The Thesis Committee for Naveen Ganta
Certifies that this is the approved version of the following thesis:

Evaluation of the Effectiveness of Smart Inverters in Mitigating Voltage Variations and Fluctuations

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Surya Santoso

Gary Hallock

Evaluation of the Effectiveness of Smart Inverters in Mitigating Voltage Variations and Fluctuations

by

Naveen Ganta, B.Tech.

Thesis

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in Engineering

The University of Texas at Austin
May 2017

Acknowledgements

I would like to express my deepest gratitude to my advisor, Dr. Surya Santoso, for his exceptional guidance, patience and providing me with an excellent atmosphere for doing research. I am deeply indebted to him for his understanding, constant care, and encouragement by steering me in the right direction throughout my graduate studies. I would also like to thank Dr. Gary Hallock as the second reader of this thesis, and I am grateful for his valuable comments on this thesis.

This study and research wouldn't have been smooth without the inputs from passionate peers of L-ASPIRES (Laboratory for Advanced Studies in Power systems and Integrated Renewable Energy Sources) group. I would like to thank Harsha Vardhana Padullaparti, Piyapath Siratarnsophon, Sundaravaradan Navalpakkam Ananthan, Suma Jothibasu, Quan Nguyen Huy, Tuan A Ngo, and Min Lwin for their help and support during my graduate study.

I would also like to thank Pisitpol Chirapongsananurak and Miguel Eduardo Hernandez Figueredo for their inspirational attitude and support during my research.

Finally, I express my profound gratefulness to my parents for providing me with unwavering love, unfailing and continuous support in every stage of my life.

Abstract

Evaluation of the Effectiveness of Smart Inverters in Mitigating Voltage Variations and Fluctuations

Naveen Ganta, MSE
The University of Texas at Austin, 2017

Supervisor: Surya Santoso

With the increasing integration level of photovoltaics (PV) generation into the distribution grid, there would be detrimental impacts on residential customers in the form of rapid voltage variations and overvoltages. These voltage disturbances are caused by intermittent and varying power injections of PV generation. They can be reduced with the installation of smart inverters alongside PV with appropriate control settings. The smart inverters monitor and manage the reactive power exchange between the distribution circuit and the PV system so as to mitigate the adversity of voltage impacts.

This thesis analyzes the effects of PV generation in regards to range and variation of voltage in a real-world distribution system. It also evaluates the effectiveness of PV generation equipped with smart inverters in mitigating these issues. A detailed study is conducted on few control strategies that would mitigate increasing range and variability of voltages. This study helps determine suitable control settings that increase the effectiveness of the discussed control functions and improve the overall performance of the distribution system.

Table of Contents

List of Tables

List of Figures

List of Illustrations

1. INTRODUCTION

1.1 MOTIVATION AND BACKGROUND

With the increase in demand for clean and green power, several energy utilities are adapting renewable energy in their generation portfolio. For example, California's Renewables Portfolio Standard (RPS) requires the state to deliver at least 33 percent of its electricity from eligible renewable energy resources by 2020. Gigawatts of solar have already been added to California grid. But as more and more PV is added, the grid must become smarter to accommodate the increased capacity. The inverters of the PV systems are regarded as the tools to provide this expanded grid functionality making them smart. Rule 21 of California Public Utility Commission (CPUC) deals with the interconnection, operating and metering requirements for generating facilities such as PV connected to utility's distribution system [1]. The IEEE 1547a, a standard for interconnecting distributed energy resources with electric power systems has been amended to allow the inverter-based generation to actively participate in voltage regulation. Smart Inverter Working Group (SIWG) which is formed by CPUC, California Energy Commission and other organizations including inverter manufacturers have been working on revising rule 21 of CPUC in improving communications and automated control to manage voltage fluctuations that could result from distributed energy resources (DER) such as PV [2]. They are currently implementing smart inverters in a phased process in meeting the demands of its policies to increase the percentage of integrated renewable energy and reach their solar installation targets. The United States is experiencing persistent growth in the total PV capacity with total installations reaching almost 15,000 MW by the end of the year 2016 [3]. This larger reliance on variable, weather-dependent and distributed generation in place of the traditional sources introduce complications in the operation of distribution systems. With such high penetration of distributed energy resources (DER), voltage and reactive power management are turning increasingly complex particularly at the distribution level. It is becoming necessary for the PV inverters to provide grid support alongside the existing traditional regulating equipment such as load tap changing transformers, voltage regulators and switching capacitors which are usually integrated on the primary side of the distribution network.

While the use of smart inverters is on a rise to meet the power quality issues, it is necessary to adopt a suitable control strategy that is efficient in reacting power management alongside the benefit of solar power to fully achieve the benefits of distribution feeder performance. Furthermore, the settings would produce varying results with varying solar conditions or load level and also the location of integrated PV.

1.2 OBJECTIVE

The objective of this study is initially to determine the factors that affect the voltage and power quality issues due to the integration PV systems and then to overcome these issues using smart inverters. The study is aimed at mitigating the under and over voltage issues due to various PV penetration levels. It also aims at regulating and smoothing the voltage profile on the customer side of the distribution network through control functions of smart inverters that manage reactive power exchange efficiently. The impact of three control functions of smart inverter namely volt-var control (VVC), adaptive volt-var control (AVVC) and dynamic reactive current control (DRCC) are studied in detail for different PV penetration levels and solar conditions. The effectiveness of these smart inverter control functions to mitigate the voltage issues are analyzed to determine the suitable control function for a given distribution network. A combined control functionality is also discussed and implemented to see its effectiveness in regulating voltage through reactive power management.

1.3 ORGANIZATION

The impact of integrating PV into the distribution network and the effectiveness of different smart inverter control functions in mitigating the voltage related issues caused by the penetration of PV is studied in this report. The work presented in this report is organized in different chapters as follows. The motivation and background of the study along with defined objectives are introduced in chapter 1.

Chapter 2 details the conversion process in setting up and modeling a real-world distribution network based on the data provided by the utility. This chapter addresses the modeling of distribution network elements in OpenDSS [4], an Open source electric power Distribution System Simulator from Synergi Electric [5], a commercial software for power distribution system planning and analysis for the electric grid. The conversion process, validated by conducting short-circuit and load flow analysis is also presented in this chapter.

Chapter 3 describes the real-world residential distribution network used to perform this study. The base case for reference of analysis is also evaluated through a snapshot and time-series simulation of the peak loading day.

Chapter 4 explains the benefits of the smart inverters in supporting the reactive power management controller existing in the distribution network. A detailed explanation on each of the control function implemented through smart inverters in this study is presented in this chapter. A mathematical analysis supported by quantitative results are presented on voltage variation due to the integrated PV listing out the parameters accountable for the impact. The objective and approach of performing the study in reference to the chosen practical residential network are discussed followed by defining the metrics used to conduct the analysis. Four different scenarios that vary in solar conditions and PV integration level are chosen to study. The effectiveness of the smart inverter control functions in mitigating the voltage related issues that are generated due to the integration of PV are analyzed in the chosen scenarios. The results are compared and analyzed in determining the effective control setting for the residential distribution network.

Chapter 5 summarizes the study by comparing the effectiveness in mitigating voltage related issues with each control function in all the scenarios studied and demonstrates the rationale behind choosing the effective smart inverter control setting. The study is then concluded with insights into the procedure to follow in determining the optimal smart invert control function for an improved performance of the integrated PV in a generic distribution network.

2. DATA CONVERSION FROM SYNERGI TO OPENDSS TO DEVELOP DISTRIBUTION NETWORK

In today's world of digitalization and automation, both medium and large electrical utilities have started using sophisticated software to monitor, analyze and have a centralized control of the distribution system and its customers. One such simulation tool is Synergi Electric. The distribution system model provided are files exported from Synergi Electric simulation tool. However, only the primary circuit is modeled in Synergi Electric and the secondary system i.e., the secondary transformers, service lines and the loads are not modeled. Renewable energy resources such as PV and energy storage systems are to be integrated on the secondary of the distribution network and so explicit modeling of secondary circuit is required for this analysis. Therefore, the distribution network in Synergi Electric is to be modeled in OpenDSS along with secondary circuit elements to study and demonstrate the effect of various voltage control techniques on smart inverters. In order to model and simulate such devices, the data of distribution system which is modeled in Synergi Electric is exported and converted to OpenDSS maintained by Electric Power Research Institute (EPRI). This chapter provides an outline of the conversion process of modeling a distribution system from Synergi Electric to OpenDSS. The accuracy of the developed OpenDSS circuit model is then validated by comparing short-circuit and load flow results with that of original Synergi Electric model.

2.1 SYNERGI ELECTRIC - DISTRIBUTION PLANNING AND ANALYSIS FOR ELECTRIC GRID

Synergi Electric [6], electrical simulation software models and analyses electric utilities and power distribution systems in a real world spatial environment in full detail from the substation to the residential customer. Synergi Electric provides power engineers the flexibility to model their power distribution systems over a 10-year period down to one second on radial, looped and mesh network systems on multiple voltages and configurations. It helps a power utility in its main tasks like utility planning, operation, protection, and in maintaining a safe and reliable system. The software provides an intuitive engineering environment for working with feeders, networks, and substations and is built around detailed models of real-world facilities, customer loads, protective devices and reliability information. Synergi Electric also facilitates advanced modeling solutions for regulations, distributed generation and extreme weather events.

Synergy Electric has a wide variety of tools and engineering applications operating off a single model and database. The software can perform through different perspectives and analysis results in various functional areas through following modules:

- Cable Ampacity
- Customer Management
- Motorstart
- Protection Module
- Reliability
- Middlelink
- Switching and Contingency

The key benefits of Synergi Electric is that real-time data can be imported from GIS (Geographic Information System), CIS (Customer Information System), SCADA (Supervisory Control and Data Acquisition) and provide a programming platform through COM interface to support client-specific analysis, automate planning and operations analysis and to publish model results to MS Access Database, SharePoint or web dashboards.

The distribution network provided for the study is an exported results file from Synergi Electric in .mdb format (Microsoft Access Database file). Though Synergi

Electric has a wide variety of applications, to conduct a study on the behavior of smart inverters under the influence of volt-var techniques, a more flexible software in terms of modeling devices and its control objects is needed. To install the PV systems and to have smart inverters with advanced volt-var controls, the distribution network is converted from Synergi Electric to OpenDSS.

2.2 OPENDSS – OPEN DISTRIBUTION SYSTEM SIMULATOR

The Open Distribution System Simulator (OpenDSS, or simply, DSS) is a comprehensive electrical power system simulation tool primarily for electric utility power distribution systems. It supports nearly all frequency domain (RMS steady-state) analyses commonly performed on electric utilities and many new types of analyses that are designed to meet future needs many of which are dictated by the advent of "smart grid". DSS was developed to provide a very flexible research platform and a foundation for special distribution analysis applications such as distributed generation analysis. Other features support energy efficiency analysis of power delivery, smart grid applications, and harmonics analysis.

A major strength of OpenDSS lies is in its "quasi-static" solution modes which lend themselves well to sequential time simulations. OpenDSS was designed to perform yearly, daily, and duty cycle simulations. OpenDSS is capable of performing power flow calculations and study harmonics, dynamics, and faults in a power system. Scripting is accomplished by creating scripts in files and by driving the OpenDSS from another program using the Component Object Model (COM) interface. The COM interface allows you to write some code in Excel, VBA, MATLAB, Python, R, etc. and make the OpenDSS do what you want it to do. This provides powerful external analytical capabilities as well as excellent graphics for displaying results.

The diverse capabilities of OpenDSS range from general distribution planning and analysis, to renewable integration and load/storage simulators. In 2008, EPRI released the program under an open source license, meaning that additional functionality can be added as necessary to support new developments and technologies. One of the unique aspects of OpenDSS is that it never makes internal simplifications related to phase balance or symmetrical components. In this study, the COM interface is used with MATLAB to compile and run OpenDSS files in order to compare the performance of various PV inverter settings for regulating local voltage. The following sections describe the structure of OpenDSS models, the solution process, and an example model of a distribution circuit in the United States.

2.3 CONVERSION FROM SYNERGI ELECTRIC TO OPENDSS

Conversion from Synergi Electric to OpenDSS is a two-step process. The initial step is to export the circuit data containing all the system parameters and results of load flow and short circuit analyses of the distribution system under Synergi Electric into a Microsoft access database file which are then used to define OpenDSS files in .dss format through an automated Visual Basic for Applications (VBA) scripting tool that accesses data from database file. Figure 2-1 describes the conversion process of modeling distribution network in OpenDSS from Synergi Electric.

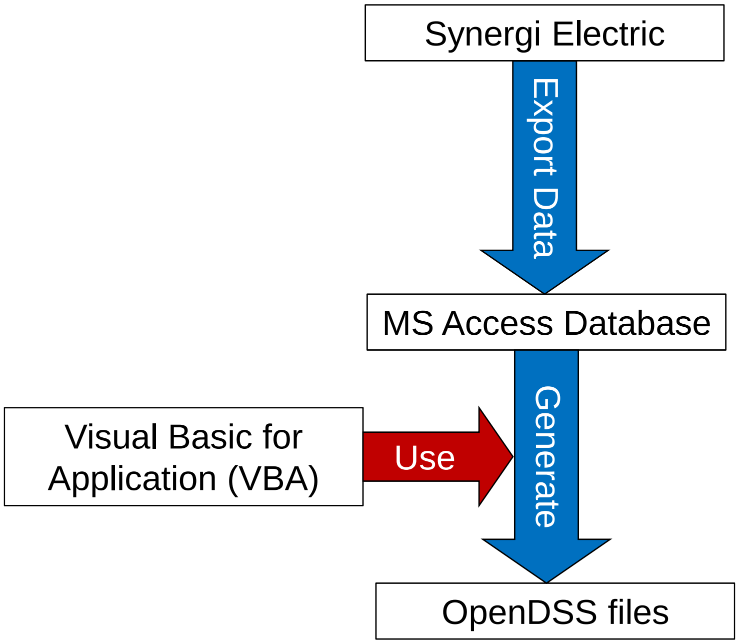


Figure 2-1: Flowchart of data conversion from Synergi Electric to OpenDSS.

2.3.1 Exporting data from Synergi Electric to Microsoft Access Database

One of the key advantages of Synergi Electric is to easily export the model results to a variety of popular tools like web, DXF, GIS, MS Excel, SharePoint or a dashboard [7]. SharePoint is a web-based application that integrates with Microsoft Office. Microsoft Access is an extra feature in SharePoint enterprise through which results and data of distribution can be exported for analysis. The data obtained so would be available in a .mdb format. The distribution system data is exported in Microsoft access database format because it helps to develop application software supported by Visual Basic for Application (VBA), an object-based programming language that can reference a variety of objects including DAO (Data Access Objects), ActiveX data objects and many other ActiveX components.

2.3.1.1 Overview of Microsoft Access Database

A Microsoft access database file would be having different access objects that help store, modify, process, automate, and interact with the user. This section briefly explains about the functions and uses of these access objects.

- **Tables:** All the data is stored in the tables. The columns of the tables are referred to as “Fields” which can be set to various data types.
- **Queries:** These are used to find and operate on the data present in the tables. These objects help match, sort, edit and combine data from different tables based on defined criteria.
- **Forms:** These are screens for displaying data from and inputting data into your tables.
- **Reports:** These are screens for displaying data to and outputting data from the database.
- **Macros:** This object is a script for doing specific tasks.
- **Modules:** This access object helps to define own functions and programs and has multi-user interface. It is mainly used to automate a repetitive process that deals with accessing, modifying the data in the access database.

2.3.2 Converting data in Microsoft Access database to OpenDSS

This section describes the conversion of circuit data in Microsoft access database to OpenDSS files using automated Visual Basic for Application (VBA) scripting tool. The VBA programming and steps for creating different network components in OpenDSS are explained in detailed in this section.

2.3.2.1 Visual Basic for Applications (VBA)

Visual Basic for Applications (VBA) is an implementation of Microsoft's event-driven programming language Visual Basic 6 and its associated integrated development environment (IDE), which are built into most Microsoft Office applications. One of the main reason in considering VBA programming in office is because it is highly effective and efficient when it comes to automation and repetition. As an example, for a practical distribution circuit, the number of devices may range from about 2000 to 6000 depending on whether it is supporting an industrial or a residential area. With such a vast data of network in Microsoft access database file, to handle, fetch and format data into various OpenDSS files need a software that is capable of automating a highly repetitive process which is the fulfilled by VBA programming. So, VBA programming is used for translating the data in MS access tables into DSS files in this study.

Basic VBA programming structures required for translating the circuit data to DSS files is discussed in brief in this section. A module is inserted in the visual basic window of the current MS Access database which acts as a platform for VBA scripting that accesses the data. Various subroutines and functions are written inside the created module which accesses the tables (also referred as records) and queries in the database file to translate corresponding circuit component data into OpenDSS files.

1. **Subroutine:** A subroutine is a part of code that performs a specific task and does not return a result. Subroutines are mainly used to break down large sections of code into small manageable parts. “Sub” keyword is used to declare a subroutine followed by the name of the subroutine. A subroutine can be assigned access level by declaring with keywords “Public” or “Private”. A Public subroutine can be called by the routines of the same module as well as by the routines of other

modules. “End Sub” is the keyword used to end the body of the subroutine. Following is the syntax of a subroutine:

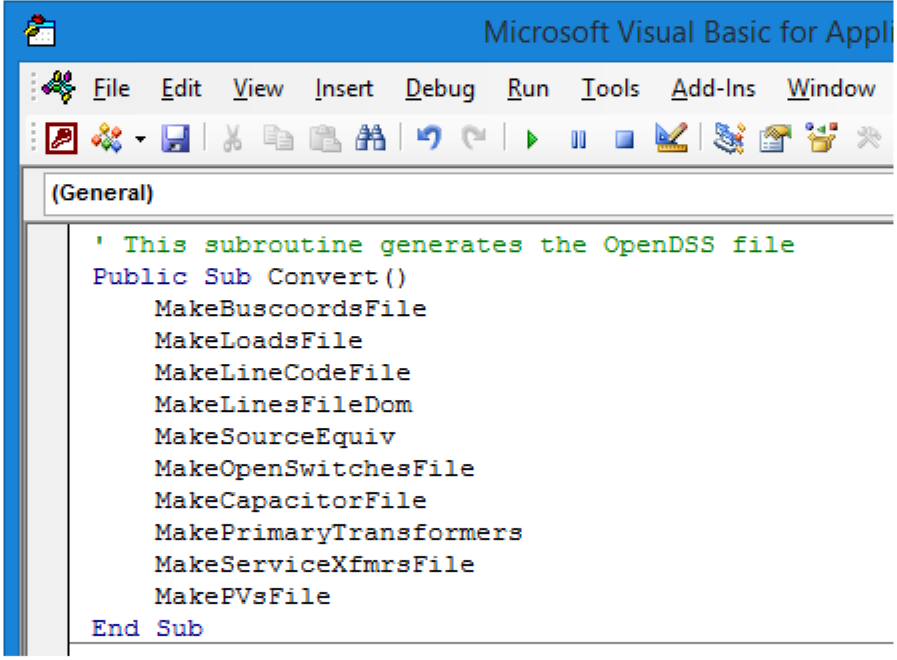
```
Public Sub mysubroutine()
…..
…..
End Sub
```

2. **Function:** A function is a part of code that performs a specific task and returns a result. Functions are mainly used to carry out repetitive tasks such as formatting data for output. SA function is similar to that of a subroutine but with a few more rules. “Function” keyword is used to declare a function followed by the name of the function. The creation of a function starts with the Function keyword and closes with End Function. Here is the syntax of a function:

```
Function myfunction()
…..
…..
End Function
```

2.3.2.2 Automating VBA for generating OpenDSS (.dss) files

In order to create a particular DSS file, one or more tables (records) across the database are to be accessed on a repetitive approach. And, the retrieved data should be made readable to DSS so that the modeled distribution system or the circuit executes without any errors. This section describes various VBA programming structures written for our application to edit the data into a readable format and to generate DSS files to model the distribution system. Specific subroutines for creating independent DSS files for bus coordinates, lines, service transformers, loads etc. are written. The central subroutine named Convert (), as shown in Figure 2-2 calls all the relevant subroutines and functions within, to generate DSS file and automate the process of modeling similar elements of the circuit into that file. As explained in the previous section, functions are used to edit the data to a DSS readable format and the subroutines are used to collect, combine and generate DSS files to model the distribution system.



```
' This subroutine generates the OpenDSS file
Public Sub Convert()
    MakeBuscoordsFile
    MakeLoadsFile
    MakeLineCodeFile
    MakeLinesFileDom
    MakeSourceEquiv
    MakeOpenSwitchesFile
    MakeCapacitorFile
    MakePrimaryTransformers
    MakeServiceXfmrsFile
    MakePVsFile
End Sub
```

Figure 2-2: VBA code snippet of central subroutine - Convert ().

Following subroutines and functions are written for this application:

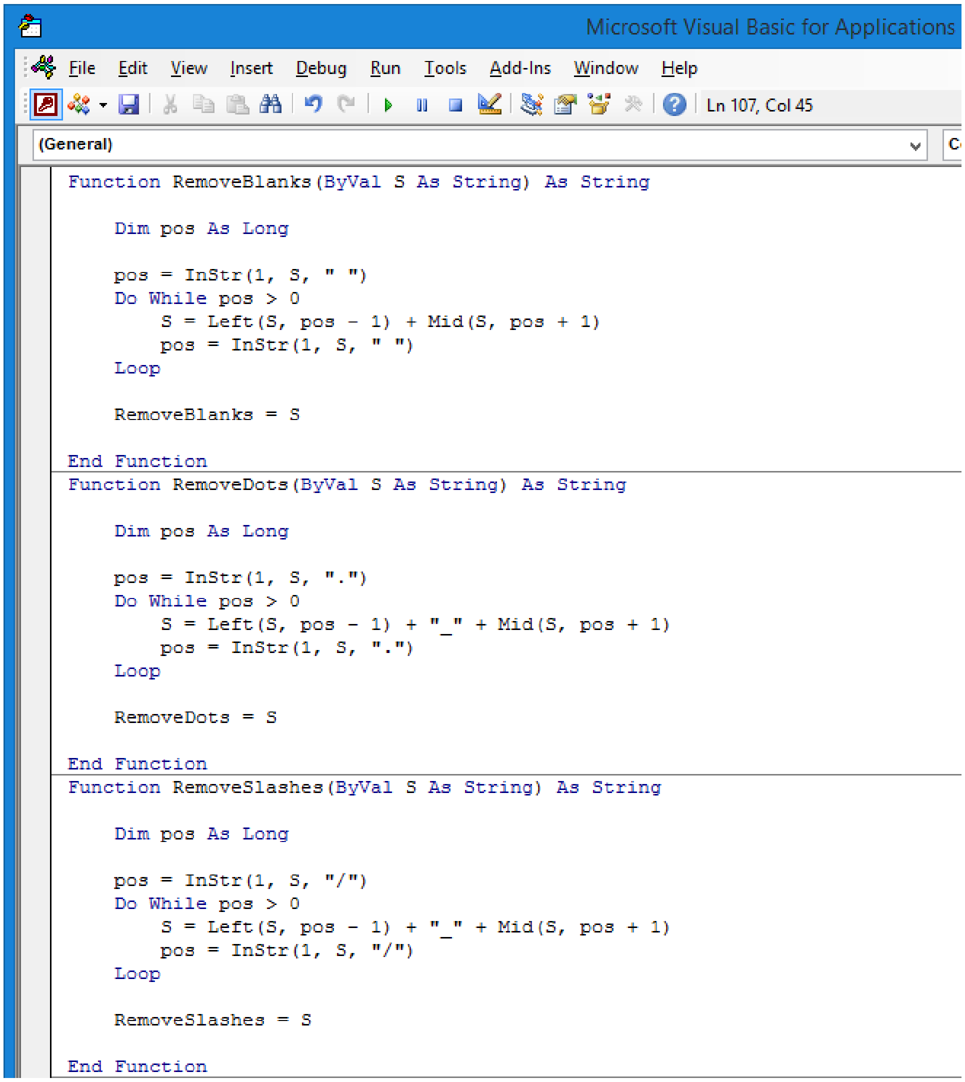
<u>Subroutines</u>

- Convert () – to call all the subroutines and functions to generate the .dss files.
- MakeBusCoordsFile() – creates DSS file that contains GIS coordinates for every node in the circuit (BusCoords.dss).
- MakeLoadsFile() – creates DSS file that contains the bus connection data and feature (power) for all the load elements in the circuit (Loads.dss).
- MakeLineCodeFile() – creates DSS file that directly contains the impedances of all service lines and feeder lines use in the circuit (Linecodes.dss).

- MakeLinesFileDom() – creates DSS file for all the feeder lines that connect substation to the primary side of service transformers in the circuit (Lines.dss).
- MakeOpenSwitchesFile() – creates DSS file containing location and status of a normally open point in the network (Switches.dss).
- MakeCapacitorFile() – creates DSS file that contains location and size of all the capacitor elements installed in the circuit (Capacitors.dss).
- MakePrimaryTransformers() – creates DSS file for all the Primary transformers in the circuit (PrimaryTransformer.dss)
- MakeServiceXfmrsFile() – creates DSS file for all the service transformers and the service lines that connect secondary distribution transformer to the loads in the circuit (xfmrs.dss and Services.dss)
- MakePVsFile() – creates DSS file for all the distributed PV elements in the circuit (PVs.dss)

Functions

- RemoveBlanks() – helps refine the data in the tables of access database to be in a DSS readable format by removing blanks (” ”).
- RemoveDots() – helps refine the data in the tables of access database to be in a DSS readable format by removing hyphen (”.”) and replacing with an underscore (”_”).
- RemoveSlashes() – helps refine the data in the tables of access database to be in a DSS readable format by removing slashes (”/”) and replacing with an underscore (”_”).
- RemoveHyphen() – helps refine the data in the tables of access database to be in a DSS readable format by removing hyphen (”-”) and replacing with an underscore (”_”).

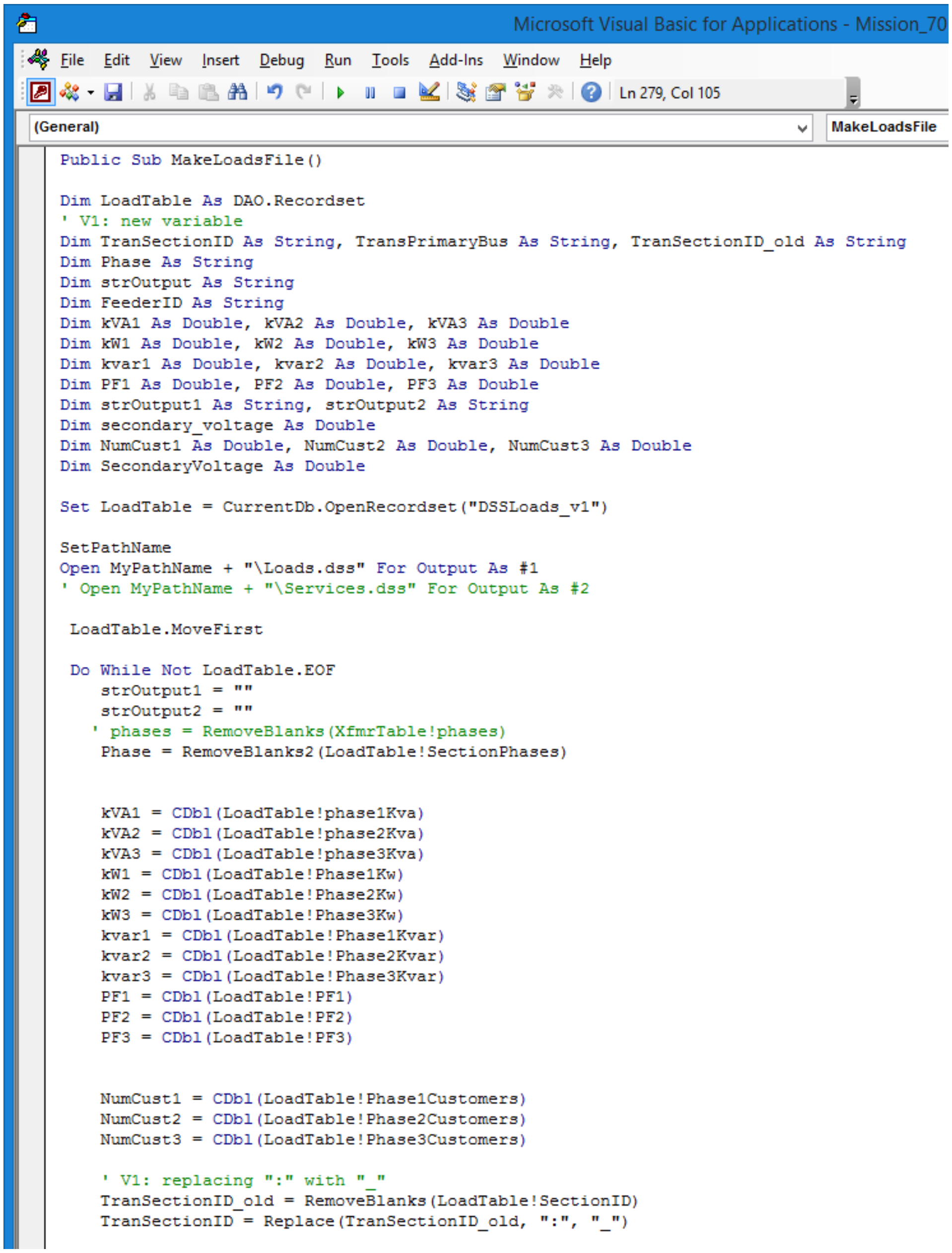


```
Function RemoveBlanks(ByVal S As String) As String

    Dim pos As Long

    pos = InStr(1, S, " ")
    Do While pos > 0
        S = Left(S, pos - 1) + Mid(S, pos + 1)
        pos = InStr(1, S, " ")
    Loop

    RemoveBlanks = S

End Function
Function RemoveDots(ByVal S As String) As String

    Dim pos As Long

    pos = InStr(1, S, ".")
    Do While pos > 0
        S = Left(S, pos - 1) + "_" + Mid(S, pos + 1)
        pos = InStr(1, S, ".")
    Loop

    RemoveDots = S

End Function
Function RemoveSlashes(ByVal S As String) As String

    Dim pos As Long

    pos = InStr(1, S, "/")
    Do While pos > 0
        S = Left(S, pos - 1) + "_" + Mid(S, pos + 1)
        pos = InStr(1, S, "/")
    Loop

    RemoveSlashes = S

End Function
```

Figure 2-3: VBA code snippet of functions.

2.3.2.3 Automation of DSS file Generation

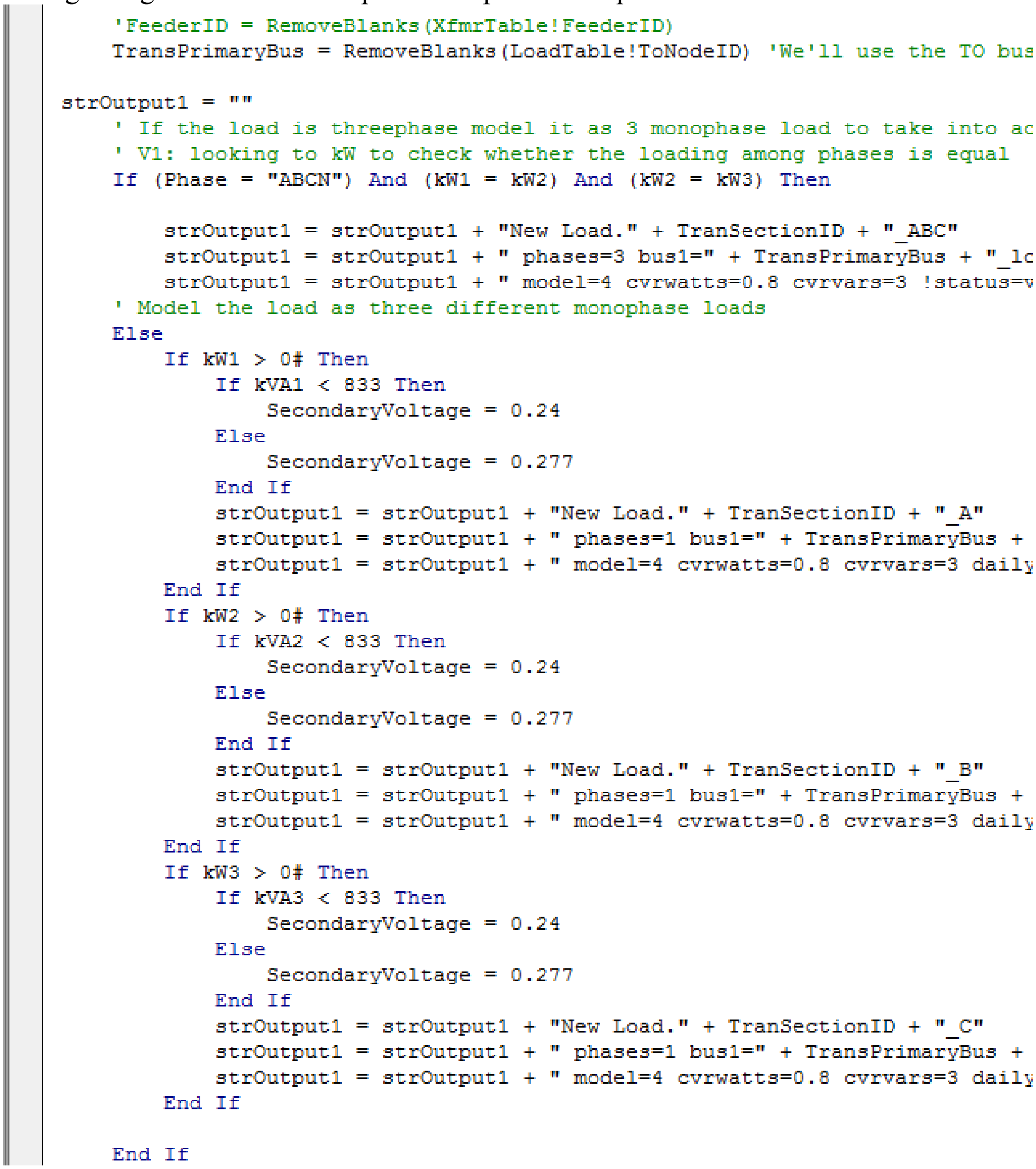
In order to understand the automation of generating DSS files, the VBA code snippets for MakeLoadFile() subroutine is shown in Figure 2-4 and Figure 2-5. The initial step in the code is to refer to an existing table in the database which would be used to fetch data to generate the required DSS file. “LoadTable” is the chosen variable name of the table in the database through “DAO.Recordset” (DAO in short) property to access data.



```
Public Sub MakeLoadsFile()

Dim LoadTable As DAO.Recordset
' V1: new variable
Dim TranSectionID As String, TransPrimaryBus As String, TranSectionID_old As String
Dim Phase As String
Dim strOutput As String
Dim FeederID As String
Dim kVA1 As Double, kVA2 As Double, kVA3 As Double
Dim kW1 As Double, kW2 As Double, kW3 As Double
Dim kvar1 As Double, kvar2 As Double, kvar3 As Double
Dim PF1 As Double, PF2 As Double, PF3 As Double
Dim strOutput1 As String, strOutput2 As String
Dim secondary_voltage As Double
Dim NumCust1 As Double, NumCust2 As Double, NumCust3 As Double
Dim SecondaryVoltage As Double

Set LoadTable = CurrentDb.OpenRecordset("DSSLoads_v1")

SetPathName
Open MyPathName + "\Loads.dss" For Output As #1
' Open MyPathName + "\Services.dss" For Output As #2

 LoadTable.MoveFirst

 Do While Not LoadTable.EOF
    strOutput1 = ""
    strOutput2 = ""
   ' phases = RemoveBlanks(XfmrTable!phases)
    Phase = RemoveBlanks2(LoadTable!SectionPhases)


    kVA1 = CDbl(LoadTable!phase1Kva)
    kVA2 = CDbl(LoadTable!phase2Kva)
    kVA3 = CDbl(LoadTable!phase3Kva)
    kW1 = CDbl(LoadTable!Phase1Kw)
    kW2 = CDbl(LoadTable!Phase2Kw)
    kW3 = CDbl(LoadTable!Phase3Kw)
    kvar1 = CDbl(LoadTable!Phase1Kvar)
    kvar2 = CDbl(LoadTable!Phase2Kvar)
    kvar3 = CDbl(LoadTable!Phase3Kvar)
    PF1 = CDbl(LoadTable!PF1)
    PF2 = CDbl(LoadTable!PF2)
    PF3 = CDbl(LoadTable!PF3)


    NumCust1 = CDbl(LoadTable!Phase1Customers)
    NumCust2 = CDbl(LoadTable!Phase2Customers)
    NumCust3 = CDbl(LoadTable!Phase3Customers)

    ' V1: replacing ":" with "_"
    TranSectionID_old = RemoveBlanks(LoadTable!SectionID)
    TranSectionID = Replace(TranSectionID_old, ":", "_")
```

Figure 2-4: VBA code snippet of MakeLoadsFile() initialization.

Visual Basic uses “DAO.Recordset” property to refer a table or a query in the database which stands for Data Access Object. A subroutine can have multiple DAOs but having a single DAO would improve computational speed.



```
    'FeederID = RemoveBlanks(XfmrTable!FeederID)
    TransPrimaryBus = RemoveBlanks(LoadTable!ToNodeID) 'We'll use the TO bus

strOutput1 = ""
    ' If the load is threephase model it as 3 monophase load to take into ac
    ' V1: looking to kW to check whether the loading among phases is equal
    If (Phase = "ABCN") And (kW1 = kW2) And (kW2 = kW3) Then

        strOutput1 = strOutput1 + "New Load." + TranSectionID + "_ABC"
        strOutput1 = strOutput1 + " phases=3 bus1=" + TransPrimaryBus + "_lc
        strOutput1 = strOutput1 + " model=4 cvrwatts=0.8 cvrvars=3 !status=v
    ' Model the load as three different monophase loads
    Else
        If kW1 > 0# Then
            If kVA1 < 833 Then
                SecondaryVoltage = 0.24
            Else
                SecondaryVoltage = 0.277
            End If
            strOutput1 = strOutput1 + "New Load." + TranSectionID + "_A"
            strOutput1 = strOutput1 + " phases=1 bus1=" + TransPrimaryBus +
            strOutput1 = strOutput1 + " model=4 cvrwatts=0.8 cvrvars=3 daily
        End If
        If kW2 > 0# Then
            If kVA2 < 833 Then
                SecondaryVoltage = 0.24
            Else
                SecondaryVoltage = 0.277
            End If
            strOutput1 = strOutput1 + "New Load." + TranSectionID + "_B"
            strOutput1 = strOutput1 + " phases=1 bus1=" + TransPrimaryBus +
            strOutput1 = strOutput1 + " model=4 cvrwatts=0.8 cvrvars=3 daily
        End If
        If kW3 > 0# Then
            If kVA3 < 833 Then
                SecondaryVoltage = 0.24
            Else
                SecondaryVoltage = 0.277
            End If
            strOutput1 = strOutput1 + "New Load." + TranSectionID + "_C"
            strOutput1 = strOutput1 + " phases=1 bus1=" + TransPrimaryBus +
            strOutput1 = strOutput1 + " model=4 cvrwatts=0.8 cvrvars=3 daily
        End If

    End If
```

Figure 2-5: VBA code snippet of MakeLoadsFile() function.

As explained in Section 2.3.1.1, a Query can be used to group all the required data in a table format from different tables which helps to avoid referrring to multiple DAOs in Visual Basic and improve the computation speed of generating DSS files. The DAO of MakeLoadFile() subroutine is "DSSLoads_v1" which has the required data to model all the load elements in OpenDSS. The detailed information about a load like kW, phases, location, etc., would be present in different columns of the query "DSSLoads_v1". A blank file "Loads.dss" is generated in the next step and would be filled repeatedly by traversing through rows of DAO (LoadTable) to the end of the file. Figure 2-6 shows the control logic behind the modeling of three-phase or monophase loads in DSS files based on the information in the DAO ("LoadTable"/ DSSLoads_v1) extracted by the function.

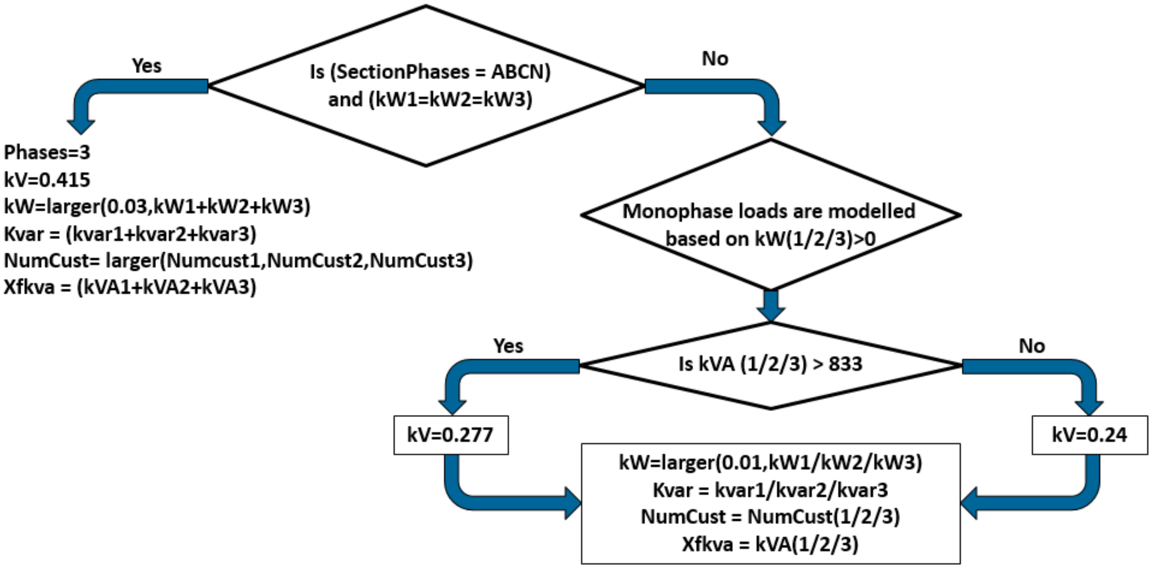
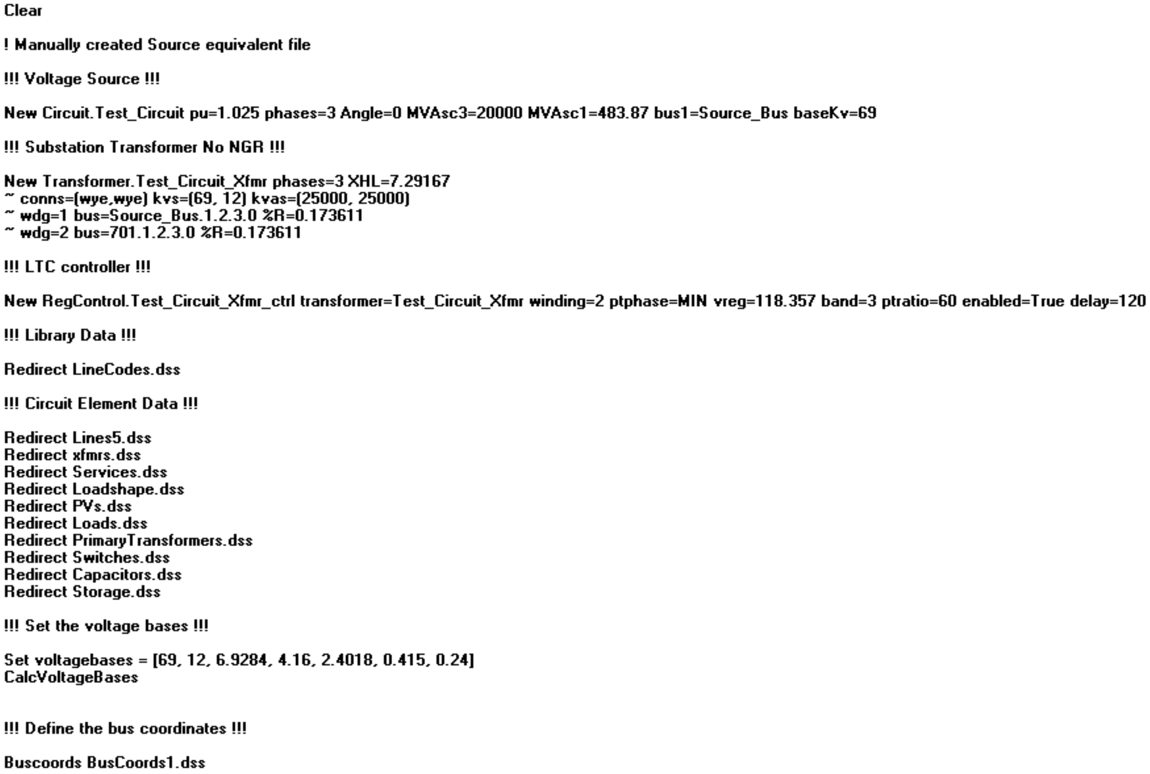


Figure 2-6: Control Algorithm of MakeLoadFile() function.

The phase and connection information about the load is extracted by selecting columns "Phase1Kw", "Phase2Kw" and "Phase3Kw" in the DSSLoads_v1 table. If it is Phase A, Phase B or Phase C, then phase information for load is written as single and the connection as ".1" for Phase A, ".2" for Phase B and ".3" for Phase C. A load with

SectionPhases = "ABCN" and have same loads on each of its phase, it is modelled as three-phase load and the load connection as “. 1.2.3”. In this similar way, the complete information of the load table is transferred to "Loads.dss".

Few OpenDSS files are to be manually generated in order to redirect and group all the generated DSS files to model the distribution circuit and to add any other circuit elements like Thevenin equivalent of the voltage source, monitors, and energy meters to monitor and simulate the modeled distribution system. Figure 2-7 describes the modeling of Master.dss file that helps define voltage source, substation transformer with LTC control (voltage regulator) and grouping of all the generated DSS files.



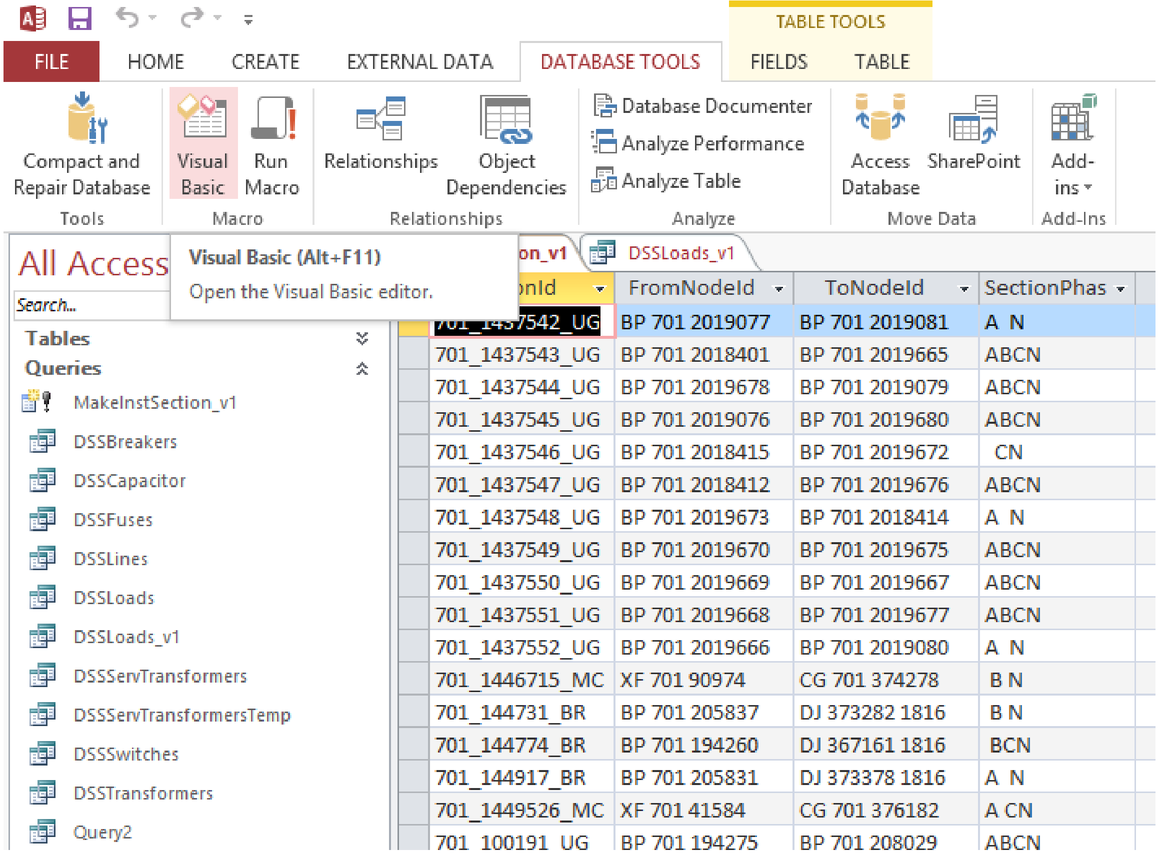
```
Clear

! Manually created Source equivalent file

!!! Voltage Source !!!

New Circuit.Test_Circuit pu=1.025 phases=3 Angle=0 MVAsc3=20000 MVAsc1=483.87 bus1=Source_Bus baseKv=69

!!! Substation Transformer No NGR !!!

New Transformer.Test_Circuit_Xfmr phases=3 XHL=7.29167
~ conns=(wye,wye) kvs=(69, 12) kvas=(25000, 25000)
~ wdg=1 bus=Source_Bus.1.2.3.0 %R=0.173611
~ wdg=2 bus=701.1.2.3.0 %R=0.173611

!!! LTC controller !!!

New RegControl.Test_Circuit_Xfmr_ctrl transformer=Test_Circuit_Xfmr winding=2 ptphase=MIN vreg=118.357 band=3 ptratio=60 enabled=True delay=120

!!! Library Data !!!

Redirect LineCodes.dss

!!! Circuit Element Data !!!

Redirect Lines5.dss
Redirect xfmrs.dss
Redirect Services.dss
Redirect Loadshape.dss
Redirect PVs.dss
Redirect Loads.dss
Redirect PrimaryTransformers.dss
Redirect Switches.dss
Redirect Capacitors.dss
Redirect Storage.dss

!!! Set the voltage bases !!!

Set voltagebases = [69, 12, 6.9284, 4.16, 2.4018, 0.415, 0.24]
CalcVoltageBases


!!! Define the bus coordinates !!!

Buscoords BusCoords1.dss
```

Figure 2-7: Example Master.dss file in OpenDSS.

Detailed VBA code for generating the OpenDSS files and the OpenDSS files corresponding to all circuit elements are discussed in the Appendix.

The following are few steps along with snapshots that summarize the process of generating DSS files from Microsoft Access Database file:

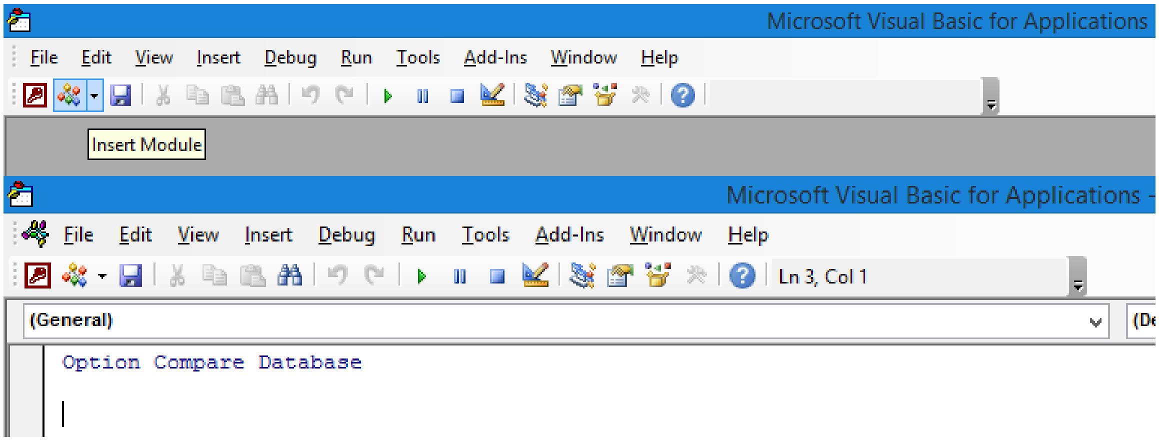
1. Open a new Visual Basic editor from the access database file generated by the Synergy Electric software which has all the data needed to model the circuit in OpenDSS (Access Database file >> Database Tools >> Visual Basic).



onId	FromNodeId	ToNodeId	SectionPhas
701_1437542_UG	BP 701 2019077	BP 701 2019081	A N
701_1437543_UG	BP 701 2018401	BP 701 2019665	ABCN
701_1437544_UG	BP 701 2019678	BP 701 2019079	ABCN
701_1437545_UG	BP 701 2019076	BP 701 2019680	ABCN
701_1437546_UG	BP 701 2018415	BP 701 2019672	CN
701_1437547_UG	BP 701 2018412	BP 701 2019676	ABCN
701_1437548_UG	BP 701 2019673	BP 701 2018414	A N
701_1437549_UG	BP 701 2019670	BP 701 2019675	ABCN
701_1437550_UG	BP 701 2019669	BP 701 2019667	ABCN
701_1437551_UG	BP 701 2019668	BP 701 2019677	ABCN
701_1437552_UG	BP 701 2019666	BP 701 2019080	A N
701_1446715_MC	XF 701 90974	CG 701 374278	B N
701_144731_BR	BP 701 205837	DJ 373282 1816	B N
701_144774_BR	BP 701 194260	DJ 367161 1816	BCN
701_144917_BR	BP 701 205831	DJ 373378 1816	A N
701_1449526_MC	XF 701 41584	CG 701 376182	A CN
701_100191_UG	BP 701 194275	BP 701 208029	ABCN

Figure 2-8: Creating a new Visual Basic editor from Access Database file.

2. Create a module in the Visual Basic editor to develop the program to fetch, arrange and edit the information provided in the database file to generate OpenDSS files (Visual Basic >> Insert Module >> Module).



```
Option Compare Database

```

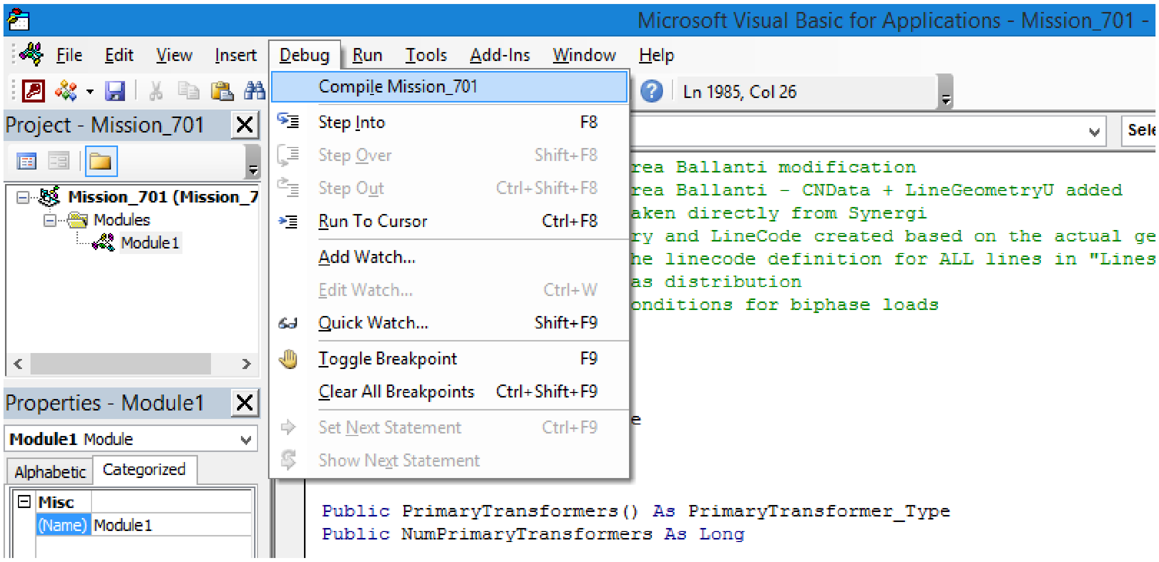
Figure 2-9: Creating a new module for VBA scripting.

3. Once the module is created, starting with setting the path for the generated files to store, write the required subroutines and functions to generate loads, lines, transformers or any other files and a master file to define the source, substation transformer and to compile all the files.

```
Sub SetPathName()
    MyPathName = "C:\Users\Naveen\Desktop\Naveen\Projects\Thesis\OpenDSSCodes"
End Sub
```

Figure 2-10: Setting the path for generated .dss files to store.

4. Compile the code containing all the functions and subroutines so as to avoid the syntax errors if any.



```
rea Ballanti modification
rea Ballanti - CNData + LineGeometryU added
aken directly from Synergi
ry and LineCode created based on the actual ge
he linecode definition for ALL lines in "Lines
as distribution
onditions for biphase loads

e


Public PrimaryTransformers() As PrimaryTransformer_Type
Public NumPrimaryTransformers As Long
```

Figure 2-11: Compiling the VBA script.

5. Run the compiled code in VBA to generate the OpenDSS files in the path specified in the code.

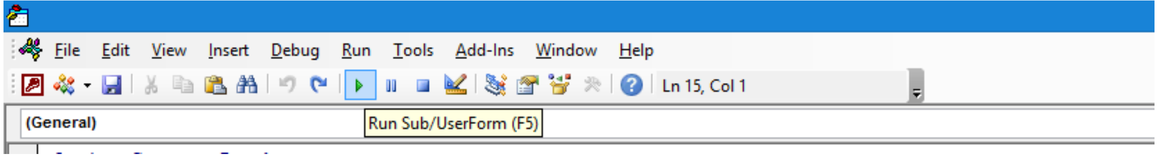


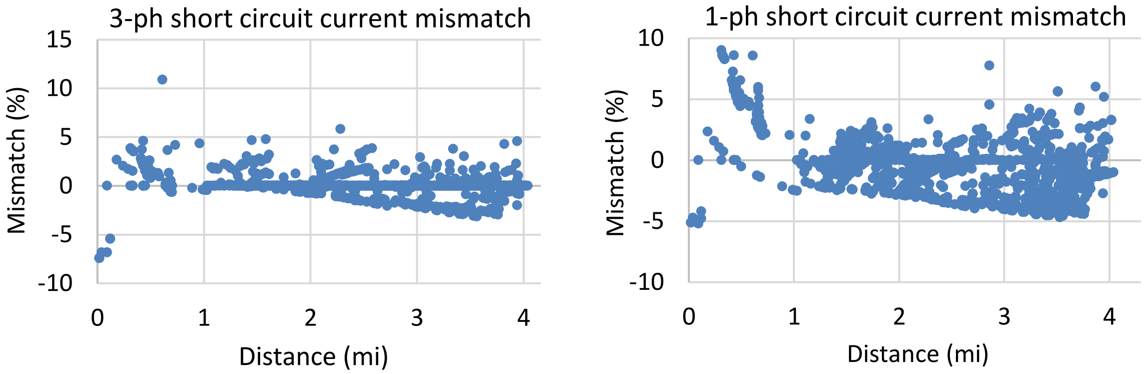
Figure 2-12: Run the VBA script to generate .dss files.

2.4 MODEL VALIDATION

This section presents the OpenDSS circuit model validation results by comparing the solutions of short-circuit (three-phase to ground and single line-to-ground faults) and load flow analyses of the OpenDSS circuit model of the distribution system with that of the original Synergi Electric circuit model. The validated model in OpenDSS would then be considered as a base case model for further simulations of various studies, scenarios, and evaluation of results. This particular real-world distribution network is based in southeastern region of the United States and other details of the circuit are not revealed.

2.4.1 Short-circuit validation

This section provides the details of validating the accuracy of lines, transformer, and equivalent source impedances modeled in OpenDSS by comparing with the short-circuit results of the circuit model in Synergi Electric. The short-circuit currents are recorded by applying both three-phase to ground and single line-to-ground faults at all the primary buses in both the OpenDSS and Synergi Electric models. Then the mismatches between the short-circuit currents obtained from both the models are computed and expressed as percentages of Synergi Electric results. The corresponding results are depicted in Figure 2-13. It is observed that the mismatches in the short-circuit currents increase as the distance of the bus from the substation increases. This is because the mismatch in the short-circuit current at a bus is also influenced by the mismatches at the upstream buses. Additionally, while the secondary circuit is not modeled in Synergi Electric, it is explicitly modeled in the OpenDSS circuit model. Furthermore, the modeling of different circuit elements including lines, transformers, and loads can differ slightly among the simulation platforms. Because of these reasons, there are slight mismatches in the short-circuit fault current results. Nevertheless, the maximum mismatches in the three-phase and single-phase short-circuit currents are limited to 10.88% and 9.01% respectively between the Synergi Electric and OpenDSS circuit models. The range of mismatches up to 15% is commonly observed in model conversion.

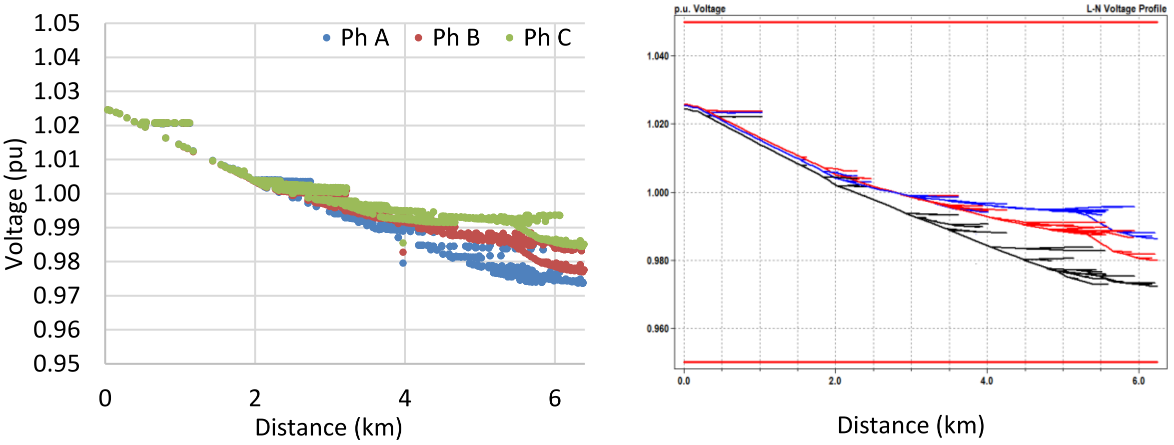


(a) (b)

Figure 2-13: Mismatches in short-circuit currents between Synergi Electric and OpenDSS circuit models (a) 3-ph short-circuit faults (b) 1-ph short circuit faults.

2.4.2 Load-Flow validation

For the purpose of load flow validation, the load flow results obtained from both Synergi Electric and OpenDSS models are compared in this section. For load flow validation, the substation voltage regulator, capacitors, switches, service lines, and loads are enabled. Generators and Fuses are disabled.



(a) (b)

Figure 2-14: Voltage profile of the distribution network in (a) Synergi Electric model (b) OpenDSS model.

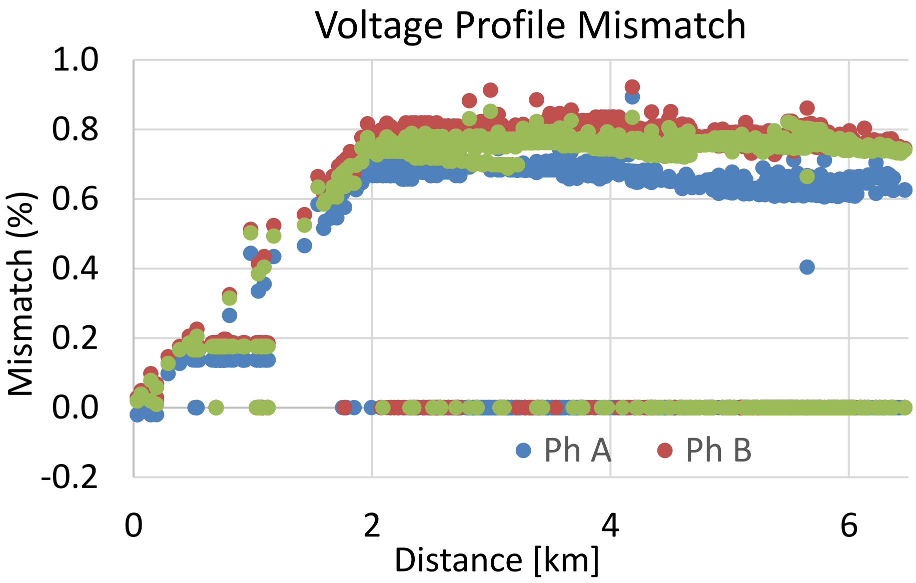


Figure 2-15: Mismatches of Voltage profiles obtained between OpenDSS and Synergi circuit models.

Voltages at all primary buses at a loading condition of the selected distribution circuit obtained in OpenDSS and Synergi Electric models are shown in Figure 2-14. Then the voltage mismatches between the two models are computed and expressed as percentages of the bus voltages recorded in Synergi model are shown in Figure 2-15. Voltage profile in the converted model in OpenDSS matches the results from the Synergi model with a maximum mismatch of 0.92%. The mismatches result from the reasons mentioned in the section 2.4.1. The voltage profile mismatches are less, thus validates the load flow results.

3. DISTRIBUTION NETWORK MODEL

The impact of integrating renewable energy resources is studied on a practical distribution network. This distribution network data is made available to public to investigate Smart Grid issues by EPRI and comes with OpenDSS software [8]. The distribution network would have been modeled in OpenDSS the similar way as explained in Chapter 2 if the raw data of the distribution network is in Synergi Electric. The major elements that exist in the modeled distribution network are listed briefly to give an estimate of the size of the network this study is carried out on. This chapter also sets the base case scenario for further analysis of the integration of renewables enabled with smart inverters. This chapter describes the real-world power distribution network model used in this study.

3.1 DISTRIBUTION NETWORK MODEL

For a better analysis and evaluation of the impact of integrated intermittent energy resources along with different control strategies, the distribution network is chosen such that the customers are predominantly residential and have a small portion of commercial customers as well. The characteristics and the base case scenario for the chosen network is described in the following sections.

3.1.1 Residential Distribution Network Model

The characteristics of the distribution network are described in this section. Figure 3-1 shows the one-line diagram of the circuit with the colors contoured in terms of the distance from the substation. The circuit has a 10 MVA 115/12.47 kV substation transformer with a peak load demand of 8.12 MVA during the entire year. The residential distribution network shown in Figure 3-1 contains 2998 buses, 3437 nodes, and supplies to 4414 devices. The total length of primary circuit compiles to be approximately 48.5

miles and to be a total of 74.5 miles including the secondary circuit. There exist four three-phase controlled switching capacitors along the feeder for both voltage regulation and power factor correction. There exist no voltage regulators in this distribution network. All the loads are modeled to be monophase and so the service transformers are also monophase with a service line connecting the secondary of the transformer to the load. The largest single-phase load present in the distribution network has a peak demand of approximately 14 kW. All the customer loads are modeled as Nominal Linear P, and Quadratic Q with Conservation Voltage Reduction (CVR) technique implemented that helps utilities in saving energy while maintaining rated voltage supplied to the customers.

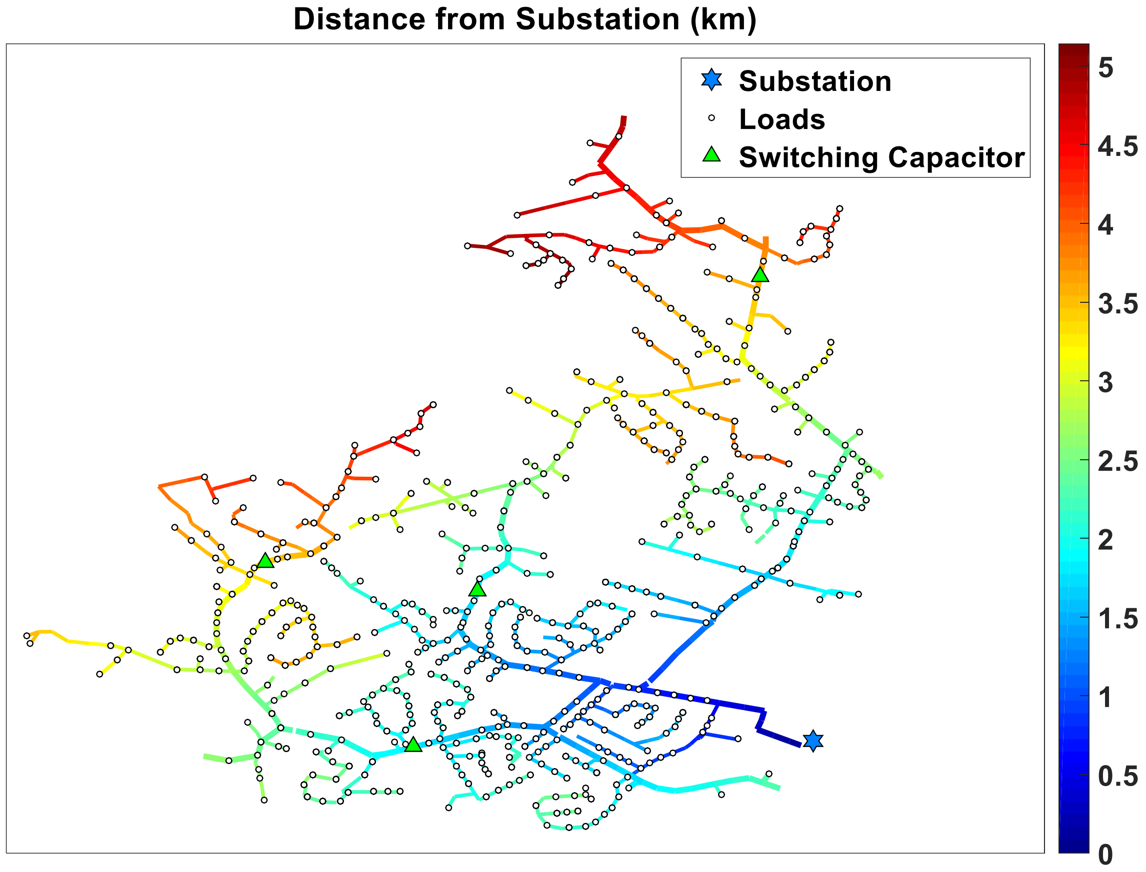


Figure 3-1: One line diagram of the residential distribution network.

3.2 BASE CASE SCENARIO

This section describes the base case scenario of the chosen distribution network that acts as a reference for the analysis. The simulations are run on modeled distribution network in OpenDSS through MATLAB interface as mentioned in Section 2.2.

3.2.1 Base Case

The base case conditions and results are detailed in this section. The base case acts as a reference for analyses of the impact of integrated renewable energy sources and the effectiveness of voltage control technologies implemented through smart inverters on the distribution network. Two simulation cases are presented in this section, i.e., snapshot simulation at the peak day and the time-series simulation during the entire peak day.

3.2.1.1 Snapshot simulation for the peak day

The snapshot simulation in this section is performed at the peak demand of 7.28 MW recorded on 52nd day of the year. Table 3-1 and Table 3-2 describe the powers at substation and capacitor status respectively. The snapshot voltage profile is shown in Figure 3-2.

Table 3-1: Base case substation power.

Substation Power	Total
Active Power (kW)	7.2813
Reactive Power (kvar)	3.5843
Apparent Power (kVA)	8.1157
Power Factor	0.897

Table 3-2: Base case reactive power injections from capacitors.

Capacitor	Reactive Power Injection (kvar)
MDV201_HN_2_116_ABC28285-1	0
MDV201_HN_2_818_ABC63707-1	0
MDV201_HN_2_345_ABC8081-1	0
MDV201_DA_8_153_ABC74433-1	0

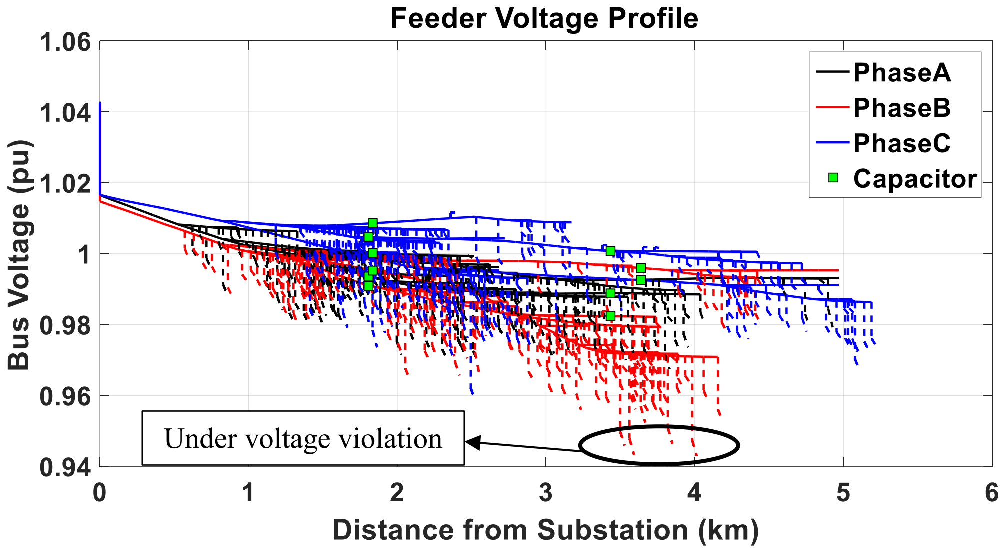


Figure 3-2: Feeder voltage profile of snapshot simulation.

It is observed that all the capacitors are switched off during the snapshot simulation. And it should be observed that there exists an under voltage violation in phase B as shown in Figure 3-2.

3.2.1.2 Time-series simulation for the peak day

The load profile as observed on the peak day which is used for time-series simulation is shown in Figure 3-3. The time-series simulation in this section is performed for a 24-hour period of the observed peak day with one-minute time-step resolutions. For each time step, the load voltages are obtained and a histogram is developed based on the density of customers experiencing voltage range which is then represented by ‘kernel’ probability density function (PDF). The time series voltage profile simulation is then visualized as contour map based on the obtained PDF that density of loads experiencing a particular range of voltage for the entire peak day.

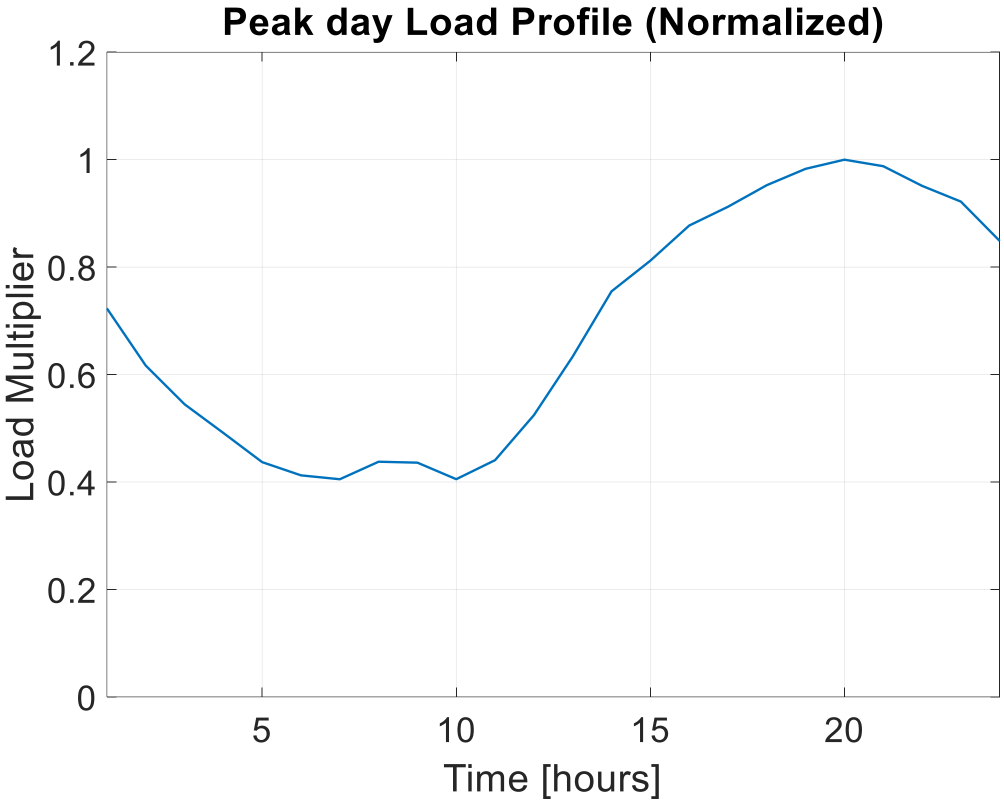


Figure 3-3: Load Profile for the peak daytime series simulation.

A total of 10 locations are observed to experience under voltage violations with the base case and details regarding duration and magnitude of experienced under voltage situation are summarized in Table 3-3. All the loads experiencing under voltage violation are observed to be in phase B.

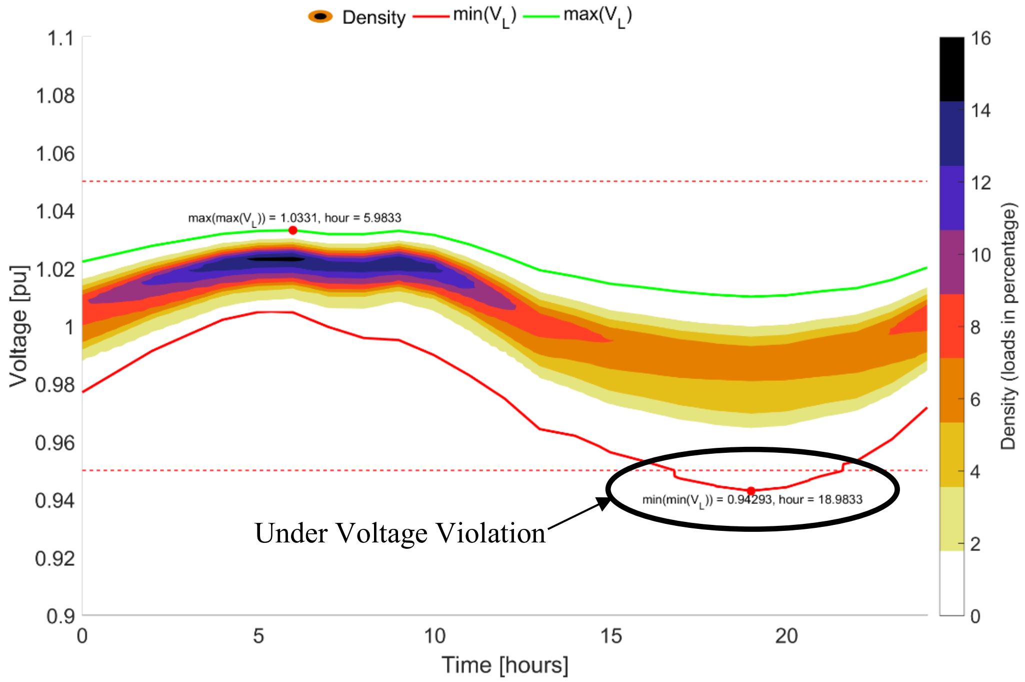


Figure 3-4: Feeder voltage profile of the base case time series simulation.

Table 3-3: Locations experiencing under-voltage violations in base case.

Location	**Under voltage duration (min)**	**Minimum voltage observed (pu)**
X_39754_3.2	287	0.9429
X_39754_2.2	287	0.9429
X_39754_1.2	287	0.9429
X_14996_3.2	281	0.9432
X_14996_3.2	281	0.9432
X_14996_3.2	281	0.9432
X_15005_4.2	160	0.9464
X_15005_3.2	160	0.9464
X_15005_2.2	160	0.9464
X_15005_1.2	160	0.9464

4. IMPACT OF SMART INVERTER CONTROLLED INTEGRATED PV ON DISTRIBUTION NETWORK

This chapter explains about the advantages of implementing smart inverters, explaining about three of the control functions in detail along with the simulations of each of the explained control functions on the chosen distribution network in a variety of selected scenarios. The smart inverter control functions discussed in this study are intelligent volt-var (VVC), adaptive volt-var (AVVC), dynamic reactive current control (DRCC) and a combined control function where both intelligent volt-var and dynamic reactive current control functions are combined to see the impact on the distribution network.

4.1 BENEFITS OF SMART INVERTER SETTINGS

This section explains the need for smart inverter settings for the integration of PV in brief. In the recent past, there have been many studies and analyses about determining the maximum PV hosting capacity that can be integrated into the distribution systems. It is also evident from the research that the PV hosting capacity of the distribution network can be improved to great extent with the combination of smart inverter technology [9]. This section briefs the significant benefits that the smart inverters have in not just improving the hosting capacity based on ANSI voltage limitations but also in improving the power quality such as voltage regulation, mitigating voltage fluctuations of the observed voltages that rise due to variable and intermittent power injections of PV.

Before focusing on determining the most effective and efficient smart inverter setting for the integrated PV, existing research on the methods to arrive at choosing these settings are discussed in brief. An intense research has been put into providing a methodology on how to determine settings for both single and multiple DER on a feeder

[10] based on power factor. The research comes up with three different methods in determining power factor settings:

1. Median feeder X/R ratio.
2. Weighted average X/R ratio at the location of PV.
3. Sensitivity – based algorithm.

The research suggests that method 3 would be effective in mitigating the voltage related issues but since that fixing a constant power factor setting for an integrated DER would result in reduced voltage rise even when not necessary. The key factors that influence in determining power factor setting or even the inverter setting for smart inverters depend on both the network parameters as well as the DER parameters. The influence of these parameters in determining inverter settings is discussed in detail later in the following sections.

The smart inverters for PV would not be limited by a fixed power factor control and rather would have a control algorithm that mitigates voltages related issues at all times based on the target parameter chosen by the control function. This chapter briefs the benefits of using smart inverter control strategies and details about the functioning and application of few smart inverter settings that control the integrated distributed PV. The effectiveness of each of the functions is studied with varying key parameters and analyzed based on the impact created by them. The study of various smart inverter settings is carried from the context of the distribution network which is discussed in Chapter 3.

DER, which also includes PV, that are integrated into the distribution grid with unity power factor setting generally result in the rise in voltage through real power injection at the point of interconnection (POI). The introduction of PV can have adverse effects on the distribution system through voltage related issues such as voltage rise and highly varying voltage. The extent of voltage related issues is dictated by various factors like PV output (active and reactive power), the equivalent impedance at the POI of PV, the variation of irradiance due to intermittent cloud coverage as well as other PV systems on the feeder [11]. The use of smart inverters can have significant benefits in overcoming these voltage related issues if the control functions are tuned accordingly. In many cases, the use of smart inverters can be the most cost-effective solution in mitigating voltage related issues due to PV [12]. It is the reactive power related inverter functions that are used to mitigate these voltage related issues but the availability of reactive power is dependent on the type of function chosen by the smart inverters which will be discussed in detail in the Sections 4.3 through 4.6 in this chapter.

There are numerous settings that can be implemented to a smart inverter but these settings should be appropriately chosen because a PV system with wrong smart inverter settings can actually worsen the grid performance beyond seen if the PV is operating at unity power factor. Sections 4.3-4.6 explain about various smart inverter functions that can be implemented in mitigating the voltage related issues. Each of the sections focuses on the scope of the chosen function, its objective, key factors that influence the effectiveness of the smart inverter in impacting the target parameters and the application methodology in OpenDSS through which the functions are implemented.

4.2 ANALYSIS OF VOLTAGE VARIATION DUE TO INTEGRATION OF PV

This section discusses about locations for integrating PV into the chosen distribution circuit, factors that effect in voltage rise at the location of PV integration, the quantitative analysis of the variation in voltage as a result of the integration of PV and finally a few conclusions on the settings that are to be implemented for smart inverters of

PV for a better functioning and performance of the customers through a regulated and smoothened voltage profile.

As shown in Figure 4-1 and Figure 4-2, the highlighted areas display the locations that experience under voltage violation during the peak daytime series simulation. Since the objective of the analysis is to find the smart inverter settings of PV that would effectively eliminate under as well as over voltage issues along with smoothing of the voltage profile, the locations for integration of PV are so chosen that the pre-integration voltages observed don't meet the standards of ANSI voltage limits i.e., per-unit voltages between 0.95 pu and 1.05 pu. The ten locations as listed in Table 3-3 which experience undervoltage condition on the peak day base case scenario are considered for integration of PV with smart inverters. The locations that experience voltages between 0.95 pu and 0.96 pu, which are prone to experiencing under voltage violations in the case of increased loading are also considered for integration of PV. Eleven such locations that are close to locations listed in Table 3-3 are considered for integration of PV with smart inverters.

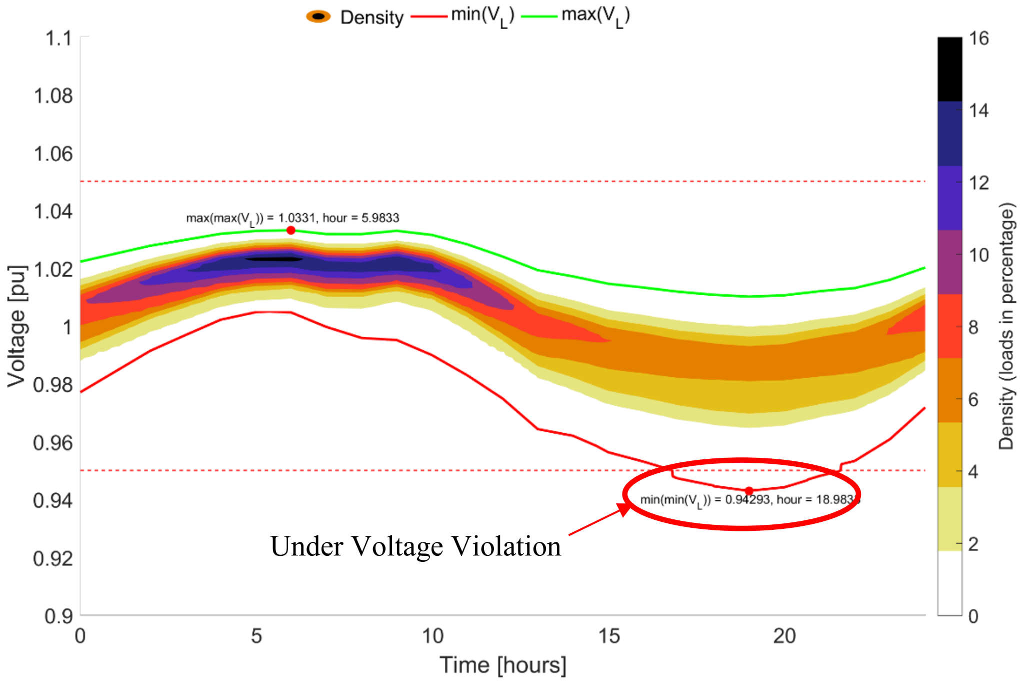


Figure 4-1: Feeder voltage profile of the base case time series simulation.

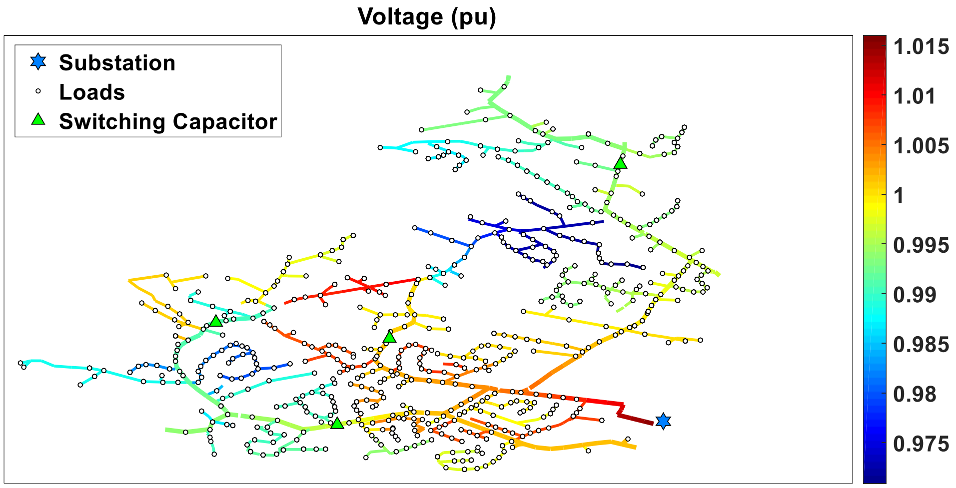


Figure 4-2: One line diagram of the residential circuit with voltage range.

A total of twenty-one PV systems are installed at 6 locations marked by the red asterisks and are installed in the area which is chosen for analysis of the impact of smart inverter settings on integrated PV in the residential distribution network. The locations are shown in Figure 4-3. It should be observed that all the 21 locations are in phase B and that is the only phase that has been observing under voltage violations and have more customers prone to under voltage violations. The number of PV appears to be 6 because few of the PV systems are integrated at the end of laterals that come from a common bus.

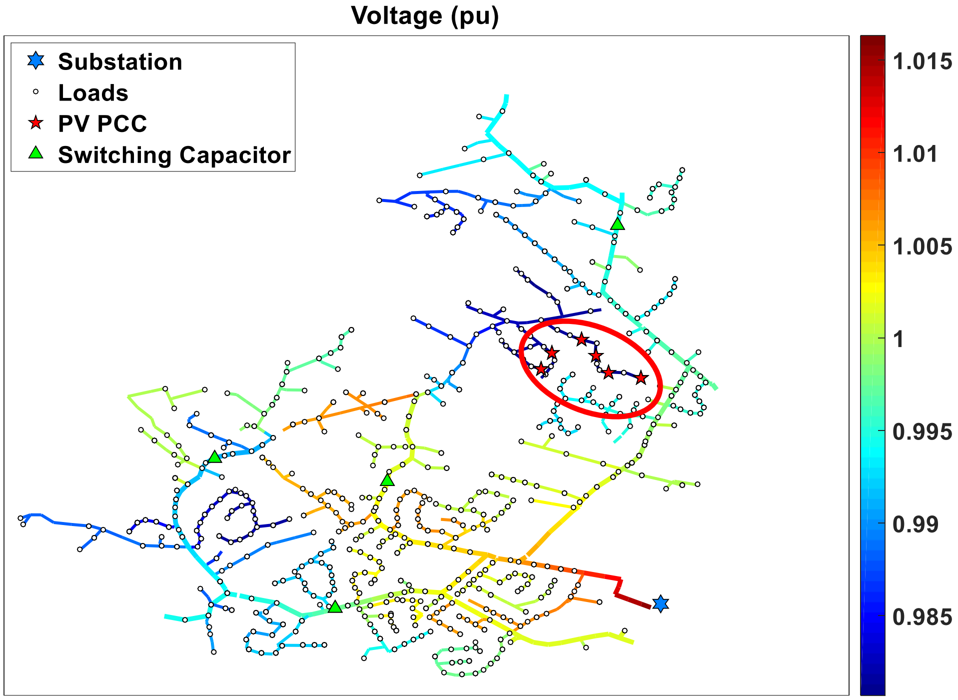


Figure 4-3: One line diagram of the residential circuit PV locations.

Figure 4-4 outlines the location of PV integration on the secondary side of the distribution system. The PV system is shown to vary its output depending on the solar irradiance. The figure also highlights the involvement of smart inverter in monitoring the voltage at the POI and also in controlling the PV in managing voltage and reactive power based on the control function.

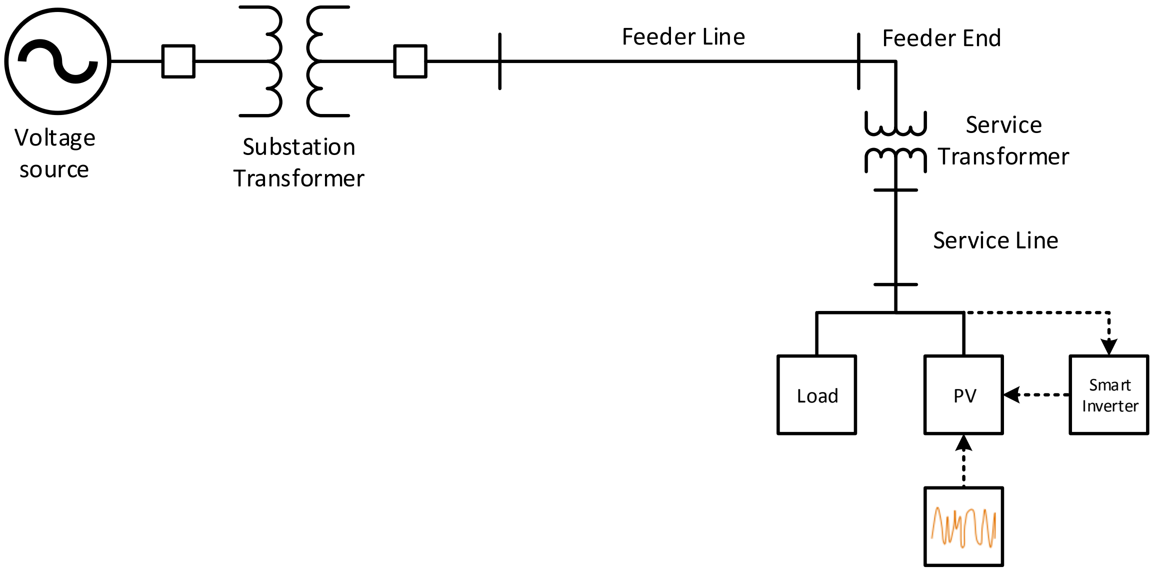


Figure 4-4: Simplified block diagram of the integration of PV enabled with a smart inverter.

This following section discusses the effect of various parameters in observing the change in voltage due to the integration of PV into the distributed network. An equivalent system as shown in Figure 4-5 is assumed to study the voltage variations that are observed at the POI of PV.

The equivalent system consists of a voltage source, V_s represented by a Thevenin equivalent in series with Thevenin impedance, R+jX, observed at the POI of the PV primary and a PV system which is represented by a current source, $I_{pv.}$ The current injected from PV can be written as shown in Equation 4-1. The Thevenin equivalent impedance at the primary of POI of the PV is obtained from fault study analysis of the distribution network. Since the change in voltage due to the integration of PV is nothing but the voltage drop across the Thevenin impedance due to injected current from the PV, the change in voltage observed at the POI of PV is calculated as shown in Equation 4-2.

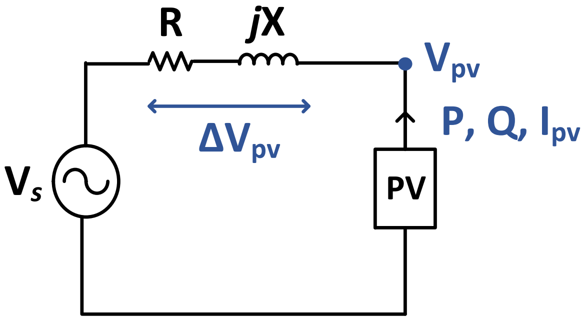


Figure 4-5: Thevenin equivalent system with integrated PV.

$$I_{pv} = \frac{P - jQ}{V_{pv}^*} \qquad \text{Equation 4-1}$$

$$\Delta V_{pv} = V_S - V_{pv} = (R + jX)I_{pv} = \frac{1}{V_{pv}^*}\left[(RP + XQ) + j(XP - RQ)\right] \qquad \text{Equation 4-2}$$

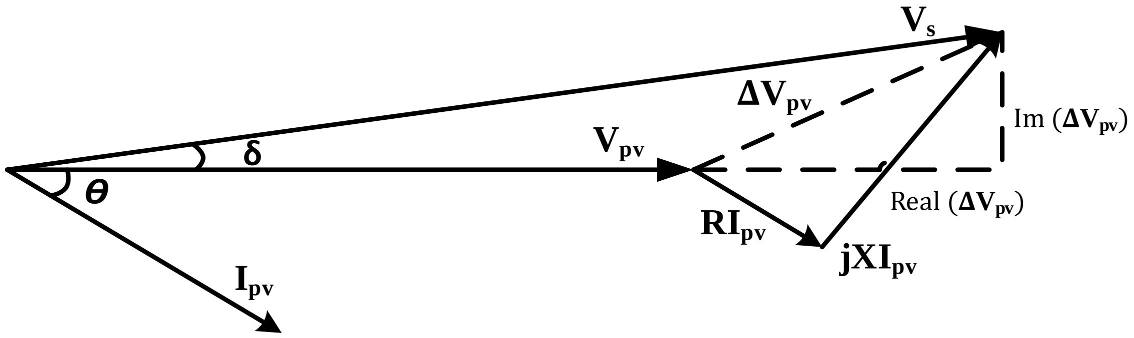


Figure 4-6: Phasor diagram to explain the induced voltage due to the integration of PV.

Few assumptions are assumed at this point of analysis to simplify the analysis. The angle, (δ) between source voltage (V_S) and the terminal voltage at POI of PV (V_{pv}) is very small as shown in Figure 4-6. Because of that, ΔV_{pv}, the voltage drop across the source and the PV is approximately equal to the real part of impedance drop [13]. So, the reactive term of Equation 4-2 can be neglected. And, since the voltage phase shift is negligible due to low distribution system reactance, the voltage at the POI is similar to that of its complex conjugate.

With the above assumptions, the change in voltage observed at the POI can be expressed as

$$\Delta V_{pv} = \frac{RP + XQ}{V_{pv}} \qquad \text{Equation 4-3}$$

$$\Delta V_{pu} = \frac{\Delta V_{pv}}{V_{base}} \qquad \text{Equation 4-4}$$

Upon simplifying the change in voltage in per unit standard as shown in Equation 4-4, the factors that are responsible for the change in voltage at the POI can be listed as follows:

1. Distribution network parameters – R and X @ POI of PV
2. PV system parameters – injected P and Q

The distribution network parameters that are responsible for the change in voltage at POI of PV are the Thevenin equivalent resistance and reactance as observed from the primary of POI of the PV to the substation. The PV system parameters that are responsible for the change in voltage are the real (P) and reactive (Q) power of the PV system.

It can be summarized from Equation 4-3 that the location of integration and the size of PV play an important role in the observed change in voltage at the POI. The distribution network parameters i.e., positive sequence short-circuit resistance (R_1) and reactance (X_1) are shown in Figure 4-7 and Figure 4-8. On comparing these to figures with the chosen locations for integration of PV into the distribution system, Figure 4-3, it can be observed that all the PV which are close to one another observe a similar impedance from the system that affects the induced change in voltage.

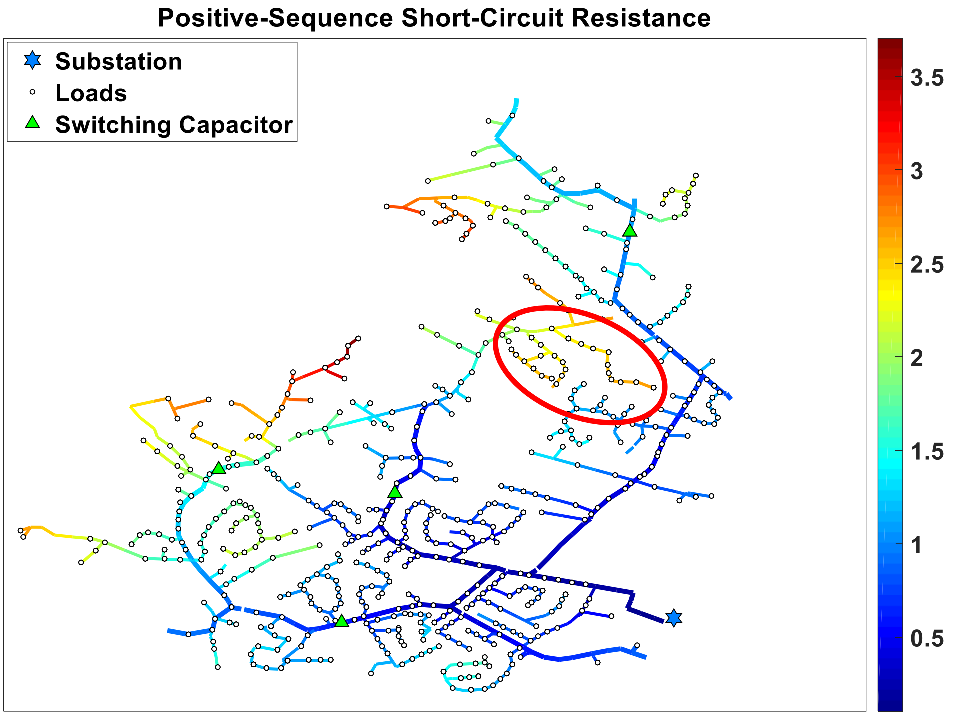


Figure 4-7: Positive sequence short-circuit resistance of the distribution circuit.

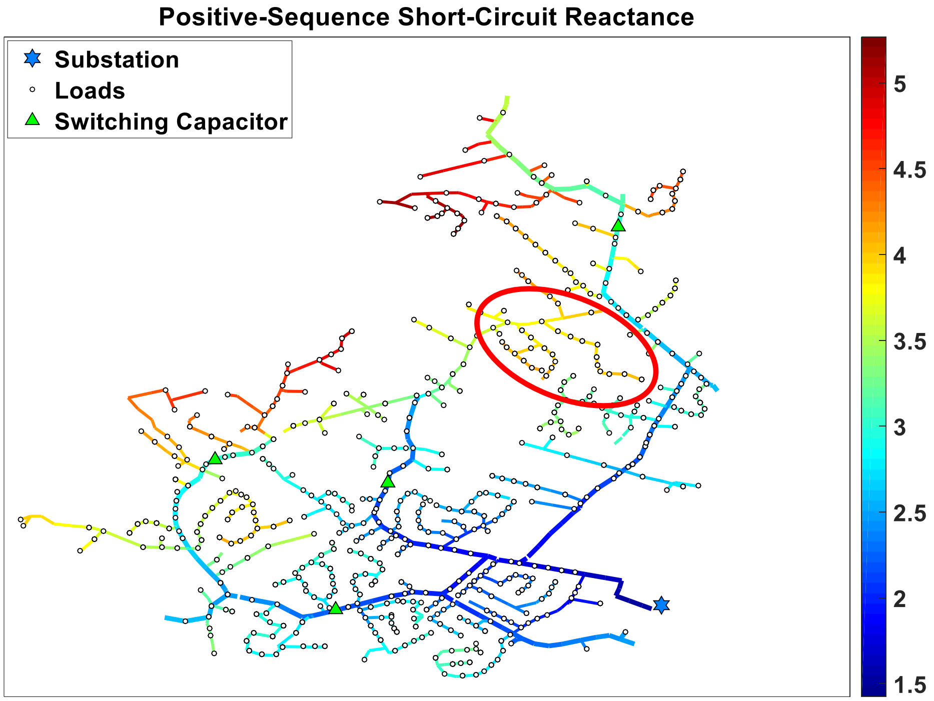


Figure 4-8: Positive sequence short-circuit reactance of the distribution circuit.

Coming to the parameters of the integrated PV, the induced change in voltage is observed to be dependent on the real and reactive power that is exchanged between the distribution network and the PV system. Real or reactive power injected from the PV system into the distribution network is considered positive and the reverse flow is considered negative in this analysis. The size distribution of PV systems that are integrated for initial analysis into the distribution network is shown in Figure 4-9. The rating of PV systems are assumed to be same as of the peak load at each integrated location in the initial analysis and then the assumed size is later increased approximately 4 times the peak load to simulate a very high penetration scenario. This analysis helps compare the effectiveness of the control algorithms implemented through smart inverters in regulating and smoothing the voltage profile of the distribution network with very high penetration levels of PV.

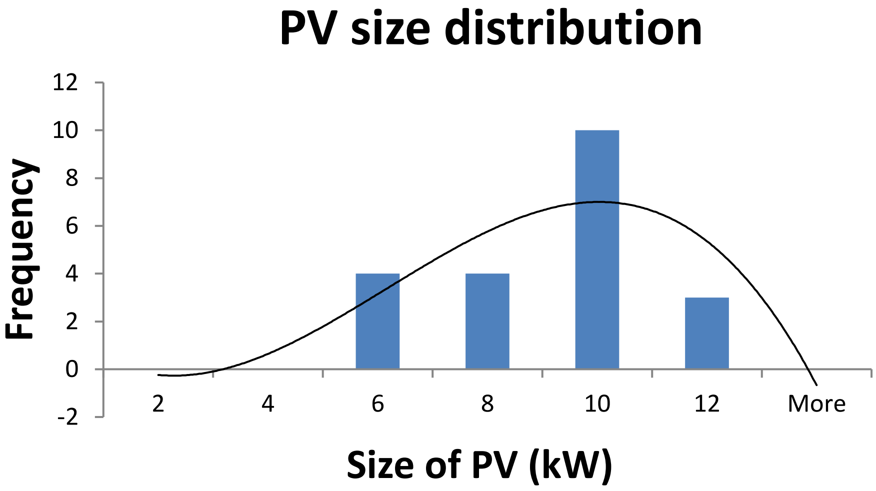


Figure 4-9: Size distribution of the integrated PV into the distribution circuit.

Based on the information provided from Figure 4-10 and Figure 4-11, and from short-circuit impedance data obtained from running fault study analysis on the distribution network, a quantitative analysis of the induced change in voltage is discussed.

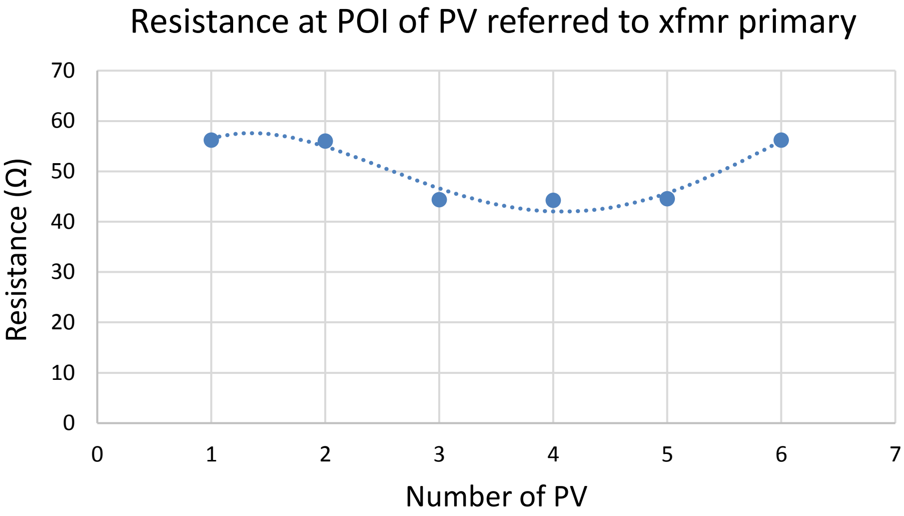


Figure 4-10: Resistance at primary of POI of the integrated PV.

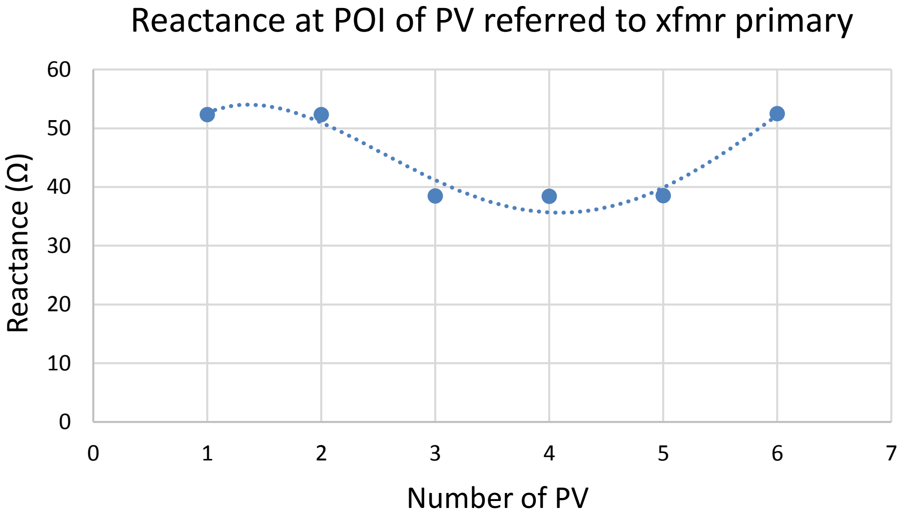


Figure 4-11: Reactance at primary of POI of the integrated PV.

The rating of PV systems integrated into distribution network ranges between 5.2 kW and 10.1 kW. The observed equivalent resistance ranges between 44.3Ω and 56.3Ω whereas the equivalent reactance observed ranges between 38.4 Ω and 52.5 Ω respectively as observed from the substation. The extremes of induced voltage are calculated for real and reactive power injection separately as following:

$$\boldsymbol{\Delta} V_{pu,\max\_P} \cong \frac{56.25 \times 10.12\ (kW)}{7.2\ (kV)} = 79.09\ V = 0.01098\ \text{pu}$$

$$\boldsymbol{\Delta} V_{pu,\min\_P} \cong \frac{44.28 \times 5.19\ (kW)}{7.2\ (kV)} = 31.93\ V = 0.0044\ pu$$

It is a general practice to assume the inverter rating to be 1.1 times the rating of the PV system integrated into the distribution network. And so, the minimum and maximum reactive power available for transfer are calculated to be: 2.38 kvar and 11.14 kvar. These values are used in quantitative analysis of finding extremes of voltage difference observed due to reactive power injection.

$$\boldsymbol{\Delta} V_{pu,\max\_Q} \cong \frac{52.46 \times 11.14\ (kvar)}{7.2\ (kV)} = 81.13\ V = 0.011\text{pu}$$

$$\boldsymbol{\Delta} V_{pu,\min\_Q} \cong \frac{38.40 \times 2.38\ (kvar)}{7.2\ (kV)} = 12.69\ V = 0.0018 pu$$

$$\boldsymbol{\Delta} V_{pu\_P} = \boldsymbol{\Delta} V_{pu,\max\_P} - \boldsymbol{\Delta} V_{pu,\min\_P}$$

$$\boldsymbol{\Delta} V_{pu\_Q} = \boldsymbol{\Delta} V_{pu,\max\_Q} - \boldsymbol{\Delta} V_{pu,\min\_Q}$$

It is observed from this analysis that the maximum variation in $\boldsymbol{\Delta} V_{pu\_P}$ is 0.0066 pu and in $\boldsymbol{\Delta} V_{pu\_Q}$ is 0.0092 pu on (7.2 kV base) which are not significant. Based on this analysis, it is assumed that any PV in the same vicinity as of other PV systems and with same base voltage would be experiencing similar voltage change due to real and reactive power exchange with the distribution network. Since the PVs are in same vicinity and because the change in voltage induced by the PVs are similar, the inverter control settings for all the integrated PV are assumed same and the need for customized smart inverter settings is eliminated.

The need for customized smart inverter settings would be essential and demanding in distribution networks with PV interconnected in a wide geographical area even if the integrated systems have a similar size distribution. This is due to the reason that there exists a noticeable difference in induced voltage change due to uncommon distribution system parameters in terms of equivalent impedance seen from the substation.

4.3 INTELLIGENT VOLT -VAR CONTROL FUNCTION

This section details about the intelligent volt-var control of smart inverters stating the scope of the function, parameters that define the function, its intelligent characteristics that help in regulating the voltage at POI of PV and the implementation of the function in OpenDSS.

The scope of intelligent volt-var function is intended to provide a mechanism through which a PV may be configured to manage its own reactive power output in response to the local service voltage at the POI [14]. This function allows each individual PV system to provide a unique reactive power response according to [15]:

1. Voltage at the POI of PV (the terminals of the DER)
2. Available apparent power capacity of the inverter at that point in time
3. Utility-defined volt-var set points as illustrated in Figure 4-12.

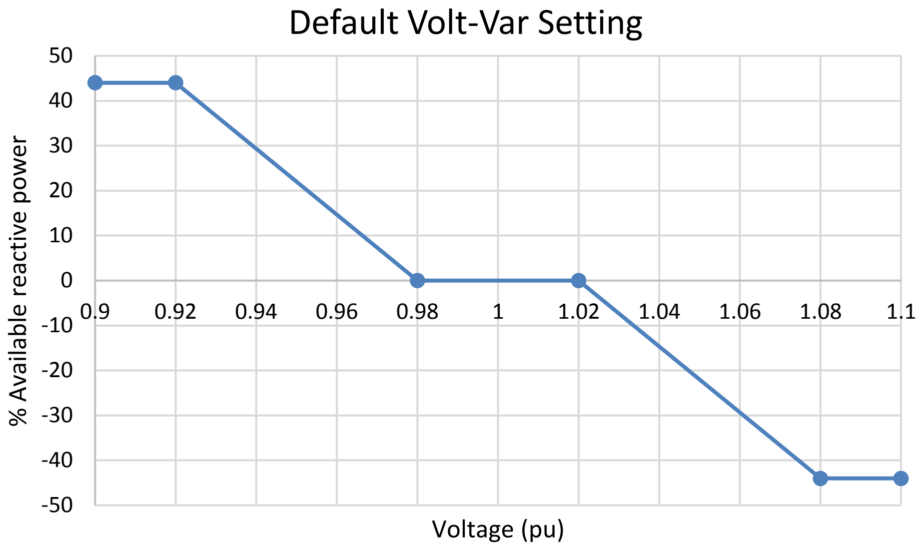


Figure 4-12: A typical volt-var curve for smart inverters.

It is generally assumed that the rating of an inverter would be 110% of that of the PV system to which it is connected and the available apparent power capacity of the inverter at any moment is governed by the following equation:

$$kvar_{available} = \sqrt{(kva\_rating)^2 - (present\_kW)^2} \qquad \text{Equation 4-5}$$

Each of the integrated PV can be made to follow this function by a user defined array which usually has a variable number of points, which together define a piece-wise linear curve of the desired volt-var behavior. This array is defined as an XYCurve in OpenDSS which defines the output reactive power as a function of the terminal voltage of the PV system.

A positive sign for reactive power output (in per unit, pu) for the y-array values indicate capacitive power or reactive power flowing in the same direction as the active power similar to a capacitor acting as a source of reactive power which is generally used as var compensator. A negative sign for reactive power output indicates inductive power, or reactive power flowing in the opposite direction as the active power. It is a general practice to have a dead band in the volt-var curve during which no reactive power transfer is allowed.

The voltage at the POI of PV is dynamic and the objective of the implementation of this function is to maintain this voltage within the ANSI limits i.e. between 0.95 pu and 1.05 pu for a better performance of the loads. Let us consider the voltage profile at the terminal of a PV without smart inverter as shown in Figure 4-13.

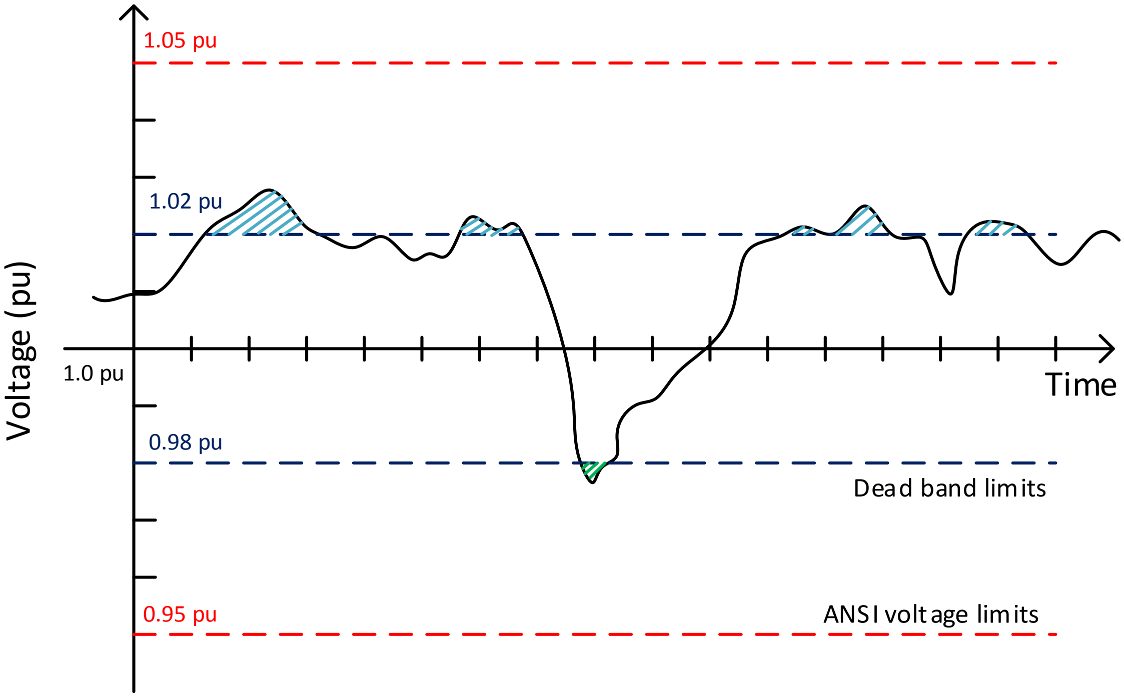


Figure 4-13: Voltage profile at the terminal of PV to explain the functioning of smart inverter control functions.

Assume each marked division on X-axis of Figure 4-13 is a time step of the simulation, i.e., a power flow solution of the system is evaluated and voltage, current, powers of all elements of the network are updated at an interval of this duration in OpenDSS.

Based on Figures 4-12 and 4-13, the functioning of smart inverter enabled with volt-var control is now explained. The reference voltage for defining an intelligent volt-var curve is always the rated voltage at the terminal of integrated PV i.e., the unit 1 on X-axis of the volt-var curve defined in Figure 4-12 will always refer to the terminal voltage of PV system. This way, the intelligent volt-var curve and the PV system will have a common reference. As the dead band defined in Figure 4-12 limits the observed voltage between 0.98 pu and 1.02 pu, the smart inverter forces the PV system to absorb reactive power at all the time-steps where the voltage profile increases above 1.02 pu (shaded blue) so that the voltage experienced immediately drops below 1.02 pu. Similarly, the smart inverter forces the PV system to inject reactive power when voltage profile drops below 0.98 pu (shaded green) so that the voltage profile rises. So, at every time-step, based on the voltage observed at the terminal of PV, the reactive power is exchanged to ensure the voltage profile is within the defined dead band. The amount of reactive power to be exchanged is also defined by the volt-var curve shown in Figure 4-12. Once the voltage observed at the terminal of PV is in the dead band of the curve, the inverter functions at unity power factor trying to remain in the defined dead band of the volt-var curve. The per-unit voltage on the volt-var curve is calculated based on Equation 4-6.

$$V_{pu} = \frac{Observed\ voltage\ at\ the\ terminal\ of\ PV}{Rated\ terminal\ voltage\ of\ PV} \qquad \text{Equation 4-6}$$

The horizontal axis of the volt-var curve is the per-unit voltage and is defined as the ratio of the observed voltage at the terminal of the PV and the reference voltage. The reference voltage for the volt-var function is the rated voltage which is 1 pu. Adaptive volt-var function, which will be discussed in detail in Section 4.4, differs with volt-var function in defining this per unit voltage.

The following are the key factors of the volt-var function that define the optimal behavior of the controlled PV:

1. Dead band in the volt-var curve
2. Slope of the curve in capacitive and inductive regions

The presence of a dead band in the volt-var curve indicates that the inverter neither absorbs nor injects available reactive power and it can be varied by varying the defined curve based on the purpose of the analysis. The slope of the curve defines the aggressiveness of the inverter in injecting or absorbing available reactive power or in other words, the aggressiveness of the smart inverter in pushing the PV system to experience voltages corresponding to defined dead band. The target parameter in choosing intelligent volt-var function is voltage at the terminal of PV.

Research has been done in defining a default volt-var curve that can improve distribution system performance and enhance acceptable DER integration levels on a feeder [11]. Figure 4-12 shows the default volt-var settings proposed by IEEE Voltage Regulation Subgroup, an IEEE working group, where the dead band is maintained between 0.98 pu to 1.02 pu and the maximum reactive power injection is limited to 44% of the available reactive power of the PV. The default volt-var curve has been defined to work under low-penetration scenarios, high penetration scenarios, urban and rural types of feeders, PV locations near the feeder head as well as locations near the feeder end. Since the aim of the study is to enhance the performance of the distribution system using smart inverter functions, this study is not just limited with default volt-var curve.

```
New XYCurve.vv_curve1 npts=4 Yarray=(0.44,0.44,0,0,-0.44,-0.44)
Xarray=(0.9,0.92,0.98,1.02,1.08,1.1)
```

Illustration 4-1: Code snippet for defining volt-var curve in OpenDSS.

The OpenDSS code snippet for defining intelligent volt-var curve has been shown in Illustration 4-1. The volt-var curve is defined initially and the inverter control is defined later and made to follow the curve in the following line of code as shown in Illustration 4-2.

```
New InvControl.InvPVCtrl1 mode=VOLTVAR voltage_curvex_ref=rated
vvc_curve1=vv_curve1 DeltaQ_factor=0.1 voltagechangetolerance=0.01
varchangetolerance=0.001 Eventlog=Yes
```

Illustration 4-2: Code snippet for defining inverter controller in volt-var mode in OpenDSS.

4.4 ADAPTIVE VOLT-VAR CONTROL FUNCTION

This section explains the objective of the adaptive volt-var function, the differences in defining adaptive volt-var setting from intelligent volt-var setting, the additional key factor that plays an important role in defining the setting and the target parameter of this function.

The adaptive volt-var function is similar to volt-var function explained in Section 4.3 but that the reference voltage in defining the curve is different. The functioning of adaptive volt-var control is explained based on Figures 4-12 and 4-14.

The typical moving average calculation of voltage is shown as a brown rectangle in Figure 4-14. The averaging window will calculate the average PV system terminal voltage over a specified period of time, up to and including the last power flow solution. The reference voltage for the adaptive volt-var curve is this average voltage calculated by a defined window whose length would be specified by the user. This average will be adapted as the center (1 pu) of the volt-var curve which changes over time with moving window. Since the average of the voltage changes as the power flow solution is computed in each time-step, the reference voltage is no longer constant and varies. This basically represents that the volt-var curve is not fixed to the terminal voltage of the PV system but is fixed to the moving average of terminal voltages which are calculated depending on the length of the window defined by the user.

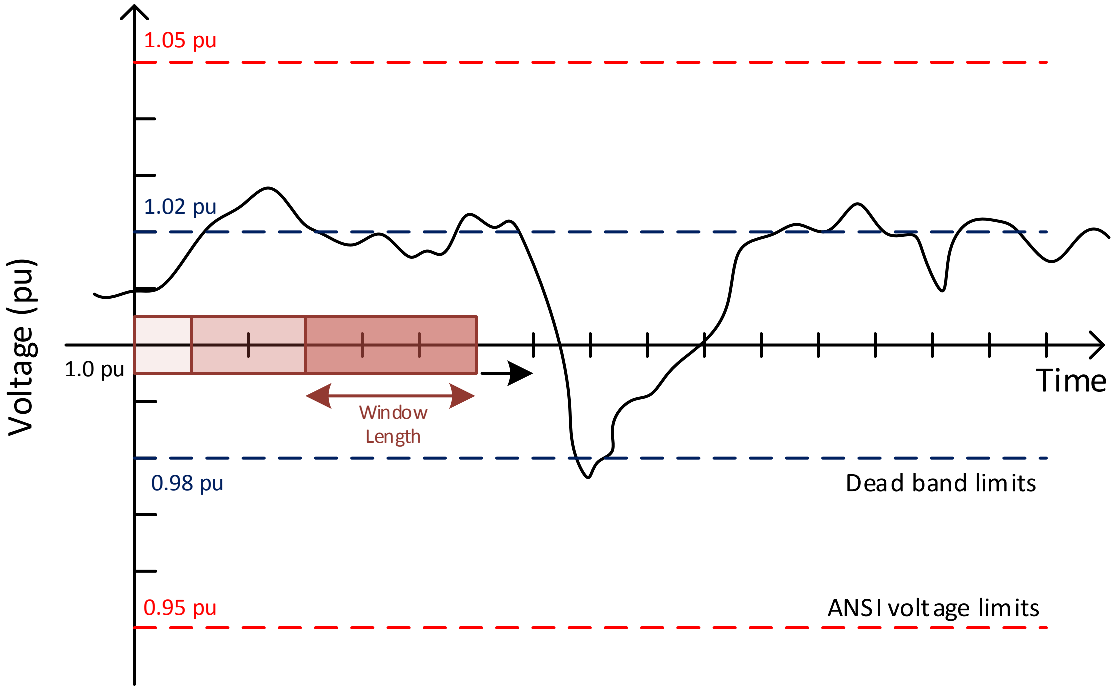


Figure 4-14: Voltage profile at the terminal of PV to explain the functioning of smart inverter control functions.

The slope of the curve still defines the aggressiveness of the inverter in pushing the observed voltage of the controlled PV to the dead band defined but since the adaptive set point is chosen, the control function tends to achieve the average voltage value which changes dynamically with time. This is analogous to intelligent volt-var curve that slides on the horizontal axis with its center point adapting the value of average of the voltages calculated for the specified averaging window length. The per-unit voltage on the adaptive volt-var curve is calculated based on Equation 4-7.

$$V_{pu} = \frac{Observed\ voltage\ at\ the\ terminal\ of\ PV}{Calculated\ moving\ Average} \qquad \text{Equation 4-7}$$

The key factor of the volt-var curve with an adaptive set point is the averaging window length defined by the user. Figure 4-15 represents the reference voltage of an adaptive volt-var control curve.

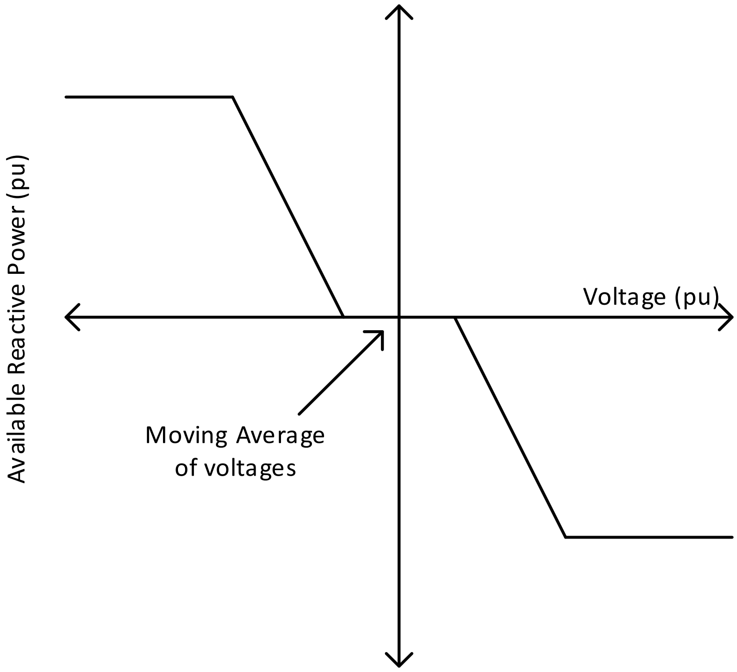


Figure 4-15: An adaptive volt-var curve highlighting the reference voltage.

It should be observed that if the monitored voltage at the point of connection were to see no change, the computed average voltage saturates and reactive power absorbed or injected would be minimal even if the saturated average voltage is either 1.02 or 0.98 pu.

The adaptive volt-var control aims at maintaining the terminal voltage of PV at average voltage calculated over a period defined by the window length which changes as the simulation is carried out. But, the intelligent volt-var control aims at maintaining the terminal voltage of PV at fixed reference voltage which is the rated voltage at the terminal of PV.

Illustration 4-3 shows the code snippet of defining a volt-var curve with adaptive set point function in OpenDSS. One should observe that the reference voltage to be considered is mentioned as “avg” meaning to calculate the average voltage from a certain interval of time defined by the parameter “avgwindowlen”.

```
New InvControl.InvPVCtrl1 mode=VOLTVAR voltage_curvex_ref=avg avgwindowlen=120m
vvc_curve1=vv_curve1 DeltaQ_factor=0.1 voltagechangetolerance=0.01
varchangetolerance=0.001 Eventlog=Yes
```

Illustration 4-3: Code snippet for defining inverter controller in an adaptive volt-var mode in OpenDSS.

4.5 DYNAMIC REACTIVE CURRENT CONTROL FUNCTION

This section is about dynamic reactive current control (DRCC) function which explains the scope of the function, key factors that distinguish the function from previously explained functions, the target parameter and the way to implement the same in OpenDSS.

The scope of the function is to provide a flexible mechanism through which the smart inverters may be configured to provide reactive power support in response to dynamic variations in voltage. This function is distinct from the existing intelligent volt-var and adaptive volt-var functions described earlier in previous sections, in that the controlling parameter is the change in voltage rather than the voltage level itself. The following are the key factors of the DRCC function that differentiates its characteristics from previously explained functions:

1. % Delta voltage (parameter that defines horizontal axis)
2. Dead band of the defined curve
3. Slope of the curve

Figure 4-16 shows the control settings that a dynamic reactive current control function follows. This would explain the basic concept and helps to differentiate other control functions explained in Sections 4.3 and 4.4.

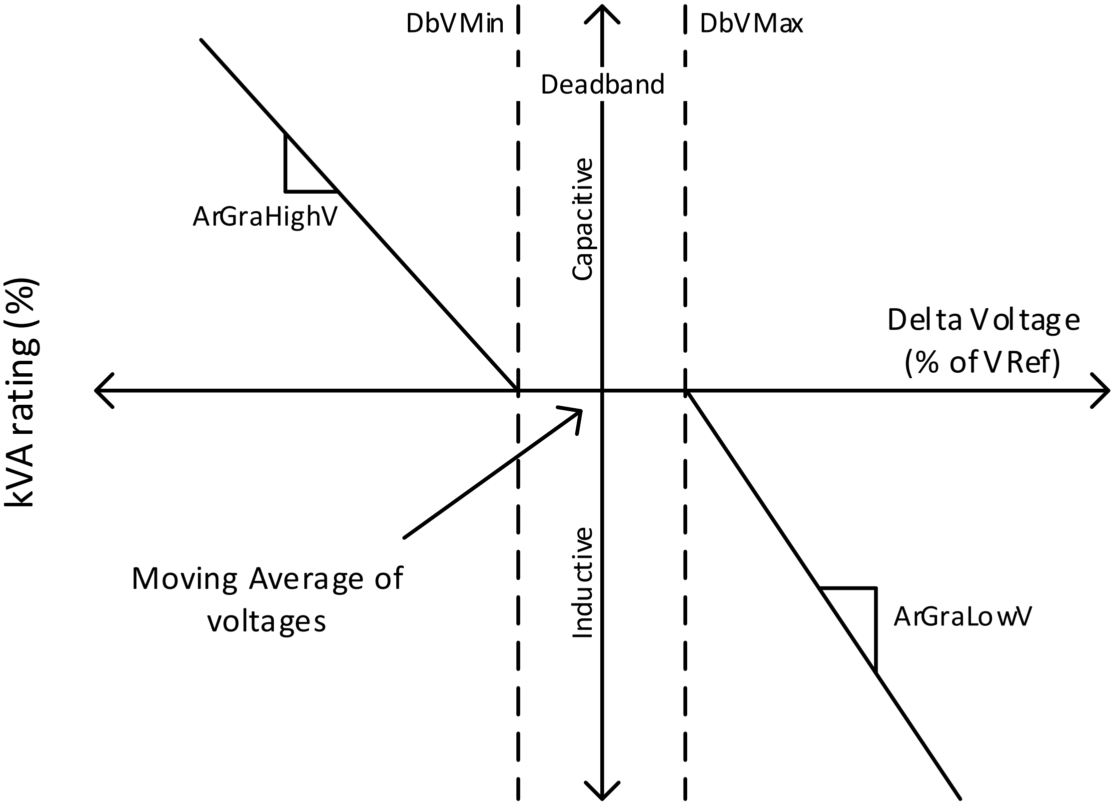


Figure 4-16: Control settings of dynamic reactive current control.

Percent delta-voltage is defined as the present voltage at the terminal of PV minus the moving average voltage, expressed as a percentage of the rated voltage for the PV. The same has been mathematically expressed in Equation 4-8. The moving average voltage is calculated in the same way as calculated in Section 4.4 based on defined window length.

$$Delta\ Voltage\ (\%) = \frac{Present\ terminal\ voltage\ of\ PV - Moving\ average\ voltage}{Rated\ voltage\ of\ PV} * 100$$

Equation 4-8

In other words, the power system voltage may be above normal, resulting in a general need for inductive reactive power, but if it is also falling rapidly, this function could produce capacitive reactive current to help counteract the dropping of the voltage [11]. The dead band is the region in which reactive current is neither supplied or absorbed i.e. the region of the curve in which the % delta voltage is neglected in forcing the controller to absorb or inject reactive power. These slope of the curves in both inductive and reactive region are unit-less and are defined as a gradient in percent of capacitive or inductive reactive power transfer divided by the percentage delta-voltage as shown in Equation 4-9. These terms basically define how quickly the reactive power is to be absorbed or injected rather than how much of the reactive power is to be absorbed or injected based on the percentage delta voltage is lower or higher than the limits of the defined dead band. More the percentage change in observed voltage, more would be the percentage change in reactive power that is exchanged. This is a type of dynamic stabilization function which creates an effect that is similar to momentum or inertia that resists rapid change in controlling parameter and can be highly helpful in mitigating power quality issues like flickering. So, this kind of function would be of good use in smoothing a control parameter which is voltage in this study.

$$Slope = \frac{\%\ Reactive\ power\ exchange}{\%\ Delta\ voltage} \qquad \text{Equation 4-9}$$

Illustration 4-4 shows the code snippet of defining a dynamic reactive current control (DRCC) function in OpenDSS. “DbvMin” and “DbvMax” are the terms that help defining the dead band of the curve which acts as limits between which the % delta voltage is accepted by the control function. “ArGraLowV” and “ArGraHiV” are unit-less terms to define the slope of the curve which basically decide how quickly the reactive power should be exchanged to see that the change is voltage is minimized or in other words, to smoothen the voltage profile. The “DynReacavgwindowlen” term determined the time interval for the average to be calculated to determine delta voltage.

```
New InvControl.InvPVCtrl13 mode=DYNAMICREACCURR DbVMin=1.0 DbVMax=1.0
ArGraLowV=40.0 ArGraHiV=40.0 DynReacavgwindowlen=2m DeltaQ_factor=0.1
voltagechangetolerance=0.0001 varchangetolerance=0.025 Eventlog=Yes
```

Illustration 4-4: Code snippet for defining inverter controller in a dynamic reactive current control mode in OpenDSS.

4.6 COMBINED CONTROL OF SMART INVERTER FUNCTIONS

This section describes the possibility and implementation of multiple smart inverter control functions on a PV. Combining control modes would help achieve multiple objectives simultaneously as well as the functions can be time scheduled to attain required objectives based on time-of-day. OpenDSS provides a feasibility of implementing two control functions simultaneously through the same smart inverter. The combined modes that are feasible are as follows:

1. Volt-var with dynamic reactive current control method.
2. Volt-var with volt-watt mode.

OpenDSS also provides the choice of choosing which control mode to precede the other in terms of implementation.

Volt-watt function is not studied in this report as it influences real power rather than the reactive power. Volt-watt function would be an appropriate function to implement in a distribution system when the high level of integrated PV and a low load is causing the feeder voltage to violate ANSI voltage limits. This function would also be a good option when more of the PV are clustered at a region and so the locality experiences a high voltage service. Implementing volt-watt in such scenarios would allow PV to share the load rather than to boost the local voltage.

The combined functionality of volt-var with dynamic reactive current control method can be helpful in two different ways. Since volt-var control is seen to be effective in mitigating over and under voltages as explained in Section 4.3 and dynamic reactive current control method compensates for the change in voltage as shown in Section 4.5, combining these functions to influence smart inverter of a PV can effectively mitigate over, under voltages and voltage fluctuations simultaneously.

The control diagram of the combined functionality is shown in Figure 4-17.

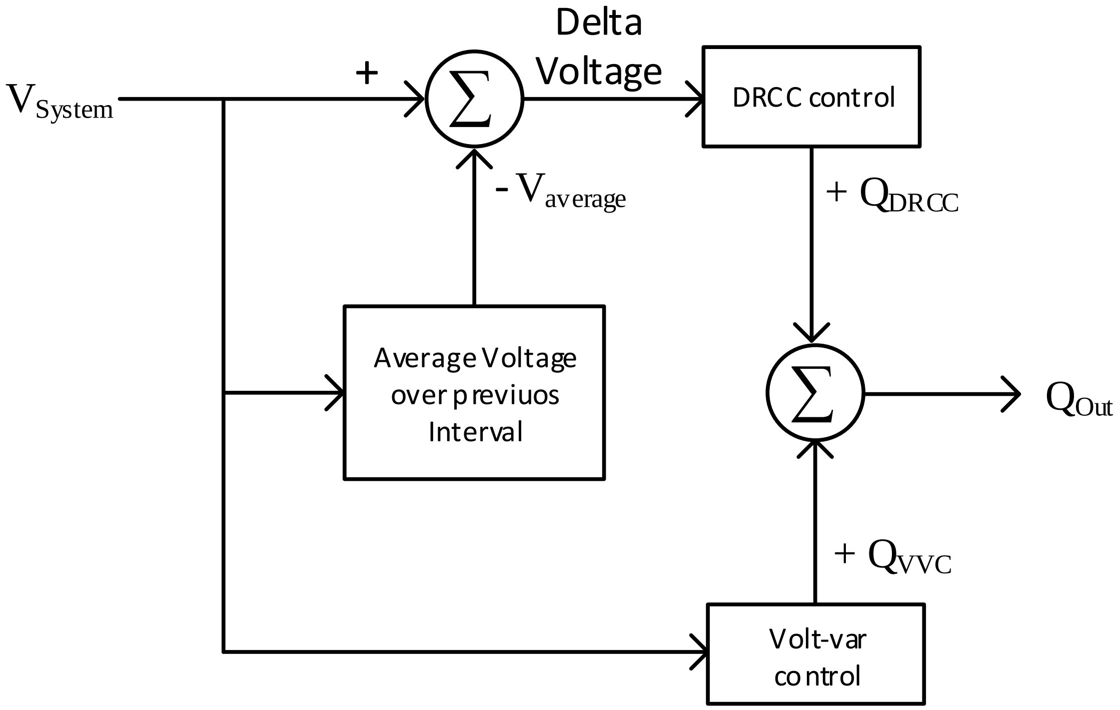


Figure 4-17: Control settings of combined function control.

Illustration 4-5 shows the code snippet of defining an inverter control setting with a combined functionality of volt-var and dynamic reactive current control methods. The precedence is given to active power output in this particular mode of smart inverter control and so the available reactive power is utilized for the purpose of mitigating over voltages and voltage fluctuations. The available reactive power may sometimes not be sufficient in achieving the desired level of voltage levels. This precedence can be changed to reactive power absorption or generation over the active power generation by varying the parameter "VV_RefRecativePower" while defining the combined functionality of volt-var setting.

```
New InvControl.InvPVCtrl1 voltage_curvex_ref=rated vvc_curve1=vv_curve1 DbVMin=1.0
DbVMax=1.0 ArGraLowV=40.0 ArGraHiV=40.0 DynReacavgwindowlen=2m DeltaQ_factor=0.1
CombiMode=VV_DRC voltagechangetolerance=0.01 varchangetolerance=0.0001
```

Illustration 4-5: Code snippet for defining inverter controller in combined control mode in OpenDSS.

Table 4-1 summarizes the main objective, the reference voltage in defining the control setting or the curve, and key factors that define the capability of each of the smart inverter control function discussed in Sections 4.3 through 4.6.

Table 4-1: A summary table describing the target parameters, reference voltages and key factors of smart inverter control functions.

Control function	**Target parameter**	**Reference Voltage**	**Key Factors**
VVC	Voltage regulation (Under and Over Voltage)	Rated terminal voltage at PV (fixed)	• Dead band • PV Terminal Voltage • Slope of the curve
AVVC	Voltage regulation (Under and Over Voltage)	Average terminal voltage at PV (Adaptive setpoint - varies)	• Dead band • PV Terminal Voltage • Slope of the curve • Averaging window length
DRCC	Mitigate faster voltage variations	Delta voltage (%) (varies)	• Dead band • PV Terminal Voltage • Slope of the curve • Averaging window length
Combined function	Mitigate over/under voltages and voltage variations	Combination of two different control functions	• Dead band • PV Terminal Voltage • Slope of the curve • Averaging window length

4.7 OBJECTIVE AND APPROACH OF THE STUDY

The following section discusses the objectives of this study, approaches undertaken in analyzing each of the control functions, metrics used in measuring the effectiveness of each of the undertaken approach of smart inverters in achieving the targeted objectives.

As shown in Figure 3-4, there exist about ten locations that experience under voltage violations in the base case simulation which are to be addressed using the integration of PV enabled with smart inverters. And since the integrated PV will be exchanging power based on the irradiance of the sun, the distribution network is expected to experience fluctuations in power as well as induced voltage at POI due to intermittent nature of irradiance. Figure 4.18 and Figure 4.19 show the irradiance plots for a clear sunny day and an intermittent cloudy day to explain the source of fluctuations that evolve in voltage profile of the distribution network.

To summarize, the following are the objectives that are taken into consideration for the analysis of the impact of smart inverter control functions on the integrated PV in the chosen distribution network.

1. Eliminating undervoltage violations existing in the distribution network.
2. Voltage regulation.
3. Mitigation of voltage fluctuations.

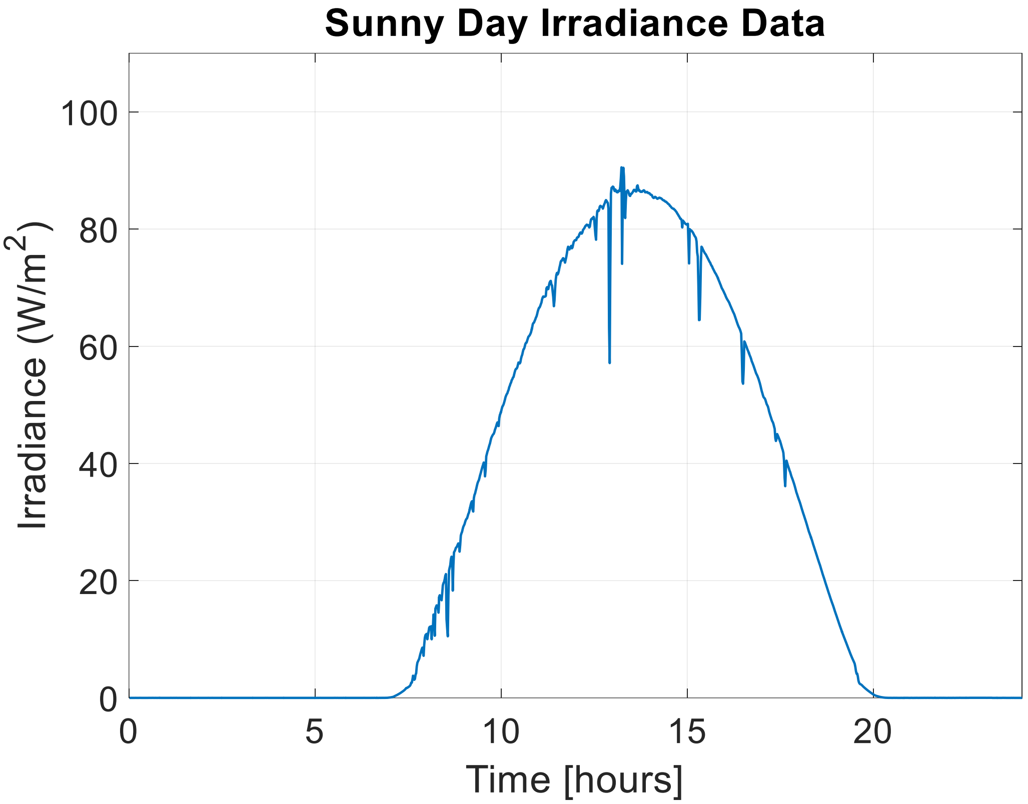


Figure 4.18: PV irradiance profile for the clear sunny day.

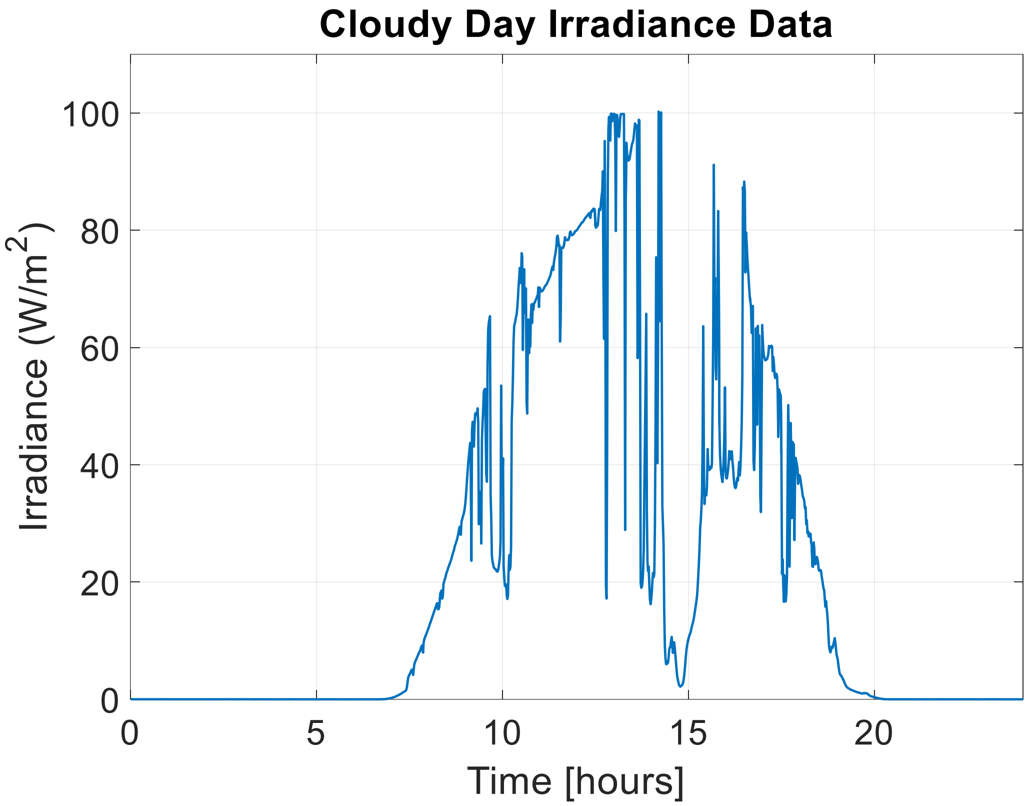


Figure 4.19: PV irradiance profile for the intermittent cloudy day.

In order to measure each of these objectives for each of the chosen scenario and control function used, different metrics are used to compare and analyze the effectiveness of the control function used. The metrics are explained in the following sections:

1. In order to analyze and quantify the under voltage violations observed during the time-series simulations, the percentage of loads experiencing under voltage violations at each time instance of the simulation are measured. The objective of the smart inverter control functions is to eliminate the under voltage violations during the entire time-series simulation and so want this metric to be zero or at the least value possible. The percentage of loads experiencing voltage value between 0.95 pu and 0.96 pu is also calculated at each time step and is analyzed for the effect imparted by control

function on the loads that are on the verge of experiencing under voltage violation if the loading is further increased.

2. Statistically, the range is defined as the difference between the maximum and minimum values of a dataset. “Range” in this context is the difference between the observed maximum and minimum voltage values of the load during the entire time-series simulation. Voltage regulation is the main objective of having intelligent volt-var control function as it directly influences the voltage to be regulated within the values defined by the dead band of the XYCurve that the function follows. It is always anticipated that the loads experience a reasonable and constant voltage without any variations for a better performance. So, to have a better performance is to have a constant and stable voltage which requires the range to be as minimal as possible. So, the advanced volt-var functions should aim to achieve a minimal voltage range.

3. The ANSI voltage limits state that it is reasonable to experience a voltage between 0.95 pu and 1.05 pu but it is not reasonable to experience a change of such high differences in short periods. Most loads and equipment are sensitive to sudden changes in voltages. In order to measure such sudden changes in voltages, “Voltage variability Index” is measured. The voltage variability index is calculated as shown in Equation 4-10 which is a modification of solar variability index used to classify solar generation [16].

$$VI = \sum_{k=2}^{n} |V_k - V_{k-1}|$$

Equation 4-10

Various approaches are experimented on the smart inverter for each of the control function to test and determine the effective smart inverter settings suitable for the integrated PV in achieving the objectives listed above. The following sections explain the approaches undertaken for each of the control functions explained in Sections 4.3 through 4.6 that help determine the effective setting.

In analyzing the effectiveness of intelligent volt-var setting of the smart inverter, as studied in Section 4.3, slopes of various aggressive limits are implemented as shown in Figure 4.20 in forcing the PV system to exchange power with the distribution network. The dead band in all the defined XYCurves are omitted to be conservative in making the distribution network experience the voltage close to 1 pu.

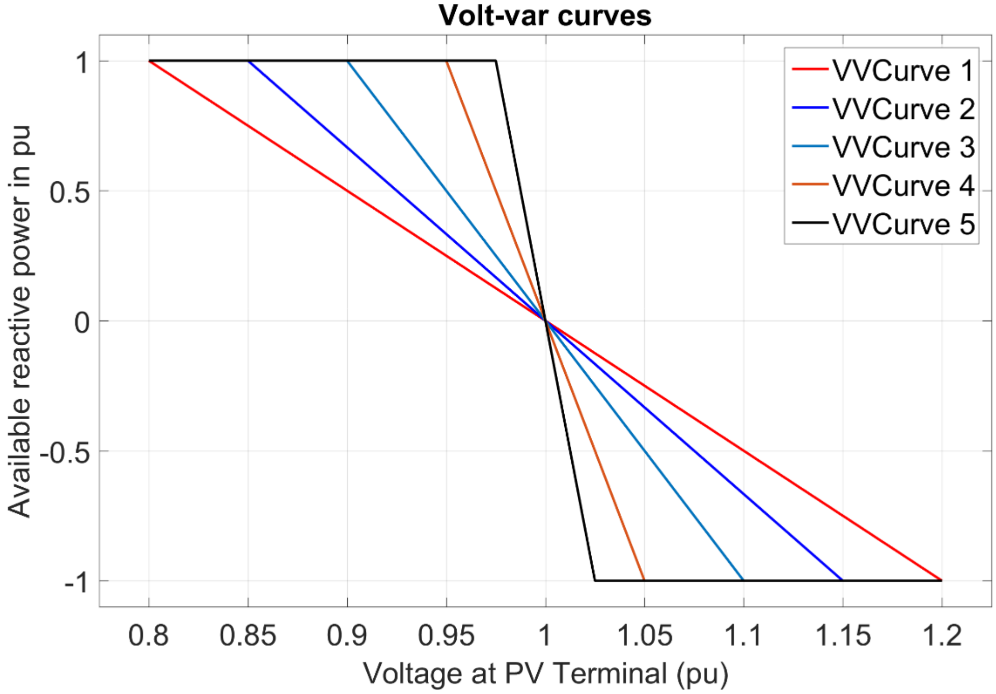


Figure 4.20: Intelligent volt-var curves with varying slopes settings.

In order to study the effectiveness of adaptive volt-var function on the integrated PV, the averaging window length is changed from 2 minutes to 400 minutes in order to see the effect of calculated average of the voltage in setting the adaptive set point as the center for regulating voltage in XYCurve. The slope of the curve is chosen to be the most aggressive curve of the intelligent volt-var curve explained in Figure 4.20. The range of the voltage observed is compared for changing window length or changing adaptive set point in other words.

The main objective of the dynamic reactive current control function is to reduce the variations observed in the control parameter which is voltage according to this study. The effectiveness of the dynamic reactive current control is also tested through changing the “DynReacavgwindowlen” from 2 minutes to 400 minutes in order to see the change observed in the variability index. The “ArGraLo” and “ArGraHi” are set as calculated from the most aggressive curve of the intelligent volt-var curve explained in Figure 4.20 for all the chosen window lengths.

The combined functionality of the intelligent volt-var control and dynamic reactive current control is also tested with the most aggressive volt-var curve in combination with varying dynamic averaging window length where both range and variability index of the voltage are considered for evaluation.

Though the inverter control is primarily focused on the mode in which it is defined to function, other terms such as “DeltaQ_Factor”, “varChangeTolerance” and “voltagechangetolerance” in defining each of the control function discussed play an important role in making the inverter function smoothly.

“DeltaQ_Factor” sets the maximum change (per unit) from the prior reactive power output level to the desired reactive power output level during each control iteration. Having a very low value would limit the reactive power exchange between the inverter and the distribution network in regulating the voltage.

“varChangeTolerance” is the change in reactive power from one control iteration solution to the next and is a determining parameter to stop additional control iterations from happening. This is the difference between the desired target reactive power (in per-unit) of the PVSystem and present reactive power output (in per-unit), as an absolute value (without sign).

“voltageChangeTolerance” is the change in voltage from one control iteration solution to the next and is also a determining parameter to stop additional control iterations. This is the difference between the present per-unit voltage at the terminals of the PVSystem and the prior control iteration PVSystem terminal voltage (in per-unit) as an absolute value.

So, even if the XYCurve is defined with a very high aggression or if the averaging window length is defined by a large window, these factors would limit the reactive power exchange and may also lead to control iteration issues if the values are not set to reasonable values.

4.8 RESULTS AND ANALYSIS OF SMART INVERTER CONTROL FUNCTIONS

This section analyzes the obtained results on the impacts of various smart inverter functions in mitigating voltage related issues like under voltage, over voltage and highly varying voltage profile due to variation in the solar irradiation. The effectiveness of each of the control functions is evaluated and analyzed in four different scenarios in this study by observing all the metrics mentioned in Section 4.7. The effectiveness of each of the control functions is compared against clear sunny day & intermittent cloudy day and between different sizes of PV integrated into the distribution network as well. Days with different irradiance fluctuations are chosen to study the ability of smart inverter controlled PV systems in smoothing the voltage profile through reactive power exchange monitored by studied control functions. Two different sizes of PV integration scenarios are discussed in this study. In the first scenario, the size of PV integrated at the location is taken to be the size of the load attached to the common bus which makes the total size of PV integrated into the distribution network to be 0.18 MW. PV integration of the size same as that of the load is chosen particularly to see the impact of smart inverter control functions studied in mitigating under voltage violations. In the next scenario, the size of PV is increased to approximately 4 times the load size making it a total of 0.7 MW of PV to simulate a very high PV penetration situation. A penetration of this level would definitely introduce an overvoltage condition at the local terminal and is chosen to study the capability of the smart inverter control functions in mitigating over voltage conditions. So, the effectiveness of the smart inverter control functions are evaluated in four scenarios as mentioned:

1. Scenario 1: Clear sunny day and low PV penetration.
2. Scenario 2: Intermittent cloudy day and low PV penetration.
3. Scenario 3: Clear sunny day and high PV penetration.
4. Scenario 4: Intermittent cloudy day and high PV penetration.

Table 4-2: List of Scenarios.

Scenario	**Size of PV integration**	**Solar condition**	**Irradiance curve**
1	0.18 MW	Clear sunny day	Figure 4.18
2	0.18 MW	Intermittent cloudy day	Figure 4.19
3	0.7 MW	Clear sunny day	Figure 4.18
4	0.7 MW	Intermittent cloudy day	Figure 4.19

Each of these Sections 4.8.1 through 4.8.4 discuss the effectiveness of all the smart inverter control functions. All of the analyses is conducted based on measurements at Bus (x_39754_3) which is in phase B and is chosen because it experiences the most under voltage violation before PV is integrated as shown in Table 3-3.

As explained in Section 4.7, the effectiveness of intelligent volt-var is tested by varying the slope of the volt-var curve that varies the amount of reactive power exchange with the distribution network. The effectiveness of adaptive volt-var, dynamic reactive current and combined function control methods are tested by varying corresponding averaging window lengths. Though both the functions depend on the window length, it should be mentioned that the purpose of determining window lengths differ in adaptive volt-var and dynamic reactive control functions.

Table 4-3: Range and variability index of the scenarios without smart inverter in service.

Scenario	Size of PV integration	Range (pu)	Variability index (pu)
1	0.18 MW	0.0714	0.2237
2	0.18 MW	0.0763	0.8022
3	0.7 MW	0.1307	0.5793
4	0.7 MW	0.1340	2.7133

Table 4-3 details the range and variability index of base cases of each scenario when PV of specified sizes are integrated into the distribution system without enabling any smart inverter functions. The range and variability index obtained when any of the control function is in service should be compared to the reference values listed in Table 4-3.

4.8.1 Scenario 1: Clear Sunny Day and Low PV Integration

This section discusses the effectiveness of intelligent volt-var, volt-var with an adaptive set point, dynamic reactive current and combined control algorithms with a 0.18 MW of PV integration on a clear sunny day.

4.8.1.1 Intelligent Volt-Var Control (VVC) Analysis

Figure 4.21 shows the voltage profile as observed at bus: x_39754_3.2. It is observed that as the slope of the volt-var curve is increased, the smart inverter forces the PV system to inject or absorb more reactive power to meet the voltage regulation requirement. It should also be noted that the scenario with no smart inverters installed experiences an under voltage scenario which is absent in other cases with volt-var control function implemented. A maximum of 0.725% loads is experiencing under voltage violations at 7:36 pm on peak loading day when no smart inverters are employed and the under voltage violations are eliminated with the implementation of volt-var controlled smart inverters.

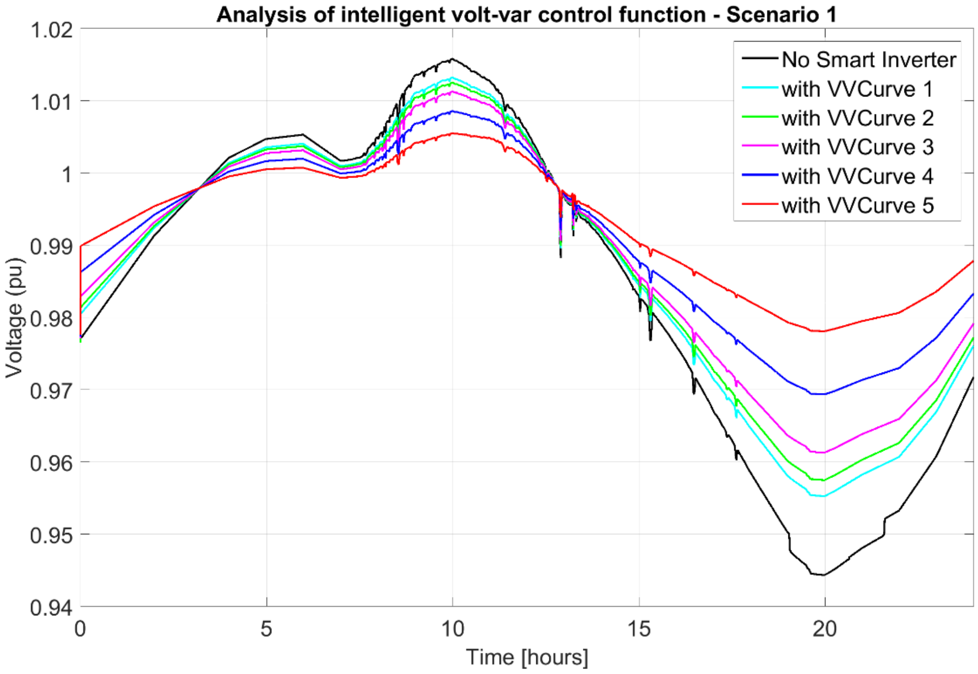


Figure 4.21: Voltage profile of PV with various volt-var curves.

Though volt-var control is implemented to reduce the voltage range, variability index is also compared alongside voltage range in this study as the variability index is also seen to be effectively lowered by this control function. This is because variability index is a function of the change in voltage and as intelligent volt-var control aims at reducing this voltage change, variability of the voltage is also addressed.

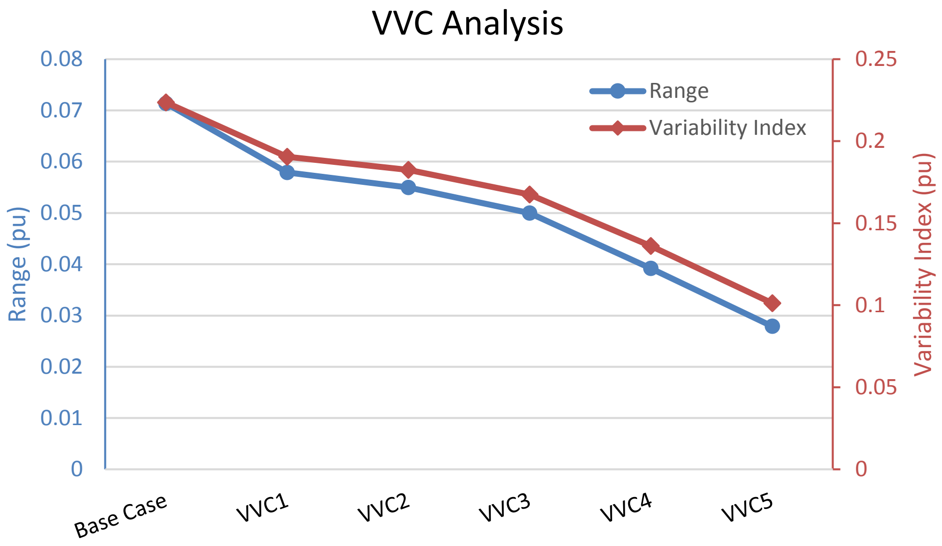


Figure 4.22: Comparison of range and variability index observed with different VVC curves.

The range and variability index of the voltage profile of the selected PV during each of the 5 chosen curves and the base case is shown in Fig 4.22. It is observed that both the metrics are lowered with the increased aggression of the intelligent volt-var control curves. The percentage of loads that are prone to experience under voltage violations is also lowered with the increased aggression of the curves as shown in Fig 4.23.

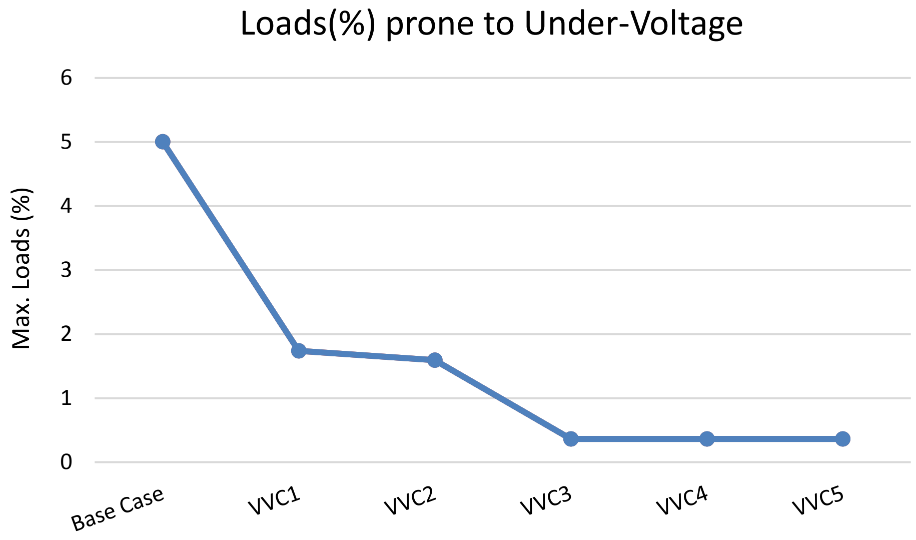


Figure 4.23: Comparison of effectiveness of different VVC curves in avoiding under voltage violations.

4.8.1.2 Adaptive Volt-Var Control (AVVC) Analysis

Figure 4.24 shows the voltage profile as observed at bus: x_39754_3.2. It is observed that as the averaging window length of the curve is increased, the smart inverter forces the PV system to inject or absorb reactive power to experience a regulated and smooth voltage profile. Based on Figure 4.25, the scenario involving adaptive volt-var curve with longest averaging window length has observed the least range as well as the variability index compared to influence by curves with other window lengths. The percentage of load that is prone to under voltage violations remain the same with an averaging window length of 2m and have reduced to zero with higher averaging window lengths.

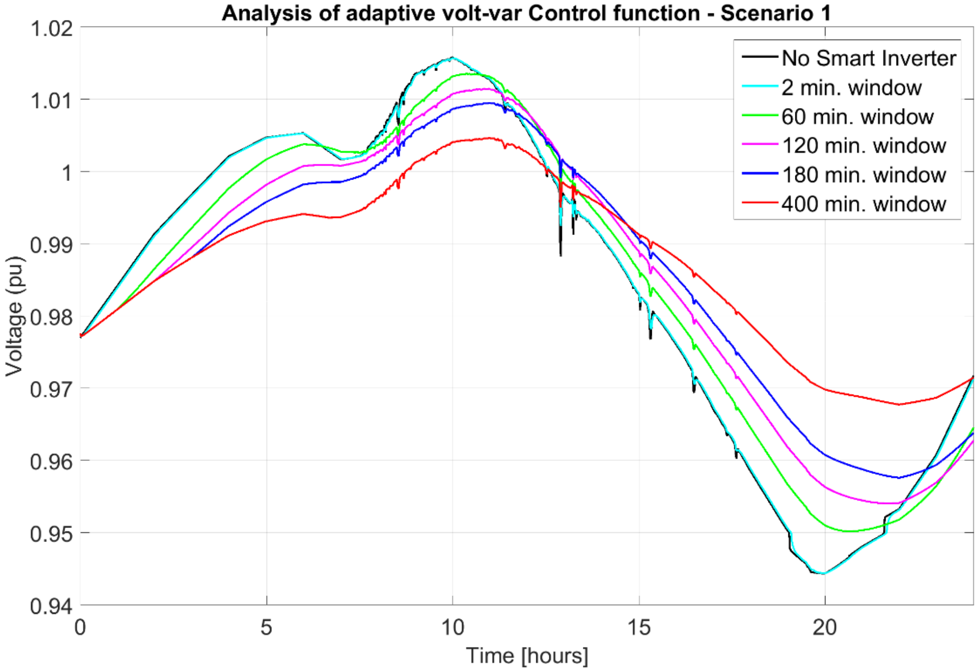


Figure 4.24: Voltage profile of PV with various averaging window lengths.

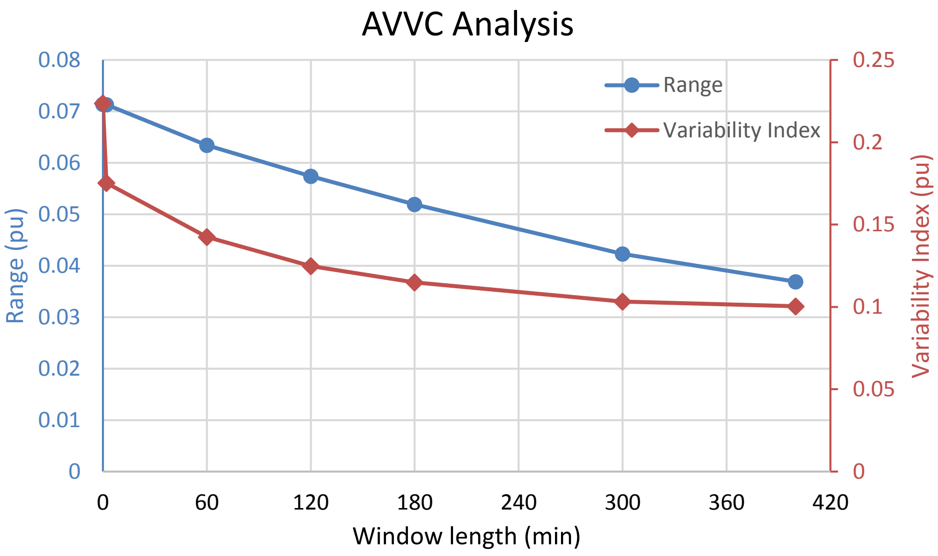


Figure 4.25: Comparison of range and variability index observed with different averaging window lengths of AVVC.

The voltage profile observed with adaptive volt-var control function in service seem to have a delayed peak which is due to the fact that the control function takes the average of previous voltages in determining the amount of reactive power to inject or absorb in order to regulate the voltage. The process of averaging can be clearly observed during starting hours of the simulation where the control functions with different averaging window lengths experience the same average.

4.8.1.3 Dynamic Reactive Current Control (DRCC) Analysis

Figure 4.26 shows the voltage profile as observed at bus: x_39754_3.2 as the dynamic averaging window length is increased. The process of averaging the voltages in determining the amount reactive power to be exchanged can also be observed in this plot during starting hours of the simulation.

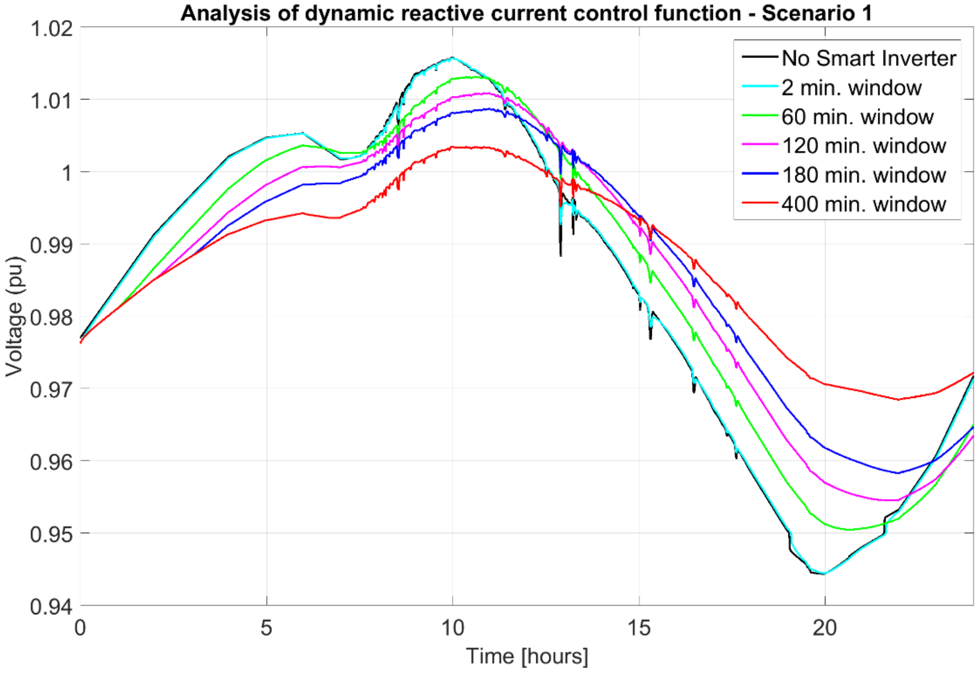


Figure 4.26: Voltage profile of PV with various averaging window lengths with DRCC.

Based on Figure 4.27, it is observed that the range and variability index are lowered as the dynamic reactive average window length is increased. It should also be noted that loads or customers experience under voltage violations with an averaging window length of 2m and have reduced to zero with higher averaging window lengths.

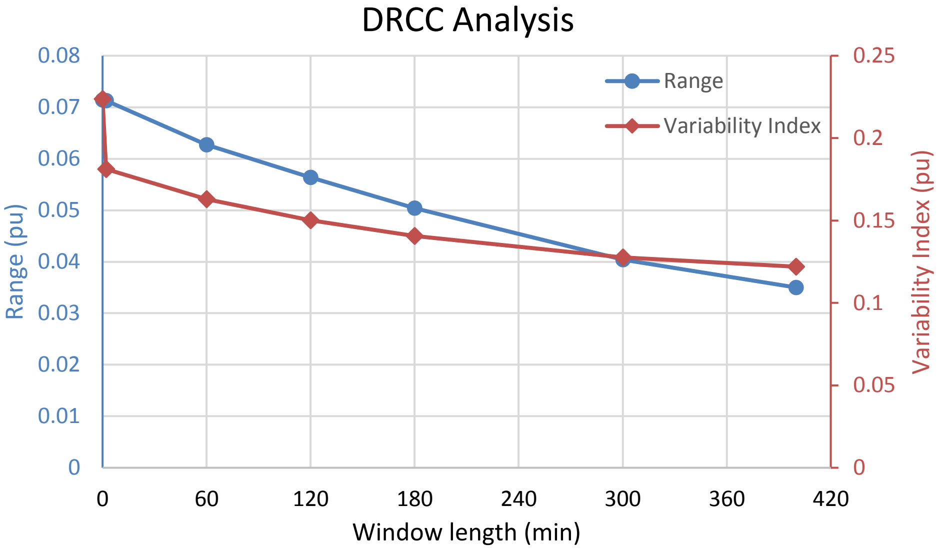


Figure 4.27: Comparison of range and variability index observed with different averaging window lengths of DRCC.

4.8.1.4 Combined Control Function Analysis

Figure 4.28 shows the voltage profile as observed at bus: x_39754_3.2 as the dynamic averaging window length is increased from 2 minutes to 400 minutes.

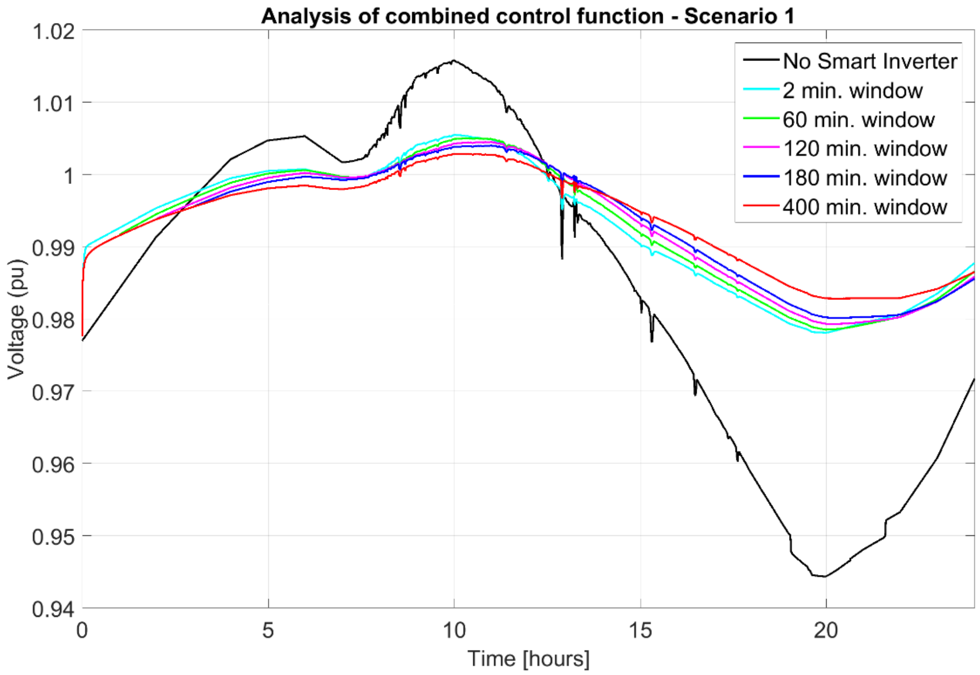


Figure 4.28: Voltage profile of PV with various averaging window lengths with combined control function.

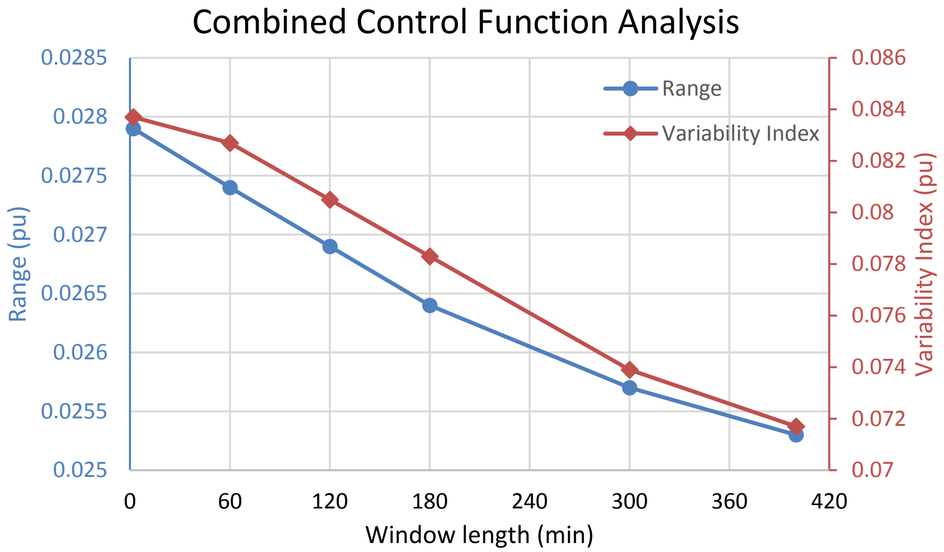


Figure 4.29: Comparison of range and variability index observed with different averaging window lengths of combined control function.

It is quite clear that combined functionality of intelligent volt-var and DRCC helps achieve required regulation and smoothing of voltage profile. A high change in range when comparing PV system with and without combined control function is observed due to the fact that in combined control function, the volt-var curve is set to most aggressive slope for all the varying averaging window lengths and so the variation in the range observed is minimal. The range and variability index of voltage profile observed at bus: x_39754_3.2 are shown in Fig 4.8.1.4-2. The range and variability index are observed to be lowered with increasing dynamic averaging window length.

4.8.2 Scenario 2: Intermittent Cloudy Day and Low PV Integration

This section discusses the effectiveness of intelligent volt-var, volt-var with an adaptive set point, dynamic reactive current and combined control algorithms with a 0.18 MW of PV integration on an intermittent cloudy day.

4.8.2.1 Intelligent Volt-Var Control (VVC) Analysis

Figure 4.30 shows the voltage profile as observed at bus: x_39754_3.2. It is observed that as the slope of the curve is increased, the smart inverter forces the PV system to inject or absorb more of reactive power to meet the voltage requirement as expected. It should also be noted that the scenario with no smart inverters experiences an under voltage scenario which is absent in other cases with volt-var control function implemented. The maximum percentage of loads experiencing under voltage violations when no smart inverters are employed is 0.7252% at 7 pm on peak loading day.

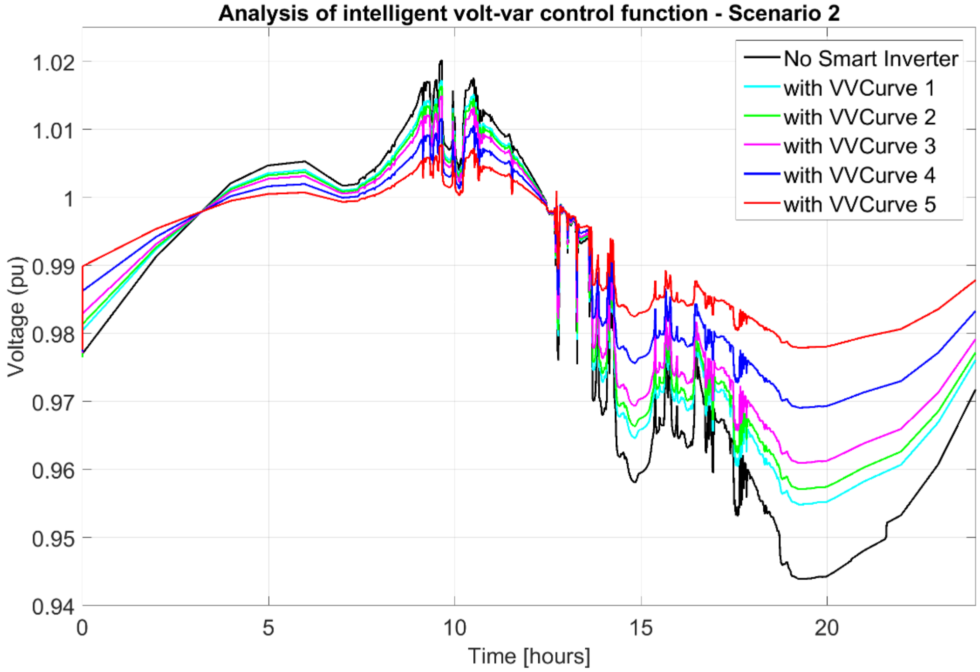


Figure 4.30: Voltage profile of PV with various volt-var curves.

The Range and Variability Index of the voltage profile of the selected PV during each of the 5 chosen curves is shown in Fig 4.31. It is observed that both the range and variability index are lowered with increasing aggression of the intelligent volt-var control curves. The percentage of loads that are prone to experience under voltage violations is also lowered with increasing aggression of the curves as shown in Fig 4.32.

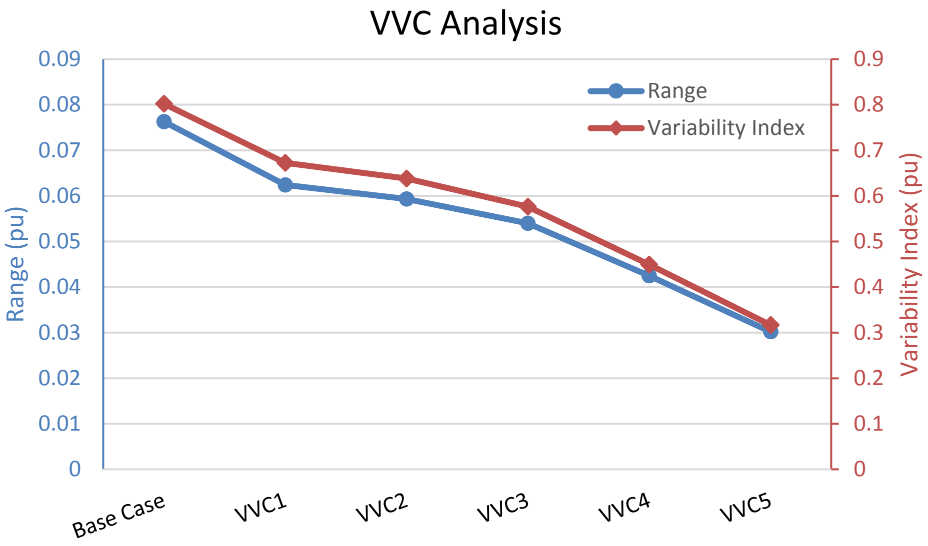


Figure 4.31: Comparison of range and variability index observed with different VVC curves.

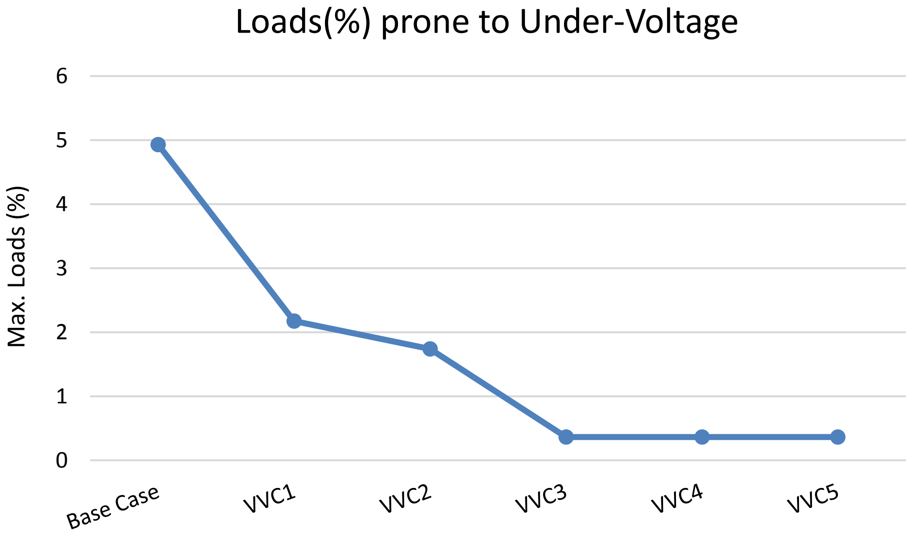


Figure 4.32: Comparison of effectiveness of different VVC curves in avoiding under voltage violations.

4.8.2.2 Adaptive Volt-Var Control (AVVC) Analysis

Figure 4.33 shows the voltage profile as observed at bus: x_39754_3.2. It is observed that as the averaging window length of the curve is increased, the smart inverter forces the PV system to inject or absorb reactive power to experience a regulated and smooth voltage profile. Based on Figure 4.34, it is observed that both range and variability index of the voltage profile are lowered as the averaging window length is increased.

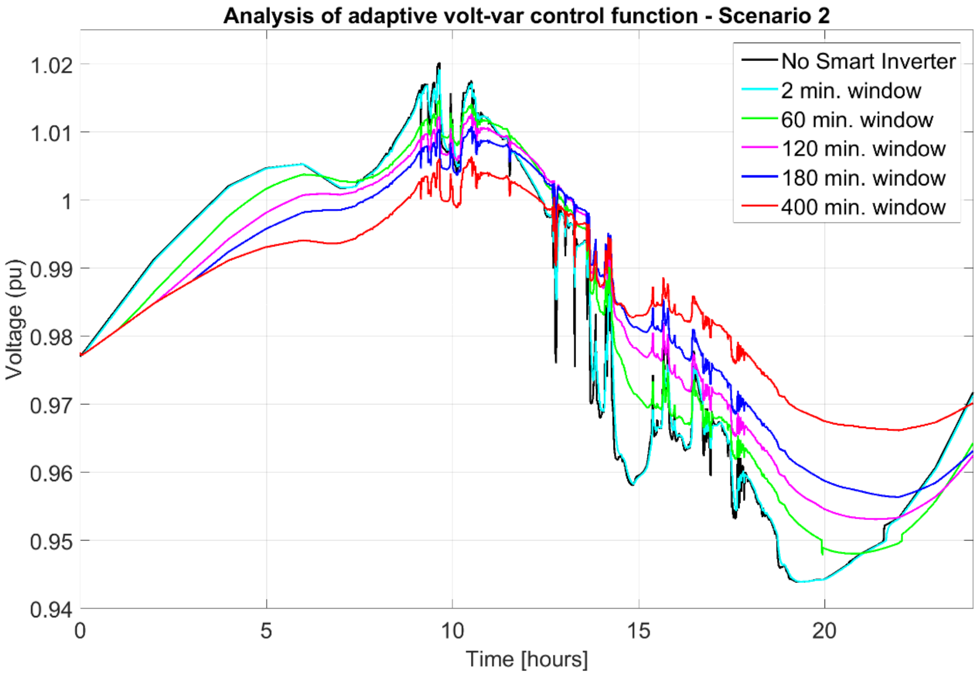


Figure 4.33: Voltage profile of PV with various averaging window lengths.

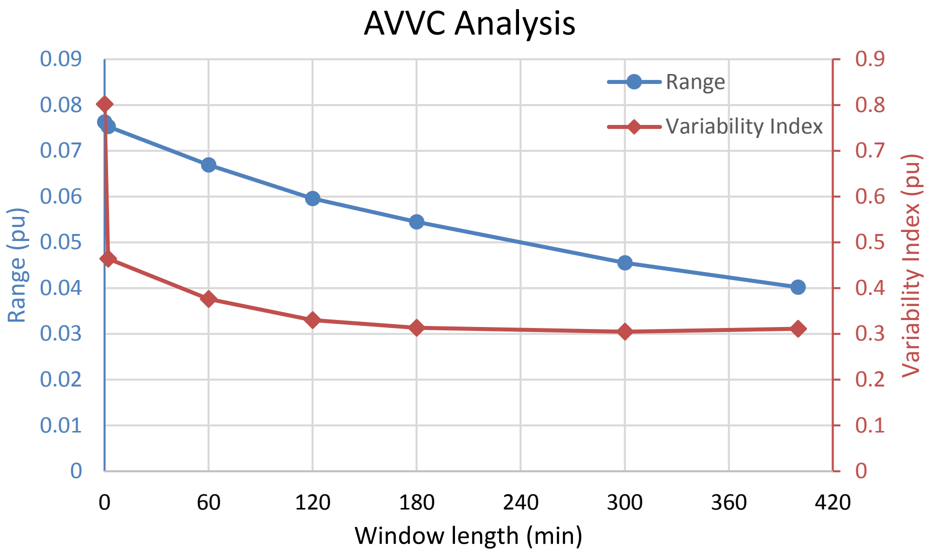


Figure 4.34: Comparison of range and variability index observed with different averaging window lengths of AVVC.

It is observed that the loads experience under voltage violations even when the averaging window length is 1 hour. With averaging window length more than 120 minutes, all the loads experienced a voltage value greater than 0.95 pu during the entire peak day.

4.8.2.3 Dynamic Reactive Current Control (DRCC) Analysis

Figure 4.35 shows the voltage profile as observed at bus: x_39754_3.2 as the dynamic window length is varied.

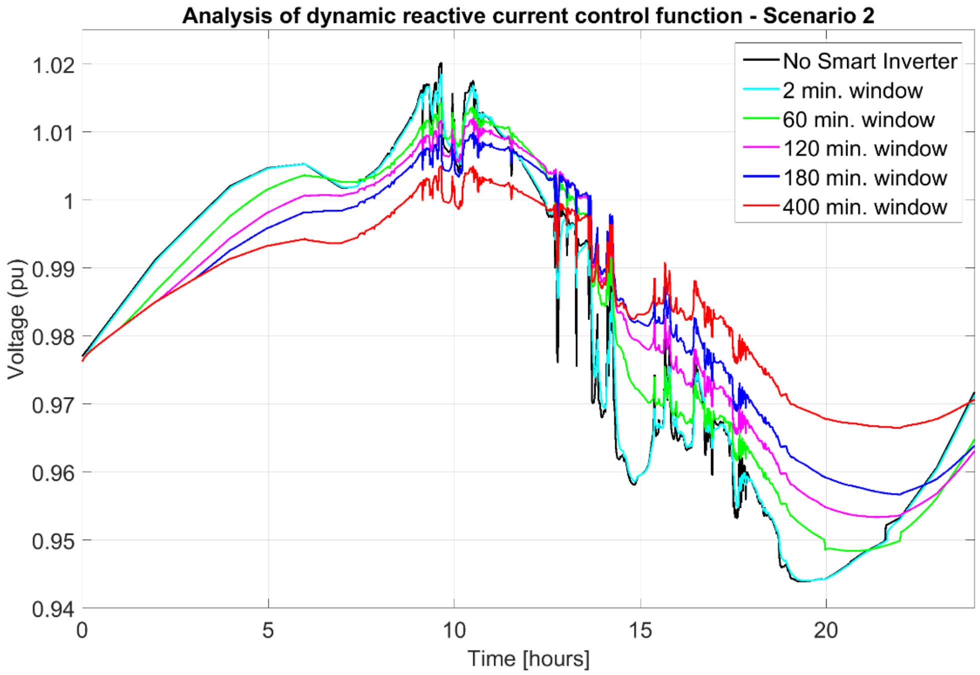


Figure 4.35: Voltage profile of PV with various averaging window lengths with DRCC.

Based on figure 4.36, it is observed that the range and variability index are lowered as the dynamic reactive average window length is increased. It is observed that the loads experience under voltage violations even with an averaging window length of 1 hour. No under voltage violation is observed when window length is increased more than 120 minutes.

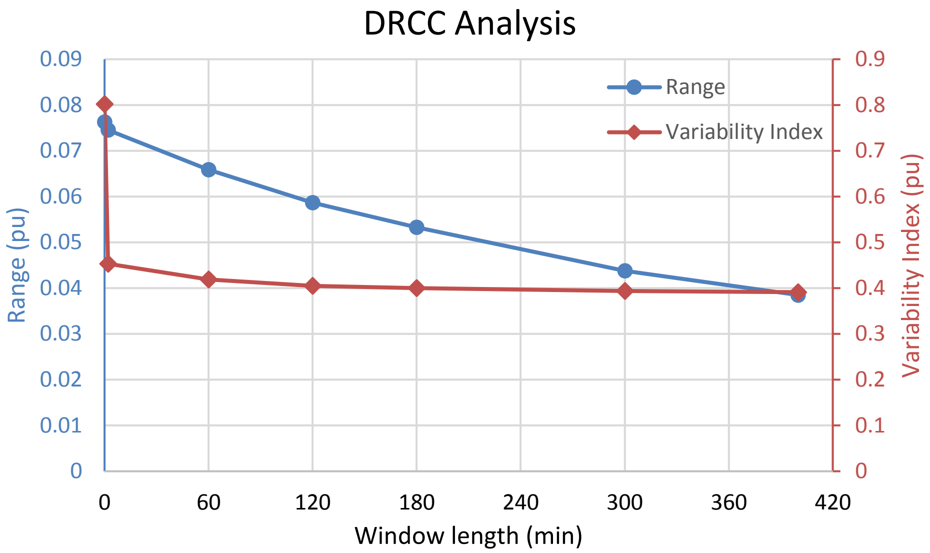


Figure 4.36: Comparison of range and variability index observed with different averaging window lengths of DRCC.

4.8.2.4 Combined Control Function Analysis

Figure 4.37 shows the voltage profile as observed at bus: x_39754_3.2 as the dynamic window length is varied.

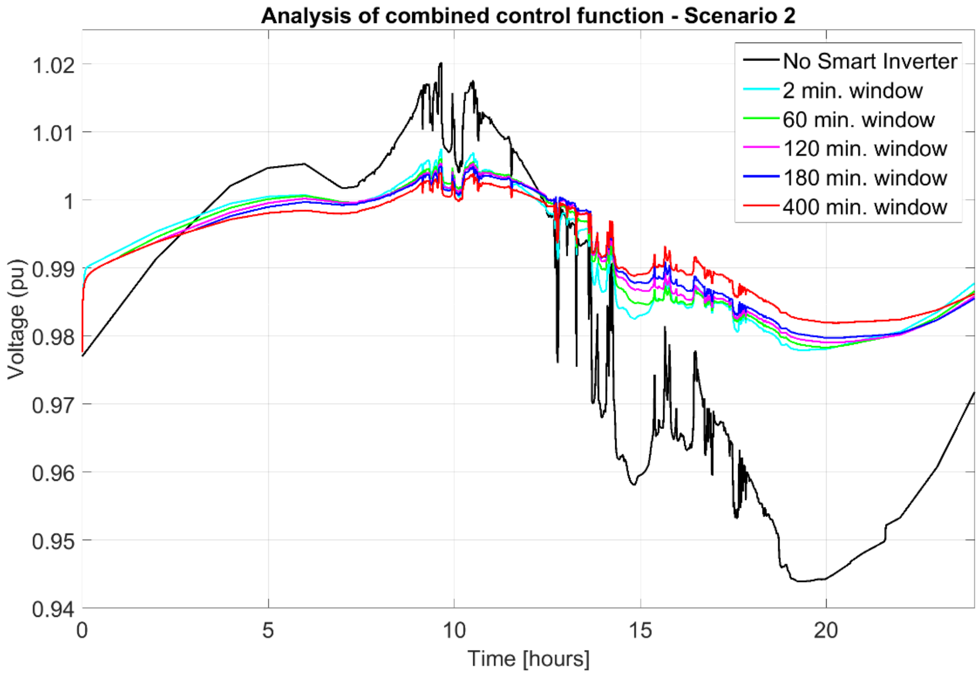


Figure 4.37: Voltage profile of PV with various averaging window lengths with combined control function.

It is quite evident that combined functionality of intelligent volt-var and DRCC helps achieve required regulation and smoothing of voltage profile. The range and variability index of voltage profile observed at bus: x_39754_3.2 is shown in Fig 4.38. The range is observed to be lowered with increasing dynamic averaging window length but variability index drops suddenly to a low value, increases and then reduces as the window length is increased. No under voltage violation are observed and no loads are observed to experience voltage between 0.95 pu and 0.96 pu for any averaging window length.

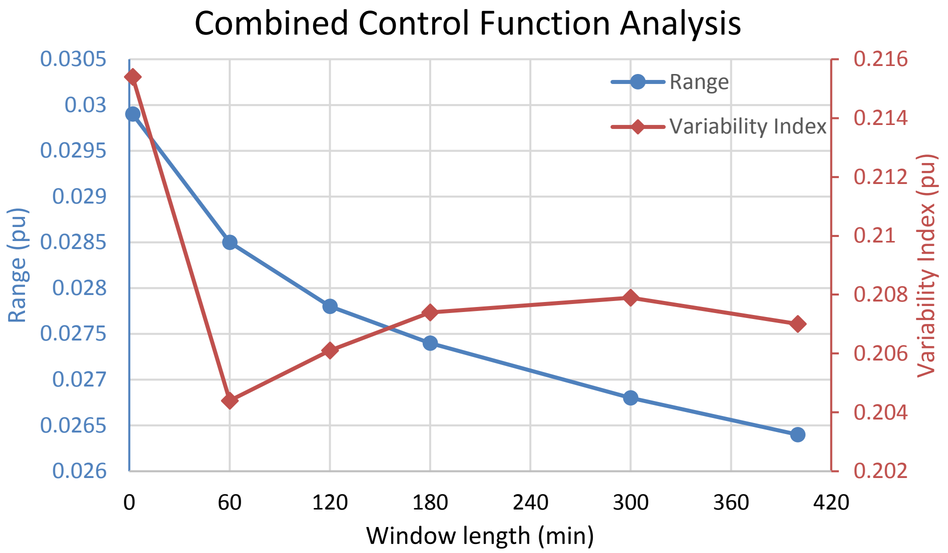


Figure 4.38: Comparison of range and variability index observed with different averaging window lengths of combined control function.

4.8.3 Scenario 3: Clear Sunny Day and High PV Integration

This section discusses the effectiveness of intelligent volt-var, volt-var with an adaptive set point, dynamic reactive current and combined control algorithms with a 0.7 MW of PV integration on a clear sunny day.

4.8.3.1 Intelligent Volt-Var Control (VVC) Analysis

Figure 4.39 shows the voltage profile as observed at bus: x_39754_3.2. It is observed that as the slope of the curve is increased, the smart inverter forces the PV system to inject or absorb more of reactive power to meet the voltage requirement. It should also be noted that the loads in the scenario with no smart inverters experiences an under voltage. With a very high PV penetration compared to the size of loads, an over voltage is observed with the low aggression of the volt-var curves and are eventually seen to be eliminated as the volt-var control curves with higher slope are implemented. The maximum percentage of loads experiencing overvoltage condition due to high PV penetration when no smart inverters are employed is 1.3053%. It should be noted that there exists an over voltage condition experienced by loads in the distribution network with volt-var curves 1 and 2 in service where 1.015% and 0.218% loads are affected respectively.

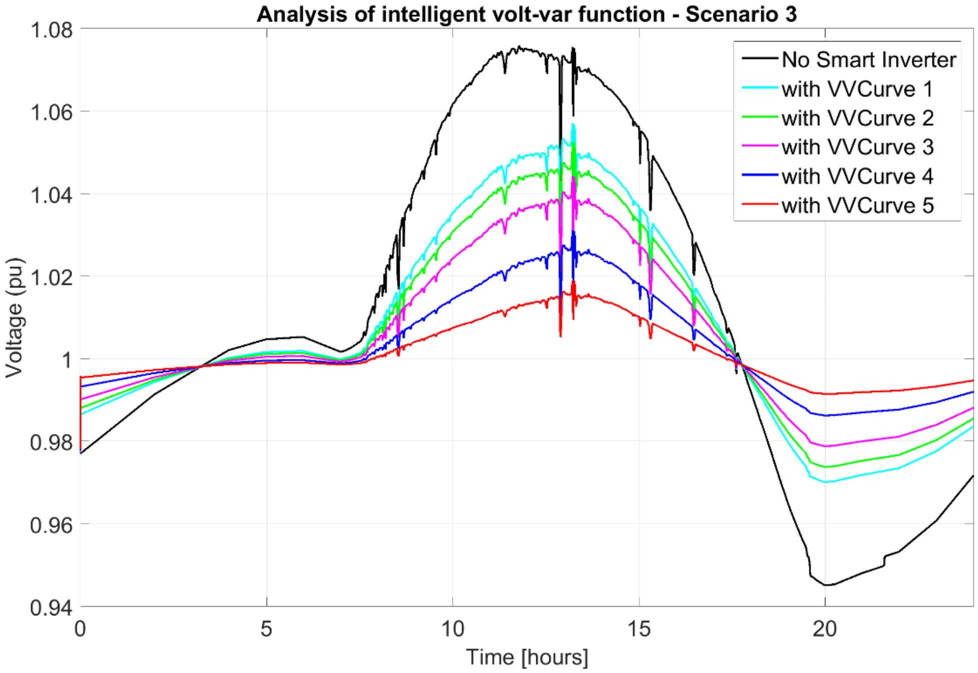


Figure 4.39: Voltage profile of PV with various volt-var curves.

The range and variability index of the voltage profile during each of the 5 chosen curves is shown in Fig 4.40. It is observed that the range and variability index of the voltage profile is lowered with the increased aggression of the intelligent volt-var curve and even the percentage of loads that are experiencing over voltage violations is lowered with the increased aggression of the curves.

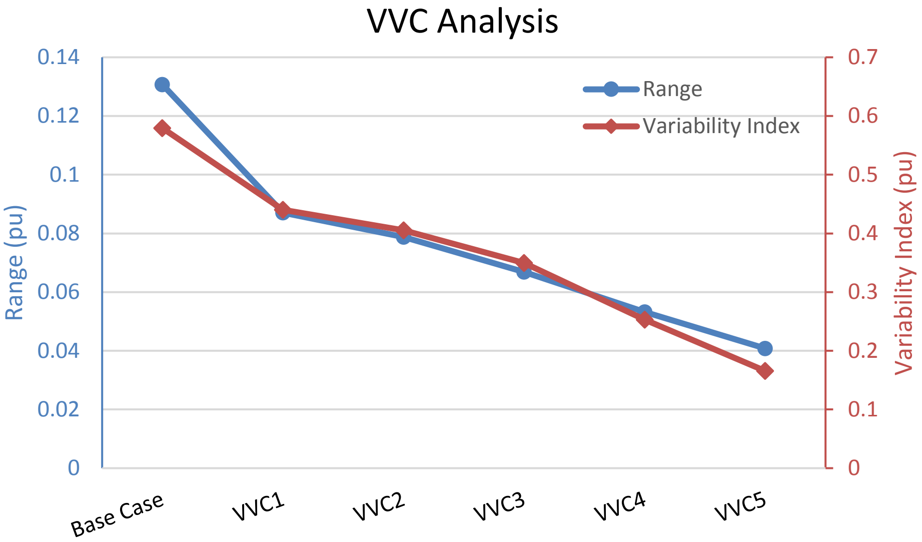


Figure 4.40: Comparison of range and variability index observed with different VVC curves.

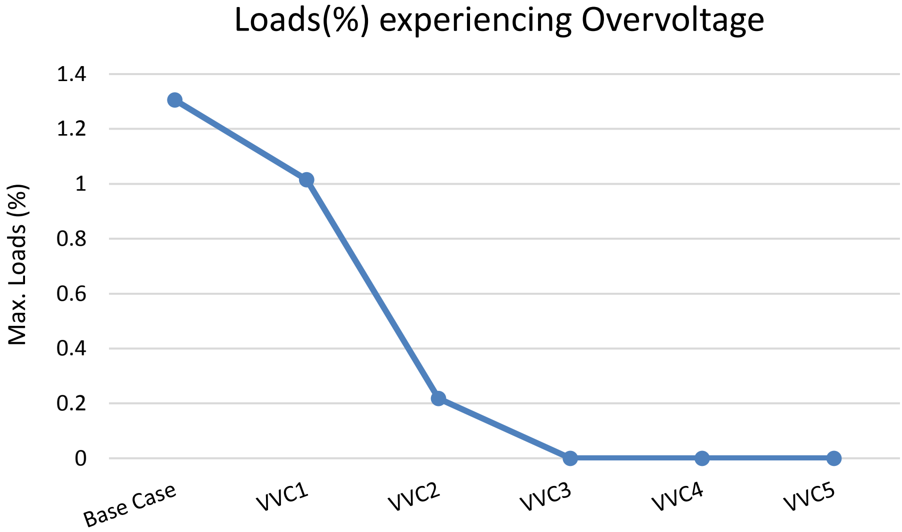


Figure 4.41: Comparison of effectiveness of different VVC curves in mitigating over voltage violations.

4.8.3.2 Adaptive Volt-Var Control (AVVC) Analysis

Figure 4.42 shows the voltage profile as observed at bus: x_39754_3.2. It is observed that as the averaging window length of the curve is increased, the smart inverter forces the PV system to inject or absorb reactive power to experience a regulated and smooth voltage profile. Based on Figure 4.43, the scenario involving adaptive volt-var curve with longest averaging window length has observed the least range as well as the variability index compared to influence by curves with other window lengths.

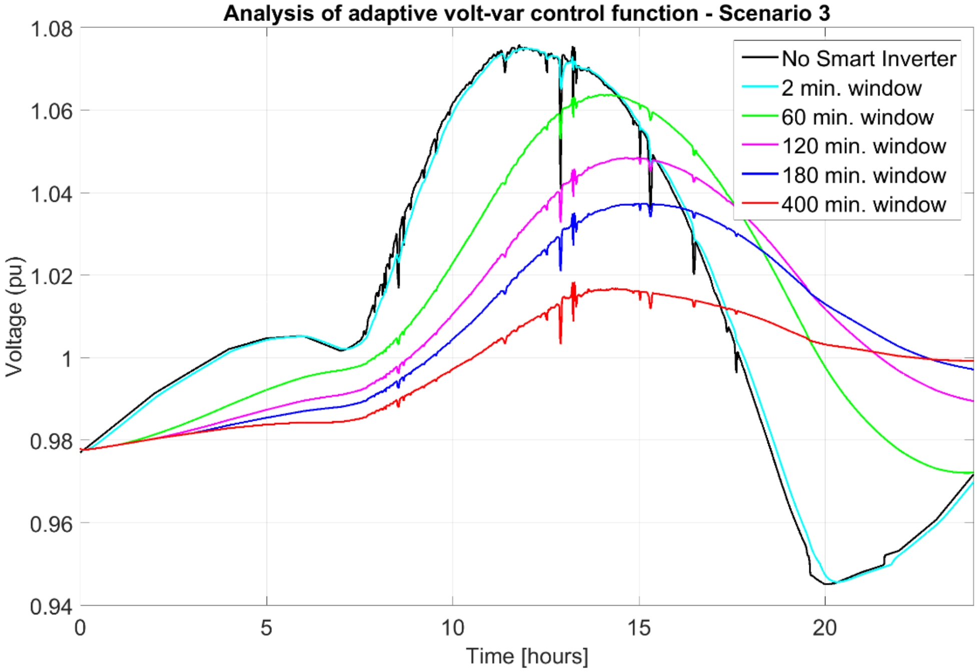


Figure 4.42: Voltage profile of PV with various averaging window lengths.

It is observed that even having a window length of 60 minutes isn't sufficient in mitigating over voltage condition where a maximum of 1.0152% of loads experienced over voltage condition at a particular time during the simulation in this scenario.

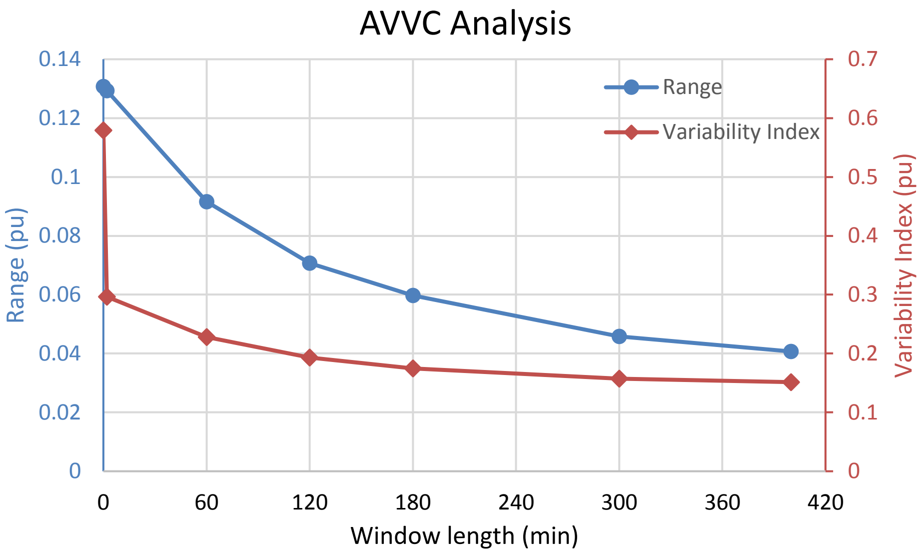


Figure 4.43: Comparison of range and variability index observed with different averaging window lengths of AVVC.

4.8.3.3 Dynamic Reactive Current Control (DRCC) Analysis

Fig 4.8.3.3-1 shows the voltage profile as observed at bus: x_39754_3.2 as the dynamic averaging window length is increased.

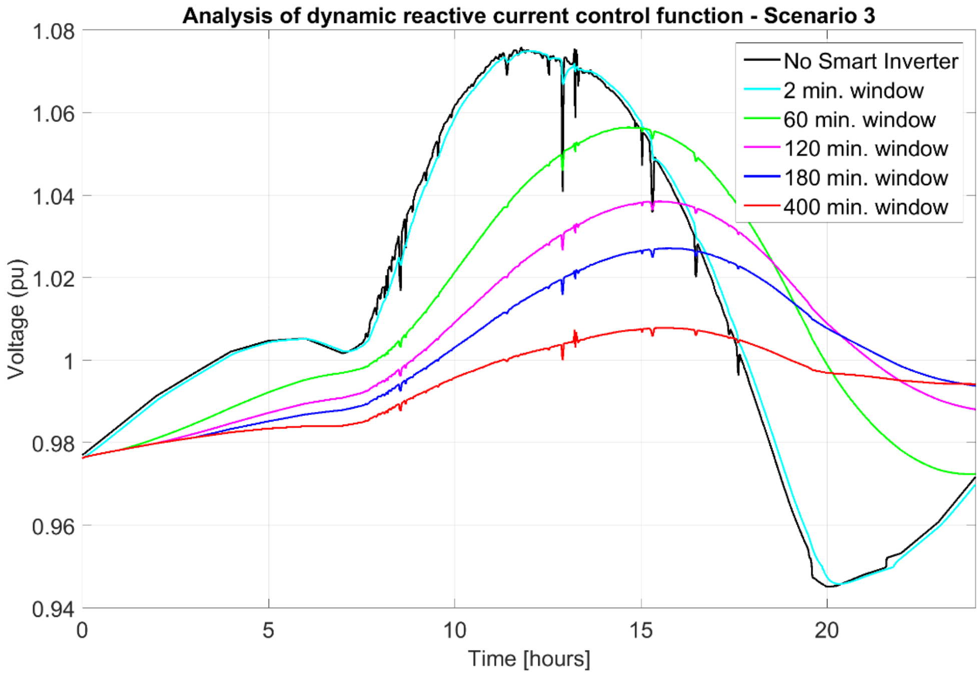


Figure 4.44: Voltage profile of PV with various averaging window lengths with DRCC.

Based on Fig 4.45, it is observed that the range and variability index is lowered as the dynamic reactive average window length is increased. It should also be noted that loads or customers don’t experience any over voltage violations even with a window length of 60 minutes.

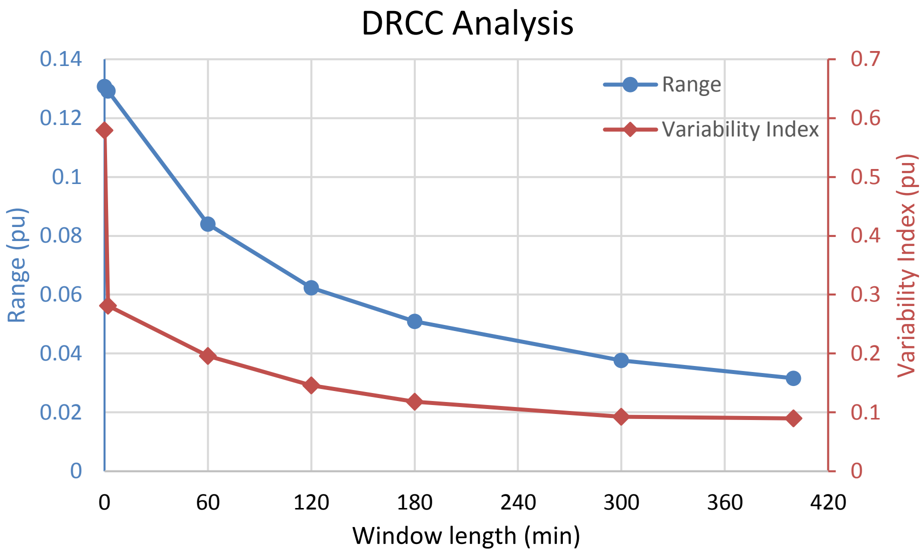


Figure 4.45: Comparison of range and variability index observed with different averaging window lengths of DRCC.

4.8.3.4 Combined Control Function Analysis

Fig 4.46 shows the voltage profile as observed at bus: x_39754_3.2 as the dynamic averaging window length is varied from 2 minutes to 400 minutes.

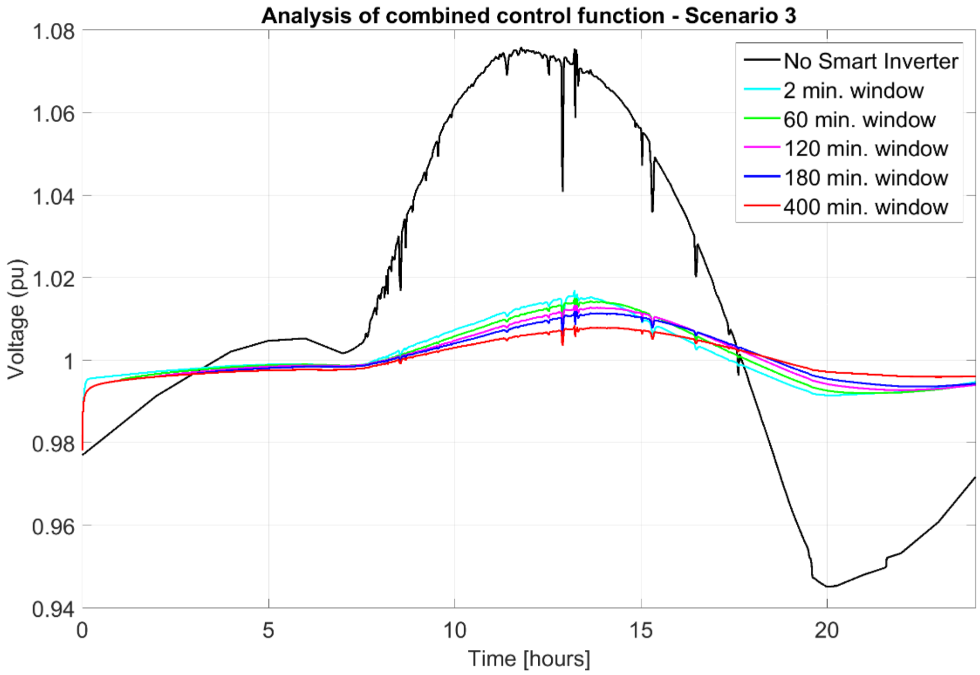


Figure 4.46: Voltage profile of PV with various averaging window lengths with combined control function.

It is quite clear that combined functionality of intelligent volt-var and DRCC helps achieve required regulation and smoothing of voltage profile. The range and variability index of voltage profile observed at bus: x_39754_3.2 is shown in Fig 4.47. The range and variability index are observed to be lowered with increasing window length.

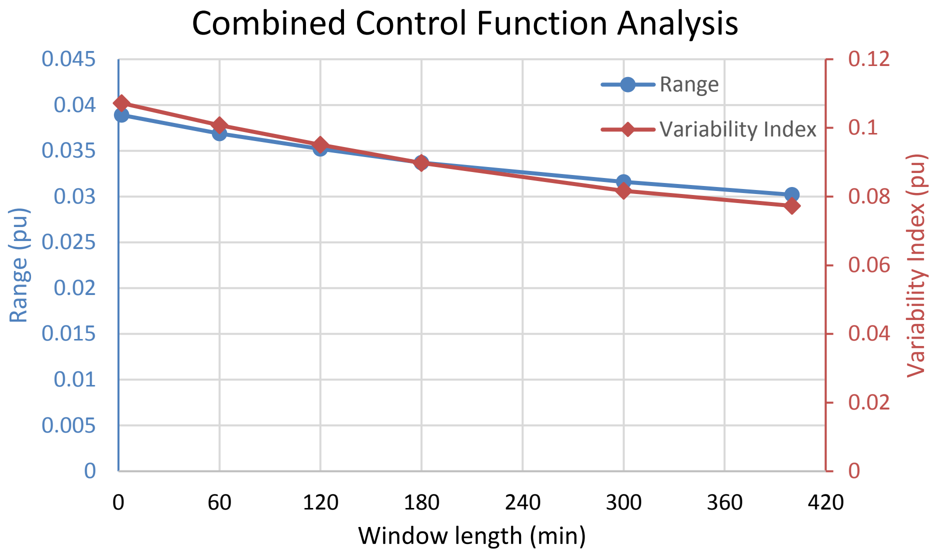


Figure 4.47: Comparison of range and variability index observed with different averaging window lengths of combined control function.

4.8.4 Scenario 4: Intermittent Cloudy Day and High PV Integration

This section discusses the effectiveness of intelligent volt-var, volt-var with an adaptive set point, dynamic reactive current and combined control algorithms with a 0.7 MW of PV integration on an intermittent cloudy day.

4.8.4.1 Intelligent Volt-Var Control (VVC) Analysis

Fig 4.48 shows the voltage profile as observed at bus: x_39754_3.2. It is observed that as the slope of the curve is increased, the smart inverter forces the PV system to inject or absorb more of reactive power to meet the voltage regulation requirement. With a very high PV penetration compared to the size of loads, an over voltage is observed with low aggression of the volt-var curves and are eventually seen to be eliminated as the volt-var control curves with higher slope are implemented. The maximum 1.523% of loads are experiencing overvoltages when no smart inverters are enabled with the integrated PV. It should be noted that there exists an over voltage condition experienced by loads in the distribution network with volt-var curves 1 and 2 in service where a maximum of 1.015% and 0.435% of loads are affected respectively.

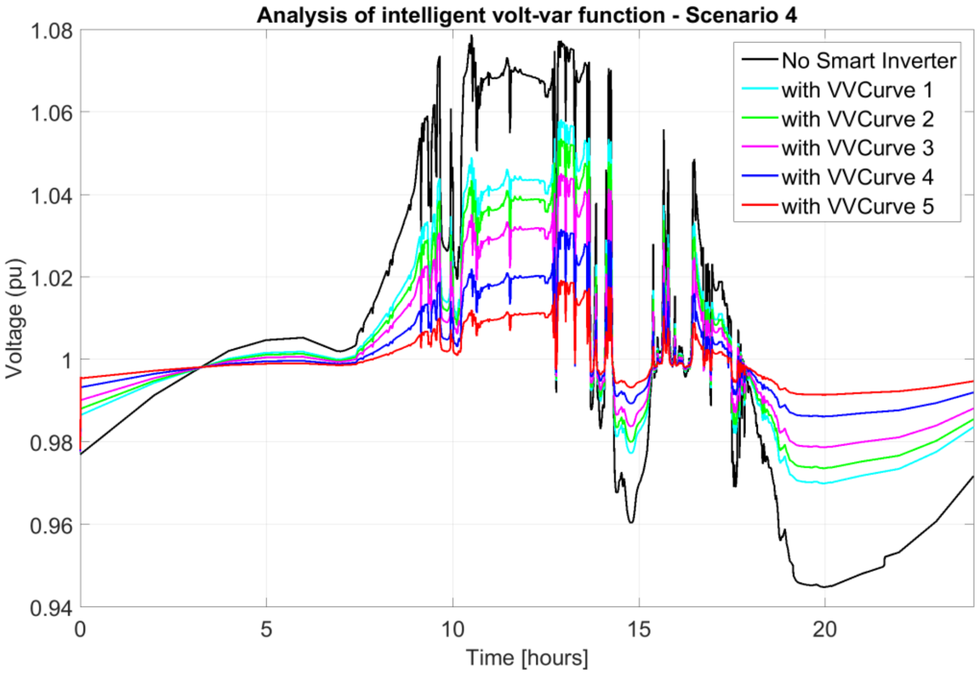


Figure 4.48: Voltage profile of PV with various volt-var curves.

The range and variability index of the voltage profile during each of the 5 chosen curves is shown in Fig 4.49. Both range and variability index metrics are seen to lower as the slope of the volt-var curve is increased.

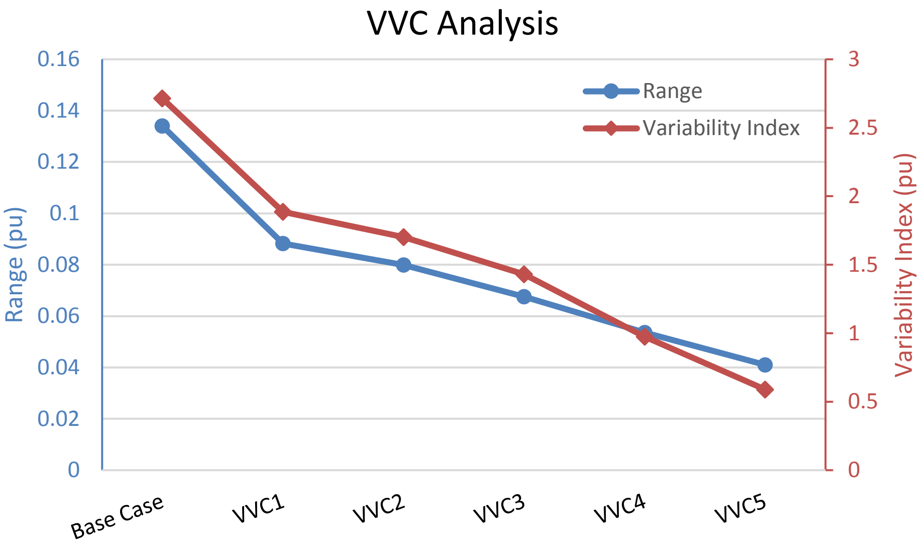


Figure 4.49: Comparison of range and variability index observed with different VVC curves.

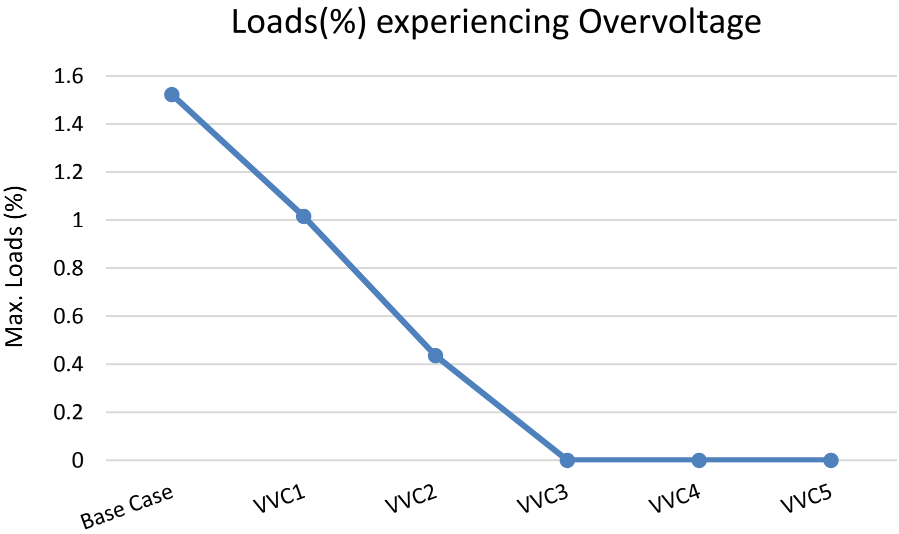


Figure 4.50: Comparison of effectiveness of different VVC curves in mitigating over voltage violations.

4.8.4.2 Adaptive Volt-Var Control (AVVC) Analysis

Fig 4.51 shows the voltage profile as observed at bus: x_39754_3.2. It is observed that as the averaging window length of the curve is increased, the smart inverter forces the PV system to inject or absorb reactive power to experience a regulated and smooth voltage profile. Based on Fig 4.52, the scenario involving adaptive volt-var curve with longest averaging window length has observed the least range as well as the variability index compared to influence by curves with other window lengths.

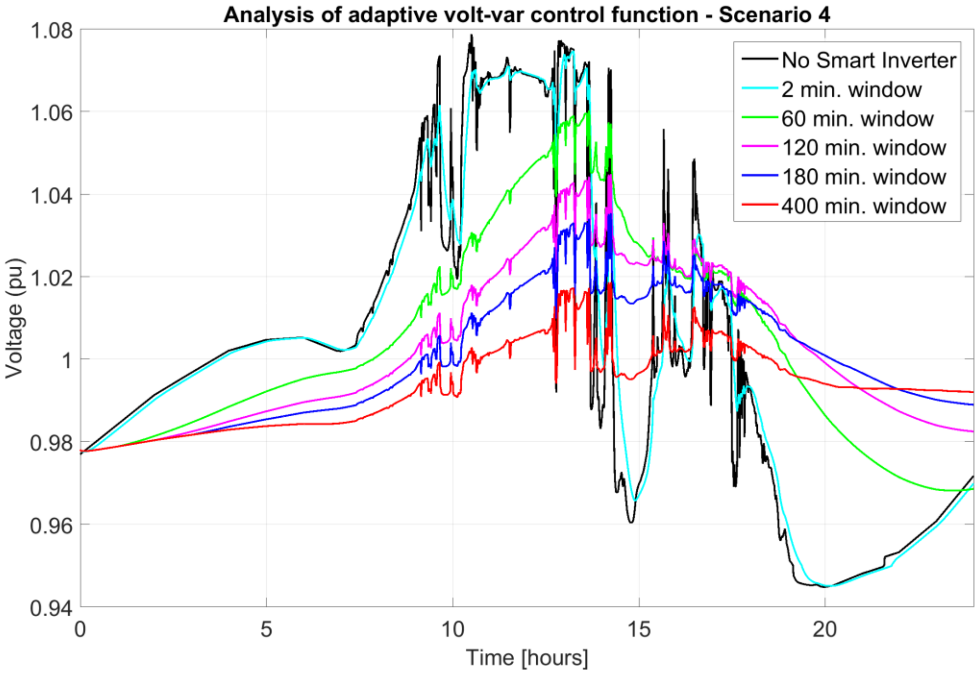


Figure 4.51: Voltage profile of PV with various averaging window lengths.

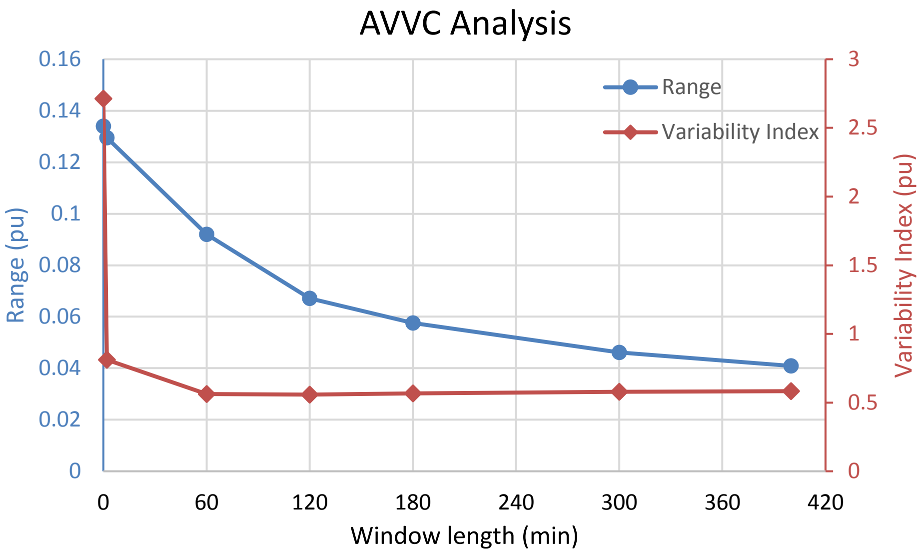


Figure 4.52: Comparison of range and variability index observed with different averaging window lengths of AVVC.

It should be observed that a window length of 60 minutes is not sufficient in mitigating over voltage condition. It is observed that the range of the voltage profile is lowered as observed at bus: x_39754_3.2 for increasing window length. It should be observed that both range and variability index tends to saturate as the averaging window length is increased.

4.8.4.3 Dynamic Reactive Current Control (DRCC) Analysis

Fig 4.53 shows the voltage profile as observed at bus: x_39754_3.2 as the dynamic window length is varied. Based on Fig 4.54, it is observed that both range and variability index have lowered and tend to saturate to particular values as the dynamic reactive average window length is increased. It should also be noted that loads don't experience any over voltage violations and with window length more than 60 minutes, all the loads experience a voltage value greater than 0.96 pu.

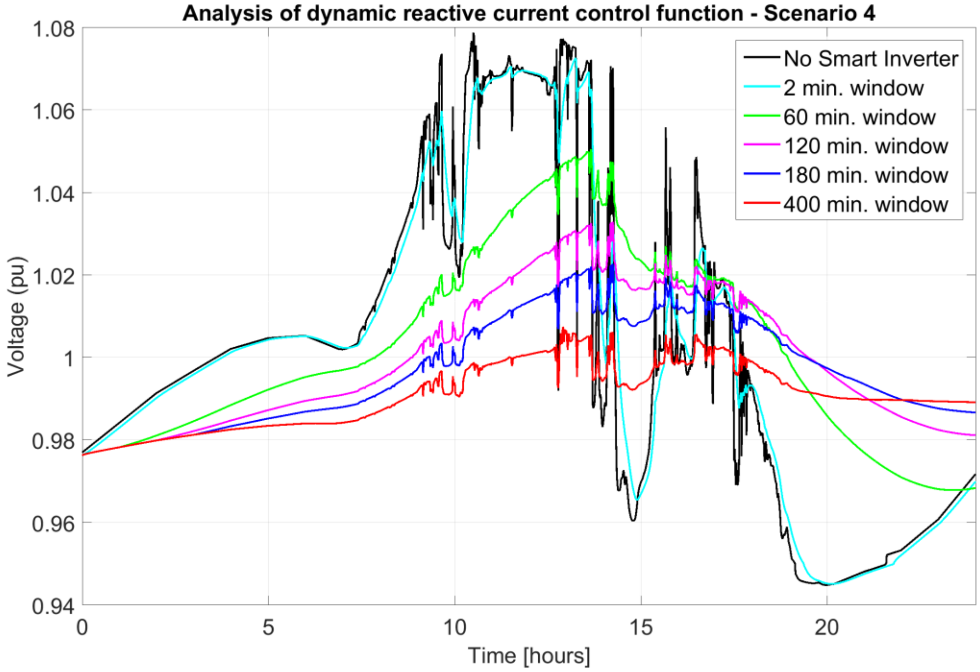


Figure 4.53: Voltage profile of PV with various averaging window lengths with DRCC.

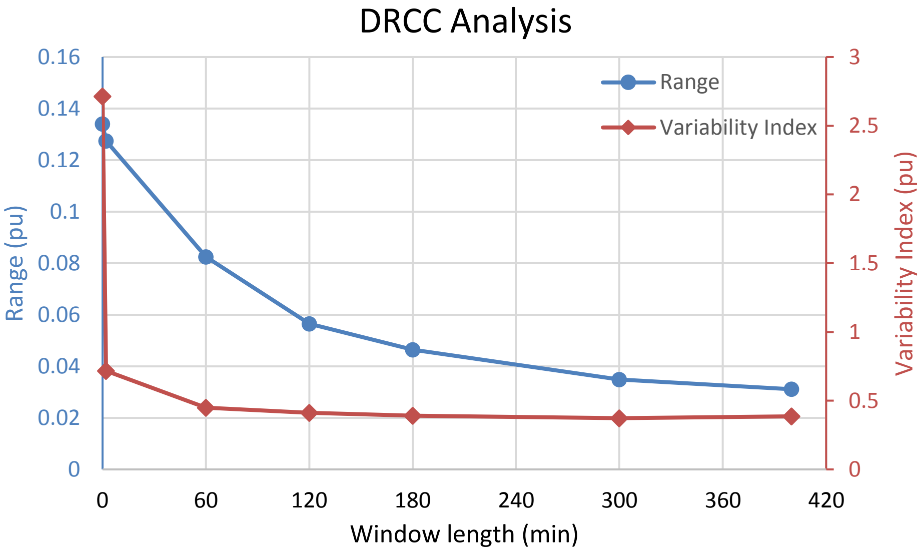


Figure 4.54: Comparison of range and variability index observed with different averaging window lengths of DRCC.

4.8.4.4 Combined Control Function Analysis

Figure 4.55 shows the voltage profile as observed at bus: x_39754_3.2 as the dynamic window length is varied from 2 minutes to 400 minutes.

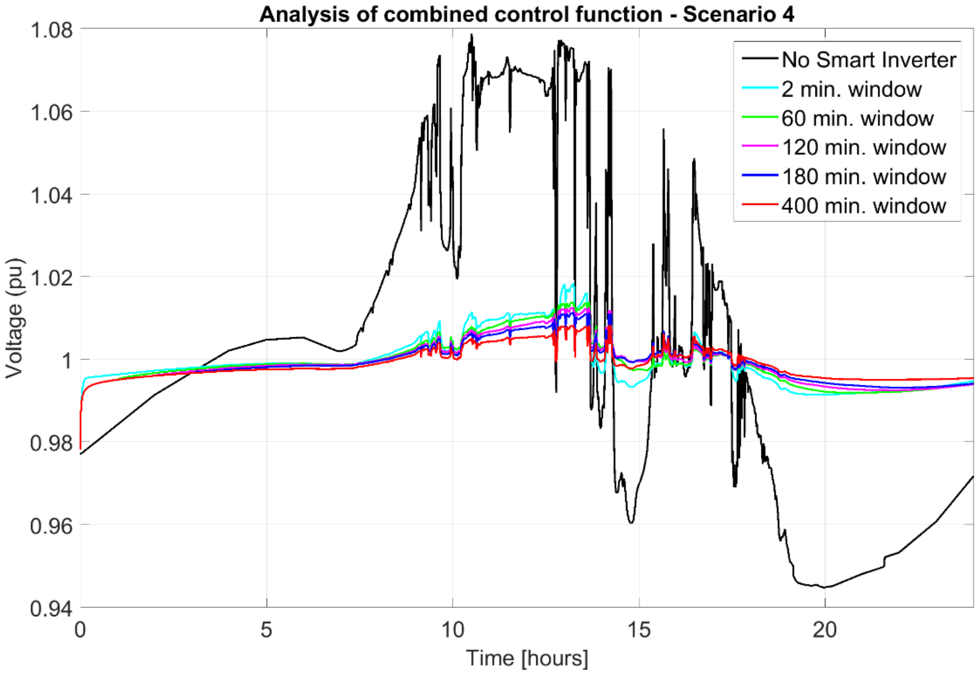


Figure 4.55: Voltage profile of PV with various averaging window lengths with combined control function.

It is quite clear that combined functionality of intelligent volt-var and DRCC helps achieve required regulation and smoothing of voltage profile. The range and variability index of voltage profile observed at bus: x_39754_3.2 is shown in Figure 4.56. The range and variability index are observed to be lowered with increase in averaging window length.

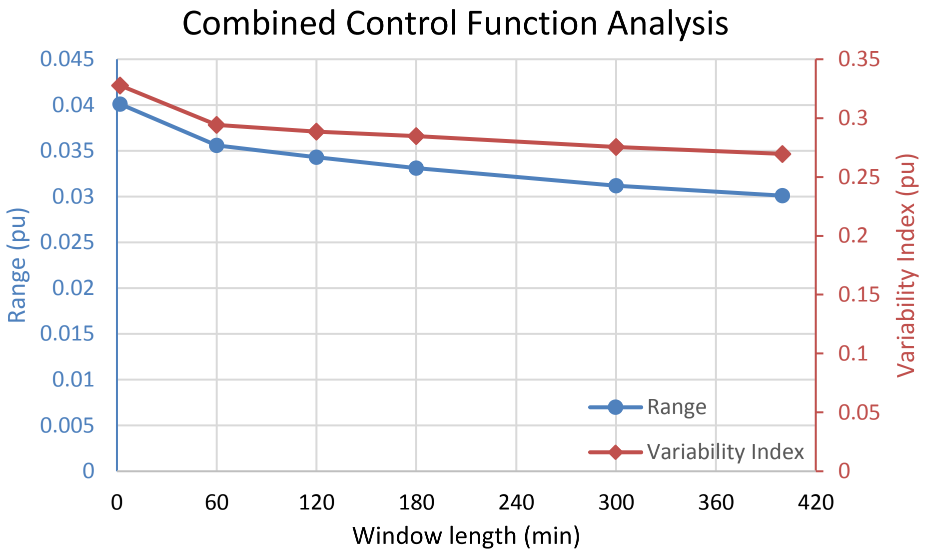


Figure 4.56: Comparison of range and variability index observed with different averaging window lengths of combined control function.

5. SUMMARY AND CONCLUSION

This chapter summarizes the effectiveness of each of the control functions discussed in this study and comes up with a conclusion in determining the suitable control function for smart inverters that would effectively mitigate voltage related issues that rise due to the integration of PV into the distribution network.

5.1 SUMMARY

Utilities generally use sophisticated commercial software for maintenance and operation of the distribution network. To model and evaluate a practical real-world distribution network and to analyze the impact of the integrated PV, the data extracted from the commercial software is to be converted to model the network on a different platform. A detailed conversion process is discussed to model the distribution network in OpenDSS tool that provides the interface to model, analyze and simulate distributed energy resources such as PV, energy storage, and smart inverter control technologies in a distribution network. The conversion process is carried out through VBA scripting that automates the process of accessing the data files and generating .dss files required to model the distribution network in OpenDSS. The conversion process is validated through verification of load flow and short-circuit analyses.

This study is conducted to determine the impacts in the distribution network that rise due to the integration of PV of various size and in different solar conditions. Mathematical derivation supported with quantitative analysis is documented to conclude that common inverter settings can be employed for clustered PVs if the size distribution of such cluster is nearly constant. A detailed study is conducted on various smart inverter technologies that help mitigate voltage variations and fluctuations.

This section summarizes the behavior of each of the control function in all the chosen scenarios. The intelligent volt-var control function has shown a progressive decrease in both the observed range and variability index for increasing capability of exchanging reactive power for all the four scenarios discussed. The intelligent volt-var control is also capable of eliminating under voltage violations with a suitable slope of the volt-var curve. The adaptive volt-var function has shown to be progressive in decreasing the range and variability index of the voltage profile observed in the case of a clear sky scenario but on a cloudy day, the range and variability index are seen to saturate as the averaging window length is increased greater than 180 minutes. The adaptive volt-var control function is not effective in eliminating under voltage violations as it does not push the system to reach any particular voltage rather it settles down to the average value which may lead the system to experience under voltage violation. The dynamic reactive current control function has also shown a similar trend as shown by the adaptive volt-var control function for all the scenarios discussed. The dynamic reactive current control function has a better performance in mitigating voltage related issues over adaptive volt-var control when the voltage fluctuations are high as in scenario 3 and 4 where the high penetration of PV causes high variations in voltage. The combined control function has shown a very minimal change in the range and variability index as the dynamic averaging window length is increased for all the scenarios discussed as the volt-var curve considered is same for all the changing window lengths. The variability index is seen to gradually decrease with increasing window length in cases with high voltage variations like in scenario 3 and 4.

The following comparisons of obtained range and variability index would give a better insight in summarizing the study and in choosing a control function based on the size of PV integrated and the variability of the irradiance.

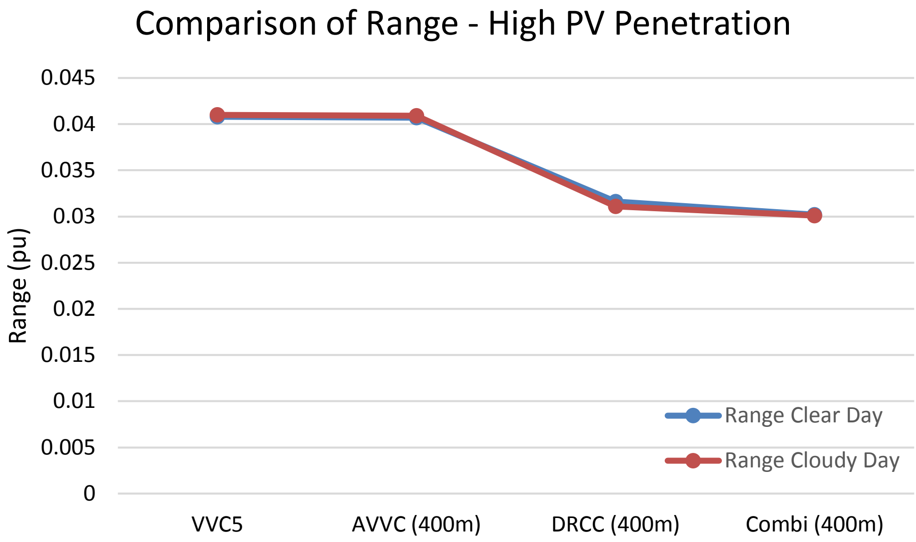


Figure 5.1: Comparison of voltage range with high PV penetration.

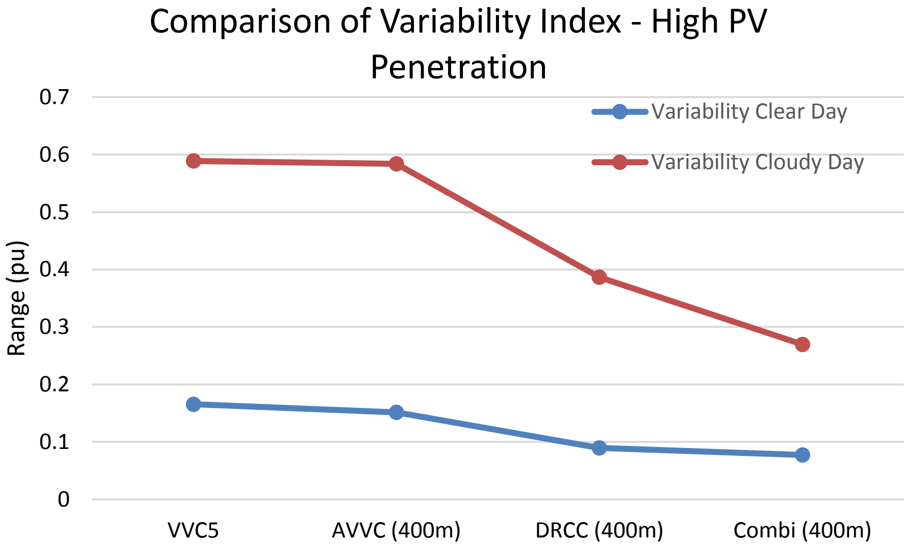


Figure 5.2: Comparison of variability index with high PV penetration.

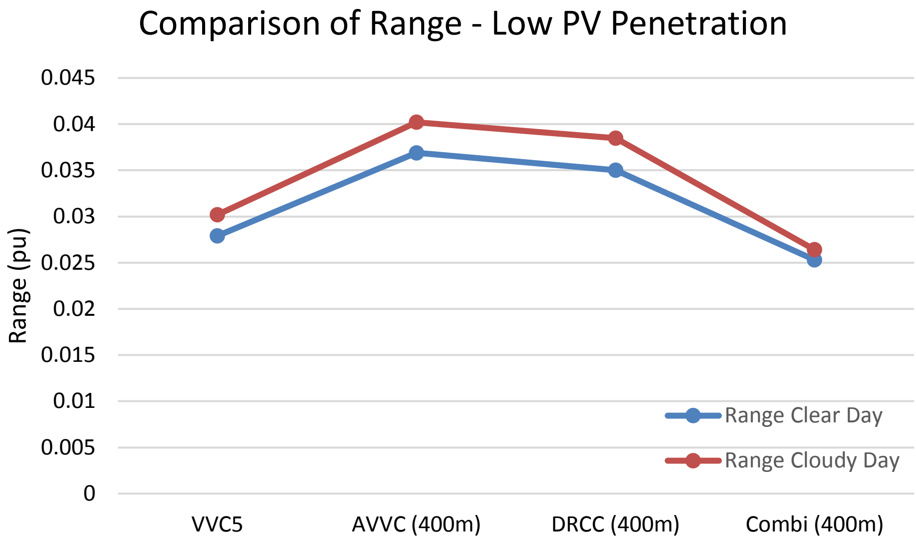


Figure 5.3: Comparison of voltage range with low PV penetration.

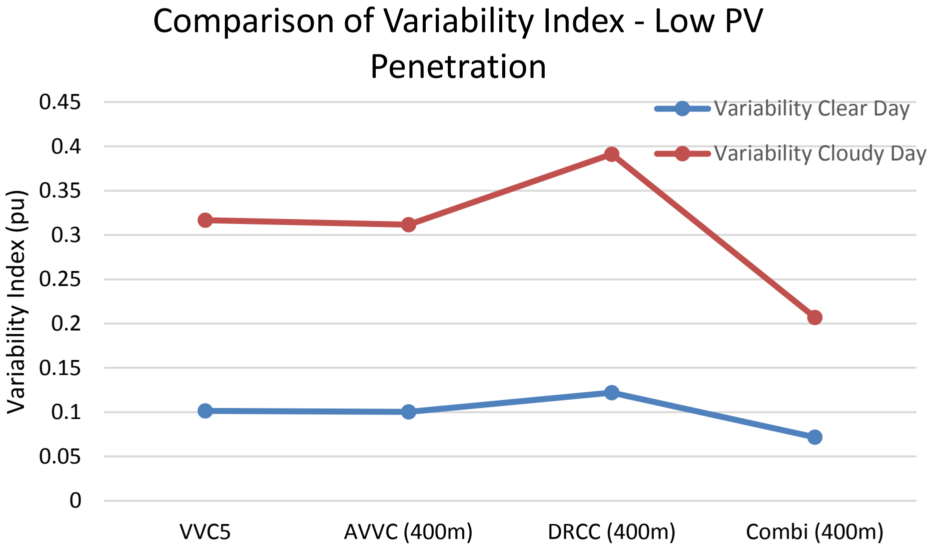


Figure 5.4: Comparison of variability index low PV penetration.

Based on the above analysis, it can be seen that the range and variability index of the voltage profile are higher in the case of a cloudy day than on a clear sunny day as expected. The range and variability index observed are similar when comparing adaptive volt-var control and dynamic reactive current control functions for most of the scenarios. Though various other factors would be involved in choosing a control function, in most of the above-shown comparisons, combined control functionality has a better capability in mitigating the voltage related issues.

In analyzing the capability of combined control function, the reactive power exchange of the smart inverters with the distribution network is compared to other control functions. Scenario 2: Intermittent Cloudy Day with Low PV Penetration is chosen to compare the reactive power exchange by smart inverter under the influence of various control functions.

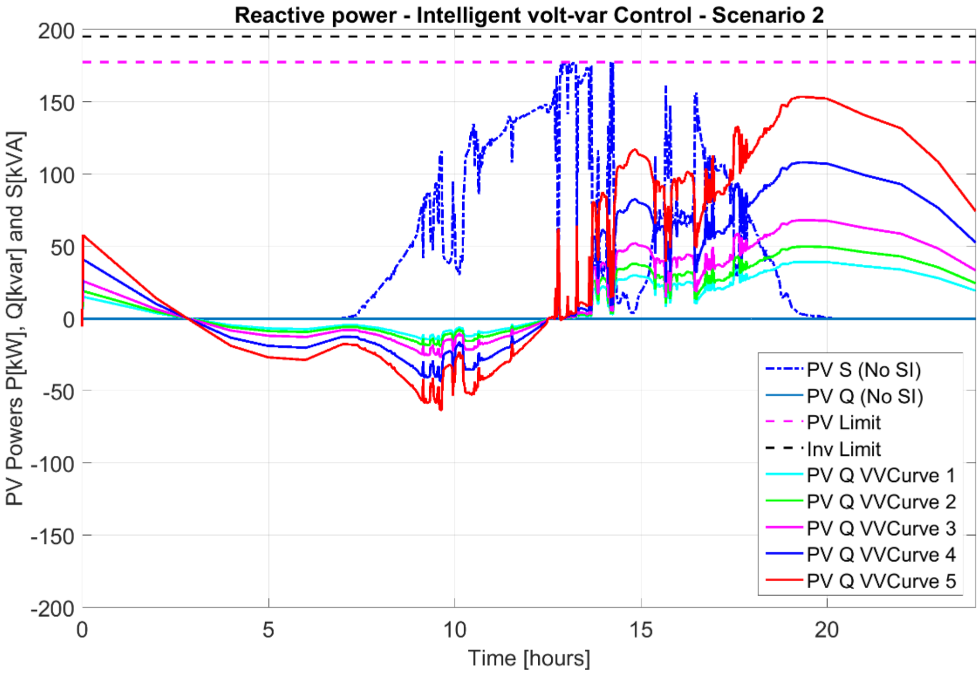


Figure 5.5: Reactive power injection with VVC on an intermittent cloudy day.

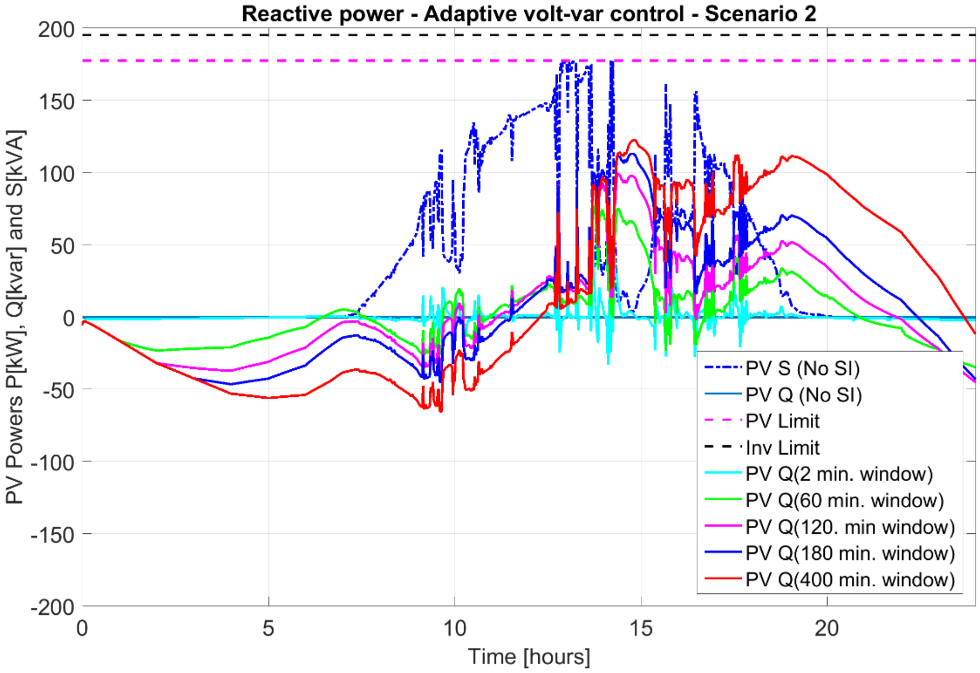


Figure 5.6: Reactive power injection with AVVC on an intermittent cloudy day.

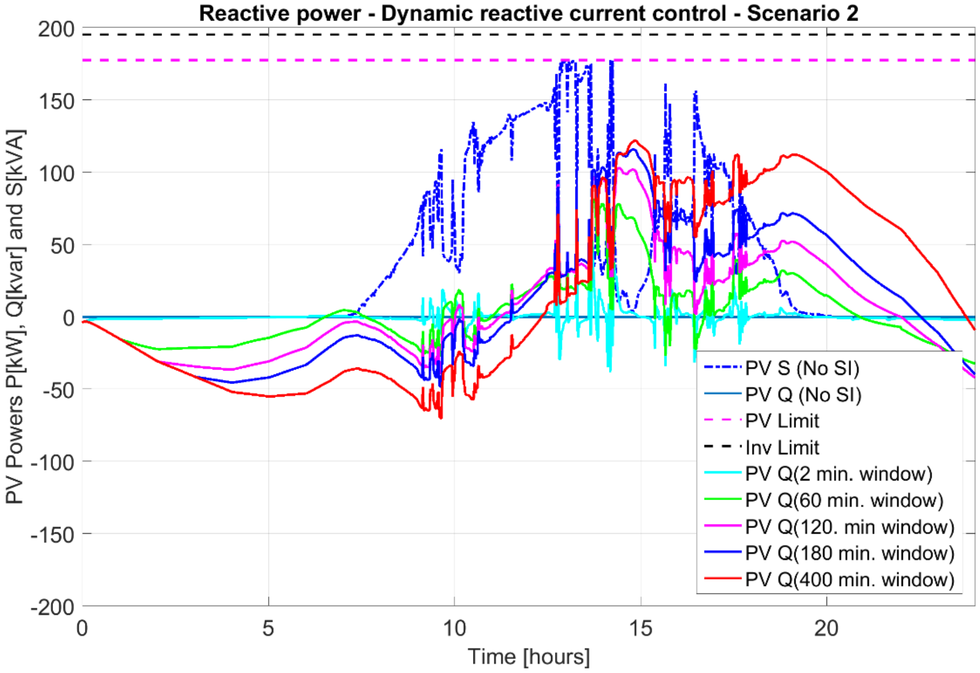


Figure 5.7: Reactive power injection with DRCC on an intermittent cloudy day.

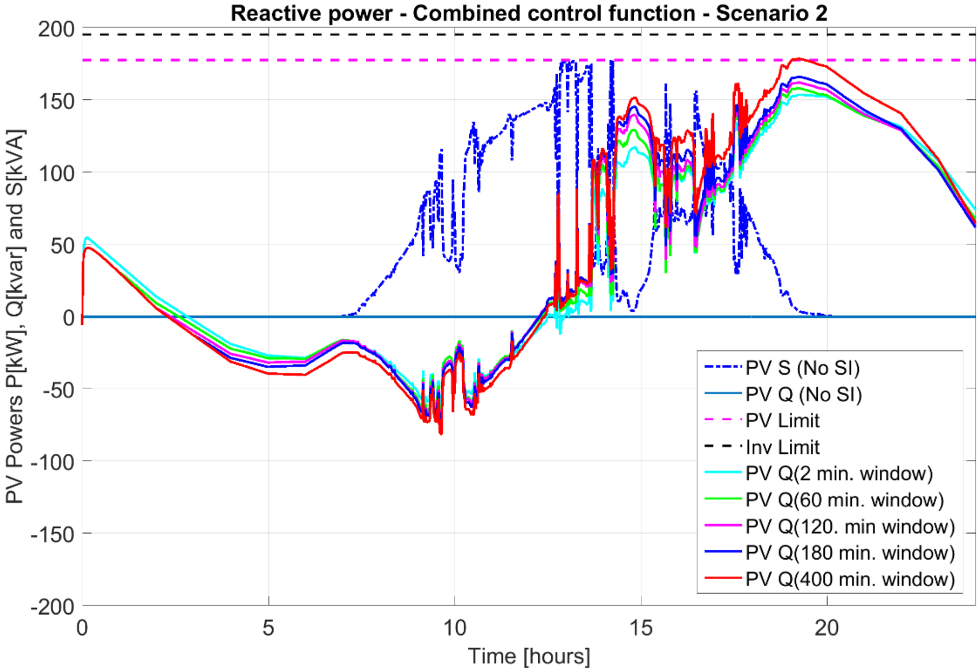


Figure 5.8: Reactive power injection with combined control function on an intermittent cloudy day.

Based on the comparison of reactive power exchanges, it can be seen that the reactive power exchange is higher and efficient in terms of utilizing the entire available reactive power for minimizing both the range and variability index of the voltage profile with the combined control function on smart inverters. The main reason for the better performance of this control function through higher reactive power exchange is that the smart inverter regulates the voltage based on the defined volt-var curve by exchanging available reactive power and also smooths the voltage profile through defined dynamic reactive current control.

5.2 CONCLUSION

Based on the study, it is observed that the loads could be experiencing voltage related issues like overvoltage conditions, voltage fluctuations due to high and intermittent power injections from the integrated PV. And it is also demonstrated that with the implementation of the smart inverter control functions, these voltage related issues can be mitigated to ensure safe operating conditions for loads at all times. Since the functionality of each of the control function is different, the purpose of selecting a control function depends on the objective of the study. It should be noted that all the smart inverter control functions studied in this are real power prioritized, meaning that the reactive power management by control functions is established only based on the available reactive power. With reactive power priority, the impact of smart inverters on the voltage profile may have been greater.

Even though the study is conducted based on common inverter settings for PV that are integrated into a cluster at feeder end of the distribution network, the same procedure can be followed for distribution networks with PV distributed over a wide geographical area, to determine the suitable inverter settings to mitigate voltage related issues. The PV distribution should then be zoned based on maximum allowable $\boldsymbol{\Delta} V_{pu\_P}$ and $\boldsymbol{\Delta} V_{pu\_Q}$ for a given region and then determining smart inverter settings that are custom to that particular region in mitigating voltage related issues.

The combined control function is advised for implementation as it effectively utilizes the available reactive power in mitigating voltage fluctuations as well as under & over voltage violations caused due to large PV integration or also due to intermittent nature of the PV irradiance.

Smart PV inverters installed in sufficient quantity at the right location can impact a distribution network voltage profile by providing reactive power support with inverters operating based on chosen control functions without the need for traditional regulatory devices like voltage regulators or capacitor banks.

5.3 FUTURE WORK

With a large fraction of utilities adapting DER and renewable energy generation, there is definitely a scope for a centralized automated system that monitors and controls all the energy sources such as Distributed Energy Resources Management System (DERMS). The present study has focused on mitigating the voltage variations and fluctuations that are caused mainly by the integration of PV and to some level, by the loads in the distribution network. The study can be extended to evaluate the effectiveness of the smart inverters in comparing different loading conditions and mainly in interacting with other voltage regulatory devices like SVCs and voltage regulators.

APPENDIX

VISUAL BASIC CODE

```
----------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit
Type PrimaryTransformer_Type
        SectionId As String
End Type
Public PrimaryTransformers() As PrimaryTransformer_Type
Public NumPrimaryTransformers As Long
----------------------------------------------------------------------------------------------------------
Public Sub Convert ()
        MakeBuscoordsFile
        MakeLoadsFile
        MakeLineCodeFile
        MakeLinesFileDom
        MakeSourceEquiv
        MakeOpenSwitchesFile
        MakeCapacitorFile
        MakePrimaryTransformers
        MakeServiceXfmrsFile
        MakePVsFile
End Sub
----------------------------------------------------------------------------------------------------------
Public Sub MakeBuscoordsFile()
Dim NodeTable As DAO.Recordset
Set NodeTable = CurrentDb.OpenRecordset("Node")
SetPathName
Open MyPathName + "\BusCoords.dss" For Output As #1
NodeTable.MoveFirst
Do While Not NodeTable.EOF
        Write #1, RemoveBlanks(NodeTable!NodeID), NodeTable!x, NodeTable!y
        NodeTable.MoveNext
Loop
Close #1
End Sub
----------------------------------------------------------------------------------------------------------
```

```
Public Sub MakeLoadsFile()

Dim LoadTable As DAO.Recordset
Dim TranSectionID As String, TransPrimaryBus As String, TranSectionID_old As String
Dim Phase As String, strOutput As String, FeederID As String
Dim kVA1 As Double, kVA2 As Double, kVA3 As Double
Dim kW1 As Double, kW2 As Double, kW3 As Double
Dim kvar1 As Double, kvar2 As Double, kvar3 As Double
Dim PF1 As Double, PF2 As Double, PF3 As Double
Dim strOutput1 As String, strOutput2 As String, SecondaryVoltage As Double
Dim NumCust1 As Double, NumCust2 As Double, NumCust3 As Double

Set LoadTable = CurrentDb.OpenRecordset("DSSLoads_v1")
SetPathName
Open MyPathName + "\Loads.dss" For Output As #1

LoadTable.MoveFirst
Do While Not LoadTable.EOF
   strOutput1 = ""
   Phase = RemoveBlanks2(LoadTable!SectionPhases)
   kVA1 = CDbl(LoadTable!phase1Kva)
   kVA2 = CDbl(LoadTable!phase2Kva)
   kVA3 = CDbl(LoadTable!phase3Kva)
   kW1 = CDbl(LoadTable!Phase1Kw)
   kW2 = CDbl(LoadTable!Phase2Kw)
   kW3 = CDbl(LoadTable!Phase3Kw)
   kvar1 = CDbl(LoadTable!Phase1Kvar)
   kvar2 = CDbl(LoadTable!Phase2Kvar)
   kvar3 = CDbl(LoadTable!Phase3Kvar)
   PF1 = CDbl(LoadTable!PF1)
   PF2 = CDbl(LoadTable!PF2)
   PF3 = CDbl(LoadTable!PF3)
   NumCust1 = CDbl(LoadTable!Phase1Customers)
   NumCust2 = CDbl(LoadTable!Phase2Customers)
   NumCust3 = CDbl(LoadTable!Phase3Customers)
   TranSectionID_old = RemoveBlanks(LoadTable!SectionID)
   TranSectionID = Replace(TranSectionID_old, ":", "_")
   TransPrimaryBus = RemoveBlanks(LoadTable!ToNodeID)
```

```
            strOutput1 = ""
            ' If the load is threephase, model it as 3 monophase loads.
If (Phase = "ABCN") And (kW1 = kW2) And (kW2 = kW3) Then
            strOutput1 = strOutput1 + "New Load." + TranSectionID + "_ABC"
            strOutput1 = strOutput1 + " phases=3 bus1=" + TransPrimaryBus + "_load.1.2.3"
+ " kV=0.415 kW=" + Format(dblLarger(0.03, kW1 + kW2 + kW3)) + " kvar=" +
Format(kvar1 + kvar2 + kvar3) + " NumCust=" + Format(dblLarger(1#, NumCust1 +
NumCust2 + NumCust3))
            strOutput1 = strOutput1 + " model=4 cvrwatts=0.8 cvrvars=3 xfkva=" +
Format(kVA1 + kVA2 + kVA3) + " daily=Load_shape status=variable   " & vbCrLf

            ' Model the load as three different monophase loads
Else
            If  kW1 > 0# Then
                        If  kVA1 < 833 Then
                            SecondaryVoltage = 0.24
                        Else
                            SecondaryVoltage = 0.277
                        End If
                        strOutput1 = strOutput1 + "New Load." + TranSectionID + "_A"
                        strOutput1 = strOutput1 + " phases=1 bus1=" + TransPrimaryBus +
"_load.1" + " kV=" + CStr(SecondaryVoltage) + " kW=" + Format(dblLarger(0.01,
kW1)) + " kvar=" + Format(kvar1) + " NumCust=" + Format(NumCust1)
                        strOutput1 = strOutput1 + " model=4 cvrwatts=0.8 cvrvars=3 xfkva=" +
Format(kVA1) + " daily=Load_shape status=variable   " & vbCrLf
            End If
            If  kW2 > 0# Then
                        If  kVA2 < 833 Then
                             SecondaryVoltage = 0.24
                        Else
                             SecondaryVoltage = 0.277
                        End If
                        strOutput1 = strOutput1 + "New Load." + TranSectionID + "_B"
                        strOutput1 = strOutput1 + " phases=1 bus1=" + TransPrimaryBus +
"_load.2" + " kV=" + CStr(SecondaryVoltage) + " kW=" + Format(dblLarger(0.01,
kW2)) + " kvar=" + Format(kvar2) + " NumCust=" + Format(NumCust2)
                        strOutput1 = strOutput1 + " model=4 cvrwatts=0.8 cvrvars=3 xfkva=" +
Format(kVA2) + " daily=Load_shape status=variable   " & vbCrLf
```

```
        End If
        If  kW3 > 0# Then
                If  kVA3 < 833 Then
                    SecondaryVoltage = 0.24
                Else
                    SecondaryVoltage = 0.277
                End If
                strOutput1 = strOutput1 + "New Load." + TranSectionID + "_C"
                strOutput1 = strOutput1 + " phases=1 bus1=" + TransPrimaryBus +
"_load.3" + " kV=" + CStr(SecondaryVoltage) + " kW=" + Format(dblLarger(0.01,
kW3)) + " kvar=" + Format(kvar3) + " NumCust=" + Format(NumCust3)
        strOutput1 = strOutput1 + " model=4 cvrwatts=0.8 cvrvars=3 xfkva=" +
Format(kVA3) + " daily=Load_shape status=variable   " & vbCrLf
        End If
End If

Print #1, strOutput1;
LoadTable.MoveNext
Loop

Close #1
End Sub
------------------------------------------------------------------------------------------------------
Public Sub MakeLineCodeFile()

Dim devConductorsTable As DAO.Recordset
Dim strConductorName, strOutput, StrSecondary, strNeutralConductor As String
Dim StrSec3P_1, StrSec3P_2, StrSec3P_3, StrSec3P_4, StrSec3P_5, StrSec3P_6 As String

Set devConductorsTable = CurrentDb.OpenRecordset("DevConductors")

   SetPathName
   Open MyPathName + "\LineCodes.dss" For Output As #1
   devConductorsTable.MoveFirst

Do While Not devConductorsTable.EOF
   strConductorName =RemoveDots(RemoveBlanks(devConductorsTable!ConductorName))
   strNeutralConductor = RemoveDots(RemoveBlanks(devConductorsTable!NeutralConductorId))
```

```
    strOutput = ""
    strOutput = strOutput & "New LineCode." & strConductorName
    strOutput = strOutput & " R1=" & CStr(devConductorsTable!PosSequenceResistance_PerLUL)
    strOutput = strOutput & " X1=" & CStr(devConductorsTable!PosSequenceReactance_PerLUL)
    strOutput = strOutput & " R0=" & CStr(devConductorsTable!ZeroSequenceResistance_PerLUL)
    strOutput = strOutput & " X0=" & CStr(devConductorsTable!ZeroSequenceReactance_PerLUL)
    strOutput = strOutput & " units=kft baseFreq=60 normamps=" &
CStr(devConductorsTable!ContinuousCurrentRating) & Chr(10) & vbCrLf

    Print #1, strOutput;
    devConductorsTable.MoveNext
Loop

' V1: Add service cables
StrSecondary = "New Linecode.defaultSecondary nphases=1 rmatrix=0.25  xmatrix=0.076
cmatrix=3 units=kft normamps=200" & vbCrLf
Print #1, StrSecondary;
StrSec3P_1 = "New Linecode.defaultSecondary3P nphases=3 rmatrix=[ 0.109716212 0.018049545
0.018049545 |0.018049545 0.109716212 0.018049545 | 0.018049545 0.018049545 0.109716212]"
& vbCrLf
StrSec3P_2 = "!ohms per 1000 ft" & vbCrLf
StrSec3P_3 = "~ xmatrix=[ 0.27765665 0.246417363 0.23845259 | 0.246417363 0.27765665
0.246417363 | 0.23845259 0.246417363 0.27765665]" & vbCrLf
StrSec3P_4 = "~ cmatrix=[15 | -12 15 | -12 -12 15]" & vbCrLf
StrSec3P_5 = "~ Normamps=299  {299 1.25 *}" & vbCrLf
StrSec3P_6 = "~ units=kft" & vbCrLf
Print #1, StrSec3P_1 & StrSec3P_2 & StrSec3P_3 & StrSec3P_4 & StrSec3P_5; StrSec3P_6;

Close #1
End Sub
----------------------------------------------------------------------------------------------------
Public Sub MakeLinesFileDom()

Dim SectTable As DAO.Recordset
Dim SectTable1 As DAO.Recordset
Dim SectID As String, FeederID As String
Dim strPhaseConductor, strNeutralConductor, strSectionPhases As String
Dim strPhaseConductorId, lngNumberPhases, strOutput, strDescription As String
```

```
Dim strphasedesig, strConfigName, ImpedanceReference As String
Dim strConfigurationId, presentConfigandWire, stravgHeight As String
Dim IsUsingActualImpedance, IsUsingEquivalentSpacing As Double
Dim pctR, nphasesLinecod, nphasesLine As Double

Set SectTable = CurrentDb.OpenRecordset("InstSection")
Set SectTable1 = CurrentDb.OpenRecordset("DevConductors")

SetPathName
Open MyPathName + "\Lines.dss" For Output As #1
SectTable.MoveFirst

Do While Not SectTable.EOF
   strOutput = ""
   SectID = Replace(RemoveBlanks(SectTable!SectionID) , ":", "_")
   FeederID = RemoveBlanks(SectTable!FeederID)
   strPhaseConductor = RemoveDots(RemoveBlanks(SectTable!PhaseConductorId))
   strNeutralConductor = RemoveDots(RemoveBlanks(SectTable!NeutralConductorId))
   strSectionPhases = RemoveBlanks2(SectTable!SectionPhases)
   stravgHeight = RemoveDots(CStr(SectTable!AveHeightAboveGround_MUL))
   strDescription = RemoveBlanks(SectTable!Description)
   strPhaseConductorId = RemoveBlanks(SectTable!PhaseConductorId)
   IsUsingActualImpedance = RemoveBlanks(SectTable1!ActualImpedance)
   IsUsingEquivalentSpacing = SectTable!UseEquivSpacing
   lngNumberPhases = Len(strSectionPhases) - 1
   strphasedesig = ""

   If (strSectionPhases = "AN") Then
      strphasedesig = ".1"
      nphasesLinecode = 1
      nphasesLine = 1
   End If
   If (strSectionPhases = "BN") Then
      strphasedesig = ".2"
      nphasesLinecode = 1
      nphasesLine = 1
   End If
   If (strSectionPhases = "CN") Then
```

```
        strphasedesig = ".3"
        nphasesLinecode = 1
        nphasesLine = 1
    End If
    If (strSectionPhases = "ABN") Or (strSectionPhases = "BAN") Then
        strphasedesig = ".1.2"
        nphasesLinecode = 1
        nphasesLine = 2
    End If
    If (strSectionPhases = "ACN") Or (strSectionPhases = "CAN") Then
        strphasedesig = ".1.3"
        nphasesLinecode = 1
        nphasesLine = 2
    End If
    If (strSectionPhases = "BCN") Or (strSectionPhases = "CBN") Then
        strphasedesig = ".2.3"
        nphasesLinecode = 1
        nphasesLine = 2
    End If
    If (strSectionPhases = "ABCN") Then
        strphasedesig = ".1.2.3"
        nphasesLinecode = 3
        nphasesLine = 3
    End If
    If (Len(strphasedesig) = 0) Then
        MsgBox ("Bad move")
    End If

    ImpedanceReference = " linecode=" & strPhaseConductor
    strOutput = "New Line." & SectID
    strOutput = strOutput & " bus1=" & RemoveBlanks(SectTable!FromNodeID) & strphasedesig
    strOutput = strOutput & " bus2=" & RemoveBlanks(SectTable!ToNodeID) & strphasedesig
    strOutput = strOutput & ImpedanceReference
    strOutput = strOutput & " phases=" & CStr(nphasesLine)
    strOutput = strOutput & " length=" & CStr(SectTable!SectionLength_MUL) & " units=ft" &
Chr(10) & vbCrLf

    Print #1, strOutput;
```

```
    SectTable.MoveNext
Loop

Close #1
End Sub
------------------------------------------------------------------------------------------------------
Public Sub MakeSourceEquiv()

Dim SectTable As DAO.Recordset
Dim SectID As String

Set SectTable = CurrentDb.OpenRecordset("InstSubstationTransformers")
SetPathName
Open MyPathName + "Source.DSS" For Output As #1

SectTable.MoveFirst
Do While Not SectTable.EOF
    SectID = RemoveBlanks(Strip69kVtap(SectTable!SubtranID))
    Print #1, "New Line." + SectID;
    Print #1, "   Sourcebus " + RemoveBlanks(SectTable!SubtranID);
    Print #1, "   R1="; SectTable!PosSequenceResistance; SectTable!PosSequenceReactance;
    Print #1, "   R0="; SectTable!ZeroSequenceResistance; SectTable!ZeroSequenceReactance;
    Print #1, "   0  0 Length=1"

    SectTable.MoveNext
Loop

Close #1
End Sub
------------------------------------------------------------------------------------------------------
Public Sub MakeOpenSwitchesFile()

Dim SectTable As DAO.Recordset
Dim SectID As String, SectID_old As String
Dim isOpen As Boolean
Set SectTable = CurrentDb.OpenRecordset("DSSSwitches")
SetPathName
Open MyPathName + "\Switches.dss" For Output As #1
```

```
SectTable.MoveFirst
Do While Not SectTable.EOF
   isOpen = SectTable!SwitchIsOpen
    If isOpen Then
       SectID = Replace(RemoveBlanks(SectTable!SectionID) , ":", "_")
       Print #1, "Edit Line." + SectID;
       If SectTable!NearFromNode Then
         Print #1, " switch=y enabled=false";
         Print #1, " !" + Format(SectTable!FeederID)
       Else
         Print #1, " switch=y enabled=false";
         Print #1, " !" + Format(SectTable!FeederID)
       End If
    Else
       SectID = Replace(RemoveBlanks(SectTable!SectionID) , ":", "_")
       Print #1, "Edit Line." + SectID;
       If SectTable!NearFromNode Then
         Print #1, " switch=y enabled=true";
         Print #1, " !" + Format(SectTable!FeederID)
       Else
         Print #1, " switch=y enabled=true";
         Print #1, " !" + Format(SectTable!FeederID)
       End If
    End If
    SectTable.MoveNext
Loop

Close #1
End Sub
-------------------------------------------------------------------------------------------------------
Public Sub MakeCapacitorFile()

Dim SectTable As DAO.Recordset
Dim SectID As String
Dim isFixed As Boolean
Dim Onsetting As Double, Offsetting As Double
```

```
Set SectTable = CurrentDb.OpenRecordset("DSSCapacitor")
SetPathName
Open MyPathName + "\Capacitors.DSS" For Output As #1

SectTable.MoveFirst
Do While Not SectTable.EOF
    isFixed = SectTable!ManualOperation
    Print #1,
    SectID = Replace(RemoveBlanks(SectTable!SectionID) , ":", "_")
    If SectTable!FixedKvarPhase1 > 0# Then
    Print #1, "New Capacitor." + SectID;
    Print #1, "  Bus1=" + RemoveBlanks(SectTable!FromNodeID);
    Print #1, "  kvar="; SectTable!FixedKvarPhase1 * 3#;
    Print #1, "  kv="; SectTable!RatedKv
    End If
    If Not isFixed Then
        If SectTable!Module1KvarPerPhase > 0 Then
            Print #1, "!--- SWITCHED BANK -----"
            Print #1, "New Capacitor." + SectID + "-1";
            Print #1, "  Bus1=" + RemoveBlanks(SectTable!FromNodeID);
            Print #1, "  kvar="; SectTable!Module1KvarPerPhase * 3#;
            Print #1, "  kv="; SectTable!RatedKv
            Print #1, "New CapControl." + SectID + "-1";
            Print #1, "   Line." + SectID + " 1 ";
            Print #1, SectID + "-1";
            Print #1, "  Type=";
            If SectTable!PrimaryControlMode = "VOLTS" Then Print #1, "Voltage ";
            If SectTable!PrimaryControlMode = "KVAR" Then Print #1, "KVAR ";
            If SectTable!PrimaryControlMode = "AMPS" Then Print #1, "Current ";
            If SectTable!PrimaryControlMode = "TOD" Then Print #1, "Time ";
            Onsetting = SectTable!Module1CapSwitchCloseValue
            Offsetting = SectTable!Module1CapSwitchTripValue
            If SectTable!PrimaryControlMode = "KVAR" Then
                Onsetting = Onsetting * 3#
                Offsetting = Offsetting * 3#
            End If
            Print #1, "  Onsetting="; Onsetting;
            Print #1, "  OFFsetting="; Offsetting;
```

```
      Print #1, "  PTratio="; SectTable!CapacitorPTRatio;
      Print #1, "  CTratio="; SectTable!CapacitorCTRating
    End If
    If SectTable!Module2KvarPerPhase > 0 Then
      Print #1, "!---"
      Print #1, "New Capacitor." + SectID + "-2";
      Print #1, "  Bus1=" + RemoveBlanks(SectTable!FromNodeID);
      Print #1, "  kvar="; SectTable!Module2KvarPerPhase * 3#;
      Print #1, "  kv="; SectTable!RatedKv
      Print #1, "New CapControl." + SectID + "-2";
      Print #1, "   Line." + SectID + " 1 ";
      Print #1, SectID + "-2";
      Print #1, "  Type=";
      If SectTable!PrimaryControlMode = "VOLTS" Then Print #1, "Voltage ";
      If SectTable!PrimaryControlMode = "KVAR" Then Print #1, "KVAR ";
      If SectTable!PrimaryControlMode = "AMPS" Then Print #1, "Current ";
      If SectTable!PrimaryControlMode = "TOD" Then Print #1, "Time ";
      Onsetting = SectTable!Module2CapSwitchCloseValue
      Offsetting = SectTable!Module2CapSwitchTripValue
      If SectTable!PrimaryControlMode = "KVAR" Then
        Onsetting = Onsetting * 3#
        Offsetting = Offsetting * 3#
      End If
      Print #1, "  Onsetting="; Onsetting;
      Print #1, "  OFFsetting="; Offsetting;
      Print #1, "  PTratio="; SectTable!CapacitorPTRatio;
      Print #1, "  CTratio="; SectTable!CapacitorCTRating
    End If
  End If
SectTable.MoveNext
Loop
Close #1
End Sub
------------------------------------------------------------------------------------------------------------------
Public Sub MakePrimaryTransformers()

Dim XfmrTable As DAO.Recordset
Dim TransPrimaryBus, TransSecondaryBus, SectID, Phases, NodeConnection As String
```

```
Dim nphases As Integer
Dim PTRatio As Double
Dim MaxPrimaryTransformers As Long

MaxPrimaryTransformers = 100
ReDim PrimaryTransformers(1 To MaxPrimaryTransformers) As
PrimaryTransformer_Type
NumPrimaryTransformers = 0
Set XfmrTable = CurrentDb.OpenRecordset("DSSTransformers")
SetPathName
Open MyPathName + "\PrimaryTransformers.dss" For Output As #1

XfmrTable.MoveFirst
Do While Not XfmrTable.EOF
    NumPrimaryTransformers = NumPrimaryTransformers + 1
    If NumPrimaryTransformers > MaxPrimaryTransformers Then
        MaxPrimaryTransformers = MaxPrimaryTransformers + 100
        ReDim Preserve PrimaryTransformers(1 To MaxPrimaryTransformers) As
PrimaryTransformer_Type
    End If

    SectID = RemoveBlanks(XfmrTable!SectionID)
    PrimaryTransformers(NumPrimaryTransformers).SectionID = SectID
    TransPrimaryBus = RemoveBlanks(XfmrTable!FromNodeID)
    TransSecondaryBus = RemoveBlanks(XfmrTable!ToNodeID)
    Phases = Replace(XfmrTable!SectionPhases, " ", "")
    If StrComp(Phases, "AN", 1) = 0 Then
        nphases = 1
        NodeConnection = ".1"
    ElseIf StrComp(Phases, "BN", 1) = 0 Then
        nphases = 1
        NodeConnection = ".2"
    ElseIf StrComp(Phases, "CN", 1) = 0 Then
        nphases = 1
        NodeConnection = ".3"
    ElseIf StrComp(Phases, "ABCN", 1) = 0 Then
        nphases = 3
        NodeConnection = ""
```

```
  Else
    MsgBox ("Phases connection unexpected:" + Phases)
  End If
  Print #1, "New Transformer."; SectID; " "; "phases="; nphases; "Windings = 2";
  Print #1, " Buses=["; TransSecondaryBus; NodeConnection; ", "; TransPrimaryBus;
NodeConnection; "]";
  Print #1, " kVs=["; CStr(XfmrTable!HighSideRatedkV); ", ";
CStr(XfmrTable!LowSideRatedkV); "]";
  Print #1, " kVAs=["; CStr(XfmrTable!TransformerRatedKva); ", ";
CStr(XfmrTable!TransformerRatedKva); "]";
  Print #1, " conns=["; XfmrTable!HighSideConnectionCode; ", ";
XfmrTable!HighSideConnectionCode; "]";
  Print #1, " XHL="; CStr(XfmrTable!PercentImpedance); " %LoadLoss= ";
CStr(XfmrTable!PercentResistance)
  PTRatio = XfmrTable!PTRatio
  If PTRatio > 0# Then
    Print #1, "New RegControl.reg_"; SectID; " Transformer="; SectID; " Winding=2
Vreg=";
    Print #1, CStr(XfmrTable!LTCFwdVoltSettingPh1); " Band=";
CStr(XfmrTable!LTCFwdBWSettingPh1);
    Print #1, " R="; CStr(XfmrTable!LTCFwdRDialPhase1); " X=";
CStr(XfmrTable!LTCFwdXDialPhase1);
    Print #1, " PTRatio="; CStr(XfmrTable!PTRatio); " CTPrim="; dblLarger(200,
XfmrTable!CTRating); ' make sure non zero
    Print #1, " Delay="; CStr(XfmrTable!TimeDelaySec); " TapDelay=";
CStr(XfmrTable!TapChangeSec)
    Print #1, " "
  End If

XfmrTable.MoveNext
Loop
Close #1
End Sub
----------------------------------------------------------------------------------------------------------
Public Sub MakeServiceXfmrsFile()

Dim LoadTable As DAO.Recordset
Dim TranSectionID As String, TransPrimaryBus As String, SectionID As String
```

```
Dim strOutput1, strOutput2, Phase, strOutput, FeederID, phased As String
Dim kVA1 As Double, kVA2 As Double, kVA3 As Double
Dim kW1 As Double, kW2 As Double, kW3 As Double
Dim kvar1 As Double, kvar2 As Double, kvar3 As Double
Dim NumCust1 As Double, NumCust2 As Double, NumCust3 As Double
Dim iServiceLine, NServiceCable, randxfmr As Integer
Dim Expected_Phase_Current, primaryLL, primarykV, secondarykV As Double
Dim Connection, Losses As String
Dim AllTrafoSizes(1 To 8) As Double
Dim iTrafoSize, TrafoSize, iTrafo, NParallelTrafo As Double
Dim TotalTrafokVA, TrafoNumbPhases As Double
Dim TrafoModel As Variant

Set LoadTable = CurrentDb.OpenRecordset("DSSLoads_v1")
SetPathName
Open MyPathName + "\xfmrs.dss" For Output As #1
Open MyPathName + "\Services.dss" For Output As #2

LoadTable.MoveFirst
Do While Not LoadTable.EOF
   strOutput1 = ""
   strOutput2 = ""

   Phase = RemoveBlanks2(LoadTable!SectionPhases)

   kVA1 = CDbl(LoadTable!phase1Kva)
   kVA2 = CDbl(LoadTable!phase2Kva)
   kVA3 = CDbl(LoadTable!phase3Kva)
   kW1 = CDbl(LoadTable!Phase1Kw)
   kW2 = CDbl(LoadTable!Phase2Kw)
   kW3 = CDbl(LoadTable!Phase3Kw)
   kvar1 = CDbl(LoadTable!Phase1Kvar)
   kvar2 = CDbl(LoadTable!Phase2Kvar)
   kvar3 = CDbl(LoadTable!Phase3Kvar)
   TotalTrafokVA = CDbl(LoadTable!ConnectedkVA)

   primaryLL = 12
   TranSectionID = Replace(RemoveBlanks(LoadTable!SectionID) , ":", "_")
```

```
    TransPrimaryBus = RemoveBlanks(LoadTable!ToNodeID)
    strOutput2 = ""

    If LoadTable!ConnectedkVA = 0 Then
        Print #1, "!";  'comment out transformers with zero kVA
    End If
    SectionID = Replace(LoadTable!SectionID, ":", "_")

    '---------------- SERVICE CABLE MODELING ----------------------

    ' Threphase load: use a threephase cable
    If (kVA1 > 0) And (kVA2 > 0) And (kVA3 > 0) Then
        Connection = ".1.2.3"
        phased = "3"
        primarykV = primaryLL
        secondarykV = 0.415
        Expected_Phase_Current = (Sqr(kW1 ^ 2 + kvar1 ^ 2) + Sqr(kW2 ^ 2 + kvar2 ^ 2) +
Sqr(kW3 ^ 2 + kvar3 ^ 2)) / secondarykV
        If Expected_Phase_Current < 300 Then
            strOutput2 = strOutput2 + "New Line.service_" + TranSectionID + "_ABC"
            strOutput2 = strOutput2 + " Bus1=" + TransPrimaryBus + "_sec.1.2.3 Bus2=" +
TransPrimaryBus + "_load.1.2.3" + " linecode=defaultSecondary3P phases=3 length=100
units=ft" + Chr(10) + vbCrLf
        Else
            NServiceCable = Round(0.5 + Expected_Phase_Current / 580)
            For iServiceLine = 1 To NServiceCable
                strOutput2 = strOutput2 + "New Line.service_" + TranSectionID + "_ABC" +
"_" + CStr(iServiceLine)
                strOutput2 = strOutput2 + " Bus1=" + TransPrimaryBus + "_sec.1.2.3 Bus2="
+ TransPrimaryBus + "_load.1.2.3" + " linecode=defaultSecondary3P phases=3
length=100 units=ft" + Chr(10) + vbCrLf
            Next

        End If
    ' monophase load: monophase cables
    Else
        If kVA1 > 0# Then
            Connection = ".1"
```

```
            phased = "1"
            primarykV = primaryLL / 1.732
            secondarykV = 0.24
            Expected_Phase_Current = Sqr(kW1 ^ 2 + kvar1 ^ 2) / secondarykV
            If Expected_Phase_Current < 200 Then
              strOutput2 = strOutput2 + "New Line.service_" + TranSectionID + "_A"
              strOutput2 = strOutput2 + " Bus1=" + TransPrimaryBus + "_sec.1 Bus2=" +
TransPrimaryBus + "_load.1" + " linecode=defaultSecondary phases=1 length=100
units=ft" + Chr(10) + vbCrLf
            Else
              NServiceCable = Round(0.5 + Expected_Phase_Current / 200)
              For iServiceLine = 1 To NServiceCable
                 strOutput2 = strOutput2 + "New Line.service_" + TranSectionID + "_A" +
"_" + CStr(iServiceLine)
                 strOutput2 = strOutput2 + " Bus1=" + TransPrimaryBus + "_sec.1 Bus2=" +
TransPrimaryBus + "_load.1" + " linecode=defaultSecondary phases=1 length=100
units=ft" + Chr(10) + vbCrLf
              Next
            End If
        End If

        If kVA2 > 0# Then
            Connection = ".2"
            phased = "1"
            primarykV = primaryLL / 1.732
            secondarykV = 0.24

            Expected_Phase_Current = Sqr(kW2 ^ 2 + kvar2 ^ 2) / secondarykV
            If Expected_Phase_Current < 200 Then
              strOutput2 = strOutput2 + "New Line.service_" + TranSectionID + "_B"
              strOutput2 = strOutput2 + " Bus1=" + TransPrimaryBus + "_sec.2 Bus2=" +
TransPrimaryBus + "_load.2" + " linecode=defaultSecondary phases=1 length=100
units=ft" + Chr(10) + vbCrLf
            Else
              NServiceCable = Round(0.5 + Expected_Phase_Current / 200)
              For iServiceLine = 1 To NServiceCable
                 strOutput2 = strOutput2 + "New Line.service_" + TranSectionID + "_B" +
"_" + CStr(iServiceLine)
```

```
          strOutput2 = strOutput2 + " Bus1=" + TransPrimaryBus + "_sec.2 Bus2=" + TransPrimaryBus + "_load.2" + " linecode=defaultSecondary phases=1 length=100 units=ft" + Chr(10) + vbCrLf
        Next
      End If
    End If

    If kVA3 > 0# Then
      Connection = ".3"
      phased = "1"
      primarykV = primaryLL / 1.732
      secondarykV = 0.24

      Expected_Phase_Current = Sqr(kW3 ^ 2 + kvar3 ^ 2) / secondarykV
      If Expected_Phase_Current < 200 Then
        strOutput2 = strOutput2 + "New Line.service_" + TranSectionID + "_C"
        strOutput2 = strOutput2 + " Bus1=" + TransPrimaryBus + "_sec.3 Bus2=" + TransPrimaryBus + "_load.3" + " linecode=defaultSecondary phases=1 length=100 units=ft" + Chr(10) + vbCrLf
      Else
        NServiceCable = Round(0.5 + Expected_Phase_Current / 200)
        For iServiceLine = 1 To NServiceCable
          strOutput2 = strOutput2 + "New Line.service_" + TranSectionID + "_C" + "_" + CStr(iServiceLine)
          strOutput2 = strOutput2 + " Bus1=" + TransPrimaryBus + "_sec.3 Bus2=" + TransPrimaryBus + "_load.3" + " linecode=defaultSecondary phases=1 length=100 units=ft" + Chr(10) + vbCrLf
        Next
      End If
    End If
  End If
 ' ----------- TRANSFORMER MODELING -------------------------

  If (kVA1 > 0) And (kVA2 > 0) And (kVA3 > 0) Then
    TrafoSize = kVA1
    TrafoModel = SelectXfmrsModel(TrafoNumbPhases, TrafoSize, SectionID)
    Losses = TrafoModel(1)
    TrafoSize = TrafoModel(2)
```

```
        If TrafoSize < 833 Then
            secondarykV = 0.24
        ElseIf TrafoSize >= 833 And TrafoSize < 1000 Then
            secondarykV = 0.277
        ElseIf TrafoSize > 833 Then
            secondarykV = 4.16 / Sqr(3)
            MsgBox "Really high load"
        Else
            MsgBox "Voltage condition unexpected"
        End If

        primarykV = primaryLL / 1.732
        ' THREE monophase transformers
        For iTrafo = 1 To 3
          Print #1, "New Transformer." + SectionID + "_" + Phase + "_" + CStr(iTrafo);
          Print #1, " phases=1" + " wdg=1";
          Print #1, " bus=" + TransPrimaryBus + "." + CStr(iTrafo);
          Print #1, " kv=" + CStr (primarykV);
          Print #1, " kVA=" + CStr (TrafoSize);
          Print #1, " wdg=2";
          Print #1, " bus=" + TransPrimaryBus + "_sec" + "." + CStr (iTrafo);
          Print #1, " kv=" + CStr (secondarykV);
          Print #1, " kVA=" + CStr(TrafoSize);
          Print #1, Losses;
          Print #1, " emerghkva=" + CStr(CDbl(1.4 * Round(LoadTable!ConnectedkVA, 0)))
        Next

   ' If load is monophase the trafo will be monophase
    Else
        primarykV = primaryLL / 1.732
        If kVA1 > 0 Then
            Phase = "AN"
            Connection = ".1"
            TrafoSize = kVA1
            TrafoNumbPhases = 1
            TrafoModel = SelectXfmrsModel(TrafoNumbPhases, TrafoSize, SectionID)
```

```
      Losses = TrafoModel(1)
      TrafoSize = TrafoModel(2)
      secondarykV = 0.24

      Print #1, "New Transformer." + SectionID + "_" + Phase;
      Print #1, " phases=" + CStr(TrafoNumbPhases) + " wdg=1";
      Print #1, " bus=" + TransPrimaryBus + Connection;
      Print #1, " kv=" + CStr(primarykV);
      Print #1, " kVA=" + CStr(TrafoSize);
      Print #1, " wdg=2";
      Print #1, " bus=" + TransPrimaryBus + "_sec" + Connection;
      Print #1, " kv=" + CStr(secondarykV);
      Print #1, " kVA=" + CStr(TrafoSize);
      Print #1, " %loadloss=0.2 %noloadloss=0.1 XHL=0.765";
      Print #1, Losses;
      Print #1, " emerghkva=" + CStr(CDbl(1.4 * Round(LoadTable!ConnectedkVA,
0)))
    End If

    If kVA2 > 0 Then
      Phase = "BN"
      Connection = ".2"
      TrafoSize = kVA2
      TrafoNumbPhases = 1
      TrafoModel = SelectXfmrsModel(TrafoNumbPhases, TrafoSize, SectionID)
      Losses = TrafoModel(1)
      TrafoSize = TrafoModel(2)
      secondarykV = 0.24

      Print #1, "New Transformer." + SectionID + "_" + Phase;
      Print #1, " phases=" + CStr(TrafoNumbPhases) + " wdg=1";
      Print #1, " bus=" + TransPrimaryBus + Connection;
      Print #1, " kv=" + CStr(primarykV);
      Print #1, " kVA=" + CStr(TrafoSize);
      Print #1, " wdg=2";
      Print #1, " bus=" + TransPrimaryBus + "_sec" + Connection;
      Print #1, " kv=" + CStr(secondarykV);
      Print #1, " kVA=" + CStr(TrafoSize);
```

```
            Print #1, " %loadloss=0.2 %noloadloss=0.1 XHL=0.765";
            Print #1, Losses;
            Print #1, " emerghkva=" + CStr(CDbl(1.4 * Round(LoadTable!ConnectedkVA, 0)))
        End If

        If kVA3 > 0 Then
            Phase = "CN"
            Connection = ".3"
            TrafoSize = kVA3
            TrafoNumbPhases = 1
            TrafoModel = SelectXfmrsModel(TrafoNumbPhases, TrafoSize, SectionID)
            Losses = TrafoModel(1)
            TrafoSize = TrafoModel(2)
            secondarykV = 0.24

            Print #1, "New Transformer." + SectionID + "_" + Phase;
            Print #1, " phases=" + CStr(TrafoNumbPhases) + " wdg=1";
            Print #1, " bus=" + TransPrimaryBus + Connection;
            Print #1, " kv=" + CStr(primarykV);
            Print #1, " kVA=" + CStr(TrafoSize);
            Print #1, " wdg=2";
            Print #1, " bus=" + TransPrimaryBus + "_sec" + Connection;
            Print #1, " kv=" + CStr(secondarykV);
            Print #1, " kVA=" + CStr(TrafoSize);
            Print #1, " %loadloss=0.2 %noloadloss=0.1 XHL=0.765";
            Print #1, Losses;
            Print #1, " emerghkva=" + CStr(CDbl(1.4 * Round(LoadTable!ConnectedkVA, 0)))
        End If
    End If

     Print #2, strOutput2;
     LoadTable.MoveNext
 Loop
    Close #1
    Close #2
```

```
   Set LoadTable = Nothing
   ' V1: advice user on the nominal voltage value adopted
   MsgBox ("WARNING ASSUMPTION: Primary side voltage adopted=" + CStr(primaryLL) + " kV (L-L)")

End Sub
------------------------------------------------------------------------------------------------
Public Sub MakePVsFile()

Dim LoadTable As DAO.Recordset
Dim TranSectionID As String, TransPrimaryBus As String, TranSectionID_old As String
Dim kW1 As Double, kW2 As Double, kW3 As Double
Dim Phase, Connection, phased, SectionID_old, SectionID As String
Dim strOutput, strOutput1, strOutput2, strOutput3, strOutput4 As String
Dim randxfmr, NServiceCable, iServiceLine As Integer
Dim primaryLL, primarykV, secondarykV, Losses As Double
Dim Expected_Phase_Current As Double
Dim AllTrafoSizes(1 To 8) As Double
Dim iTrafoSize, TrafoSize, iTrafo, NParallelTrafo As Double
Dim TrafoNumbPhases As Double , TotalTrafokVA As Double
Dim TrafoModel As Variant

Set LoadTable = CurrentDb.OpenRecordset("InstGenerators")

SetPathName
Open MyPathName + "\PVs.dss" For Output As #1
LoadTable.MoveFirst

Do While Not LoadTable.EOF
  strOutput1 = ""
  strOutput2 = ""
  strOutput3 = ""
  strOutput4 = ""

   Phase = RemoveBlanks2(LoadTable!ConnectedPhases)
   kW1 = CStr(0.45 * Format(CDbl(LoadTable!GenPhase1Kw)))
   kW2 = CStr(0.45 * Format(CDbl(LoadTable!GenPhase2Kw)))
   kW3 = CStr(0.45 * Format(CDbl(LoadTable!GenPhase3Kw)))
```

```
    primaryLL = 12
    primarykV = primaryLL / 1.732
    TranSectionID = Replace(RemoveBlanks(LoadTable!SectionID) , ":", "_")
    TransPrimaryBus = RemoveBlanks(LoadTable!ToNodeID)
    SectionID = Replace(LoadTable!SectionID, ":", "_")

strOutput1 = ""
    ' If the load is threephase model it as 3 monophase load
    If kW1 > 0 Then
            Phase = "AN"
            Connection = ".1"
            TrafoSize = kW1
            TrafoNumbPhases = 1
            TrafoModel = SelectXfmrsModel(TrafoNumbPhases, TrafoSize, SectionID)
            Losses = TrafoModel(1)
            TrafoSize = TrafoModel(2)
            secondarykV = 0.24
            strOutput1 = strOutput1 + "New Transformer." + TranSectionID + "_xfmr_A"
            strOutput1 = strOutput1 + " phases=" + CStr(TrafoNumbPhases) + " wdg=1"
            strOutput1 = strOutput1 + " bus=" + TransPrimaryBus + Connection
            strOutput1 = strOutput1 + " kv=" + CStr(primarykV)
            strOutput1 = strOutput1 + " kVA=" + CStr(TrafoSize)
            strOutput1 = strOutput1 + " wdg=2"
            strOutput1 = strOutput1 + " bus=" + TransPrimaryBus + "_secPV" + Connection
            strOutput1 = strOutput1 + " kv=" + CStr(secondarykV)
            strOutput1 = strOutput1 + " kVA=" + CStr(TrafoSize)
            strOutput1 = strOutput1 + Losses
            strOutput1 = strOutput1 + " emerghkva=" + CStr(CDbl(1.4 * Round(kW1, 0))) & vbCrLf

            strOutput2 = strOutput2 + "makebuslist" & vbCrLf

            strOutput3 = strOutput3 + "setkvbase Bus=" + TransPrimaryBus + "_secPV kVLN=0.240" & vbCrLf

            strOutput4 = strOutput4 + "New PVSystem." + TranSectionID + "_PV"
```

```
        strOutput4 = strOutput4 + " phases=1 bus1=" + TransPrimaryBus + "_secPV.1
kV=0.24 kVA=" + CStr(1.1 * Format(kW1)) + " pf=1 irradiance=1 Temperature=25
Pmpp=" + Format(kW1) + " %cutin=0.1 %cutout=0.1 daily=PVshape " & vbCrLf &
vbCrLf
    End If
    If kW2 > 0 Then
        Phase = "BN"
        Connection = ".2"
        TrafoSize = kW2
        TrafoNumbPhases = 1
        TrafoModel = SelectXfmrsModel(TrafoNumbPhases, TrafoSize, SectionID)
        Losses = TrafoModel(1)
        TrafoSize = TrafoModel(2)
        secondarykV = 0.24
        strOutput1 = strOutput1 + "New Transformer." + TranSectionID + "_xfmr_B"
        strOutput1 = strOutput1 + " phases=" + CStr(TrafoNumbPhases) + " wdg=1"
        strOutput1 = strOutput1 + " bus=" + TransPrimaryBus + Connection
        strOutput1 = strOutput1 + " kv=" + CStr(primarykV)
        strOutput1 = strOutput1 + " kVA=" + CStr(TrafoSize)
        strOutput1 = strOutput1 + " wdg=2"
        strOutput1 = strOutput1 + " bus=" + TransPrimaryBus + "_secPV" + Connection
        strOutput1 = strOutput1 + " kv=" + CStr(secondarykV)
        strOutput1 = strOutput1 + " kVA=" + CStr(TrafoSize)
        strOutput1 = strOutput1 + Losses
        strOutput1 = strOutput1 + " emerghkva=" + CStr(CDbl(1.4 * Round(kW2, 0))) &
vbCrLf

        strOutput2 = strOutput2 + "makebuslist" & vbCrLf

        strOutput3 = strOutput3 + "setkvbase Bus=" + TransPrimaryBus + "_secPV
kVLN=0.240" & vbCrLf

        strOutput4 = strOutput4 + "New PVSystem." + TranSectionID + "_PV"
        strOutput4 = strOutput4 + " phases=1 bus1=" + TransPrimaryBus + "_secPV.2
kV=0.24 kVA=" + CStr(1.1 * Format(kW2)) + " pf=1 irradiance=1 Temperature=25
Pmpp=" + Format(kW2) + " %cutin=0.1 %cutout=0.1 daily=PVshape " & vbCrLf &
vbCrLf
    End If
```

```
  If kW3 > 0 Then
        Phase = "CN"
        Connection = ".3"
        TrafoSize = kW3
        TrafoNumbPhases = 1
        TrafoModel = SelectXfmrsModel(TrafoNumbPhases, TrafoSize, SectionID)
        Losses = TrafoModel(1)
        TrafoSize = TrafoModel(2)
        secondarykV = 0.24
        strOutput1 = strOutput1 + "New Transformer." + TranSectionID + "_xfmr_C"
        strOutput1 = strOutput1 + " phases=" + CStr(TrafoNumbPhases) + " wdg=1"
        strOutput1 = strOutput1 + " bus=" + TransPrimaryBus + Connection
        strOutput1 = strOutput1 + " kv=" + CStr(primarykV)
        strOutput1 = strOutput1 + " kVA=" + CStr(TrafoSize)
        strOutput1 = strOutput1 + " wdg=2"
        strOutput1 = strOutput1 + " bus=" + TransPrimaryBus + "_secPV" + Connection
        strOutput1 = strOutput1 + " kv=" + CStr(secondarykV)
        strOutput1 = strOutput1 + " kVA=" + CStr(TrafoSize)
        strOutput1 = strOutput1 + Losses
        strOutput1 = strOutput1 + " emerghkva=" + CStr(CDbl(1.4 * Round(kW3, 0))) &
vbCrLf

        strOutput2 = strOutput2 + "makebuslist" & vbCrLf

        strOutput3 = strOutput3 + "setkvbase Bus=" + TransPrimaryBus + "_secPV
kVLN=0.240" & vbCrLf

        strOutput4 = strOutput4 + "New PVSystem." + TranSectionID + "_PV"
        strOutput4 = strOutput4 + " phases=1 bus1=" + TransPrimaryBus + "_secPV.3
kV=0.24 kVA=" + CStr(1.1 * Format(kW3)) + " pf=1 irradiance=1 Temperature=25
Pmpp=" + Format(kW3) + " %cutin=0.1 %cutout=0.1 daily=PVshape " & vbCrLf &
vbCrLf
  End If

  Print #1, strOutput1;
  Print #1, strOutput2;
  Print #1, strOutput3;
```

```
   Print #1, strOutput4;

 LoadTable.MoveNext
 Loop
 'MsgBox "WARNING ASSUMPTION: Default load model assumed: Z=0.3 I=0.45 and P=0.25 "
Close #1
Close #2
   Set LoadTable = Nothing
   ' V1: advice user on the nominal voltage value adopted
   MsgBox ("WARNING ASSUMPTION: Primary side voltage adopted=" + CStr(primaryLL) + " kV (L-L)")

End Sub
----------------------------------------------------------------------------------------------------
Function SelectXfmrsModel(TrafoNumbPhases As Double, TrafoSize As Double, SectionID As String) As Variant

Dim AllTrafoSizes_mono(14) As Double, AllTrafoSizes_Tri(9) As Double
Dim mismatch_tri As Double, mismatch_mono As Double, randxfmr As Double
Dim Losses As String
Dim Results(1 To 2) As Variant
Dim iSize As Integer

' Monophase transformer model size
AllTrafoSizes_mono(1) = 10
AllTrafoSizes_mono(2) = 15
AllTrafoSizes_mono(3) = 25
AllTrafoSizes_mono(4) = 33
AllTrafoSizes_mono(5) = 37.5
AllTrafoSizes_mono(6) = 38
AllTrafoSizes_mono(7) = 50
AllTrafoSizes_mono(8) = 75
AllTrafoSizes_mono(9) = 100
AllTrafoSizes_mono(10) = 167
AllTrafoSizes_mono(11) = 250
AllTrafoSizes_mono(12) = 333
AllTrafoSizes_mono(13) = 500
```

```
AllTrafoSizes_mono(14) = 833

' Threephase transformer modelled size
AllTrafoSizes_Tri(1) = 75
AllTrafoSizes_Tri(2) = 150
AllTrafoSizes_Tri(3) = 225
AllTrafoSizes_Tri(4) = 300
AllTrafoSizes_Tri(5) = 500
AllTrafoSizes_Tri(6) = 750
AllTrafoSizes_Tri(7) = 1000
AllTrafoSizes_Tri(8) = 1500
AllTrafoSizes_Tri(9) = 2000

' If the trafo is THREEPHASE: check if the size is nominal
    If TrafoNumbPhases = 3 Then
        For iSize = 1 To 9
            mismatch_tri = (TrafoSize - AllTrafoSizes_Tri(iSize))
            ' If the transformer size is among the standard ones
            If mismatch_tri = 0 Then
                Exit For
            End If
        Next iSize

        ' If the trafo does not have a standard size: assing a standard size and advice
        If mismatch_tri <> 0 Then
            MsgBox "WARNING ASSUMPTION: The transformer " + SectionID + " has not
standard size of " + CStr(TrafoSize) + " kVA"

            For iSize = 1 To 9
                mismatch_tri = (TrafoSize - AllTrafoSizes_Tri(iSize))
                If mismatch_tri < 0 Then
                    Exit For
                End If
            Next iSize
            TrafoSize = CDbl(AllTrafoSizes_Tri(iSize))
            MsgBox "The adopted size is: " + CStr(TrafoSize) + " kVA"
        End If
```

```
    ' If the transformer is MONOPHASE: check if the size is nominal
    Else
        ' Check if there is a trafo with the same size
        For iSize = 1 To 14
            mismatch_mono = (TrafoSize - AllTrafoSizes_mono(iSize))
            If mismatch_mono = 0 Then
                Exit For
            End If
        Next iSize

        ' If the trafo does not have a standard size: check the closest one
        If mismatch_mono <> 0 Then
            MsgBox "WARNING ASSUMPTION: The transformer " + SectionID + " has not
standard size of " + CStr(TrafoSize) + " kVA"
            For iSize = 1 To 14
                mismatch_mono = (TrafoSize - AllTrafoSizes_mono(iSize))
                If mismatch_mono < 0 Then
                    Exit For
                End If
            Next iSize

            'Assign the immediately larger size
            TrafoSize = CDbl(AllTrafoSizes_mono(iSize))
            MsgBox "The adopted size is: " + CStr(TrafoSize) + " kVA"
        End If
    End If

    Randomize
    randxfmr = Int((7 * Rnd))
    If randxfmr > 3 Then

        ' Choose the size for the threephase
        If TrafoNumbPhases = 1 Then

"Typical_Transformer_Data_USA" 1986-87
            Select Case TrafoSize
                Case 10
                Losses = " %loadloss=1.022   %noloadloss=0.322   XHL=1.707"
```

```
        Case 15
        Losses = " %loadloss=0.939   %noloadloss=0.266   XHL=1.685"
        Case 25
        Losses = " %loadloss=0.858   %noloadloss=0.261   XHL=1.751"
        Case 37.5, 33, 38
        Losses = " %loadloss=0.722   %noloadloss=0.242   XHL=1.516"
        Case 50
        Losses = " %loadloss=0.738   %noloadloss=0.199   XHL=1.753"
        Case 75
        Losses = " %loadloss=0.666   %noloadloss=0.196   XHL=2.002"
        Case 100
        Losses = " %loadloss=0.724   %noloadloss=0.152   XHL=2.156"
        Case 167
        Losses = " %loadloss=0.595   %noloadloss=0.158   XHL=2.199"
        ' From 1988
        Case 250, 333
        Losses = " %loadloss=0.599   %noloadloss=0.129   XHL=1.885"
        ' From 1989
        Case 500, 833
        Losses = " %loadloss=0.536   %noloadloss=0.120   XHL=2.385"
      End Select
    ElseIf TrafoNumbPhases = 3 Then

"Typical_Transformer_Data_USA" 1986-87
      ' Three phase transformer
      Select Case TrafoSize
        Case 75
        Losses = " %loadloss=0.898   %noloadloss=0.249   XHL=3.122"
        Case 150
        Losses = " %loadloss=0.687   %noloadloss=0.216   XHL=3.074"
        Case 225
        Losses = " %loadloss=0.653   %noloadloss=0.182   XHL=4.203"
        Case 300
        Losses = " %loadloss=0.689   %noloadloss=0.163   XHL=3.735"
        Case 500
        Losses = " %loadloss=0.572   %noloadloss=0.157   XHL=4.442"
        Case 750
        Losses = " %loadloss=0.673   %noloadloss=0.115   XHL=5.489"
```

```
            Case 1000
            Losses = " %loadloss=0.567   %noloadloss=0.121   XHL=5.731"
            Case 1500
            Losses = " %loadloss=0.557   %noloadloss=0.121   XHL=5.689"
            Case 2000
            Losses = " %loadloss=0.507   %noloadloss=0.115   XHL=5.429"
        End Select
    End If

' If the random number is larger than 3
Else
    ' Choose the size for monophase transformer
    If TrafoNumbPhases = 1 Then
"Typical_Transformer_Data_USA" 1986-2006
        ' Monophase transformer data
        Select Case TrafoSize
            Case 10
            Losses = " %loadloss=1.153 %noloadloss=.318 XHL=1.888"
            Case 25
            Losses = " %loadloss=1.008 %noloadloss=.230 XHL=1.803"
            Case 37.5, 33, 38
            Losses = " %loadloss=0.831 %noloadloss=.219 XHL=1.686"
            Case 50
            Losses = " %loadloss=0.811 %noloadloss=.184 XHL=1.765"
            Case 75
            Losses = " %loadloss=0.749 %noloadloss=.171 XHL=1.773"
            Case 100
            Losses = " %loadloss=0.715 %noloadloss=.173 XHL=1.845"
            Case 150
            Losses = " %loadloss=0.7268 %noloadloss=.1559 XHL=2.828"
            Case 167
            Losses = " %loadloss=0.633   %noloadloss=0.156   XHL=2.219"
            Case 250, 333
            Losses = " %loadloss=0.642   %noloadloss=0.142   XHL=2.211"
            Case 500, 833
            Losses = " %loadloss=0.582   %noloadloss=0.127   XHL=2.570"
        End Select
    ' If it is threephase
```

```
    Else
      Select Case TrafoSize
        Case 75
        Losses = " %loadloss=0.935   %noloadloss=0.259   XHL=2.183"
        Case 150
        Losses = " %loadloss=0.753   %noloadloss=0.176   XHL=2.867"
        Case 225
        Losses = " %loadloss=0.668   %noloadloss=0.157   XHL=2.816"
        Case 300
        Losses = " %loadloss=0.664   %noloadloss=0.139   XHL=3.053"
        Case 500
        Losses = " %loadloss=0.607   %noloadloss=0.128   XHL=3.501"
        Case 750
        Losses = " %loadloss=0.629   %noloadloss=0.109   XHL=5.737"
        Case 1000
        Losses = " %loadloss=0.594   %noloadloss=0.097   XHL=5.719"
        Case 1500
        Losses = " %loadloss=0.537   %noloadloss=0.111   XHL=5.623"
        Case 2000
        Losses = " %loadloss=0.536   %noloadloss=0.096   XHL=5.476"
        Case 2500
        Losses = " %loadloss=0.528   %noloadloss=0.079   XHL=5.694"
        Case 3000
        Losses = " %loadloss=0.541   %noloadloss=0.090   XHL=5.722"
      End Select
    End If
End If
Results(1) = Losses
Results(2) = TrafoSize
SelectXfmrsModel = Results

End Function
----------------------------------------------------------------------------------------------------
```

OPENDSS FILES

Code Snippet for Buscoords.dss

```
"701","1598160.8077","11900809.4823"
"BP_701_1573569","1597324.201084","11909776.3183945"
"BP_701_1573570","1597324.48881309","11909777.009338"
"BP_701_1573571","1597413.65333292","11909361.892058"
"BP_701_1588610","1598473.82542517","11902026.1928187"
"BP_701_1588611","1598477.36970942","11902026.202005"
"BP_701_1588612","1598476.18467242","11902026.1997084"
"BP_701_1588613","1598472.647606","11902026.1954434"
"BP_701_1588614","1598475.00554092","11902026.198068"
"BP_701_1700305","1597021.1288555","11910009.4514584"
...
```

Code Snippet for Loads.dss

```
New Load.701_101237_OH_B phases=1 bus1=DJ_383512_1816_load.2
kV=0.24 kW=4.975469023603 kvar=1.60244252764682 NumCust=2.5
model=4 cvrwatts=0.8 cvrvars=3 xfkva=12.5  daily=Load_shape
status=variable
New Load.701_101237_OH_C phases=1 bus1=DJ_383512_1816_load.3
kV=0.24 kW=4.85698965776036 kvar=1.59519768712842 NumCust=2.5
model=4 cvrwatts=0.8 cvrvars=3 xfkva=12.5  daily=Load_shape
status=variable
New Load.701_101463_OH_B phases=1 bus1=DJ_378668_1816_load.2
kV=0.24 kW=13.3389058044761 kvar=4.2960432136082 NumCust=7
model=4 cvrwatts=0.8 cvrvars=3 xfkva=12.5  daily=Load_shape
status=variable
...
```

Code Snippet for Linecodes.dss

```
New LineCode.4_0_PECN_CU R1=0.066 X1=0.041 R0=0.236 X0=0.068
units=kft baseFreq=60 normamps=275
New LineCode.4_0_PILC R1=0.066 X1=0.041 R0=0.236 X0=0.068
units=kft baseFreq=60 normamps=250
New LineCode.4_0_PILC_CU R1=0.0001 X1=0.0001 R0=0.0001 X0=0.0001
units=kft baseFreq=60 normamps=250
...
```

Code Snippet for Lines.dss

```
New Line.701_101802_UG bus1=BP_701_207260.2 bus2=BP_701_204446.2
linecode=#2_XLPECN_PEJ_AL_(1_ph) phases=1 length=185 units=ft
New Line.701_101806_OH bus1=DJ_373206_1816.1.2.3
bus2=DJ_373211_1816.1.2.3 linecode=636_ACSR phases=3 length=284
units=ft
New Line.701_101807_OH bus1=DJ_383484_1816.2.3
bus2=DJ_383486_1816.2.3 linecode=#4_B_STRD phases=2 length=122
units=ft
...
```

Code Snippet for Master.dss

The code snippet of an example Master.dss file has already been discussed in chapter 2 of this report.

Code Snippet for Switches.dss

```
Edit Line.701_100191_UG switch=y enabled=false !701
Edit Line.701_1063457_BR switch=y enabled=true !701
Edit Line.701_1063458_BR switch=y enabled=true !701
...
```

Code Snippet for Capacitors.dss

```
New Capacitor.701_70799_MC  Bus1=CF_701_311  kvar= 1200   kv= 12
! 701-104CF
New Capacitor.701_57838_MC  Bus1=CW_701_347  kvar= 1200   kv= 12
! 701-319CM
New CapControl.701_57838_MC   element=Line.701_57838_MC
Capacitor=701_57838_MC  terminal=2 Type=Voltage   Onsetting= 119
OFFsetting= 125   PTratio= 57.8   CTratio= 100
...
```

Code Snippet for PrimaryTransformers.dss

```
New Transformer.701_71563_MC phases= 3 Windings = 2
Buses=[BP_701_84998, UD_701_48083] kVs=[12, 4.16] kVAs=[300, 300]
conns=[YG, YG] XHL=5 %LoadLoss= 0.8
...
```

Code Snippet for xfmrs.dss

```
New Transformer.701_101237_OH_BN phases=1 wdg=1
bus=DJ_383512_1816.2 kv=6.9284064665127 kVA=15 wdg=2
```

```
bus=DJ_383512_1816_sec.2 kv=0.24 kVA=15 %loadloss=0.939
%noloadloss=0.266   XHL=1.685 emerghkva=35
New Transformer.701_101237_OH_CN phases=1 wdg=1
bus=DJ_383512_1816.3 kv=6.9284064665127 kVA=15 wdg=2
bus=DJ_383512_1816_sec.3 kv=0.24 kVA=15 emerghkva=35
New Transformer.701_101463_OH_BN phases=1 wdg=1
bus=DJ_378668_1816.2 kv=6.9284064665127 kVA=15 wdg=2
bus=DJ_378668_1816_sec.2 kv=0.24 kVA=15 emerghkva=35
...
```

Code Snippet for Services.dss

```
New Line.service_701_101237_OH_B Bus1=DJ_383512_1816_sec.2
Bus2=DJ_383512_1816_load.2 linecode=defaultSecondary phases=1
length=100 units=ft

New Line.service_701_101237_OH_C Bus1=DJ_383512_1816_sec.3
Bus2=DJ_383512_1816_load.3 linecode=defaultSecondary phases=1
length=100 units=ft

New Line.service_701_101463_OH_B Bus1=DJ_378668_1816_sec.2
Bus2=DJ_378668_1816_load.2 linecode=defaultSecondary phases=1
length=100 units=ft
...
```

Code Snippet for PVs.dss

```
New Transformer.701_2626414_MC_xfmr_B phases=1 windings=2
buses=(CG_701_826457.2,CG_701_826457_secPV.2)  kVs=(6.9282,0.240)
kVAs=(65,65) %loadloss=0.7639   %noloadloss=.0851   XHL=2.1
emerghkva=86.8
makebuslist
setkvbase Bus=CG_701_826457_secPV.2 kVLN=0.240
New PVSystem.701_2626414_MC_PV phases=1
bus1=CG_701_826457_secPV.2 kV=0.240 kVA=56.779 pf=1 irradiance=1
Temperature=25 Pmpp=51.617 %cutin=0.1 %cutout=0.1 daily=PVshape

New Transformer.701_2626415_MC_xfmr_B phases=1 windings=2
buses=(CG_701_826458.2,CG_701_826458_secPV.2)  kVs=(6.9282,0.240)
kVAs=(75,75) %loadloss=0.749   %noloadloss=.171   XHL=1.773
emerghkva=105
makebuslist
setkvbase Bus=CG_701_826458_secPV.2 kVLN=0.240
New PVSystem.701_2626415_MC_PV phases=1
bus1=CG_701_826458_secPV.2 kV=0.240 kVA=68.581 pf=1 irradiance=1
Temperature=25 Pmpp=62.346 %cutin=0.1 %cutout=0.1 daily=PVshape
...
```

REFERENCES

[1] Rule 21 of California Public Utilities Commission (CPUC) – http://www.cpuc.ca.gov/Rule21/

[2] Understanding what California Rule 21 means for solar inverters – http://www.solarpowerworldonline.com/2014/10/understanding-california-rule-21-means-solar-inverters/

[3] Solar Industry Data – http://www.seia.org/research-resources/solar-industry-data

[4] OpenDSS – https://sourceforge.net/projects/electricdss/

[5] Synergi Electric – Distribution planning and analysis for electric grid – https://www.dnvgl.com/services/power-distribution-system-and-electrical-simulation-software-synergi-electric-5005

[6] Synergi Electric brochure

[7] VBA – Visual Basic for Applications

[8] EPRI Test circuits - http://smartgrid.epri.com/SimulationTool.aspx.

[9] John Seuss, Matthew J. Reno, Robert J. Broderick, "Improving distribution network PV hosting capacity via smart inverter reactive power support", in 2015 IEEE Power & Energy Society General Meeting, pp.1-5, July 2015.

[10] Matthew Rylander, Matthew J. Reno, Jimmy E. Quiroz, Fei Ding, Huijuan Li, Robert J. Broderick, Barry Mather, Jeff Smith, "Methods to determine recommended feeder-wide advanced inverter settings for improving distribution system performance", in 2016 IEEE 43rd Photovoltaics Specialists Conference (PVSC), pp. 1393-1398, June 2016.

[11] Matthew Rylander, Huijuan Li, Jeff Smith, Wes Sunderman, "Default volt-var inverter settings to improve distribution system performance", in 2016 IEEE Power and Energy Society General Meeting (PESGM), pp. 1-5, July 2016.

[12] Huijuan Li, Matthew Rylander, Jeff Smith, "Analysis to Inform CA Grid Integration: Methods and Default Settings to Effectively Use Advanced Inverter Functions in the Distribution System", EPRI, Palo Alto, CA: 2015.

[13] William H. Kersting, "Distribution System Modeling and Analysis", CRC Press LLC, 2002.

[14] B. Seal, "Common functions for smart inverters – Version 3", February 2014.

[15] Wes Sunderman, Roger Dugan, Jeff Smith, B. Seal, "Modelling High-Penetration PV for Distribution Interconnection Studies – Smart Inverter Modeling in OpenDSS, Rev.2", October 2013.

[16] Joshua S. Stein, Clifford W. Hansen, Matthew J. Reno, “The variability index: A new and novel metric for quantifying irradiance and PV output variability”, World Renewable Energy Forum, 2012.