summer Meeting, San Francisco, July 1994.

- [A15] J. M. Hawkins and T. Robbins, *Vanadium Energy Storage Concepts for Telecommunications Applications*, IEEE Intelec99, 1999.
- [A16] C. Mostert, S. Thiel, J.H.R. Enslin, R. Herman and R. Stephen, *Investigating the different combinations of μ FACTS devices in low cost rural electrification*, CIGRÉ SC-14 International Colloquium on HVDC and FACTS, Johannesburg, South Africa, 28 - 30 September 1997.
- [A17] C. Billingham, R. McLaren, J. Boshoff and S. Boshoff, *Relocation of Facts Devices – The Zimbane Experience*, 1998 Cigré Power Applications Symposium, Midrand, 1998.

Books and manuals

- [B1] *Facts Overview*, IEEE Inc., New York, 1995.
- [B2] N. G. Hingorani and L. Gyugi, *Understanding FACTS*, IEEE Press, New York, 2000.
- [B3] T.J.E. Miller, *Reactive Power Control in Electric Systems*, John Wiley & Sons, New York, 1982.
- [B4] NRS 048, *Electricity Supply - Quality of Supply*, South African Bureau of Standards, Pretoria, 1996.
- [B5] A.E. Fitzgerald, C. Kingsley and S.D. Umans, *Electric Machinery*, McGraw-Hill Inc., London, 1992.
- [B6] J.D. Glover and M. Sarma, *Power System Analysis and Design*, PWS Publishing Company, Boston, 1994.
- [B7] *Building GUIs with Matlab*, The Mathworks Inc., Natick M.A, 1997.
- [B8] N. Mohan, T.M. Undeland and W.P. Robbins, *Power Electronics*, John Wiley & Sons, Inc., New York, 1995.

Other references

- [C1] Conversation with B. Meyer and HJ Beukes, Stellenbosch, October 2000.
- [C2] J.H.R. Enslin, H. du T. Mouton, D.D. Bester, A.D. le Roux and A.J. Visser, *Control and Implementation Considerations of a 2 MVA Power Quality Compensator*, Eskom SAPSSI Research Report, Stellenbosch, October 1999.
- [C3] *UPS SubstationTM: Evaluation, Conceptual Design and Generic Specification*, EPRI, Palo Alto, CA: 1999. TR-111091.
- [C4] C. Carter-Brown, *IEP Planning Proposal: Cathedral Peak Pilot*, Internal Eskom Report, South Africa, 1998.

- [C5] H.J. Beukes, *Network Options and Integration: Volume 1*, Eskom SAPSSI Research Report SAPSSI/98/026, Stellenbosch, 1998.
- [C6] Telephone conversation with R. Jung, Eskom branch in East London, 10 October 2000.
- [C7] Telephone conversation with N. Meyer, Eskom branch in East London, 11 October 2000.

ADDENDUM A CALCULATION OF THE IMPEDANCE OF 25 KM HARE LINE

From the standard Aluminium Conductor Steel Reinforced (ACSR) property tables, the following data of Hare conductor can be obtained:

- Number of strands: 7
- Strand diameter: 4.72 mm
- Resistance: 0.2733 Ω /km

From Glover & Sarma [B6], the line inductance per meter for a three-phase three-wire line with equal phase spacing is given by (A.1), where D is the distance between conductors and D_S is the geometric mean radius (GMR) of the multi-stranded hare conductor.

$$L = 2 \times 10^{-7} \ln \left(\frac{D}{D_S} \right) \text{ [H/m]} \quad (\text{A.1})$$

The line structure is typically a T-structure with $D = 1.8$ m between the conductors, as shown in Fig. A.1.

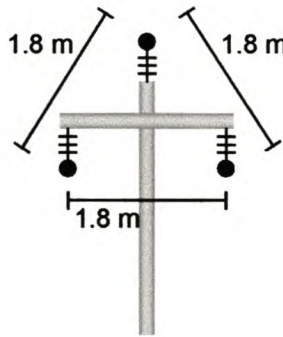


Fig. A.1: Pole structure of a typical 33 kV Hare line

D_S is given by (A.2) for the 7-stranded Hare conductor. D_{km} is the distance between the centres of strands k and m and r is the strand radius, which is $4.72 / 2 = 2.36$ mm.

$$\begin{aligned} D_S &= \sqrt[7]{\prod_{k=1}^7 \prod_{m=1}^7 D_{km}} & (\text{A.2}) \\ &= \sqrt[49]{\left\{ 0.7788r(2r)^3(\sqrt{12}r)^2(4r) \right\}^6 \left\{ 0.7788r(2r)^6 \right\}} \\ &= 2.177r \\ &= 5.14 \text{ mm} \end{aligned}$$

Substituting D and D_S in (A.1), the inductance per meter is calculated as 1.17 μ H/m. The total impedance for the 25 km Hare line is therefore given by Z_R .

ADDENDUM A CALCULATION OF THE IMPEDANCE OF 25 KM HARE LINE

$$\begin{aligned}Z_R &= 25 \times (0.2733 + j1.17 \times 10^{-6} \times 10^3 \times 2 \times \Pi \times 50) \\&= 6.83 + j9.19\Omega \\&= 0.628 + j0.844 \text{ p.u.}\end{aligned}$$

The per unit value for Z_R (for 33 kV) is calculated with $Z_{base} = \frac{V_{base}^2}{S_{base}} = \frac{(33 \times 10^3)^2}{100 \times 10^6} = 10.89\Omega$.

$$\text{For 11 kV, } Z_{base} = \frac{V_{base}^2}{S_{base}} = \frac{(11 \times 10^3)^2}{100 \times 10^6} = 1.21\Omega$$

ADDENDUM B PROGRAM LISTINGS

B.1 MAPLE CODE

B.1.1 SolPQ.mws

B.1.2 SolTo4thOrder.mws

B.2 PSAT CODE

B.2.1 Voltreg.m

B.2.2 CalcVpcc.m

B.2.3 SolvePQ.m

B.2.4 Solve4thOrder.m

B.2.5 No.m

B.2.6 NoGui.m

B.2.7 VoltregGui.m

B.2.8 Graph2dAng.m

B.2.9 Graph2dpos.m

B.2.10 Graph2dComp.m

B.2.11 Graph2dLoad.m

B.2.12 Graph2dVector.m

B.2.13 Graph3d.m

B.1 MAPLE CODE

B.1.1 SolPQ.mws

```

1  Declarations
2  restart;
3  assume(thetaVpcc,realcons);assume(Vpcc,realcons);assume(Ppccs,realcons);assume(Qpccs,realcons);
4  assume(ca,realcons);assume(cb,realcons);assume(cc,realcons);assume(cd,realcons);assume(ce,realcons);assume(cf,realcons);
5  Algebra
6  > Peq:=Vpcc*ca*cos(thetaVpcc)-Vpcc*cb*sin(thetaVpcc)+Vpcc^2*cc-Ppccs;
7
8  Peq = Vpcc~ ca~ cos(thetaVpcc~) - Vpcc~ cb~ sin(thetaVpcc~) + Vpcc~^2 cc~ - Ppccs~
9
10 > Qeq:=Vpcc*ca*sin(thetaVpcc)+Vpcc*cb*cos(thetaVpcc)+Vpcc^2*cd-Qpccs;
11
12 Qeq = Vpcc~ ca~ sin(thetaVpcc~) + Vpcc~ cb~ cos(thetaVpcc~) + Vpcc~^2 cd~ - Qpccs~
13
14 > sol:=solve({Peq,Qeq},{thetaVpcc,Vpcc}); % solve thetaVpcc and Vpcc from equations Peq and Qeq
15
16 sol = {Vpcc~ = RootOf((cc~^2 + cd~^2) _Z^4 + (-2 Qpccs~ cd~ - cb~^2 - 2 cc~ Ppccs~ - ca~^2) _Z^2 + Qpccs~^2 + Ppccs~^2),
17 thetaVpcc~ = arctan( (-cb~ RootOf((cc~^2 + cd~^2) _Z^4 + (-2 Qpccs~ cd~ - cb~^2 - 2 cc~ Ppccs~ - ca~^2) _Z^2 + Qpccs~^2 + Ppccs~^2) cc~
18 + cb~ Ppccs~ + RootOf((cc~^2 + cd~^2) _Z^4 + (-2 Qpccs~ cd~ - cb~^2 - 2 cc~ Ppccs~ - ca~^2) _Z^2 + Qpccs~^2 + Ppccs~^2) cd~
19 ca~ - Qpccs~ ca~) / RootOf((cc~^2 + cd~^2) _Z^4 + (-2 Qpccs~ cd~ - cb~^2 - 2 cc~ Ppccs~ - ca~^2) _Z^2 + Qpccs~^2 + Ppccs~^2), - (
20 cb~ RootOf((cc~^2 + cd~^2) _Z^4 + (-2 Qpccs~ cd~ - cb~^2 - 2 cc~ Ppccs~ - ca~^2) _Z^2 + Qpccs~^2 + Ppccs~^2) cd~
21 - cb~ Qpccs~ + RootOf((cc~^2 + cd~^2) _Z^4 + (-2 Qpccs~ cd~ - cb~^2 - 2 cc~ Ppccs~ - ca~^2) _Z^2 + Qpccs~^2 + Ppccs~^2) cc~
22 ca~ - Ppccs~ ca~) / RootOf((cc~^2 + cd~^2) _Z^4 + (-2 Qpccs~ cd~ - cb~^2 - 2 cc~ Ppccs~ - ca~^2) _Z^2 + Qpccs~^2 + Ppccs~^2) )};
23
24 > subs(RootOf((cc^2+cd^2)_Z^4+(-2*Qpccs*cd-ca^2-2*cc*Ppccs-cb^2)_Z^2+Qpccs^2+Ppccs^2)=RR,sol); % simplify solution by substituting constants for RR
25
26 {Vpcc~ = RR, thetaVpcc~ = arctan( (-cb~ RR^2 cc~ + cb~ Ppccs~ + RR^2 cd~ ca~ - Qpccs~ ca~,
27 -cb~ RR^2 cd~ - cb~ Qpccs~ + RR^2 cc~ ca~ - Ppccs~ ca~) / RR)};
28
29 > thetaVpcc:=arctan(cb*RR^2*cc-cb*Ppccs-RR^2*cd*ca+Qpccs*ca,-cb*RR^2*cd+cb*Qpccs-RR^2*cc*ca+Ppccs*ca); % thetaVpcc solution 1 in terms of RR
30
31 thetaVpcc = arctan(cb RR^2 cc - cb Ppccs - RR^2 cd ca + Qpccs ca, -cb RR^2 cd + cb Qpccs - RR^2 cc ca + Ppccs ca)
32
33 > thetaVpcc:=arctan(RR^2*(cb*cc-cd*ca)-cb*Ppccs+Qpccs*ca,-RR^2*(cb*cd+cc*ca)+cb*Qpccs+Ppccs*ca); % thetaVpcc solution in terms of RR
34
35 thetaVpcc = arctan(RR^2 (cb cc - cd ca) - cb Ppccs + Qpccs ca, -RR^2 (cb cd + cc ca) + cb Qpccs + Ppccs ca)
36
37 > RR:=RootOf((cc^2+cd^2)_Z^4+(-2*Qpccs*cd-ca^2-2*cc*Ppccs-cb^2)_Z^2+Qpccs^2+Ppccs^2); % calculate RR by substituting constants
38
39 RR = RootOf((cc~^2 + cd~^2) _Z^4 + (-2 Qpccs~ cd~ - cb~^2 - 2 cc~ Ppccs~ - ca~^2) _Z^2 + Qpccs~^2 + Ppccs~^2)
40
41 > subs({(cc^2+cd^2)=C1,(-2*Qpccs*cd-ca^2-2*cc*Ppccs-cb^2)=C3,Qpccs^2+Ppccs^2=C5},RR);
42
43 RootOf(C1 _Z^4 + C3 _Z^2 + Qpccs~^2 + Ppccs~^2)
44
45 > solall:=allvalues(RootOf(C1*_Z^4+C3*_Z^2+C5)); % solve RR in terms of constants

```

ADDENDUM B PROGRAM LISTINGS

```

30  solall =  $\frac{1}{2} \frac{\sqrt{2} \sqrt{C1(-C3 + \sqrt{C3^2 - 4 C1 C5})}}{C1}$ ,  $-\frac{1}{2} \frac{\sqrt{2} \sqrt{C1(-C3 + \sqrt{C3^2 - 4 C1 C5})}}{C1}$ ,  $\frac{1}{2} \frac{\sqrt{-2 C1(C3 + \sqrt{C3^2 - 4 C1 C5})}}{C1}$ ,
31   $-\frac{1}{2} \frac{\sqrt{-2 C1(C3 + \sqrt{C3^2 - 4 C1 C5})}}{C1}$ 
32  > RR1:=combine(1/2*sqrt(2)*sqrt(C1*(-C3+sqrt(C3^2-4*C1*C5)))/C1); % simplify first solution of RR
33
34  > RR2:=combine(-1/2*sqrt(2)*sqrt(C1*(-C3+sqrt(C3^2-4*C1*C5)))/C1); % simplify second solution of RR
35
36  > RR3:=combine(1/2*sqrt(-2*C1*(C3+sqrt(C3^2-4*C1*C5)))/C1); % simplify third solution of RR
37
38  > RR4:=combine(-1/2*sqrt(-2*C1*(C3+sqrt(C3^2-4*C1*C5)))/C1); % simplify lastt solution of RR
39
40
41  >
42  >
43  > Vpcc1:=RR1; % four solutions to RR give four solutions to Vpcc
44  > Vpcc2:=RR2;
45  > Vpcc3:=RR3;
46  > Vpcc4:=RR4;
47  > %formulae for solving thetaVpcc in matlab (SolPQ.m)
48  > thetaVpcc1:=arctan(-(cb*RR1^2*cc+cb*Ppccs+RR1^2*cd*ca-Qpccs*ca)/RR1,-(cb*RR1^2*cd-cb*Qpccs+RR1^2*cc*ca-Ppccs*ca)/RR1);
49  > thetaVpcc2:=arctan(-(cb*RR2^2*cc+cb*Ppccs+RR2^2*cd*ca-Qpccs*ca)/RR2,-(cb*RR2^2*cd-cb*Qpccs+RR2^2*cc*ca-Ppccs*ca)/RR2);
50  > thetaVpcc3:=arctan(-(cb*RR3^2*cc+cb*Ppccs+RR3^2*cd*ca-Qpccs*ca)/RR3,-(cb*RR3^2*cd-cb*Qpccs+RR3^2*cc*ca-Ppccs*ca)/RR3);
51  > thetaVpcc4:=arctan(-(cb*RR4^2*cc+cb*Ppccs+RR4^2*cd*ca-Qpccs*ca)/RR4,-(cb*RR4^2*cd-cb*Qpccs+RR4^2*cc*ca-Ppccs*ca)/RR4);
52  >
53  > C1=cc^2+cd^2; % formulae of constants used above
54  C2=0;
55  C3=-2*Qpccs*cd-ca^2-2*cc*Ppccs-cb^2;
56  C4=0;
57  C5=Qpccs^2+Ppccs^2;

```

B.1.2 SolTo4thOrder.mws

This file is not shown here due to its excessive length (77 pages). The method followed in the file is quite simple, but the long file length is caused by the very long formulae that solves the fourth-order equations. The method is discussed below and the results of *SolTo4thOrder.mws* are implemented in *Solto4thOrder.m*.

1. The equation describing θ_{PCCR} and V_{PCCR} is solved for θ_{PCCR} . Maple gives the results in the form of a fourth-order polynomial. The roots of the polynomial are the answers to θ_{PCCR} .
2. The above polynomial is simplified by grouping constants and substituting each of these groups for a single constant. The resulting polynomial is given in Maple notation in terms of $_Z$:

$$C1*_Z^4+C2*_Z^3+C3*_Z^2+C4*_Z+C5 = 0 \quad (B.1)$$

3. (B.1) is then solved explicitly by Maple. This is where the long formulae originates.
4. The four solutions to $_Z$ is simplified by substituting groups of constants with a single constant. This substitution is repeated until the following formulae are obtained as solutions to θ_{PCCR} . The formulae are in Maple format and all parameters in the equations are known constants for a specific network situation.

$$\text{thetaVpccr(1)}:=\arctan(-(\text{sol1}^2*\text{K1}-\text{sol1}^2*\text{K2}+\text{sol1}*\text{K3}+\text{K2}+\text{K7})/(\text{sol1}*\text{K5}+\text{K4}),\text{sol1}): \quad (B.2)$$

$$\text{thetaVpccr(2)}:=\arctan(-(\text{sol2}^2*\text{K1}-\text{sol2}^2*\text{K2}+\text{sol2}*\text{K3}+\text{K2}+\text{K7})/(\text{sol2}*\text{K5}+\text{K4}),\text{sol2}): \quad (B.3)$$

$$\text{thetaVpccr(3)}:=\arctan(-(\text{sol3}^2*\text{K1}-\text{sol3}^2*\text{K2}+\text{sol3}*\text{K3}+\text{K2}+\text{K7})/(\text{sol3}*\text{K5}+\text{K4}),\text{sol3}): \quad (B.4)$$

$$\text{thetaVpccr(4)}:=\arctan(-(\text{sol4}^2*\text{K1}-\text{sol4}^2*\text{K2}+\text{sol4}*\text{K3}+\text{K2}+\text{K7})/(\text{sol4}*\text{K5}+\text{K4}),\text{sol4}): \quad (B.5)$$

5. These equations, along with the formulae for the constants, are implemented in *Solto4thOrder.m*, which is listed in Addendum B.2.4.

B.2 PSAT CODE

B.2.1 Voltreg.m

```

1  function Data = VoltReg(DeviceType, Technology, Screen, delta1, P, N)                                % Voltage regulation for network to customer
2
3  Vs = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','VsEdit'),'String'));                % get value of Vs parameter from GUI
4  Zs = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','ZsEdit'),'String'));                % get value of Zs parameter from GUI
5  Zr = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','ZrEdit'),'String'));                % get value of Zr parameter from GUI
6  Vr = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','VrEdit'),'String'));                % get value of Vr parameter from GUI
7  if or(isempty(Zr),isempty(Vr))                                                                    % set default values for Vr and Zr
8      Zr = inf;
9      Vr = 0;
10 end;
11 Iload = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','IloadEdit'),'String'));          % get value of Iload parameter from GUI
12 IZload = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','IZloadEdit'),'String'));        % get value of IZload parameter from GUI
13 if or(isempty(Iload),isempty(IZload))                                                            % set default values for Iload and IZload
14     Iload = 0;
15     IZload = inf;
16 end;
17 Vload = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','VloadEdit'),'String'));          % get value of Vload parameter from GUI
18 VZload = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','VZloadEdit'),'String'));        % get value of VZload parameter from GUI
19 if or(isempty(Vload),isempty(VZload))                                                            % set default values for Vload and VZload
20     Vload = 0;
21     VZload = inf;
22 end;
23 Zload = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','ZloadEdit'),'String'));          % get value of Zload parameter from GUI
24 if isempty(Zload)                                                                                % set default value for Zload
25     Zload = inf;
26 end;
27 Pload = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','PloadEdit'),'String'));          % get value of Pload parameter from GUI
28 if isempty(Pload)                                                                                % set default values for Pload
29     Pload = 0;
30 end;
31 Qload = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','QloadEdit'),'String'));          % get value of Qload parameter from GUI
32 if isempty(Qload)                                                                                % set default values for Qload
33     Qload = 0;
34 end;
35 tol = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','ToleranceEdit'),'String'));        % get tolerance value from VReg GUI
36
37 delta = [0;(P*pi/N);P*pi-(P*pi/N)];                                                            % delta = voltage angle across network
38 delta(1) = pi/N/10;                                                                              % delta(1) <> 0 to prevent div 0 errors
39 Vs = Vs .* exp(j*delta);                                                                        % set Vs angle
40 ZrThev = inv(1/Zr+1/VZload+1/Zload+1/IZload);                                                  % Thevenin equivalent impedance for receiving-end and load
41 VrThev = (Vr/Zr+Vload/VZload-Iload)*ZrThev;                                                    % Thevenin equivalent voltage for receiving-end and load
42 if isnan(VrThev)                                                                                  % test if VrThev valid
43     VrThev = zeros(1,N);                                                                        % set invalid VrThev to zero
44 else
45     VrThev = repmat(VrThev,1,N);                                                                % create 1xN array from valid VrThev
46 end;
47 Pload = repmat(Pload,1,N);                                                                      % create 1xN Pload array
48 Qload = repmat(Qload,1,N);                                                                      % create 1xN Qload array
49 Sload = Pload+j*Qload;
50
51 % ----- CALCULATE Vpccr UNDER NO COMPENSATION
52
53 a = real(conj(Vs/Zs));                                                                            % calculate constants
54 b = imag(conj(Vs/Zs));
55 c = real(-1/conj(Zs));
56 d = imag(-1/conj(Zs));
57 e = real(-conj(VrThev/ZrThev));
58 f = imag(-conj(VrThev/ZrThev));
59 g = real(1/conj(ZrThev));
60 h = imag(1/conj(ZrThev));
61 K3 = (a-e).*ones(1,N);

```

```

62 K4 = (b-f).*ones(1,N);
63 K6 = (c-g).*ones(1,N);
64 K8 = (d-h).*ones(1,N);
65 Vpcert = SolvePQ(K3,K4,K6,K8,Pload,Qload); % solve Vpcert (magnitude and angle)
66 [Vrows Vcols] = size(Vpcert); % determine number of solutions (Vrows)
67 PpcesTest = abs(Vpcert).*repmat(K3,Vrows,1).*cos(angle(Vpcert))-abs(Vpcert).*repmat(K4,Vrows,1)... % calculate control values for Vpcert
68 .*sin(angle(Vpcert))+abs(Vpcert).^2.*repmat(K6,Vrows,1)-repmat(Pload,Vrows,1);
69 QpcesTest = abs(Vpcert).*repmat(K4,Vrows,1).*cos(angle(Vpcert))+abs(Vpcert).*repmat(K3,Vrows,1)...
70 .*sin(angle(Vpcert))+abs(Vpcert).^2.*repmat(K8,Vrows,1)-repmat(Qload,Vrows,1);
71 thetaValid = and(abs(PpcesTest)<tol,abs(QpcesTest)<tol); % test if Vpcert valid solution
72 Sctot = ones(Vrows,N); % reset total compensator power variable
73 FailCounter = 0; % reset solution failure counter
74 for sols1=1:Vrows-1 % eliminate duplicate solutions
75     for sols2=2:Vrows
76         if and(thetaValid(sols2,:),and(or(abs(Vpcert(sols1,:))-Vpcert(sols2,:))<tol,...
77         and(isnan(Vpcert(sols1,:)),isnan(Vpcert(sols2,:))),sols1==sols2))
78             Vpcert(sols2,:) = zeros(1,N); % set duplicate Vpcert to zero
79             Sctot(sols2,:) = repmat(NaN,1,N); % set duplicate compensator powers to NaN
80         end;
81     end;
82 end;
83 for n = 1:N
84     if ~thetaValid(:,n) % all solutions of Vpcert invalid - voltage collapse
85         FailCounter = FailCounter+1; % increase solution failure counter
86         % Failed_on_n = n % remove % at start of line for diagnostic feedback
87         Vpcert(:,n) = NaN; % set Vpcert to NaN under voltage collapse
88         Sctot(:,n) = NaN; % set compensator power to NaN under voltage collapse
89     else
90         for m = 1:4 % eliminate individual invalid solutions of Vpcert
91             if ~thetaValid(m,n) % Vpcert(m,n) not valid solution - voltage collapse for this specific solution
92                 Vpcert(m,n) = NaN; % set Vpcert to NaN under voltage collapse
93                 Sctot(m,n) = NaN; % set compensator power to NaN under voltage collapse
94             end;
95         end;
96     end;
97     [temp, index] = sort(abs(abs(Vs(n))-abs(Vpcert(:,n)))); % sort Vpcert solutions in order closest to Vs
98     VpcerNocomp(n) = Vpcert(index(1),n); % VpcerNocomp is best solution under no compensation - closest to Vs
99 end;
100 if FailCounter == 0
101 elseif FailCounter == N % feedback to GUI if no solution is found for some values of delta
102     set(findobj('Tag','NoDeviceRadio'),'BackgroundColor',[1 0 0]); % no solutions for any values of delta - background red in figure 1
103     set(findobj('Tag','NoDeviceRadio2'),'BackgroundColor',[1 0 0]); % no solutions for any values of delta - background red in figure 2
104 else
105     set(findobj('Tag','NoDeviceRadio'),'BackgroundColor',[1 1 0]); % no solutions for some values of delta - background yellow in figure 1
106     set(findobj('Tag','NoDeviceRadio2'),'BackgroundColor',[1 1 0]); % no solutions for some values of delta - background yellow in figure 2
107 end;
108
109 % ----- CALCULATE Vpcer FOR DIFFERENT COMPENSATORS
110
111 switch(DeviceType)
112
113 % ----- CALCULATE Vpcer UNDER NO COMPENSATION
114
115     case 'No__Device' % solve circuit for no compensator
116         Vpcert = VpcerNocomp;
117         [Vrows Vcols] = size(Vpcert); % determine number of solutions (Vrows)
118         Vst = repmat(Vs,Vrows,1); % create Vst matrix to fit number of solutions
119         Vrt = repmat(Vr,Vrows,N); % create Vrt matrix to fit number of solutions
120         Iloadt = (Vpcert-Vload)/VZload+Vpcert/Zload+Vpcert/IZload+Iload+conj(Sload/Vpcert); % solve circuit
121         Irt = (Vpcert-Vrt)/Zr;
122         Ipcert = Iloadt+Irt;
123         Vpcest = Vpcert;
124         Ipcest = (repmat(Vs,Vrows,1)-Vpcest)/Zs;
125         Vpcent = Vpcert;
126         Vcompt = zeros(Vrows,N); % no compensator present
127         Icompt = zeros(Vrows,N);
128         Vcomp2t = zeros(Vrows,N);

```

ADDENDUM B PROGRAM LISTINGS

```

129     lcomp2t = zeros(Vrows,N);
130
131 % ----- CALCULATE Vpccr UNDER SHUNT COMPENSATION
132
133 case 'ShuntDevice'
134     Vpccr = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','UserInput1Edit'),'String')); % get Vpccr magnitude from GUI
135     a = real(conj(Vs/Zs)); % calculate constants
136     b = imag(conj(Vs/Zs));
137     c = real(-1/conj(Zs));
138     d = imag(-1/conj(Zs));
139     e = real(conj(-VrThev/ZrThev));
140     f = imag(conj(-VrThev/ZrThev));
141     g = real(1/conj(ZrThev));
142     h = imag(1/conj(ZrThev));
143     switch(get(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData')) % get Energy store control strategy from GUI
144     case('EnergyStoreNone') % no energystore
145         K3_1 = (a-e).*ones(1,N); % calculate constants
146         K3_3 = zeros(1,N);
147         K4_1 = (f-b).*ones(1,N);
148         K4_3 = zeros(1,N);
149         K6_0 = -Pload.*ones(1,N);
150         K6_2 = (c-g).*ones(1,N);
151         K6_4 = zeros(1,N);
152         [Vpccr, Sctot, FailCounter] = CalcVpcc(K3_1,K3_3,K4_1,K4_3,K6_0,K6_2,K6_4,...
153         Vs,Vpccr,VpccrNocomp,N,tol);
154     case('EnergyStore_PQ') % Pcomp/Qcomp = Kpq
155         Kpq = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','... % get Kpq from GUI
156         'UserInput3Edit'),'String'));
157         K3_1 = (a-e-Kpq*(b-f)).*ones(1,N); % calculate constants
158         K3_3 = zeros(1,N);
159         K4_1 = -(b-f+Kpq*(a-e)).*ones(1,N);
160         K4_3 = zeros(1,N);
161         K6_0 = (-Pload+Kpq*Qload).*ones(1,N);
162         K6_2 = (c-g-Kpq*(d-h)).*ones(1,N);
163         K6_4 = zeros(1,N);
164         [Vpccr, Sctot, FailCounter] = CalcVpcc(K3_1,K3_3,K4_1,K4_3,K6_0,K6_2,K6_4,...
165         Vs,Vpccr,VpccrNocomp,N,tol);
166     case('EnergyStorePcomp') % Pcomp specified
167         Pcomp = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','... % get Pcomp from GUI
168         'UserInput3Edit'),'String'));
169         K3_1 = (a-e).*ones(1,N); % calculate constants
170         K3_3 = zeros(1,N);
171         K4_1 = -(b-f).*ones(1,N);
172         K4_3 = zeros(1,N);
173         K6_0 = (Pcomp-Pload).*ones(1,N);
174         K6_2 = (c-g).*ones(1,N);
175         K6_4 = zeros(1,N);
176         [Vpccr, Sctot, FailCounter] = CalcVpcc(K3_1,K3_3,K4_1,K4_3,K6_0,K6_2,K6_4,...
177         Vs,Vpccr,VpccrNocomp,N,tol);
178     case('EnergyStoreQcomp') % Qcomp specified
179         Qcomp = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','... % get Qcomp from GUI
180         'UserInput3Edit'),'String'));
181         K3_1 = (b-f).*ones(1,N); % calculate constants
182         K3_3 = zeros(1,N);
183         K4_1 = (a-e).*ones(1,N);
184         K4_3 = zeros(1,N);
185         K6_0 = (Qcomp-Qload).*ones(1,N);
186         K6_2 = (d-h).*ones(1,N);
187         K6_4 = zeros(1,N);
188         [Vpccr, Sctot, FailCounter] = CalcVpcc(K3_1,K3_3,K4_1,K4_3,K6_0,K6_2,K6_4,...
189         Vs,Vpccr,VpccrNocomp,N,tol);
190     case('EnergyStoreMins') % minimum energystore
191         K3 = Vpccr^2*((a-e).*(d-h)-(b-f).*(c-g))+Pload.*(b-f)-Qload.*(a-e); % calculate constants
192         K4 = -Vpccr^2*((a-e).*(c-g)+(b-f).*(d-h))+Pload.*(a-e)+Qload.*(b-f);
193         thetaVpccr(1,:) = atan2(-K3,K4); % calculate angle(Vpccr) - thetaVpccr
194         thetaVpccr(2,:) = atan2(K3,-K4);
195         Vpccr = Vpccr.*exp(j*thetaVpccr); % solve Vpccr for thetaVpccr(1:2)

```


ADDENDUM B PROGRAM LISTINGS

```

196 Test = [K3,K3] *cos(thetaVpccr)+[K4,K4] *sin(thetaVpccr); % calculate control values for thetaVpccr
197 thetaValid = and(abs(Test)<tol,isfinite(thetaVpccr)); % test if thetaVpccr valid solution
198 % thetaValid = and(thetaValid,abs(Vpccrt-[Vr,Vr])>tol); % test if power is transferred to receiving-end
199 thetaValid = and(thetaValid,abs(Vpccrt)>abs([VpccrNocomp;VpccrNocomp])); % test if solution is better than no compensation
200 Sctot = ones(2,N); % reset total compensator power variable
201 VpccrSpec = Vpccr; % set user specified Vpccr
202 FailCounter = 0; % reset solution failure counter
203 for n = 1:N
204     VpccrTemp(n) = VpccrSpec; % reset temporary value of VpccrTemp
205     if ~thetaValid(:,n) % loop to lower Vpccr until at least 1 valid solution is found for every delta
206         while and(~thetaValid(:,n),VpccrTemp(n) > abs(VpccrNocomp(n))) % repeat until Vpccr give valid solution or VpccrNocomp better solution
207             VpccrTemp(n) = VpccrTemp(n)-VpccrSpec/100; % reduce Vpccr with 1% of original value
208             K3(n) = VpccrTemp(n)^2*((a(n)-e(n)).*(d(n)-h(n))... % calculate constants
209                 -(b(n)-f(n)).*(c(n)-g(n)))+Pload(n)...
210                 *(b(n)-f(n))-Qload(n).*(a(n)-e(n));
211             K4(n) = -VpccrTemp(n)^2*((a(n)-e(n)).*(c(n)-g(n))...
212                 +(b(n)-f(n)).*(d(n)-h(n)))+Pload(n)...
213                 *(a(n)-e(n))+Qload(n).*(b(n)-f(n));
214             thetaVpccr(1,n) = atan2(-K3(n),K4(n)); % calculate solution1 of angle of Vpccrt - thetaVpccr
215             thetaVpccr(2,n) = atan2(K3(n),-K4(n)); % calculate solution2 of angle of Vpccrt - thetaVpccr
216             Test(:,n) = [K3(n);K3(n)] *cos(thetaVpccr(:,n))... % calculate control values for thetaVpccr
217                 +[K4(n);K4(n)] *sin(thetaVpccr(:,n));
218             thetaValid(:,n) = and(abs(Test(:,n))<tol,... % test if thetaVpccr valid solution
219                 isfinite(thetaVpccr(:,n)));
220         end;
221         Vpccrt(:,n) = VpccrTemp(n) *exp(j*thetaVpccr(:,n)); % solve Vpccrt for thetaVpccr(1:2)
222         FailCounter = FailCounter+1; % increase solution failure counter
223     end;
224     for m = 1:2
225         if ~thetaValid(m,n) % thetaVpccr(m,n) not valid solution
226             Vpccrt(m,n) = VpccrNocomp(n); % compensator not working - reset Vpccrt to no compensation value
227             Sctot(m,n) = NaN; % compensator power excluded from sort to determine best solution
228         end;
229     end;
230 end;
231 end;
232 [Vrows Vcols] = size(Vpccrt);
233 for sols1=[1:Vrows-1] % eliminate duplicate solutions
234     for sols2=[2:Vrows]
235         if and(or(abs(Vpccrt(sols1,:)-Vpccrt(sols2,:))<tol,...
236             and(isnan(Vpccrt(sols1,:)),isnan(Vpccrt(sols2,:))),sols1~=sols2)
237             Vpccrt(sols2,:) = VpccrNocomp;
238             Sctot(sols2,:) = NaN*ones(1,N);
239         end;
240     end;
241 end;
242 Vst = repmat(Vs,Vrows,1); % adjust matrices to size of solutions
243 Vrt = repmat(Vr,Vrows,N); % create Vrt matrix to fit number of solutions
244 Pload = repmat(Pload,Vrows,1);
245 Qload = repmat(Qload,Vrows,1);
246 lloadt = (Vpccrt-Vload)/VZload+Vpccrt./Zload+Vpccrt./IZload+lload... % solve circuit for Vpccrt
247     +conj((Pload+j*Qload)/Vpccrt);
248 Irt = (Vpccrt-Vrt)/Zr;
249 Ipcct = lloadt+Irt;
250 Vpccst = Vpccrt;
251 Ipcct = (Vst-Vpccst)/Zs;
252 Vpccnt = zeros(Vrows,N);
253 Vcompt = zeros(Vrows,N); % no series compensator present
254 Icompt = zeros(Vrows,N);
255 Vcomp2t = Vpccrt; % shunt compensator voltage
256 Icomp2t = Ipcct-Ipcct; % shunt compensator current
257 Scomp2t = Vcompt2t.*conj(Icompt2t); % total compensator power
258 Sctot = Scomp2t.*Sctot; % eliminate invalid solutions from compensator power matrix
259 if FailCounter == 0
260     elseif FailCounter == N % feedback to GUI if no solution is found for some values of delta
261         set(findobj('Tag','ShuntRadio'),'BackgroundColor',[1 0 0]); % no solutions for any values of delta - background red in figure1
262         set(findobj('Tag','ShuntRadio2'),'BackgroundColor',[1 0 0]); % no solutions for any values of delta - background red in figure2

```

ADDENDUM B PROGRAM LISTINGS

```

263 else
264     set(findobj('Tag','ShuntRadio'),'BackgroundColor',[1 1 0]); % no solutions for some values of delta - background yellow in figure1
265     set(findobj('Tag','ShuntRadio2'),'BackgroundColor',[1 1 0]); % no solutions for some values of delta - background yellow in figure2
266 end;
267
268 % ----- CALCULATE Vpccr UNDER SERIES COMPENSATION
269
270 case 'SerieDevice' % series compensation
271     Vpccr = str2num(get(findobj(findobj('Tag','CircuitFigure'),Tag,'UserInput1Edit'),'String')); % get Vpccr magnitude and angle from GUI
272     a = real(1/conj(ZrThev)+Zs/abs(ZrThev).^2)*ones(1,N);
273     b = imag(1/conj(ZrThev)+Zs/abs(ZrThev).^2)*ones(1,N);
274     c = real(-conj(VrThev./ZrThev)-conj(VrThev).*Zs/abs(ZrThev).^2);
275     d = imag(-conj(VrThev./ZrThev)-conj(VrThev).*Zs/abs(ZrThev).^2);
276     e = real(-Vs/conj(ZrThev)-VrThev.*Zs/abs(ZrThev).^2);
277     f = imag(-Vs/conj(ZrThev)-VrThev.*Zs/abs(ZrThev).^2);
278     g = real(Sload+Sload.*Zs./ZrThev+Vs.*conj(VrThev./ZrThev)+Zs*abs(VrThev./ZrThev).^2...
279         +Zs*conj(Sload./ZrThev));
280     h = imag(Sload+Sload.*Zs./ZrThev+Vs.*conj(VrThev./ZrThev)+Zs*abs(VrThev./ZrThev).^2...
281         +Zs*conj(Sload./ZrThev));
282     k = real(-Vs.*Sload-VrThev.*Zs./ZrThev.*Sload);
283     l = imag(-Vs.*Sload-VrThev.*Zs./ZrThev.*Sload);
284     o = real(-conj(Sload.*VrThev./ZrThev).*Zs);
285     p = imag(-conj(Sload.*VrThev./ZrThev).*Zs);
286     q = real(abs(Sload).^2*Zs);
287     r = imag(abs(Sload).^2*Zs);
288     switch(get(findobj(findobj('Tag','CircuitFigure'),Tag,'EnergyStoreText'),'UserData'))
289         case('EnergyStoreNone') % In-line compensator with no energy storage
290             K3_1 = (k+o).*ones(1,N);
291             K3_3 = (c+e).*ones(1,N);
292             K4_1 = (l-p).*ones(1,N);
293             K4_3 = (f-d).*ones(1,N);
294             K6_0 = q.*ones(1,N);
295             K6_2 = g.*ones(1,N);
296             K6_4 = a.*ones(1,N);
297             [Vpccr, Sctot, FailCounter] = CalcVpcc(K3_1,K3_3,K4_1,K4_3,K6_0,K6_2,K6_4,...
298                 Vs,Vpccr,VpccrNocomp,N,tol);
299         case('EnergyStore_PQ') % series compensation for fixed P/Q ratio: P/Q = Kpq
300             Kpq = str2num(get(findobj(findobj('Tag','CircuitFigure'),Tag,... % get Kpq from GUI
301                 'UserInput3Edit'),'String'));
302             K3_1 = (k+o-Kpq*(l+p)).*ones(1,N);
303             K3_3 = (c+e-Kpq*(d+f)).*ones(1,N);
304             K4_1 = (l-p-Kpq*(o-k)).*ones(1,N);
305             K4_3 = (f-d-Kpq*(c-e)).*ones(1,N);
306             K6_0 = (q-Kpq*r).*ones(1,N);
307             K6_2 = (g-Kpq*h).*ones(1,N);
308             K6_4 = (a-Kpq*b).*ones(1,N);
309             [Vpccr, Sctot, FailCounter] = CalcVpcc(K3_1,K3_3,K4_1,K4_3,K6_0,K6_2,K6_4,...
310                 Vs,Vpccr,VpccrNocomp,N,tol);
311         case('EnergyStorePcomp') % series compensation for fixed Pcomp
312             Pcomp = str2num(get(findobj(findobj('Tag','CircuitFigure'),Tag,... % get Pcomp from GUI
313                 'UserInput3Edit'),'String'));
314             K3_1 = (k+o).*ones(1,N);
315             K3_3 = (c+e).*ones(1,N);
316             K4_1 = (l-p).*ones(1,N);
317             K4_3 = (f-d).*ones(1,N);
318             K6_0 = q.*ones(1,N);
319             K6_2 = (g-Pcomp).*ones(1,N);
320             K6_4 = a.*ones(1,N);
321             [Vpccr, Sctot, FailCounter] = CalcVpcc(K3_1,K3_3,K4_1,K4_3,K6_0,K6_2,K6_4,...
322                 Vs,Vpccr,VpccrNocomp,N,tol);
323         case('EnergyStoreQcomp') % series compensation for fixed Qcomp
324             Qcomp = str2num(get(findobj(findobj('Tag','CircuitFigure'),Tag,... % get Qcomp from GUI
325                 'UserInput3Edit'),'String'));
326             K3_1 = (l+p).*ones(1,N);
327             K3_3 = (d+f).*ones(1,N);
328             K4_1 = (o-k).*ones(1,N);
329             K4_3 = (c-e).*ones(1,N);

```

ADDENDUM B PROGRAM LISTINGS

```

330     K6_0 = r.*ones(1,N);
331     K6_2 = (h-Qcomp).*ones(1,N);
332     K6_4 = b.*ones(1,N);
333     [Vpccrt, Sctot, FailCounter] = CalcVpccr(K3_1,K3_3,K4_1,K4_3,K6_0,K6_2,K6_4,...
334         Vs,Vpccr,VpccrNocomp,N,tol);
335 case('EnergyStoreMinS') % series compensation with minimum compensator rating
336     K1 = 2*(Vpccr^6*(c.*f-d.*e)+Vpccr^4*(c.*l-e.*p-d.*k+f.*o)...
337         +Vpccr^2*(l.*o-k.*p)).*ones(1,N);
338     K3 = (Vpccr^7*(a.*f-a.*d+b.*c-b.*e)...
339         +Vpccr^5*(a.*l-a.*p+g.*f-g.*d+b.*o-b.*k+c.*h-e.*h)...
340         +Vpccr^3*(g.*l-g.*p+f.*q-d.*q+h.*o-h.*k+c.*r-e.*r)...
341         +Vpccr*(l.*q-p.*q+o.*r-k.*r)).*ones(1,N);
342     K4 = -(Vpccr^7*(a.*c+a.*e+b.*d+b.*f)...
343         +Vpccr^5*(a.*k+a.*o+c.*g+e.*g+b.*l+b.*p+d.*h+f.*h)...
344         +Vpccr^3*(g.*k+g.*o+c.*q+e.*q+h.*h.*p+d.*r+f.*r)...
345         +Vpccr*(k.*q+o.*q+l.*r+p.*r)).*ones(1,N);
346     K5 = -4*(Vpccr^6*(c.*e+d.*f)+Vpccr^4*(c.*k+e.*o+f.*p+d.*l)...
347         +Vpccr^2*(k.*o+l.*p)).*ones(1,N);
348 if and(K1 == 0,K5 == 0)
349     thetaVpccr = atan2(-K3,K4);
350     [Vrows Vcols] = size(thetaVpccr)
351     Vpccrt = Vpccr*exp(j*thetaVpccr);
352     Sctot = ones(Vrows,N); % reset total compensator power variable
353 else
354     thetaVpccr = Solve4thOrder(K1,K3,K4,K5);
355     [Vrows Vcols] = size(thetaVpccr);
356     Vpccrt = Vpccr*exp(j*thetaVpccr);
357     Test = repmat(K1,Vrows,1).*cos(thetaVpccr).^2... % calculate control values for thetaVpccr
358         -repmat(K1,Vrows,1).*sin(thetaVpccr).^2+...
359         repmat(K3,Vrows,1).*cos(thetaVpccr)...
360         +repmat(K4,Vrows,1).*sin(thetaVpccr)+...
361         repmat(K5,Vrows,1).*sin(thetaVpccr).*cos(thetaVpccr);
362     thetaValid = abs(Test)<tol; % test if thetaVpccr valid solution
363     thetaValid = and(thetaValid,abs(Vpccrt)>abs(repmat(VpccrNocomp,...
364         Vrows,1)));
365     Sctot = ones(Vrows,N); % reset total compensator power variable
366     VpccrSpec = Vpccr;
367     FailCounter = 0;
368     for n = 1:N
369         VpccrTemp(n) = VpccrSpec;
370         if ~thetaValid(:,n)
371             while and(~thetaValid(:,n),... % repeat until Vpccr low enough for valid solution
372                 VpccrTemp(n)>abs(VpccrNocomp(n)))
373
374                 VpccrTemp(n) = VpccrTemp(n)-VpccrSpec/100;
375                 K1(n) = 2*(VpccrTemp(n)^6*(c(n).*f(n)...
376                     -d(n).*e(n))+VpccrTemp(n)^4...
377                     *(c(n).*l(n)-e(n).*p(n)-d(n).*k(n)...
378                     +f(n).*o(n))+VpccrTemp(n)^2...
379                     *(l(n).*o(n)-k(n).*p(n)));
380                 K3(n) = (VpccrTemp(n)^7*(a(n).*f(n)...
381                     -a(n).*d(n)+b(n).*c(n)-b(n).*e(n)...
382                     +VpccrTemp(n)^5*(a(n).*l(n)...
383                     -a(n).*p(n)+g(n).*f(n)-g(n).*d(n)...
384                     +b(n).*o(n)-b(n).*k(n)+c(n).*h(n)...
385                     -e(n).*h(n))+VpccrTemp(n)^3...
386                     *(g(n).*l(n)-g(n).*p(n)+f(n).*q(n)...
387                     -d(n).*q(n)+h(n).*o(n)-h(n).*k(n)...
388                     +c(n).*r(n)-e(n).*r(n)...
389                     +VpccrTemp(n)*(l(n).*q(n)-p(n).*q(n)...
390                     +o(n).*r(n)-k(n).*r(n)));
391                 K4(n) = -(VpccrTemp(n)^7*(a(n).*c(n)...
392                     +a(n).*e(n)+b(n).*d(n)+b(n).*f(n)...
393                     +VpccrTemp(n)^5*(a(n).*k(n)...
394                     +a(n).*o(n)+c(n).*g(n)+e(n).*g(n)...
395                     +b(n).*l(n)+b(n).*p(n)+d(n).*h(n)...
396                     +f(n).*h(n))+VpccrTemp(n)^3...

```

ADDENDUM B PROGRAM LISTINGS

```

397         *(g(n).*k(n)+g(n).*o(n)+c(n).*q(n)...
398         +e(n).*q(n)+h(n).*l(n)+h(n).*p(n)...
399         +d(n).*r(n)+f(n).*r(n))...
400         +VpccrTemp(n)*(k(n).*q(n)+o(n).*q(n)...
401         +l(n).*r(n)+p(n).*r(n));
402     K5(n) = -4*(VpccrTemp(n)^6*(c(n).*e(n)...
403         +d(n).*f(n))+VpccrTemp(n)^4*(c(n)...
404         .*k(n)+e(n).*o(n)+f(n).*p(n)...
405         +d(n).*l(n))+VpccrTemp(n)^2*(k(n)...
406         .*o(n)+l(n).*p(n));
407     thetaVpccr(:,n) = Solve4thOrder(K1(n),K3(n),...
408         K4(n),K5(n));
409     Vpccr(:,n) = VpccrTemp(n)...
410         .*exp(j*thetaVpccr(:,n));
411     Test(:,n) = repmat(K1(n),Vrows,1)...
412         .*cos(thetaVpccr(:,n))^2...
413         -repmat(K1(n),Vrows,1)...
414         .*sin(thetaVpccr(:,n))^2...
415         +repmat(K3(n),Vrows,1)...
416         .*cos(thetaVpccr(:,n))...
417         +repmat(K4(n),Vrows,1)...
418         .*sin(thetaVpccr(:,n))...
419         +repmat(K5(n),Vrows,1)...
420         .*sin(thetaVpccr(:,n))...
421         .*cos(thetaVpccr(:,n));
422     abs(Vpccr(:,n)-Vr(n))>tol;
423     thetaValid(:,n) = abs(Test(:,n))<tol;
424     end;
425     FailCounter = FailCounter+1;
426     end;
427     if ~thetaValid(:,n)
428         Vpccr(1,n) = VpccrNocomp(n);
429         Vpccr(2:Vrows,n) = NaN;
430         Sctot(:,n) = NaN;
431     else
432         for m = 1:Vrows
433             if ~thetaValid(m,n)
434                 Vpccr(m,n) = NaN;
435                 Sctot(m,n) = NaN;
436             end;
437         end;
438     end
439     end;
440     end;
441     end;
442     [Vrows Vcols] = size(Vpccr);
443     for sols1=1:Vrows-1
444         for sols2=[2:Vrows]
445             if and(or(abs(Vpccr(sols1,:)-Vpccr(sols2,:))<tol,...
446                 and(isnan(Vpccr(sols1,:)),isnan(Vpccr(sols2,:))),sols1~=sols2)
447                 Vpccr(sols2,:) = VpccrNocomp;
448                 Sctot(sols2,:) = repmat(NaN,1,N);
449             end;
450         end;
451     end;
452     Vst = repmat(Vs,Vrows,1);
453     Vrt = repmat(Vr,Vrows,N);
454     Pload = repmat(Pload,Vrows,1);
455     Qload = repmat(Qload,Vrows,1);
456     Iloadt = (Vpccr-Vload)/VZload+Vpccr/Zload+Vpccr./IZload+Iload...
457         +conj((Pload+j*Qload)/Vpccr);
458     Irt = (Vpccr-Vrt)/Zr;
459     Ipcert = Iloadt+Irt;
460     Ipcst = Ipcert;
461     Vpcst = Vst-Ipcst*Zs;
462     Vpcent = Vpcst;
463     Vcompt = Vpccr-Vpcst;

```

% solve circuit for thetaVpccr(1:2)

% calculate control values for thetaVpccr

% test if thetaVpccr valid solution

% no valid solution for specific n (delta)

% set first solution to no compensation voltage

% no other solutions

% at least one valid solution for specific n (delta)

% thetaVpccr(m,n) not valid solution

% eliminate this solution

% eliminate duplicate solutions

% create Vrt matrix to fit number of solutions

```

464 Icompt = Ipcst;
465 Scompt = Vcompt.*conj(Icompt);
466 Setot = Scompt.*Scot;
467 Vcomp2t = zeros(Vrows,N); % no second compensator present
468 Icomp2t = zeros(Vrows,N);
469 if FailCounter == 0
470 elseif FailCounter == N % feedback to GUI if no solution is found for some values of delta
471     set(findobj('Tag','SeriesRadio'),'BackgroundColor',[1 0 0]); % no solutions for any values of delta - background red
472     set(findobj('Tag','SeriesRadio2'),'BackgroundColor',[1 0 0]); % no solutions for any values of delta - background red
473 else
474     set(findobj('Tag','SeriesRadio'),'BackgroundColor',[1 1 0]); % no solutions for some values of delta - background yellow
475     set(findobj('Tag','SeriesRadio2'),'BackgroundColor',[1 1 0]); % no solutions for some values of delta - background yellow
476 end;
477
478 % ----- CALCULATE Vpccs UNDER SERIE-SHUNT COMPENSATION (Vpccr SPECIFIED BY USER)
479
480 case 'SeriesShunt' % Series-shunt compensation
481     Vpccr = str2num(get(findobj(findobj('Tag','CircuitFigure'),... % get Vpccr magnitude and angle from GUI
482     'Tag','UserInput1Edit'),'String'))*exp(j*str2num(get(findobj(findobj('Tag',...
483     'CircuitFigure'),'Tag','UserInput2Edit'),'String')));
484     PF = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','UserInput3Edit'),'String')); % get power factor at PCCS from GUI
485     a = real(conj(Vs/Zs)).*ones(1,N);
486     b = imag(conj(Vs/Zs)).*ones(1,N);
487     c = real(-1/conj(Zs)).*ones(1,N);
488     d = imag(-1/conj(Zs)).*ones(1,N);
489     switch(get(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData'))
490         case('EnergyStoreNone') % In-line compensator with no energy storage
491             Spccr = Sload+Vpccr*conj((Vpccr-VrThev)/ZrThev);
492             Ppccs = real(Spccr);
493             Qpccs = sqrt(1-PF^2)/PF*Ppccs;
494             Vpccst = SolvePQ(a,b,c,d,Ppccs,Qpccs); % solve Vpcc and thetaVpcc
495             [Vrows Vcols] = size(Vpccst);
496             PpccsTest = abs(Vpccst).*repmat(a,Vrows,1).*cos(angle(Vpccst))-abs(Vpccst)... % calculate control values for thetaVpccr
497             .*repmat(b,Vrows,1).*sin(angle(Vpccst))+abs(Vpccst).^2...
498             .*repmat(c,Vrows,1)-repmat(Ppccs,Vrows,1);
499             QpccsTest = abs(Vpccst).*repmat(b,Vrows,1).*cos(angle(Vpccst))+abs(Vpccst)...
500             .*repmat(a,Vrows,1).*sin(angle(Vpccst))+abs(Vpccst).^2...
501             .*repmat(d,Vrows,1)-repmat(Qpccs,Vrows,1);
502             thetaValid = and(abs(PpccsTest)<tol,abs(QpccsTest)<tol); % test if thetaVpccr valid solution
503             thetaValid = and(thetaValid,repmat(abs(Vpccr)>abs(VpccrNocomp),4,1));
504             Scot = ones(Vrows,N); % reset total compensator power variable
505             VpccrSpec = abs(Vpccr);
506             FailCounter = 0;
507             for n = 1:N
508                 VpccrTemp(n) = VpccrSpec;
509                 if ~thetaValid(:,n)
510                     while and(~thetaValid(:,n),VpccrTemp(n) > abs(VpccrNocomp(n))) % repeat until Vpccr low enough for valid solution
511                         VpccrTemp(n) = VpccrTemp(n)-VpccrSpec/100;
512                         Spccr(n) = Sload(n)+VpccrTemp(n)*exp(j*angle(Vpccr))...
513                         *conj((VpccrTemp(n)*exp(j*angle(Vpccr))...
514                         -VrThev(n))/ZrThev);
515                         Ppccs(n) = real(Spccr(n));
516                         Qpccs(n) = sqrt(1-PF^2)/PF*Ppccs(n);
517                         Vpccst(:,n) = SolvePQ(a(n),b(n),c(n),d(n),Ppccs(n),... % solve Vpcc and thetaVpcc
518                         Qpccs(n));
519                         PpccsTest(:,n) = abs(Vpccst(:,n))... % calculate control values for thetaVpccr
520                         .*[a(n),a(n),a(n),a(n)]...
521                         .*cos(angle(Vpccst(:,n)))...
522                         -abs(Vpccst(:,n)).*[b(n),b(n),b(n),b(n)]...
523                         .*sin(angle(Vpccst(:,n)))...
524                         +abs(Vpccst(:,n)).^2.*[c(n),c(n),c(n),c(n)]...
525                         -[Ppccs(n);Ppccs(n);Ppccs(n);Ppccs(n)];
526                         QpccsTest(:,n) = abs(Vpccst(:,n))...
527                         .*[b(n),b(n),b(n),b(n)]...
528                         .*cos(angle(Vpccst(:,n)))...
529                         +abs(Vpccst(:,n)).*[a(n),a(n),a(n),a(n)]...
530                         .*sin(angle(Vpccst(:,n)))...

```

Stellenbosch University <http://scholar.sun.ac.za>
ADDENDUM B PROGRAM LISTINGS

```

531         +abs(Vpccst(:,n)).^2.*[d(n);d(n);d(n);d(n)]...
532         -[Qpccs(n);Qpccs(n);Qpccs(n);Qpccs(n)];
533         thetaValid(:,n) = and(abs(PpccsTest(:,n))<tol,...           % test if theta Vpccr valid solution
534         abs(QpccsTest(:,n))<tol);
535     end;
536     FailCounter = FailCounter+1;
537 end;
538 Vpccert(:,n) = repmat(VpccrTemp(n)*exp(j*angle(Vpccr)),4,1);
539 if ~thetaValid(:,n)                                             % no valid solution for specific n (delta)
540     Vpccst(1,n) = VpccrNocomp(n);                               % set first solution to no compensation voltage
541     Vpccst(2:Vrows,n) = NaN;                                    % no other solutions
542     Vpccrt(1,n) = VpccrNocomp(n);                               % set first solution to no compensation voltage
543     Vpccrt(2:Vrows,n) = NaN;                                    % no other solutions
544     Sctot(:,n) = NaN;
545 else
546     for m = 1:Vrows                                             % at least one valid solution for specific n (delta)
547         if ~thetaValid(m,n)                                     % theta Vpccr(m,n) not valid solution
548             Vpccst(m,n) = NaN;                                  % eliminate this solution
549             Vpccrt(m,n) = NaN;                                  % eliminate this solution
550             Sctot(m,n) = NaN;
551         end;
552     end;
553 end
554 end;
555 case('EnergyStore_PQ')
556 case('EnergyStoreMins')
557 case('EnergyStore Vpccr')
558 end;
559 [Vrows Vcols] = size(Vpccst);
560 for sols1=1:Vrows-1                                           % eliminate duplicate solutions
561     for sols2=2:Vrows
562         if and(or(abs(Vpccst(sols1,:)-Vpccst(sols2,:))<tol,...
563         and(isnan(Vpccst(sols1,:)),isnan(Vpccst(sols2,:)))),sols1~=sols2)
564             Vpccst(sols2,:) = VpccrNocomp;
565             Vpccrt(sols2,:) = VpccrNocomp;
566             Sctot(sols2,:) = repmat(NaN,1,N);
567         end;
568     end;
569 end;
570 Vst = repmat(Vs,Vrows,1);
571 Vrt = repmat(Vr,Vrows,N);                                     % create Vrt matrix to fit number of solutions
572 Pload = repmat(Pload,Vrows,1);
573 Qload = repmat(Qload,Vrows,1);
574 Iloadt = (Vpccr-Vload)/VZload+Vpccrt/Zload+Vpccrt/IZload+Iload...
575         +conj((Pload+j*Qload)/Vpccr);
576 Irt = (Vpccr-Vrt)/Zr;
577 Ipcert = Iloadt+Irt;
578 Ipcst = (Vst-Vpccst)/Zs;
579 Vpcent = Vpccst;
580 Vcompt = Vpccr-Vpccst;
581 Icompt = Ipcst;
582 Scompt = Vcompt.*conj(Icompt);
583 Vcomp2t = Vpccr;
584 Icomp2t = Ipcert-Ipcst;
585 Scomp2t = Vcomp2t.*conj(Icomp2t);
586 Setot = (Scompt+Scomp2t).*Setot;
587 if FailCounter == 0
588 elseif FailCounter == N                                       % feedback to GUI if no solution is found for some values of delta
589     set(findobj('Tag','SeriesShuntRadio'),'BackgroundColor',[1 0 0]); % no solutions for any values of delta - background red
590     set(findobj('Tag','SeriesShuntRadio2'),'BackgroundColor',[1 0 0]); % no solutions for any values of delta - background red
591 else
592     set(findobj('Tag','SeriesShuntRadio'),'BackgroundColor',[1 1 0]); % no solutions for some values of delta - background yellow
593     set(findobj('Tag','SeriesShuntRadio2'),'BackgroundColor',[1 1 0]); % no solutions for some values of delta - background yellow
594 end;
595
596 % ----- CALCULATE Vpccs UNDER IN-LINE COMPENSATION (Vpccr SPECIFIED BY USER)
597

```

ADDENDUM B PROGRAM LISTINGS

```

598 case 'InLineDvice' % In-line compensation
599 Vpccr = str2num(get(findobj(findobj('Tag','CircuitFigure'),... % get Vpccr magnitude and angle from GUI
600 'Tag','UserInput1Edit','String'))*exp(j*str2num(get(findobj(findobj('Tag',...
601 'CircuitFigure'),'Tag','UserInput2Edit','String')));
602 PF = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','UserInput3Edit','String')); % get power factor at PCCS from GUI
603 Vdcr = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','UserInput4Edit','String')); % get receiving-end dc voltage magnitude from GUI
604 Rdc = str2num(get(findobj(findobj('Tag','CircuitFigure'),'Tag','UserInput5Edit','String')); % get dc line resistance from GUI
605 a = real(conj(Vs/Zs))*ones(1,N);
606 b = imag(conj(Vs/Zs))*ones(1,N);
607 c = real(-1/conj(Zs))*ones(1,N);
608 d = imag(-1/conj(Zs))*ones(1,N);
609 switch(get(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData'))
610 case('EnergyStoreNone') % In-line compensator with no energy storage
611 Spccr = Sload+Vpccr*conj((Vpccr-VrThev)/ZrThev);
612 Pdcr = real(Spccr);
613 Idc = Pdcr/Vdcr;
614 Vdcs = Vdcr+Idc*Rdc;
615 Pdc = Vdcs*Idc;
616 Ppccs = Pdc;
617 Qpccs = Ppccs*tan(PF); %sqrt(1-PF^2)/PF*Ppccs;
618 Vpccst = SolvePQ(a,b,c,d,Ppccs,Qpccs); % solve Vpcc and theta Vpcc
619 [Vrows Vcols] = size(Vpccst);
620 PpccsTest = abs(Vpccst)*repmat(a,Vrows,1)*cos(angle(Vpccst))-abs(Vpccst)... % calculate control values for theta Vpccr
621 .*repmat(b,Vrows,1)*sin(angle(Vpccst))+abs(Vpccst).^2...
622 .*repmat(c,Vrows,1)-repmat(Ppccs,Vrows,1);
623 QpccsTest = abs(Vpccst)*repmat(b,Vrows,1)*cos(angle(Vpccst))+abs(Vpccst)...
624 .*repmat(a,Vrows,1)*sin(angle(Vpccst))+abs(Vpccst).^2...
625 .*repmat(d,Vrows,1)-repmat(Qpccs,Vrows,1);
626 thetaValid = and(abs(PpccsTest)<tol,abs(QpccsTest)<tol); % test if theta Vpccr valid solution
627 thetaValid = and(thetaValid,repmat(abs(Vpccr)>abs(VpccrNocomp),4,1));
628 Sctot = ones(Vrows,N); % reset total compensator power variable
629 VpccrSpec = abs(Vpccr);
630 FailCounter = 0;
631 for n = 1:N
632 VpccrTemp(n) = VpccrSpec;
633 if ~thetaValid(:,n)
634 while and(~thetaValid(:,n),VpccrTemp(n) > abs(VpccrNocomp(n))) % repeat until Vpccr low enough for valid solution
635 VpccrTemp(n) = VpccrTemp(n)-VpccrSpec/100;
636 Spccr(n) = Sload(n)+VpccrTemp(n)*exp(j*angle(Vpccr))...
637 .*conj((VpccrTemp(n)*exp(j*angle(Vpccr))...
638 -VrThev(n))/ZrThev);
639 Pdcr(n) = real(Spccr(n));
640 Idc(n) = Pdcr(n)/Vdcr;
641 Vdcs(n) = Vdcr+Idc(n)*Rdc;
642 Pdc(n) = Vdcs(n)*Idc(n);
643 Ppccs(n) = Pdc(n);
644 Qpccs(n) = Ppccs(n)*tan(PF); %sqrt(1-PF^2)/PF*Ppccs(n);
645 Vpccst(:,n) = SolvePQ(a(n),b(n),c(n),d(n),Ppccs(n),... % solve Vpcc and theta Vpcc
646 Qpccs(n));
647 PpccsTest(:,n) = abs(Vpccst(:,n))... % calculate control values for theta Vpccr
648 .*[a(n),a(n),a(n),a(n)]...
649 .*cos(angle(Vpccst(:,n)))...
650 -abs(Vpccst(:,n)).*[b(n),b(n),b(n),b(n)]...
651 .*sin(angle(Vpccst(:,n)))...
652 +abs(Vpccst(:,n)).^2.*[c(n),c(n),c(n),c(n)]...
653 -[Ppccs(n);Ppccs(n);Ppccs(n);Ppccs(n)];
654 QpccsTest(:,n) = abs(Vpccst(:,n))...
655 .*[b(n),b(n),b(n),b(n)]...
656 .*cos(angle(Vpccst(:,n)))...
657 +abs(Vpccst(:,n)).*[a(n),a(n),a(n),a(n)]...
658 .*sin(angle(Vpccst(:,n)))...
659 +abs(Vpccst(:,n)).^2.*[d(n),d(n),d(n),d(n)]...
660 -[Qpccs(n);Qpccs(n);Qpccs(n);Qpccs(n)];
661 thetaValid(:,n) = and(abs(PpccsTest(:,n))<tol,... % test if theta Vpccr valid solution
662 abs(QpccsTest(:,n))<tol);
663 end;
664 FailCounter = FailCounter+1;

```

ADDENDUM B PROGRAM LISTINGS

```

665     end;
666     Vpccr(:,n) = repmat(VpccrTemp(n)*exp(j*angle(Vpccr)),4,1);
667     if ~thetaValid(n) % no valid solution for specific n (delta)
668         Vpccst(1,n) = VpccrNocomp(n); % set first solution to no compensation voltage
669         Vpccst(2:Vrows,n) = NaN; % no other solutions
670         Vpccrt(1,n) = VpccrNocomp(n); % set first solution to no compensation voltage
671         Vpccrt(2:Vrows,n) = NaN; % no other solutions
672         Sctot(:,n) = NaN;
673     else
674         for m = 1:Vrows % at least one valid solution for specific n (delta)
675             if ~thetaValid(m,n) % theta Vpccr(m,n) not valid solution
676                 Vpccst(m,n) = NaN; % eliminate this solution
677                 Vpccrt(m,n) = NaN; % eliminate this solution
678                 Sctot(m,n) = NaN;
679             end;
680         end;
681     end
682     end;
683     case('EnergyStore_PQ')
684     case('EnergyStoreMinS')
685     case('EnergyStoreVpcc')
686 end;
687 [Vrows Vcols] = size(Vpccst);
688 for sols1=[1:Vrows-1] % eliminate duplicate solutions
689     for sols2=[2:Vrows]
690         if and(or(abs(Vpccst(sols1,:)-Vpccst(sols2,:))<tol,...
691             and(isnan(Vpccst(sols1,:)),isnan(Vpccst(sols2,:))),sols1~=sols2)
692             Vpccst(sols2,:) = VpccrNocomp;
693             Vpccrt(sols2,:) = VpccrNocomp;
694             Sctot(sols2,:) = repmat(NaN,1,N);
695         end;
696     end;
697 end;
698 Vst = repmat(Vs,Vrows,1);
699 Vrt = repmat(Vr,Vrows,N); % create Vrt matrix to fit number of solutions
700 Pload = repmat(Pload,Vrows,1);
701 Qload = repmat(Qload,Vrows,1);
702 Iloadt = (Vpccrt-Vload)/VZload+Vpccrt/Zload+Vpccrt./Zload+Iload...
703     +conj((Pload+j*Qload)/Vpccrt);
704 Irt = (Vpccrt-Vrt)/Zr;
705 Ipcprt = Iloadt+Irt;
706 Ipcpst = (Vst-Vpccst)/Zs;
707 Vpcent = zeros(4,N);
708 Vcomp2t = repmat(Vdcr,4,N);
709 Scomp2t = real(Vpccrt.*conj(Ipcprt));
710 Icomp2t = Scomp2t./Vcomp2t;
711 Vcompt = Vcomp2t+Icomp2t*Rdc;
712 Icompt = Icomp2t;
713 Scompt = Vcompt.*conj(Icompt);
714 Sctot = (Scompt+Scomp2t).*Sctot;
715 if FailCounter == 0
716 elseif FailCounter == N % feedback to GUI if no solution is found for some values of delta
717     set(findobj('Tag','InLineRadio'),'BackgroundColor',[1 0 0]); % no solutions for any values of delta - background red
718     set(findobj('Tag','InLineRadio2'),'BackgroundColor',[1 0 0]); % no solutions for any values of delta - background red
719 else
720     set(findobj('Tag','InLineRadio'),'BackgroundColor',[1 1 0]); % no solutions for some values of delta - background yellow
721     set(findobj('Tag','InLineRadio2'),'BackgroundColor',[1 1 0]); % no solutions for some values of delta - background yellow
722 end;
723 end;
724
725 % ----- ELIMINATE DUPLICATE SOLUTIONS
726
727 for sols1=[1:Vrows]
728     if isnan(Sctot(sols1,:))
729         if get(findobj(findobj('Tag','VRegFigure'),'Tag','strcat('Solution',num2str(sols1),'Radio')),...
730             'Value')
731             set(findobj(findobj('Tag','VRegFigure'),'Tag','SolutionORadio'),'Value',1);

```


ADDENDUM B PROGRAM LISTINGS

```

732         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData',1);
733         set(findobj(findobj('Tag','VRegFigure'),'Tag','strcat('Solution',...
734             num2str(sols1),'Radio')),'Enable','off','Value',0);
735     end
736     set(findobj(findobj('Tag','VRegFigure'),'Tag','strcat('Solution',num2str(sols1),'Radio')),...
737         'Enable','off)
738 end
739 end
740
741 % ----- ELIMINATE ERATIC PHASE JUMPS WHEN COMPENSATOR CURRENT OR VOLTAGE ZERO
742
743 for m = 1:Vrows
744     for n = 3:N
745         if abs(Vcompt(m,n-2:n))<tol
746             Vcompt(m,n-2:n) = 0;
747         end
748         if abs(lcomp2t(m,n-2:n))<tol
749             lcomp2t(m,n-2:n) = 0;
750         end
751     end
752 end
753
754 % ----- DETERMINE OPTIMUM SOLUTION
755
756 if Vrows > 1 % determine optimum solution if more than one solution exist
757     sort1 = round(abs(abs(Vpccr)-abs(Vpccrt))/tol);
758     [temp,index] = sort(sort1); % sort according to abs(Vpccr)
759     for n = 1:N
760         if or(abs(sort1(index(1,n),n)-sort1(index(2,n),n))<tol,isnan(sort1(index(1:2,n),n))) % Difference between best 2 solutions in abs(Vpccr) insignificant
761             if Vrows > 2
762                 if or(abs(sort1(index(1,n),n)-sort1(index(3,n),n))<tol,... % Difference between best 3 solutions in abs(Vpccr) insignificant
763                     isnan(sort1(index(1:3,n),n)))
764                     if or(abs(sort1(index(1,n),n)-sort1(index(4,n),n))<tol,... % Difference between best 4 solutions in abs(Vpccr) insignificant
765                         isnan(sort1(index(1:4,n),n)))
766                         max = 4;
767                     else
768                         max = 3;
769                     end;
770                 else
771                     max = 2;
772                 end;
773             else
774                 max = 2;
775             end;
776             sort2 = round(abs(Sctot(index(1:max,n),n))/tol);
777             if or(abs(sort2(1)-sort2(2))<tol,isnan(sort2)) % Difference between best 2 solutions in abs(Sctot) insignificant
778                 sort3 = round(angle(Sctot(index(1:max,n),n))/tol);
779                 if or(abs(sort3(1)-sort3(2))<tol,isnan(sort3)) % Difference between best 2 solutions in angle(Sctot) insignificant
780                     sort4 = round(abs(angle(Vpccr(index(1:max,n),n)))/tol);
781                     if or(abs(sort4(1)-sort4(2))<tol,isnan(sort4)) % Difference between best 2 solutions in angle(Vpccr) insignificant
782                         sort5 = -sign(angle(Vpccr(index(1:max,n),n)));
783                         if or(abs(sort5(1)-sort5(2))<tol,isnan(sort5)) % Difference between best 2 solutions in sign(angle(Vpccr)) insignificant
784                             sort6 = round(abs(Vst(index(1:max,n),n))... % sort according to abs(Vpccs)
785                                 -abs(Vpccst(index(1:max,n),n)))/tol);
786                             [temp(1:max,n),index(1:max,n)] = sort(sort6);
787                             index(1:max,n) = index(index(1:max,n),n);
788                         else
789                             [temp(1:max,n),index(1:max,n)] = sort(sort5); % sort according to sign of angle(Vpccr) - positive first
790                             index(1:max,n) = index(index(1:max,n),n);
791                         end;
792                     else
793                         [temp(1:max,n),index(1:max,n)] = sort(sort4); % sort according to abs(Vpccs)
794                         index(1:max,n) = index(index(1:max,n),n);
795                     end;
796                 else
797                     [temp(1:max,n),index(1:max,n)] = sort(sort3); % sort according to angle(Sctot)
798                     index(1:max,n) = index(index(1:max,n),n);

```

```

799         end;
800     else
801         [temp(1:max,n),index(1:max,n)] = sort(sort2);           % sort according to abs(Sciot)
802         index(1:max,n) = index(index(1:max,n),n);
803     end;
804 end;
805 Vpccs(1,n) = Vpcst(index(1,n),n);
806 VVpccr(1,n) = Vpcrt(index(1,n),n);
807 Vpcen(1,n) = Vpcnt(index(1,n),n);
808 Vcomp(1,n) = Vcompt(index(1,n),n);
809 Vcomp2(1,n) = Vcomp2t(index(1,n),n);
810 Ipces(1,n) = Ipcst(index(1,n),n);
811 Ipcer(1,n) = Ipcrt(index(1,n),n);
812 Iload(1,n) = Iloadt(index(1,n),n);
813 Ir(1,n) = Irt(index(1,n),n);
814 Icomp(1,n) = Icompt(index(1,n),n);
815 Icomp2(1,n) = Icomp2t(index(1,n),n);
816 end;
817 else
818     Vpccs(1,:) = Vpcst(1,:);
819     VVpccr(1,:) = Vpcrt(1,:);
820     Vpcen(1,:) = Vpcnt(1,:);
821     Vcomp(1,:) = Vcompt(1,:);
822     Vcomp2(1,:) = Vcomp2t(1,:);
823     Ipces(1,:) = Ipcst(1,:);
824     Ipcer(1,:) = Ipcrt(1,:);
825     Iload(1,:) = Iloadt(1,:);
826     Ir(1,:) = Irt(1,:);
827     Icomp(1,:) = Icompt(1,:);
828     Icomp2(1,:) = Icomp2t(1,:);
829 end
830
831 % ----- ADD OTHER SOLUTIONS TO ARRAYS
832 Vs = [Vst; Vs];
833 for m = 1:Vrows
834     Vpccs(m+1,:) = Vpcst(m,:);
835     VVpccr(m+1,:) = Vpcrt(m,:);
836     Vpcen(m+1,:) = Vpcnt(m,:);
837     Vcomp(m+1,:) = Vcompt(m,:);
838     Vcomp2(m+1,:) = Vcomp2t(m,:);
839     Ipces(m+1,:) = Ipcst(m,:);
840     Ipcer(m+1,:) = Ipcrt(m,:);
841     Iload(m+1,:) = Iloadt(m,:);
842     Ir(m+1,:) = Irt(m,:);
843     Icomp(m+1,:) = Icompt(m,:);
844     Icomp2(m+1,:) = Icomp2t(m,:);
845 end;
846 if isnf(Zr)           % no receiving-end source or impedance
847     Vr = VVpccr;
848 end;
849
850 % ----- ASSIGN ALL SOLUTIONS TO DATA MATRIX
851
852 Data(:,1) = Vs;
853 Data(:,2) = Vpccs;
854 Data(:,3) = VVpccr;
855 Data(:,4) = Vr;
856 Data(:,5) = Vcomp;
857 Data(:,6) = Vcomp2;
858 Data(:,7) = Vpcen;
859 Data(:,8) = Ipces;
860 Data(:,9) = Iload;
861 Data(:,10) = Ir;
862 Data(:,11) = Ipcer;
863 Data(:,12) = Icomp;
864 Data(:,13) = Icomp2;

```

B.2.2 CalcVpcc.m

```

1 function [Vpccrt, Sctot, FailCounter] = CalcVpcc(K3_1,K3_3,K4_1,K4_3,K6_0,K6_2,K6_4,...
2     Vs,Vpccr,VpccrNocomp,N,tol);
3
4 K3 = Vpccr^3*K3_3+Vpccr*K3_1;
5 K4 = Vpccr^3*K4_3+Vpccr*K4_1;
6 K6 = Vpccr^4*K6_4+Vpccr^2*K6_2+K6_0;
7 thetaVpccr(1,:) = atan2(-K3.*sign(K4).*sqrt(K4.^2+K4.^2-K6.^2)-K6.*K4,-K6.*K3...
8     +sqrt(K3.^2+K4.^2-K6.^2).*abs(K4));
9 thetaVpccr(2,:) = atan2(K3.*sign(K4).*sqrt(K4.^2+K4.^2-K6.^2)-K6.*K4,-K6.*K3...
10     -sqrt(K3.^2+K4.^2-K6.^2).*abs(K4));
11 [Vrows Vcols] = size(thetaVpccr);
12 Vpccrt = Vpccr.*exp(j*thetaVpccr); % solve circuit for thetaVpccr(1:2)
13 Test = repmat(K3,Vrows,1).*cos(thetaVpccr)+repmat(K4,Vrows,1).*sin(thetaVpccr)+repmat(K6,Vrows,1); % calculate control values for thetaVpccr
14 thetaValid = abs(Test)<tol; % test if thetaVpccr valid solution
15 thetaValid = and(thetaValid,abs(Vpccrt)>abs(repmat(VpccrNocomp,Vrows,1)));
16 Sctot = ones(Vrows,N); % reset total compensator power variable
17 VpccrSpec = Vpccr;
18 FailCounter = 0;
19 for n = 1:N
20     VpccrTemp(n) = VpccrSpec;
21     if ~thetaValid(:,n)
22         while and(~thetaValid(:,n),VpccrTemp(n)>abs(VpccrNocomp(n))) % repeat until Vpccr low enough for valid solution
23             VpccrTemp(n) = VpccrTemp(n)-VpccrSpec/100;
24             K3(n) = VpccrTemp(n)^3*K3_3(n)+VpccrTemp(n)*K3_1(n);
25             K4(n) = VpccrTemp(n)^3*K4_3(n)+VpccrTemp(n)*K4_1(n);
26             K6(n) = VpccrTemp(n)^4*K6_4(n)+VpccrTemp(n)^2*K6_2(n)+K6_0(n);
27             % solve thetaVpccr with formulae from Solve2ndOrder.mws
28             thetaVpccr(1,n) = atan2(-K3(n).*sign(K4(n)).*sqrt(K3(n).^2+K4(n).^2-K6(n).^2)-K6(n).*K4(n),...
29                 -K6(n).*K3(n)+sqrt(K3(n).^2+K4(n).^2-K6(n).^2).*abs(K4(n)));
30             thetaVpccr(2,n) = atan2(K3(n).*sign(K4(n)).*sqrt(K3(n).^2+K4(n).^2-K6(n).^2)-K6(n).*K4(n),...
31                 -K6(n).*K3(n)-sqrt(K3(n).^2+K4(n).^2-K6(n).^2).*abs(K4(n)));
32             Vpccrt(:,n) = VpccrTemp(n).*exp(j*thetaVpccr(:,n)); % solve circuit for thetaVpccr(1:2)
33             Test(:,n) = repmat(K3(n),Vrows,1).*cos(thetaVpccr(:,n))... % calculate control values for thetaVpccr
34                 +repmat(K4(n),Vrows,1).*sin(thetaVpccr(:,n))+K6(n);
35             thetaValid(:,n) = abs(Test(:,n))<tol; % test if thetaVpccr valid solution
36         end;
37         FailCounter = FailCounter+1;
38     end;
39     if ~thetaValid(:,n) % no valid solution for specific n (delta)
40         Vpccrt(1,n) = VpccrNocomp(n); % set first solution to no compensation voltage
41         Vpccrt(2:Vrows,n) = NaN; % no other solutions
42         Sctot(:,n) = NaN;
43     else
44         for m = 1:Vrows % at least one valid solution for specific n (delta)
45             if ~thetaValid(m,n) % thetaVpccr(m,n) not valid solution
46                 Vpccrt(m,n) = NaN; % eliminate this solution
47                 Sctot(m,n) = NaN;
48             end;
49         end;
50     end
51 end;

```

B.2.3 SolvePQ.m

```

1 function Vpcc = SolvePQ(a,b,c,d,Pload,Qload)
2 % solve Vpcc and thetaVpccs for Pload, Qload
3 % where Vpcc*a*cos(thetaVpcc)-Vpcc*b*sin(thetaVpcc)+Vpcc^2*c-Pload=0
4 % Vpcc*b*cos(thetaVpcc)+Vpcc*a*sin(thetaVpcc)+Vpcc^2*d-Qload=0
5 % formulae from Maple file SolPQ.mws
6
7 C1 = c.^2+d.^2;
8 C3 = -2*Qload.*d-a.^2-2*c.*Pload-b.^2;
9 C5 = Qload.^2+Pload.^2;
10
11 RR(1,:) = 1/2.*sqrt(2).*sqrt(C1.*(-C3+sqrt(C3.^2-4.*C1.*C5)))/C1;
12 RR(2,:) = -1/2.*sqrt(2).*sqrt(C1.*(-C3+sqrt(C3.^2-4.*C1.*C5)))/C1;
13 RR(3,:) = 1/2.*sqrt(-2.*C1.*(C3+sqrt(C3.^2-4.*C1.*C5)))/C1;
14 RR(4,:) = -1/2.*sqrt(-2.*C1.*(C3+sqrt(C3.^2-4.*C1.*C5)))/C1;
15
16 ca = [a; a; a; a];
17 cb = [b; b; b; b];
18 cc = [c; c; c; c];
19 cd = [d; d; d; d];
20 Pload = [Pload;Pload;Pload;Pload];
21 Qload = [Qload;Qload;Qload;Qload];
22
23 Vpcc = RR; % calculate 4 solutions for Vpcc
24 warning off
25 thetaVpcc = atan2((-cb.*RR.^2.*cc+cb.*Pload+RR.^2.*cd.*ca-Qload.*ca)/RR,... % calculate 4 solutions for thetaVpcc
26 (-cb.*RR.^2.*cd-cb.*Qload+RR.^2.*cc.*ca-Pload.*ca)/RR);
27 warning on
28 Vpcc = Vpcc.*exp(j*thetaVpcc);

```

B.2.4 Solve4thOrder.m

```

1 function thetaVpccr = Solve4thOrder(K1,K3,K4,K5)
2 % solve 4th order equation
3 % formulae from Maple file SolTo4thOrderSimple.mws
4
5 C1 = 4.*K1.^2+K5.^2;
6 C2 = 2.*K5.*K4+4.*K3.*K1;
7 C3 = K4.^2-4.*K1.^2+K3.^2-K5.^2;
8 C4 = -2.*K3.*K1-2.*K5.*K4;
9 C5 = K1.^2-K4.^2;
10
11 sub1 = sqrt(3).*sqrt(-C2.^2.*C4.^2.*C3.^2+128.*C5.^2.*C1.^2.*C3.^2-16.*C5.*C1.*C3.^4+4.*C4.^2.*C1.*C3.^3+6.*C2.^2.*C4.^2.*C5.*C1-...
12 144.*C5.^2.*C3.*C1.*C2.^2+27.*C4.^4.*C1.^2+192.*C2.*C4.*C5.^2.*C1.^2-18.*C3.*C2.*C4.^3.*C1-18.*C3.*C2.^3.*C4.*C5-...
13 144.*C5.*C3.*C1.^2.*C4.^2-256.*C5.^3.*C1.^3+80.*C2.*C4.*C5.*C1.*C3.^2+4.*C5.*C2.^2.*C3.^3+4.*C2.^3.*C4.^3+27.*C5.^2.*C2.^4);
14 sub2 = -36.*C3.*C2.*C4-288.*C5.*C3.*C1+108.*C4.^2.*C1+108.*C5.*C2.^2+8.*C3.^3+12.*sub1;
15 sub3 = sqrt(-9.*C2.^2.*sub2.^(1/3)+24.*C3.*C1.*sub2.^(1/3)-6.*sub2.^(2/3).*C1+72.*C4.*C2.*C1-288.*C5.*C1.^2-24.*C1.*C3.^2)/(sub2.^(1/3));
16
17
18 sol(1,:) = -1/4.*C2./C1+1/12.*sub3./C1+1/12.*sqrt(-18.*C2.^2.*sub2.^(1/3).*sub3+48.*C3.*C1.*sub2.^(1/3).*sub3+...
19 6.*sub3.*sub2.^(2/3).*C1-72.*sub3.*C4.*C2.*C1+288.*sub3.*C5.*C1.^2+24.*sub3.*C1.*C3.^2+432.*sub2.^(1/3).*C4.*C1.^2+...
20 54.*sub2.^(1/3).*C2.^3-216.*sub2.^(1/3).*C2.*C3.*C1)/(sub2.^(1/3).*sqrt(-9.*C2.^2.*sub2.^(1/3)+24.*C3.*C1.*sub2.^(1/3)-...
21 6.*sub2.^(2/3).*C1+72.*C4.*C2.*C1-288.*C5.*C1.^2-24.*C1.*C3.^2)/(sub2.^(1/3)))/C1;
22 sol(2,:) = -1/4.*C2./C1+1/12.*sub3./C1-1/12.*sqrt(-18.*C2.^2.*sub2.^(1/3).*sub3+48.*C3.*C1.*sub2.^(1/3).*sub3+...
23 6.*sub3.*sub2.^(2/3).*C1-72.*sub3.*C4.*C2.*C1+288.*sub3.*C5.*C1.^2+24.*sub3.*C1.*C3.^2+432.*sub2.^(1/3).*C4.*C1.^2+...
24 54.*sub2.^(1/3).*C2.^3-216.*sub2.^(1/3).*C2.*C3.*C1)/(sub2.^(1/3).*sqrt(-9.*C2.^2.*sub2.^(1/3)+24.*C3.*C1.*sub2.^(1/3)-...
25 6.*sub2.^(2/3).*C1+72.*C4.*C2.*C1-288.*C5.*C1.^2-24.*C1.*C3.^2)/(sub2.^(1/3)))/C1;
26 sol(3,:) = -1/4.*C2./C1-1/12.*sub3./C1+1/12.*sqrt(6).*sqrt(3.*C2.^2.*sub2.^(1/3).*sub3-8.*C3.*C1.*sub2.^(1/3).*sub3-...
27 sub3.*sub2.^(2/3).*C1+12.*sub3.*C4.*C2.*C1-48.*sub3.*C5.*C1.^2-4.*sub3.*C1.*C3.^2+72.*sub2.^(1/3).*C4.*C1.^2+...
28 9.*sub2.^(1/3).*C2.^3-36.*sub2.^(1/3).*C2.*C3.*C1)/(sub2.^(1/3).*sqrt(-9.*C2.^2.*sub2.^(1/3)+24.*C3.*C1.*sub2.^(1/3)-...
29 6.*sub2.^(2/3).*C1+72.*C4.*C2.*C1-288.*C5.*C1.^2-24.*C1.*C3.^2)/(sub2.^(1/3)))/C1;
30 sol(4,:) = -1/4.*C2./C1-1/12.*sub3./C1-1/12.*sqrt(6).*sqrt(3.*C2.^2.*sub2.^(1/3).*sub3-8.*C3.*C1.*sub2.^(1/3).*sub3-...
31 sub3.*sub2.^(2/3).*C1+12.*sub3.*C4.*C2.*C1-48.*sub3.*C5.*C1.^2-4.*sub3.*C1.*C3.^2+72.*sub2.^(1/3).*C4.*C1.^2+...
32 9.*sub2.^(1/3).*C2.^3-36.*sub2.^(1/3).*C2.*C3.*C1)/(sub2.^(1/3).*sqrt(-9.*C2.^2.*sub2.^(1/3)+24.*C3.*C1.*sub2.^(1/3)-...
33 6.*sub2.^(2/3).*C1+72.*C4.*C2.*C1-288.*C5.*C1.^2-24.*C1.*C3.^2)/(sub2.^(1/3)))/C1;
34
35 K1 = [K1; K1; K1; K1];
36 K3 = [K3; K3; K3; K3];
37 K4 = [K4; K4; K4; K4];
38 K5 = [K5; K5; K5; K5];
39
40 thetaVpccr = atan2(-(2.*K1.*sol.^2+sol.*K3-K1)/(sol.*K5+K4),sol);

```

B.2.5 No.m

```
1 function h = no
2 % This function was created by saveas(...,'mfig'), or print -dmfile.
3 % It loads an HG object saved in binary format in the FIG-file of the
4 % same name. NOTE: if you want to see the old-style M code
5 % representation of a saved object, previously created by print -dmfile,
6 % you can obtain this by using saveas(...,'mmat'). But be advised that the
7 % M-file/MAT-file format does not preserve some important information due
8 % to limitations in that format, including ApplicationData stored using
9 % setappdata. Also, references to handles stored in UserData or Application-
10 % Data will no longer be valid if saved in the M-file/MAT-file format.
11 % ApplicationData and stored handles (excluding integer figure handles)
12 % are both correctly saved in FIG-files.
13 %
14 %load the saved object
15 open('NO_Circuit.fig')
16 open('NO_VReg.fig')
17 open('NO_VStab.fig')
18 figure(findobj('Tag','CircuitFigure'))
19 if get(findobj(findobj('Tag','CircuitFigure'),'Tag','HighResFiguresCheck'),'Value')
20     set(findobj(findobj('Tag','CircuitFigure'),'Tag','HighResFiguresCheck'),'Value',0)
21 end;
22 if nargin > 0, h =(gcf), end
```

B.2.6 NoGui.m

```

1 function NOgui(action)
2 currentGUI = get(gcf,'Tag');
3 switch(action)
4     case 'No__Device'
5         currentFig = gcf;
6         figure(findobj('Tag','CircuitFigure'))
7         image(imread('comp_none.bmp','bmp'))
8         set(gca,'XTick',[],...
9             'YTick',[])
10        figure(currentFig)
11        set(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData',action);
12        set(findobj('Tag','NoDeviceRadio'),'Value',1)
13        set(findobj('Tag','ShuntRadio'),'Value',0)
14        set(findobj('Tag','SeriesRadio'),'Value',0)
15        set(findobj('Tag','SeriesShuntRadio'),'Value',0)
16        set(findobj('Tag','InLineRadio'),'Value',0)
17        set(findobj(findobj('Tag','CircuitFigure'),'Tag','PassiveRadio'),'Enable','off')
18        set(findobj(findobj('Tag','CircuitFigure'),'Tag','MechanicalRadio'),'Enable','off')
19        set(findobj(findobj('Tag','CircuitFigure'),'Tag','ThyristorRadio'),'Enable','off')
20        set(findobj(findobj('Tag','CircuitFigure'),'Tag','TransistorRadio'),'Enable','off')
21        set(findobj('Tag','EnergyStoreNone'),'Enable','off')
22        set(findobj('Tag','EnergyStore__PQ'),'Enable','off')
23        set(findobj('Tag','EnergyStorePcmp'),'Enable','off')
24        set(findobj('Tag','EnergyStoreQcmp'),'Enable','off')
25        set(findobj('Tag','EnergyStoreMinS'),'Enable','off')
26        set(findobj('Tag','EnergyStoreVpcc'),'Enable','off')
27        set(findobj('Tag','UserInput1Text'),'Enable','off');
28        set(findobj('Tag','UserInput1Edit'),'Enable','off');
29        set(findobj('Tag','UserInput2Text'),'Enable','off');
30        set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
31        set(findobj('Tag','UserInput2Edit'),'Enable','off');
32        set(findobj('Tag','UserInput3Text'),'Enable','off');
33        set(findobj('Tag','UserInput3Edit'),'Enable','off');
34        set(findobj('Tag','UserInput4Text'),'Enable','off');
35        set(findobj('Tag','UserInput4Edit'),'Enable','off');
36        set(findobj('Tag','UserInput5Text'),'Enable','off');
37        set(findobj('Tag','UserInput5Edit'),'Enable','off');
38    case 'ShuntDevice'
39        currentFig = gcf;
40        figure(findobj('Tag','CircuitFigure'))
41        image(imread('comp_shunt.bmp','bmp'))
42        set(gca,'XTick',[],...
43            'YTick',[])
44        figure(currentFig)
45        set(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData',action);
46        set(findobj('Tag','NoDeviceRadio'),'Value',0)
47        set(findobj('Tag','ShuntRadio'),'Value',1)
48        set(findobj('Tag','SeriesRadio'),'Value',0)
49        set(findobj('Tag','SeriesShuntRadio'),'Value',0)
50        set(findobj('Tag','InLineRadio'),'Value',0)
51        set(findobj(findobj('Tag','CircuitFigure'),'Tag','PassiveRadio'),'Enable','off')
52        set(findobj(findobj('Tag','CircuitFigure'),'Tag','MechanicalRadio'),'Enable','off')
53        set(findobj(findobj('Tag','CircuitFigure'),'Tag','ThyristorRadio'),'Enable','off')
54        set(findobj(findobj('Tag','CircuitFigure'),'Tag','TransistorRadio'),'Enable','off')
55        set(findobj('Tag','EnergyStoreNone'),'Enable','on')
56        set(findobj('Tag','EnergyStore__PQ'),'Enable','on')
57        set(findobj('Tag','EnergyStorePcmp'),'Enable','on')
58        set(findobj('Tag','EnergyStoreQcmp'),'Enable','on')
59        set(findobj('Tag','EnergyStoreMinS'),'Enable','on')
60        set(findobj('Tag','EnergyStoreVpcc'),'Enable','on')
61        set(findobj('Tag','UserInput1Text'),'Enable','on');
62        set(findobj('Tag','UserInput1Edit'),'Enable','on');
63        switch(get(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData'))
64            case 'EnergyStore__PQ'

```

ADDENDUM B PROGRAM LISTINGS

```

65     set(findobj('Tag','UserInput2Text'),'Enable','off');
66     set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
67     set(findobj('Tag','UserInput2Edit'),'Enable','off');
68     set(findobj('Tag','UserInput3Text'),'Enable','on',...
69         'String','comp:P/Q');
70     set(findobj('Tag','UserInput3Edit'),'Enable','on');
71 case 'EnergyStorePcomp'
72     set(findobj('Tag','UserInput2Text'),'Enable','off');
73     set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
74     set(findobj('Tag','UserInput2Edit'),'Enable','off');
75     set(findobj('Tag','UserInput3Text'),'Enable','on',...
76         'String','Pcomp');
77     set(findobj('Tag','UserInput3Edit'),'Enable','on');
78 case 'EnergyStoreQcomp'
79     set(findobj('Tag','UserInput2Text'),'Enable','off');
80     set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
81     set(findobj('Tag','UserInput2Edit'),'Enable','off');
82     set(findobj('Tag','UserInput3Text'),'Enable','on',...
83         'String','Qcomp');
84     set(findobj('Tag','UserInput3Edit'),'Enable','on');
85 case 'EnergyStoreVpcc'
86     set(findobj('Tag','UserInput2Text'),'Enable','on');
87     set(findobj('Tag','UserInput2TextAngle'),'Enable','on');
88     set(findobj('Tag','UserInput2Edit'),'Enable','on');
89     set(findobj('Tag','UserInput3Text'),'Enable','off');
90     set(findobj('Tag','UserInput3Edit'),'Enable','off');
91 otherwise
92     set(findobj('Tag','UserInput2Text'),'Enable','off');
93     set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
94     set(findobj('Tag','UserInput2Edit'),'Enable','off');
95     set(findobj('Tag','UserInput3Text'),'Enable','off');
96     set(findobj('Tag','UserInput3Edit'),'Enable','off');
97 end;
98 set(findobj('Tag','UserInput4Text'),'Enable','off');
99 set(findobj('Tag','UserInput4Edit'),'Enable','off');
100 set(findobj('Tag','UserInput5Text'),'Enable','off');
101 set(findobj('Tag','UserInput5Edit'),'Enable','off');
102 case 'SerieDevice'
103     currentFig =(gcf);
104     figure(findobj('Tag','CircuitFigure'))
105     image(imread('comp_series.bmp','bmp'))
106     set(gca,'XTick',[],...
107         'YTick',[])
108     figure(currentFig)
109     set(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData',action);
110     set(findobj('Tag','NoDeviceRadio'),'Value',0)
111     set(findobj('Tag','ShuntRadio'),'Value',0)
112     set(findobj('Tag','SeriesRadio'),'Value',1)
113     set(findobj('Tag','SeriesShuntRadio'),'Value',0)
114     set(findobj('Tag','InLineRadio'),'Value',0)
115     set(findobj(findobj('Tag','CircuitFigure'),'Tag','PassiveRadio'),'Enable','off)
116     set(findobj(findobj('Tag','CircuitFigure'),'Tag','MechanicalRadio'),'Enable','off)
117     set(findobj(findobj('Tag','CircuitFigure'),'Tag','ThyristorRadio'),'Enable','off)
118     set(findobj(findobj('Tag','CircuitFigure'),'Tag','TransistorRadio'),'Enable','off)
119     set(findobj('Tag','EnergyStoreNone'),'Enable','on)
120     set(findobj('Tag','EnergyStore__PQ'),'Enable','on)
121     set(findobj('Tag','EnergyStorePcomp'),'Enable','on)
122     set(findobj('Tag','EnergyStoreQcomp'),'Enable','on)
123     set(findobj('Tag','EnergyStoreMinS'),'Enable','on)
124     set(findobj('Tag','EnergyStoreVpcc'),'Enable','on)
125     set(findobj('Tag','UserInput1Text'),'Enable','on');
126     set(findobj('Tag','UserInput1Edit'),'Enable','on');
127     switch(get(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData'))
128     case 'EnergyStore__PQ'
129         set(findobj('Tag','UserInput2Text'),'Enable','off');
130         set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
131         set(findobj('Tag','UserInput2Edit'),'Enable','off');

```


ADDENDUM B PROGRAM LISTINGS

```

132         set(findobj('Tag','UserInput3Text'),'Enable','on',...
133             'String','comp:P/Q),
134         set(findobj('Tag','UserInput3Edit'),'Enable','on');
135     case 'EnergyStorePcomp'
136         set(findobj('Tag','UserInput2Text'),'Enable','off');
137         set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
138         set(findobj('Tag','UserInput2Edit'),'Enable','off');
139         set(findobj('Tag','UserInput3Text'),'Enable','on',...
140             'String','Pcomp);
141         set(findobj('Tag','UserInput3Edit'),'Enable','on');
142     case 'EnergyStoreQcomp'
143         set(findobj('Tag','UserInput2Text'),'Enable','off');
144         set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
145         set(findobj('Tag','UserInput2Edit'),'Enable','off');
146         set(findobj('Tag','UserInput3Text'),'Enable','on',...
147             'String','Qcomp);
148         set(findobj('Tag','UserInput3Edit'),'Enable','on');
149     case 'EnergyStoreVpcc'
150         set(findobj('Tag','UserInput2Text'),'Enable','on');
151         set(findobj('Tag','UserInput2TextAngle'),'Enable','on');
152         set(findobj('Tag','UserInput2Edit'),'Enable','on');
153         set(findobj('Tag','UserInput3Text'),'Enable','off');
154         set(findobj('Tag','UserInput3Edit'),'Enable','off');
155     otherwise
156         set(findobj('Tag','UserInput2Text'),'Enable','off');
157         set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
158         set(findobj('Tag','UserInput2Edit'),'Enable','off');
159         set(findobj('Tag','UserInput3Text'),'Enable','off');
160         set(findobj('Tag','UserInput3Edit'),'Enable','off');
161     end;
162     set(findobj('Tag','UserInput4Text'),'Enable','off');
163     set(findobj('Tag','UserInput4Edit'),'Enable','off');
164     set(findobj('Tag','UserInput5Text'),'Enable','off');
165     set(findobj('Tag','UserInput5Edit'),'Enable','off');
166 case 'SeriesShunt'
167     currentFig = gcf;
168     figure(findobj('Tag','CircuitFigure'))
169     image(imread('comp_seresshunt.bmp','bmp'))
170     set(gca,'XTick',[],...
171         'YTick',[])
172     figure(currentFig)
173     set(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData',action);
174     set(findobj('Tag','NoDeviceRadio'),'Value',0)
175     set(findobj('Tag','ShuntRadio'),'Value',0)
176     set(findobj('Tag','SeriesRadio'),'Value',0)
177     set(findobj('Tag','SeriesShuntRadio'),'Value',1)
178     set(findobj('Tag','InLineRadio'),'Value',0)
179     set(findobj(findobj('Tag','CircuitFigure'),'Tag','PassiveRadio'),'Enable','off)
180     set(findobj(findobj('Tag','CircuitFigure'),'Tag','MechanicalRadio'),'Enable','off)
181     set(findobj(findobj('Tag','CircuitFigure'),'Tag','ThyristorRadio'),'Enable','off)
182     set(findobj(findobj('Tag','CircuitFigure'),'Tag','TransistorRadio'),'Enable','off)
183     set(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData','EnergyStoreNone');
184     set(findobj('Tag','EnergyStoreNone'),'Enable','on',...
185         'Value',1)
186     set(findobj('Tag','EnergyStore__PQ'),'Enable','off',...
187         'Value',0)
188     set(findobj('Tag','EnergyStorePcomp'),'Enable','off',...
189         'Value',0)
190     set(findobj('Tag','EnergyStoreQcomp'),'Enable','off',...
191         'Value',0)
192     set(findobj('Tag','EnergyStoreMinS'),'Enable','off',...
193         'Value',0)
194     set(findobj('Tag','EnergyStoreVpcc'),'Enable','off',...
195         'Value',0)
196     set(findobj('Tag','UserInput1Text'),'Enable','on');
197     set(findobj('Tag','UserInput1Edit'),'Enable','on');
198     set(findobj('Tag','UserInput2Text'),'Enable','on');

```

ADDENDUM B PROGRAM LISTINGS

```

199     set(findobj('Tag','UserInput2TextAngle'),'Enable','on');
200     set(findobj('Tag','UserInput2Edit'),'Enable','on');
201     set(findobj('Tag','UserInput3Text'),'Enable','on',...
202         'String','PF');
203     set(findobj('Tag','UserInput3Edit'),'Enable','on');
204     set(findobj('Tag','UserInput4Text'),'Enable','off');
205     set(findobj('Tag','UserInput4Edit'),'Enable','off');
206     set(findobj('Tag','UserInput5Text'),'Enable','off');
207     set(findobj('Tag','UserInput5Edit'),'Enable','off');
208 case 'InLineDvice'
209     currentFig = gcf;
210     figure(findobj('Tag','CircuitFigure'))
211     image(imread('comp_inline.bmp','bmp'))
212     set(gca,'XTick',[],...
213         'YTick',[])
214     figure(currentFig)
215     set(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData',action);
216     set(findobj('Tag','NoDeviceRadio'),'Value',0)
217     set(findobj('Tag','ShuntRadio'),'Value',0)
218     set(findobj('Tag','SeriesRadio'),'Value',0)
219     set(findobj('Tag','SeriesShuntRadio'),'Value',0)
220     set(findobj('Tag','InLineRadio'),'Value',1)
221     set(findobj(findobj('Tag','CircuitFigure'),'Tag','PassiveRadio'),'Enable','off')
222     set(findobj(findobj('Tag','CircuitFigure'),'Tag','MechanicalRadio'),'Enable','off')
223     set(findobj(findobj('Tag','CircuitFigure'),'Tag','ThyristorRadio'),'Enable','off')
224     set(findobj(findobj('Tag','CircuitFigure'),'Tag','TransistorRadio'),'Enable','off')
225     set(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData','EnergyStoreNone');
226     set(findobj('Tag','EnergyStoreNone'),'Enable','on',...
227         'Value',1)
228     set(findobj('Tag','EnergyStore__PQ'),'Enable','off',...
229         'Value',0)
230     set(findobj('Tag','EnergyStorePcmp'),'Enable','off',...
231         'Value',0)
232     set(findobj('Tag','EnergyStoreQcmp'),'Enable','off',...
233         'Value',0)
234     set(findobj('Tag','EnergyStoreMinS'),'Enable','off',...
235         'Value',0)
236     set(findobj('Tag','EnergyStoreVpcc'),'Enable','off',...
237         'Value',0)
238     set(findobj('Tag','UserInput1Text'),'Enable','on');
239     set(findobj('Tag','UserInput1Edit'),'Enable','on');
240     set(findobj('Tag','UserInput2Text'),'Enable','on');
241     set(findobj('Tag','UserInput2TextAngle'),'Enable','on');
242     set(findobj('Tag','UserInput2Edit'),'Enable','on');
243     set(findobj('Tag','UserInput3Text'),'Enable','on',...
244         'String','PF');
245     set(findobj('Tag','UserInput3Edit'),'Enable','on');
246     set(findobj('Tag','UserInput4Text'),'Enable','on',...
247         'String','Vdcr');
248     set(findobj('Tag','UserInput4Edit'),'Enable','on');
249     set(findobj('Tag','UserInput5Text'),'Enable','on',...
250         'String','Rdc');
251     set(findobj('Tag','UserInput5Edit'),'Enable','on');
252 case 'EnergyStoreNone'
253     set(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData',action);
254     set(findobj('Tag','EnergyStoreNone'),'Value',1)
255     set(findobj('Tag','EnergyStore__PQ'),'Value',0)
256     set(findobj('Tag','EnergyStorePcmp'),'Value',0)
257     set(findobj('Tag','EnergyStoreQcmp'),'Value',0)
258     set(findobj('Tag','EnergyStoreMinS'),'Value',0)
259     set(findobj('Tag','EnergyStoreVpcc'),'Value',0)
260     switch(get(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData'))
261     case 'SeriesShunt'
262     case 'InLineDvice'
263     otherwise
264         set(findobj('Tag','UserInput2Text'),'Enable','off');
265         set(findobj('Tag','UserInput2TextAngle'),'Enable','off');

```

```

266         set(findobj('Tag','UserInput2Edit'),'Enable','off');
267         set(findobj('Tag','UserInput3Text'),'Enable','off');
268         set(findobj('Tag','UserInput3Edit'),'Enable','off');
269         set(findobj('Tag','UserInput4Text'),'Enable','off');
270         set(findobj('Tag','UserInput4Edit'),'Enable','off');
271         set(findobj('Tag','UserInput5Text'),'Enable','off');
272         set(findobj('Tag','UserInput5Edit'),'Enable','off');
273     end;
274 case 'EnergyStore__PQ'
275     set(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData',action);
276     set(findobj('Tag','EnergyStoreNone'),'Value',0)
277     set(findobj('Tag','EnergyStore__PQ'),'Value',1)
278     set(findobj('Tag','EnergyStorePcomp'),'Value',0)
279     set(findobj('Tag','EnergyStoreQcomp'),'Value',0)
280     set(findobj('Tag','EnergyStoreMinS'),'Value',0)
281     set(findobj('Tag','EnergyStoreVpcc'),'Value',0)
282     set(findobj('Tag','UserInput2Text'),'Enable','off');
283     set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
284     set(findobj('Tag','UserInput2Edit'),'Enable','off');
285     set(findobj('Tag','UserInput3Text'),'Enable','on',...
286         'String','comp:P/Q');
287     set(findobj('Tag','UserInput3Edit'),'Enable','on');
288     set(findobj('Tag','UserInput4Text'),'Enable','off');
289     set(findobj('Tag','UserInput4Edit'),'Enable','off');
290     set(findobj('Tag','UserInput5Text'),'Enable','off');
291     set(findobj('Tag','UserInput5Edit'),'Enable','off');
292 case 'EnergyStorePcomp'
293     set(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData',action);
294     set(findobj('Tag','EnergyStoreNone'),'Value',0)
295     set(findobj('Tag','EnergyStore__PQ'),'Value',0)
296     set(findobj('Tag','EnergyStorePcomp'),'Value',1)
297     set(findobj('Tag','EnergyStoreQcomp'),'Value',0)
298     set(findobj('Tag','EnergyStoreMinS'),'Value',0)
299     set(findobj('Tag','EnergyStoreVpcc'),'Value',0)
300     set(findobj('Tag','UserInput2Text'),'Enable','off');
301     set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
302     set(findobj('Tag','UserInput2Edit'),'Enable','off');
303     set(findobj('Tag','UserInput3Text'),'Enable','on',...
304         'String','Pcomp');
305     set(findobj('Tag','UserInput3Edit'),'Enable','on');
306     set(findobj('Tag','UserInput4Text'),'Enable','off');
307     set(findobj('Tag','UserInput4Edit'),'Enable','off');
308     set(findobj('Tag','UserInput5Text'),'Enable','off');
309     set(findobj('Tag','UserInput5Edit'),'Enable','off');
310 case 'EnergyStoreQcomp'
311     set(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData',action);
312     set(findobj('Tag','EnergyStoreNone'),'Value',0)
313     set(findobj('Tag','EnergyStore__PQ'),'Value',0)
314     set(findobj('Tag','EnergyStorePcomp'),'Value',0)
315     set(findobj('Tag','EnergyStoreQcomp'),'Value',1)
316     set(findobj('Tag','EnergyStoreMinS'),'Value',0)
317     set(findobj('Tag','EnergyStoreVpcc'),'Value',0)
318     set(findobj('Tag','UserInput2Text'),'Enable','off');
319     set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
320     set(findobj('Tag','UserInput2Edit'),'Enable','off');
321     set(findobj('Tag','UserInput3Text'),'Enable','on',...
322         'String','Qcomp');
323     set(findobj('Tag','UserInput3Edit'),'Enable','on');
324     set(findobj('Tag','UserInput4Text'),'Enable','off');
325     set(findobj('Tag','UserInput4Edit'),'Enable','off');
326     set(findobj('Tag','UserInput5Text'),'Enable','off');
327     set(findobj('Tag','UserInput5Edit'),'Enable','off');
328 case 'EnergyStoreMinS'
329     set(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData',action);
330     set(findobj('Tag','EnergyStoreNone'),'Value',0)
331     set(findobj('Tag','EnergyStore__PQ'),'Value',0)
332     set(findobj('Tag','EnergyStorePcomp'),'Value',0)

```

```

333     set(findobj('Tag','EnergyStoreQcmp'),'Value',0)
334     set(findobj('Tag','EnergyStoreMinS'),'Value',1)
335     set(findobj('Tag','EnergyStoreVpcc'),'Value',0)
336     set(findobj('Tag','UserInput2Text'),'Enable','off');
337     set(findobj('Tag','UserInput2TextAngle'),'Enable','off');
338     set(findobj('Tag','UserInput2Edit'),'Enable','off');
339     set(findobj('Tag','UserInput3Text'),'Enable','off');
340     set(findobj('Tag','UserInput3Edit'),'Enable','off');
341     set(findobj('Tag','UserInput4Text'),'Enable','off');
342     set(findobj('Tag','UserInput4Edit'),'Enable','off');
343     set(findobj('Tag','UserInput5Text'),'Enable','off');
344     set(findobj('Tag','UserInput5Edit'),'Enable','off');
345 case 'Energy Store Vpcc'
346     set(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData',action);
347     set(findobj('Tag','EnergyStoreNone'),'Value',0)
348     set(findobj('Tag','EnergyStore__PQ'),'Value',0)
349     set(findobj('Tag','EnergyStorePcmp'),'Value',0)
350     set(findobj('Tag','EnergyStoreQcmp'),'Value',0)
351     set(findobj('Tag','EnergyStoreMinS'),'Value',0)
352     set(findobj('Tag','EnergyStoreVpcc'),'Value',1)
353     set(findobj('Tag','UserInput2Text'),'Enable','on');
354     set(findobj('Tag','UserInput2TextAngle'),'Enable','on');
355     set(findobj('Tag','UserInput2Edit'),'Enable','on');
356     set(findobj('Tag','UserInput3Text'),'Enable','off');
357     set(findobj('Tag','UserInput3Edit'),'Enable','off');
358     set(findobj('Tag','UserInput4Text'),'Enable','off');
359     set(findobj('Tag','UserInput4Edit'),'Enable','off');
360     set(findobj('Tag','UserInput5Text'),'Enable','off');
361     set(findobj('Tag','UserInput5Edit'),'Enable','off');
362 case 'Passive'
363     set(gcbo,'Value',1)
364     set(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData',action);
365     set(findobj(findobj('Tag','CircuitFigure'),'Tag','MechanicalRadio'),'Value',0)
366     set(findobj(findobj('Tag','CircuitFigure'),'Tag','ThyristorRadio'),'Value',0)
367     set(findobj(findobj('Tag','CircuitFigure'),'Tag','TransistorRadio'),'Value',0)
368 case 'Mechanical'
369     set(gcbo,'Value',1)
370     set(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData',action);
371     set(findobj(findobj('Tag','CircuitFigure'),'Tag','PassiveRadio'),'Value',0)
372     set(findobj(findobj('Tag','CircuitFigure'),'Tag','ThyristorRadio'),'Value',0)
373     set(findobj(findobj('Tag','CircuitFigure'),'Tag','TransistorRadio'),'Value',0)
374 case 'Thyristor'
375     set(gcbo,'Value',1)
376     set(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData',action);
377     set(findobj(findobj('Tag','CircuitFigure'),'Tag','PassiveRadio'),'Value',0)
378     set(findobj(findobj('Tag','CircuitFigure'),'Tag','MechanicalRadio'),'Value',0)
379     set(findobj(findobj('Tag','CircuitFigure'),'Tag','TransistorRadio'),'Value',0)
380 case 'Transistor'
381     set(gcbo,'Value',1)
382     set(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData',action);
383     set(findobj(findobj('Tag','CircuitFigure'),'Tag','PassiveRadio'),'Value',0)
384     set(findobj(findobj('Tag','CircuitFigure'),'Tag','MechanicalRadio'),'Value',0)
385     set(findobj(findobj('Tag','CircuitFigure'),'Tag','ThyristorRadio'),'Value',0)
386 case 'Tolerance'
387     set(findobj(findobj('Tag','CircuitFigure'),'Tag','ToleranceEdit'),'String',... % erase input on error
388         num2str(str2num(get(gcbo,'String'))))
389 case 'Vs'
390     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
391     set(findobj('Tag','RunPushbutton'),'Enable','on');
392 case 'Zs'
393     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
394 case 'Zr'
395     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
396 case 'Vr'
397     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
398 case 'Iload'
399     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error

```

ADDENDUM B PROGRAM LISTINGS

```

400 case 'Zload'
401     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
402 case 'Vload'
403     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
404 case 'VZload'
405     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
406 case 'Zload'
407     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
408 case 'Pload'
409     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
410 case 'Qload'
411     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
412 case 'UserInput1'
413     set(findobj('Tag','UserInput1Edit'),'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
414 case 'UserInput2'
415     set(findobj('Tag','UserInput2Edit'),'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
416 case 'UserInput3'
417     set(findobj('Tag','UserInput3Edit'),'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
418 case 'UserInput4'
419     set(findobj('Tag','UserInput4Edit'),'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
420 case 'UserInput5'
421     set(findobj('Tag','UserInput5Edit'),'String',num2str(str2num(get(gcbo,'String')))) % erase input on error
422 case 'reset'
423     set(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData','No ___ Device');
424     set(findobj('Tag','NoDeviceRadio'),'Value',1,...
425         'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588]);
426     set(findobj('Tag','ShuntRadio'),'Value',0,...
427         'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588]);
428     set(findobj('Tag','SeriesRadio'),'Value',0,...
429         'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588]);
430     set(findobj('Tag','SeriesShuntRadio'),'Value',0,...
431         'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588]);
432     set(findobj('Tag','InLineRadio'),'Value',0,...
433         'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588]);
434     set(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData','Transistor');
435     set(findobj(findobj('Tag','CircuitFigure'),'Tag','PassiveRadio'),'Enable','off')
436     set(findobj(findobj('Tag','CircuitFigure'),'Tag','MechanicalRadio'),'Enable','off')
437     set(findobj(findobj('Tag','CircuitFigure'),'Tag','ThyristorRadio'),'Enable','off')
438     set(findobj(findobj('Tag','CircuitFigure'),'Tag','TransistorRadio'),'Enable','off')
439     set(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData','EnergyStoreNone');
440     set(findobj('Tag','EnergyStoreNone'),'Value',1,...
441         'Enable','off')
442     set(findobj('Tag','EnergyStore__PQ'),'Value',0,...
443         'Enable','off')
444     set(findobj('Tag','EnergyStorePcmp'),'Value',0,...
445         'Enable','off')
446     set(findobj('Tag','EnergyStoreQcmp'),'Value',0,...
447         'Enable','off')
448     set(findobj('Tag','EnergyStoreMinS'),'Value',0,...
449         'Enable','off')
450     set(findobj('Tag','EnergyStoreVpcc'),'Value',0,...
451         'Enable','off')
452     set(findobj('Tag','UserInput1Text'),'String','\Vpccr',...
453         'Enable','off');
454     set(findobj('Tag','UserInput1Edit'),'String','',...
455         'Enable','off');
456     set(findobj('Tag','UserInput2TextAngle'),'String','D',...
457         'Enable','off');
458     set(findobj('Tag','UserInput2Text'),'String','\Vpccr',...
459         'Enable','off');
460     set(findobj('Tag','UserInput2Edit'),'String','',...
461         'Enable','off');
462     set(findobj('Tag','UserInput3Text'),'String','PF',...
463         'Enable','off');
464     set(findobj('Tag','UserInput3Edit'),'String','',...
465         'Enable','off');
466     set(findobj('Tag','UserInput4Text'),'Enable','off',...

```

```

467     'String','Vdcrf');
468 set(findobj('Tag','UserInput4Edit'),'String','...',
469     'Enable','off');
470 set(findobj('Tag','UserInput5Text'),'Enable','off',...
471     'String','Rdc');
472 set(findobj('Tag','UserInput5Edit'),'String','...',
473     'Enable','off');
474 set(findobj('Tag','RunPushbutton'),'Enable','off');
475 set(findobj(findobj('Tag','CircuitFigure'),'Tag','VsEdit'),'String','')
476 set(findobj(findobj('Tag','CircuitFigure'),'Tag','ZsEdit'),'String','')
477 set(findobj(findobj('Tag','CircuitFigure'),'Tag','ZrEdit'),'String','')
478 set(findobj(findobj('Tag','CircuitFigure'),'Tag','VrEdit'),'String','')
479 set(findobj(findobj('Tag','CircuitFigure'),'Tag','PloadEdit'),'String','')
480 set(findobj(findobj('Tag','CircuitFigure'),'Tag','QloadEdit'),'String','')
481 set(findobj(findobj('Tag','CircuitFigure'),'Tag','IloadEdit'),'String','')
482 set(findobj(findobj('Tag','CircuitFigure'),'Tag','IZloadEdit'),'String','')
483 set(findobj(findobj('Tag','CircuitFigure'),'Tag','VloadEdit'),'String','')
484 set(findobj(findobj('Tag','CircuitFigure'),'Tag','VZloadEdit'),'String','')
485 set(findobj(findobj('Tag','CircuitFigure'),'Tag','ZloadEdit'),'String','')
486 set(findobj(findobj('Tag','CircuitFigure'),'Tag','ToleranceEdit'),'String','num2str(str2num('1e-6'))')
487 if get(findobj(findobj('Tag','CircuitFigure'),'Tag','HighResFiguresCheck'),'Value')
488     set(findobj(findobj('Tag','CircuitFigure'),'Tag','HighResFiguresCheck'),'Value',0)
489     fig = findobj('Tag','3dMagFigure');
490     if ~isempty(fig)
491         close.figure(fig);
492     end;
493     fig = findobj('Tag','3dAngFigure');
494     if ~isempty(fig)
495         close.figure(fig);
496     end;
497     fig = findobj('Tag','PosFigure');
498     if ~isempty(fig)
499         close.figure(fig);
500     end;
501     fig = findobj('Tag','AngleFigure');
502     if ~isempty(fig)
503         close.figure(fig);
504     end;
505     fig = findobj('Tag','LoadFigure');
506     if ~isempty(fig)
507         close.figure(fig);
508     end;
509     fig = findobj('Tag','CompFigure');
510     if ~isempty(fig)
511         close.figure(fig);
512     end;
513     fig = findobj('Tag','VectorFigure');
514     if ~isempty(fig)
515         close.figure(fig);
516     end;
517 end;
518 figure(findobj('Tag','CircuitFigure'))
519 image(imread('comp_none.bmp','bmp'))
520 set(gca,'XTick',[],...
521     'YTick',[])
522 case 'close'
523 if get(findobj(findobj('Tag','CircuitFigure'),'Tag','HighResFiguresCheck'),'Value')
524     fig = findobj('Tag','3dMagFigure');
525     if ~isempty(fig)
526         close.figure(fig);
527     end;
528     fig = findobj('Tag','3dAngFigure');
529     if ~isempty(fig)
530         close.figure(fig);
531     end;
532     fig = findobj('Tag','PosFigure');
533     if ~isempty(fig)

```

```

534         close(figure(fig));
535     end;
536     fig = findobj('Tag','AngleFigure');
537     if ~isempty(fig)
538         close(figure(fig));
539     end;
540     fig = findobj('Tag','LoadFigure');
541     if ~isempty(fig)
542         close(figure(fig));
543     end;
544     fig = findobj('Tag','CompFigure');
545     if ~isempty(fig)
546         close(figure(fig));
547     end;
548     fig = findobj('Tag','VectorFigure');
549     if ~isempty(fig)
550         close(figure(fig));
551     end;
552 end;
553 fig = findobj('Tag','CircuitFigure');
554 saveas(figure(fig).get(fig,'FileName'),'fig')
555 close(figure(fig));
556 fig = findobj('Tag','VRegFigure');
557 saveas(figure(fig).get(fig,'FileName'),'fig')
558 close(figure(fig));
559 fig = findobj('Tag','VStabFigure');
560 saveas(figure(fig).get(fig,'FileName'),'fig')
561 close(figure(fig));
562 end
563
564 switch(action)
565     case('run')
566         VoltRegGUI(action);
567         VoltStabGUI(action);
568     case('reset')
569         VoltRegGUI(action);
570         VoltStabGUI(action);
571         figure(findobj('Tag','CircuitFigure'))
572     case('Vs') % Accounts for case when action is 'Vs' in CircuitFigure
573         VoltRegGUI(action);
574         VoltStabGUI(action);
575     case 'HighResFigures' % Accounts for change in 'HighResFiguresCheck' in CircuitFigure
576         VoltRegGUI(action);
577         VoltStabGUI(action);
578     otherwise
579         switch(currentGUI)
580             case('VRegFigure')
581                 VoltRegGUI(action);
582             case('VStabFigure')
583                 VoltStabGUI(action);
584         end
585     end
586
587 set(findobj(findobj('Tag','CircuitFigure'),'Tag','VsEdit'),'Visible','on'); % avoid object property 'Visible'='off' when 'Value' = 0
588 set(findobj(findobj('Tag','CircuitFigure'),'Tag','ZsEdit'),'Visible','on');
589 set(findobj(findobj('Tag','CircuitFigure'),'Tag','ZrEdit'),'Visible','on');
590 set(findobj(findobj('Tag','CircuitFigure'),'Tag','VrEdit'),'Visible','on');
591 set(findobj(findobj('Tag','CircuitFigure'),'Tag','IloadEdit'),'Visible','on');
592 set(findobj(findobj('Tag','CircuitFigure'),'Tag','LZloadEdit'),'Visible','on');
593 set(findobj(findobj('Tag','CircuitFigure'),'Tag','VloadEdit'),'Visible','on');
594 set(findobj(findobj('Tag','CircuitFigure'),'Tag','VZloadEdit'),'Visible','on');
595 set(findobj(findobj('Tag','CircuitFigure'),'Tag','ZloadEdit'),'Visible','on');
596 set(findobj(findobj('Tag','CircuitFigure'),'Tag','PloadEdit'),'Visible','on');
597 set(findobj(findobj('Tag','CircuitFigure'),'Tag','QloadEdit'),'Visible','on');
598 set(findobj('Tag','UserInput1Edit'),'Visible','on');
599 set(findobj('Tag','UserInput2Edit'),'Visible','on');
600 set(findobj('Tag','UserInput3Edit'),'Visible','on');

```

ADDENDUM B PROGRAM LISTINGS

```
601 set(findobj('Tag','UserInput4Edit'),'Visible','on');
602 set(findobj('Tag','UserInput5Edit'),'Visible','on');
```


B.2.7 VoltregGui.m

```

1 function VoltRegGUI(action)
2 switch(action)
3     case 'No__Device'
4         set(findobj(findobj('Tag','VRegFigure'),'Tag','SolutionORadio'),'Enable','on');
5         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','off');
6         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','off');
7         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off');
8         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off');
9         set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp1Radio'),'Enable','off',...
10             'Value',0)
11         set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp2Radio'),'Enable','off',...
12             'Value',0)
13         set(findobj(findobj('Tag','VRegFigure'),'Tag','LoadRadio'),'Value',1)
14         set(findobj(findobj('Tag','VRegFigure'),'Tag','CompLoadFrame'),'UserData','Load ')
15     case 'ShuntDevice'
16         set(findobj(findobj('Tag','VRegFigure'),'Tag','SolutionORadio'),'Enable','on');
17         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','off');
18         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','off');
19         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off');
20         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off');
21         if get(findobj(findobj('Tag','VRegFigure'),'Tag','Comp1Radio'),'Value')
22             set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp1Radio'),'Enable','off',...
23                 'Value',0)
24             set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp2Radio'),'Enable','on',...
25                 'Value',1)
26             set(findobj(findobj('Tag','VRegFigure'),'Tag','LoadRadio'),'Value',0)
27             set(findobj(findobj('Tag','VRegFigure'),'Tag','CompLoadFrame'),'UserData','Comp2')
28         else
29             set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp1Radio'),'Enable','off')
30             set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp2Radio'),'Enable','on')
31         end
32     case 'SerieDevice'
33         set(findobj(findobj('Tag','VRegFigure'),'Tag','SolutionORadio'),'Enable','on');
34         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','off');
35         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','off');
36         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off');
37         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off');
38         if get(findobj(findobj('Tag','VRegFigure'),'Tag','Comp2Radio'),'Value')
39             set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp2Radio'),'Enable','off',...
40                 'Value',0)
41             set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp1Radio'),'Enable','on',...
42                 'Value',1)
43             set(findobj(findobj('Tag','VRegFigure'),'Tag','LoadRadio'),'Value',0)
44             set(findobj(findobj('Tag','VRegFigure'),'Tag','CompLoadFrame'),'UserData','Comp1')
45         else
46             set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp1Radio'),'Enable','on')
47             set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp2Radio'),'Enable','off')
48             set(findobj(findobj('Tag','VRegFigure'),'Tag','CompLoadFrame'),'UserData','Comp1')
49         end
50     case 'SeriesShunt'
51         set(findobj(findobj('Tag','VRegFigure'),'Tag','SolutionORadio'),'Enable','on');
52         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','off');
53         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','off');
54         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off');
55         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off');
56         set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp1Radio'),'Enable','on')
57         set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp2Radio'),'Enable','on')
58         set(findobj(findobj('Tag','VRegFigure'),'Tag','LoadRadio'),'Enable','on')
59     case 'InLineDvice'
60         set(findobj(findobj('Tag','VRegFigure'),'Tag','SolutionORadio'),'Enable','on');
61         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','off');
62         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','off');
63         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off');
64         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off');

```

ADDENDUM B PROGRAM LISTINGS

```

65     set(findobj(findobj('Tag','VRegFigure'),Tag,'Comp1Radio'),'Enable','on')
66     set(findobj(findobj('Tag','VRegFigure'),Tag,'Comp2Radio'),'Enable','on')
67     set(findobj(findobj('Tag','VRegFigure'),Tag,'LoadRadio'),'Enable','on')
68 case 'EnergyStoreNone'
69     set(findobj(findobj('Tag','VRegFigure'),Tag,'SolutionORadio'),'Enable','on');
70     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution1Radio'),'Enable','off');
71     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution2Radio'),'Enable','off');
72     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution3Radio'),'Enable','off');
73     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution4Radio'),'Enable','off');
74 case 'EnergyStore__PQ'
75     set(findobj(findobj('Tag','VRegFigure'),Tag,'SolutionORadio'),'Enable','on');
76     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution1Radio'),'Enable','off');
77     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution2Radio'),'Enable','off');
78     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution3Radio'),'Enable','off');
79     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution4Radio'),'Enable','off');
80 case 'EnergyStorePcmp'
81     set(findobj(findobj('Tag','VRegFigure'),Tag,'SolutionORadio'),'Enable','on');
82     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution1Radio'),'Enable','off');
83     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution2Radio'),'Enable','off');
84     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution3Radio'),'Enable','off');
85     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution4Radio'),'Enable','off');
86 case 'EnergyStoreQcmp'
87     set(findobj(findobj('Tag','VRegFigure'),Tag,'SolutionORadio'),'Enable','on');
88     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution1Radio'),'Enable','off');
89     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution2Radio'),'Enable','off');
90     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution3Radio'),'Enable','off');
91     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution4Radio'),'Enable','off');
92 case 'EnergyStoreMinS'
93     set(findobj(findobj('Tag','VRegFigure'),Tag,'SolutionORadio'),'Enable','on');
94     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution1Radio'),'Enable','off');
95     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution2Radio'),'Enable','off');
96     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution3Radio'),'Enable','off');
97     set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution4Radio'),'Enable','off');
98 case 'EnergyStoreVpcc'
99     set(findobj(findobj('Tag','VRegFigure'),Tag,'SolutionORadio'),'Enable','on');
100    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution1Radio'),'Enable','off');
101    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution2Radio'),'Enable','off');
102    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution3Radio'),'Enable','off');
103    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution4Radio'),'Enable','off');
104 case 'SolutionO'
105    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution'),'UserData',1);
106    set(findobj(findobj('Tag','VRegFigure'),Tag,'SolutionORadio'),'Value',1)
107    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution1Radio'),'Value',0)
108    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution2Radio'),'Value',0)
109    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution3Radio'),'Value',0)
110    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution4Radio'),'Value',0)
111    PlotsChange;
112 case 'Solution1'
113    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution'),'UserData',2);
114    set(findobj(findobj('Tag','VRegFigure'),Tag,'SolutionORadio'),'Value',0)
115    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution1Radio'),'Value',1)
116    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution2Radio'),'Value',0)
117    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution3Radio'),'Value',0)
118    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution4Radio'),'Value',0)
119    PlotsChange;
120 case 'Solution2'
121    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution'),'UserData',3);
122    set(findobj(findobj('Tag','VRegFigure'),Tag,'SolutionORadio'),'Value',0)
123    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution1Radio'),'Value',0)
124    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution2Radio'),'Value',1)
125    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution3Radio'),'Value',0)
126    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution4Radio'),'Value',0)
127    PlotsChange;
128 case 'Solution3'
129    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution'),'UserData',4);
130    set(findobj(findobj('Tag','VRegFigure'),Tag,'SolutionORadio'),'Value',0)
131    set(findobj(findobj('Tag','VRegFigure'),Tag,'Solution1Radio'),'Value',0)

```

```

132     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Value',0)
133     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Value',1)
134     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Value',0)
135     PlotsChange;
136 case 'Solution4'
137     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData',5);
138     set(findobj(findobj('Tag','VRegFigure'),'Tag','SolutionORadio'),'Value',0)
139     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Value',0)
140     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Value',0)
141     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Value',0)
142     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Value',1)
143     PlotsChange;
144 case 'Voltage'
145     set(gcbo,'Value',1)
146     set(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData',action);
147     set(findobj(findobj('Tag','VRegFigure'),'Tag','CurrentRadio'),'Value',0)
148     set(findobj(findobj('Tag','VRegFigure'),'Tag','PowerRadio'),'Value',0)
149     PlotsChange;
150 case 'Current'
151     set(gcbo,'Value',1)
152     set(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData',action);
153     set(findobj(findobj('Tag','VRegFigure'),'Tag','VoltageRadio'),'Value',0)
154     set(findobj(findobj('Tag','VRegFigure'),'Tag','PowerRadio'),'Value',0)
155     PlotsChange;
156 case 'Power'
157     set(gcbo,'Value',1)
158     set(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData',action);
159     set(findobj(findobj('Tag','VRegFigure'),'Tag','VoltageRadio'),'Value',0)
160     set(findobj(findobj('Tag','VRegFigure'),'Tag','CurrentRadio'),'Value',0)
161     PlotsChange;
162 case 'Vs'
163     set(findobj(findobj('Tag','VRegFigure'),'Tag','VoltageRadio'),'Enable','on');
164     set(findobj(findobj('Tag','VRegFigure'),'Tag','CurrentRadio'),'Enable','on');
165     set(findobj(findobj('Tag','VRegFigure'),'Tag','PowerRadio'),'Enable','on');
166     set(findobj(findobj('Tag','VRegFigure'),'Tag','AngleSlider'),'Enable','on');
167     set(findobj(findobj('Tag','VRegFigure'),'Tag','PositionSlider'),'Enable','on');
168 case 'Comp1Graph'
169     set(gcbo,'Value',1)
170     set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp2Radio'),'Value',0)
171     set(findobj(findobj('Tag','VRegFigure'),'Tag','LoadRadio'),'Value',0)
172     set(findobj(findobj('Tag','VRegFigure'),'Tag','CompLoadFrame'),'UserData','Comp1')
173     P = 1; %deltamax in terms of pi
174     N = 101; %delta divisions
175     DeviceType = get(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData');
176     Technology = get(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData');
177     Screen = get(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData');
178     Data = get(findobj('Tag','CircuitFigure'),'Userdata');
179     Sol = get(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData');
180     Graph2dcomp(DeviceType, Technology, Screen, P, N, 1, Data, Sol);
181 case 'Comp2Graph'
182     set(gcbo,'Value',1)
183     set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp1Radio'),'Value',0)
184     set(findobj(findobj('Tag','VRegFigure'),'Tag','LoadRadio'),'Value',0)
185     set(findobj(findobj('Tag','VRegFigure'),'Tag','CompLoadFrame'),'UserData','Comp2')
186     P = 1; %deltamax in terms of pi
187     N = 101; %delta divisions
188     DeviceType = get(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData');
189     Technology = get(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData');
190     Screen = get(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData');
191     Data = get(findobj('Tag','CircuitFigure'),'Userdata');
192     Sol = get(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData');
193     Graph2dcomp(DeviceType, Technology, Screen, P, N, 2, Data, Sol);
194 case 'LoadGraph'
195     set(gcbo,'Value',1)
196     set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp1Radio'),'Value',0)
197     set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp2Radio'),'Value',0)
198     set(findobj(findobj('Tag','VRegFigure'),'Tag','CompLoadFrame'),'UserData','Load ')

```

ADDENDUM B PROGRAM LISTINGS

```

199     P = 1; %deltamax in terms of pi
200     N = 101; %delta divisions
201     DeviceType = get(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData');
202     Technology = get(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData');
203     Screen = get(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData');
204     Data = get(findobj('Tag','CircuitFigure'),'Userdata');
205     Sol = get(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData');
206     Graph2dload(DeviceType, Technology, Screen, P, N, Data, Sol);
207 case 'PositionSlider'
208     position = round(get(gcbo,'Value'));
209     set(findobj(findobj('Tag','VRegFigure'),'Tag','PositionSlider'),'Value',position)
210     P = 1; %deltamax in terms of pi
211     N = 101; %delta divisions
212     DeviceType = get(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData');
213     Technology = get(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData');
214     Screen = get(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData');
215     delfaref = get(findobj(findobj('Tag','VRegFigure'),'Tag','AngleSlider'),'Value');
216     if delfaref == 0,
217         delfaref = 0.0157;
218     end;
219     delta1 = round(delfaref*N/(P*pi));
220     Data = get(findobj('Tag','CircuitFigure'),'Userdata');
221     Sol = get(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData');
222     Graph2dang(DeviceType, Technology, Screen, delta1, P, N, position, Data, Sol);
223 case 'AngleSlider'
224     delfaref = get(gcbo,'Value');
225     set(findobj(findobj('Tag','VRegFigure'),'Tag','AngleSlider'),'Value',delfaref)
226     P = 1; %deltamax in terms of pi
227     N = 101; %delta divisions
228     DeviceType = get(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData');
229     Technology = get(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData');
230     Screen = get(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData');
231     if delfaref == 0,
232         delfaref = 0.0157;
233     end;
234     delta1 = round(delfaref*N/(P*pi));
235     Vs_Angle = delfaref
236     position = round(get(findobj(findobj('Tag','VRegFigure'),'Tag','PositionSlider'),'Value'));
237     Data = get(findobj('Tag','CircuitFigure'),'Userdata');
238     Sol = get(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData');
239     Graph2dpos(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);
240     Graph2dVector(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);
241 case 'HighResFigures'
242     if get(findobj(findobj('Tag','CircuitFigure'),'Tag','HighResFiguresCheck'),'Value')
243         P = 1; %deltamax in terms of pi
244         N = 101; %delta divisions
245         DeviceType = get(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData');
246         Technology = get(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData');
247         Screen = get(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData');
248         delfaref = get(findobj(findobj('Tag','VRegFigure'),'Tag','AngleSlider'),'Value');
249         if delfaref == 0,
250             delfaref = 0.0157;
251         end;
252         delta1 = round(delfaref*N/(P*pi));
253         position = round(get(findobj(findobj('Tag','VRegFigure'),'Tag','PositionSlider'),'Value'));
254         Data = get(findobj('Tag','CircuitFigure'),'Userdata');
255         Sol = get(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData');
256         switch(get(findobj(findobj('Tag','VRegFigure'),'Tag','CompLoadFrame'),'UserData'))
257             case('Comp1')
258                 Graph2dcomp(DeviceType, Technology, Screen, P, N, 1, Data, Sol);
259             case('Comp2')
260                 Graph2dcomp(DeviceType, Technology, Screen, P, N, 2, Data, Sol);
261             case('Load ')
262                 Graph2dload(DeviceType, Technology, Screen, P, N, Data, Sol);
263         end;
264         Graph2dang(DeviceType, Technology, Screen, delta1, P, N, position, Data, Sol);
265         Graph2dpos(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);

```

ADDENDUM B PROGRAM LISTINGS

```

266     Graph2dVector(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);
267     Graph3d(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);
268 else
269     fig = findobj('Tag','3dMagFigure');
270     if ~isempty(fig)
271         close.figure(fig);
272     end;
273     fig = findobj('Tag','3dAngFigure');
274     if ~isempty(fig)
275         close.figure(fig);
276     end;
277     fig = findobj('Tag','PosFigure');
278     if ~isempty(fig)
279         close.figure(fig);
280     end;
281     fig = findobj('Tag','AngleFigure');
282     if ~isempty(fig)
283         close.figure(fig);
284     end;
285     fig = findobj('Tag','LoadFigure');
286     if ~isempty(fig)
287         close.figure(fig);
288     end;
289     fig = findobj('Tag','CompFigure');
290     if ~isempty(fig)
291         close.figure(fig);
292     end;
293     fig = findobj('Tag','VectorFigure');
294     if ~isempty(fig)
295         close.figure(fig);
296     end;
297 end;
298 case 'run'
299     P = 1; %deltamax in terms of pi
300     N = 101; %delta divisions
301     DeviceType = get(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData');
302     Technology = get(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData');
303     Screen = get(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData');
304     deltaref = get(findobj(findobj('Tag','VRegFigure'),'Tag','AngleSlider'),'Value');
305     if deltaref == 0,
306         deltaref = 0.0157;
307     end;
308     delta1 = round(deltaref*N/(P*pi));
309     Vs_Angle = deltaref
310     position = round(get(findobj(findobj('Tag','VRegFigure'),'Tag','PositionSlider'),'Value'));
311     set(findobj('Tag','NoDeviceRadio'),'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588]);
312     set(findobj('Tag','ShuntRadio'),'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588]);
313     set(findobj('Tag','SeriesRadio'),'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588]);
314     set(findobj('Tag','SeriesShuntRadio'),'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588]);
315     set(findobj('Tag','InLineRadio'),'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588]);
316     switch(DeviceType)
317     case 'No ___ Device' % no compensation
318         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','on');
319         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','off');
320         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off');
321         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off');
322         MaxSol = 2;
323     case 'ShuntDevice' % shunt compensation
324         switch(get(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData'))
325         case 'EnergyStoreVpcc' % shunt compensation with Vpcc magnitude and angle known
326             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','on');
327             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','off');
328             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off');
329             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off');
330             MaxSol = 2;
331         otherwise
332             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','on');

```

ADDENDUM B PROGRAM LISTINGS

```

333         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','on');
334         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off');
335         set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off');
336         MaxSol = 3;
337     end;
338 case 'SerieDevice' % series compensation
339     switch(get(findobj(findobj('Tag','CircuitFigure'),'Tag','EnergyStoreText'),'UserData'))
340         case('EnergyStoreMinS') % series compensation with minimum compensator rating
341             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','on');
342             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','on');
343             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','on');
344             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','on');
345             MaxSol = 5;
346         case('EnergyStoreVpcc') % series compensator for Vpcc magnitude and angle known
347             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','on');
348             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','off');
349             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off');
350             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off');
351             MaxSol = 2;
352         otherwise
353             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','on');
354             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','on');
355             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off');
356             set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off');
357             MaxSol = 3;
358     end;
359 otherwise
360     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','on');
361     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','on');
362     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','on');
363     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','on');
364     MaxSol = 5;
365 end;
366 Sol = get(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData');
367 if Sol > MaxSol
368     set(findobj(findobj('Tag','VRegFigure'),'Tag','SolutionORadio'),'Value',1);
369     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData',1);
370     set(findobj(findobj('Tag','VRegFigure'),'Tag','strcat('Solution',num2str(Sol-1),'Radio')),'Enable','off',...
371         'Value',0);
372     set(findobj(findobj('Tag','VRegFigure'),'Tag','strcat('Solution',num2str(Sol-1),'Radio3')),'Enable','off',...
373         'Value',0);
374 end
375 Data = VoltReg(DeviceType, Technology, Screen, delta1, P, N);
376 set(findobj('Tag','CircuitFigure'),'Userdata',Data);
377 Sol = get(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData');
378 switch(get(findobj(findobj('Tag','VRegFigure'),'Tag','ComplLoadFrame'),'UserData'))
379     case('Comp1')
380         Graph2dcomp(DeviceType, Technology, Screen, P, N, 1, Data, Sol);
381     case('Comp2')
382         Graph2dcomp(DeviceType, Technology, Screen, P, N, 2, Data, Sol);
383     case('Load ')
384         Graph2dload(DeviceType, Technology, Screen, P, N, Data, Sol);
385 end;
386 Graph2dang(DeviceType, Technology, Screen, delta1, P, N, position, Data, Sol);
387 Graph2dpos(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);
388 Graph2dVector(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);
389 Graph3d(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);
390 case 'reset'
391     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData',1);
392     set(findobj(findobj('Tag','VRegFigure'),'Tag','SolutionORadio'),'Enable','on',...
393         'Value',1);
394     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution1Radio'),'Enable','off',...
395         'Value',0);
396     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution2Radio'),'Enable','off',...
397         'Value',0);
398     set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution3Radio'),'Enable','off',...
399         'Value',0);

```

ADDENDUM B PROGRAM LISTINGS

```

400 set(findobj(findobj('Tag','VRegFigure'),'Tag','Solution4Radio'),'Enable','off',...
401     'Value',0);
402 set(findobj(findobj('Tag','VRegFigure'),'Tag','VoltageRadio'),'Enable','off');
403 set(findobj(findobj('Tag','VRegFigure'),'Tag','CurrentRadio'),'Enable','off');
404 set(findobj(findobj('Tag','VRegFigure'),'Tag','PowerRadio'),'Enable','off');
405 set(findobj(findobj('Tag','VRegFigure'),'Tag','AngleSlider'),'Enable','off');
406 set(findobj(findobj('Tag','VRegFigure'),'Tag','PositionSlider'),'Enable','off');
407 set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp1Radio'),'Enable','off',...
408     'Value',0)
409 set(findobj(findobj('Tag','VRegFigure'),'Tag','Comp2Radio'),'Enable','off',...
410     'Value',0)
411 set(findobj(findobj('Tag','VRegFigure'),'Tag','LoadRadio'),'Value',1)
412 set(findobj(findobj('Tag','VRegFigure'),'Tag','CompLoadFrame'),'UserData','Load ')
413 set(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData','Voltage');
414 set(findobj(findobj('Tag','VRegFigure'),'Tag','VoltageRadio'),'Value',1)
415 set(findobj(findobj('Tag','VRegFigure'),'Tag','CurrentRadio'),'Value',0)
416 set(findobj(findobj('Tag','VRegFigure'),'Tag','PowerRadio'),'Value',0)
417 set(findobj(findobj('Tag','VRegFigure'),'Tag','AngleSlider'),'Value',0);
418 set(findobj(findobj('Tag','VRegFigure'),'Tag','PositionSlider'),'Value',0);
419 axes(findobj(findobj('Tag','VRegFigure'),'Tag','DeviceAxes'))
420 cla reset
421 set(gca,'Tag','DeviceAxes')
422 axes(findobj(findobj('Tag','VRegFigure'),'Tag','AngleAxes'))
423 cla reset
424 set(gca,'Tag','AngleAxes')
425 axes(findobj(findobj('Tag','VRegFigure'),'Tag','PositionAxes'))
426 cla reset
427 set(gca,'Tag','PositionAxes')
428 axes(findobj(findobj('Tag','VRegFigure'),'Tag','VectorAxes'))
429 cla reset
430 set(gca,'Tag','VectorAxes')
431 axes(findobj(findobj('Tag','VRegFigure'),'Tag','ActiveAxes'))
432 cla reset
433 set(gca,'Tag','ActiveAxes')
434 axes(findobj(findobj('Tag','VRegFigure'),'Tag','ReactiveAxes'))
435 cla reset
436 set(gca,'Tag','ReactiveAxes')
437 end
438
439 function PlotsChange                                % change angle, position and device plots
440     P = 1;                                           % deltamax in terms of pi
441     N = 101;                                         % delta divisions
442     DeviceType = get(findobj(findobj('Tag','CircuitFigure'),'Tag','DeviceType'),'UserData');
443     Technology = get(findobj(findobj('Tag','CircuitFigure'),'Tag','Technology'),'UserData');
444     Screen = get(findobj(findobj('Tag','VRegFigure'),'Tag','Screen'),'UserData');
445     deltaref = get(findobj(findobj('Tag','VRegFigure'),'Tag','AngleSlider'),'Value');
446     if deltaref == 0,
447         deltaref = 0.0157;
448     end;
449     delta1 = round(deltaref*N/(P*pi));
450     position = round(get(findobj(findobj('Tag','VRegFigure'),'Tag','PositionSlider'),'Value'));
451     Data = get(findobj('Tag','CircuitFigure'),'Userdata');
452     Sol = get(findobj(findobj('Tag','VRegFigure'),'Tag','Solution'),'UserData');
453     switch(get(findobj(findobj('Tag','VRegFigure'),'Tag','CompLoadFrame'),'UserData'))
454     case('Comp1')
455         Graph2dcomp(DeviceType, Technology, Screen, P, N, 1, Data, Sol);
456     case('Comp2')
457         Graph2dcomp(DeviceType, Technology, Screen, P, N, 2, Data, Sol);
458     case('Load ')
459         Graph2dload(DeviceType, Technology, Screen, P, N, Data, Sol);
460     end;
461     Graph2dang(DeviceType, Technology, Screen, delta1, P, N, position, Data, Sol);
462     Graph2dpos(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);
463     Graph2dVector(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);
464     Graph3d(DeviceType, Technology, Screen, delta1, P, N, Data, Sol);
465 end

```

B.2.8 Graph2dAng.m

```

1  function y = Graph2d(DeviceType, Technology, Screen, delta1, P, N, position, Data, Sol)
2
3  Vs = Data(Sol,1);
4  Vpccs = Data(Sol,2);
5  Vpccr = Data(Sol,3);
6  Vr = Data(Sol,4);
7  Ipccs = Data(Sol,8);
8  Ir = Data(Sol,10);
9  Ipccr = Data(Sol,11);
10
11  Ss = Vs.*conj(Ipccs);
12  Spccs = Vpccs.*conj(Ipccs);
13  Spccr = Vpccr.*conj(Ir);
14  Spccr = Vpccr.*conj(Ipccr);
15  Sr = Vr.*conj(Ir);
16
17  delta = [0;(P*pi/N);P*pi-(P*pi/N)];
18  interv = N;
19
20  switch(position)
21  case 0
22      switch(Screen)
23      case 'Voltage'
24          System = Vs(1:round(N/interv):N);
25          System1 = abs(System);
26          System2 = angle(System);
27      case 'Current'
28          System = Ipccs(1:round(N/interv):N);
29          System1 = abs(System);
30          System2 = angle(System);
31      case 'Power'
32          System = Ss(1:round(N/interv):N);
33          System1 = real(System);
34          System2 = imag(System);
35      end;
36  case 1
37      switch(Screen)
38      case 'Voltage'
39          System = Vpccs(1:round(N/interv):N);
40          System1 = abs(System);
41          System2 = angle(System);
42      case 'Current'
43          System = Ipccs(1:round(N/interv):N);
44          System1 = abs(System);
45          System2 = angle(System);
46      case 'Power'
47          System = Spccs(1:round(N/interv):N);
48          System1 = real(System);
49          System2 = imag(System);
50      end;
51  case 2
52      switch(Screen)
53      case 'Voltage'
54          System = Vpccr(1:round(N/interv):N);
55          System1 = abs(System);
56          System2 = angle(System);
57      case 'Current'
58          System = Ipccr(1:round(N/interv):N);
59          System1 = abs(System);
60          System2 = angle(System);
61      case 'Power'
62          System = Spccr(1:round(N/interv):N);
63          System1 = real(System);
64          System2 = imag(System);

```



```

65     end;
66 case 3
67     switch(Screen)
68     case 'Voltage'
69         System = Vpccr(1:round(N/interv):N);
70         System1 = abs(System);
71         System2 = angle(System);
72     case 'Current'
73         System = Ipccr(1:round(N/interv):N);
74         System1 = abs(System);
75         System2 = angle(System);
76     case 'Power'
77         System = Spccr(1:round(N/interv):N);
78         System1 = real(System);
79         System2 = imag(System);
80     end;
81 case 4
82     switch(Screen)
83     case 'Voltage'
84         System = Vr(1:round(N/interv):N);
85         System1 = abs(System);
86         System2 = angle(System);
87     case 'Current'
88         System = Ir(1:round(N/interv):N);
89         System1 = abs(System);
90         System2 = angle(System);
91     case 'Power'
92         System = Sr(1:round(N/interv):N);
93         System1 = real(System);
94         System2 = imag(System);
95     end;
96 end
97 S = delta(1:round(N/interv):N);
98
99 axes(findobj(findobj('Tag','VRegFigure'),'AngleAxes'));
100 cla;
101 H=plot(S,System1,'b');
102 set(gca,'Tag','AngleAxes');
103 set(H,'LineWidth',1)
104 hold on
105 H=plot(S,System2,'r');
106 set(H,'LineWidth',1)
107 hold off
108 xlabel('\delta [rad]');
109 set(gca,'XTick',[0 pi/2 pi],...
110     'XTickLabel',{'0 ','pi/2','pi '})
111 axis tight
112 a = axis;
113 AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
114 switch(Screen)
115     case 'Voltage'
116         title('System Voltage vs. delta');
117         legend('Magnitude [p.u.]','Angle [rad]',0)
118     case 'Current'
119         title('System Current vs. delta');
120         legend('Magnitude [p.u.]','Angle [rad]',0)
121     case 'Power'
122         title('System Power vs. delta');
123         legend('Active [p.u.]','Reactive [p.u.],0)
124 end;
125 set(findobj('String','0'),'Visible','OFF');
126 grid on
127
128 if get(findobj(findobj('Tag','CircuitFigure'),'Tag','HighResFiguresCheck'),'Value')
129     fig = findobj('Tag','AngleFigure');
130     if isempty(fig)
131         figure

```

ADDENDUM B PROGRAM LISTINGS

```

132     else
133         figure(fig)
134     clf
135 end;
136 set(gcf,'IntegerHandle','off',...
137     'Name','Angle Graph',...
138     'NumberTitle','off',...
139     'Tag','AngleFigure');
140 fsize = 36;
141 set(gcf,'Position',[1 29 1024 672]);
142 set(gca,'Position',[0.15 0.15 0.7 0.7]);
143 [H, HlineMag, HlineAng]=plotyy(S,System1,S,System2);
144 set(H,'LineWidth',1.5)
145 set(HlineMag,'LineWidth',1.5)
146 set(HlineAng,'LineWidth',1.5)
147 for n = 1:2
148     axes(H(n))
149     a = axis;
150     AXIS([-0.2 3.342 a(3) a(4)]);
151     set(gca,'FontSize',fsize*0.7);
152     set(gca,'XTick',[0 pi/2 pi]);
153     set(gcf,'Color',[1 1 1]);
154     set(findobj('String','0'),'Visible','Off');
155 end;
156
157 axes(H(1))
158 set(gca,'XTickLabel',{'0 ','pi/2','pi '})
159 switch(Screen)
160     case 'Voltage'
161         ht=title('System Voltage vs. delta');
162         hy=ylabel('Magnitude [p.u.]');
163         grid on
164     case 'Current'
165         ht=title('System Current vs. delta');
166         hy=ylabel('Magnitude [p.u.]');
167         grid on
168     case 'Power'
169         ht=title('System Power vs. delta');
170         hy=ylabel('Active Power [p.u.]');
171         grid on
172 end;
173 set(ht,'FontSize',fsize);
174 set(hy,'FontSize',fsize*0.8);
175 set(gca,'YTickMode','auto');
176 hx=xlabel('delta [rad]');
177 posx = get(hx,'Position');
178 set(hx,'FontSize',fsize*0.8);
179
180 axes(H(2))
181 set(gca,'XTickLabel','');
182 switch(Screen)
183     case 'Voltage'
184         hy=ylabel('Angle [rad]');
185     case 'Current'
186         hy=ylabel('Angle [rad]');
187     case 'Power'
188         hy=ylabel('Reactive Power [p.u.]');
189 end;
190 set(hy,'FontSize',fsize*0.8);
191
192 figure(findobj('Tag','VRegFigure'))
193 end;

```

B.2.9 Graph2dpos.m

```

1  function y = Graph2d(DeviceType, Technology, Screen, delta1, P, N, Data, Sol)
2
3  Vs = Data(Sol,1);
4  Vpccs = Data(Sol,2);
5  Vpccr = Data(Sol,3);
6  Vr = Data(Sol,4);
7  Vcomp = Data(Sol,5);
8  Vcomp2 = Data(Sol,6);
9  Ipccs = Data(Sol,8);
10 Ir = Data(Sol,10);
11 Ipccr = Data(Sol,11);
12 Icomp = Data(Sol,12);
13 Icomp2 = Data(Sol,13);
14
15 Ss = Vs.*conj(Ipccs);
16 Spccs = Vpccs.*conj(Ipccs);
17 Spccr = Vpccr.*conj(Ir);
18 Spccr = Vpccr.*conj(Ipccr);
19 Sr = Vr.*conj(Ir);
20 Scomp = Vcomp.*conj(Icomp);
21 Scomp2 = Vcomp2.*conj(Icomp2);
22
23 delta = [0:(P*pi/N):P*pi-(P*pi/N)];
24 switch(DeviceType)
25     case 'InLineDevice'
26         switch(Screen)
27             case 'Voltage'
28                 System = [Vs(delta1), Vpccs(delta1), Vcomp(delta1), Vcomp2(delta1), Vpccr(delta1), Vpccr(delta1), Vr(delta1)];
29                 S = [0 1 1 1.4 1.4 1.8 2.8];
30                 System1 = abs(System);
31                 System2 = angle(System);
32
33             case 'Current'
34                 if isnan(Vr)
35                     System = [Ipccs(delta1), Ipccs(delta1), Icomp(delta1), Icomp2(delta1), Ipccr(delta1), Ipccr(delta1), Ir(delta1), Vr(delta1)];
36                 else
37                     System = [Ipccs(delta1), Ipccs(delta1), Icomp(delta1), Icomp2(delta1), Ipccr(delta1), Ipccr(delta1), Ir(delta1), Ir(delta1)];
38                 end;
39                 S = [0 1 1 1.4 1.4 1.8 1.8 2.8];
40                 System1 = abs(System);
41                 System2 = angle(System);
42
43             case 'Power'
44                 System = [Ss(delta1), Spccs(delta1), Scomp(delta1), Scomp2(delta1), Spccr(delta1), Spccr(delta1), Spccr(delta1), Sr(delta1)];
45                 S = [0 1 1 1.4 1.4 1.8 1.8 2.8];
46                 System1 = real(System);
47                 System2 = imag(System);
48         end;
49     otherwise
50         switch(Screen)
51             case 'Voltage'
52                 System = [Vs(delta1), Vpccs(delta1), Vpccr(delta1), Vpccr(delta1), Vr(delta1)];
53                 S = [0 1 1.4 1.8 2.8];
54                 System1 = abs(System);
55                 System2 = angle(System);
56
57             case 'Current'
58                 if isnan(Vr)
59                     System = [Ipccs(delta1), Ipccs(delta1), Ipccs(delta1), Ipccr(delta1), Ipccr(delta1), Ir(delta1), Vr(delta1)];
60                 else
61                     System = [Ipccs(delta1), Ipccs(delta1), Ipccs(delta1), Ipccr(delta1), Ipccr(delta1), Ir(delta1), Ir(delta1)];
62                 end;
63                 S = [0 1 1.4 1.4 1.8 1.8 2.8];
64                 System1 = abs(System);

```

```

65         System2 = angle(System);
66
67         case 'Power'
68             System = [Ss(delta1); Spccs(delta1); Spocs(delta1)+Scomp(delta1); Spccr(delta1); Spccr(delta1); Spcc1(delta1); Sr(delta1)];
69             S = [0 1 1.4 1.4 1.8 1.8 2.8];
70             System1 = real(System);
71             System2 = imag(System);
72         end;
73     end;
74     if isnan(Vr)
75         XTickLabels = ['S-end'; 'PS '; 'PR '; 'PL '];
76         XTicks = [0 1 1.4 1.8];
77         AxisLimits = [0 pi 0 1.8 -2 2];
78     else
79         XTickLabels = ['S-end'; 'PS '; 'PR '; 'PL '; 'R-end'];
80         XTicks = [0 1 1.4 1.8 2.8];
81         AxisLimits = [0 pi 0 2.8 -2 2];
82     end;
83
84     axes(findobj(findobj(Tag,'VRegFigure'),Tag,'PositionAxes'));
85     cla;
86     H=plot(S,System1,'b');
87     set(gca,Tag,'PositionAxes');
88     set(H,'LineWidth',1)
89     hold on
90     H=plot(S,System2,'r');
91     set(H,'LineWidth',1)
92     hold off
93     set(gca,'XTick',XTicks,...
94         'XTickLabel',XTickLabels);
95     axis tight
96     a = axis;
97     AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
98     switch(Screen)
99         case 'Voltage'
100             title('Voltage vs. Position');
101             legend('Magnitude [p.u.];Angle [rad]',0)
102         case 'Current'
103             title('Current vs. Position');
104             legend('Magnitude [p.u.];Angle [rad]',0)
105         case 'Power'
106             title('Power vs. Position');
107             ylabel('Power [p.u.]');
108             legend('Active [p.u.];Reactive [p.u.],0)
109     end;
110     set(findobj('String','0'),'Visible','Off');
111     grid on
112
113     if get(findobj(findobj(Tag,'CircuitFigure'),Tag,'HighResFiguresCheck'),'Value')
114         fig = findobj(Tag,'PosFigure');
115         if isempty(fig)
116             figure
117         else
118             figure(fig)
119             clf
120         end;
121         set(gcf,'IntegerHandle','off',...
122             'Name','Position Graph',...
123             'NumberTitle','off',...
124             'Tag','PosFigure');
125         fsize = 36;
126         set(gcf,'Position',[1 29 1024 672]);
127         set(gca,'Position',[0.15 0.15 0.7 0.7]);
128         [H, HlineMag, HlineAng]=plotyy(S,System1,S,System2);
129         set(H,'LineWidth',1.5)
130         set(HlineMag,'LineWidth',1.5)
131         set(HlineAng,'LineWidth',1.5)

```

ADDENDUM B PROGRAM LISTINGS

```

132     for n = 1:2
133         axes(H(n))
134         a = axis;
135         if isnan(Vr)
136             AXIS([-0.2 2 a(3) a(4)]);
137         else
138             AXIS([-0.2 3 a(3) a(4)]);
139         end;
140         set(gca,'FontSize',fsize*0.7);
141         set(gca,'XTick',XTicks);
142         set(gcf,'Color',[1 1 1]);
143         set(findobj('String','0'),'Visible','Off');
144     end;
145
146     axes(H(1))
147     set(gca,'XTickLabel',XTickLabels);
148     switch(Screen)
149         case 'Voltage'
150             ht=title('Voltage vs. Position');
151             hy=ylabel('Magnitude [p.u.]');
152             grid on
153         case 'Current'
154             ht=title('Current vs. Position');
155             hy=ylabel('Magnitude [p.u.]');
156             grid on
157         case 'Power'
158             ht=title('Power vs. Position');
159             hy=ylabel('Active Power [p.u.]');
160             grid on
161     end;
162     hx=xlabel('Position along network');
163     set(ht,'FontSize',fsize);
164     set(hy,'FontSize',fsize*0.8);
165     set(hx,'FontSize',fsize*0.8);
166     set(gca,'YTickMode','auto');
167
168     axes(H(2))
169     set(gca,'XTickLabel','');
170     switch(Screen)
171         case 'Voltage'
172             hy=ylabel('Angle [rad]');
173         case 'Current'
174             hy=ylabel('Angle [rad]');
175         case 'Power'
176             hy=ylabel('Reactive Power [p.u.]');
177     end;
178     set(hy,'FontSize',fsize*0.8);
179
180     figure(findobj('Tag','VRegFigure'))
181 end;

```

B.2.10 Graph2dComp.m

```

1 function y = Graph2d(DeviceType, Technology, Screen, P, N, Comp, Data, Sol)
2 if Comp == 1
3     Vcomp = Data(Sol,5); % Compensator 1
4     Icomp = Data(Sol,12);
5     Scomp = Vcomp.*conj(Icomp);
6 else
7     Vcomp = Data(Sol,6); % Compensator 2
8     Icomp = Data(Sol,13);
9     Scomp = Vcomp.*conj(Icomp);
10 end;
11
12 delta = [0:(P*pi/N):P*pi-(P*pi/N)];
13 interv = N;
14
15 switch(Screen)
16 case 'Voltage'
17     System = Vcomp(1:round(N/interv):N);
18     System1 = abs(System);
19     System2 = angle(System);
20 case 'Current'
21     System = Icomp(1:round(N/interv):N);
22     System1 = abs(System);
23     System2 = angle(System);
24 case 'Power'
25     System = Scomp(1:round(N/interv):N);
26     System1 = real(System);
27     System2 = imag(System);
28     System3 = abs(System);
29 end
30 S = delta(1:round(N/interv):N);
31
32 axes(findobj(findobj('Tag','VRagFigure'),'Tag','DeviceAxes'));
33 cla;
34 H=plot(S,System1,'b');
35 set(gca,'Tag','DeviceAxes');
36 set(H,'LineWidth',1)
37 hold on
38 H=plot(S,System2,'r');
39 set(H,'LineWidth',1)
40 switch(Screen)
41 case 'Power'
42     H=plot(S,System3,'g');
43     set(H,'LineWidth',1)
44 end
45 hold off
46 xlabel('\delta [rad]');
47 set(gca,'XTick',[0 pi/2 pi],...
48 'XTickLabel',{'0 ','pi/2','pi '})
49 axis tight
50 a = axis;
51 AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
52 switch(Screen)
53 case 'Voltage'
54     title('Compensator Voltage');
55     legend('Magnitude [p.u.]','Angle [rad]',0);
56 case 'Current'
57     title('Compensator Current');
58     legend('Magnitude [p.u.]','Angle [rad]',0);
59 case 'Power'
60     title('Compensator Power');
61     legend('Active [p.u.]','Reactive [p.u.]','Magnitude [p.u.]',0);
62 end;
63 set(findobj('String','0'),'Visible','Off');
64 grid on

```

ADDENDUM B PROGRAM LISTINGS

```

65
66 if get(findobj(findobj('Tag','CircuitFigure'),Tag,'HighResFiguresCheck'),'Value')
67     fig = findobj('Tag','CompFigure');
68     if isempty(fig)
69         figure
70     else
71         figure(fig)
72         clf
73     end;
74     set(gcf,'IntegerHandle','off',...
75         'Name','Compensator Graph',...
76         'NumberTitle','off',...
77         'Tag','CompFigure');
78     fsize = 36;
79     set(gcf,'Position',[1 29 1024 672]);
80     set(gca,'Position',[0.15 0.15 0.7 0.7]);
81     [H, HlineMag, HlineAng]=plotyy(S,System1,S,System2);
82     set(H,'LineWidth',1.5)
83     set(HlineMag,'LineWidth',1.5)
84     set(HlineAng,'LineWidth',1.5)
85     for n = 1:2
86         axes(H(n))
87         a = axis;
88         AXIS([-0.2 3.34 a(3) a(4)]);
89         set(gca,'FontSize',fsize*0.7);
90         set(gca,'XTick',[0 pi/2 pi]);
91         set(gcf,'Color',[1 1 1]);
92         set(findobj('String','0'),'Visible','Off');
93     end;
94
95     axes(H(1))
96     set(gca,'XTickLabel',{'0 ','pi/2','pi '});
97     switch(Screen)
98     case 'Voltage'
99         ht=title('Compensator Voltage');
100        hy=ylabel('Magnitude [p.u.]');
101        grid on
102     case 'Current'
103        ht=title('Compensator Current');
104        hy=ylabel('Magnitude [p.u.]');
105        grid on
106     case 'Power'
107        ht=title('Compensator Power');
108        hy=ylabel('Active Power [p.u.]');
109        grid on
110     end;
111     hx=xlabel('^delta [rad]');
112     set(ht,'FontSize',fsize);
113     set(hy,'FontSize',fsize*0.8);
114     set(hx,'FontSize',fsize*0.8);
115     set(gca,'YTickMode','auto');
116
117     axes(H(2))
118     set(gca,'XTickLabel','');
119     switch(Screen)
120     case 'Voltage'
121        hy=ylabel('Angle [rad]');
122     case 'Current'
123        hy=ylabel('Angle [rad]');
124     case 'Power'
125        hy=ylabel('Reactive Power [p.u.]');
126     end;
127     set(hy,'FontSize',fsize*0.8);
128
129     figure(findobj('Tag','VRegFigure'))
130 end;

```

B.2.11 Graph2dLoad.m

```

1 function y = Graph2d(DeviceType, Technology, Screen, P, N, Data, Sol)
2
3 Vpccr = Data(Sol,3);
4 Iload = Data(Sol,9);
5 Sload = Vpccr.*conj(Iload);
6
7 delta = [0:(P*pi/N):P*pi-(P*pi/N)];
8 interv = N;
9
10 switch(Screen)
11     case 'Voltage'
12         System = Vpccr(1:round(N/interv):N);
13         System1 = abs(System);
14         System2 = angle(System);
15     case 'Current'
16         System = Iload(1:round(N/interv):N);
17         System1 = abs(System);
18         System2 = angle(System);
19     case 'Power'
20         System = Sload(1:round(N/interv):N);
21         System1 = real(System);
22         System2 = imag(System);
23         System3 = abs(System);
24     end
25 S = delta(1:round(N/interv):N);
26
27 axes(findobj(findobj('Tag','VRagFigure'),'Tag','DeviceAxes'));
28 cla;
29 H=plot(S,System1,'b');
30 set(gca,'Tag','DeviceAxes');
31 set(H,'LineWidth',1)
32 hold on
33 H=plot(S,System2,'r');
34 set(H,'LineWidth',1)
35 switch(Screen)
36     case 'Power'
37         H=plot(S,System3,'g');
38         set(H,'LineWidth',1)
39     end
40 hold off
41 xlabel('\delta [rad]');
42 set(gca,'XTick',[0 pi/2 pi],...
43     'XTickLabel',{'0 ','pi/2','pi '})
44 axis tight
45 a = axis;
46 AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
47 switch(Screen)
48     case 'Voltage'
49         title('Load Voltage');
50         legend('Magnitude [p.u.'],'Angle [rad]',0);
51     case 'Current'
52         title('Load Current');
53         legend('Magnitude [p.u.'],'Angle [rad]',0);
54     case 'Power'
55         title('Load Power');
56         legend('Active [p.u.'],'Reactive [p.u.'],'Magnitude [p.u.]',0);
57     end;
58 set(findobj('String','0'),'Visible','Off');
59 grid on
60
61
62 if get(findobj(findobj('Tag','CircuitFigure'),'Tag','HighResFiguresCheck'),'Value')
63     fig = findobj('Tag','LoadFigure');
64     if isempty(fig)

```



```

65     figure
66 else
67     figure(fig)
68     clf
69 end;
70 set(gcf,'IntegerHandle','off',...
71     'Name','Load Graph',...
72     'NumberTitle','off',...
73     'Tag','LoadFigure');
74 fsize = 36;
75 set(gcf,'Position',[1 29 1024 672]);
76 set(gca,'Position',[0.15 0.15 0.7 0.7]);
77 [H, HlineMag, HlineAng]=plotyy(S,System1,S,System2);
78 set(H,'LineWidth',1.5)
79 set(HlineMag,'LineWidth',1.5)
80 set(HlineAng,'LineWidth',1.5)
81 for n = 1:2
82     axes(H(n))
83     a = axis;
84     AXIS([-0.2 3.34 a(3) a(4)]);
85     set(gca,'FontSize',fsize*0.7);
86     set(gca,'XTick',[0 pi/2 pi])
87     set(gcf,'Color',[1 1 1]);
88     set(findobj('String','0'),'Visible','Off');
89 end;
90
91 axes(H(1))
92 set(gca,'XTickLabel',{'0 ','pi/2','pi '})
93 switch(Screen)
94     case 'Voltage'
95         ht=title('Load Voltage');
96         hy=ylabel('Magnitude [p.u.]');
97         grid on
98     case 'Current'
99         ht=title('Load Current');
100        hy=ylabel('Magnitude [p.u.]');
101        grid on
102    case 'Power'
103        ht=title('Load Power');
104        hy=ylabel('Active Power [p.u.]');
105        grid on
106 end;
107 hx=xlabel('delta [rad]');
108 set(ht,'FontSize',fsize);
109 set(hy,'FontSize',fsize*0.8);
110 set(hx,'FontSize',fsize*0.8);
111 set(gca,'YTickMode','auto');
112
113 axes(H(2))
114 set(gca,'XTickLabel','');
115 switch(Screen)
116     case 'Voltage'
117         hy=ylabel('Angle [rad]');
118     case 'Current'
119         hy=ylabel('Angle [rad]');
120     case 'Power'
121         hy=ylabel('Reactive Power [p.u.]');
122 end;
123 set(hy,'FontSize',fsize*0.8);
124
125 figure(findobj('Tag','VRegFigure'))
126 end;

```

B.2.12 Graph2dVector.m

```

1 function y = VectorDiagram(DeviceType, Technology, Screen, delta1, P, N, Data, Sol)
2
3 Vs = Data(Sol,1);
4 Vpccs = Data(Sol,2);
5 Vpccr = Data(Sol,3);
6 Vr = Data(Sol,4);
7 Vcomp = Data(Sol,5);
8 Vcomp2 = Data(Sol,6);
9 Vpcen = Data(Sol,7);
10 Ipccs = Data(Sol,8);
11 Iload = Data(Sol,9);
12 Ir = Data(Sol,10);
13 Ipccr = Data(Sol,11);
14 Icomp = Data(Sol,12);
15 Icomp2 = Data(Sol,13);
16
17 Ss = Vs.*conj(Ipccs);
18 Spccs = Vpccs.*conj(Ipccs);
19 Spccr = Vpccr.*conj(Ir);
20 Spccr = Vpccr.*conj(Ipccr);
21 Sload = Vpccr.*conj(Iload);
22 Sr = Vr.*conj(Ir);
23 Scomp = Vcomp.*conj(Icomp);
24 Scomp2 = Vcomp2.*conj(Icomp2);
25
26
27 delta = [0:(P*pi/N):P*pi-(P*pi/N)];
28
29 axes(findobj(findobj('Tag','VRegFigure'),'Tag','VectorAxes'));
30 cla reset;
31 set(gca,'Tag','VectorAxes');
32
33 switch(Screen)
34     case 'Voltage'
35         V1text = 'V_{S}';
36         V1Xs = 0;
37         V1Ys = 0;
38         V1Xe = V1Xs+real(Vs(delta1));
39         V1Ye = V1Ys+imag(Vs(delta1));
40         V2text = 'I_{PCCS}Z_{S}';
41         V2Xs = V1Xe;
42         V2Ys = V1Ye;
43         V2Xe = V2Xs-real(Vs(delta1)-Vpccs(delta1));
44         V2Ye = V2Ys-imag(Vs(delta1)-Vpccs(delta1));
45         V3text = "";
46         V3Xs = 0;
47         V3Ys = 0;
48         V3Xe = 0;
49         V3Ye = 0;
50         V4text = "";
51         V4Xs = 0;
52         V4Ys = 0;
53         V4Xe = 0;
54         V4Ye = 0;
55         V5text = "";
56         V5Xs = 0;
57         V5Ys = 0;
58         V5Xe = 0;
59         V5Ye = 0;
60         V6text = "";
61         V6Xs = 0;
62         V6Ys = 0;
63         V6Xe = 0;
64         V6Ye = 0;

```

ADDENDUM B PROGRAM LISTINGS

```

65     V7text = 'V_{PCCR}';
66     V7Xs = 0;
67     V7Ys = 0;
68     V7Xe = V7Xs+real(Vpccr(delta1));
69     V7Ye = V7Ys+imag(Vpccr(delta1));
70     V8text = 'I_{PCCR}Z_{R}';
71     V8Xs = V7Xe;
72     V8Ys = V7Ye;
73     V8Xe = V8Xs-real(Vpccr(delta1)-Vr(delta1));
74     V8Ye = V8Ys-imag(Vpccr(delta1)-Vr(delta1));
75     V9text = 'V_{R}';
76     V9Xs = 0;
77     V9Ys = 0;
78     V9Xe = V9Xs+real(Vr(delta1));
79     V9Ye = V9Ys+imag(Vr(delta1));
80     V10text = ";
81     V10Xs = 0;
82     V10Ys = 0;
83     V10Xe = 0;
84     V10Ye = 0;
85 case 'Current'
86     V1text = ";
87     V1Xs = 0;
88     V1Ys = 0;
89     V1Xe = 0;
90     V1Ye = 0;
91     V2text = ";
92     V2Xs = 0;
93     V2Ys = 0;
94     V2Xe = 0;
95     V2Ye = 0;
96     V3text = 'I_{PCCR}';
97     V3Xs = 0;
98     V3Ys = 0;
99     V3Xe = V3Xs+real(Ipccr(delta1));
100    V3Ye = V3Ys+imag(Ipccr(delta1));
101    V4text = 'I_{Load}';
102    V4Xs = 0;
103    V4Ys = 0;
104    V4Xe = V4Xs+real(ILoad(delta1));
105    V4Ye = V4Ys+imag(ILoad(delta1));
106    V5text = 'I_{R}';
107    V5Xs = V4Xe;
108    V5Ys = V4Ye;
109    V5Xe = V5Xs+real(Ir(delta1));
110    V5Ye = V5Ys+imag(Ir(delta1));
111    V6text = ";
112    V6Xs = 0;
113    V6Ys = 0;
114    V6Xe = 0;
115    V6Ye = 0;
116    V7text = ";
117    V7Xs = 0;
118    V7Ys = 0;
119    V7Xe = 0;
120    V7Ye = 0;
121    V8text = '0';
122    V8Xs = 0;
123    V8Ys = 0;
124    V8Xe = 0;
125    V8Ye = 0;
126    V9text = '0';
127    V9Xs = 0;
128    V9Ys = 0;
129    V9Xe = 0;
130    V9Ye = 0;
131    V10text = ";

```

ADDENDUM B PROGRAM LISTINGS

```

132     V10Xs = 0;
133     V10Ys = 0;
134     V10Xe = 0;
135     V10Ye = 0;
136     case 'Power'
137         V1text = 'S_{S}';
138         V1Xs = 0;
139         V1Ys = 0;
140         V1Xe = V1Xs+real(Ss(delta1));
141         V1Ye = V1Ys+imag(Ss(delta1));
142         V2text = 'S_{ZS}';
143         V2Xs = V1Xe;
144         V2Ys = V1Ye;
145         V2Xe = V2Xs+real(Spccs(delta1)-Ss(delta1));
146         V2Ye = V2Ys+imag(Spccs(delta1)-Ss(delta1));
147         V3text = ";
148         V3Xs = 0;
149         V3Ys = 0;
150         V3Xe = 0;
151         V3Ye = 0;
152         V4text = ";
153         V4Xs = 0;
154         V4Ys = 0;
155         V4Xe = 0;
156         V4Ye = 0;
157         V5text = ";
158         V5Xs = 0;
159         V5Ys = 0;
160         V5Xe = 0;
161         V5Ye = 0;
162         V6text = 'S_{PCCR}';
163         V6Xs = 0;
164         V6Ys = 0;
165         V6Xe = V6Xs+real(Spccr(delta1));
166         V6Ye = V6Ys+imag(Spccr(delta1));
167         V7text = 'S_{PCCL}';
168         V7Xs = 0;
169         V7Ys = 0;
170         V7Xe = V7Xs+real(Spccl(delta1));
171         V7Ye = V7Ys+imag(Spccl(delta1));
172         V8text = 'S_{Load}';
173         V8Xs = V7Xe;
174         V8Ys = V7Ye;
175         V8Xe = V8Xs+real(Sload(delta1));
176         V8Ye = V8Ys+imag(Sload(delta1));
177         V9text = 'S_{ZR}';
178         V9Xs = V7Xe;
179         V9Ys = V7Ye;
180         V9Xe = V9Xs+real(Sr(delta1)-Spccl(delta1));
181         V9Ye = V9Ys+imag(Sr(delta1)-Spccl(delta1));
182         V10text = 'S_{R}';
183         V10Xs = V9Xe;
184         V10Ys = V9Ye;
185         V10Xe = V10Xs-real(Sr(delta1));
186         V10Ye = V10Ys-imag(Sr(delta1));
187     end;
188
189     switch(DeviceType)
190     case 'ShuntDevice'
191         switch(Screen)
192         case 'Current'
193             V1text = 'I_{PCCS}';
194             V1Xe = V1Xs+real(Ipccs(delta1));
195             V1Ye = V1Ys+imag(Ipccs(delta1));
196             V2text = 'I_{Comp2}';
197             V2Xs = V1Xe;
198             V2Ys = V1Ye;

```

```

199         V2Xe = V2Xs+real(Icomp2(delta1));
200         V2Ye = V2Ys+imag(Icomp2(delta1));
201     case 'Power'
202         V3text = 'S_{PCCS}';
203         V3Xe = V3Xs+real(Specs(delta1));
204         V3Ye = V3Ys+imag(Specs(delta1));
205         V5text = 'S_{Comp2}';
206         V5Xs = V3Xe;
207         V5Ys = V3Ye;
208         V5Xe = V5Xs+real(Scomp2(delta1));
209         V5Ye = V5Ys+imag(Scomp2(delta1));
210     end;
211 case 'SerieDevice'
212     switch(Screen)
213     case 'Voltage'
214         V3text = 'V_{PCCS}';
215         V3Xe = V3Xs+real(Vpccs(delta1));
216         V3Ye = V3Ys+imag(Vpccs(delta1));
217         V4text = 'V_{Comp}';
218         V4Xs = V3Xe;
219         V4Ys = V3Ye;
220         V4Xe = V4Xs+real(Vcomp(delta1));
221         V4Ye = V4Ys+imag(Vcomp(delta1));
222     case 'Power'
223         V3text = 'S_{PCCS}';
224         V3Xe = V3Xs+real(Specs(delta1));
225         V3Ye = V3Ys+imag(Specs(delta1));
226         V4text = 'S_{Comp}';
227         V4Xs = V3Xe;
228         V4Ys = V3Ye;
229         V4Xe = V4Xs+real(Scomp(delta1));
230         V4Ye = V4Ys+imag(Scomp(delta1));
231     end;
232 case 'SeriesShunt'
233     switch(Screen)
234     case 'Voltage'
235         V3text = 'V_{PCCS}';
236         V3Xe = V3Xs+real(Vpccs(delta1));
237         V3Ye = V3Ys+imag(Vpccs(delta1));
238         V4text = 'V_{Comp}';
239         V4Xs = V3Xe;
240         V4Ys = V3Ye;
241         V4Xe = V4Xs+real(Vcomp(delta1));
242         V4Ye = V4Ys+imag(Vcomp(delta1));
243     case 'Current'
244         V1text = 'I_{PCCS}';
245         V1Xe = V1Xs+real(Ipccs(delta1));
246         V1Ye = V1Ys+imag(Ipccs(delta1));
247         V2text = 'I_{Comp2}';
248         V2Xs = V1Xe;
249         V2Ys = V1Ye;
250         V2Xe = V2Xs+real(Icomp2(delta1));
251         V2Ye = V2Ys+imag(Icomp2(delta1));
252     case 'Power'
253         V3text = 'S_{PCCS}';
254         V3Xe = V3Xs+real(Spccs(delta1));
255         V3Ye = V3Ys+imag(Spccs(delta1));
256         V4text = 'S_{Comp}';
257         V4Xs = V3Xe;
258         V4Ys = V3Ye;
259         V4Xe = V4Xs+real(Scomp(delta1));
260         V4Ye = V4Ys+imag(Scomp(delta1));
261         V5text = 'S_{Comp2}';
262         V5Xs = V4Xe;
263         V5Ys = V4Ye;
264         V5Xe = V5Xs+real(Scomp2(delta1));
265         V5Ye = V5Ys+imag(Scomp2(delta1));

```

```

266     end;
267 case 'InLineDvice'
268     switch(Screen)
269     case 'Voltage'
270         V3text = 'V_{PCCS}';
271         V3Xe = V3Xs+real(Vpccs(delta1));
272         V3Ye = V3Ys+imag(Vpccs(delta1));
273         V4text = 'V_{DCS}';
274         V4Xe = V4Xs+real(Vcomp(delta1));
275         V4Ye = V4Ys+imag(Vcomp(delta1));
276         V5text = 'I_{DC}Z_{DC}';
277         V5Xs = V4Xe;
278         V5Ys = V4Ye;
279         V5Xe = V5Xs-real(Vcomp(delta1)-Vcomp2(delta1));
280         V5Ye = V5Ys-imag(Vcomp(delta1)-Vcomp2(delta1));
281         V6text = 'V_{DCR}';
282         V6Xe = V6Xs+real(Vcomp2(delta1));
283         V6Ye = V6Ys+imag(Vcomp2(delta1));
284         V7text = 'V_{PCCR}';
285         V7Xe = V7Xs+real(Vpccr(delta1));
286         V7Ye = V7Ys+imag(Vpccr(delta1));
287     case 'Current'
288         V1text = 'I_{PCCS}';
289         V1Xe = V1Xs+real(Ipccs(delta1));
290         V1Ye = V1Ys+imag(Ipccs(delta1));
291         V2text = 'I_{DC}';
292         V2Xe = V2Xs+real(Icomp(delta1));
293         V2Ye = V2Ys+imag(Icomp(delta1));
294     case 'Power'
295         V3text = 'S_{PCCS}';
296         V3Xe = V3Xs+real(Spccs(delta1));
297         V3Ye = V3Ys+imag(Spccs(delta1));
298         V4text = 'S_{DCS}';
299         V4Xe = V4Xs+real(Scomp(delta1));
300         V4Ye = V4Ys+imag(Scomp(delta1));
301         V5text = 'S_{ZDC}';
302         V5Xs = V4Xe;
303         V5Ys = V4Ye;
304         V5Xe = V5Xs+real(Scomp2(delta1)-Scomp(delta1));
305         V5Ye = V5Ys+imag(Scomp2(delta1)-Scomp(delta1));
306         V6text = 'S_{DCR}';
307         V6Xe = V6Xs+real(Scomp2(delta1));
308         V6Ye = V6Ys+imag(Scomp2(delta1));
309         V7text = 'S_{PCCR}';
310         V7Xe = V7Xs+real(Spccr(delta1));
311         V7Ye = V7Ys+imag(Spccr(delta1));
312     end;
313
314 end;
315
316 lineV1 = line([V1Xs V1Xe],[V1Ys V1Ye]);
317 headV1 = line([V1Xe V1Xe],[V1Ye V1Ye]);
318 set(lineV1,'Color','b');
319
320 lineV2 = line([V2Xs V2Xe],[V2Ys V2Ye]);
321 headV2 = line([V2Xe V2Xe],[V2Ye V2Ye]);
322 set(lineV2,'Color','k');
323
324 lineV3 = line([V3Xs V3Xe],[V3Ys V3Ye]);
325 headV3 = line([V3Xe V3Xe],[V3Ye V3Ye]);
326 set(lineV3,'Color','m');
327
328 lineV4 = line([V4Xs V4Xe],[V4Ys V4Ye]);
329 headV4 = line([V4Xe V4Xe],[V4Ye V4Ye]);
330 set(lineV4,'Color','e');
331
332 lineV5 = line([V5Xs V5Xe],[V5Ys V5Ye]);

```

ADDENDUM B PROGRAM LISTINGS

```

333 headV5 = line([V5Xe V5Xe],[V5Ye V5Ye]);
334 set(lineV5,'Color','r');
335
336 lineV6 = line([V6Xs V6Xe],[V6Ys V6Ye]);
337 headV6 = line([V6Xe V6Xe],[V6Ye V6Ye]);
338 set(lineV6,'Color','g');
339
340 lineV7 = line([V7Xs V7Xe],[V7Ys V7Ye]);
341 headV7 = line([V7Xe V7Xe],[V7Ye V7Ye]);
342 set(lineV7,'Color','b');
343
344 lineV8 = line([V8Xs V8Xe],[V8Ys V8Ye]);
345 headV8 = line([V8Xe V8Xe],[V8Ye V8Ye]);
346 set(lineV8,'Color','k');
347
348 lineV9 = line([V9Xs V9Xe],[V9Ys V9Ye]);
349 headV9 = line([V9Xe V9Xe],[V9Ye V9Ye]);
350 set(lineV9,'Color','m');
351
352 lineV10 = line([V10Xs V10Xe],[V10Ys V10Ye]);
353 headV10 = line([V10Xe V10Xe],[V10Ye V10Ye]);
354 set(lineV10,'Color','c');
355
356 axis tight
357 a = axis;
358 xSize = a(2)-a(1);
359 ySize = a(4)-a(3);
360 AXIS([a(1)-(xSize)*0.1 a(2)+(xSize)*0.1 a(3)-(ySize)*0.1 a(4)+(ySize)*0.1])
361 textV1 = text((V1Xe+V1Xs)/2+xSize*0.01,(V1Ye+V1Ys)/2+ySize*.04,V1text);
362 textV2 = text((V2Xe+V2Xs)/2+xSize*0.01,(V2Ye+V2Ys)/2+ySize*.04,V2text);
363 textV3 = text((V3Xe+V3Xs)/2+xSize*0.01,(V3Ye+V3Ys)/2+ySize*.04,V3text);
364 textV4 = text((V4Xe+V4Xs)/2+xSize*0.01,(V4Ye+V4Ys)/2+ySize*.04,V4text);
365 textV5 = text((V5Xe+V5Xs)/2+xSize*0.01,(V5Ye+V5Ys)/2+ySize*.04,V5text);
366 textV6 = text((V6Xe+V6Xs)/2+xSize*0.01,(V6Ye+V6Ys)/2+ySize*.04,V6text);
367 textV7 = text((V7Xe+V7Xs)/2+xSize*0.01,(V7Ye+V7Ys)/2+ySize*.04,V7text);
368 textV8 = text((V8Xe+V8Xs)/2+xSize*0.01,(V8Ye+V8Ys)/2+ySize*.04,V8text);
369 textV9 = text((V9Xe+V9Xs)/2+xSize*0.01,(V9Ye+V9Ys)/2+ySize*.04,V9text);
370 textV10 = text((V10Xe+V10Xs)/2+xSize*0.01,(V10Ye+V10Ys)/2+ySize*.04,V10text);
371
372 switch(Screen)
373 case 'Voltage'
374     title('Voltage Vector Diagram')
375     xlabel('Active Voltage [p.u.]')
376     ylabel('Reactive Voltage [p.u.]')
377 case 'Current'
378     title('Current Vector Diagram')
379     xlabel('Active Current [p.u.]')
380     ylabel('Reactive Current [p.u.]')
381 case 'Power'
382     title('Power Vector Diagram')
383     xlabel('Active Power [p.u.]')
384     ylabel('Reactive Power [p.u.]')
385 end;
386 set(textV1,'FontSize',8);
387 set(textV2,'FontSize',8);
388 set(textV3,'FontSize',8);
389 set(textV4,'FontSize',8);
390 set(textV5,'FontSize',8);
391 set(textV6,'FontSize',8);
392 set(textV7,'FontSize',8);
393 set(textV8,'FontSize',8);
394 set(textV9,'FontSize',8);
395 set(textV10,'FontSize',8);
396 set(findobj('String','0'),'Visible','Off');
397 axis equal
398
399 if get(findobj(findobj('Tag','CircuitFigure'),'Tag','HighResFiguresCheck'),'Value')

```

```

400     fsize = 36;
401     fig = findobj('Tag','VectorFigure');
402     if isempty(fig)
403         figure
404         set(gcf,'IntegerHandle','off',...
405             'Name','Vector Diagram',...
406             'NumberTitle','off',...
407             'Tag','VectorFigure');
408     else
409         figure(fig)
410     end;
411     clf
412     set(gcf,'Position',[1    29    1024    672]);
413     set(gca,'Position',[0.15 0.15 0.7 0.7]);
414
415     lineV1 = line([V1Xs V1Xe],[V1Ys V1Ye]);
416     headV1 = line([V1Xe V1Xe],[V1Ye V1Ye]);
417     set(lineV1,'Color','b');
418
419     lineV2 = line([V2Xs V2Xe],[V2Ys V2Ye]);
420     headV2 = line([V2Xe V2Xe],[V2Ye V2Ye]);
421     set(lineV2,'Color','k');
422
423     lineV3 = line([V3Xs V3Xe],[V3Ys V3Ye]);
424     headV3 = line([V3Xe V3Xe],[V3Ye V3Ye]);
425     set(lineV3,'Color','m');
426
427     lineV4 = line([V4Xs V4Xe],[V4Ys V4Ye]);
428     headV4 = line([V4Xe V4Xe],[V4Ye V4Ye]);
429     set(lineV4,'Color','c');
430
431     lineV5 = line([V5Xs V5Xe],[V5Ys V5Ye]);
432     headV5 = line([V5Xe V5Xe],[V5Ye V5Ye]);
433     set(lineV5,'Color','r');
434
435     lineV6 = line([V6Xs V6Xe],[V6Ys V6Ye]);
436     headV6 = line([V6Xe V6Xe],[V6Ye V6Ye]);
437     set(lineV6,'Color','g');
438
439     lineV7 = line([V7Xs V7Xe],[V7Ys V7Ye]);
440     headV7 = line([V7Xe V7Xe],[V7Ye V7Ye]);
441     set(lineV7,'Color','b');
442
443     lineV8 = line([V8Xs V8Xe],[V8Ys V8Ye]);
444     headV8 = line([V8Xe V8Xe],[V8Ye V8Ye]);
445     set(lineV8,'Color','k');
446
447     lineV9 = line([V9Xs V9Xe],[V9Ys V9Ye]);
448     headV9 = line([V9Xe V9Xe],[V9Ye V9Ye]);
449     set(lineV9,'Color','m');
450
451     lineV10 = line([V10Xs V10Xe],[V10Ys V10Ye]);
452     headV10 = line([V10Xe V10Xe],[V10Ye V10Ye]);
453     set(lineV10,'Color','c');
454
455     axis tight
456     a = axis;
457     xSize = a(2)-a(1);
458     ySize = a(4)-a(3);
459     AXIS([a(1)-(xSize)*0.1 a(2)+(xSize)*0.1 a(3)-(ySize)*0.1 a(4)+(ySize)*0.1])
460     textV1 = text((V1Xe+V1Xs)/2+xSize*0.01,(V1Ye+V1Ys)/2+ySize*.04,V1text);
461     textV2 = text((V2Xe+V2Xs)/2+xSize*0.01,(V2Ye+V2Ys)/2+ySize*.04,V2text);
462     textV3 = text((V3Xe+V3Xs)/2+xSize*0.01,(V3Ye+V3Ys)/2+ySize*.04,V3text);
463     textV4 = text((V4Xe+V4Xs)/2+xSize*0.01,(V4Ye+V4Ys)/2+ySize*.04,V4text);
464     textV5 = text((V5Xe+V5Xs)/2+xSize*0.01,(V5Ye+V5Ys)/2+ySize*.04,V5text);
465     textV6 = text((V6Xe+V6Xs)/2+xSize*0.01,(V6Ye+V6Ys)/2+ySize*.04,V6text);
466     textV7 = text((V7Xe+V7Xs)/2+xSize*0.01,(V7Ye+V7Ys)/2+ySize*.04,V7text);

```



```

467 textV8 = text((V8Xe+V8Xs)/2+xSize*0.01,(V8Ye+V8Ys)/2+ySize*.04,V8text);
468 textV9 = text((V9Xe+V9Xs)/2+xSize*0.01,(V9Ye+V9Ys)/2+ySize*.04,V9text);
469 textV10 = text((V10Xe+V10Xs)/2+xSize*0.01,(V10Ye+V10Ys)/2+ySize*.04,V10text);
470
471 switch(Screen)
472 case 'Voltage'
473     ht=title('Voltage Vector Diagram');
474     hx=xlabel('Active Voltage [p.u.]');
475     hy=ylabel('Reactive Voltage [p.u.]');
476 case 'Current'
477     ht=title('Current Vector Diagram');
478     hx=xlabel('Active Current [p.u.]');
479     hy=ylabel('Reactive Current [p.u.]');
480 case 'Power'
481     ht=title('Power Vector Diagram');
482     hx=xlabel('Active Power [p.u.]');
483     hy=ylabel('Reactive Power [p.u.]');
484 end;
485 set(ht,'FontSize',fsize);
486 posx = get(hx,'Position');
487 set(hx,'FontSize',fsize*0.8,...
488     'Position',[posx(1) posx(2) posx(3)]);
489 posy = get(hy,'Position');
490 set(hy,'FontSize',fsize*0.8,...
491     'Position',[posy(1) posy(2) posy(3)]);
492 set(gcf,'Color',[1 1 1]);
493 set(gca,'FontSize',fsize*0.7);
494 set(textV1,'FontSize',fsize*0.8);
495 set(textV2,'FontSize',fsize*0.8);
496 set(textV3,'FontSize',fsize*0.8);
497 set(textV4,'FontSize',fsize*0.8);
498 set(textV5,'FontSize',fsize*0.8);
499 set(textV6,'FontSize',fsize*0.8);
500 set(textV7,'FontSize',fsize*0.8);
501 set(textV8,'FontSize',fsize*0.8);
502 set(textV9,'FontSize',fsize*0.8);
503 set(textV10,'FontSize',fsize*0.8);
504 set(findobj('String','0'),'Visible','Off');
505 figure(findobj('Tag','VRegFigure'))
506 end;

```

B.2.13 Graph3d.m

```

1 function y = Graph3d(DeviceType, Technology, Screen, delta1, P, N, Data, Sol)
2
3 Vs = Data(Sol,1);
4 Vpccs = Data(Sol,2);
5 Vpccr = Data(Sol,3);
6 Vr = Data(Sol,4);
7 Vcomp = Data(Sol,5);
8 Vcomp2 = Data(Sol,6);
9 Ipccs = Data(Sol,8);
10 Ir = Data(Sol,10);
11 Ipcr = Data(Sol,11);
12 Icomp = Data(Sol,12);
13 Icomp2 = Data(Sol,13);
14
15 Ss = Vs.*conj(Ipccs);
16 Spccs = Vpccs.*conj(Ipccs);
17 Spccr = Vpccr.*conj(Ir);
18 Spcr = Vpccr.*conj(Ipcr);
19 Sr = Vr.*conj(Ir);
20 Scomp = Vcomp.*conj(Icomp);
21 Scomp2 = Vcomp2.*conj(Icomp2);
22
23 delta = [0:(P*pi/N):P*pi-(P*pi/N)];
24 interval = 10;
25 switch(DeviceType)
26     case 'InLineDvice'
27         switch(Screen)
28             case 'Voltage'
29                 System = [Vs(1:round(N/interval):N); Vpccs(1:round(N/interval):N); Vcomp(1:round(N/interval):N); Vcomp2(1:round(N/interval):N);...
30                     Vpccr(1:round(N/interval):N); Vpccr(1:round(N/interval):N); Vr(1:round(N/interval):N)];
31                 [D,S] = meshgrid(delta(1:round(N/interval):N),[0 1 1.4 1.4 1.8 2.8]);
32                 System1 = abs(System);
33                 System2 = angle(System);
34             case 'Current'
35                 if isnan(Vr)
36                     System = [Ipccs(1:round(N/interval):N); Ipccs(1:round(N/interval):N); Icomp(1:round(N/interval):N); Icomp2(1:round(N/interval):N);...
37                         Ipcr(1:round(N/interval):N); Ipcr(1:round(N/interval):N); Ir(1:round(N/interval):N); Vr(1:round(N/interval):N)];
38                 else
39                     System = [Ipccs(1:round(N/interval):N); Ipccs(1:round(N/interval):N); Icomp(1:round(N/interval):N); Icomp2(1:round(N/interval):N);...
40                         Ipcr(1:round(N/interval):N); Ipcr(1:round(N/interval):N); Ir(1:round(N/interval):N); Ir(1:round(N/interval):N)];
41                 end;
42                 [D,S] = meshgrid(delta(1:round(N/interval):N),[0 1 1.4 1.4 1.8 2.8]);
43                 System1 = abs(System);
44                 System2 = angle(System);
45             case 'Power'
46                 System = [Ss(1:round(N/interval):N); Spccs(1:round(N/interval):N); Scomp(1:round(N/interval):N); Scomp2(1:round(N/interval):N);...
47                     Spccr(1:round(N/interval):N); Spccr(1:round(N/interval):N); Spccr(1:round(N/interval):N); Sr(1:round(N/interval):N)];
48                 [D,S] = meshgrid(delta(1:round(N/interval):N),[0 1 1.4 1.4 1.8 2.8]);
49                 System1 = real(System);
50                 System2 = imag(System);
51         end;
52     otherwise
53         switch(Screen)
54             case 'Voltage'
55                 System = [Vs(1:round(N/interval):N); Vpccs(1:round(N/interval):N); Vpccr(1:round(N/interval):N); Vpccr(1:round(N/interval):N);...
56                     Vr(1:round(N/interval):N)];
57                 [D,S] = meshgrid(delta(1:round(N/interval):N),[0 1 1.4 1.8 2.8]);
58                 System1 = abs(System);
59                 System2 = angle(System);
60             case 'Current'
61                 if isnan(Vr)
62                     System = [Ipccs(1:round(N/interval):N); Ipccs(1:round(N/interval):N); Ipccs(1:round(N/interval):N);...
63                         Ipcr(1:round(N/interval):N); Ipcr(1:round(N/interval):N); Ir(1:round(N/interval):N); Vr(1:round(N/interval):N)];
64                 else

```

ADDENDUM B PROGRAM LISTINGS

```

65         System = [Ipcos(1/round(N/interval):N)]; Ipcos(1/round(N/interval):N)]; Ipcos(1/round(N/interval):N)];...
66         Ipcer(1/round(N/interval):N)]; Ipcer(1/round(N/interval):N)]; Ir(1/round(N/interval):N)]; Ir(1/round(N/interval):N)];
67     end;
68     [D,S] = meshgrid(delta(1/round(N/interval):N),[0 1 1.4 1.4 1.8 1.8 2.8]);
69     System1 = abs(System);
70     System2 = angle(System);
71     case 'Power'
72         System = [Ss(1/round(N/interval):N)]; Spcos(1/round(N/interval):N)]; Spcos(1/round(N/interval):N)]; Scomp(1/round(N/interval):N)];...
73         Spcer(1/round(N/interval):N)]; Spcer(1/round(N/interval):N)]; Spcl(1/round(N/interval):N)]; Sr(1/round(N/interval):N)];
74     [D,S] = meshgrid(delta(1/round(N/interval):N),[0 1 1.4 1.4 1.8 1.8 2.8]);
75     System1 = real(System);
76     System2 = imag(System);
77 end;
78 end;
79 YTickLabels = ['S-end'; 'PS '; 'PR '; 'PL '; 'R-end'];
80 YTicks = [0 1 1.4 1.8 2.8];
81 AxisLimits = [0 pi 0 2.8 -2 2];
82
83 axes(findobj(findobj('Tag','VRegFigure'),'Tag','ActiveAxes'));
84 cla;
85 surfH=surf(D,S,System1);
86 set(gca,'Tag','ActiveAxes',...
87     'YTickLabel',YTickLabels,...
88     'YTick',YTicks,...
89     'XTick',[0 pi/2 pi],...
90     'XTickLabel',{'0 ','pi/2','pi '});
91 h=xlabel('\delta [rad]');
92 colormap('autumn')
93 shading interp
94 set(surfH,'EdgeColor',[0 0 0]);
95 currAxisLim = axis;
96 axis([currAxisLim(1) currAxisLim(2) AxisLimits(3) AxisLimits(4) currAxisLim(5) currAxisLim(6)])
97 view(60,14)
98 switch(Screen)
99     case 'Voltage'
100         title('Voltage Magnitude');
101         zlabel('Magnitude [p.u.]');
102     case 'Current'
103         title('Current Magnitude');
104         zlabel('Magnitude [p.u.]');
105     case 'Power'
106         title('Active Power');
107         zlabel('Power [p.u.]');
108 end;
109
110 axes(findobj(findobj('Tag','VRegFigure'),'Tag','ReactiveAxes'));
111 cla;
112 surfH=surf(D,S,System2,...
113     'EdgeColor',[0 0 0]);
114 set(gca,'Tag','ReactiveAxes',...
115     'YTickLabel',YTickLabels,...
116     'YTick',YTicks,...
117     'XTick',[0 pi/2 pi],...
118     'XTickLabel',{'0 ','pi/2','pi '});
119 h=xlabel('\delta [rad]');
120 colormap('autumn')
121 shading interp
122 set(surfH,'EdgeColor',[0 0 0]);
123 currAxisLim = axis;
124 axis([currAxisLim(1) currAxisLim(2) AxisLimits(3) AxisLimits(4) currAxisLim(5) currAxisLim(6)])
125 view(60,14)
126 switch(Screen)
127     case 'Voltage'
128         title('Voltage Angle');
129         zlabel('Angle [rad]');
130     case 'Current'
131         title('Current Angle');

```

ADDENDUM B PROGRAM LISTINGS

```

132     xlabel('Angle [rad]');
133 case 'Power'
134     title('Reactive Power');
135     xlabel('Power [p.u.]');
136 end;
137
138 set(findobj('String','0'),'Visible','Off');
139
140 if get(findobj(findobj('Tag','CircuitFigure'),'Tag','HighResFiguresCheck'),'Value')
141     fsize = 36;
142     fig = findobj('Tag','3dMagFigure');
143     if isempty(fig)
144         figure
145         set(gcf,'IntegerHandle','off',...
146             'Name','3d Magnitude Graph',...
147             'NumberTitle','off',...
148             'Tag','3dMagFigure');
149     else
150         figure(fig)
151     end;
152     set(gcf,'Position',[1 29 1024 672]);
153     set(gca,'Position',[0.17 0.1 0.7 0.75]);
154     surfH=surf(D,S,System1);
155     set(gca,'YTickLabel',YTickLabels,...
156         'YTick',YTicks,...
157         'XTick',[0 pi/2 pi],...
158         'XTickLabel',{'0 ','pi/2','pi'});
159     h=xlabel('\delta [rad]');
160     set(h,'FontSize',fsize*0.8);
161     colormap('autumn')
162     shading interp
163     set(surfH,'EdgeColor',[0 0 0]);
164     currAxisLim = axis;
165     axis([currAxisLim(1) currAxisLim(2) AxisLimits(3) AxisLimits(4) currAxisLim(5) currAxisLim(6)])
166     view(60,14)
167     switch(Screen)
168     case 'Voltage'
169         ht=title('Voltage Magnitude');
170         hz=xlabel('Magnitude [p.u.]');
171     case 'Current'
172         ht=title('Current Magnitude');
173         hz=xlabel('Magnitude [p.u.]');
174     case 'Power'
175         ht=title('Active Power');
176         hz=xlabel('Power [p.u.]');
177     end;
178     set(ht,'FontSize',fsize);
179     posz = get(hz,'Position');
180     set(hz,'FontSize',fsize*0.8,...
181         'Position',[posz(1) posz(2) posz(3)]);
182     set(gcf,'Color',[1 1 1]);
183     set(gca,'FontSize',fsize*0.7);
184
185     fig = findobj('Tag','3dAngFigure');
186     if isempty(fig)
187         figure
188         set(gcf,'IntegerHandle','off',...
189             'Name','3d Angle Graph',...
190             'NumberTitle','off',...
191             'Tag','3dAngFigure');
192     else
193         figure(fig)
194     end;
195     set(gcf,'Position',[1 29 1024 672]);
196     set(gca,'Position',[0.17 0.1 0.7 0.75]);
197     surfH=surf(D,S,System2,...
198         'EdgeColor',[0 0 0]);

```

```
199 set(gca,'YTickLabel',YTickLabels,...
200     'YTick',YTicks,...
201     'XTick',[0 pi/2 pi],...
202     'XTickLabel',{'0 ','pi/2','pi '});
203 h=xlabel('\delta [rad]');
204 set(h,'FontSize',fsize*0.8);
205 colormap('autumn')
206 shading interp
207 set(surfH,'EdgeColor',[0 0 0]);
208 currAxisLim = axis;
209 axis([currAxisLim(1) currAxisLim(2) AxisLimits(3) AxisLimits(4) currAxisLim(5) currAxisLim(6)])
210 view(60,14)
211 switch(Screen)
212     case 'Voltage'
213         ht=title('Voltage Angle');
214         hz=zlabel('Angle [rad]');
215     case 'Current'
216         ht=title('Current Angle');
217         hz=zlabel('Angle [rad]');
218     case 'Power'
219         ht=title('Reactive Power');
220         hz=zlabel('Power [p.u.]');
221 end;
222 set(ht,'FontSize',fsize);
223 posz = get(hz,'Position');
224 set(hz,'FontSize',fsize*0.8);
225 set(gcf,'Color',[1 1 1]);
226 set(gca,'FontSize',fsize*0.7);
227 set(findobj('String','0'),'Visible','Off');
228 figure(findobj('Tag','VRegFigure'))
229 end
```